



**SISTEM INFORMASI PENJUALAN PADA “GROSIR
FASHION ONLINE” DENGAN SARAN PEMBELIAN PAKET
PRODUK MENGGUNAKAN ALGORITMA APRIORI**

SKRIPSI

oleh:

Mita Adindayu

112410101008

PROGRAM STUDI SISTEM INFORMASI

UNIVERSITAS JEMBER

2015



**SISTEM INFORMASI PENJUALAN PADA “GROSIR
FASHION ONLINE” DENGAN SARAN PEMBELIAN PAKET
PRODUK MENGGUNAKAN ALGORITMA APRIORI**

SKRIPSI

oleh:

Mita Adindayu

112410101008

PROGRAM STUDI SISTEM INFORMASI

UNIVERSITAS JEMBER

2015

PERSEMBAHAN

Skripsi ini saya persembahkan untuk :

1. Allah SWT, yang telah memberikan kelancaran dan kemudahan dalam menyelesaikan skripsi ini;
2. Ayahanda Taufiq dan Ibunda tercinta Mamiiek;
3. Kakak tersayang Mita Anandayu;
4. Saudara-saudaraku beserta seluruh keluarga besar;
5. Sahabatku bersama doa dan bantuannya;
6. Guru-guruku sejak taman kanak-kanak hingga perguruan tinggi;
7. Almamater Program Studi Sistem Informasi Universitas Jember.

MOTO

“Allah akan meninggikan orang-orang yang beriman di antara kamu dan orang-orang yang diberi ilmu pengetahuan beberapa derajat”. (*QS. Al-Mujadalah : 11*)¹



¹ Departemen Agama Republik Indonesia. 1998. *Al Qur'an dan Terjemahannya*. Semarang: PT Kumudasmoro Grafindo

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Mita Adindayu

NIM : 112410101008

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul Sistem Informasi Penjualan Pada “Grosir Fashion Online” dengan Saran Pembelian Paket Produk Menggunakan Algoritma Apriori adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika di kemudian hari pernyataan ini tidak benar.

Jember, 7 September 2015

Yang menyatakan,

Mita Adindayu

NIM 112410101008

PENGESAHAN PEMBIMBING

Skripsi berjudul Sistem Informasi Penjualan Pada “Grosir Fashion Online” dengan Saran Pembelian Paket Produk Menggunakan Algoritma Apriori, telah diuji dan disahkan pada:

hari, tanggal : Kamis, 17 September 2015

tempat : Program Studi Sistem Informasi Universitas Jember.

Disetujui oleh :

Pembimbing I,

Pembimbing II

Drs. Antonius Cahya Prihandoko, M.App.Sc

NIP 196909281993021001

Muhamad Arief Hidayat S.Kom, M.Kom.

NIP 198101232010121003

SKRIPSI

**SISTEM INFORMASI PENJUALAN PADA “GROSIR FASHION ONLINE”
DENGAN SARAN PEMBELIAN PAKET PRODUK MENGGUNAKAN
ALGORITMA APRIORI**

oleh:

Mita Adindayu

NIM 112410101008

Pembimbing:

Dosen Pembimbing Utama : Drs. Antonius Cahya Prihandoko, M.App.Sc
Dosen Pembimbing Pendamping : Muhamad Arief Hidayat S.Kom, M.Kom.

PENGESAHAN

Skripsi berjudul Sistem Informasi Penjualan Pada “Grosir Fashion Online” dengan Saran Pembelian Paket Produk Menggunakan Algoritma Apriori, telah diuji dan disahkan pada:

hari, tanggal : Kamis, 17 September 2015

tempat : Program Studi Sistem Informasi Universitas Jember.

Tim Penguji :

Penguji I,

Penguji II,

Dr. Saiful Bukhori ST., M.Kom
NIP 196811131994121001

Anang Andrianto ST., MT
NIP 196906151997021002

Mengesahkan
Ketua Program Studi,

Prof. Drs. Slamini, M.Comp.Sc.,Ph.D
NIP 196704201992011001

RINGKASAN

Sistem Informasi Penjualan Pada “Grosir Fashion Online” dengan Saran Pembelian Paket Produk Menggunakan Algoritma Apriori; Mita Adindayu; 112410101008; 354 halaman; Program Studi Sistem Informasi Universitas Jember.

Bisnis online shop yang semakin berkembang seiring berkembangnya kemajuan teknologi membuat daya saing antar online shop semakin tinggi pula. Grosir Fashion Online (GFO) merupakan perusahaan bisnis *online* yang menawarkan berbagai produk aksesoris *import* Korea dengan harga lebih terjangkau dibandingkan dengan *online shop* lainnya.

Untuk menghadapi persaingan yang semakin ketat, GFO memiliki strategi agar konsumen tidak beralih ke online shop lainnya, yaitu dengan memberikan harga grosir tanpa syarat dan diskon berdasarkan *level member* bagi konsumen loyal. Salah satu strategi promosi lain yang masih belum diterapkan oleh GFO yaitu dengan mengelompokkan beberapa produk menjadi satu paket dan menawarkannya dengan harga yang lebih rendah. Cara ini pernah dilakukan GFO namun hanya pada saat-saat tertentu, sebagai contoh semisal pada saat perayaan Hari Raya Imlek, GFO memberikan diskon lebih tinggi jika dalam transaksi pembeli terdapat produk yang telah ditentukan. Sesuai dengan tujuannya, pemaketan produk akan sangat memberikan banyak keuntungan bagi konsumen begitu pula bagi GFO, maka akan lebih baik jika pemaketan produk juga diterapkan.

Banyaknya transaksi yang masuk akan sangat bermanfaat jika dapat membantu GFO untuk menentukan paket produk, namun tidak memungkinkan jika *admin* GFO secara manual meneliti satu persatu transaksi tersebut. Untuk itulah dibutuhkan suatu sistem yang dapat memudahkan pemilihan paket produk.

Pada penelitian ini dibangun Sistem Informasi Penjualan dengan tambahan fitur rekomendasi paket produk yang menggunakan transaksi konsumen sebagai kriterianya. Metode yang digunakan yaitu metode *market basket analysis* dengan algoritma

Apriori. Pada proses penentuan ini dibutuhkan nilai minimum *support* dan minimum *confidence* untuk melihat seberapa sering produk tersebut dibeli dalam satu kali transaksi sebelum aturan asosiasi final dapat ditentukan.

Penelitian ini dilaksanakan dalam tiga tahap yaitu pengumpulan data, tahap analisis, dan tahap pengembangan sistem. Tahap pengumpulan data dilakukan melalui studi literatur dan wawancara kepada Chief Executive Officer (CEO) GFO. Tahap analisis dilakukan dengan menentukan dan memasukkan nilai minimum *support*, minimum *confidence*, dan sejumlah data transaksi yang digunakan ke dalam sistem. Tahap pengembangan sistem dilakukan dengan menggunakan model *prototype* yang diawali dengan mendengarkan pelanggan, membangun dan memperbaiki *prototype*, dan yang terakhir pelanggan menguji coba *prototype*. Sistem ini dibangun dengan menggunakan bahasa pemrograman *Page Hypertext Pre-Processor* (PHP) dan *framework Code Igniter* (CI). Berdasarkan hasil pengujian *prototype* sistem, sistem ini dapat memberikan rekomendasi paket produk berdasarkan besarnya nilai *support* dan *confidence* dari masing-masing kelompok produk dalam satu transaksi. Dari situlah kemudian *admin* dapat menentukan produk apa saja yang dapat disatukan menjadi paket produk.

PRAKATA

Puji syukur kehadirat Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul Sistem Informasi Penjualan Pada “Grosir Fashion Online” dengan Saran Pembelian Paket Produk Menggunakan Algoritma Apriori. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sistem Informasi Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Prof. Drs. Slamini, M.Comp.Sc., Ph.D., selaku Ketua Program Studi Sistem Informasi Universitas Jember;
2. Drs. Antonius Cahya Prihandoko, M.App.Sc, selaku Dosen Pembimbing Utama dan Muhamad Arief Hidayat S.Kom, M.Kom. selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi;
3. Prof. Drs. Slamini, M.Comp.Sc., Ph.D., selaku Dosen Pembimbing Akademik yang telah membimbing selama penulis menjadi mahasiswa;
4. Seluruh Bapak dan Ibu dosen beserta staf karyawan di Program Studi Sistem Informasi Universitas Jember;
5. Bapak Taufiq, Ibu Mamiek, dan Kakak tercinta Cocom yang telah sangat mendukung baik dalam hal moril dan materil serta selalu memotivasi penulis;
6. Perusahaan grosirfashiononline.com selaku objek penelitian yang telah meluangkan waktu untuk membalas *e-mail* penulis dan membantu dalam melancarkan pengerjaan skripsi ini.
7. Teman-teman seperjuangan diantaranya geng Programmer 2015, Nur Lele, Mumu Ruka, dan Kadek Markodek; geng SKOM yang selalu menghibur dan memberi dukungan; geng Chugil yang selalu mendukung dan memberi semangat walau terpisah jarak; geng KKN tercinta yang tetap jadi pemerhati setia.

8. Semua pihak yang tidak dapat disebutkan satu-persatu.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna, oleh sebab itu penulis mengharapkan adanya masukan yang bersifat membangun dari semua pihak. Penulis berharap skripsi ini dapat bermanfaat bagi semua pihak.

Jember, 7 September 2015



DAFTAR ISI

PERSEMBAHAN	ii
MOTO	iii
PERNYATAAN	iv
PENGESAHAN PEMBIMBING	v
SKRIPSI	vi
PENGESAHAN	vii
RINGKASAN	viii
PRAKATA	x
DAFTAR ISI	xii
DAFTAR TABEL	xv
DAFTAR GAMBAR	xvii
DAFTAR LAMPIRAN	xviii
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Tujuan dan Manfaat	4
1.3.1 Tujuan	4
1.3.2 Manfaat	4
1.4 Ruang Lingkup	5
1.5 Sistematika Penulisan	6
BAB 2. TINJAUAN PUSTAKA	7

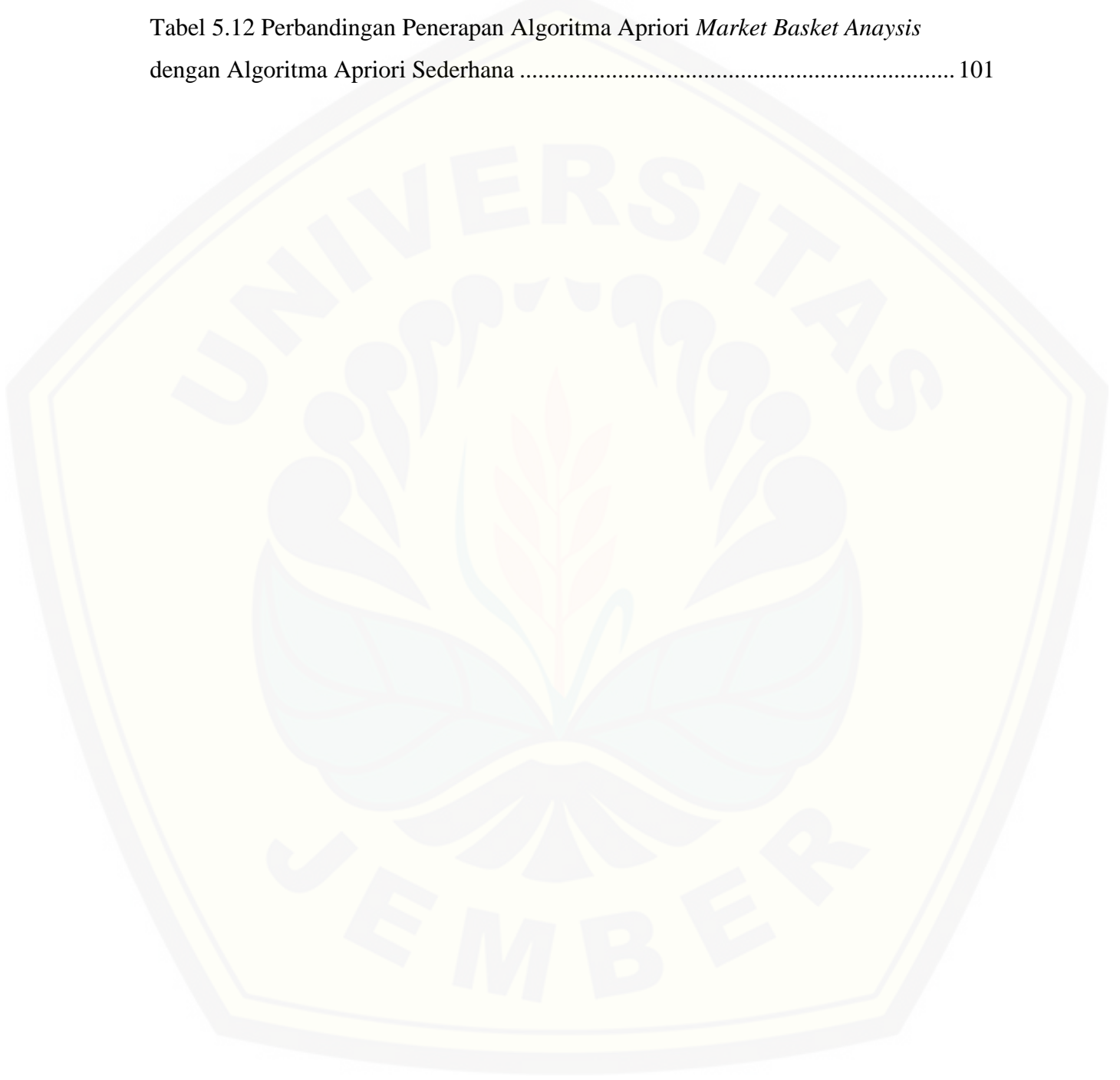
2.1	Penelitian Terdahulu	7
2.2	<i>Online Shopping</i>	9
2.3	Promosi Paket Produk	10
2.4	Data Mining	12
2.5	<i>Market Basket Analysis</i>	12
2.6	Algoritma Apriori	13
BAB 3.	METODOLOGI PENELITIAN	16
3.1	Jenis Penelitian	16
3.2	Tempat dan Waktu Penelitian	16
3.3	Perancangan Sistem	16
3.1.1	Mendengarkan Pelanggan.....	17
3.1.2	Membangun <i>Prototype</i>	17
3.1.3	Pelanggan Menguji Coba <i>Prototype</i>	19
BAB 4.	PENGEMBANGAN SISTEM	20
4.1	<i>Statement Of Purpose</i>	20
4.2	Analisis Kebutuhan Sistem	21
4.2.1	Kebutuhan Fungsional.....	21
4.2.2	Kebutuhan Non-Fungsional.....	22
4.3	Desain Sistem	22
4.3.1	<i>Bussiness Process</i>	22
4.3.2	<i>Usecase Diagram</i>	25
4.3.3	<i>Usecase Skenario</i>	31
4.3.4	<i>Activity Diagram</i>	37

4.3.5	<i>Sequence Diagram</i>	44
4.3.6	<i>Class Diagram</i>	57
4.3.7	<i>Entity Relationship Diagram</i>	60
4.4	Penulisan Kode Program	61
4.5	Pengujian Sistem	70
4.5.1.	Pengujian <i>White Box</i>	71
4.5.2.	Pengujian <i>Black box</i>	86
BAB 5.	HASIL DAN PEMBAHASAN	88
5.1	Hasil Penerapan Algoritma Apriori Pada Proses Perhitungan Rekomendasi Paket Produk	88
5.2	Hasil Pembuatan Sistem Informasi Penjualan GFO	95
5.2.1.	Menu <i>Manage Product Packages</i>	96
5.3	Hasil Perbandingan Proses Penentuan Rekomendasi Paket Produk Secara Konvensional dan Menggunakan Sistem	98
5.4	Perbandingan Penerapan Algoritma Apriori Pada <i>Market Basket Analysis</i> dengan Algoritma Apriori Sederhana	100
BAB 6.	PENUTUP	103
6.1	Kesimpulan	103
6.2	Saran	104
DAFTAR PUSTAKA	105
LAMPIRAN	107

DAFTAR TABEL

Tabel 4.1 Definisi Aktor	26
Tabel 4.2 Definisi <i>Usecase</i>	27
Tabel 4.3 <i>Usecase</i> Skenario Menentukan Rekomendasi Paket Produk	31
Tabel 4.4 Kode Program <i>My_lib</i> Fungsi <i>setData()</i>	61
Tabel 4.5 Kode Program <i>My_lib</i> Fungsi <i>makeTable()</i>	61
Tabel 4.6 Kode Program <i>My_lib</i> Fungsi <i>scan()</i>	63
Tabel 4.7 Kode Program <i>My_lib</i> Fungsi <i>combine()</i>	64
Tabel 4.8 Kode Program <i>My_lib</i> Fungsi <i>realName()</i>	65
Tabel 4.9 Kode Program <i>My_lib</i> Fungsi <i>checkRule()</i>	65
Tabel 4.10 Kode Program <i>My_lib</i> Fungsi <i>confidence()</i>	66
Tabel 4.11 Kode Program <i>My_lib</i> Fungsi <i>freqItemset()</i>	67
Tabel 4.12 Kode Program <i>My_lib</i> Fungsi <i>process()</i>	68
Tabel 4.13 Kode Program <i>My_lib</i> Fungsi <i>getFreqItemset()</i>	69
Tabel 4.14 Kode Program <i>My_lib</i> Fungsi <i>getAssociationRules()</i>	70
Tabel 4.15 Hasil Pengujian <i>Black Box Usecase</i> Menentukan Rekomendasi Paket Produk	86
Tabel 5.1 Nama Produk Sebagai Sampel	89
Tabel 5.2 Pola Transaksi Penjualan Produk GFO	89
Tabel 5.3 Nilai <i>Support</i> dari Masing-masing <i>Item</i>	90
Tabel 5.4 Bentuk 1- <i>Itemset</i>	91
Tabel 5.5 Calon 2- <i>Itemset</i>	91
Tabel 5.6 Bentuk 2- <i>itemset</i>	92
Tabel 5.7 Calon 3- <i>Itemset</i>	92
Tabel 5.8 Nilai <i>Confidence</i> dari Calon Aturan Asosiasi	93
Tabel 5.9 Hasil Aturan Asosiasi Final	94
Tabel 5.10 Hasil Aturan Asosiasi Final Berdasarkan Nama Produk	94

Tabel 5.11 Perbandingan Penentuan Paket Secara Konvensional dengan Menggunakan Sistem	99
Tabel 5.12 Perbandingan Penerapan Algoritma Apriori <i>Market Basket Anaysis</i> dengan Algoritma Apriori Sederhana	101



DAFTAR GAMBAR

Gambar 3.1 Model Prototype Paradigma (Pressman, 2002).....	17
Gambar 4.1 <i>Business Process</i> Sistem Informasi Penjualan GFO	24
Gambar 4.2 <i>Usecase</i> Diagram Sistem Informasi Grosir <i>Fashion Online</i>	25
Gambar 4.3 <i>Activity</i> Diagram Menentukan Rekomendasi Paket Produk	39
Gambar 4.4 <i>Sequence</i> Diagram Menentukan Rekomendasi Paket Produk.....	45
Gambar 4.5 <i>Class Diagram</i> Sistem Informasi Penjualan GFO	59
Gambar 4.6 Entity Relationship Diagram Sistem Informasi Penjualan GFO.....	60
Gambar 4.7 <i>Listing</i> Program Menampilkan Halaman Masukan Kriteria	71
Gambar 4.8 Diagram Alir Fitur Menentukan Rekomendasi Paket Produk	72
Gambar 4.9 Grafik Alir Fitur Menentukan Rekomendasi Paket Produk	72
Gambar 4.10 Listing Program Fungsi Penentuan Nilai Kriteria.....	75
Gambar 4.11 Contoh Penggambaran Diagram Alir <i>Library</i> Algoritma Apriori <i>My_lib</i>	76
Gambar 4.12 Contoh Penggambaran Grafik Alir <i>Library</i> Algoritma Apriori <i>My_lib</i>	77
Gambar 5.9 Halaman Input Kriteria.....	96
Gambar 5.10 Halaman Rekomendasi Paket Produk	98

DAFTAR LAMPIRAN

Lampiran A. Usecase Skenario	107
Lampiran B. Activity Diagram	136
Lampiran C. Sequence Diagram	161
Lampiran D. Kode Program	205
Lampiran E. Listing Program Fitur Menentukan Rekomendasi Paket Produk	317
Lampiran F. Listing Program Library Algoritma Apriori My_lib	318
Lampiran G. Diagram Alir Algoritma Apriori My_lib	325
Lampiran H. Grafik Alir Library Algoritma Apriori My_lib	331
Lampiran I. Pengujian Black Box	337
Lampiran J. Hasil Pembuatan Sistem	342

BAB 1. PENDAHULUAN

Bab ini merupakan bab awal dari laporan tugas akhir. Pada bab ini akan dibahas tentang latar belakang, perumusan masalah, tujuan dan manfaat, ruang lingkup studi, dan sistematika penulisan.

1.1 Latar Belakang

Semakin banyaknya persaingan dalam dunia bisnis khususnya dalam industri penjualan, menuntut para pengembang untuk menemukan suatu strategi yang dapat meningkatkan penjualan dan pemasaran produk yang dijual, salah satunya adalah dengan pemanfaatan data penjualan produk. Tapi perkembangan teknologi yang semakin pesat memberikan kontribusi yang signifikan terhadap perkembangan jumlah data penjualan produk yang disimpan dalam suatu basis data.

Kegiatan penjualan yang dilakukan setiap hari mengakibatkan semakin bertambah banyaknya data. Selain berguna sebagai arsip bagi perusahaan, data tersebut juga dapat diolah menjadi informasi yang berguna untuk peningkatan penjualan dan promosi produk.

Bisnis *online* atau yang sering disebut *online shop* masa kini sangat menjanjikan, namun juga harus siap bersaing mengingat kecanggihan teknologi yang semakin maju. Grosir *Fashion Online* (GFO) merupakan perusahaan bisnis *online* yang menawarkan berbagai produk aksesoris *import* Korea dengan harga lebih terjangkau dibandingkan dengan *online shop* lainnya. Grosir *Fashion Online* sudah berdiri sejak April 2009, dan tercatat hingga bulan Oktober 2013, telah dipercaya oleh lebih dari 870.000 *fans* di *facebook page*-nya dan lebih dari 75.000 *member* terdaftar (Grosir Fashion Online, 2013).

Untuk mengatasi pesaing yang semakin meningkat, GFO memberikan harga produk yang sangat menarik, yaitu dengan harga grosir yang ditujukan bagi para *reseller* di seluruh Indonesia. Cara lainnya yaitu bagi setiap konsumen yang membeli

di website GFO diwajibkan untuk mendaftar sebagai *member*, pemberlakuan sistem *member* ini juga sangat menguntungkan konsumen, karena semakin banyak transaksi yang dilakukan, maka semakin tinggi *level* *membersnya* dan semakin tinggi pula diskon yang akan didapat.

Salah satu strategi promosi lain yang diterapkan oleh beberapa *online shop* yaitu dengan menawarkan paket produk dan dengan harga yang lebih rendah dibanding harga produk jika dibeli satuan. Namun cara ini belum diterapkan pada sistem GFO karena harga yang ditawarkan pada produk GFO sudah merupakan harga grosir yang ditujukan bagi para *reseller*, jika kemudian GFO menambahkan strategi promosi dengan membuat paket-paket produk, maka ini akan membuat perubahan besar pada sistem yang sudah diterapkan selama bertahun-tahun dan telah dipercaya oleh puluhan ribu *member*. Meski begitu, GFO pernah memberikan tawaran paket produk menarik bagi calon konsumen. Cara ini pernah dilakukan GFO meski hanya pada suatu waktu tertentu, semisal pada saat perayaan Hari Raya Imlek, GFO memberikan diskon lebih tinggi jika dalam transaksi pembeli terdapat produk yang telah ditentukan. Sesuai dengan tujuannya, pemaketan produk akan sangat memberikan banyak keuntungan bagi konsumen begitu pula bagi GFO, maka akan lebih baik jika pemaketan produk juga diterapkan.

Grosir *Fashion Online* memiliki ribuan produk dan masing-masing produk memiliki ketertarikan tersendiri bagi calon konsumennya. Menentukan paket-paket produk tidak mungkin dilakukan secara manual oleh karyawan karena karyawan tidak dapat menentukan produk apa yang lebih diminati konsumen dan dibeli secara bersamaan dalam satu kali transaksi. Jumlah data yang begitu besar justru bisa menjadi masalah bagi perusahaan GFO jika tidak bisa dimanfaatkan. Semakin banyak data, maka perusahaan tersebut semakin memerlukan usaha untuk memilah data mana yang dapat diolah menjadi informasi. Dalam waktu empat bulan saja, sudah terdapat dua puluh ribuan transaksi yang masuk sistem GFO, namun jika data sebanyak itu dibiarkan saja, maka hanya akan menjadi sampah yang tidak berarti bagi perusahaan tersebut.

Berdasarkan beberapa permasalahan tersebut, maka diperlukan sebuah aplikasi yang mampu memilah dan memilih data yang besar, sehingga dapat diperoleh informasi yang berguna bagi penggunaanya (Setiawati, 2009). Maka untuk pemilahan produk yang dapat dijadikan paket produk dapat didukung dengan menggunakan suatu algoritma untuk menentukan target promosi yang potensial.

Data Mining digunakan untuk ekstraksi informasi penting yang tersembunyi dari dataset yang besar. Dengan adanya *data mining* maka akan didapatkan pengetahuan di dalam kumpulan data-data yang banyak jumlahnya. Salah satu area penerapan data mining adalah di dalam bidang promosi. Bila sasaran promosi tidak ditentukan secara baik, dalam arti tidak diupayakan mencari sasaran promosi yang potensial, maka hanya akan menghabiskan banyak (Nurcahyo, 2013).

Algoritma dalam datamining yang sesuai dengan data penjualan produk (*market basket analysis*) yaitu algoritma Apriori. Algoritma Apriori adalah algoritma pengambilan data dengan aturan asosiatif (Association rule) untuk menentukan hubungan asosiatif suatu kombinasi *item* (Kusrini dalam Nurcahyo, 2013). Algoritma Apriori yang bertujuan untuk menemukan *frequent item sets* dijalankan pada sekumpulan data. Analisis Apriori didefinisikan suatu proses untuk menemukan semua aturan apriori yang memenuhi syarat minimum untuk *support* dan syarat minimum untuk *confidence* (Pane, 2013).

Berdasarkan penjelasan tersebut, maka algoritma Apriori sangat cocok jika digunakan untuk solusi masalah yang ada. Pada proses mendapatkan target promosi yang potensial, nantinya membutuhkan data produk dan data transaksi penjualan. Dari data-data tersebut dapat diketahui produk-produk apa saja yang dibeli konsumen secara bersamaan dalam satu kali transaksi, sehingga dari hasil pencarian *frequent itemset* dapat dikelompokkan menjadi satu paket yang akan ditawarkan.

1.2 Rumusan Masalah

Dari latar belakang masalah diatas timbul pertanyaan yang dapat dijadikan rumusan masalah, antara lain:

1. Bagaimana merancang dan membangun sistem informasi penjualan Grosir *Fashion Online*?
2. Bagaimana menerapkan algoritma apriori untuk menentukan paket-paket produk berdasarkan data detail transaksi penjualan?
3. Sejauh mana efektifitas sistem informasi penjualan yang akan dibangun?

1.3 Tujuan dan Manfaat

Tujuan dan manfaat dalam penulisan ini merupakan jawaban dari perumusan masalah yang telah disebutkan.

1.3.1 Tujuan

Tujuan dari penelitian ini adalah:

1. Merancang dan membangun sistem informasi penjualan untuk perusahaan Grosir *Fashion Online*.
2. Mengimplementasikan hasil aturan asosiasi final algoritma Apriori berdasarkan data detail transaksi penjualan ke dalam sistem informasi penjualan Grosir *Fashion Online* sebagai paket-paket produk dengan harga lebih rendah.
3. Memahami kegunaan dan efektifitas dari sistem informasi penjualan perusahaan Grosir Fashion online dengan saran pembelian berupa paket produk menggunakan algoritma apriori.

1.3.2 Manfaat

Manfaat yang ingin didapatkan dari penelitian ini adalah:

1. Manfaat Akademis

Hasil penelitian ini diharapkan dapat bermanfaat dengan memberikan kontribusi dan masukan bagi yang membutuhkan informasi yang berhubungan dengan judul penelitian ini. Selain itu, hasil penelitian ini merupakan suatu upaya untuk menambah varian judul penelitian yang ada di Program Studi Sistem Informasi Universitas Jember.

2. Manfaat Bagi Peneliti

Mengetahui bagaimana proses penerapan algoritma Apriori pada proses *datamining* penentuan paket produk pada *online shop* Grosir Fashion Online.

3. Manfaat bagi objek

Memberikan inovasi baru kepada perusahaan tempat penelitian mengenai penggunaan algoritma Apriori guna penentuan paket produk pada waktu atau *event* tertentu.

1.4 Ruang Lingkup

Ruang lingkup dalam penulisan ini merupakan batasan–batasan masalah dalam penulisan. Penulis memberi batasan masalah untuk objek dan tema yang dibahas sehingga tidak terjadi penyimpangan dalam proses penulisan dan pembuatan sistem informasi. Berikut adalah batasan masalah yang dicantumkan:

1. Perancangan dan pembagunan sistem ini guna menerapkan strategi promosi penjualan.
2. Data transaksi penjualan yang digunakan ditentukan oleh admin sesuai dengan batasan tanggal awal dan akhir transaksi yang diinginkan.
3. Metode yang digunakan dalam menentukan *itemset* adalah metode *market basket analysis* dengan algoritma Apriori sehingga dapat diperoleh produk yang ingin dijadikan paket.
4. Pengelolaan paket produk dilakukan oleh *admin* dengan menentukan harga yang lebih rendah dibanding harga produk jika dibeli satuan.

5. Pada sistem yang akan dibangun, tidak terdapat pengelolaan data *stock* barang karena pada GFO sendiri barang diambil dari *supplier* sehingga tidak dapat menentukan jumlah barang yang tersedia.

1.5 Sistematika Penulisan

Sistematika penulisan dan keruntutan skripsi ini disusun sebagai berikut:

1. Pendahuluan
Bab ini menjelaskan tentang latar belakang, perumusan masalah, tujuan dan manfaat, ruang lingkup studi dan sistematika penulisan.
2. Tinjauan Pustaka
Bab ini menjelaskan tentang materi, informasi, tinjauan pustaka, dan studi terdahulu yang menjadi kerangka pemikiran dalam penelitian.
3. Metodologi Penelitian
Bab ini menjelaskan tentang metode penelitian yang digunakan dalam penelitian.
4. Pengembangan Sistem
Bab ini menjelaskan tentang pengembangan sistem yang dikembangkan.
5. Hasil dan Pembahasan
Bab ini menjelaskan tentang hasil dan pembahasan dari penelitian yang dilakukan.
6. Penutup
Bab ini berisi tentang kesimpulan dari penelitian dan saran untuk penelitian selanjutnya.

BAB 2. TINJAUAN PUSTAKA

Bagian ini menjelaskan tinjauan yang berkaitan dengan masalah yang dibahas, kajian teori yang berkaitan dengan masalah, kerangka pemikiran yang merupakan sintesis dari kajian teori yang dikaitkan dengan permasalahan yang dihadapi. Teori-teori ini diambil dari buku literatur, jurnal dan internet. Berikut merupakan teori-teori yang digunakan dan dibahas dalam penelitian:

2.1 Penelitian Terdahulu

Penelitian ini dilakukan berdasarkan teori dan studi-studi yang dilakukan oleh peneliti terdahulu. Adapun penelitian terdahulu tersebut antara lain:

2.1.1 Penelitian Terdahulu dengan judul “Penerapan Metode *Data Mining Market Basket Analysis* Terhadap Data Penjualan Produk Buku dengan Menggunakan Algoritma Apriori dan *Frequent Pattern Growth (FP-Growth)*: Studi Kasus Percetakan PT. Gramedia” yang dilakukan oleh Goldie Gunadi dan Dana Indra Sensuse. Penelitian ini mencoba meneliti tentang penerapan algoritma FP-Growth terhadap *market basket* guna memanfaatkan banyak data untuk mendukung pengambilan keputusan bisnis. Penelitian ini menggunakan algoritma *apriori* dan *FP-Growth*. Menurut Goldie dan Dana dalam jurnalnya menyatakan bahwa algoritma Apriori merupakan algoritma dalam melakukan pencarian *frequent itemset* dengan menggunakan pengetahuan tentang *frequent itemset* yang telah diketahui sebelumnya untuk memroses informasi selanjutnya, sedangkan algoritma *FP-Growth* merupakan perluasan dari algoritma *apriori* yang lebih efektif dan efisien karena hanya melakukan dua langkah untuk melakukan *scan* basisdata. Salah satu hasil yang didapat dalam penelitian ini adalah dari analisa yang dilakukan terhadap tingkat kekuatan aturan-aturan asosiasi menunjukkan bahwa aturan-aturan asosiasi yang dihasilkan oleh

algoritma *apriori* memiliki tingkat kekuatan yang lebih tinggi dibandingkan dengan yang dihasilkan oleh algoritma *FP-Growth*.

Berdasarkan hasil penelitian yang telah dilakukan tersebut, maka dalam penelitian selanjutnya akan lebih baik jika menggunakan algoritma Apriori.

2.1.2 Penelitian terdahulu dengan Judul “Penggunaan Metode Apriori Untuk Analisa Keranjang Pasar Pada Data Transaksi Penjualan Minimarket Menggunakan Java & MySQL” yang dilakukan oleh Devi Dinda Setiawati. Pada penelitian ini, peneliti menerapkan algoritma Apriori terhadap beberapa transaksi pada sebuah minimarket. Hasil asosiasi pada penelitian tersebut didapatkan bahwa:

- a. Hasil asosiasi dapat digunakan untuk merancang kupon diskon. Kupon ini nantinya diberikan pada pelanggan yang membeli produk tertentu, semisal untuk data *itemset* biskuit \Rightarrow minuman. Apabila hasil analisis pembelian biskuit lebih rendah daripada minuman, ada dua strategi yang bisa dilakukan yaitu dengan menawarkan jika seseorang membeli minuman sejumlah x , maka akan mendapatkan biskuit sebagai hadiah atau diskon, atau sebaliknya jika seseorang membeli biskuit sejumlah x maka akan mendapat minuman sebagai hadiah atau diskon. Hal ini dapat meningkatkan jumlah penjualan biskuit.

Hasil asosiasi juga dapat digunakan pada peluncuran produk baru. Semisal seseorang membeli minuman sejumlah x , maka akan mendapat biskuit sebagai hadiah atau diskon. Cara ini dapat menjadi sarana promosi produk pada pelanggan.

- b. Digunakan dalam membantu minimarket untuk menyusun layout toko sebaik mungkin dan sesuai kebutuhan pelanggan. Dari hasil asosiasi, pada data *itemset* biskuit \Rightarrow minuman, efek pada layout toko adalah letak rak biskuit tidak akan jauh dari minuman. Hal ini akan menimbulkan kenyamanan bagi pelanggan yang berbelanja di minimarkete tersebut.

- c. Dapat digunakan dalam membantu minimarket untuk menentukan keputusan persediaan. Sebagai contoh data yang digunakan adalah data *itemset* biskuit \Rightarrow minuman. Dari data tersebut dapat diambil kesimpulan seandainya pelanggan biasa membeli biskuit dan minuman secara bersamaan, maka minimarket setidaknya harus menyediakan kedua barang tersebut dalam jumlah yang sama.
- d. Digunakan untuk melihat hubungan antara penjualan satu produk dengan produk lainnya, misalkan pencarian pola penjualan biskuit sedemikian rupa sehingga dapat diketahui barang apa saja yang juga dibeli oleh pembeli biskuit. Dengan demikian dapat diketahui jika minimarket tidak lagi menjual biskuit.

Dari hasil penelitian tersebut telah banyak diketahui manfaat yang dapat diambil dari penerapan metode market basket analysis menggunakan algoritma Apriori, namun masih ada kegunaan lainnya yaitu dari data *itemset* bisa digunakan untuk membuat suatu paket produk dengan penawaran harga yang lebih rendah. Hal ini mirip dengan penerapan kupon, namun jika pada kupon menyarankan jumlah pembelian tertentu, lain halnya dengan paket produk. Semisal untuk data *itemset* biskuit \Rightarrow minuman, maka kedua produk tersebut bisa dijadikan dalam satu paket, dengan kata lain pembeli harus membeli keduanya untuk mendapatkan harga lebih rendah. Penawaran ini akan semakin menarik jika paket produk ditampilkan di tengah-tengah minimarket atau di depan pintu masuk dengan tampilan semenarik mungkin. Hal ini akan mencuri perhatian pembeli yang baru masuk minimarket.

2.2 Online Shopping

Online Shopping adalah kecenderungan seseorang untuk melakukan pembelian melalui internet (Levin, Levin, & Weller, 2005). Kecenderungan yang dimaksud

adalah pengalaman melalui proses pembelian dari tahap pencarian (search stage) sampai pada tahap keputusan pembelian, bukan saja niat (*intention*) untuk melakukan pembelian *online*. Secara spesifik, pada proses pembelian konsumen, preferensi belanja *online*, terletak pada tahap evaluasi alternatif, karena pada tahap ini konsumen mengadakan pemilihan atribut-atribut produk (Kotler, 2002). *Online shopping* adalah proses dimana konsumen membeli produk atau jasa di internet. Proses seorang konsumen menggunakan media internet untuk melakukan pembelian sebuah produk atau jasa dimulai dengan timbulnya awareness (kesadaran) konsumen akan suatu informasi atau produk yang dapat diperoleh di internet (Roberts, 2003).

2.3 Promosi Paket Produk

Promosi adalah salah satu unsur dalam bauran pemasaran perusahaan yang didayagunakan untuk memberitahukan, membujuk, dan meningkatkan produk perusahaan (Rangkuti, 2009).

Diferensiasi produk adalah kegiatan memodifikasi produk agar lebih menarik dan dilakukan dengan cara mengubah karakter produk. Diferensiasi memerlukan penelitian pasar yang cukup serius agar bisa benar-benar berbeda dan diperlukan juga pengetahuan tentang produk pesaing. Diferensiasi produk ini biasanya hanya mengubah sedikit karakter produk, antara lain kemasan dan tema promosi tanpa mengubah spesifikasi fisik produk, meskipun hal tersebut diperbolehkan.

Tujuan dari strategi diferensiasi adalah mengembangkan *positioning* yang tepat sesuai keinginan konsumen potensial yang ingin dituju diferensiasi produk yang berhasil adalah diferensiasi yang mampu mengalihkan basis persaingan dari harga ke faktor lain, seperti karakteristik produk, strategi distribusi atau variabel-variabel promotif lainnya. Kelemahan dari diferensiasi adalah perlunya biaya produksi tambahan dan iklan besar-besaran.

Beberapa strategi dari diferensiasi produk yaitu dalam *marketing*, *product bundling* adalah strategi untuk menggabungkan penjualan beberapa produk menjadi

satu paket penjualan. Strategi ini sangat umum digunakan dalam bisnis *software*. *Product bundling* sangat tepat untuk produk yang memiliki *volume* penjualan dan tingkat keuntungan yang tinggi (Global, 2012).

Tujuan penjualan melalui *bundling* tersebut adalah untuk meningkatkan penjualan produk dengan memberikan kemudahan prosedur pembelian, penghematan waktu, harga yang relatif lebih murah, serta nilai tambah yang lebih besar kepada konsumen. Disamping itu melalui penjualan *bundling* ini juga dapat menunjukkan adanya *networking* dan kemitraan bisnis antara produsen dari produk yang dijual bersama-sama sehingga saling memperkuat merek produk. Karena tidak semua penjualan *bundling* harus berasal dari satu produsen.

Menurut cara pembelian produk maka penjualan *bundling* dapat dibedakan menjadi: (1) *Bundling* Alternatif yaitu penjualan produk dengan alternatif konsumen bisa memilih antara membeli keseluruhan paket atau dapat membeli produk secara terpisah. (2) *Bundling* Murni yaitu penjualan produk dalam satu paket, konsumen harus membeli dalam satu paket utuh.

Sedangkan menurut keterkaitan fungsi produknya maka penjualan *bundling* dapat dipisahkan menjadi: (1) *Bundling* yang terintegrasi, yaitu paket penjualan produk yang memiliki fungsi saling melengkapi sehingga manfaat yang diperoleh konsumen menjadi maksimal. (2) *Bundling* yang tidak terintegrasi artinya paket produk yang dijual tidak dalam satu fungsi yang terintegrasi.

Strategi penjualan *bundling* akan sangat tepat untuk mendongkrak penjualan bagi produk-produk yang memiliki masa kadaluwarsa produk yang cepat, produk baru yang ketat persaingannya, adanya pasar potensial cukup besar, *marginal cost* rendah, dan tingkat keuntungan yang tinggi. Produk baru akan lebih cepat dikenal dan naik *volume* penjualannya jika dijual secara *bundling* dengan produk yang sudah dikenal dan disukai konsumen. Keuntungan lainnya adalah produsen dapat menghemat biaya promosi karena iklan produk dapat digabungkan menjadi satu (Feriyanto, 2009).

2.4 Data Mining

Data mining adalah suatu istilah yang digunakan untuk menguraikan penemuan pengetahuan di dalam basis data. *Data mining* adalah proses yang menggunakan teknik statistik, matematika, kecerdasan buatan, dan *machine learning* untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai basis data besar.

Data mining adalah proses mencari pola atau informasi menarik dari data terpilih dengan menggunakan metode tertentu. Dalam usaha *retail* atau penjualan umumnya menggunakan metode analisa keranjang belanja (*market basket analysis*) dengan algoritma Apriori (Turban & Aronson, 2000).

Salah satu metode yang seringkali digunakan dalam teknologi *data mining* adalah metode asosiasi atau *association rule mining*. Di dalam bidang usaha *retail* metode *association rule mining* ini lebih dikenal dengan istilah analisa keranjang belanja (*market basket analysis*). *Market basket analysis* adalah suatu metode analisa atas perilaku konsumen secara spesifik dari suatu golongan/kelompok tertentu. Sumber data dari *market basket analysis* antara lain dapat bersumber dari transaksi kartu kredit, kartu lotere, kupon diskon, panggilan keluhan pelanggan. *Market basket analysis* umumnya dimanfaatkan sebagai titik awal pencarian pengetahuan dari suatu transaksi data ketika kita tidak mengetahui pola spesifik apa yang kita cari. Kebutuhan *market basket analysis* berawal dari keakuratan dan manfaat yang dihasilkannya dalam wujud aturan asosiasi (*association rules*). *Association rules* merupakan pola-pola keterkaitan data dalam basis data (Gunadi & Sensuse, 2012).

2.5 Market Basket Analysis

Market Basket Analysis adalah suatu metodologi untuk melakukan analisis *buying habit* konsumen dengan menemukan asosiasi antar beberapa *item* yang berbeda, yang diletakkan konsumen dalam suatu transaksi tertentu. Tujuan dari *market basket analysis* adalah untuk mengetahui produk-produk mana yang mungkin akan dibeli

secara bersamaan (Gunadi & Sensuse, 2012). Dengan mengetahui produk-produk tersebut, maka sebuah sistem *e-commerce* dapat membuat maupun mengembangkan sebuah sistem customer profiles dan dapat menentukan layout katalog pelanggannya sendiri. Perhitungan yang dapat digunakan untuk menentukan asosiasi antara beberapa atribut salah satunya dengan menggunakan algoritma Apriori, karena algoritma Apriori termasuk jenis aturan asosiasi pada data mining.

Seperti pada jurnal Humboldt Universitat Berlin yang telah ada mengenai penerapan metode Market Basket Analysis (Gemala, 2013) Market Basket Analysis adalah Asosiasi berpasangan menggunakan ukuran asosiasi sederhana untuk menunjukkan kebetulan atau afinitas *item* dalam keranjang pasar untuk mengidentifikasi kategori produk yang berhubungan. Seringkali teknik multidimensional scaling atau analisis cluster diterapkan untuk pertama mengurangi jumlah besar kategori.

2.6 Algoritma Apriori

Algoritma Apriori adalah suatu algoritma dasar yang diusulkan oleh Agrawal & Srikant pada tahun 1994 untuk penentuan frequent *itemsets* untuk aturan asosiasi boolean (Gunadi & Sensuse, 2012).

Bentuk algoritma Apriori dapat dituliskan sebagai berikut:

1	<i>L</i> ₁ = {frequent itemset with one element}
2	for (<i>k</i> =2; <i>L</i> _{<i>k</i>-1} ≠ ∅; <i>k</i> ++)
3	{
4	<i>C</i> _{<i>k</i>} = apriori-gen(<i>L</i> _{<i>k</i>-1}); //pembuatan kandidat baru
5	for all transactions <i>t</i>
6	{
7	<i>C</i> ' _{<i>t</i>} = subset(<i>C</i> _{<i>k</i>} , <i>t</i>); //kandidat yang tampil pada <i>t</i>
8	for all candidates <i>c</i> ∈ <i>C</i> ' _{<i>t</i>} do <i>c</i> .count ++;
9	}
10	<i>L</i> _{<i>k</i>} = { <i>c</i> ∈ <i>C</i> ' _{<i>t</i>} <i>c</i> .count ≥ minsup }
11	}
12	return <i>UkLk</i> ;

Dimana:

L: himpunan frequent *itemset*

minsup: minimum *support*

C: himpunan kandidat *itemset*

c: kandidat *itemset*

t: transaksi

Algoritma Apriori termasuk jenis aturan asosiasi pada *data mining*. Aturan yang menyatakan asosiasi antara beberapa atribut sering disebut *affinity analysis* atau *market basket analysis*. Analisis asosiasi atau *association rule mining* adalah teknik *data mining* untuk menemukan aturan suatu kombinasi *item*. Salah satu tahap analisis asosiasi yang menarik perhatian banyak peneliti untuk menghasilkan algoritma yang efisien adalah analisis pola frekuensi tinggi (*frequent pattern mining*). Penting tidaknya suatu asosiasi dapat diketahui dengan dua tolak ukur, yaitu *support* dan *confidence*.

Support adalah ukuran dari seberapa sering koleksi *item* dalam asosiasi terjadi bersama sebagai presentase dari seluruh transaksi. Sedangkan *confidence* adalah ukuran ketidakpastian atau kepercayaan kelayakan yang terkait dengan setiap pola yang ditemukan.

1. Analisis Pola Frekuensi Tinggi dengan Algoritma Apriori

Tahap ini mencari kombinasi *item* yang memenuhi syarat minimum dari nilai *support* dalam basis data (Gunadi & Sensuse, 2012).

Nilai *support item* didapat dengan Persamaan (2.1):

$$Support(A) = \frac{\Sigma \text{Transaksi mengandung } A}{\Sigma \text{Transaksi}} \times 100\% \quad (2.1)$$

Nilai *support* dari dua *item* diperoleh dari Persamaan (2.2):

$$Support(A, B, C) = \frac{\Sigma \text{Transaksi mengandung } A, B, C}{\Sigma \text{Transaksi}} \times 100\% \quad (2.2)$$

Frequent *itemset* menunjukkan *itemset* yang memiliki frekuensi kemunculan lebih dari nilai minimum yang ditentukan (\emptyset). Misalkan $\emptyset = 2$, maka semua *itemsets* yang frekuensi kemunculannya lebih dari atau sama dengan 2 kali disebut frequent. Himpunan dari frequent *k-itemset* dilambangkan dengan F_k .

2. Pembentukan Aturan Asosiasi

Setelah semua pola frekuensi tinggi ditemukan, barulah dicari aturan asosiasi yang memenuhi syarat minimum untuk *confidence* dengan menghitung *confidence* aturan asosiatif $A \Rightarrow B$ diperoleh dari Persamaan (2.3).

$$Confidence = P(B|A) = \frac{\Sigma \text{Transaksi mengandung } A \text{ dan } B}{\Sigma \text{Transaksi mengandung } A} \times 100\% \quad (2.3)$$

Untuk menentukan aturan asosiasi yang akan dipilih maka harus diurutkan berdasarkan $Support \times Confidence$. Aturan diambil sebanyak n aturan yang memiliki hasil terbesar (Pane, 2013).

BAB 3. METODOLOGI PENELITIAN

Bab ini menjelaskan tentang jenis penelitian, metode penelitian, tempat dan waktu penelitian, serta tahapan yang dilakukan untuk membangun Sistem Informasi Penjualan Pada “Grosir Fashion Online” dengan Saran Pembelian Paket Produk Menggunakan Algoritma Apriori.

3.1 Jenis Penelitian

Penelitian ini termasuk jenis penelitian pengembangan. Termasuk penelitian pengembangan karena peneliti bermaksud untuk mengembangkan sistem informasi penjualan pada perusahaan Grosir Fashion Online yang dapat memenuhi tiga komponen penelitian pengembangan yaitu model pengembangan, prosedur pengembangan, dan uji coba produk.

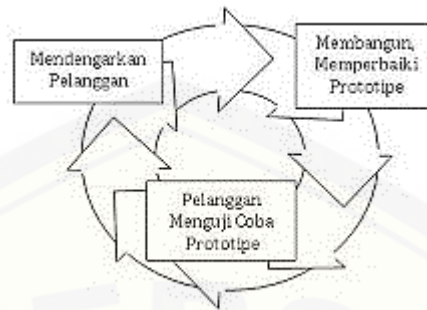
3.2 Tempat dan Waktu Penelitian

Penelitian ini dilakukan pada website www.grosirfashiononline.com. Waktu penelitian dilakukan selama empat bulan, sejak bulan April hingga Agustus 2015.

3.3 Perancangan Sistem

Perancangan sistem ini menggunakan model *prototype*. Penggunaan model perancangan *prototype* ini bertujuan untuk lebih memahami kebutuhan *user* serta dapat menghemat waktu dalam pengembangan sistem.

Model ini dimulai dengan mengumpulkan kebutuhan dimana pengembang dan pelanggan bertemu dan mengidentifikasi objektif keseluruhan dari perangkat lunak, mengidentifikasi segala kebutuhan yang diketahui, dan area garis besar dimana definisi lebih jauh merupakan keharusan dan kemudian dilakukan perancangan kilat (Pressman, 2002).



Gambar 3.1 Model Prototype Paradigma (Pressman, 2002)

Seperti pada Gambar 3.1, pada proses perancangan sistem ini, akan dilakukan dengan tiga tahap siklus seperti pada model *prototype*, yaitu:

3.1.1 Mendengarkan Pelanggan

Tahap ini dimulai dengan pengumpulan kebutuhan dengan menganalisis kebutuhan dan mengidentifikasi segala kebutuhan. Maka dapat ditetapkan tujuan perancangan, pengajuan usulan solusi yang dapat diterima.

Tahap yang dilakukan antara lain:

1. Menganalisa sistem yang sudah ada pada website Grosir Fashion Online.
2. Memberikan usulan pengembangan sistem dengan menerapkan strategi pemasaran produk.

3.1.2 Membangun *Prototype*

Setelah memahami sistem yang akan dikembangkan, maka tahap berikutnya yaitu melakukan perancangan sistem dan pengkodean.

3.1.2.1 Perancangan Sistem

Perancangan sistem merupakan proses yang berfokus pada struktur arsitektur perangkat lunak, representasi *interface*, dan detail algoritma. Tahapan ini akan menghasilkan dokumen yang disebut *software requirement*.

Perancangan sistem ini digunakan *Unified Modeling Language* (UML) untuk membuat suatu model sehingga bisa menghasilkan diagram grafis yaitu *bussiness process, use case diagram, use case scenario, activity diagram, sequence diagram* dan *class diagram*.

3.1.2.2 Pengkodean

Pada tahap ini desain yang telah dibuat akan diimplementasikan ke dalam kode program. Beberapa hal yang dilakukan dalam tahap implementasi antara lain:

1. Penulisan kode program menggunakan bahasa *Page Hypertext Preprocessor (PHP), Cascading Style Sheet (CSS), dan Hyper Text Markup Language (HTML)* yang dibantu oleh *tool* NetBeans IDE 8.0.1.
2. Basis data Management System (DBMS) yang digunakan adalah MySQL. Menggunakan Code Igniter sebagai *frame work*.

3.1.2.3 Pengujian

Setelah program selesai dibuat, berikutnya dilakukan pengujian program untuk bisa menghasilkan program yang lebih maksimal, benar-benar berjalan dengan baik dan sesuai harapan. Pengujian dilakukan dengan metode *white box testing* dan *black box testing* (Nidhra & Dondeti, 2012).

1. *White box Testing*

White box testing digunakan untuk mendeteksi kesalahan logic dalam kode program. Hal ini digunakan untuk mencari kesalahan sebuah kode, mencari kesalahan penulisan acak, dan mengungkap pemrograman yang salah asumsi.

2. *Black box Testing*

Black box testing membantu validasi fungsi keseluruhan sistem. *Black box* testing dilakukan berdasarkan requirements sehingga requirements yang diinginkan pelanggan yang tidak lengka atau tidak terduga dapat dengan mudah diidentifikasi dan dapat diatasi kemudian. *Black box* testing dilakukan berdasarkan perspektif pengguna akhir.

3.1.3 Pelanggan Menguji Coba Prototype

Tahap ini dimana sistem telah selesai dibuat dan diuji, kemudian diperlihatkan pada pelanggan. Disini juga terjadi evaluasi oleh pelanggan apakah sistem tersebut sesuai dengan yang diharapkan atau masih perlu tambahan agar dapat sesuai.



BAB 4. PENGEMBANGAN SISTEM

Pada bab ini dibahas mengenai pengembangan Sistem Informasi Penjualan Pada “Grosir Fashion Online” dengan Saran Pembelian Paket Produk Menggunakan Algoritma Apriori. Tahap pengembangan sistem dilaksanakan berdasarkan model *prototype*, dimulai dari mendengarkan pelanggan, membangun dan memperbaiki *prototype*, kemudian pelanggan menguji coba *prototype*.

4.1 *Statement Of Purpose*

Sistem informasi penjualan dengan saran pembelian paket produk digunakan untuk memberikan rekomendasi paket produk berdasarkan data transaksi yang memenuhi nilai minimum *support* dan *confidence*. Tujuan dikembangkan sistem ini adalah untuk membantu *admin* dalam menentukan produk apa saja yang dapat dikombinasikan menjadi paket produk. Pada sistem ini terdapat tiga *user*, yaitu *admin*, CEO, dan *member*. CEO hanya dapat menambah dan mengubah *user admin*. *Admin* dapat mengelola data produk, data *member*, data transaksi, data testimonial, mengelola kategori, mengelola gambar *landing page*, mengubah *level member*, mengubah status *order*, serta mengubah informasi toko. Sedangkan *member* dapat mengubah data *member*, mengelola data *order*, mengelola data konfirmasi bayar, menambah testimonial, melihat testimonial, serta melihat halaman cara *order*.

Pembentukan paket produk pada sistem informasi penjualan GFO ini dimulai ketika admin memasukkan data paket berupa nama paket, harga paket, dan detail paket lainnya pada halaman *Add Product Package* dan menekan tombol “Tentukan Isis Paket”. Kemudian proses kerja algoritma apriori dimulai dari admin menentukan nilai minimum *support*, minimum *confidence*, dan batasan tanggal pada halaman *Input Kriteria*. Kemudian ketika tombol “Submit” ditekan, sistem akan melakukan perhitungan terhadap data transaksi menggunakan algoritma apriori. Hasil dari perhitungan inilah yang akan menjadi rekomendasi paket produk.

4.2 Analisis Kebutuhan Sistem

Berdasarkan tahapan yang harus dilakukan sesuai model perancangan Prototype, tahap yang pertama dilakukan adalah mendengarkan pelanggan. Pada tahap ini peneliti menganalisis kebutuhan sistem sesuai dengan keinginan pelanggan. Seluruh kebutuhan penggunaan didefinisikan dan diformulasikan pada tahap kebutuhan fungsional dan kebutuhan non-fungsional.

4.2.1 Kebutuhan Fungsional

Kebutuhan fungsional merupakan hal utama yang harus dimiliki sistem yang ingin dibuat sehingga dapat menerima masukan untuk diproses dan menghasilkan keuaran. Kebutuhan fungsional berisi proses-proses yang nantinya dilakukan oleh sistem informasi penjualan Pada Grosir Fashion Online dengan saran pembelian paket produk menggunakan algoritma apriori, yaitu:

1. Sistem dapat mengelola (input, ubah, hapus) data dan menyimpan data pendukung pengambil keputusan seperti data *user* dan data produk oleh *admin*, serta data *order* oleh *member*.
2. Sistem dapat menampilkan data produk agar *member* dapat melakukan transaksi.
3. Sistem dapat menyediakan halaman masukan kriteria untuk proses penentuan rekomendasi paket produk.
4. Sistem dapat mengelola dan mengatur proses pengambilan keputusan paket produk.
5. Sistem dapat menganalisis *itemset* yang dapat dijadikan rekomendasi sebagai paket produk berdasarkan kriteria yang telah ditentukan.
6. Sistem dapat menampilkan hasil analisis *itemset* sebagai aturan asosiasi final yang dapat direkomendasikan sebagai paket produk.
7. Sistem memiliki tiga tampilan yaitu *admin* panel untuk *admin* yang berisi menu Mengelola *Member*, Mengelola Data Produk, Menentukan Rekomendasi Paket

Produk, Mengelola Kategori, Gambar Landing Page, Informasi Toko, serta Testimonial. Tampilan kedua yaitu tampilan CEO yang berisi menu *User Admin*. Berikutnya tampilan ketiga untuk *member* yang dapat mengubah data *member*, mengelola data *order*, melihat status *order*, menambah konfirmasi bayar, melihat informasi toko, menambah dan melihat testimonial, serta melihat halaman Cara *Order*.

4.2.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional merupakan hal yang dibutuhkan oleh sistem untuk mendukung aktivitas sistem sesuai dengan kebutuhan fungsional yang telah disusun. Kebutuhan non-fungsional menitikberatkan pada properti perilaku yang dimiliki oleh sistem. Kebutuhan non-fungsional pada sistem informasi penjualan Grosir Fashion Online dengan algoritma Apriori adalah sebagai berikut:

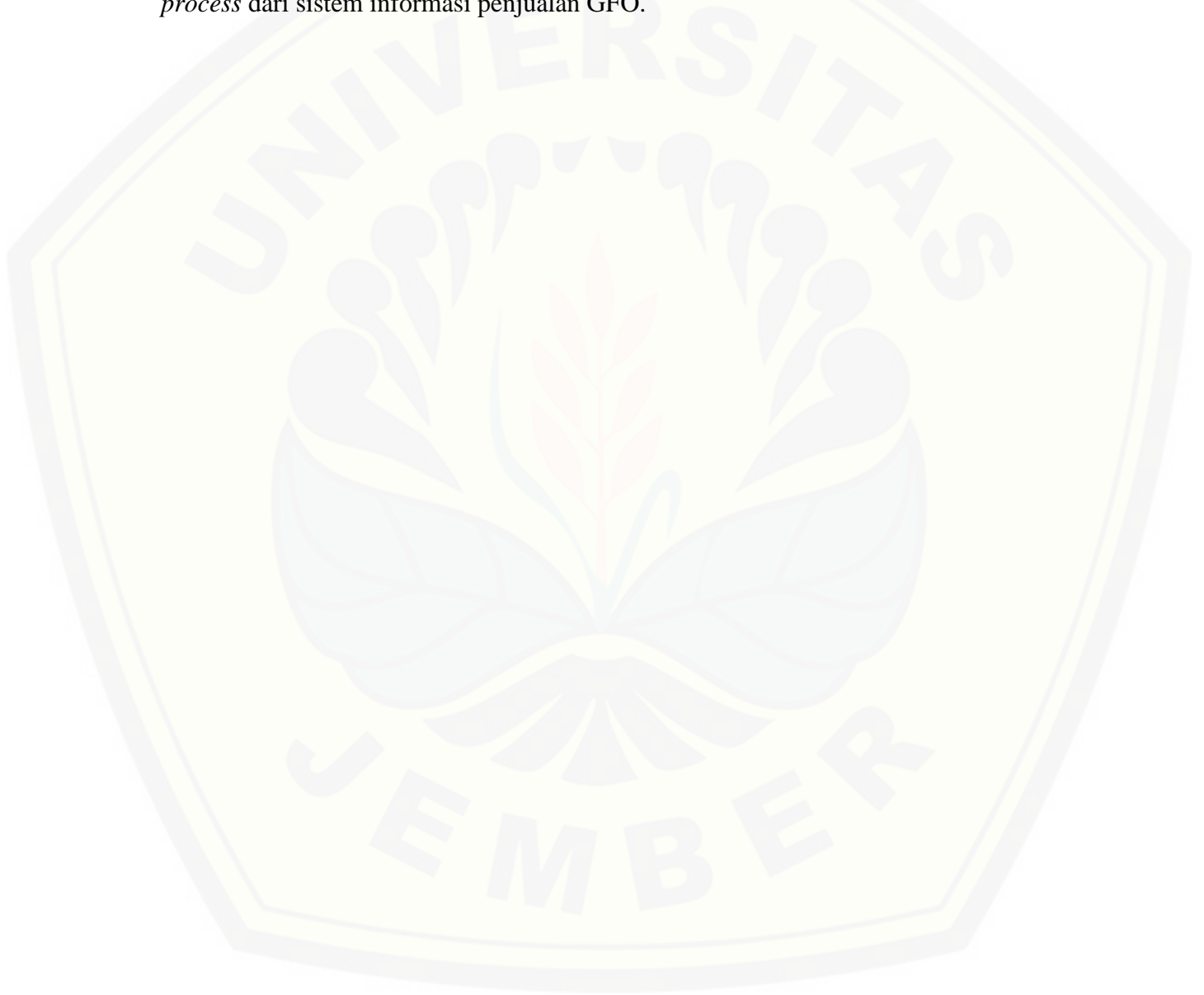
1. Sistem ini dapat digunakan oleh siapapun dan kapanpun bahkan pada saat yang bersamaan dengan menggunakan komputer yang berbeda.
2. Sistem menggunakan *username* dan *password* untuk autentifikasi akses pengguna terhadap sistem.
3. Tampilan dan bahasa yang digunakan pada sistem dapat dimengerti oleh pengguna sehingga dapat dengan mudah digunakan.
4. Sistem mudah dioperasikan pada beberapa sistem operasi.

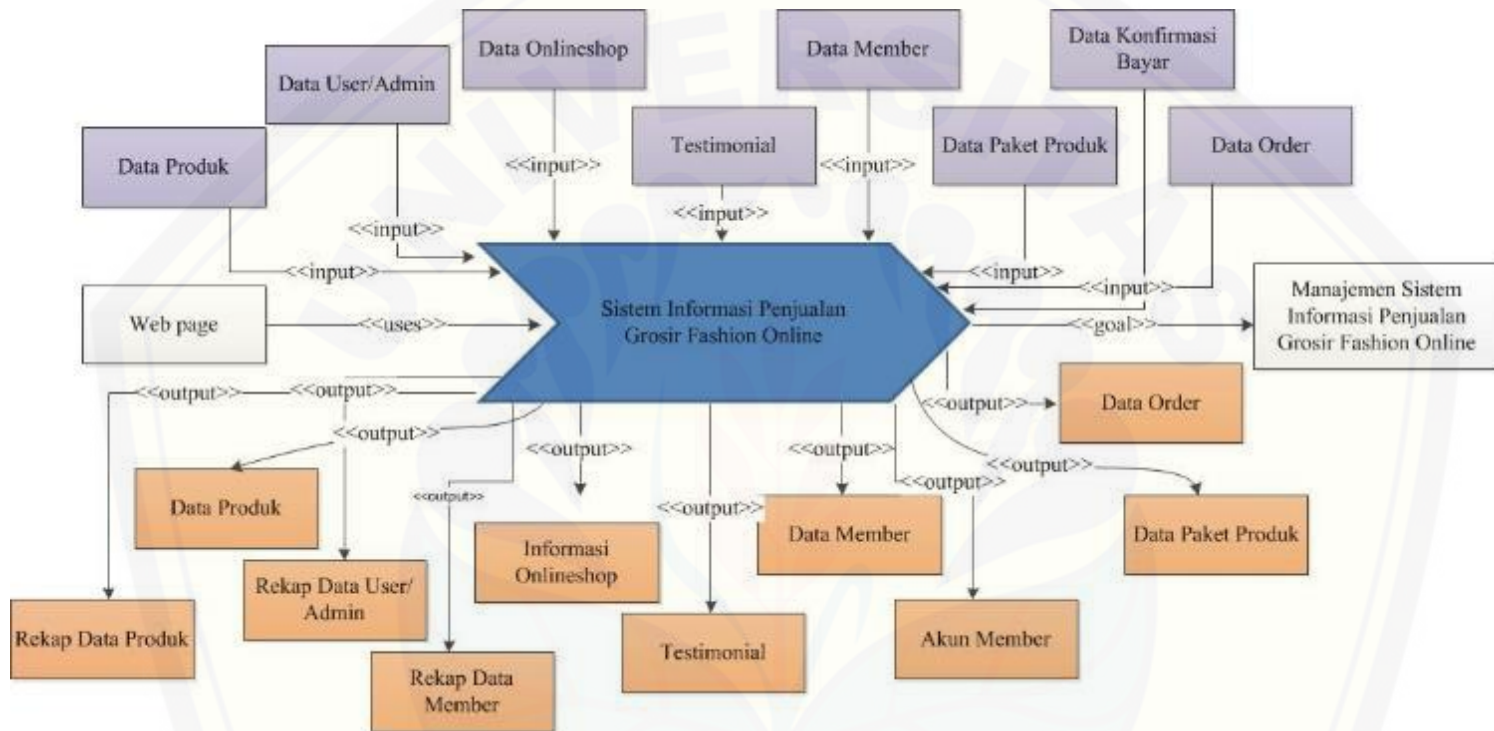
4.3 Desain Sistem

Desain sistem adalah tahapan untuk memodelkan sistem informasi penjualan Grosir Fashion Online. Desain sistem yang dibuat yaitu *Bussiness Process*, *Usecase Diagram*, *Usecase Skenario*, *Activity Diagram*, *Sequence Diagram*, *Class Diagram*, dan *Entity Relationship Diagram (ERD)*.

4.3.1 *Bussiness Process*

Bussiness process merupakan sekumpulan proses yang dilakukan pada sistem yang dibuat untuk mencapai hasil yang diinginkan. Komponen yang dibutuhkan untuk mencapai tujuan tersebut yaitu data yang menjadi masukan (*input*) dan media yang digunakan sehingga akan menghasilkan data keluaran (*output*) untuk melakukan tujuan (*goal*) yang ingin dicapai dari sistem tersebut. Pada Gambar 4.1 dapat dilihat *business process* dari sistem informasi penjualan GFO.

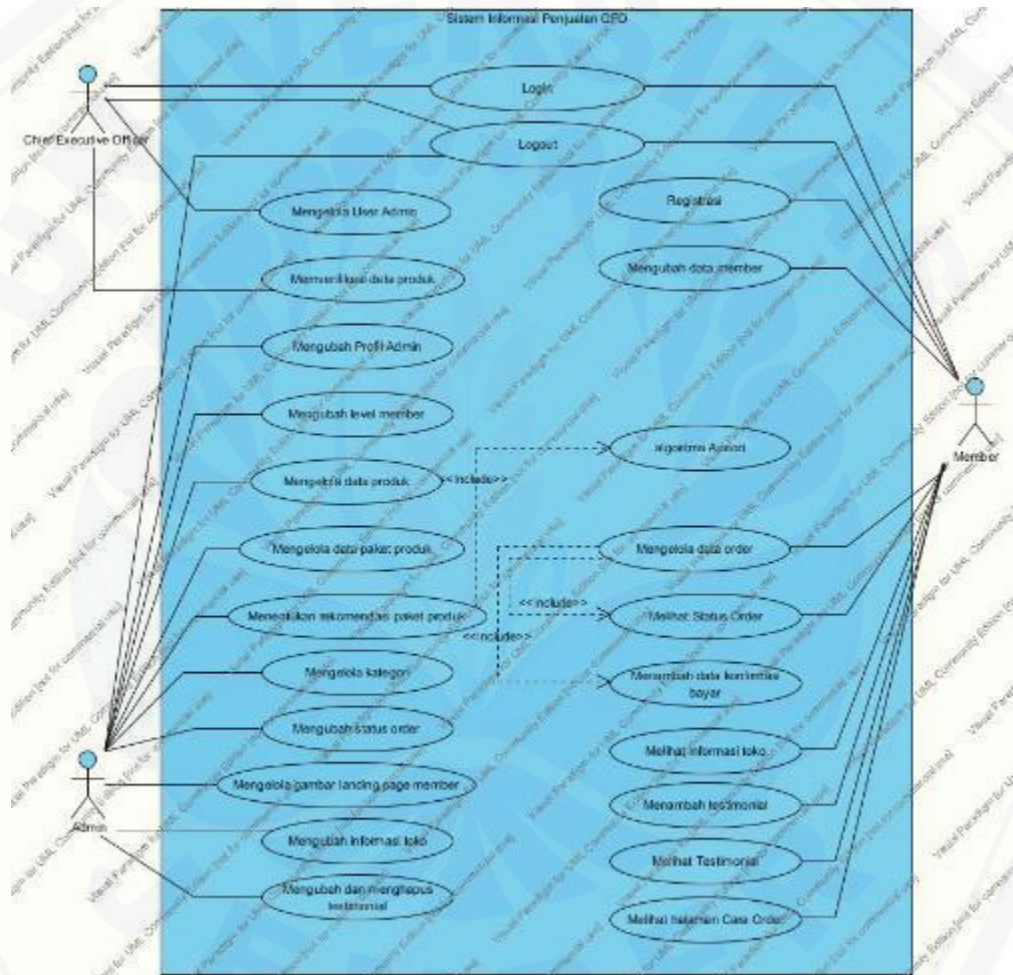




Gambar 4.1 *Business Process* Sistem Informasi Penjualan GFO

4.3.2 Usecase Diagram

Usecase Diagram adalah dokumentasi untuk menggambarkan fitur dan aktor yang terdapat pada sistem yang akan dibuat. *Usecase diagram* menampilkan aktor mana yang menggunakan *usecase* mana, *usecase* mana yang memasukkan *usecase* lain, dan hubungan antara aktor dan *usecase*. Pada Gambar 4.2 dapat dilihat *usecase diagram* dari sistem informasi penjualan GFO.



Gambar 4.2 *Usecase Diagram* Sistem Informasi Grosir Fashion Online

Penjelasan mengenai definisi aktor dan definisi *usecase* dalam *usecase* diagram sistem informasi penjualan GFO adalah sebagai berikut:

1. Definisi Aktor

Definisi aktor merupakan penjelasan tentang aktor-aktor sebagai pengguna sistem informasi penjualan GFO yang akan dibangun. Berdasarkan hasil analisis, terdapat tiga aktor pada sistem ini yang akan dijelaskan pada Tabel 4.1.

Tabel 4.1 Definisi Aktor

No.	Aktor	Deskripsi
1.	CEO	Aktor CEO hanya bertugas untuk mengelola data <i>admin</i> , yaitu menambah dan menghapus data <i>admin</i> . CEO juga dapat memberikan <i>approval</i> pada data produk yang baru ditambahkan <i>admin</i> . Aktor ini dapat melakukan <i>login</i> dan <i>logout</i> untuk masuk ke sistem.
2.	<i>Admin</i>	Aktor ini memiliki sebagian besar akses dari sistem, yaitu dapat melakukan <i>login</i> dan <i>logout</i> , mengelola profil <i>admin</i> , mengubah <i>level member</i> , mengelola data produk, menentukan rekomendasi paket produk, mengelola status <i>order</i> , mengelola gambar <i>landing page member</i> , mengubah informasi toko, serta mengubah dan menghapus testimonial.
3.	<i>Member</i>	Akses yang dapat dilakukan oleh aktor <i>member</i> yaitu registrasi, <i>login</i> , <i>logout</i> , mengubah data <i>member</i> , mengelola data <i>order</i> , melihat status <i>order</i> , menambah data konfirmasi bayar, melihat informasi toko, menambah testimonial, melihat testimonial, serta melihat halaman informasi Cara <i>Order</i> .

2. Definisi *Usecase*

Definisi *usecase* merupakan penjelasan dari masing-masing *usecase* atau fitur-fitur pada sistem informasi penjualan GFO, seperti yang dapat dilihat pada Tabel 4.2.

Tabel 4.2 Definisi *Usecase*

No.	<i>Usecase</i>	Penjelasan
GFO100-1	<i>Login</i>	Merupakan <i>usecase</i> yang menggambarkan proses <i>login</i> atau autentifikasi agar <i>user</i> dapat memasuki sistem.
GFO100-2	<i>Logout</i>	Merupakan <i>usecase</i> yang menggambarkan proses <i>user</i> keluar dari sistem.
GFO301	Mengelola <i>Admin</i>	Merupakan <i>usecase</i> yang menggambarkan proses CEO menambah dan menghapus data <i>admin</i> .
GFO102	Mengubah Data Profil <i>Admin</i>	Merupakan <i>usecase</i> yang menggambarkan proses mengubah data profil <i>admin</i> .
GFO201	Register <i>Member</i>	Merupakan <i>usecase</i> yang menggambarkan proses register <i>member</i> baru.
GFO203	Mengubah Data <i>Member</i>	Merupakan <i>usecase</i> yang menggambarkan proses <i>member</i> mengubah data nama, alamat, nomor telepon, <i>username</i> , serta <i>password</i> .
GFO103	Mengubah <i>Level</i> <i>Member</i>	Merupakan <i>usecase</i> yang menggambarkan proses <i>admin</i> mengelola data <i>member</i> dengan mengubah <i>level member</i> .
Berlanjut		

Lanjutan		
GFO104	Mengelola Data Produk	Merupakan <i>usecase</i> yang menggambarkan proses <i>admin</i> menambah, mengubah, dan menghapus data produk. Dalam proses ini juga termasuk proses <i>admin</i> menentukan kategori produk serta <i>upload</i> gambar produk.
GFO105	Menentukan Rekomendasi Paket Produk	Merupakan <i>usecase</i> yang menggambarkan proses <i>admin</i> dalam menentukan paket produk. <i>Admin</i> hanya perlu memasukkan nilai minimal <i>support</i> , minimal <i>confidence</i> , serta batasan tanggal transaksi yang ingin ditinjau.
GFO106	Mengelola Kategori Produk	Merupakan <i>usecase</i> yang menggambarkan proses <i>admin</i> menambah, mengubah, dan menghapus kategori-kategori produk.
GFO204	Mengelola Data <i>Order</i>	Merupakan <i>usecase</i> yang menggambarkan proses <i>member</i> melakukan transaksi, yang di dalamnya termasuk proses menambah produk, mengubah <i>quantity</i> , dan menghapus produk pada keranjang belanja. Selain itu pada proses ini <i>member</i> juga diberi pilihan apakah ingin mengubah alamat pengiriman atau tidak.
Berlanjut		

Lanjutan		
GFO205	Melihat Daftar <i>Order</i>	Merupakan <i>usecase</i> yang menggambarkan proses <i>member</i> melihat daftar <i>order</i> yang pernah dilakukan, detail <i>order</i> , serta melihat status <i>order</i> -nya.
GFO107	Mengubah Status <i>Order</i>	Merupakan <i>usecase</i> yang menggambarkan proses <i>admin</i> mengelola data dengan mengubah status <i>order</i> jika konfirmasi bayar sudah dipenuhi oleh <i>member</i> .
GFO206	Menambah Konfirmasi Bayar	Merupakan <i>usecase</i> yang menggambarkan <i>member</i> melakukan konfirmasi bayar dengan mengisi <i>form</i> konfirmasi bayar yang disediakan setelah melakukan pembayaran terhadap <i>order</i> yang telah dilakukan.
GFO108	Mengelola Gambar Landing Page	Merupakan <i>usecase</i> yang menggambarkan proses <i>admin</i> menambah dan menghapus gambar dan <i>link</i> gambar pada halaman <i>landing page member</i> .
GFO109	Mengubah Informasi Toko	Merupakan <i>usecase</i> yang menggambarkan proses <i>admin</i> mengubah informasi toko.
GFO207	Melihat Informasi Toko	Merupakan <i>usecase</i> yang menggambarkan proses <i>member</i> melihat informasi toko.
Berlanjut		

		Lanjutan
GFO110	Mengelola Testimonial	Merupakan <i>usecase</i> yang menggambarkan proses <i>admin</i> mengubah serta menghapus testimonial jika tidak ada hubungannya dengan GFO atau menggunakan bahasa yang kurang sopan.
GFO208	Menambah Testimonial	Merupakan <i>usecase</i> yang menggambarkan proses <i>member</i> menambah kesan setelah belanja di GFO.
GFO209	Melihat Testimonial	Merupakan <i>usecase</i> yang menggambarkan proses <i>member</i> melihat testimonial-testimonial sebelumnya yang pernah ditambahkan oleh <i>member</i> lain.
GFO210	Melihat Cara <i>Order</i>	Merupakan <i>usecase</i> yang menggambarkan proses <i>member</i> melihat halaman Cara <i>Order</i> .
GFO211	Memverifikasi data produk	Merupakan <i>usecase</i> yang menggambarkan proses CEO mengubah verifikasi data produk agar ditampilkan ke halaman <i>member</i> atau tidak.
GFO212	Mengelola data paket produk	Merupakan <i>usecase</i> yang menggambarkan proses <i>admin</i> menambah, mengubah, dan menghapus data paket produk. Dalam proses ini juga termasuk proses <i>admin</i> menentukan kategori produk serta <i>upload</i> gambar paket produk.

4.3.3 *Usecase* Skenario

Usecase skenario adalah dokumentasi terhadap kebutuhan fungsional sistem.

Usecase skenario untuk sistem informasi penjualan GFO adalah sebagai berikut:

1. *Usecase* Skenario Menentukan Rekomendasi Paket Produk

Urutan reaksi aktor dan sistem pada *usecase* skenario proses *admin* menentukan rekomendasi paket produk dapat dilihat pada Tabel 4.3.

Tabel 4.3 *Usecase* Skenario Menentukan Rekomendasi Paket Produk

Nama <i>Usecase</i>	: Menentukan Rekomendasi Paket Produk
ID <i>Usecase</i>	: GFO105
Aktor	: <i>Admin</i>
Pre Kondisi	: Aktor memilih menu “Manajemen Paket Produk”
Post Kondisi	: Aktor berhasil mendapatkan rekomendasi paket produk

NORMAL SKENARIO MENENTUKAN REKOMENDASI PAKET PRODUK

Aktor	Sistem
1. Klik submenu Add Product Package	2. Menampilkan kolom teks <i>input</i> data paket produk disertai tombol “Save”
3. Mengisi kolom teks input data paket	
4. Menekan tombol “Save”	5. Menampilkan halaman Add Product Package serta tombol “Tentukan Isi Paket Produk”
6. Menekan tombol “Tentukan Isi Paket Produk”	7. Menampilkan kolom teks Kriteria berupa nilai minimum support, confidence, dan batasan tanggal
8. Mengisi kolom teks sesuai dengan kriteria yang diinginkan	
9. Menekan tombol “Hitung Apriori”	
Berlanjut	

 Lanjutan

10. Menampilkan hasil aturan asosiasi final sebagai rekomendasi paket produk pada halaman Hasil Rekomendasi Paket Produk

NORMAL SKENARIO MENGUBAH DETAIL PAKET PRODUK

11. Menekan tombol “Ubah Harga”
12. Menampilkan kolom teks *input* data paket produk disertai tombol “Save”
13. Mengisi kolom teks input data paket
14. Menekan tombol “Save”
15. Menampilkan kolom teks *input* data paket produk disertai tombol “Save”

**NORMAL SKENARIO TOMBOL “UPLOAD PIC”
PADA UPDATE PRODUCT**

16. Klik tombol “Upload Pic”
17. Menampilkan halaman Upload Product Picture
18. Klik tombol “Browse”
19. Menampilkan window File Upload
20. Memilih salah satu foto produk
21. Klik tombol “Open”
22. Klik tombol “Upload”
23. Menyimpan nama *file* gambar ke dalam basis data
24. Menampilkan halaman Upload Product Picture disertai dengan gambar paket produk yang baru
-

Alternatif Skenario Upload Pic: Ukuran file yang dipilih terlalu besar

**ALTERNATIF SKENARIO
UKURAN FILE YANG DIPILIH TERLALU BESAR**

- 22b. Ukuran file yang dipilih lebih besar dari yang diminta
- 23b. Menampilkan pesan error
-

2. *Usecase Skenario Login*

Pada *usecase* skenario login dijelaskan mengenai proses aktor memasukkan *username* dan *password* pada portal login untuk dapat memasuki sistem. Penjelasan

urutan reaksi aktor dan sistem pada *usecase* skenario proses *user login* dapat dilihat pada Lampiran A.1 (*Usecase* Skenario Login Admin dan CEO).

3. *Usecase* Skenario Logout

Usecase skenario *logout* menjelaskan tentang proses aktor keluar dari sistem. Penjelasan urutan reaksi aktor dan sistem pada *usecase* skenario *logout* dapat dilihat pada Lampiran A.2 (*Usecase* Skenario Logout).

4. *Usecase* Skenario Mengelola User Admin

Usecase skenario mengelola user admin menjelaskan mengenai proses *admin* dapat menambah, mengubah, dan menghapus data *admin* dengan memilih menu *Manage Admin* pada *CEO Panel*. Penjelasan urutan reaksi aktor dan sistem pada *usecase* skenario mengelola *user admin* dapat dilihat pada Lampiran A.3 (*Usecase* Skenario Mengelola User Admin).

5. *Usecase* Skenario Register Member

Usecase skenario register member menjelaskan mengenai proses member memasukkan data diri pada halaman *register member* kemudian dapat menampilkan halaman *member account*. Penjelasan urutan reaksi aktor dan sistem pada *usecase* skenario register *member* dapat dilihat pada Lampiran A.5 (*Usecase* Skenario Register Member).

6. *Usecase* Skenario Login Member

Pada *usecase* skenario login dijelaskan mengenai proses member memasukkan *username* dan *password* pada portal login untuk dapat memasuki sistem. Penjelasan urutan reaksi aktor dan sistem pada *usecase* skenario proses *member login* dapat dilihat pada Lampiran A.6 (*Usecase* Skenario Login Member).

7. *Usecase Skenario Logout Member*

Usecase skenario *logout* menjelaskan tentang proses member keluar dari sistem. Penjelasan urutan reaksi aktor dan sistem pada *usecase* skenario *logout member* dapat dilihat pada Lampiran A.7 (*Usecase Skenario Logout Member*).

8. *Usecase Skenario Mengubah Data Member*

Usecase skenario mengubah data member menjelaskan tentang proses member mengubah data diri pada halaman update member account. Halaman ini berisi kolom teks nama, *email*, alamat, kota, kode pos, dan nomor telepon dan dapat disimpan dengan menekan tombol “Submit”. Penjelasan urutan reaksi aktor dan sistem pada *usecase* skenario mengubah data *member* dapat dilihat pada Lampiran 0 (*Usecase Skenario Mengubah Data Member*).

9. *Usecase Skenario Mengubah Level Member*

Usecase skenario mengubah level member menjelaskan mengenai proses proses admin memilih menu Manage Member kemudian memilih salah satu data member untuk dapat diubah levelnya dengan menekan tombol “Ubah Level”. Penjelasan urutan reaksi aktor dan sistem pada *usecase* skenario mengubah *level member* dapat dilihat pada Lampiran A.9 (*Usecase Skenario Mengubah Level Member*).

10. *Usecase Skenario Mengelola Data Produk*

Usecase skenario mengelola data produk menjelaskan mengenai proses proses admin menambah data produk dengan mengisi kolom teks pada halaman Add Product, mengubah data produk pada halaman Manage Product, dan menghapus data produk dengan menekan tombol “Delete” disertai dengan pesan konfirmasi hapus data. Penjelasan urutan reaksi aktor dan sistem pada *usecase* skenario mengelola data produk dapat dilihat pada Lampiran A.10 (*Usecase Skenario Mengelola Data Produk*).

11. *Usecase Skenario Mengelola Kategori*

Usecase skenario mengelola kategori menjelaskan mengenai proses admin menambah nama kategori pada halaman Tambah Kategori dengan mengisi kolom teks nama kategori, mengubah kategori pada halaman Ubah Kategori, dan menghapus kategori yang dipilih dengan menekan tombol “Delete”. Penjelasan urutan reaksi aktor dan sistem pada *usecase* skenario mengelola kategori dapat dilihat pada Lampiran A.11 (*Usecase* Skenario Mengelola Kategori).

12. *Usecase* Skenario Mengelola Data *Order*

Usecase skenario mengelola data *order* menjelaskan mengenai proses member melakukan transaksi dengan menambah produk yang akan dibeli pada Tas Belanja. Skenario ini menjelaskan hingga proses transaksi selesai dan menampilkan halaman yang berisi nomor *order* dan nomor rekening GFO. Penjelasan urutan reaksi aktor dan sistem pada *usecase* skenario mengelola data *order* dapat dilihat pada Lampiran 12 (*Usecase* Skenario Mengelola Data *Order*).

13. *Usecase* Skenario Melihat Daftar *Order*

Usecase skenario melihat daftar *order* menjelaskan mengenai proses member memilih menu *Order* Saya untuk menampilkan daftar *order* dan dapat menampilkan halaman Kelengkapan *Order* jika member memilih tombol “Kelengkapan *Order*”. Penjelasan urutan reaksi aktor dan sistem pada *usecase* skenario melihat status *order* dapat dilihat pada Lampiran 13 (*Usecase* Skenario Melihat Daftar *Order*).

14. *Usecase* Skenario Mengubah Status *Order*

Usecase skenario mengubah status *order* menjelaskan mengenai proses admin dapat mengubah status *order* pada data *order* yang dipilih setelah menekan tombol “Update Status” dan memilih salah satu status *order*. Penjelasan urutan reaksi aktor dan sistem pada *usecase* skenario mengubah status *order* dapat dilihat pada Lampiran 14 (*Usecase* Skenario Mengubah Status *Order*).

15. *Usecase* Skenario Menambah Data Konfirmasi Bayar

Usecase skenario menambah data konfirmasi bayar menjelaskan mengenai proses member mengisi *form* Konfirmasi Bayar yang berisi data-data pembayaran dan menyimpannya ke dalam basis data dengan menekan tombol “*Submit*”. Penjelasan urutan reaksi aktor dan sistem pada *usecase* skenario menambah data konfirmasi bayar dapat dilihat pada Lampiran 15 (*Usecase* Skenario Menambah Data Konfirmasi Bayar).

16. *Usecase* Skenario Mengelola Gambar Landing Page *Member*

Usecase skenario mengelola gambar *landing page* menjelaskan mengenai proses admin menambah gambar landing page dengan menekan tombol “Browse” dan “Upload” pada halaman *Upload Homepage Pictures*. Pada halaman ini admin juga dapat menghapus gambar yang telah diunggah dengan menekan tombol “Delete”. Penjelasan urutan reaksi aktor dan sistem pada *usecase* skenario mengelola gambar landing page halaman *member* dapat dilihat pada Lampiran 16 (*Usecase* Skenario Mengelola Gambar Landing Page *Member*).

17. *Usecase* Skenario Mengubah Informasi Toko

Usecase skenario mengubah informasi toko menjelaskan proses admin mengisi kolom teks *Store Info* dan menyimpannya ke dalam database. Penjelasan urutan reaksi aktor dan sistem pada *usecase* mengubah informasi toko dapat dilihat pada Lampiran 17 (*Usecase* Skenario Mengubah Informasi Toko).

18. *Usecase* Skenario Melihat Informasi Toko

Usecase skenario melihat informasi toko menjelaskan mengenai proses member menampilkan halaman Informasi Toko dengan memilih menu *Store Info*. Penjelasan urutan reaksi aktor dan sistem pada *usecase* skenario melihat informasi toko dapat dilihat pada Lampiran 18 (*Usecase* Skenario Melihat Informasi Toko).

19. *Usecase* Skenario Mengubah dan Menghapus Testimonial

Usecase skenario mengubah dan menghapus testimonial menjelaskan mengenai proses *admin* menampilkan halaman Ubah Testimonial dengan menekan tombol “*Edit*” dan menampilkan halaman konfirmasi. Penjelasan urutan reaksi aktor dan sistem pada *usecase* skenario mengubah dan menghapus testimonial dapat dilihat pada Lampiran 19 (*Usecase* Skenario Mengubah dan Menghapus Testimonial).

20. *Usecase* Skenario Menambah Testimonial

Usecase skenario menambah testimonial menjelaskan mengenai proses member membuka halaman Tambah Testimonial, kemudian mengisi dan menekan tombol “*Submit*” untuk menambahkan testimonials. Penjelasan urutan reaksi aktor dan sistem pada *usecase* skenario menambah testimonial dapat dilihat pada Lampiran 20 (*Usecase* Skenario Menambah Testimonial).

21. *Usecase* Skenario Melihat Testimonial

Usecase skenario melihat testimonial menjelaskan proses member menampilkan halaman yang berisi seluruh testimonial. Penjelasan urutan reaksi aktor dan sistem pada *usecase* skenario melihat testimonial dapat dilihat pada Lampiran 21 (*Usecase* Skenario Melihat Testimonial).

22. *Usecase* Skenario Melihat Halaman Cara *Order*

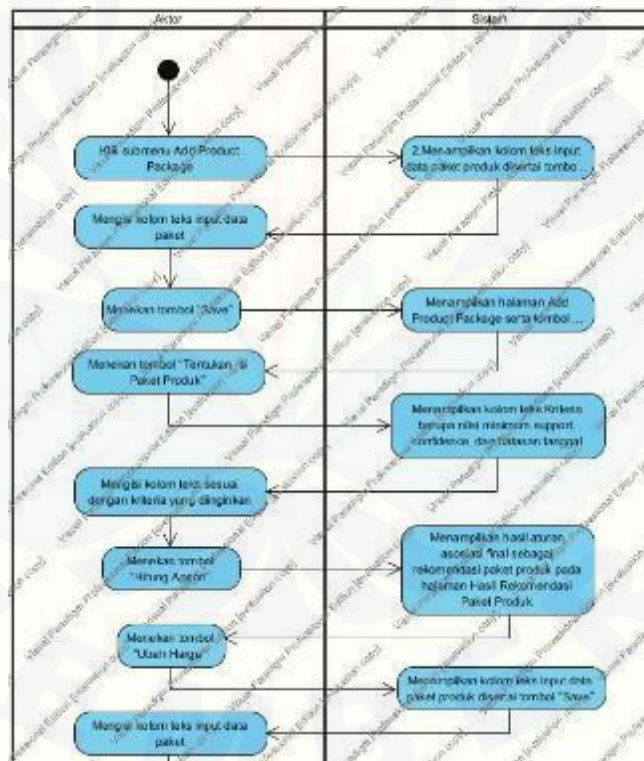
Usecase skenario melihat halaman cara *order* menjelaskan proses member melihat urutan cara *order* pada halaman Cara *Order*. Penjelasan urutan reaksi aktor dan sistem pada *usecase* skenario melihat halaman cara *order* dapat dilihat pada Lampiran 22 (*Usecase* Skenario Melihat Halaman Cara *Order*).

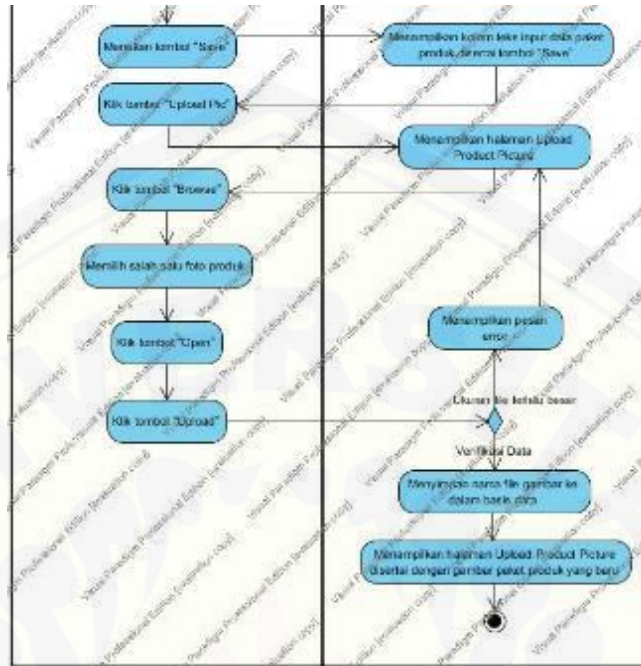
4.3.4 Activity Diagram

Activity Diagram menggambarkan aliran aktivitas antara aktor dan sistem pada sistem informasi penjualan GFO yang kana dibangun. Sistem informasi penjualan GFO ini memiliki dua puluh satu activity diagram, yaitu:

1. Activity Diagram Menentukan Rekomendasi Paket Produk

Activity diagram menentukan rekomendasi paket produk dilakukan oleh *admin*. Activity diagram ini menjelaskan tentang proses *admin* menggunakan sistem informasi penjualan GFO untuk menentukan nilai minimum *support*, minimum *confidence*, batasan tanggal transaksi yang ingin diproses serta melihat hasil rekomendasi paket produk oleh sistem. Activity diagram menentukan rekomendasi paket produk dapat dilihat pada Gambar 4.3.





Gambar 4.3 Activity Diagram Menentukan Rekomendasi Paket Produk

2. Activity Diagram Login

Activity diagram *login* dapat dilakukan oleh semua *user* yaitu *admin*, CEO, dan *member*. Activity diagram *login* menjelaskan aliran proses *user* dapat masuk ke dalam sistem. Activity ini dapat dilihat pada Lampiran B.1 (Activity Diagram Login).

3. Activity Diagram Logout

Activity diagram *logout* dapat dilakukan oleh semua *user* yaitu *admin*, CEO, dan *member*. Activity diagram *logout* menjelaskan aliran proses *user* keluar dari sistem. Activity ini dapat dilihat pada Lampiran B.2 (Activity Diagram Logout).

4. Activity Diagram Mengelola User Admin

Activity diagram mengelola *user admin* yang dilakukan oleh CEO ini menjelaskan tentang proses CEO menambah dan menghapus data *admin*. Activity

diagram mengelola *user admin* dapat dilihat pada Lampiran B.3 (*Activity Diagram Mengelola User Admin*).

5. *Activity Mengubah Profil Admin*

Activity diagram mengubah profil admin yang dilakukan oleh admin ini menjelaskan tentang proses admin mengubah data diri admin pada menu Profil Admin. *Activity* diagram register *member* dapat dilihat pada Lampiran 4 (*Activity Diagram Mengubah Profil Admin*).

6. *Activity Diagram Register Member*

Activity diagram register *member* yang dilakukan oleh *member* baru ini menjelaskan tentang proses *member* mendaftar sebelum dapat melakukan transaksi pada sistem informasi penjualan GFO. *Activity* diagram register *member* dapat dilihat pada Lampiran 5 (*Activity Diagram Register Member*).

7. *Activity Diagram Login Member*

Activity diagram *login member* yang dilakukan oleh *member* ini menjelaskan tentang proses *member* masuk ke dalam sistem. *Activity* diagram *login member* dapat dilihat pada Lampiran B.6 (*Activity Diagram Login Member*).

8. *Activity Diagram Logout Member*

Activity diagram *logout member* yang dilakukan *member* ini menjelaskan tentang proses *member* keluar dari sistem. *Activity* diagram *logout member* dapat dilihat pada Lampiran B.7 (*Activity Diagram Logout Member*).

9. *Activity Diagram Mengubah Data Member*

Activity diagram mengubah data *member* yang dilakukan oleh *member* ini menjelaskan tentang proses *member* mengubah data berupa nama, alamat, nomor

handphone, dan sebagainya. *Activity* diagram mengubah data *member* dapat dilihat pada Lampiran B.8 (*Activity* Diagram Mengubah Data *Member*).

10. *Activity* Diagram Mengubah *Level Member*

Activity diagram mengubah *level member* yang dilakukan oleh *admin* ini menjelaskan proses *admin* mengelola data *member* dengan mengubah *level member* sesuai dengan poin yang telah didapat *member*. *Activity* diagram mengubah *level member* dapat dilihat pada Lampiran 0 (*Activity* Diagram Mengubah *Level Member*).

11. *Activity* Diagram Mengelola Data Produk

Activity diagram mengelola data produk yang dilakukan oleh *admin* ini menjelaskan tentang proses *admin* menambah, mengubah, dan menghapus data produk, serta menentukan kategori produk. *Activity* diagram mengelola data produk dapat dilihat pada Lampiran B.10 (*Activity* Diagram Mengelola Data Produk).

12. *Activity* Diagram Mengelola Kategori

Activity diagram mengelola kategori yang dilakukan oleh *admin* ini menjelaskan tentang proses *admin* menambah, mengubah, dan menghapus kategori. *Activity* diagram mengelola kategori dapat dilihat pada Lampiran B.11 (*Activity* Diagram Mengelola Kategori).

13. *Activity* Diagram Mengelola Data *Order*

Activity diagram mengelola data *member* yang dilakukan oleh *member* ini menjelaskan tentang proses *member* melakukan transaksi, dimulai dari menambah barang pada keranjang belanja, mengubah quantity, menghapus, hingga checkout. *Activity* diagram mengelola data *order* dapat dilihat pada Lampiran B.12 (*Activity* Diagram Mengelola Data *Order*).

14. *Activity Diagram Melihat Status Order*

Activity diagram melihat status order yang dilakukan oleh *member* ini menjelaskan tentang proses *member* melihat halaman daftar *order* yang telah dilakukan *member* untuk mengetahui sampai mana proses *ordernya* dan melihat detail *order* dapat dilihat pada Lampiran B.13 (*Activity Diagram Melihat Status Order*).

15. *Activity Diagram Mengubah Status Order*

Activity diagram mengubah status order yang dilakukan oleh *admin* ini menjelaskan tentang proses *admin* mengelola data *order* dengan mengubah status *order* sesuai dengan yang dilakukan pegawai perusahaan terhadap *order* dengan nomor *order* tersebut. *Activity diagram mengubah status order* dapat dilihat pada Lampiran B.14 (*Activity Diagram Mengubah Status Order*).

16. *Activity Diagram Menambah Data Konfirmasi Bayar*

Activity diagram menambah data konfirmasi bayar yang dilakukan oleh *member* ini menjelaskan tentang proses *member* memasukkan data pembayaran seperti tanggal pembayaran, nama pemilik rekening, umlah yang dibayarkan, dan sebagainya. *Activity diagram menambah data konfirmasi bayar* dapat dilihat pada Lampiran B.15 (*Activity Diagram Menambah Data Konfirmasi Bayar*).

17. *Activity Diagram Mengelola Gambar Landing Page*

Activity diagram mengelola gambar landing page yang dilakukan oleh *admin* ini menjelaskan tentang proses *admin* menambah dan menghapus gambar serta link untuk halaman landing page *member*. *Activity diagram mengelola gambar landing page* dapat dilihat pada Lampiran B.16 (*Activity Diagram Mengelola Gambar Landing Page*).

18. *Activity Diagram Mengubah Informasi Toko*

Activity diagram mengubah informasi toko yang dilakukan oleh *admin* ini menjelaskan tentang proses *admin* mengubah informasi toko berupa profil *online shop*, nomor layanan, jam kerja, serta informasi *reseller*. *Activity* diagram mengubah informasi toko dapat dilihat pada Lampiran B.17 (*Activity* Diagram Mengubah Informasi Toko).

19. *Activity* Diagram Melihat Informasi toko

Activity diagram melihat informasi toko yang dilakukan oleh *member* ini menjelaskan tentang proses *member* melihat penjelasan mengenai *online shop*. *Activity* diagram melihat informasi toko dapat dilihat pada Lampiran B.18 (*Activity* Diagram Melihat Informasi toko).

20. *Activity* Diagram Mengubah dan Menghapus Testimonial

Activity diagram mengubah dan menghapus testimonial yang dilakukan oleh *admin* ini menjelaskan tentang proses *admin* mengubah atau menghapus sebagian kata yang kurang pantas untuk ditampilkan. *Activity* diagram mengubah dan menghapus testimonial dapat dilihat pada Lampiran B.19 (*Activity* Diagram Mengubah dan Menghapus Testimonial).

21. *Activity* Diagram Menambah Testimonial

Activity diagram menambah testimonial yang dilakukan oleh *member* ini menjelaskan tentang proses *member* memberikan kesan setelah melakukan transaksi melalui sistem informasi penjualan GFO. *Activity* diagram menambah testimonial dapat dilihat pada Lampiran B.20 (*Activity* Diagram Menambah Testimonial).

22. *Activity* Diagram Melihat Testimonial

Activity diagram melihat testimonial yang dilakukan oleh *member* ini menjelaskan tentang proses *member* melihat seluruh testimonial yang pernah

ditambahkan oleh *member* lain. *Activity* diagram melihat testimonial dapat dilihat pada Lampiran B.21 (*Activity* Diagram Melihat Testimonial).

23. *Activity* Diagram Melihat Halaman Cara *Order*

Activity diagram melihat halaman cara *order* yang dilakukan oleh *member* ini menjelaskan tentang proses *member* membuka halaman cara *order* kemudian melihat tahap-tahap untuk dapat menambah data *order* pada sistem informasi penjualan GFO. *Activity* diagram melihat halaman cara *order* dapat dilihat pada Lampiran B.22 (*Activity* Diagram Melihat Halaman Cara *Order*).

4.3.5 *Sequence* Diagram

Sequence diagram adalah suatu diagram yang menggambarkan interaksi antar obyek dan mengindikasikan komunikasi diantara obyek-obyek tersebut. Diagram ini juga menunjukkan serangkaian pesan yang dipertukarkan oleh obyek-obyek yang melakukan suatu tugas atau aksi tertentu. Pembuatan *sequence* diagram ini bertujuan untuk mengkomunikasikan *requirement* kepada tim teknis karena diagram ini dapat lebih mudah untuk dielaborasi menjadi model desain, selain itu *sequence* diagram juga merupakan diagram yang paling cocok untuk mengembangkan model deskripsi *usecase* menjadi spesifikasi desain.

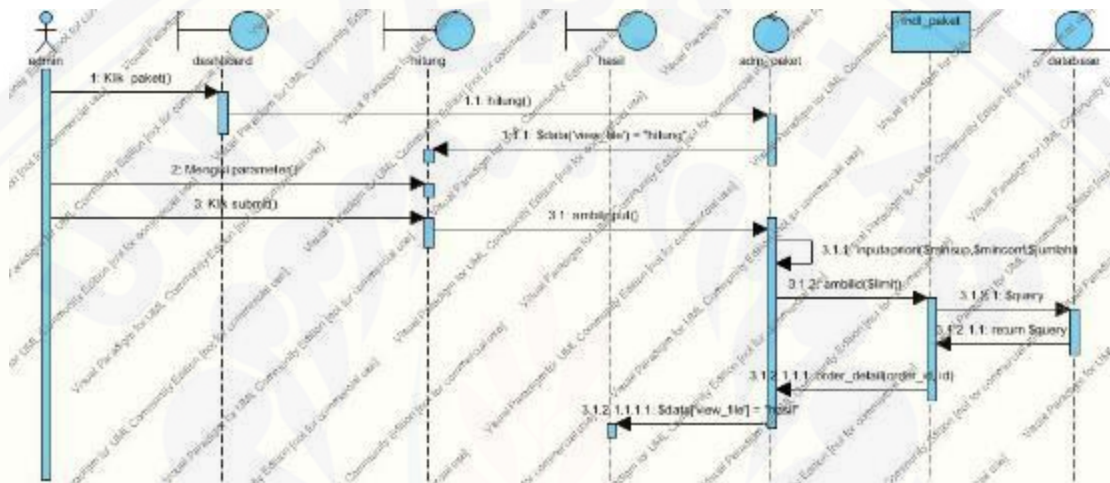
Sequence diagram dari sistem informasi penjualan GFO adalah sebagai berikut:

1. *Sequence* Diagram Menentukan Rekomendasi Paket Produk

Sequence diagram menentukan rekomendasi paket produk menggambarkan dan menjelaskan mengenai fungsi atau *method* yang akan digunakan pada program. Masing-masing *class* akan ditampilkan visual dengan gambar. *Sequence* ini terdiri dari *class view* hitung dan hasil, *class controller adm_paket*, dan *class model mdl_paket* seperti yang dapat dilihat pada Gambar 4.4.

Proses ini diawali dengan *admin* memilih menu Paket, sehingga sistem menjalankan fungsi *hitung()* untuk menampilkan *view* hitung. Minsup, minconf, dan jumlah merupakan parameter masukan yang ditentukan oleh *admin*. Fungsi

`ambilinput()` digunakan untuk menghitung parameter yang dimasukkan sesuai dengan *library* algoritma Apriori pada sistem. Kemudian dari fungsi `ambilinput()` dilanjutkan pada fungsi `inputapriori($minsup,$minconf,$jumlah)`. Fungsi `ambilid($limit)` adalah untuk mengambil `order_id` dari tabel `order_detail`, sedangkan `$limit` adalah batasan data yang diambil yang ditentukan dengan parameter `$jumlah`.



Gambar 4.4 *Sequence* Diagram Menentukan Rekomendasi Paket Produk

2. *Sequence* Diagram Login

Sequence diagram *login* menjelaskan mengenai penggambaran fungsi atau *method* proses aktor memasuki sistem yang digunakan pada program yang akan dibuat. Masing-masing *class* akan ditampilkan visual dengan gambar. Karena pada sistem informasi penjualan GFO ini memiliki tiga *user*, yaitu CEO, *admin*, dan *member*, maka pada *login*, *controller* yang digunakan akan sesuai dengan *user*-nya. Pada modul CEO, *class* yang dimiliki *sequence* diagram *login* yaitu *login*, *home*, *ceo*, dan *mdl_ceo*. Pada *sequence* diagram *login* *user* lain, *class* yang dimiliki sama kecuali *controller*. Pada *login* *user* *admin*, *controller* yang dimiliki bernama *admin*, sedangkan pada *user* *member* *controller* yang dimiliki bernama *member*. Namun proses yang dilakukan memiliki aliran yang sama, pertama yaitu aktor memasukkan syarat utama untuk dapat

memasuki sistem, yaitu *username* dan *password*. Setelah itu aktor menekan tombol “Login” sehingga sistem menjalankan fungsi *process()*, dari sini sistem memeriksa apakah sesuai dengan basis data atau tidak melalui fungsi *validasi()* pada *mdl_ceo*. Jika data yang dimasukkan sesuai dengan basis data, maka sistem akan menampilkan *view home*. Namun jika data masukan tidak sesuai dengan basis data, maka sistem menampilkan halaman *login* kembali disertai dengan pesan bahwa data masukan tidak sesuai. *Sequence diagram login* oleh CEO, *admin* dilampirkan pada Lampiran C.1 (*Sequence Diagram Login*).

3. *Sequence Diagram Logout*

Sequence diagram logout menjelaskan mengenai penggambaran fungsi atau *method* proses aktor keluar dari sistem yang digunakan pada program yang akan dibuat. Masing-masing *class* akan ditampilkan visual dengan gambar. Sama seperti *login*, pada *sequence diagram logout* ini dapat dilakukan oleh ketiga user. Pada *sequence diagram logout* CEO dan *admin* terdapat *class view home* dan *login* serta *class controller adm_dahsboard*. Sedangkan pada *sequence diagram logout user member* terdapat *class view account* dan *landing_page*, serta *class controller member* dan *mem_homepage*. Namun fungsi yang digunakan sama, yaitu fungsi *do_logout()* untuk menghapus semua *session* yang baru saja dilakukan. *Sequence diagram logout* dapat dilihat pada Lampiran 0 (*Sequence Diagram Logout*).

4. *Sequence Diagram Mengelola User Admin*

Sequence diagram login menjelaskan mengenai penggambaran fungsi atau *method* proses CEO menambah dan menghapus data *admin* yang digunakan pada program yang akan dibuat. Masing-masing *class* akan ditampilkan visual dengan gambar. *Class* yang terdapat pada *sequence diagram mengelola user admin* yaitu pada *view* terdapat *home, create, manage, delete_conf*; pada *controller* terdapat *ceo*; dan pada *model* terdapat *mdl_ceo*.

Proses yang pertama yaitu aktor memasukkan data *admin*, kemudian memilih tombol “*Submit*” pada *view create*. Tombol “*Submit*” kemudian meneruskan ke fungsi *submit()*, *controller* kemudian mengambil data masukan dengan fungsi *get_data_from_post()* dan menambahkan ke basis data melalui model dengan fungsi *_insert(\$user_id, \$data)*. Setelah data masuk ke basis data, sistem menampilkan kembali *view create* disertai pesan bahwa data berhasil disimpan. Jika aktor batal menambahkan data dan ingin menghapus data yang telah dimasukkan pada form isian, maka aktor memilih tombol “*Reset*” sehingga dilakukan fungsi *reset()* untuk mengosongkan *form* isian.

Jika aktor ingin menghapus data *admin*, maka aktor dapat memilih menu “*Manage Admin*” yang akan diteruskan oleh *controller* dengan fungsi *manage()* untuk menampilkan *view manage*, dalam *view manage* terdapat fungsi *Modules::run('ceo/_display_user_table')* yang bermaksud untuk memanggil fungsi *_display_user_table()* pada *controller* yang dapat menampilkan tabel data *user* yang dipanggil melalui model dengan fungsi *user()*. Setelah data diambil dari basis data, kemudian ditampilkan pada *view manage* dengan fungsi *\$this->load->view('user_table', \$data)*. Pada fungsi *delete_admin(\$user_id)*, sistem menampilkan halaman konfirmasi hapus data. Jika aktor ingin menghapus data, sistem menjalankan fungsi *_delete(\$user_id)*, jika tidak maka aktor memilih tombol “*No*” dan kembali menampilkan *view manage* dengan menjalankan fungsi *manage()*. *Sequence* diagram mengelola user admin dapat dilihat pada Lampiran C.3 (*Sequence Diagram Mengelola User Admin*).

5. *Sequence Diagram Mengubah Profil Admin*

Sequence diagram mengubah profil *admin* menjelaskan mengenai penggambaran fungsi atau *method* proses CEO menambah dan menghapus data *admin* yang digunakan pada program yang akan dibuat. Masing-masing *class* akan ditampilkan visual dengan gambar. *Class* yang terdapat pada *sequence* diagram mengubah profil *admin* pada *view*

terdapat *class home* dan *update*, pada *controller* terdapat *class admin*, dan pada model terdapat *class mdl_user*.

Proses awalnya yaitu ketika *admin* memilih tombol *dropdown* pada *header admin panel* dan memilih tombol “*User Profile*” sehingga sistem menjalankan *update(\$user_id)* dan menampilkan *view update*. Pada *view update*, terdapat fungsi *submit()* untuk mengubah data pada basis data sesuai data masukan, dan tombol “*Cancel*” untuk kembali ke *view home*. *Sequence* diagram mengubah profil *admin* dapat dilihat pada Lampiran C.4 (*Sequence Diagram Mengubah Profil Admin*).

6. *Sequence Diagram Registrasi Member*

Sequence diagram registrasi *member* menjelaskan mengenai penggambaran proses *member* mendaftar untuk mendapatkan akses masuk ke sistem. Pada *sequence* ini terdapat *class view landing_page*, *account*, dan *login*; *class controller member*; dan *class model mdl_member*.

Proses ini diawali dengan fungsi *login()* untuk menampilkan *view login*. Kemudian fungsi *form_register()* untuk menampilkan *form register member*. Fungsi *submit()* dijalankan saat *member* menekan tombol “*Submit*” yang kemudian menjalankan fungsi *get_data_from_post()* yang berguna mengambil data masukan pada *form* dan menyimpannya ke dalam basis data menggunakan fungsi *_insert(\$member_id, \$data)*. Jika terjadi kesalahan, maka fungsi *error_message()* dijalankan untuk menampilkan pesan kesalahan. *Sequence diagram register member* dapat dilihat pada Lampiran C.5 (*Sequence Diagram Registrasi Member*).

7. *Sequence Diagram Mengubah Data Member*

Sequence diagram mengubah data *member* menjelaskan mengenai proses *member* mengubah data yang telah dimasukkan sebelumnya. Pada *sequence* ini terdapat *class view account* dan *update*, *class controller member*, serta *class model mdl_member*. Fungsi *update(\$member_id)* berguna untuk menampilkan *view update*. Pada *view update* berisi data *member* yang diambil dari basis data menggunakan fungsi

get_data_from_db(). Fungsi *_update(\$member_id, \$data)* digunakan untuk mengubah data pada basis data dengan mengambil data masukan terlebih dahulu menggunakan fungsi *get_data_from_post()* kemudian *_insert(\$member_id, \$data)* untuk mengubah data dalam basis data. Sequence diagram mengubah data member dapat dilihat pada Lampiran C.8 (*Sequence Diagram Mengubah Data Member*).

8. *Sequence Diagram Mengubah Level Member*

Sequence diagram mengubah level member menjelaskan mengenai penggambaran fungsi atau *method* proses *admin* mengubah *level member* yang digunakan pada program yang akan dibuat. Masing-masing *class* akan ditampilkan visual dengan gambar. *Class* yang terdapat pada *sequence diagram* ini yaitu *home*, *manage*, *level*, *adm_member*, dan *mdl_member*.

Proses awal yaitu pada saat aktor memilih menu Manajemen *Member* sehingga menjalankan fungsi *manage()* untuk menampilkan *view manage*. Pada *view manage* aktor kemudian memilih tombol “Ubah *Level Member*” pada salah satu data *member* sehingga sistem menampilkan *view level*, ketika aktor memilih salah satu *level member* dan menekan tombol “*Submit*”, maka sistem menjalankan fungsi *submit()* untuk dapat menjalankan fungsi *_update(\$member_id, \$data)* guna menyimpan data masukan ke basis data melalui fungsi model *_update(\$member_id, \$data)* dan kemudian kembali menampilkan *view manage*. *Sequence diagram mengubah level member* dapat dilihat pada Lampiran C.9 (*Sequence Diagram Mengubah Level Member*).

9. *Sequence Diagram Mengelola Data Produk*

Sequence diagram mengelola data produk menjelaskan mengenai penggambaran fungsi atau *method* proses *admin* menambah, mengubah, dan menghapus data produk yang digunakan pada program yang akan dibuat. Masing-masing *class* akan ditampilkan visual dengan gambar. *Class view* yang terdapat pada *sequence* ini yaitu *home*, *create*, *manage*, *delete_conf*, *additional_options*, *warna_produk*, *upload_pic*, *kategori_produk*; pada *class controller* terdapat *kategori_produk*, *adm_produk*,

adm_warna_produk, dan *adm_kategori_produk*; sedangkan pada model terdapat *class mdl_produk*, *mdl_warna_produk*, dan *mdl_kategori_produk*.

Proses awal yaitu saat aktor memilih menu Manajemen Produk dan submenu *Add Product* maka sistem menjalankan fungsi *create()* sehingga menampilkan *view create*. Pada *view create*, aktor mengisi data produk pada *form* isian dan menekan tombol “*Submit*”, di dalam fungsi *submit()* terdapat fungsi *_insert(\$data)* untuk memasukkan data ke basis data. Sistem kemudian menampilkan *view update* yang berisi *form* yang telah terisi data produk. Pada tampilan *additional_options* aktor dapat memilih “*Update Color*”, “*Upload Pic*”, dan “*Kategori Produk*”. Pada halaman *Update Color* terdapat fungsi *update(\$produk_id)* untuk mengubah warna produk. Pada halaman *Upload Picture* terdapat fungsi *upload_pic(\$produk_id)* dan *do_upload(\$produk_id)* untuk menambahkan gambar produk. Sedangkan pada halaman *Kategori Produk* terdapat fungsi *kategori_produk()* untuk menambahkan kategori pada produk. Jika ingin mengubah data produk tanpa menambah data produk terlebih dahulu, aktor dapat memilih submenu “*Update Product*” yang di dalamnya terdapat fungsi *manage()* untuk menampilkan halaman *Update Product*. Fungsi *delete_produk(\$update_id)* digunakan untuk menghapus data pada basis data. Sequence diagram mengelola data produk dapat dilihat pada Lampiran 0 (*Sequence Diagram Mengelola Data Produk*).

10. *Sequence Diagram Mengelola Kategori*

Sequence diagram mengelola kategori menjelaskan mengenai penggambaran fungsi atau *method* proses *admin* menambah, mengubah, dan menghapus data kategori yang digunakan pada program yang akan dibuat. Masing-masing *class* akan ditampilkan visual dengan gambar. *Class* yang terdapat pada *sequence* ini berupa *view*, *controller*, dan *model*. Pada *view* terdapat *class home*, *manage*, *create*, dan *delete_conf*; *class controller* terdapat *adm_kategori*; dan *class model* terdapat *mdl_kategori*.

Proses berawal dari aktor memilih menu Manajemen Produk dan submenu *Categories* sehingga *controller* menjalankan fungsi *manage()* yang berguna untuk menampilkan *view manage*. Fungsi *_display_kategori_table()* digunakan untuk

menampilkan data kategori produk berupa tabel. Fungsi *create()* digunakan untuk menambah kategori baru yang kemudian disimpan ke dalam basis data menggunakan fungsi *_insert(\$data)*. Untuk mengubah data digunakan pula fungsi *create()*, namun mengambil data dari basis data dahulu sebelum ditampilkan dengan menggunakan fungsi *get_data_from_db()*. Sedangkan jika ingin menghapus data menggunakan fungsi *delete_kategori(\$kategori_id)* untuk menampilkan halaman konfirmasi hapus data, kemudian saat tombol “Yes” ditekan, maka fungsi *_delete(\$kategori_id)* dijalankan untuk menghapus data kategori dalam basis data. Sequence diagram mnegelola kategori produk dapat dilihat pada Lampiran C.11 (*Sequence Diagram Mengelola Kategori*).

11. *Sequence Diagram Mengelola Data Order*

Sequence diagram mengelola data *order* menjelaskan mengenai proses *member* melakukan pemesanan online pada sistem informasi penjualan GFO. Pada *sequence* diagram ini terdapat *view landing_page*, *show_produk*, *cat_contents*, alamat, pembayaran, dan *confirm_selesai*; *class controller mem_produk* dan *mem_cart*; serta *class* model *mdl_produk* dan *mdl_cart*.

Proses *sequence* ini berawal dari fungsi *landing_page()* yang berfungsi menampilkan *view landing_page*, *view* ini berisi deretan data produk sesuai dengan kategorinya yang dijalankan dengan fungsi *products_landing_page()*. Fungsi *show(\$produk_id)* berfungsi menampilkan data satu produk yang dipilih *member*. Data produk diambil melalui *mdl_produk* menggunakan fungsi *data_produk()*.

Saat *member* menekan tombol “Beli Sekarang”, maka sistem menjalankan fungsi *add()* pada *controller mem_cart()* sehingga data produk ditambahkan pada *session cart* menggunakan fungsi *insert_cart_session()*. Fungsi *update(\$rowid)* berguna untuk mengubah *quantity* produk yang dibeli. Fungsi *delete(\$rowid)* berguna untuk menghapus data salah satu produk yang batal dibeli. Pada tombol “Trash” terdapat fungsi *trash()* yang di dalamnya terdapat fungsi *session_destroy()* untuk menghapus *session* dan mengosongkan keranjang belanja. Fungsi *alamat()* berguna untuk

menampilkan *view* alamat yang berisi data alamat *member* yang diambil dari basis data menggunakan fungsi *get_data_from_db()*. Untuk mengubah data alamat *member* dijalankan fungsi *submit()* untuk menjalankan fungsi *get_data_from_post()* yang berguna mengambil data masukan dari form, data masukan tersebut kemudian disimpan ke dalam basis data dengan mengubah data lama menggunakan fungsi *_update(\$member_id, \$data)*. Fungsi *pembayaran()* digunakan untuk menampilkan *view* pembayaran yang berisi data alamat dan *cart contents*. Tahap terakhir yaitu menyimpan data *order* ke dalam basis data menggunakan fungsi *save_order()* untuk menjalankan fungsi *get_data_from_post()* yang berguna mengambil data masukan dan kemudian disimpan ke dalam basis data menggunakan fungsi *_insert(\$order_id, \$data)*. Sequence diagram mengelola data order dapat dilihat pada Lampiran C.12 (*Sequence Diagram Mengelola Data Order*).

12. *Sequence Diagram Melihat Daftar Order*

Sequence diagram melihat status *order* menjelaskan mengenai *member* melihat daftar *order* yang disertai perubahan status *order* dan kelengkapan *order*. Pada *sequence diagram* ini terdapat *class view account*, *order_saya*, dan *kelengkapan_order*; *class controller member*; serta model *mdl_member*.

Proses *sequence diagram* melihat status *order* diawali ketika fungsi *order_saya()* dijalankan untuk menampilkan *view order_saya* yang berisi data *order* yang diambil dari basis data menggunakan fungsi *_display_order_table()*. Fungsi *order_detail(\$order_id, \$data)* digunakan untuk menampilkan *view kelengkapan_order* yang berisi tabel data *order detail*. Data *order detail* diambil dari basis data melalui model menggunakan fungsi *order_detail(\$order_id, \$data)*. *Sequence diagram* melihat status order dapat dilihat pada Lampiran 0 (*Sequence Diagram Melihat Daftar Order*).

13. *Sequence Diagram Mengubah Status Order*

Sequence diagram mengubah status *order* menjelaskan mengenai penggambaran fungsi atau method proses *admin* mengubah status *order* yang digunakan pada program yang akan dibuat. Masing-masing *class* akan ditampilkan visual dengan gambar. *Class view* yang terdapat pada *sequence* diagram ini yaitu *home*, *manage*, dan *status*; *class controller* yaitu berupa *adm_order*; dan *class model* yaitu *mdl_order*. Untuk menampilkan halaman Manajemen *Order*, menggunakan fungsi *manage()*. Kemudian fungsi *orders()* digunakan untuk mengambil data *order* dari basis data. Fungsi *status()* adalah untuk menampilkan *view* status. Setelah *admin* memilih status *order*, kemudian memilih tombol “*Submit*” untuk menjalankan fungsi *submit()*, fungsi ini digunakan untuk menjalankan fungsi *_update(\$order_id, \$data)* untuk mengubah data *order* sesuai dengan status yang baru. Setelah itu kembali lagi ke *view* status. *Sequence* diagram mengubah status *order* dapat dilihat pada Lampiran C.14 (*Sequence Diagram Mengubah Status Order*).

14. *Sequence* Diagram Menambah Data Konfirmasi Bayar

Sequence diagram menambah data konfirmasi bayar menjelaskan mengenai proses *member* menambah data konfirmasi bayar setelah melakukan pembayaran. Pada *sequence* diagram ini terdapat *class view account* dan *konfirmasi_bayar*; *class controller mem_cart*; serta *class model mdl_cart*.

Proses ini diawali dengan dijalankannya fungsi *konfirmasi_bayar()* untuk menampilkan *view konfirmasi_bayar*. Fungsi *submit()* digunakan untuk menjalankan fungsi *get_data_from_post()* yang dapat mengambil data masukan dari *form* yang kemudian disimpan ke dalam basis data menggunakan fungsi *_insert(\$order_id, \$data)*. Fungsi *error_message()* dijalankan ketika terjadi kesalahan pada saat menyimpan data. *Sequence* diagram menambah data konfirmasi bayar dapat dilihat pada Lampiran C.15 (*Sequence Diagram Menambah Data Konfirmasi Bayar*).

15. *Sequence* Diagram Mengelola Gambar Landing Page *Member*

Sequence diagram mengelola gambar landing page *member* menjelaskan mengenai gambaran proses *admin* menambah, mengubah, dan menghapus gambar landing page pada halaman *member*. Terdapat *class view home* dan *upload_pic*, *class controller adm_homepage*, dan *class model mdl_homepage*.

Proses berawal dari *admin* memilih menu *Upload Gambar Homepage* sehingga sistem menjalankan fungsi *upload_pic()* untuk menampilkan view *upload_pic*. Fungsi *do_upload()* digunakan untuk menjalankan fungsi *insert(\$data)* yang berguna untuk menambah data gambar pada basis data. Jika data berhasil dimasukkan, maka data tersebut akan muncul pada *view upload_pic*, jika tidak maka akan tampil *error_message()* yang berisi kesalahan pada proses *do_upload()*. Untuk menghapus gambar, digunakan fungsi *ditch(\$gbr_homepage_id)* sehingga sistem dapat menjalankan fungsi *_delete(\$gbr_homepage_id)* untuk menghapus data yang dipilih pada halaman tersebut dari basis data. *Sequence* diagram mengelola gambar landing page dapat dilihat pada Lampiran 0 (*Sequence Diagram Mengelola Gambar Landing Page Member*).

16. *Sequence* Diagram Mengubah Informasi Toko

Sequence diagram mengubah informasi toko menjelaskan mengenai penggambaran proses *admin* mengubah informasi yang terkait dengan GFO. Pada *sequence* ini terdapat *class view home* dan *update*, *class controller adm_about_us*, serta *class model mdl_about_us*.

Proses berawal saat *admin* memilih menu *About Us* sehingga fungsi *update(\$about_us)* dijalankan untuk menampilkan *view update*. Pada *view update* ditampilkan informasi toko yang secara default telah ditentukan sehingga dilakukan pengambilan data dari basis data menggunakan fungsi *get_data_from_db()*. Fungsi *submit()* dijalankan ketika *admin* selesai memasukkan data baru yang kemudian masukan tersebut dibaca menggunakan fungsi *get_data_from_post()* untuk dimasukkan ke dalam basis data menggunakan fungsi *_update(\$id, \$data)*. Jika *admin* tidak jadi mengubah data, maka *admin* tidak perlu menekan tombol “*Submit*”,

melainkan menjalankan fungsi *reset()* dengan menekan tombol “Reset”. Sequence diagram menguabh informasi toko dapat dilihat pada Lampiran C.17 (*Sequence Diagram Mengubah Informasi Toko*).

17. *Sequence Diagram Melihat Informasi Toko*

Sequence diagram melihat informasi toko menjelaskan mengenai penggambaran proses *member* membuka halaman *Store Info* yang berisi informasi seputar GFO. Pada *sequence* ini terdapat *class view account* dan *show_about_us class controller mem_about*; serta *class model mdl_about*.

Proses ini diawali dengan menjalankan fungsi *show_about_us()* untuk menampilkan *view show_about_us* yang berisi data informasi toko yang diambil dari basis data menggunakan fungsi *_show_about_us()*. *Sequence diagram* melihat informasi toko dapat dilihat pada Lampiran C.18 (*Sequence Diagram Melihat Informasi Toko*).

18. *Sequence Diagram Mengubah dan Menghapus Testimonial*

Sequence diagram mengubah dan menghapus testimonial menjelaskan mengenai proses mengubah dan menghapus testimonial. Pada *sequence* ini terdapat *class view home, manage, update, dan delete_conf*; *class controller adm_testimonial*; serta *class model mdl_testimonial*.

Proses ini berawal ketika *admin* memilih menu testimonial sehingga fungsi *manage()* dijalankan untuk menampilkan *view manage*. Fungsi *_display_testi_table()* digunakan untuk menampilkan data testimonial yang diambil dari basis data menggunakan fungsi *get_data_from_db()*. Jika *admin* ingin mengubah testimonial, maka dijalankan fungsi *update()* untuk menampilkan *view update*. Untuk mengubah data testimonial, fungsi *submit()* dijalankan sehingga sistem mengambil data masukan menggunakan *get_data_from_post()* dan mengubah data pada basis data menggunakan fungsi *_update(\$data)*. Untuk menghapus data, digunakan fungsi *delete_testi(\$update_id)* sehingga sistem menampilkan halaman Konfirmasi Hapus.

Jika tombol “Yes” ditekan, maka sistem menjalankan fungsi *_delete(\$testi_id)* untuk menghapus data, sedangkan jika tombol “No” ditekan, maka sistem kembali menjalankan fungsi *manage()*. *Sequence* diagram mengubah dan menghapus testimonial dapat dilihat pada Lampiran C.19 (*Sequence* Diagram Mengubah dan Menghapus Testimonial).

19. *Sequence* Diagram Menambah Testimonial

Sequence diagram menambah testimonial menjelaskan mengenai penggambaran proses *member* menambahkan kesan dan pesan setelah berbelanja di GFO. Pada *sequence* ini terdapat *class view account* dan *create; class controller mem_testimonial;* serta model *mdl_testimonial*.

Proses ini diawali ketika *member* menekan tombol “Tambah Testimonial” sehingga fungsi *create()* dijalankan untuk menampilkan *view create*. Fungsi *submit()* berguna untuk menjalankan fungsi *get_data_from_post()* yang dapat mengambil data masukan dari *form* yang kemudian disimpan ke dalam basis data menggunakan fungsi *_insert(\$testi_id, \$data)*. Jika terjadi kesalahan pada saat menyimpan data, maka fungsi *error_message()* akan dijalankan untuk menampilkan pesan kesalahan. Fungsi *reset()* digunakan jika ingin mengembalikan *form* dalam keadaan kosong. *Sequence* diagram menambah testimonial dapat dilihat pada Lampiran C.20 (*Sequence* Diagram Menambah Testimonial).

20. *Sequence* Diagram Melihat Testimonial

Sequence diagram melihat testimonial menjelaskan mengenai penggambaran proses *member* melihat seluruh testimonial yang pernah ditambahkan oleh seluruh *member* GFO. Pada *sequence* ini terdapat *class view account* dan *view_all_testi; class controller mem_testimonial;* serta *class* model *mdl_testimonial*.

Proses ini diawali dengan dijalankannya fungsi *view_all_testi()* yang berfungsi untuk menampilkan *view view_all_testi* yang berisi seluruh data testimonial yang diambil dari basis data menggunakan fungsi *_view_all_testi()*. *Sequence* diagram

melihat testimonial dapat dilihat pada Lampiran C.21 (*Sequence Diagram Melihat Testimonial*).

21. *Sequence Diagram Melihat Halaman Cara Order*

Sequence diagram melihat halaman cara *order* menjelaskan mengenai penggambaran proses *member* melihat tahap-tahap cara *order* pada halaman Cara Order. Pada *sequence* ini terdapat *class view account* dan *cara_order*, serta *class controller mem_about_us*. Untuk melihat cara *order* perlu dijalankan fungsi *cara_order()* sehingga sistem akan menampilkan *view cara_order*. *Sequence diagram* melihat halaman Cara Order dapat dilihat pada Lampiran C.22 (*Sequence Diagram Melihat Halaman Cara Order*).

4.3.6 *Class Diagram*

Class diagram merupakan diagram UML yang menggambarkan kelas-kelas dalam sebuah sistem dan hubungannya antara satu dengan yang lain, serta dimasukkan pula atribut dan operasi. Berdasarkan *sequence diagram* yang telah dibangun, *class diagram* sistem informasi penjualan GFO terdiri dari 20 *package* dan berisi 20 *controller*, 20 *model*, serta 65 *view* seperti yang dapat dilihat pada Gambar 4.5. *Class diagram* ini menggunakan relasi *dependency* antar *class*-nya karena *class controller* membutuhkan *class model* untuk mendapatkan data dari database, begitu *class controllers* dan *class views* pada masing-masing *package* membutuhkan *class loader* CI untuk dapat ditampilkan sebagai *interface*. *Package-package* yang terdapat pada *class diagram* sistem informasi penjualan GFO yaitu *package admin*, *package adm_dashboard*, *package adm_produk*, *package adm_paket_produk*, *package adm_order*, *package adm_kategori_produk*, *package adm_about_us*, *package adm_kategori*, *package adm_testimonial*, *package adm_paket*, *package ceo*, *package ceo_produk*, *package warna_produk*, *package member*, *package mem_homepage*, *package mem_cart*, *package mem_produk*, *package mem_warna_produk*, *package mem_testimonial*, *package mem_about_us*, *package template*.

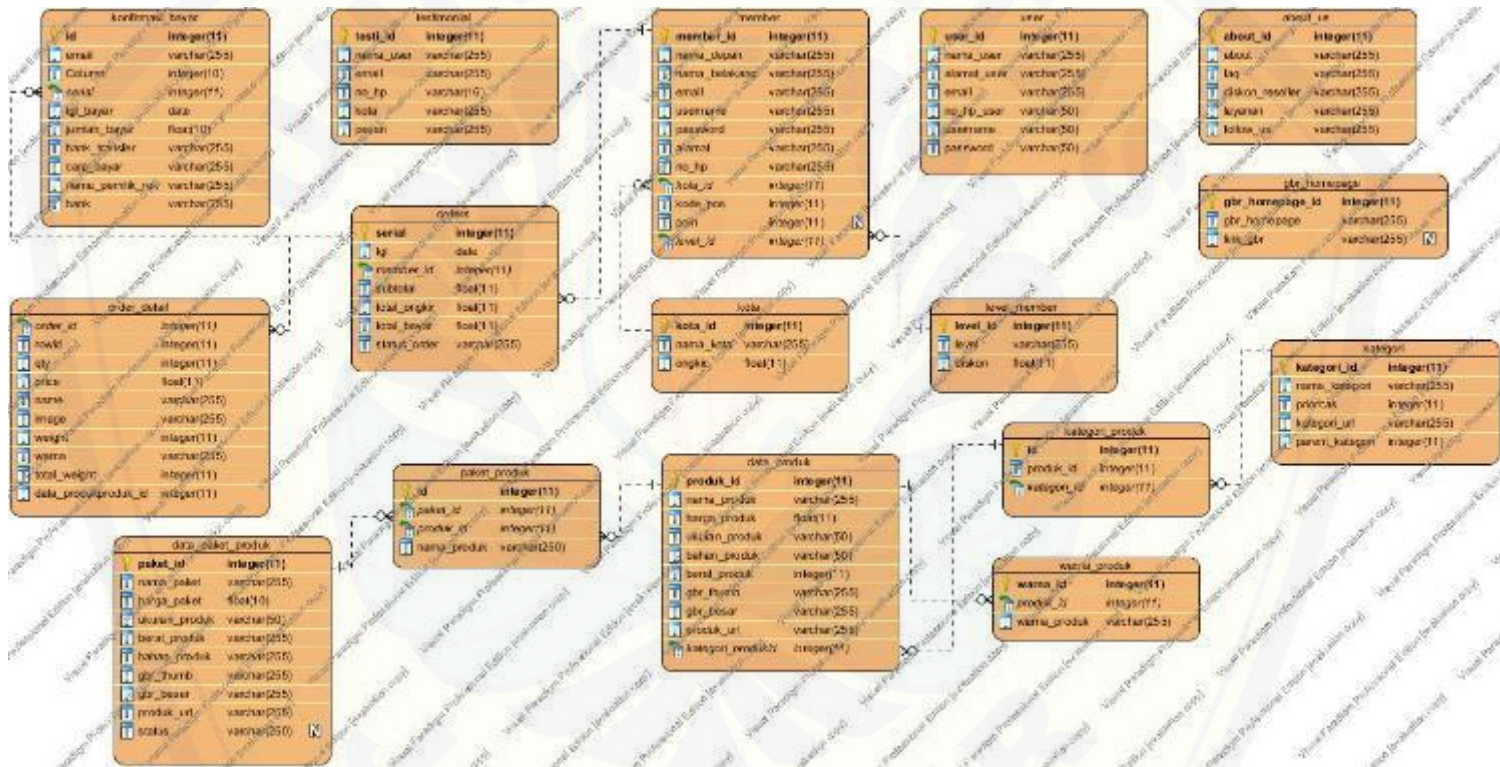




Gambar 4.5 *Class Diagram* Sistem Informasi Penjualan GFO

4.3.7 Entity Relationship Diagram

Entity Relationship Diagram (ERD) pada sistem informasi penjualan GFO menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. ERD yang diimplementasikan pada sistem ini terdiri dari 15 entitas seperti yang dapat dilihat pada Gambar 4.6.



Gambar 4.6 Entity Relationship Diagram Sistem Informasi Penjualan GFO

4.4 Penulisan Kode Program

Tahap kedua pada perancangan model Prototype yaitu membangun prototype. Hal ini dilakukan setelah penggambaran desain sistem. Desain yang telah dibuat kemudian diimplementasikan ke dalam kode program. Beberapa hal yang dilakukan dalam tahap implementasi antara lain:

1. Penulisan kode program menggunakan bahasa pemrograman *Page Hyper Text Pre-Processor* (PHP) dengan bantuan *framework Code Igniter* (CI).
2. Manajemen basis data menggunakan DBMS MySQL.

Kode program perhitungan algoritma Apriori terletak pada *libraries My_lib* dan fungsi-fungsi tambahannya terdapat pada *class adm_paket* dalam *folder controller* modul *adm_paket*. Penulisan kode program perhitungan algoritma tersebut dapat dilihat pada Tabel 4.4 hingga Tabel 4.12.

Tabel 4.4 Kode Program *My_lib* Fungsi *setData()*

21	<code>public function setData(\$MaxScan = ", \$MinSup = ", \$MinConf = ", \$Delimiter = ',') {</code>
22	<code> \$this->maxPhase = \$MaxScan;</code>
23	<code> \$this->minSup = \$MinSup;</code>
24	<code> \$this->minConf = \$MinConf;</code>
25	<code> \$this->delimiter = \$Delimiter;</code>
26	<code>}</code>

Kode program pada Tabel 4.4 merupakan fungsi *setData()* yang digunakan untuk mengambil parameter masukan yang ditentukan oleh *admin*, sehingga nantinya dapat digunakan untuk perhitungan algoritma Apriori.

Tabel 4.5 Kode Program *My_lib* Fungsi *makeTable()*

28	<code>private function makeTable(\$db) {</code>
29	<code> \$table = \$array = array();</code>
30	<code> \$count = 1;</code>
31	
32	<code> for (\$i = 0, \$num = count(\$db); \$i < \$num; \$i++) {</code>
33	<code> \$tmp = \$db[\$i];</code>
34	<code> \$x = array();</code>
35	<code> for (\$j = 0, \$num1 = count(\$tmp); \$j < \$num1; \$j++) {</code>


```
36     $x = trim($tmp[$j]);
37     if ($x === "") {
38         continue;
39     }
40
41     if (!isset($this->keys['v->k'][$x])) {
42         $this->keys['v->k'][$x] = $x;
43         $this->keys['k->v'][$counter] = $x;
44         $counter++;
45     }
46
47     if (!isset($array[$this->keys['v->k'][$x]])) {
48         $array[$this->keys['v->k'][$x]] = 1;
49         $this->allsups[$this->keys['v->k'][$x]] = 1;
50     } else {
51         $array[$this->keys['v->k'][$x]] ++;
52         $this->allsups[$this->keys['v->k'][$x]] ++;
53     }
54
55     $table[$i][$this->keys['v->k'][$x]] = 1;
56 }
57 }
58
59 $tmp = array();
60 foreach ($array as $item => $sup) {
61     if ($sup >= $this->minSup) {
62         $tmp[] = array($item);
63     }
64 }
65
66 $this->allthings[$this->phase] = $tmp;
67 $this->table = $table;
68 }
```

Kode program pada Tabel 4.5 menjelaskan mengenai fungsi *makeTable()* yang digunakan untuk mengambil data transaksi dari basis data, data ini kemudian dihitung masing-masing minimum *support*nya, jika data tersebut memenuhi nilai minimum *support* yang telah ditentukan, maka data tersebut akan dihitung lagi sebagai *itemset*

berikutnya sehingga terbentuk *k-itemset*. Proses perhitungan dilakukan berulang sehingga untuk data yang tidak memenuhi nilai minimum *support* yang telah ditentukan, data tersebut akan dihilangkan atau tidak termasuk *itemset* berikutnya untuk dihitung kembali.

Tabel 4.6 Kode Program *My_lib* Fungsi *scan()*

```
70 private function scan($arr, $implodeArr = ") {
71     $scr = 0;
72
73     if ($implodeArr) {
74         if (isset($this->allsups[$implodeArr])) {
75             return $this->allsups[$implodeArr];
76         }
77     } else {
78         sort($arr);
79         $implodeArr = implode($this->delimiter, $arr);
80         if (isset($this->allsups[$implodeArr])) {
81             return $this->allsups[$implodeArr];
82         }
83     }
84
85     for ($i = 0, $num = count($this->table); $i < $num; $i++) {
86         $bool = true;
87         for ($j = 0, $num1 = count($arr); $j < $num1; $j++) {
88             if (!isset($this->table[$i][$arr[$j]])) {
89                 $bool = false;
90                 break;
91             }
92         }
93
94         if ($bool) {
95             $scr++;
96         }
97     }
98
99     $this->allsups[$implodeArr] = $scr;
100 }
```

101	return \$cr;
102	}

Kode program pada Tabel 4.6 menjelaskan mengenai fungsi *scan()* yang digunakan untuk menyatukan beberapa data sebagaimana fungsi *implode* pada PHP. Data yang disatukan didapat dari data yang telah dihitung minimum *supportnya* seperti pada Tabel 4.5. Data yang tadinya berurutan kemudian dijadikan *array* dengan menggunakan pembatas yang telah ditentukan pada parameter *\$delimiter*. Hal ini bertujuan untuk memudahkan fungsi selanjutnya yang dijelaskan pada Tabel 4.7.

Tabel 4.7 Kode Program *My_lib* Fungsi *combine()*

104	private function combine(\$arr1, \$arr2) {
105	\$result = array();
106	
107	for (\$i = 0, \$num = count(\$arr1); \$i < \$num; \$i++) {
108	if (!isset(\$result['k'][\$arr1[\$i]])) {
109	\$result['v'][] = \$arr1[\$i];
110	\$result['k'][\$arr1[\$i]] = 1;
111	}
112	}
113	
114	for (\$i = 0, \$num1 = count(\$arr2); \$i < \$num1; \$i++) {
115	if (!isset(\$result['k'][\$arr2[\$i]])) {
116	\$result['v'][] = \$arr2[\$i];
117	\$result['k'][\$arr2[\$i]] = 1;
118	}
119	}
120	
121	return \$result['v'];
122	}

Kode program pada Tabel 4.7 merupakan fungsi *combine()* yang digunakan untuk mengkombinasikan hasil *array* yang dibentuk pada fungsi *scan()* yang dijelaskan

pada Tabel 4.6. Fungsi *combine()* bertujuan untuk menyatukan hasil data yang memenuhi minimum *support* pada *k-itemset*.

Tabel 4.8 Kode Program *My_lib* Fungsi *realName()*

```

124 private function realName($arr) {
125     $result = "";
126
127     for ($j = 0, $num = count($arr); $j < $num; $j++) {
128         if ($j) {
129             $result .= $this->delimiter;
130         }
131
132         $result .= $this->mdl_paket->ambilnama($arr[$j]);
133     }
134
135     return $result;
136 }

```

Kode program pada Tabel 4.8 merupakan fungsi *realName()* yang digunakan untuk mengambil nama produk dari basis data. Nama yang diambil ini kemudian ditampilkan sesuai dengan hasil *array* yang didapat, kemudian diberi pembatas sesuai parameter *\$delimiter* yang telah ditentukan.

Tabel 4.9 Kode Program *My_lib* Fungsi *checkRule()*

```

138 private function checkRule($a, $b) {
139     for ($i = 0, $a_num = count($a); $i < $a_num; $i++) {
140         for ($j = 0, $b_num = count($b); $j < $b_num; $j++) {
141             if ($a[$i] == $b[$j]) {
142                 return false;
143             }
144         }
145     }
146
147     return true;

```

148	}
-----	---

Kode program pada Tabel 4.9 menjelaskan tentang fungsi *checkRule()* yang digunakan untuk memeriksa kesesuaian aturan, apakah jumlah data *k-itemset* tidak kurang dari jumlah transaksi. Jika hasil penggabungan *array* sesuai dengan aturan, maka dapat dilanjutkan ke proses selanjutnya, jika tidak maka tidak dapat dilanjutkan ke proses selanjutnya.

Tabel 4.10 Kode Program *My_lib* Fungsi *confidence()*

150	private function <i>confidence</i> (\$sup_a, \$sup_ab) {
151	return round((\$sup_ab / \$sup_a) * 100, 2);
152	}
153	
154	private function <i>subsets</i> (\$items) {
155	\$result = array();
156	for (\$i = 0, \$num = count(\$items), \$members = pow(2, \$num); \$i < \$members; \$i++)
157	{
158	\$b = sprintf("%0" . \$num . "b", \$i);
159	\$tmp = array();
160	for (\$j = 0; \$j < \$num; \$j++) {
161	if (\$b[\$j] == '1') {
162	\$tmp[] = \$items[\$j];
163	}
164	}
165	
166	if (\$tmp) {
167	sort(\$tmp);
168	\$result[] = \$tmp;
169	}
170	}
171	
172	return \$result;
173	}

Kode program pada Tabel 4.10 menjelaskan mengenai fungsi *confidence()* yang digunakan untuk menghitung nilai *confidence* masing-masing data sesuai dengan minimum *confidence* yang telah ditentukan. Perhitungan nilai *confidence* sesuai dengan rumus perhitungan *confidence* pada algoritma Apriori. Hasil nilai *confidence* masing-masing data ditampilkan dalam bentuk persen.

Tabel 4.11 Kode Program *My_lib* Fungsi *freqItemset()*

```
175 private function freqItemsets($db) {
176     $this->makeTable($db);
177     while (1) {
178         if ($this->phase >= $this->maxPhase) {
179             break;
180         }
181
182         $scr = 0;
183         for ($i = 0, $num = count($this->allthings[$this->phase]); $i < $num; $i++) {
184             for ($j = $i; $j < $num; $j++) {
185                 if ($i == $j) {
186                     continue;
187                 }
188
189                 $item = $this->combine($this->allthings[$this->phase][$i], $this->allthings[$this->phase][$j]);
190                 sort($item);
191                 $implodeArr = implode($this->delimiter, $item);
192                 if (!isset($this->freqItemsets[$implodeArr])) {
193                     $sup = $this->scan($item, $implodeArr);
194                     if ($sup >= $this->minSup) {
195                         $this->allthings[$this->phase + 1][] = $item;
196                         $this->freqItemsets[$implodeArr] = 1;
197                         $scr++;
198                     }
199                 }
200             }
201         }
202
203         if ($scr <= 1) {
```

```

204         break;
205     }
206
207     $this->phase++;
208 }
209
210 foreach ($this->freqItemsts as $k => $v) {
211     $sarr = explode($this->delimiter, $k);
212     $num = count($sarr);
213     if ($num >= 3) {
214         $subsets = $this->subsets($sarr);
215         $num1 = count($subsets);
216         for ($i = 0; $i < $num1; $i++) {
217             if (count($subsets[$i]) < $num) {
218                 unset($this->freqItemsts[implode($this->delimiter, $subsets[$i])]);
219             } else {
220                 break;
221             }
222         }
223     }
224 }
225 }

```

Kode program pada Tabel 4.11 menjelaskan tentang fungsi *freqItemset()* yang digunakan untuk membuat *itemset* yang sudah memenuhi semua aturan yang dihitung pada fungsi-fungsi sebelumnya. Pada fungsi ini dihitung seberapa sering kemunculan *itemset* tersebut terhadap seluruh transaksi.

Tabel 4.12 Kode Program *My_lib* Fungsi *process()*

```

227 public function process($db) {
228     $checked = $result = array();
229
230     $this->freqItemsets($db);
231
232     foreach ($this->freqItemsts as $k => $v) {
233         $subsets = $this->subsets(explode($this->delimiter,$k));

```

```

234
235     for ($i = 0, $num = count($subsets); $i < $num; $i++) {
236         for ($j = 0; $j < $num; $j++) {
237             if ($this->checkRule($subsets[$i], $subsets[$j])) {
238                 $n1 = $this->realName($subsets[$i]);
239                 $n2 = $this->realName($subsets[$j]);
240
241                 $scan = $this->scan($this->combine($subsets[$i], $subsets[$j]));
242                 $c1 = $this->confidence($this->scan($subsets[$i]), $scan);
243                 $c2 = $this->confidence($this->scan($subsets[$j]), $scan);
244
245                 if ($c1 >= $this->minConf) {
246                     $result[$n1][$n2] = $c1;
247                 }
248
249                 if ($c2 >= $this->minConf) {
250                     $result[$n2][$n1] = $c2;
251                 }
252
253                 $checked[$n1 . $this->delimiter . $n2] = 1;
254                 $checked[$n2 . $this->delimiter . $n1] = 1;
255             }
256         }
257     }
258 }
259
260 return $this->rules = $result;
261 }

```

Kode program pada Tabel 4.12 merupakan fungsi *process()* yang digunakan untuk memproses beberapa fungsi sebelumnya terhadap data yang diambil dari database. *Itemset-itemset* dihitung sesuai aturan asosiasi dan diambil yang memenuhi nilai minimum *support* dan *confidence*.

Tabel 4.13 Kode Program *My_lib* Fungsi *getFreqItemset()*

```

263 public function getFreqItemsets() {

```


264	\$result = array();
265	
266	foreach (\$this->freqItemsts as \$k => \$v) {
267	\$tmp = array();
268	\$tmp['sup'] = \$this->allsups[\$k];
269	\$k = explode(\$this->delimiter, \$k);
270	
271	for (\$i = 0, \$num = count(\$k); \$i < \$num; \$i++) {
272	\$tmp[] = \$this->realName(\$k[\$i]);
273	}
274	
275	\$result[] = \$tmp;
276	}
277	
278	return \$result;
279	}

Kode program pada Tabel 4.13 merupakan fungsi *getFreqItemset()* yang digunakan untuk menampilkan hasil dari fungsi *freqItemset()* yang telah dijalankan sebelumnya.

Tabel 4.14 Kode Program *My_lib* Fungsi *getAssociationRules()*

281	public function getAssociationRules() {
282	return \$this->rules;
283	}

Setelah semua fungsi dijalankan, tahap terakhir yang menampilkan hasil aturan asosiasi final yang telah memenuhi semua aturan yang telah ditentukan dengan menggunakan fungsi *getAssociationRules()* seperti yang dijelaskan pada Tabel 4.14.

4.5 Pengujian Sistem

Tahap terakhir pada model Prototype yaitu menguji coba sistem. Pada penelitian ini peneliti menggunakan dua metode pengujian sistem yaitu *Black box Testing* dan *White box Testing*. Berikut adalah hasil dari pengujian sistem:

4.5.1. Pengujian *White Box*

Pengujian *white box* digunakan untuk menjamin operasi-operasi internal sesuai dengan spesifikasi yang telah ditetapkan dengan menggunakan struktur kendali dari prosedur yang dirancang. Pengujian *white box* yang dilakukan pada penelitian ini meliputi *Listing* program, grafik alir, kompleksitas siklomatis, pengujian jalur program, dan *test case*. Pengujian ini dilakukan pada fitur menentukan rekomendasi paket produk menggunakan algoritma Apriori. Hasil pengujian yang telah dilakukan adalah sebagai berikut:

1. Pengujian *White box* Menentukan Rekomendasi Paket Produk

Pengujian *white box* menentukan rekomendasi paket produk meliputi input nilai minimum *support*, minimum *confidence*, dan maksimal data transaksi yang digunakan, serta menampilkan aturan asosiasi final sebagai rekomendasi paket produk.

a. *Listing* Program Fitur Menentukan Rekomendasi Paket Produk

Listing program dibuat untuk dapat memudahkan peneliti dalam melakukan pengujian pada *cyclomatic complexity*. *Listing* program pada fitur menentukan rekomendasi paket produk dibuat untuk mengetahui penomoran pada blok-blok program yang kemudian dapat digunakan untuk pengujian tahap selanjutnya. Contoh pembuatan *listing* program dapat dilihat pada Gambar 4.7.

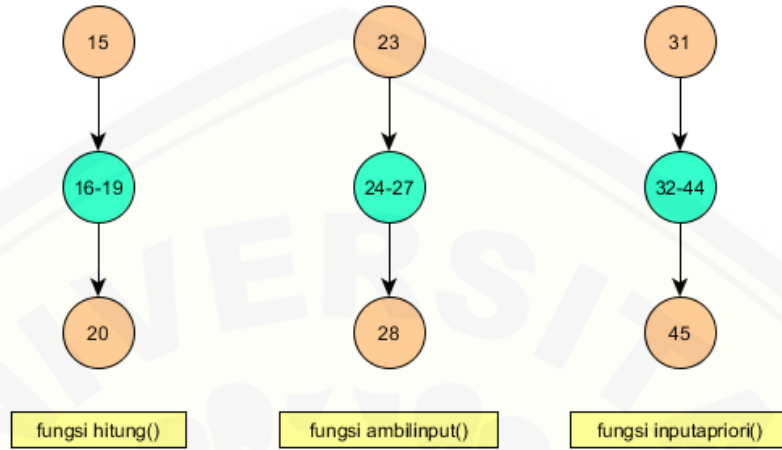
1) *Listing* Program Fungsi Menampilkan Halaman Masukan Nilai Kriteria

```
14 // halaman input
15 public function hitung() { } 1
16     $template = "admin";
17     $data['view_file'] = "hitung";
18     $this->load->module('template');
19     $this->template->$template($data);
20 } } 2
    } } 3
```

Gambar 4.7 *Listing* Program Menampilkan Halaman Masukan Kriteria

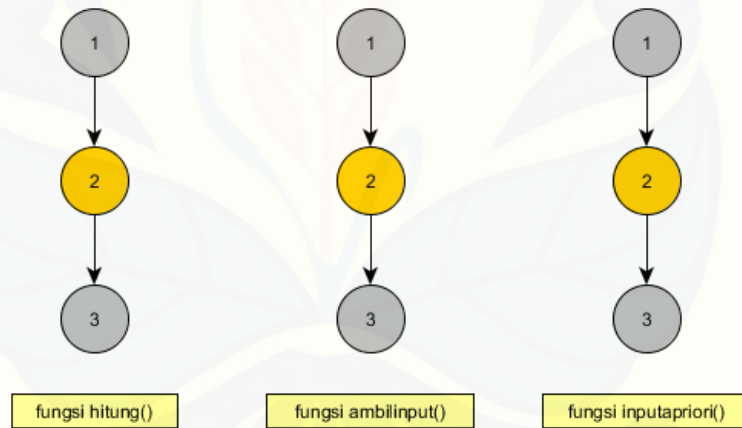
Listing program lain selengkapnya dapat dilihat pada Lampiran E (*Listing* Program Fitur Menentukan Rekomendasi Paket Produk).

b. Diagram Alir Fitur Menentukan Rekomendasi Paket Produk



Gambar 4.8 Diagram Alir Fitur Menentukan Rekomendasi Paket Produk

c. Grafik Alir Fitur Menentukan Rekomendasi Paket Produk



Gambar 4.9 Grafik Alir Fitur Menentukan Rekomendasi Paket Produk

d. Kompleksitas Siklometik

Grafik alir pada Gambar 4.9 dapat dihitung kompleksitas siklomatik-nya menggunakan rumus sebagai berikut:

$$V(G) = E - N + 2$$

Dimana:

E = jumlah *edge* grafik alir

N = jumlah *node* grafik alir

Sehingga perhitungan kompleksitas siklomatik pada Gambar 4.9 yaitu:

Fungsi hitung() $V(G) = E - N + 2 = 2 - 3 + 2 = 1$

Fungsi ambilinput() $V(G) = E - N + 2 = 2 - 3 + 2 = 1$

Fungsi inputapriori() $V(G) = E - N + 2 = 2 - 3 + 2 = 1$

e. Pengujian Jalur Program Fitur Menentukan Rekomendasi Paket Produk

Pengujian jalur program fitur menentukan rekomendasi paket produk berdasarkan Gambar 4.9 adalah sebagai berikut:

1) Fungsi hitung():

Jalur : 1 - 2 - 3

2) Fungsi ambilinput():

Jalur : 1 - 2 - 3

3) Fungsi inputapriori():

Jalur : 1 - 2 - 3

f. Test Case Fitur Menentukan Rekomendasi Paket Produk

Test Case Fungsi hitung()	
Test Case	Berhasil menampilkan halaman masukan nilai kriteria
Target yang Diharapkan	Menampilkan halaman masukan nilai kriteria pada <i>Admin Panel</i>
Hasil Pengujian	Benar
Jalur	1 – 2 – 3
Test Case Fungsi ambilinput()	
Test Case	Berhasil mengambil nilai kriteria
Target yang Diharapkan	Mengambil data nilai kriteria dari form masukan nilai kriteria
Hasil Pengujian	Benar
Jalur	1 – 2 – 3
Test Case Fungsi inputapriori()	
Test Case	Berhasil menghitung algoritma Apriori dan menampilkan halaman hasil
Target yang diharapkan	Sistem menghitung algoritma Apriori menggunakan <i>library My_lib</i> yang sesuai dengan data masukan nilai kriteria dan kemudian menampilkan halaman hasil yang berisi hasil aturan asosiasi final
Hasil Pengujian	Benar
Jalur	1 – 2 – 3

2. Pengujian *White Box Library* Algoritma Apriori *My_lib*

Pengujian *white box* pada fungsi *library My_lib* meliputi penentuan data nilai kriteria, pembuatan tabel data perhitungan *support* masing-masing data, penggabungan data satu dengan lainnya yang memenuhi nilai minimum *support* menjadi sebuah array, mengkombinasikan *array* satu dengan *array* lain, mengubah *id* produk yang digunakan ke nama produk untuk ditampilkan, memeriksa kembali kesesuaian data terhadap aturan asosiasi, menghitung nilai *confidence* masing-masing data, menentukan *frequent itemset*, memproses data masukan, mengambil *frequent itemset*, dan mengambil hasil aturan asosiasi final.

a. *Listing Program Library Algoritma Apriori My_lib*

Adanya *listing* program pada library algoritma Apriori membuat pengerjaan *cyclomatic complexity* lebih mudah karena penomoran masing-masing blok program telah ditentukan. Salah satu pembuatan *listing* program pada library algoritma Apriori dapat dilihat pada Gambar 4.10. *Listing* program pada blok kode lain selengkapnya dapat dilihat pada Lampiran F (*Listing Program Library Algoritma Apriori My_lib*).

1) *Listing Program Fungsi Penentuan Nilai Kriteria*

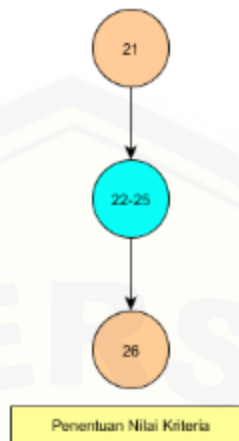
```

21 public function setData($MaxScan = '', $MinSup = '', $MinConf = '', $Delimiter = ',') {1
22     $this->maxPhase = $MaxScan;
23     $this->minSup = $MinSup;
24     $this->minConf = $MinConf;
25     $this->delimiter = $Delimiter;
26 }3
  
```

Gambar 4.10 Listing Program Fungsi Penentuan Nilai Kriteria

b. Diagram Alir *Library Algoritma Apriori My_lib*

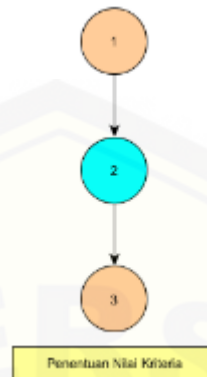
Diagram alir *library* algoritma Apriori pada *class my_lib* meliputi fitur penentuan nilai kriteria, penentuan tabel data perhitungan Apriori, penggabungan data menjadi *array*, mengkombinasikan *array*, mengubah nama variabel menjadi nama produk, memeriksa kembali kesesuaian data dengan aturan asosiasi, menghitung nilai *confidence*, mengurutkan data sesuai nilai *support* dan *confidence*, menentukan *frequent itemset*, proses data masukan, mengambil hasil *frequent itemset*, dan mengambil hasil aturan asosiasi final. Salah satu penggambaran diagram alir *library* algoritma Apriori pada *class my_lib* dapat dilihat pada Gambar 4.11. Diagram alir selengkapnya dapat dilihat pada Lampiran G (Diagram Alir *Library Algoritma Apriori My_lib*).



Gambar 4.11 Contoh Penggambaran Diagram Alir *Library* Algoritma Apriori *My_lib*

c. Grafik Alir *Library* Algoritma Apriori *My_lib*

Guna penggambaran grafik alir ini adalah untuk memudahkan mengetahui runtutan alir program dan jalur kerja program. Grafik alir *library* algoritma Apriori pada *class my_lib* meliputi fitur penentuan nilai kriteria, penentuan tabel data perhitungan Apriori, penggabungan data menjadi *array*, mengkombinasikan *array*, mengubah nama variabel menjadi nama produk, memeriksa kembali kesesuaian data dengan aturan asosiasi, menghitung nilai *confidence*, mengurutkan data sesuai nilai *support* dan *confidence*, menentukan *frequent itemset*, proses data masukan, mengambil hasil *frequent itemset*, dan mengambil hasil aturan asosiasi final. Salah satu penggambaran grafik alir *library* algoritma Apriori pada *class my_lib* dapat dilihat pada Gambar 4.12. Diagram alir selengkapnya dapat dilihat pada Lampiran G (Diagram Alir *Library* Algoritma Apriori *My_lib*).



Gambar 4.12 Contoh Penggambaran Grafik Alir *Library* Algoritma Apriori *My_lib*

d. Kompleksitas Siklometik

Grafik alir pada Gambar 4.25 dapat dihitung kompleksitas siklomatik-nya menggunakan rumus sebagai berikut:

$$V(G) = E - N + 2$$

Dimana:

E = jumlah *edge* grafik alir

N = jumlah *node* grafik alir

Sehingga perhitungan kompleksitas siklomatik pada Gambar 4.25 yaitu:

1) Penentuan Nilai Kriteria

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 2 - 3 + 2 \\
 &= 1
 \end{aligned}$$

2) Pembuatan Tabel Data Perhitungan *Support*

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 28 - 20 + 2 \\
 &= 10
 \end{aligned}$$

3) Penggabungan Data Menjadi Array

$$\begin{aligned}V(G) &= E - N + 2 \\ &= 24 - 17 + 2 \\ &= 9\end{aligned}$$

4) Mengkombinasikan Array

$$\begin{aligned}V(G) &= E - N + 2 \\ &= 14 - 10 + 2 \\ &= 6\end{aligned}$$

5) Mengubah Nama Variabel Menjadi Nama Produk

$$\begin{aligned}V(G) &= E - N + 2 \\ &= 11 - 8 + 2 \\ &= 5\end{aligned}$$

6) Memeriksa Kembali Kesesuaian Data dengan Aturan Asosiasi

$$\begin{aligned}V(G) &= E - N + 2 \\ &= 10 - 7 + 2 \\ &= 5\end{aligned}$$

7) Menghitung Nilai *Confidence*

$$\begin{aligned}V(G) &= E - N + 2 \\ &= 2 - 3 + 2 \\ &= 1\end{aligned}$$

- 8) Mengurutkan Data Sesuai Nilai *Support* dan *Confidence*

$$\begin{aligned}V(G) &= E - N + 2 \\ &= 16 - 11 + 2 \\ &= 7\end{aligned}$$

- 9) Menentukan Frequent *Itemset*

$$\begin{aligned}V(G) &= E - N + 2 \\ &= 35 - 27 + 2 \\ &= 10\end{aligned}$$

- 10) Proses Data Masukan

$$\begin{aligned}V(G) &= E - N + 2 \\ &= 19 - 16 + 2 \\ &= 5\end{aligned}$$

- 11) Mengambil Hasil Frequent *Itemset*

$$\begin{aligned}V(G) &= E - N + 2 \\ &= 10 - 9 + 2 \\ &= 3\end{aligned}$$

- 12) Mengambil Hasil Aturan Asosiasi Final

$$\begin{aligned}V(G) &= E - N + 2 \\ &= 2 - 3 + 2 \\ &= 1\end{aligned}$$

- e. Pengujian Jalur Program *Library* Algoritma Apriori *My_lib*

Pengujian jalur program *library* untuk algoritma Apriori pada class *My_lib.php* berdasarkan Gambar 4.25 adalah sebagai berikut:

1) Penentuan Nilai Kriteria

Jalur : 1 – 2 – 3

2) Pembuatan Tabel Data Perhitungan *Support*

Jalur 1 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 14 – 15 – 16 –
17 – 18 – 19 – 20

Jalur 2 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 15 – 16 – 17 – 18 – 19 – 20

Jalur 3 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 15 – 16 – 17 – 18 – 19 – 20

Jalur 4 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 11 – 12 – 14 – 15 – 16 – 17 –
18 – 19 – 20

Jalur 5 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 11 – 13 – 14 – 15 – 16 – 17 –
18 – 19 – 20

Jalur 6 : 1 – 2 – 3 – 15 – 16 – 17 – 18 – 19 – 20

Jalur 7 : 1 – 2 – 3 – 4 – 5 – 15 – 16 – 17 – 18 – 19 – 20

Jalur 8 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 14 – 15 – 16 –
19 – 20

Jalur 9 : 1 – 2 – 3 – 15 – 16 – 19 – 20

Jalur 10 : 1 – 2 – 3 – 4 – 5 – 15 – 16 – 19 – 20

3) Penggabungan Data Menjadi Array

Jalur 1 : 1 – 2 – 3 – 4 – 5 – 9 – 10 – 11 – 12 – 13 – 14 – 15 – 16 – 17

Jalur 2 : 1 – 2 – 3 – 4 – 6 – 7 – 8 – 10 – 11 – 12 – 13 – 14 – 15 – 16 – 17

Jalur 3 : 1 – 2 – 3 – 6 – 7 – 8 – 12 – 14 – 15 – 16 – 17

Jalur 4 : 1 – 2 – 3 – 6 – 7 – 9 – 10 – 11 – 12 – 13 – 14 – 15 – 16 – 17

Jalur 5 : 1 – 2 – 3 – 6 – 7 – 9 – 10 – 11 – 12 – 14 – 15 – 16 – 17

Jalur 6 : 1 – 2 – 3 – 4 – 5 – 9 – 16 – 17

Jalur 7 : 1 – 2 – 3 – 4 – 5 – 9 – 10 – 11 – 12 – 13 – 14 – 16 – 17

Jalur 8 : 1 – 2 – 3 – 4 – 5 – 9 – 10 – 11 – 14 – 16 – 17

Jalur 9 : 1 – 2 – 3 – 4 – 5 – 9 – 10 – 11 – 12 – 14 – 16 – 17

- 4) Mengkombinasikan Array
Jalur 1 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10
Jalur 2 : 1 – 2 – 3 – 4 – 6 – 7 – 8 – 9 – 10
Jalur 3 : 1 – 2 – 3 – 6 – 7 – 8 – 9 – 10 (4 kombinasi)
- 5) Mengubah Nama Variabel Menjadi Nama Produk
Jalur 1 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8
Jalur 2 : 1 – 2 – 3 – 7 – 8
Jalur 3 : 1 – 2 – 3 – 4 – 7 – 8
Jalur 4 : 1 – 2 – 3 – 4 – 5 – 7 – 8
Jalur 5 : 1 – 2 – 3 – 4 – 6 – 7 – 8
- 6) Memeriksa Kembali Kesesuaian Data dengan Aturan Asosiasi
Jalur 1 : 1 – 2 – 3 – 4 – 5 – 6 – 7
Jalur 2 : 1 – 2 – 6 – 7
Jalur 3 : 1 – 2 – 3 – 6 – 7
Jalur 4 : 1 – 2 – 3 – 4 – 6 – 7 (2 kombinasi)
- 7) Menghitung Nilai *Confidence*
Jalur : 1 – 2 – 3
- 8) Mengurutkan Data Sesuai Nilai *Support* dan *Confidence*
Jalur 1 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11
Jalur 2 : 1 – 2 – 3 – 10 – 11
Jalur 3 : 1 – 2 – 3 – 4 – 5 – 8 – 9 – 10 – 11
Jalur 4 : 1 – 2 – 3 – 4 – 5 – 6 – 8 – 9 – 10 – 11
Jalur 5 : 1 – 2 – 3 – 4 – 5 – 8 – 10 – 11
Jalur 6 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 10 – 11 (2 kombinasi)

9) Menentukan Frequent *Itemset*

Jalur 1 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 14 – 15 – 16 –
17 – 19 – 20 – 21 – 22 – 23 – 24 – 25 – 27

Jalur 2 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 14 – 15 – 16 –
18 – 19 – 20 – 21 – 22 – 23 – 24 – 25 – 27

Jalur 3 : 1 – 2 – 3 – 19 – 20 – 21 – 22 – 23 – 24 – 25 – 27

Jalur 4 : 1 – 2 – 3 – 4 – 19 – 20 – 21 – 22 – 23 – 24 – 25 – 27

Jalur 5 : 1 – 2 – 3 – 4 – 19 – 20 – 21 – 22 – 23 – 24 – 26 – 25 – 27

Jalur 6 : 1 – 2 – 3 – 4 – 19 – 20 – 21 – 22 – 23 – 27

Jalur 7 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 16 – 18 – 19 – 20 – 21 – 22 – 23 – 24
– 25 – 27

Jalur 8 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 16 – 17 – 19 – 20 – 21 – 22 – 23 – 24
– 25 – 27

Jalur 9 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 16 – 18 – 19 – 20 – 21 – 22 – 23 – 24
– 26 – 25 – 27

Jalur 10 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 16 – 17 – 19 – 20 – 21 – 22 – 23 – 24
– 26 – 25 – 27

10) Proses Data Masukan

Jalur 1 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 13 – 14 – 15 –
16

Jalur 2 : 1 – 2 – 3 – 15 – 16 (2 kombinasi)

Jalur 3 : 1 – 2 – 3 – 4 – 5 – 15 – 16 (2 kombinasi)

11) Mengambil Hasil Frequent *Itemset*

Jalur 1 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9

Jalur 2 : 1 – 2 – 3 – 4 – 5 – 7 – 8 – 9

Jalur 3 : 1 – 2 – 3 – 8 – 9

12) Mengambil Hasil Aturan Asosiasi Final

Jalur : 1 – 2 – 3

f. Test Case Fitur Menentukan Rekomendasi Paket Produk

Test Case Penentuan Nilai Kriteria

Test Case	Berhasil menentukan nilai kriteria
Target yang Diharapkan	Membuat inisialisasi nilai kriteria
Hasil Pengujian	Benar
Jalur	1 – 2 – 3

Test Case Fungsi Pembuatan Tabel Data Penghitungan Nilai Support

Jalur 1

Test Case	Berhasil membuat tabel data hasil perhitungan nilai support pada masing-masing data
Target yang Diharapkan	Membuat tabel data hasil perhitungan nilai support pada masing-masing data
Hasil Pengujian	Benar
Jalur	1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 14 – 15 – 16 – 17 – 18 – 19 – 20

Jalur 2

Test Case	Berhasil membuat tabel data hasil perhitungan nilai support pada masing-masing data
Target yang Diharapkan	Membuat tabel data hasil perhitungan nilai support pada masing-masing data
Hasil Pengujian	Benar
Jalur	1 – 2 – 3 – 4 – 5 – 6 – 7 – 15 – 16 – 17 – 18 – 19 – 20

Jalur 3

Test Case	Berhasil membuat tabel data hasil perhitungan nilai support pada masing-masing data
Target yang Diharapkan	Membuat tabel data hasil perhitungan nilai support pada masing-masing data
Hasil Pengujian	Benar
Jalur	1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 15 – 16 – 17 – 18 – 19 – 20

Jalur 4

Test Case	Berhasil membuat tabel data hasil perhitungan nilai support pada masing-masing data
Target yang Diharapkan	Membuat tabel data hasil perhitungan nilai support pada masing-masing data
Hasil Pengujian	Benar
Jalur	1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 11 – 12 – 14 – 15 – 16 –17 – 18 – 19 – 20

Jalur 5

Test Case	Berhasil membuat tabel data hasil perhitungan nilai support pada masing-masing data
Target yang Diharapkan	Membuat tabel data hasil perhitungan nilai support pada masing-masing data
Hasil Pengujian	Benar
Jalur	1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 11 – 13 – 14 – 15 – 16 –17 – 18 – 19 – 20

Jalur 6

Test Case	Berhasil membuat tabel data hasil perhitungan nilai support pada masing-masing data
Target yang Diharapkan	Membuat tabel data hasil perhitungan nilai support pada masing-masing data
Hasil Pengujian	Benar
Jalur	1 – 2 – 3 – 15 – 16 –17 – 18 –19 – 20

Jalur 7

Test Case	Berhasil membuat tabel data hasil perhitungan nilai support pada masing-masing data
Target yang Diharapkan	Membuat tabel data hasil perhitungan nilai support pada masing-masing data
Hasil Pengujian	Benar
Jalur	1 – 2 – 3 – 4 – 5 – 15 – 16 –17 – 18 –19 – 20

Jalur 8

Test Case	Berhasil membuat tabel data hasil perhitungan nilai support pada masing-masing data
Target yang Diharapkan	Membuat tabel data hasil perhitungan nilai support pada masing-masing data
Hasil Pengujian	Benar
Jalur	1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 14 – 15 – 16 – 19 – 20

Jalur 9

Test Case	Berhasil membuat tabel data hasil perhitungan nilai support pada masing-masing data
Target yang Diharapkan	Membuat tabel data hasil perhitungan nilai support pada masing-masing data
Hasil Pengujian	Benar
Jalur	1 – 2 – 3 – 15 – 16 – 19 – 20

Jalur 10

Test Case	Berhasil membuat tabel data hasil perhitungan nilai support pada masing-masing data
Target yang Diharapkan	Membuat tabel data hasil perhitungan nilai support pada masing-masing data
Hasil Pengujian	Benar
Jalur	1 – 2 – 3 – 4 – 5 – 15 – 16 – 19 – 20

Test Case Mengubah Nama Variabel Menjadi Nama Produk

Jalur 1

Test Case	Berhasil mengubah nama variabel menjadi nama produk
Target yang diharapkan	Nama variabel sementara dapat diubah kembali sesuai dengan nama produk
Hasil Pengujian	Benar
Jalur	1 – 2 – 3 – 4 – 5 – 6 – 7 – 8

Jalur 2

Test Case	Berhasil mengubah nama variabel menjadi nama produk
Target yang diharapkan	Nama variabel sementara dapat diubah kembali sesuai dengan nama produk
Hasil Pengujian	Benar
Jalur	1 – 2 – 3 – 7 – 8

Jalur 3

Test Case	Berhasil mengubah nama variabel menjadi nama produk
Target yang diharapkan	Nama variabel sementara dapat diubah kembali sesuai dengan nama produk
Hasil Pengujian	Benar
Jalur	1 – 2 – 3 – 4 – 7 – 8

Jalur 4

Test Case	Berhasil mengubah nama variabel menjadi nama produk
Target yang diharapkan	Nama variabel sementara dapat diubah kembali sesuai dengan nama produk
Hasil Pengujian	Benar
Jalur	1 – 2 – 3 – 4 – 5 – 7 – 8

Jalur 5

Test Case	Berhasil mengubah nama variabel menjadi nama produk
-----------	---

Target yang diharapkan	Nama variabel sementara dapat diubah kembali sesuai dengan nama produk
Hasil Pengujian	Benar
Jalur	1 – 2 – 3 – 4 – 6 – 7 – 8
Test Case Memeriksa Kembali Kesesuaian Data dengan Aturan Asosiasi	
Test Case	Data yang dihasilkan sesuai dengan aturan asosiasi
Target yang diharapkan	Sistem dapat memeriksa hasil data dengan aturan asosiasi yang seharusnya
Hasil Pengujian	Benar
Jalur	1 – 2 – 3 – 4 – 5 – 6 – 7
Test Case Menghitung Nilai <i>Confidence</i>	
Test Case	Berhasil menghitung nilai confidence
Target yang diharapkan	Nilai confidence masing-masing data dapat dihitung sesuai dengan nilai minimum confidence
Hasil Pengujian	Benar
Jalur	1 – 2 – 3

4.5.2. Pengujian *Black box*

Pengujian *black box* dilakukan untuk mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Pengujian dilakukan pada *form* untuk setiap *usecase*. Pengujian ini dilakukan oleh *developer*. Hasil pengujian *black box* pada *usecase* menentukan rekomendasi paket produk dapat dilihat pada Tabel 4.15.

Tabel 4.15 Hasil Pengujian *Black Box Usecase* Menentukan Rekomendasi Paket Produk

No.	Fitur	Aksi	Hasil	Kesimpulan	
				Berhasil	Gagal
1.	Memasukkan nilai kriteria	Memilih menu Paket	a. Menampilkan kolom teks isian nilai kriteria serta tombol “Submit” dan “Clear”	✓	
Berlanjut					

			Lanjutan
2.	Menentukan dan menampilkan hasil aturan asosiasi final sebagai rekomendasi paket produk	Mengisi kolom teks nilai kriteria sesuai yang diinginkan, kemudian klik tombol "Submit"	<p>a. Menghitung nilai masukan kriteria terhadap data transaksi menggunakan algoritma Apriori.</p> <p>b. Menentukan hasil aturan asosiasi final</p> <p>c. Menampilkan hasil aturan asosiasi final sebagai rekomendasi paket produk pada halaman Hasil Rekomendasi Paket Produk</p>
		Klik tombol "Clear"	Menghapus isian pada kolom teks tambah data
		Klik tombol "Back"	Menampilkan kolom teks isian nilai kriteria serta tombol "Submit" dan "Clear"

Tabel 4.15 menunjukkan bahwa pengujian *black box* pada *usecase* menentukan rekomendasi paket produk telah sesuai dengan harapan *developer* dan pengguna. Pengujian *black box* selengkapnya dapat dilihat pada Lampiran I (Pengujian *Black Box*).

BAB 5. HASIL DAN PEMBAHASAN

Bab ini menjelaskan tentang hasil pembangunan sistem dan juga pembahasannya. Penelitian ini menghasilkan sebuah sistem informasi penjualan pada Grosir Fashion Online dengan saran pembelian paket produk menggunakan algoritma apriori yang digunakan sebagai sistem informasi penjualan sekaligus menghasilkan dan menampilkan hasil rekomendasi paket produk. Pembahasan bertujuan untuk menjelaskan lebih detail mengenai jawaban dari perumusan masalah, serta tujuan dan manfaat dari penelitian.

5.1 Hasil Penerapan Algoritma Apriori Pada Proses Perhitungan Rekomendasi Paket Produk

Proses penentuan rekomendasi paket produk pada sistem informasi penjualan GFO dilakukan oleh *admin* sebagai pengelola sistem. Algoritma Apriori digunakan sebagai metode perhitungan untuk memberikan hasil rekomendasi paket produk yang didapat dari beberapa produk yang dibeli secara bersamaan dalam satu kali transaksi. Perhitungan ini berdasarkan kriteria algoritma Apriori berupa nilai minimum *support*, minimum *confidence*, dan masukan tanggal untuk menentukan tenggat waktu tertentu pada data yang akan digunakan. *Admin* bertugas memasukkan data kriteria ke dalam sistem.

Algoritma Apriori digunakan untuk menghitung nilai kriteria terhadap sejumlah transaksi yang telah ditentukan. Peneliti terlebih dahulu menganalisa data produk serta data transaksi pada sistem informasi penjualan yang telah ada untuk mengetahui kecocokan perhitungan algoritma Apriori terhadap proses penentuan rekomendasi paket produk pada sistem informasi penjualan GFO yang akan dibangun.

Sebagai contoh proses penentuan rekomendasi paket produk, peneliti menggunakan beberapa data produk dan data transaksi yang telah dimasukkan ke dalam sistem. Tabel 5.1 menunjukkan sebagian produk GFO yang akan dijadikan sampel proses penentuan rekomendasi paket produk menggunakan algoritma Apriori.

Nama produk diberi nama alias untuk memudahkan algoritma dalam melakukan perhitungan terhadap data.

Tabel 5.1 Nama Produk Sebagai Sampel

No.	ID Produk	Nama Produk	Nama Alias
1.	12	Cute Camera Ring	1
2.	14	Cincin Korea Creaved Shape	2
3.	15	Cincin Korea Rose Hollow Design	3
4.	17	Kalung Korea Diamond Decorated Leaf Shape Design	4
5.	20	Kalung Fashion Diamond Decorated Waterdrop Shape	5
6.	21	Gold Color Personality Bow Design Alloy Korean Ring	6
7.	23	Black Stones Engraved Vintage Personalized Ring	7

Berdasarkan beberapa produk tersebut, *member* kemudian dapat melakukan transaksi pembelian barang. Adapaun pola transaksi dari penjualan produk pada website GFO ditampilkan dalam Tabel 5.2 berikut ini:

Tabel 5.2 Pola Transaksi Penjualan Produk GFO

No.	Order_id	Itemset
1.	33	5
2.	35	1, 3
3.	36	4
4.	37	5
5.	40	2
6.	41	5
7.	42	6
8.	45	3, 4
9.	47	3, 4
10.	50	2, 6, 7
11.	52	6, 7

Berlanjut

Lanjutan		
12.	53	7
13.	54	7
14.	55	6
15.	56	6

a. Pembentukan *Itemset*

Tahap awal yaitu pembentukan C_1 atau disebut dengan 1 *itemset*. Sebagai contoh nilai minimum *support* yang dimasukkan adalah 10%, maka dapat dihitung dengan menggunakan rumus seperti di bawah ini:

$$Support(A) = \frac{\Sigma \text{Transaksi mengandung } A}{\Sigma \text{Transaksi}} \times 100\%$$

Maka didapat nilai *support* masing-masing produk seperti pada Tabel 5.3 berikut ini:

Tabel 5.3 Nilai *Support* dari Masing-masing *Item*

<i>Itemset</i>	<i>Support</i>
1	6,67%
2	13,3%
3	20%
4	20%
5	20%
6	33,3%
7	26,7%

Minimum *support* yang ditentukan yaitu 10%, sehingga *itemset* yang tidak memenuhi nilai minimum *support* dihilangkan, sehingga beberapa *itemset* yang diambil seperti pada Tabel 5.4.

Tabel 5.4 Bentuk 1-Itemset

<i>Itemset</i>	<i>Support</i>
2	13,3%
3	20%
4	20%
5	20%
6	33,3%
7	26,7%

b. Kombinasi 2-Itemset

Proses pembentukan C_2 atau disebut dengan 2 *itemset* dengan jumlah minimum *support* 10% dapat diselesaikan dengan rumus $Support(A,B) = P(A \cap B)$, sehingga didapatkan bentuk calon 2 *itemset*.

$$Support(A, B, C) = \frac{\Sigma \text{Transaksi mengandung } A, B, C}{\Sigma \text{Transaksi}} \times 100\%$$

Tabel 5.5 Calon 2-Itemset

<i>Itemset</i>	<i>Jumlah</i>	<i>Support</i>
2, 3	0	0
2, 4	0	0
2, 5	0	0
2, 6	0	0
2, 7	1	6,67%
3, 4	2	13,3%
3, 5	0	0
3, 6	0	0
3, 7	0	0
4, 5	0	0
4, 6	0	0
4, 7	0	0
5, 6	0	0
5, 7	0	0
6, 7	2	13,3%

Berdasarkan calon *2-itemset* pada Tabel 5.5, maka didapat *2-itemset* yang memenuhi nilai minimum *support* seperti pada Tabel 5.6 berikut ini:

Tabel 5.6 Bentuk *2-itemset*

<i>Itemset</i>	Jumlah	<i>Support</i>
3, 4	2	13,3%
6, 7	2	13,3%

c. Kombinasi *3-Itemset*

Berdasarkan bentuk *2-itemset*, maka dapat terbentuk calon *3-itemset* dengan bentuk seperti pada Tabel 5.8 berikut ini:

Tabel 5.7 Calon *3-Itemset*

<i>Itemset</i>	Jumlah
3, 4, 6	0
3, 4, 7	0
3, 6, 7	0
3, 4, 7	0
4, 6, 7	0

Tabel 5.7 menunjukkan bahwa tidak ada calon *3-itemset* yang dapat memenuhi nilai minimum *support* sehingga tidak dapat dijadikan *3-itemset*. Maka pada tahap selanjutnya *itemset* yang digunakan adalah *2-itemset*.

d. Pembentukan Aturan Asosiasi

Setelah semua pola frekuensi tinggi ditemukan, barulah dicari aturan asosiasi yang memenuhi syarat minimum untuk nilai *confidence* dengan menghitung *confidence* aturan asosiatif $A \Rightarrow B$.

Semisal nilai minimum *confidence* yang ditentukan sebesar 50%. Maka nilai *confidence* dari aturan $A \Rightarrow B$ diperoleh dengan rumus berikut:

$$Confidence = P(B|A) = \frac{\Sigma \text{Transaksi mengandung } A \text{ dan } B}{\Sigma \text{Transaksi mengandung } A} \times 100\%$$

Sehingga dari kombinasi *2-itemset* yang telah diketahui, dapat dilihat besarnya nilai *support* dan *confidence* dari calon aturan asosiasi seperti pada Tabel 5.8 berikut ini:

Tabel 5.8 Nilai *Confidence* dari Calon Aturan Asosiasi

Aturan		<i>Confidence</i>
Jika membeli produk 3, maka akan membeli produk 4	2/3	66,67%
Jika membeli produk 4, maka akan membeli produk 3	2/3	66,67%
Jika membeli produk 6, maka akan membeli produk 7	2/5	40%
Jika membeli produk 7, maka akan membeli produk 6	2/4	50%

e. Aturan Asosiasi Final

Aturan asosiasi final terurut yang dihasilkan berdasarkan nilai minimum *support* dan minimum *confidence* yang telah ditentukan dapat dilihat pada Tabel 5.9 di bawah ini.

Tabel 5.9 Hasil Aturan Asosiasi Final

Aturan	Support	Confidence
Jika membeli produk 2, maka akan membeli produk 4	13,3%	66,67%
Jika membeli produk 4, maka akan membeli produk 2	13,3%	66,67%
Jika membeli produk 7, maka akan membeli produk 6	13,3%	50%

Maka jika nama alias dikembalikan ke nama produk, hasil aturan asosiasi final yang didapat yaitu seperti pada Tabel 5.10.

Tabel 5.10 Hasil Aturan Asosiasi Final Berdasarkan Nama Produk

Aturan	Support	Confidence
Jika membeli Cincin Korea Creaved Shape, maka akan membeli Kalung Korea Diamond Decorated Leaf Shape Design	13,3%	66,67%
Jika membeli Kalung Korea Diamond Decorated Leaf Shape Design, maka akan membeli Cincin Korea Creaved Shape	13,3%	66,67%
Jika membeli Black Stones Engraved Vintage Personalized Ring, maka akan membeli Gold Color Personality Bow Design Alloy Korean Ring	13,3%	50%

Berdasarkan hasil penerapan algoritma Apriori pada sejumlah transaksi yang ada, didapatkan hasil aturan asosiasi final seperti pada Tabel 5.10 yang menunjukkan bahwa terdapat tiga kombinasi produk yang dapat dijadikan rekomendasi untuk menentukan paket produk.

Kesimpulan yang dapat diperoleh dari penerapan algoritma Apriori pada proses penentuan rekomendasi paket produk pada sejumlah transaksi tersebut antara lain:

1. Algoritma apriori dapat menghitung nilai *support* dan *confidence* dari masing-masing kombinasi *itemset* yang telah didapat.
2. Batasan tanggal transaksi yang akan dihitung dapat ditentukan oleh *admin*. Begitu pula dengan nilai minimum *support* dan *confidence*. Sehingga admin dapat menentukan pola transaksi konsumen selama waktu tertentu.
3. Satu produk yang sering dibeli dalam satu kali transaksi belum tentu bisa direkomendasikan untuk menjadi paket produk. Perhitungan algoritma apriori berdasar pada kombinasi dua atau lebih produk dalam satu kali transaksi.
4. Hasil aturan asosiasi final dapat dijadikan acuan bagi perusahaan untuk menentukan paket produk yang ingin ditawarkan.

Berdasarkan kesimpulan tersebut, maka algoritma Apriori terbukti dapat diterapkan pada proses penentuan rekomendasi paket produk pada sistem informasi penjualan GFO.

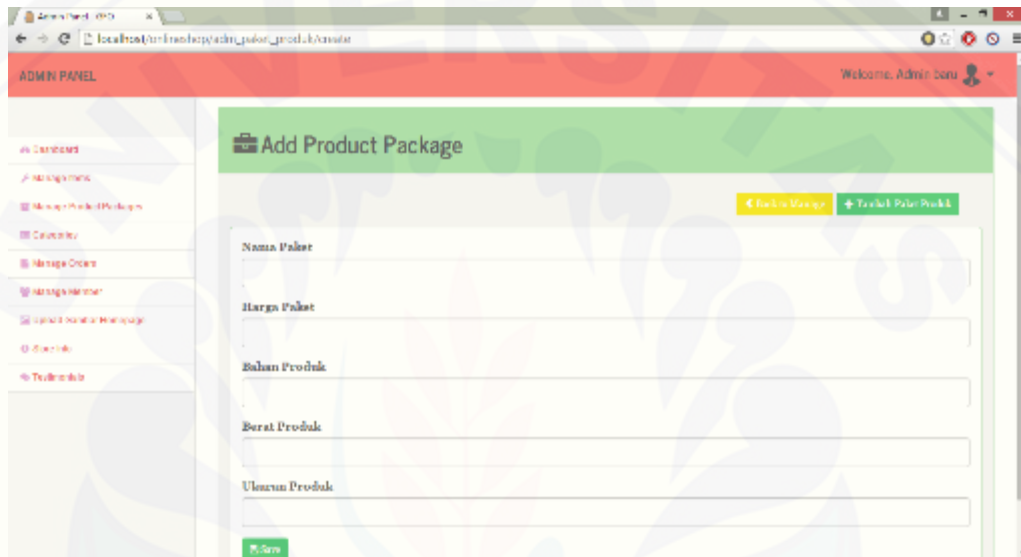
5.2 Hasil Pembuatan Sistem Informasi Penjualan GFO

Sistem informasi penjualan GFO memiliki tiga hak akses, yaitu CEO, *admin*, dan *member*. CEO sebagai pemilik perusahaan dapat mengelola data *admin*, baik untuk menambah, mengubah, serta menghapus data. *Admin* sebagai pengelola sistem dapat mengubah profil *admin*, mengubah level *member*, mengelola data produk, mengelola kategori, mengubah status *order*, mengelola gambar *landing page*, mengubah informasi toko, serta mengubah dan menghapus testimonial. *Member* sebagai pelaku transaksi dapat mengubah data *member*, mengelola data *order*, melihat status *order*, menambah data konfirmasi bayar, melihat informasi toko, menambah testimonial,

melihat testimonial, dan melihat halaman cara *order*. Fitur-fitur tersebut pada sistem informasi penjualan GFO dijelaskan pada sub-bab berikut:

5.2.1. Menu *Manage Product Packages*

Saat admin memilih menu *Manage Product Package*, sistem menampilkan halaman *input* data paket produk seperti pada Gambar 5.1.

The image shows a web browser window displaying an admin panel. The page title is "ADMIN PANEL" and the user is logged in as "Admin Baru". The main content area is titled "Add Product Package" and contains a form with the following fields: "Nama Paket", "Harga Paket", "Bahan Produk", "Berat Produk", and "Volume Produk". There is a "Save" button at the bottom of the form. Navigation buttons "Back to Menu" and "Tambah Paket Produk" are also visible.

Gambar 5.1 Halaman Input Kriteria

Kemudian setelah tombol “Save” ditekan, muncul tombol “Tentukan Isi Paket” untuk menampilkan halaman input kriteria perhitungan algoritma Apriori seperti pada Gambar 5.2.



ADMIN PANEL

Welcome, Admin baru

Input Kriteria

Nilai minimum persentase frekuensi kemunculan produk sesuai bersamaan

dalam %

Nilai persentase bobot/sumbu produk

dalam %

Batas yang Digunakan

Start Tanggal

mm/dd/yyyy

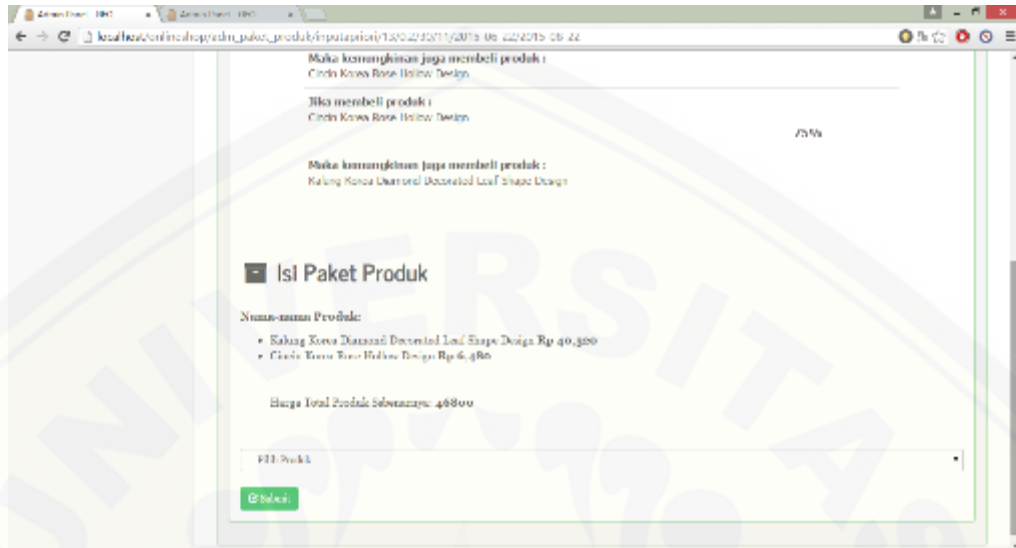
Batas Tanggal

mm/dd/yyyy

Hitung Apriori

Gambar 5.2 Halaman Input Kriteria

Ketika *admin* sudah memasukkan ketiga kriteria tersebut dan menekan tombol “Hitung Apriori”, maka sistem akan langsung menghitung dan menampilkan hasil aturan asosiasi final dari perhitungan tersebut sekaligus dapat menambah isi paket produk sesuai hasil rekomendasi pada halaman Rekomendasi Paket Produk seperti yang terlihat pada Gambar 5.3 berikut ini:



Gambar 5.3 Halaman Rekomendasi Paket Produk

5.3 Hasil Perbandingan Proses Penentuan Rekomendasi Paket Produk Secara Konvensional dan Menggunakan Sistem

Sistem informasi penjualan GFO diharapkan dapat memudahkan perusahaan untuk menentukan paket produk yang akan ditawarkan sesuai dengan kriteria yang ditentukan. Harapan tersebut akan terpenuhi apabila sistem dapat memberikan rekomendasi paket produk sesuai kriteria yang ditentukan. Hal ini sangat membantu karena dengan menggunakan sistem akan lebih mudah dibandingkan tanpa menggunakan sistem.

Jika penentuan rekomendasi paket produk dilakukan tanpa menggunakan sistem, dinilai memberikan kesulitan bagi perusahaan. Kesulitan yang didapat dari proses penentuan rekomendasi paket produk secara konvensional diharapkan dapat diatasi dengan penggunaan sistem informasi penjualan GFO.

Beberapa hal yang dapat dijadikan perbandingan antara proses penentuan rekomendasi paket produk secara konvensional dengan menggunakan sistem dapat dilihat pada Tabel 5.11.

Tabel 5.11 Perbandingan Penentuan Paket Secara Konvensional dengan Menggunakan Sistem

Penentuan Paket Produk Secara Konvensional	Penentuan Paket Produk dengan Menggunakan Sistem
<ol style="list-style-type: none"> 1. Pengambilan paket produk secara acak tanpa memperhatikan pola transaksi <i>member</i> kurang efektif karena tidak dapat memahami keinginan calon <i>customer</i>. 2. Analisa data yang banyak akan mempersulit <i>admin</i> karena harus memilah-milah data transaksi yang semakin hari semakin bertambah. 3. Waktu yang dibutuhkan lebih lama mengingat pada poin 1 dikatakan bahwa data akan bertambah tiap harinya. Proses akan semakin lambat jika ditengah-tengah proses analisa terjadi kesalahan perhitungan, maka harus kembali lagi menghitung dari awal. 	<ol style="list-style-type: none"> 1. Rekomendasi yang dihasilkan sesuai dengan pola transaksi <i>member</i> sehingga kombinasi paket produk yang ditawarkan akan banyak memikat calon customer. 2. Waktu yang dibutuhkan untuk menentukan rekomendasi paket produk tidak banyak walau data yang digunakan semakin banyak pula. 3. Data transaksi akan lebih bermanfaat, tidak hanya tersimpan di dalam database.
Berlanjut	

 Lanjutan

- | | |
|--|--|
| <p>4. Kesulitan mendapatkan data sesuai waktu yang diinginkan. Jika data terlalu banyak dan membutuhkan data menurut batasan waktu tertentu, maka akan ada tambahan waktu untuk memilah-milah data.</p> | <p>4. Penentuan rekomendasi produk yang berulang-ulang tidak akan merugikan perusahaan karena tidak membuang banyak waktu.</p> |
| <p>5. Proses ini akan terus berulang karena paket produk akan terus berganti mengingat pola transaksi <i>member</i> juga terus berubah mengikuti jaman. Waktu yang terbuang juga akan semakin bertambah.</p> | <p>5. Hasil yang didapat akan sangat berguna bagi perusahaan karena data yang diambil berdasarkan batasan tanggal yang ditentukan. Sehingga dapat digunakan untuk menentukan paket produk pada saat <i>event</i> tertentu.</p> |
-

Berdasarkan beberapa kemudahan yang dihasilkan oleh sistem informasi penjualan GFO sebagai penentu rekomendasi paket produk tersebut, maka dapat disimpulkan bahwa sistem informasi penjualan GFO dapat memberikan kemudahan dalam proses penentuan rekomendasi paket produk yang sesuai dengan kriteria yang ditentukan oleh *admin*. Namun aturan asosisasi final yang dihasilkan oleh fitur penentuan rekomendasi paket produk ini belum tentu digunakan oleh perusahaan sebagai paket produk yang ditawarkan, mengingat fungsi dari fitur ini hanya memberikan rekomendasi berdasarkan kombinasi produk dari pola transaksi *member*. Paket produk yang ditawarkan akan tetap diputuskan oleh perusahaan melalui proses berikutnya.

5.4 Perbandingan Penerapan Algoritma Apriori Pada *Market Basket Analysis* dengan Algoritma Apriori Sederhana

Algoritma Apriori merupakan salah satu algoritma asosiasi yang digunakan pada metode *market basket analysis* untuk dapat melihat pola transaksi konsumen terhadap keranjang belanja dalam satu kali transaksi yang dilakukan. Namun dalam beberapa kasus, algoritma Apriori tidak membandingkan *frequent itemset* yang didapat terhadap sejumlah transaksi yang ditentukan, sehingga nilai *support* hanya berdasarkan hasil dari pencarian *frequent itemset*.

Perbandingan perhitungan menggunakan algoritma Apriori pada metode *market basket analysis* dengan algoritma Apriori sederhana dapat dilihat pada Tabel 5.12.

Tabel 5.12 Perbandingan Penerapan Algoritma Apriori *Market Basket Analysis* dengan Algoritma Apriori Sederhana

Algoritma Apriori Pada <i>Market Basket Analysis</i>	Algoritma Apriori Sederhana
1. Pencarian k-itemset berdasarkan perbandingan kombinasi beberapa produk yang ditemui dalam satu kali transaksi terhadap seluruh transaksi.	1. Pencarian k-itemset hanya berdasarkan banyaknya kombinasi beberapa produk dalam satu kali transaksi tanpa membandingkan dengan keseluruhan transaksi.
2. Semakin banyak data transaksi yang digunakan akan semakin sempit pula kemungkinan hasil aturan asosiasi dijadikan rekomendasi paket produk.	2. Banyaknya data transaksi yang digunakan tidak berpengaruh besar terhadap aturan asosiasi <i>final</i> yang dihasilkan.
3. Hasil aturan asosiasi final lebih akurat.	3. Hasil aturan asosiasi final kurang akurat.

Sesuai dengan perbedaan perhitungan algoritma Apriori pada metode *market basket analysis* dan algoritma Apriori sederhana seperti yang telah disebutkn pada Tabel, maka hasil aturan asosiasi dari perhitungan algoritma Apriori pada metode

market basket analysis lebih efektif dibandingkan dengan yang menggunakan perhitungan sederhana.



BAB 6. PENUTUP

Bab ini berisi kesimpulan dan saran dari peneliti tentang penelitian yang telah dilakukan. Kesimpulan dan saran tersebut diharapkan dapat digunakan sebagai acuan pada penelitian selanjutnya.

6.1 Kesimpulan

Kesimpulan dari penelitian yang telah dilakukan adalah:

1. Sistem informasi penjualan GFO dengan saran pembelian paket produk dibuat dengan tiga hak akses, yaitu CEO, *admin*, dan *member* dengan fitur utama menentukan rekomendasi paket produk yang dilakukan oleh *admin*.
Perancangan sistem informasi penjualan GFO menggunakan model *prototype* sehingga pembangunan sistem dapat disesuaikan dengan keinginan user dan dibuat menggunakan bahasa pemrograman *Hypertext Preprocessor (PHP)* dan berbasis *web* sehingga dapat diakses dimanapun dan kapanpun.
2. Algoritma Apriori digunakan sebagai perhitungan pola transaksi berdasarkan kriteria yang ditentukan *admin*. Hasil aturan asosiasi final dapat dijadikan sebagai rekomendasi paket produk oleh GFO. Pada proses perhitungan, *admin* dapat menentukan nilai minimum *support* dan *confidence*, serta batasan tanggal transaksi yang akan dihitung.
3. Proses penentuan rekomendasi paket produk pada sistem informasi penjualan GFO dapat mengatasi kesulitan-kesulitan pada proses penentuan rekomendasi paket produk secara konvensional pada perusahaan GFO. Berdasarkan perhitungan penentuan paket produk menggunakan algoritma Apriori ini, rekomendasi paket produk yang dihasilkan sudah akurat karena perhitungan berdasarkan beberapa produk yang ditemui secara bersamaan dalam satu kali transaksi terhadap keseluruhan atau sejumlah transaksi sesuai batasan tanggal yang telah ditentukan, sehingga membuat rekomendasi paket produk yang

dihasilkan dapat berguna bagi perusahaan untuk menentukan paket produk sesuai dengan waktu atau *event* tertentu.

6.2 Saran

Beberapa saran dan masukan berikut yang diharapkan dapat memberikan perbaikan untuk penelitian selanjutnya, yaitu:

1. Pemanfaatan perhitungan algoritma Apriori pada sistem informasi penjualan GFO hanya sampai pada proses penentuan rekomendasi paket produk, namun akan lebih baik lagi jika terdapat fitur yang dapat membuktikan bahwa penggabungan produk berdasarkan rekomendasi paket produk dapat meningkatkan daya beli calon konsumen sehingga perusahaan juga dapat memantau hasil paket produk yang ditawarkan.
2. Perbandingan algoritma Apriori yang dibahas hanya terhadap perhitungan konvensional dengan perhitungan menggunakan sistem, dan juga perbandingan hasil dari algoritma Apriori pada *market basket analysis* dengan algoritma Apriori sederhana. Ada baiknya jika membandingkan metode *market basket analysis* dengan menggunakan algoritma Apriori ini dengan metode atau teknik lainnya.

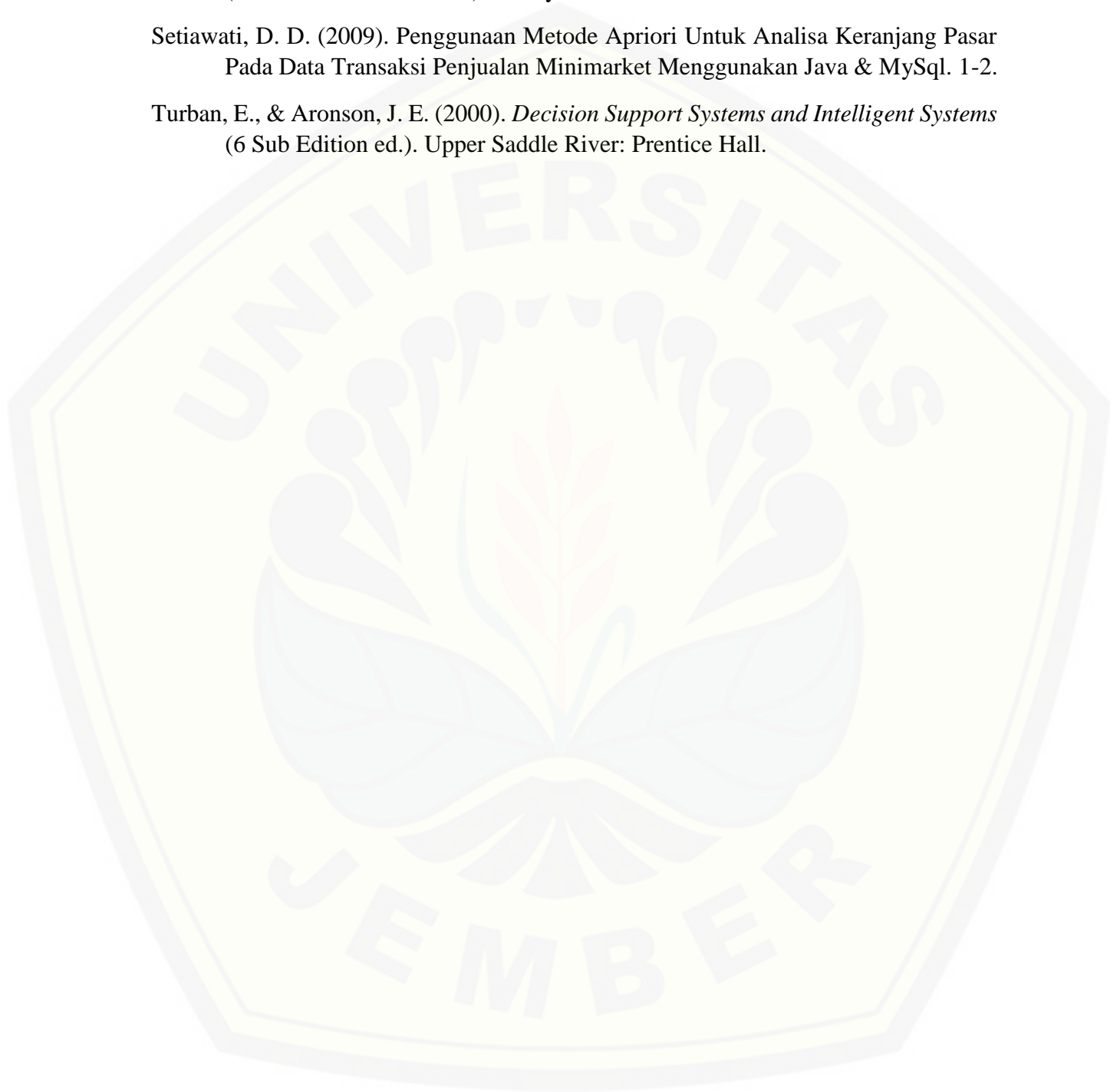
DAFTAR PUSTAKA

- Feriyanto, N. (2009, Mei 20). *Penjualan Bundling*. Dipetik Maret 9, 2015, dari Bisnis dan Ekonomi: <http://nur-feriyanto.blogspot.com/2009/05/penjualan-bundling.html>
- Gemala, M. (2013). Penerapan Metode Market Basket Analysis Pada Situs Web E-Commerce (Studi Kasus: Dhian Handicraft Indonesia). 2.
- Global, P. K. (2012, Agustus 23). Dipetik Maret 3, 2015, dari Facebook: <https://www.facebook.com/KurniaPrimaGlobal/posts/276116802493610>
- Grosir Fashion Online. (2013). *Tentang Kami: Grosir Fashion Online*. Dipetik Februari 1, 2015, dari Grosir Fashion Online: <http://www.grosirfashiononline.com/about.html>
- Gunadi, G., & Sensuse, D. I. (2012). Penerapan Metode Datamining Market Basket Analysis Terhadap Data Penjualan Produk Buku dengan Menggunakan Algoritma Apriori dan Frequent Pattern Growth (FP-Growth) : Studi Kasus Percetakan PT. Gramedia. *Jurnal TELEMATIKA MKOM, Vol. 4 No. 1*.
- Kotler, P. (2002). *Marketing Management* (11th Edition ed.). New Jersey: Prentice Hall International Inc.
- Levin, A., Levin, I. P., & Weller, J. (2005). A multi-attribute analysis of preferences for online and offline shopping: Differences between products, consumers and shopping stages. *Journal of Electronic Commerce Research, vol 6, 281-290*.
- Nidhra, S., & Dondeti, J. (2012). Black Box and White Box Testing Techniques - A Literature Review. *International Journal of Embedded Systems and Application (IJESA), Vol. 2, No. 2*.
- Nurchahyo, G. W. (2013). Penerapan Data Mining dengan Algoritma Apriori untuk Mendukung Strategi Promosi Pendidikan. *COREIT, Vol. 1 No. 1, 1*.
- Pane, D. K. (2013). Implementasi Data Mining Pada Penjualan produk Elektronik dengan Algoritma Apriori (Studi Kasus: Kreditplus). *Pelita Informatika Budi Darma, Vol: IV, Nomor: 3, 1*.
- Pressman, R. S. (2002). *Rekayasa Perangkat Lunak Pendekatan Praktisi*. Yogyakarta: Andi.
- Rangkuti, F. (2009). *Strategi Promosi yang Kreatif dan Analisis Kasus – Integrated Marketing Communications*. Jakarta: Gramedia.

Roberts, M. L. (2003). *Internet Marketing: Integrating Online and Offline Strategies* (International Edition ed.). New York: McGraw-Hill/Irwin.

Setiawati, D. D. (2009). Penggunaan Metode Apriori Untuk Analisa Keranjang Pasar Pada Data Transaksi Penjualan Minimarket Menggunakan Java & MySQL. 1-2.

Turban, E., & Aronson, J. E. (2000). *Decision Support Systems and Intelligent Systems* (6 Sub Edition ed.). Upper Saddle River: Prentice Hall.



LAMPIRAN

A. *Usecase* SkenarioA.1 *Usecase* Skenario *Login Admin* dan CEO

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario *login admin* dan *CEO* dapat dilihat pada Tabel 1.

Tabel 1. *Usecase* Skenario Login Admin

Nama <i>Usecase</i>	: <i>Login Admin</i>
ID <i>Usecase</i>	: GFO100-1
Aktor	: <i>Admin</i>
Pre Kondisi	: Aktor mengakses halaman sistem
Post Kondisi	: Aktor berhasil masuk ke <i>admin</i> panel

NORMAL SKENARIO *LOGIN USER*

Aktor	Sistem
1. Mengakses halaman sistem	2. Menampilkan halaman <i>Login</i>
3. Mengisi <i>field username</i> dan <i>password</i>	5. Menerima dan memverifikasi <i>username</i> dan <i>password</i>
4. Klik tombol " <i>Login</i> "	6. Data yang telah terisi adalah benar (valid)
	7. Menampilkan halaman Dashboard <i>Admin Panel</i>

Alternatif Skenario *Login*: Username dan Password tidak sesuai

ALTERNATIF SKENARIO *LOGIN*

- | |
|--|
| 6. Username dan password tidak sesuai |
| 7. Menampilkan pesan "Invalid username or password. Please try again." |

A.2 *Usecase* Skenario Logout Admin dan CEO

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario *logout admin* dan *CEO* dapat dilihat pada Tabel 2.

Tabel 2. *Usecase* Skenario Logout Admin dan CEO

Nama <i>Usecase</i>	: Logout Admin
ID <i>Usecase</i>	: GFO100-2
Aktor	: Admin
Pre Kondisi	: Aktor memilih panah dropdown di pojok kanan atas admin panel
Post Kondisi	: Aktor kembali ke halaman login

NORMAL SKENARIO LOGOUT USER

1. Klik panah dropdown pada pojok kanan atas
2. Klik tombol “Logout”
3. *Destroy all session*
4. Menampilkan halaman Login

A.3 *Usecase* Skenario Mengelola Admin

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengelola data *admin* dapat dilihat pada Tabel 3.

Tabel 3. *Usecase* Skenario Mengelola data Admin

Nama <i>Usecase</i>	: Mengelola Data Admin
ID <i>Usecase</i>	: GFO301
Aktor	: Chief Executive Officer
Pre Kondisi	: Aktor memilih menu “Manajemen User”
Post Kondisi	: Aktor berhasil menambah dan menghapus data admin

NORMAL SKENARIO MENAMBAH DATA ADMIN

Aktor	Sistem
1. Klik menu “Manajemen User”	
2. Klik submenu “Add Admin”	

	3. Menampilkan form Tambah Data Admin
4. Mengisi form tambah data admin	
NORMAL SKENARIO TOMBOL “SUBMIT” TAMBAH DATA ADMIN	
5. Klik tombol “Submit”	
	6. Menyimpan data ke basis data
	7. Menampilkan form data admin yang sudah terisi data disertai pesan “Data telah berhasil diubah”
NORMAL SKENARIO TOMBOL “CLEAR” PADA TAMBAH DATA ADMIN	
8. Klik tombol “Clear”	
	9. Menghapus isian pada kolom teks tambah data
NORMAL SKENARIO MENGHAPUS DATA ADMIN	
10. Klik submenu “Update Admin”	
	11. Mengambil data admin dari basis data
	12. Menampilkan daftar admin
13. Klik tombol “Delete” pada salah satu data admin	
	14. Menampilkan halaman Delete Data
NORMAL SKENARIO TOMBOL “YES” PADA DELETE DATA ADMIN	
15a. Klik tombol “Yes”	
	15b. Menghapus data pada basis data
	15c. Menampilkan daftar admin
NORMAL SKENARIO TOMBOL “NO” PADA DELETE DATA ADMIN	
15a. Klik tombol “No”	
	15b. Menampilkan daftar admin
Alternatif Skenario Menambah dan Menghapus Data Admin: Format data tidak sesuai	
ALTERNATIF SKENARIO FORMAT DATA TIDAK SESUAI	
	6a. Format data yang dimasukkan tidak sesuai
	7a. Menampilkan pesan error

Alternatif Skenario Menambah dan Menghapus Data Admin: Kolom teks masih kosong atau tidak diisi

ALTERNATIF SKENARIO KOLOM TEKS MASIH KOSONG

- 6b. Kolom teks masih kosong atau tidak diisi
- 7b. Menampilkan pesan error

A.4 *Usecase* Skenario Mengubah Data Profil Admin

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengubah data profil admin dapat dilihat pada Tabel 4.

Tabel 4. *Usecase* Skenario Mengubah Data Profil Admin

Nama <i>Usecase</i>	: Mengubah Data Profil Admin
ID <i>Usecase</i>	: GFO102
Aktor	: Admin
Pre Kondisi	: Aktor memilih tombol panah dropdown pada pojok kanan atas admin panel
Post Kondisi	: Aktor berhasil mengubah profil admin

NORMAL SKENARIO MENGUBAH DATA PROFIL ADMIN

Aktor	Sistem
1. Memilih tombol panah dropdown pada pojok kanan atas admin panel	
2. Klik tombol "User Profile"	
	3. Mengambil data admin dari basis data
	4. Menampilkan form data admin yang sudah terisi data
5. Mengubah data profil admin	

NORMAL SKENARIO TOMBOL "SUBMIT" UBAH USER PROFILE

6. Klik tombol "Submit"	
	7. Menyimpan data ke basis data
	8. Menampilkan form data admin yang sudah terisi data disertai pesan "Data telah berhasil diubah"

NORMAL SKENARIO TOMBOL "CANCEL" MENGUBAH DATA PROFIL ADMIN

9. Klik tombol “Cancel”

10. Menampilkan halaman
Dahsboard Admin Panel

Alternatif Skenario Mengubah Data Profil Admin: Format data yang dimasukkan tidak sesuai

ALTERNATIF SKENARIO FORMAT DATA TIDAK SESUAI

7.a. Format data yang dimasukkan tidak sesuai

8.a. Menampilkan pesan error

Alternatif Skenario Mengubah Data Profil Admin: Kolom teks masih kosong atau tidak diisi

ALTERNATIF SKENARIO KOLOM TEKS MASIH KOSONG

7.b Kolom teks masih kosong atau tidak diisi

8.b Menampilkan pesan error

A.5 Usecase Skenario Register Member

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario *register member* dapat dilihat pada Tabel 5.

Tabel 5. *Usecase* Skenario Register Member

Nama *Usecase* : Register Member
 ID *Usecase* : GFO201
 Aktor : Member
 Pre Kondisi : Aktor mengakses sistem
 Post Kondisi : Aktor berhasil register

NORMAL SKENARIO REGISTER MEMBER

Aktor	Sistem
1. Mengakses sistem	
2. Klik tombol “Login”	3. Menampilkan halaman <i>Member Login</i>
4. Klik tombol collapse “Daftar Disini”	
	5. Menampilkan form register <i>member</i>

6. Mengisi data <i>member</i>	
NORMAL SKENARIO TOMBOL “SUBMIT” MEMBER REGISTER	
7. Klik tombol “Submit”	
	8. Menyimpan data ke basis data
	9. Menampilkan halaman Akun <i>Member</i>

Alternatif Skenario Register *Member*: Format data yang dimasukkan tidak sesuai

ALTERNATIF SKENARIO FORMAT DATA TIDAK SESUAI	
	8.a. Format data yang dimasukkan tidak sesuai
	9.a. Menampilkan pesan error

Alternatif Skenario Register *Member*: Kolom teks masih kosong atau tidak diisi

ALTERNATIF SKENARIO KOLOM TEKS MASIH KOSONG	
	8.b. Kolom teks masih kosong atau tidak diisi
	9.b. Menampilkan pesan error

A.6 *Usecase Skenario Login Member*

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase skenario login member* dapat dilihat pada Tabel 6.

Tabel 6. *Usecase Skenario Login Member*

Nama <i>Usecase</i>	: <i>Login Member</i>
ID <i>Usecase</i>	: GFO202-1
Aktor	: <i>Member</i>
Pre Kondisi	: Aktor mengakses halaman sistem
Post Kondisi	: Aktor masuk ke halaman Akun <i>Member</i>

NORMAL SKENARIO LOGIN USER	
Aktor	Sistem
1. Mengakses sistem	
2. Klik tombol “Login”	
	3. Menampilkan halaman Login
4. Mengisi kolom teks <i>username</i> dan <i>password</i>	

-
5. Menerima dan memverifikasi *username* dan *password*
 6. Data yang telah terisi adalah benar (valid)
 7. Menampilkan halaman Akun *Member*

NORMAL SKENARIO LOGOUT USER

8. Klik tombol “Logout”

9. Menampilkan halaman Login
-

Alternatif Skenario Login: Username dan Password tidak sesuai

ALTERNATIF SKENARIO LOGIN

6. Username dan password tidak sesuai
 7. Menampilkan pesan “Invalid username or password. Please try again.”
-

A.7 *Usecase* Skenario Logout *Member*

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario *logout member* dapat dilihat pada Tabel 7.

Tabel 7. *Usecase* Skenario Logout *Member*

Nama <i>Usecase</i>	: Logout <i>Member</i>
ID <i>Usecase</i>	: GFO202-2
Aktor	: <i>Member</i>
Pre Kondisi	: Aktor memilih tombol “Logout”
Post Kondisi	: Aktor kembali ke halaman login

NORMAL SKENARIO LOGOUT MEMBER

1. Klik tombol “Logout”

2. *Destroy all session*
 3. Menampilkan halaman Login
-

A.8 *Usecase* Skenario Mengubah Data *Member*

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengubah data *member* dapat dilihat pada Tabel 8.

Tabel 8. *Usecase* Skenario Mengubah Data *Member*

Nama <i>Usecase</i>	: Mengubah Data <i>Member</i>
ID <i>Usecase</i>	: GFO203
Aktor	: <i>Member</i>
Pre Kondisi	: Aktor memilih tombol “Akun Saya”
Post Kondisi	: Aktor berhasil mengubah data <i>member</i>

NORMAL SKENARIO MENGUBAH DATA <i>MEMBER</i>	
Aktor	Sistem
1. Klik tombol “Akun Saya”	2. Mengambil data level dan <i>order member</i> dari basis data
	3. Menampilkan halaman Akun <i>Member</i>
4. Klik tombol “Ubah Profil”	5. Mengambil data <i>member</i> dari basis data
	6. Menampilkan halaman Ubah Profil yang berisi kolom teks yang telah berisi data <i>member</i>
7. Mengubah data profil <i>member</i>	
NORMAL SKENARIO TOMBOL “SUBMIT” UBAH DATA <i>MEMBER</i>	
8. Klik tombol “Submit”	9. Menyimpan data ke basis data
	10. Menampilkan halaman Ubah Profil
Alternatif Skenario Mengubah Data <i>Member</i> : Format data yang dimasukkan tidak sesuai	
ALTERNATIF SKENARIO FORMAT DATA TIDAK SESUAI	
	9.a. Format data yang dimasukkan tidak sesuai
	10.a. Menampilkan pesan error

Alternatif Skenario Mengubah Data *Member*: Kolom teks masih kosong atau tidak diisi

ALTERNATIF SKENARIO KOLOM TEKS MASIH KOSONG	
9.b	Kolom teks masih kosong atau tidak diisi
10.b	Menampilkan pesan error

A.9 *Usecase* Skenario Mengubah Level *Member*

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengubah level *member* dapat dilihat pada Tabel 9.

Tabel 9. *Usecase* Skenario Mengubah Level *Member*

Nama <i>Usecase</i>	: Mengubah Level <i>Member</i>
ID <i>Usecase</i>	: GFO103
Aktor	: Admin
Pre Kondisi	: Aktor memilih menu “Manajemen <i>Member</i> ”
Post Kondisi	: Aktor berhasil mengubah level <i>member</i>

NORMAL SKENARIO MENGUBAH LEVEL <i>MEMBER</i>	
Aktor	Sistem
1. Klik menu “Manajemen <i>Member</i> ”	2. Mengambil data <i>member</i> dari basis data
	3. Menampilkan daftar data <i>member</i>
4. Memilih salah satu pilihan level <i>member</i>	
NORMAL SKENARIO TOMBOL “SUBMIT” UBAH LEVEL <i>MEMBER</i>	
5. Klik tombol “Submit”	6. Menyimpan data ke basis data
	7. Menampilkan daftar data <i>member</i>

A.10 *Usecase* Skenario Mengelola Data Produk

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengelola data produk dapat dilihat pada Tabel 10.

Tabel 10. *Usecase* Skenario Mengelola Data Produk

Nama <i>Usecase</i>	: Mengelola Data Produk
ID <i>Usecase</i>	: GFO104
Aktor	: Admin
Pre Kondisi	: Aktor memilih menu “Manajemen Produk”
Post Kondisi	: Aktor berhasil mengelola Data Produk

NORMAL SKENARIO MENGELOLA PRODUK	
Aktor	Sistem
1. Klik menu “Manajemen Produk”	
NORMAL SKENARIO TAMBAH PRODUK	
2. Klik submenu “Add Product”	3. Menampilkan halaman form Tambah Produk disertai tombol “Submit”, “Clear”, “Update Color”, “Update Pic”, dan “Kategori Produk”
4. Mengisi data produk	
NORMAL SKENARIO TOMBOL “SUBMIT” TAMBAH DATA PRODUK	
5.a. Klik tombol “Submit”	5.a. Menyimpan ke basis data 6.a. Menampilkan halaman Update Product yang berisi daftar data produk
NORMAL SKENARIO TOMBOL “CLEAR” PADA TAMBAH DATA PRODUK	
5b. Klik tombol “Clear”	6b. Menghapus isian kolom teks Update Product
NORMAL SKENARIO MENGUBAH DATA PRODUK	
7. Klik submenu “Update Product”	8. Mengambil data produk dari basis data 9. Menampilkan halaman Update Product yang berisi daftar data produk, disertai tombol “Edit”, “Delete”, dan “Back to Manage”
10. Klik tombol “Edit” pada salah satu data produk	

	11. Mengambil data produk yang dipilih dari basis data
	12. Menampilkan halaman Update Product yang berisi kolom teks yang berisi data produk, disertai tombol "Submit" dan "Clear"
13. Mengubah data Product	
NORMAL SKENARIO TOMBOL "SUBMIT" PADA UBAH DATA PRODUK	
14.a. Klik tombol "Submit"	15.a. Menyimpan data ke dalam basis data
	16.a. Mengambil data produk dari basis data
	17.a. Menampilkan halaman Update Product
NORMAL SKENARIO TOMBOL "CLEAR" PADA UBAH DATA PRODUK	
14.a. Klik tombol "Clear"	15.a. Mengembalikan isian kolom teks seperti semula tanpa ada perubahan
NORMAL SKENARIO TOMBOL "UPDATE COLORS" PADA UPDATE PRODUCT	
16. Klik tombol "Update Colors"	17. Mengambil data warna produk dari basis data
	18. Menampilkan halaman Update Colors yang berisi kolom teks warna produk, pilihan warna yang ada, disertai tombol "Submit", "Delete", dan "Back to Manage"
19. Mengisi kolom teks warna	
NORMAL SKENARIO TOMBOL "SUBMIT" PADA UPDATE COLORS	
20.a. Klik tombol "Submit"	21.a. Menyimpan dalam basis data
	22.a. Menambahkan daftar warna produk
NORMAL SKENARIO TOMBOL "DELETE" PADA UPDATE COLOR	

20.a. Klik tombol “Delete” pada salah satu pilihan warna

21.a. Mengapus pilihan warna pada basis data

22.a. Menampilkan daftar pilihan warna produk

**NORMAL SKENARIO TOMBOL “UPLOAD PIC”
PADA UPDATE PRODUCT**

25. Klik tombol “Upload Pic”

26. Menampilkan halaman Upload Product Picture

27. Klik tombol “Browse”

28. Menampilkan window File Upload

29. Memilih salah satu foto produk

30. Klik tombol “Open”

31. Klik tombol “Upload”

32. Menyimpan nama *file* gambar ke dalam basis data

33. Menampilkan halaman Upload Product Picture disertai dengan gambar produk yang baru

**NORMAL SKENARIO TOMBOL “KATEGORI PRODUK”
PADA UPDATE PRODUCT**

34. Klik tombol “Kategori Produk”

35. Menampilkan halaman Manage Kategori Produk

36. Memilih salah satu kategori produk

**NORMAL SKENARIO TOMBOL “SUBMIT”
PADA KATEGORI PRODUCT**

37. Klik tombol “Submit”

38. Menampilkan daftar Kategori yang Dipilih

**NORMAL SKENARIO TOMBOL “SELESAI”
PADA KATEGORI PRODUCT**

37a. Klik tombol “Selesai”

38a. Kembali ke halaman Update Product

NORMAL SKENARIO MENGHAPUS DATA PRODUK

37b. Klik tombol “Delete” pada salah satu produk

-
- 38b. Menampilkan halaman Delete Produk disertai tombol “Yes” dan “No”

**NORMAL SKENARIO TOMBOL “YES”
PADA DELETE DATA PRODUK**

- 39a. Klik tombol “Yes”

- 40a. Menghapus data dari basis data
41a. Mengambil data produk dari basis data
42a. Menampilkan halaman Update Product yang berisi daftar data produk disertai tombol “Edit”, “Delete”, dan “Back to Manage”

**NORMAL SKENARIO TOMBOL “NO”
PADA DELETE PRODUK**

- 39b. Klik tombol “No”

- 40b. Kembali menampilkan halaman Update Product yang berisi daftar data produk disertai tombol “Edit”, “Delete”, dan “Back to Manage”
-

Alternatif Skenario Menambah Data Produk: Format data yang dimasukkan tidak sesuai

ALTERNATIF SKENARIO FORMAT DATA TIDAK SESUAI

- 5b. Format data yang dimasukkan tidak sesuai
6b. Menampilkan pesan error
-

Alternatif Skenario Menambah Data Produk: Kolom teks masih kosong atau tidak diisi

ALTERNATIF SKENARIO KOLOM TEKS MASIH KOSONG

- 5c. Kolom teks masih kosong atau tidak diisi
6c. Menampilkan pesan error
-

Alternatif Skenario Mengubah Data Produk: Format data yang dimasukkan tidak sesuai

ALTERNATIF SKENARIO FORMAT DATA TIDAK SESUAI

- 15b. Format data yang dimasukkan tidak sesuai
-

16b. Menampilkan pesan error

Alternatif Skenario Mengubah Data Produk: Kolom teks masih kosong atau tidak diisi

ALTERNATIF SKENARIO KOLOM TEKS MASIH KOSONG

- 15c. Kolom teks masih kosong atau tidak diisi
 - 16c. Menampilkan pesan error
-

Alternatif Skenario Mengubah Data Produk Upload Pic: Ukuran file yang dipilih terlalu besar

**ALTERNATIF SKENARIO
UKURAN FILE YANG DIPILIH TERLALU BESAR**

- 24b. Ukuran file yang dipilih lebih besar dari yang diminta
 - 25b. Menampilkan pesan error
-

A.11 *Usecase* Skenario Mengelola Kategori Produk

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengelola kategori produk dapat dilihat pada Tabel 11.

Tabel 11. *Usecase* Skenario Mengelola Kategori

Nama *Usecase* : Mengelola Kategori
 ID *Usecase* : GFO106
 Aktor : Admin
 Pre Kondisi : Aktor memilih menu “Manajemen Produk”
 Post Kondisi : Aktor berhasil mengelola kategori

NORMAL SKENARIO MENGELOLA KATEGORI PRODUK

Aktor	Sistem
1. Memilih menu Manajemen Produk	
2. Memilih submenu Categories	

-
3. Mengambil data kategori dari basis data
 4. Menampilkan daftar kategori produk

**NORMAL SKENARIO TOMBOL “TAMBAH KATEGORI”
PADA MENGELOLA KATEGORI PRODUK**

5. Klik tombol “Tambah Kategori”
6. Menampilkan halaman Tambah Kategori yang berisi kolom teks nama kategori dan tombol “Submit”
7. Mengisi kolom teks Nama Kategori
8. Klik tombol “Submit”
9. Menyimpan kategori ke basis data
10. Mengambil data kategori dari basis data
11. Menampilkan daftar kategori produk disertai tombol "Tambah Kategori", "Edit", dan "Delete"

**NORMAL SKENARIO TOMBOL “EDIT”
PADA MENGELOLA KATEGORI PRODUK**

12. Klik tombol “Edit” pada salah satu kategori
13. Mengambil data sub kategori dari database dengan parent kategori sesuai dengan yang telah dipilih
14. Menampilkan halaman manajemen kategori yang berisi tombol “Tambah Kategori”, “Update Parent Kategori”, tombol “Edit”, dan “Delete”

NORMAL SKENARIO TOMBOL “TAMBAH SUB KATEGORI”

15. Klik tombol “Tambah Kategori”
16. Menampilkan halaman Tambah Kategori
17. Mengisi kolom teks Nama Kategori

**NORMAL SKENARIO TOMBOL “SUBMIT”
TAMBAH SUB KATEGORI**

18. Klik tombol “Submit”
-

-
19. Menyimpan sub kategori ke basis data
 20. Mengambil data sub kategori dari basis data
 21. Menampilkan halaman manajemen kategori disertai pesan bahwa sub kategori produk telah ditambahkan

**NORMAL SKENARIO TOMBOL “BACK TO MANAGE”
TAMBAH SUB KATEGORI**

22. Klik tombol “Back to Manage”
23. Kembali ke halaman Manajemen Kategori

NORMAL SKENARIO TOMBOL “UPDATE PARENT KATEGORI”

24. Klik tombol “Update Parent Kategori”
25. Mengambil data parent kategori dari database
26. Menampilkan halaman Update Parent Kategori
27. Mengisi kolom teks Nama Kategori

**NORMAL SKENARIO TOMBOL “SUBMIT”
TAMBAH UPDATE PARENT KATEGORI**

28. Klik tombol “Submit”
29. Menyimpan kategori ke basis data
30. Mengambil data kategori dari basis data
31. Menampilkan daftar kategori

**NORMAL SKENARIO TOMBOL “BACK TO MANAGE”
TAMBAH UPDATE PARENT KATEGORI**

32. Klik tombol “Back to Manage”
33. Kembali ke halaman Manajemen Kategori

NORMAL SKENARIO TOMBOL “DELETE” KATEGORI

34. Klik tombol “Delete” pada salah satu kategori produk
35. Menampilkan halaman Hapus Kategori

**NORMAL SKENARIO TOMBOL “YES”
PADA DELETE KATEGORI**

- 36a. Klik tombol “Yes”
-

	37a. Menghapus data kategori dari basis data
	38a. Menampilkan daftar kategori
NORMAL SKENARIO TOMBOL “YES” PADA DELETE KATEGORI	
36b. Klik tombol “No”	37b. Kembali ke daftar kategori

12. *Usecase* Skenario Mengelola Data *Order*

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengelola data *order* dapat dilihat pada Tabel 12.

Tabel 12. *Usecase* Skenario Mengelola Data *Order*

Nama <i>Usecase</i>	: Mengelola Data <i>Order</i>
ID <i>Usecase</i>	: GFO204
Aktor	: <i>Member</i>
Pre Kondisi	: Aktor memilih salah satu produk
Post Kondisi	: Aktor berhasil melakukan transaksi pemesanan

NORMAL SKENARIO MENGELOLA DATA <i>ORDER</i>	
Aktor	Sistem
1. Mengakses sistem	2. Mengambil data produk dari basis data
	3. Menampilkan Landing Page berisi baris produk
4. Memilih salah satu produk pada halaman baris produk	5. Mengambil data produk dari database
	6. Menampilkan data produk yang dipilih
7. Memilih warna produk dari dropdown pilihan warna produk	
8. Memasukkan jumlah barang yang diinginkan	
9. Klik tombol “Beli Sekarang”	10. Memasukkan data ke dalam <i>cart session</i>
	11. Mengambil isi <i>cart session</i> dari <i>cart contents</i>

-
12. Menampilkan halaman Tas Belanja disertai tombol “Update cart”, “Delete”, “Trash”, “Continue Shopping”, dan “Lanjut ke langkah selanjutnya”

**NORMAL SKENARIO TOMBOL “UPDATE CART”
PADA HALAMAN TAS BELANJA**

13. Ubah jumlah barang yang dipesan pada kolom teks Qty
14. Klik tombol “Update Cart”
15. Mengubah jumlah barang yang dipesan pada *cart contents*
16. Mengambil isi *cart session* dari *cart contents*
17. Menampilkan halaman Cart Contents disertai tombol “Update cart”, “Delete”, “Trash”, “Continue Shopping”, dan “Lanjut ke langkah selanjutnya”

**NORMAL SKENARIO TOMBOL “DELETE”
PADA HALAMAN TAS BELANJA**

18. Klik tombol “Delete” pada salah satu barang di daftar Tas Belanja
19. Menghapus barang yang dipilih dari *cart contents*
20. Mengambil isi *cart session* dari *cart contents*
21. Menampilkan halaman Cart Contents disertai tombol “Update cart”, “Delete”, “Trash”, “Continue Shopping”, dan “Lanjut ke langkah selanjutnya”

**NORMAL SKENARIO TOMBOL “CONTINUE SHOPPING”
PADA HALAMAN TAS BELANJA**

22. Klik tombol “Continue Shopping”
23. Mengambil data produk dari basis data
24. Kembali ke halaman baris produk

**NORMAL SKENARIO TOMBOL “LANJUT KE LANGKAH BERIKUTNYA”
PADA HALAMAN TAS BELANJA**

25. Klik tombol “Lanjut ke Langkah Berikutnya”
-

26. Mengambil data *member* dari basis data

**NORMAL SKENARIO UBAH ALAMAT PENGIRIMAN
PADA MENGELOLA DATA *ORDER***

27. Menampilkan halaman Data Pembeli berupa kolom teks yang telah terisi data *member* disertai tombol “Simpan di Akun” dan “Lanjut ke Ongkir dan Pembayaran”

28. Mengubah data pembeli sesuai dengan alamat pengiriman

29. Klik tombol “Simpan di Akun”

30. Menyimpan data *member* ke basis data

31. Mengambil data *member* dari basis data

32. Menampilkan halaman Data Pembeli berupa kolom teks yang telah terisi data *member* disertai tombol “Simpan di Akun” dan “Lanjut ke Ongkir dan Pembayaran”

**NORMAL SKENARIO “LANJUT KE ONGKIR
DAN PEMBAYARAN”
PADA HALAMAN UBAH ALAMAT PENGIRIMAN**

33. Klik tombol “Lanjut ke Ongkir dan Pembayaran”

34. Mengambil data *member* dan *cart contents* pada *cart session*

NORMAL SKENARIO HALAMAN ONGKIR DAN PEMBAYARAN

35. Menampilkan halaman Ongkir dan Pembayaran yang berisi Rangkuman Data Pembeli dan Tas Belanja lengkap dengan jumlah ongkos kirim dan total yang harus dibayarkan, disertai tombol “Ubah Data Alamat” dan “Pemesanan Selesai”

**NORMAL SKENARIO TOMBOL “UBAH DATA ALAMAT”
PADA HALAMAN ONGKIR DAN PEMBAYARAN**

36. Klik tombol “Ubah Data Alamat”

-
- 37. Mengambil data dari basis data
 - 38. Menampilkan halaman Data Pembeli berupa kolom teks yang telah terisi data *member* disertai tombol “Simpan di Akun” dan “Lanjut ke Ongkir dan Pembayaran”

**NORMAL SKENARIO TOMBOL “PEMESANAN SELESAI”
PADA HALAMAN ONGKIR DAN PEMBAYARAN**

- 39. Klik tombol “Pemesanan Selesai”
 - 40. Menyimpan Rangkuman Data Pembeli dan Data *Order* ke basis data
 - 41. Menampilkan halaman Konfirmasi Pembayaran dengan menampilkan nomor *order*, total bayar, dan nomor rekening *online shop*
-

13. *Usecase* Skenario Melihat Daftar *Order*

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario melihat daftar *order* dapat dilihat pada Tabel 13.

Tabel 13. *Usecase* Skenario Melihat Daftar *Order*

Nama <i>Usecase</i>	: Melihat Daftar <i>Order</i>
ID <i>Usecase</i>	: GFO205
Aktor	: <i>Member</i>
Pre Kondisi	: Aktor memilih menu “ <i>Order</i> Saya”
Post Kondisi	: Aktor melihat Daftar <i>Order</i> dan Kelengkapan <i>Order</i>

NORMAL SKENARIO MELIHAT DAFTAR *ORDER*

Aktor	Sistem
1. Memilih menu <i>Order</i> Saya	2. Mengambil data <i>order</i> dari basis data
	3. Menampilkan halaman <i>Order</i> Saya yang berisi daftar <i>order</i> dan tombol “Kelengkapan <i>Order</i> ”

**NORMAL SKENARIO MELIHAT KELENGKAPAN *ORDER*
PADA HALAMAN *ORDER* SAYA**

-
4. Klik tombol “Kelengkapan *Order*” pada salah satu daftar *order*
 5. Mengambil data *order* detail dari basis data
 6. Menampilkan halaman Kelengkapan *Order* yang berisi data pembeli dan detail *order*
-

14. *Usecase* Skenario Mengubah Status *Order*

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengubah status *order* dapat dilihat pada Tabel

14.

Tabel 14. *Usecase* Skenario Mengubah Status *Order*

Nama <i>Usecase</i>	: Mengubah status <i>order</i>
ID <i>Usecase</i>	: GFO107
Aktor	: Admin
Pre Kondisi	: Aktor memilih menu “Manajemen <i>Order</i> ”
Post Kondisi	: Aktor berhasil mengubah status <i>order</i>

NORMAL SKENARIO MENGUBAH STATUS *ORDER*

Aktor	Sistem
1. Memilih menu Manajemen <i>Order</i>	2. Mengambil data <i>order</i> dari basis data
	3. Menampilkan halaman Manajemen <i>Order</i> yang berisi daftar <i>order</i>
4. Pilih salah satu status pada tombol dropdown Status <i>Order</i>	
5. Klik tombol “Submit”	6. Menyimpan status <i>order</i> pada basis data
	7. Menampilkan halaman Manajemen <i>Order</i> yang berisi daftar <i>order</i>

15. *Usecase* Skenario Menambah Data Konfirmasi Bayar

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario menambah data konfirmasi bayar dapat dilihat pada Tabel 15.

Tabel 15. *Usecase* Skenario Menambah Data Konfirmasi Bayar

Nama <i>Usecase</i>	: Menambah Data Konfirmasi Bayar
ID <i>Usecase</i>	: GFO206
Aktor	: <i>Member</i>
Pre Kondisi	: Aktor memilih menu “Konfirmasi Bayar”
Post Kondisi	: Aktor berhasil menambah data konfirmasi bayar

NORMAL SKENARIO MENAMBAH DATA KONFIRMASI BAYAR	
Aktor	Sistem
1. Memilih menu “Konfirmasi Bayar”	
	2. Menampilkan halaman Konfirmasi Bayar
3. Mengisi kolom teks konfirmasi bayar sesuai dengan data <i>order</i>	
4. Klik tombol Submit	
	5. Menyimpan data yang dimasukkan ke dalam basis data
	6. Menampilkan halaman Konfirmasi Bayar disertai pesan bahwa data telah disimpan

Alternatif Skenario Menambah Data Konfirmasi Bayar: Format data yang dimasukkan tidak sesuai

ALTERNATIF SKENARIO FORMAT DATA TIDAK SESUAI
5.a. Format data yang dimasukkan tidak sesuai
6.a. Menampilkan pesan error

Alternatif Skenario Menambah Data Konfirmasi Bayar: Kolom teks masih kosong atau tidak diisi

ALTERNATIF SKENARIO KOLOM TEKS MASIH KOSONG	
---	--

- | |
|--|
| 5.b Kolom teks masih kosong atau tidak diisi |
| 6.b Menampilkan pesan error |

16. *Usecase* Skenario Mengelola Gambar Landing Page *Member*

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengelola gambar *landing page member* dapat dilihat pada Tabel 16.

Tabel 16. *Usecase* Skenario Mengelola Gambar Landing Page *Member*

Nama <i>Usecase</i>	: Mengelola Gambar Landing Page <i>Member</i>
ID <i>Usecase</i>	: GFO108
Aktor	: Admin
Pre Kondisi	: Aktor memilih menu “Upload Gambar Homepage”
Post Kondisi	: Aktor berhasil mengelola gambar landing page <i>member</i>

NORMAL SKENARIO MENGELOLA GAMBAR LANDING PAGE <i>MEMBER</i>	
---	--

Aktor	Sistem
1. Memilih menu “Upload Gambar Homepage”	2. Mengambil data gambar <i>homepage</i> dari basis data
	3. Menampilkan halaman Upload Homepage Pictures disertai tombol “Browse”, kolom teks link gambar, dan tombol “Upload”, serta tombol “Delete” pada masing-masing gambar yang telah diupload

NORMAL SKENARIO MENAMBAH GAMBAR PADA MENGELOLA GAMBAR LANDING PAGE <i>MEMBER</i>	
--	--

4. Klik tombol “Browse”	5. Menampilkan <i>window</i> Open File
6. Memilih salah satu gambar	
7. Klik tombol “Open”	
8. Memasukkan <i>link</i> gambar	
9. Klik tombol “Upload”	

-
10. Menyimpan nama *file* gambar dan *link* gambar pada basis data
 11. Menampilkan halaman Upload Homepage Pictures disertai tombol “Browse”, kolom teks link gambar, dan tombol “Upload”, serta tombol “Delete” pada masing-masing gambar yang telah diupload

**NORMAL SKENARIO HAPUS GAMBAR
PADA MENGELOLA GAMBAR LANDING PAGE *MEMBER***

11. Klik tombol “Delete” pada salah satu gambar yang ingin dihapus
 12. Menghapus gambar dari basis data
 13. Menampilkan halaman Upload Homepage Pictures disertai tombol “Browse”, kolom teks link gambar, dan tombol “Upload”, serta tombol “Delete” pada masing-masing gambar yang telah diupload
-

Alternatif Skenario Menambah Gambar: Ukuran file yang dipilih terlalu besar

**ALTERNATIF SKENARIO
UKURAN FILE YANG DIPILIH TERLALU BESAR**

17. Ukuran file yang dipilih lebih besar dari yang diminta
 18. Menampilkan pesan error
-

17. *Usecase* Skenario Mengubah Informasi Toko

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengubah informasi toko dapat dilihat pada Tabel 17.

Tabel 17. *Usecase* Skenario Mengubah Informasi Toko

Nama <i>Usecase</i>	: Mengubah Informasi Toko
ID <i>Usecase</i>	: GFO109
Aktor	: Admin
Pre Kondisi	: Aktor memilih menu “Store Info”
Post Kondisi	: Aktor berhasil mengubah informasi toko

NORMAL SKENARIO MENGUBAH INFORMASI TOKO	
Aktor	Sistem
1. Memilih menu “Store Info”	2. Mengambil data informasi toko
	3. Menampilkan halaman Store Info disertai kolom teks yang berisi informasi toko baku disertai tombol “Submit” dan “Reset”
4. Mengubah informasi toko	
NORMAL SKENARIO TOMBOL “SUBMIT” PADA MENGUBAH INFORMASI TOKO	
5. Klik tombol “Submit”	6. Menyimpan masukan pada kolom teks ke dalam basis data
	7. Menampilkan halaman Store Info disertai kolom teks yang berisi informasi toko baku disertai tombol “Submit” dan “Reset”
NORMAL SKENARIO TOMBOL “RESET” PADA MENGUBAH INFORMASI TOKO	
12. Klik tombol “Reset”	13. Menghapus isian pada kolom teks informasi toko
Alternatif Skenario Mengubah Informasi Toko: Kolom teks masih kosong atau tidak diisi	
ALTERNATIF SKENARIO KOLOM TEKS MASIH KOSONG	
	13.a Kolom teks masih kosong atau tidak diisi
	14.a Menampilkan pesan error

18. *Usecase* Skenario Melihat Informasi Toko

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario melihat informasi toko dapat dilihat pada Tabel 18.

Tabel 18. *Usecase* Skenario Melihat Informasi Toko

Nama <i>Usecase</i>	: Melihat Informasi Toko
ID <i>Usecase</i>	: GFO207
Aktor	: <i>Member</i>
Pre Kondisi	: Aktor memilih menu “Store Info”
Post Kondisi	: Aktor melihat informasi toko

NORMAL SKENARIO MELIHAT INFORMASI TOKO	
Aktor	Sistem
1. Memilih menu “Store Info”	2. Mengambil data informasi toko 3. Menampilkan halaman Store Info

19. *Usecase* Skenario Mengelola Testimonial

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengelola testimonial dapat dilihat pada Tabel 19.

Tabel 19. *Usecase* Skenario Mengelola Testimonial

Nama <i>Usecase</i>	: Mengelola Testimonial
ID <i>Usecase</i>	: GFO110
Aktor	: Admin
Pre Kondisi	: Aktor memilih menu “Testimonials”
Post Kondisi	: Aktor berhasil mengelola testimonial

NORMAL SKENARIO MENGELOLA TESTIMONIAL	
Aktor	Sistem
1. Memilih menu “Testimonials”	2. Mengambil data testimonial 3. Menampilkan halaman Manage Testimonial yang berisi daftar testimonial disertai tombol “Edit”, “Delete”, dan “Back to Manage”

NORMAL SKENARIO MENGUBAH TESTIMONIAL	
4. Klik tombol “Edit” pada salah satu testimonial yang ingin diubah	5. Mengambil data testimonial dari basis data 6. Menampilkan halaman edit testimonial yang berisi kolom teks testimonial

-
7. Mengubah kalimat yang perlu
dibenahi pada testimonial

**NORMAL SKENARIO TOMBOL “SUBMIT”
PADA HALAMAN EDIT TESTIMONIAL**

8. Klik tombol “Submit”
9. Menyimpan testimonial ke basis data
10. Menampilkan halaman edit testimonial yang berisi kolom teks testimonial disertai tombol "Submit", "Reset", dan "Back to Manage"

**NORMAL SKENARIO TOMBOL “RESET”
PADA HALAMAN EDIT TESTIMONIAL**

15. Klik tombol “Reset”
16. Mengembalikan isian testimonial seperti semula

**NORMAL SKENARIO TOMBOL “BACK TO MANAGE”
PADA HALAMAN EDIT TESTIMONIAL**

15. Klik tombol “Back to Manage”
17. Mengambil data testimonial
18. Kembali ke halaman Manage Testimonial yang berisi daftar testimonial disertai tombol “Edit”, “Delete”, dan “Back to Manage”

NORMAL SKENARIO MENGHAPUS TESTIMONIAL

11. Klik tombol “Delete” pada salah satu testimonial yang ingin diubah
12. Menampilkan halaman hapus testimonial disertai tombol “Yes” dan “No”

**NORMAL SKENARIO TOMBOL “YES”
PADA HAPUS TESTIMONIAL**

13. Klik tombol “Yes”
14. Menghapus data testimonial dari basis data
15. Mengambil data testimonial dari basis data
16. Menampilkan halaman Manage Testimonial yang berisi daftar testimonial disertai tombol “Edit”, “Delete”, dan “Back to Manage”
-

**NORMAL SKENARIO TOMBOL “NO”
PADA HAPUS TESTIMONIAL**

- | | |
|----------------------|--|
| 13. Klik tombol “No” | 17. Mengambil data testimonial dari basis data |
| | 18. Menampilkan halaman Manage Testimonial yang berisi daftar testimonial disertai tombol “Edit”, “Delete”, dan “Back to Manage” |

Alternatif Skenario Tombol “Submit” Pada Ubah Testimonial: Kolom teks masih kosong atau tidak diisi

ALTERNATIF SKENARIO KOLOM TEKS MASIH KOSONG

- | |
|--|
| 16. Kolom teks masih kosong atau tidak diisi |
| 17. Menampilkan pesan error |

20. *Usecase* Skenario Menambah Testimonial

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario menambah testimonial dapat dilihat pada Tabel 20.

Tabel 20. *Usecase* Skenario Menambah Testimonial

Nama <i>Usecase</i>	: Menambah Testimonial
ID <i>Usecase</i>	: GFO208
Aktor	: <i>Member</i>
Pre Kondisi	: Aktor memilih tombol “Tulis Testimonial”
Post Kondisi	: Aktor berhasil menambah testimonial

NORMAL SKENARIO MENAMBAH TESTIMONIAL

Aktor	Sistem
1. Klik tombol “Tulis Testimonial” pada kolom testimonial	2. Menampilkan halaman Testimonial yang berisi kolom teks testimonial disertai tombol “Submit” dan “Reset”
3. Mengisi kolom teks testimonial	

**NORMAL SKENARIO TOMBOL “SUBMIT”
PADA HALAMAN TESTIMONIAL**

-
- 4. Klik tombol “Submit”
 - 5. Menyimpan data ke basis data
 - 6. Menampilkan halaman Testimonial yang berisi kolom teks testimonial dan pesan bahwa testimonial telah ditambahkan, disertai tombol “Submit” dan “Reset”

**NORMAL SKENARIO TOMBOL “RESET”
PADA HALAMAN TESTIMONIAL**

- 7. Klik tombol “Reset”
 - 8. Menghapus semua isian pada kolom teks testimonial
-

Alternatif Skenario Menambah Testimonial: Kolom teks masih kosong atau tidak diisi

ALTERNATIF SKENARIO KOLOM TEKS MASIH KOSONG

- 6.a Kolom teks masih kosong atau tidak diisi
 - 7.a Menampilkan pesan error
-

21. *Usecase* Skenario Melihat Testimonial

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario melihat testimonial dapat dilihat pada Tabel 21.

Tabel 21. *Usecase* Skenario Melihat Testimonial

Nama <i>Usecase</i>	: Melihat Testimonial
ID <i>Usecase</i>	: GFO209
Aktor	: <i>Member</i>
Pre Kondisi	: Aktor memilih tombol “Lihat Semua Testimonial”
Post Kondisi	: Aktor melihat semua testimonial

NORMAL SKENARIO MELIHAT SEMUA TESTIMONIAL

Aktor	Sistem
-------	--------

-
1. Klik tombol “Lihat Semua Testimonial” pada kolom Testimonial
 2. Mengambil data testimonial dari basis data
 3. Menampilkan halaman Testimonial berisi seluruh testimonial yang pernah ditambahkan
-

22. *Usecase* Skenario Melihat Cara *Order*

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario melihat cara *order* dapat dilihat pada Tabel 22.

Tabel 22. *Usecase* Skenario Melihat Cara *Order*

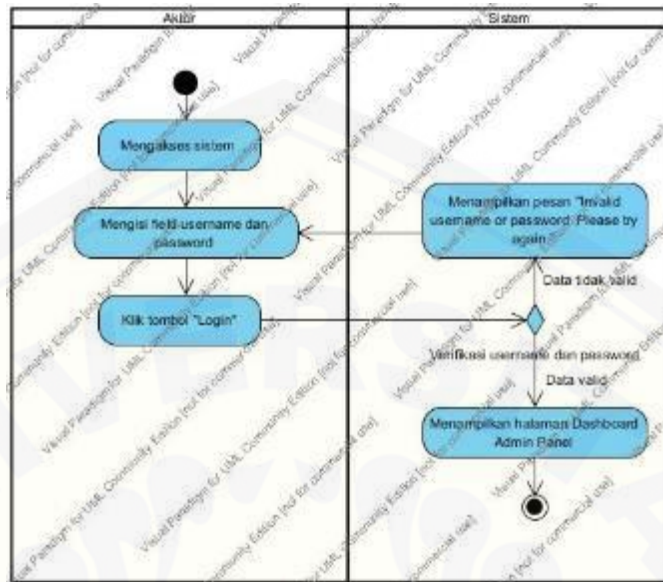
Nama <i>Usecase</i>	: Melihat Cara <i>Order</i>
ID <i>Usecase</i>	: GFO210
Aktor	: <i>Member</i>
Pre Kondisi	: Aktor memilih menu “Cara <i>Order</i> ”
Post Kondisi	: Aktor melihat cara <i>order</i>

NORMAL SKENARIO MELIHAT CARA <i>ORDER</i>	
Aktor	Sistem
1. Klik menu “Cara <i>Order</i> ”	2. Menampilkam halaman Cara <i>Order</i>

B. *Activity Diagram*

B.1. *Activity Diagram Login Admin dan CEO*

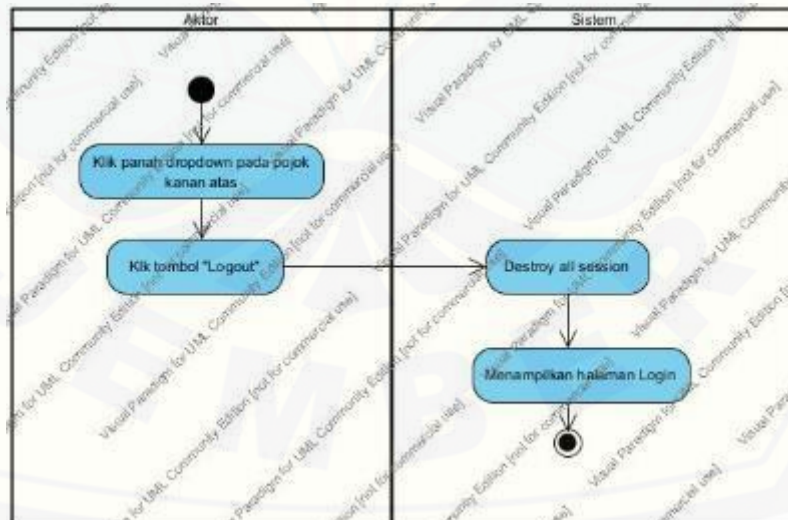
Activity diagram login dapat dilakukan oleh *admin* dan CEO. *Activity diagram login* menjelaskan tentang bagaimana *admin* dan CEO dapat masuk ke dalam sistem informasi penjualan GFO. *Activity diagram login* dapat dilihat pada Gambar 1.



Gambar 1. Activity Diagram Login CEO

B.2. Activity Diagram Logout Admin dan CEO

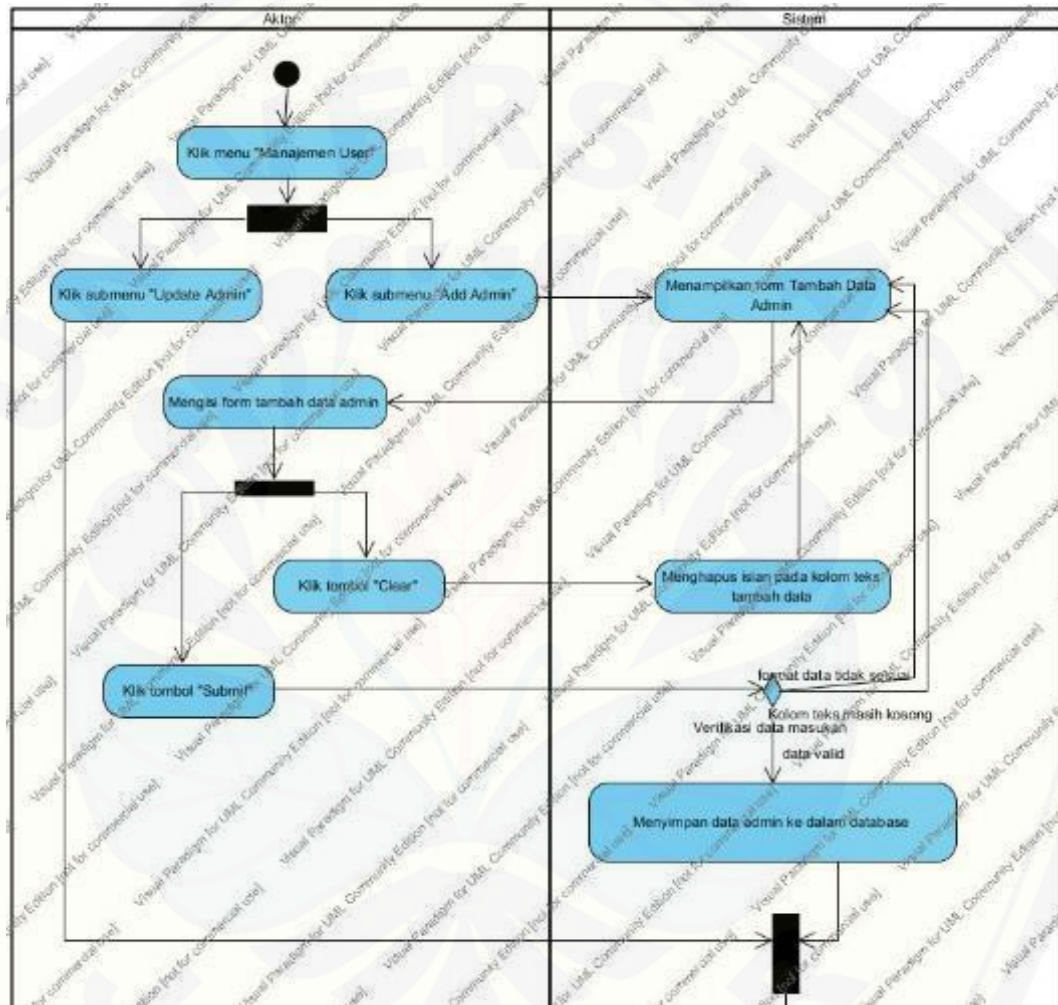
Activity diagram *logout* dapat dilakukan oleh *admin* dan CEO. Activity diagram *logout* menjelaskan tentang bagaimana *admin* dan CEO keluar dari sistem informasi penjualan GFO. Activity diagram *logout* dapat dilihat pada Gambar 2.

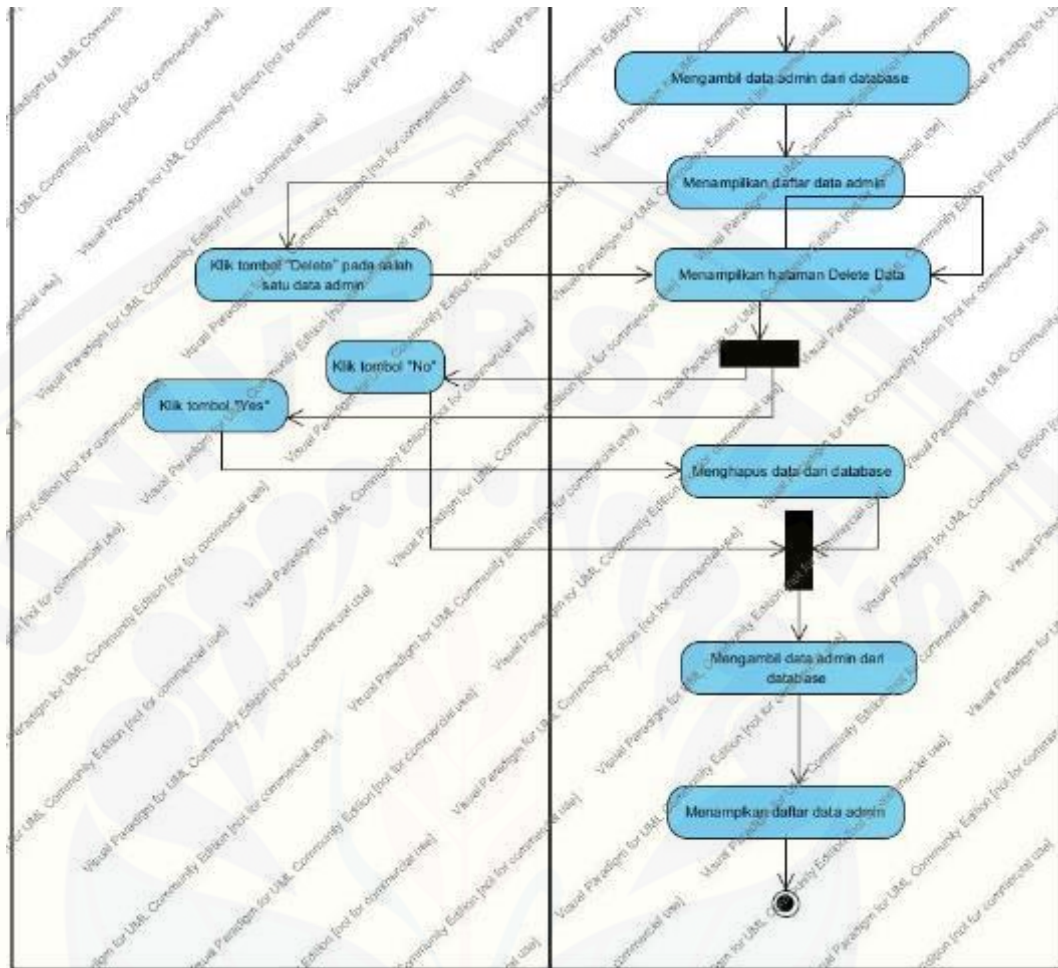


Gambar 2. Activity Diagram Logout Admin dan CEO

B.3. Activity Diagram Mengelola User Admin

Activity diagram mengelola *admin* dapat dilakukan oleh CEO. Activity diagram mengelola *admin* menjelaskan tentang bagaimana CEO dapat menambah dan menghapus data admin. Activity diagram *logout* dapat dilihat pada Gambar 3.

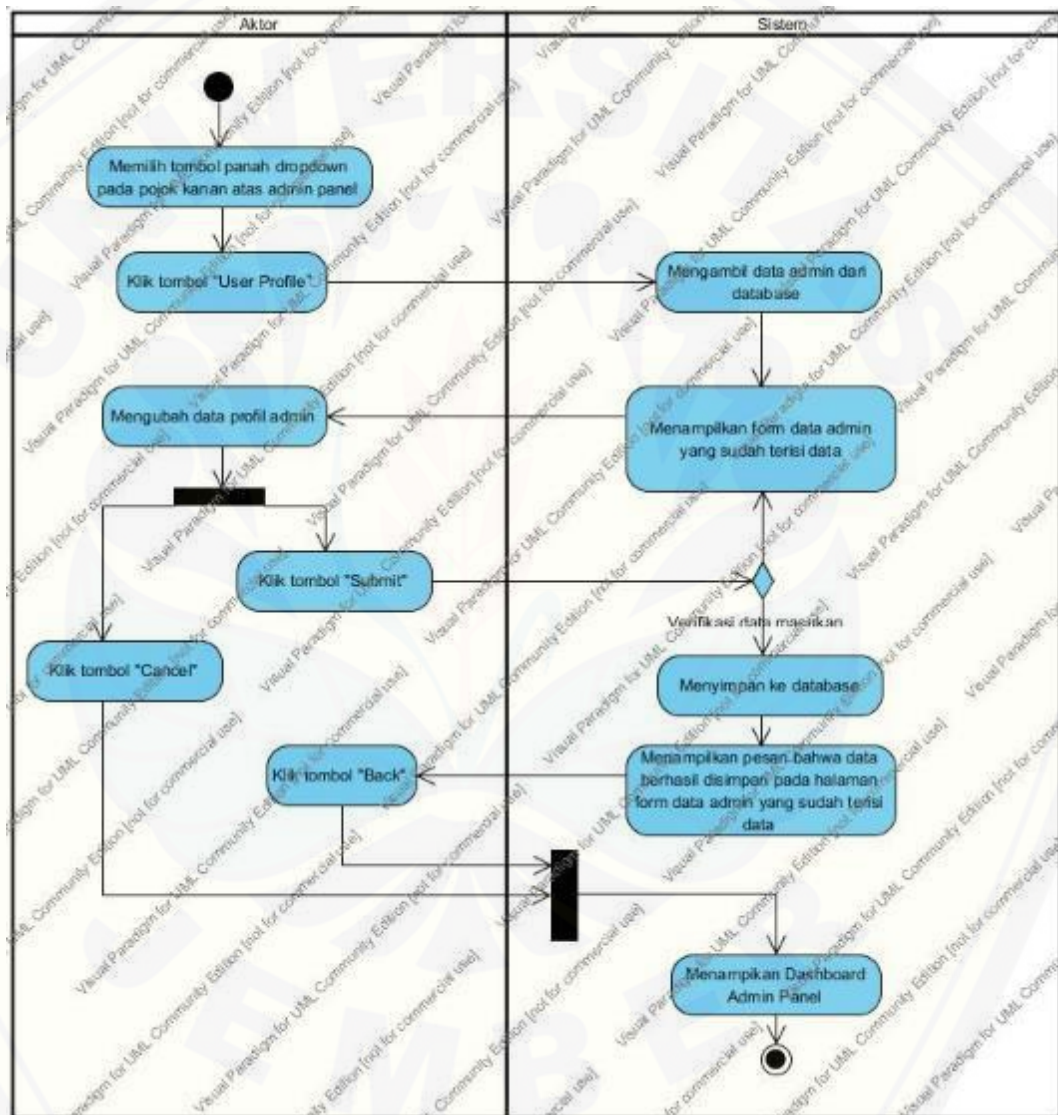




Gambar 3. Activity Diagram Mengelola User Admin

B.4. Activity Diagram Mengubah Data Profil Admin

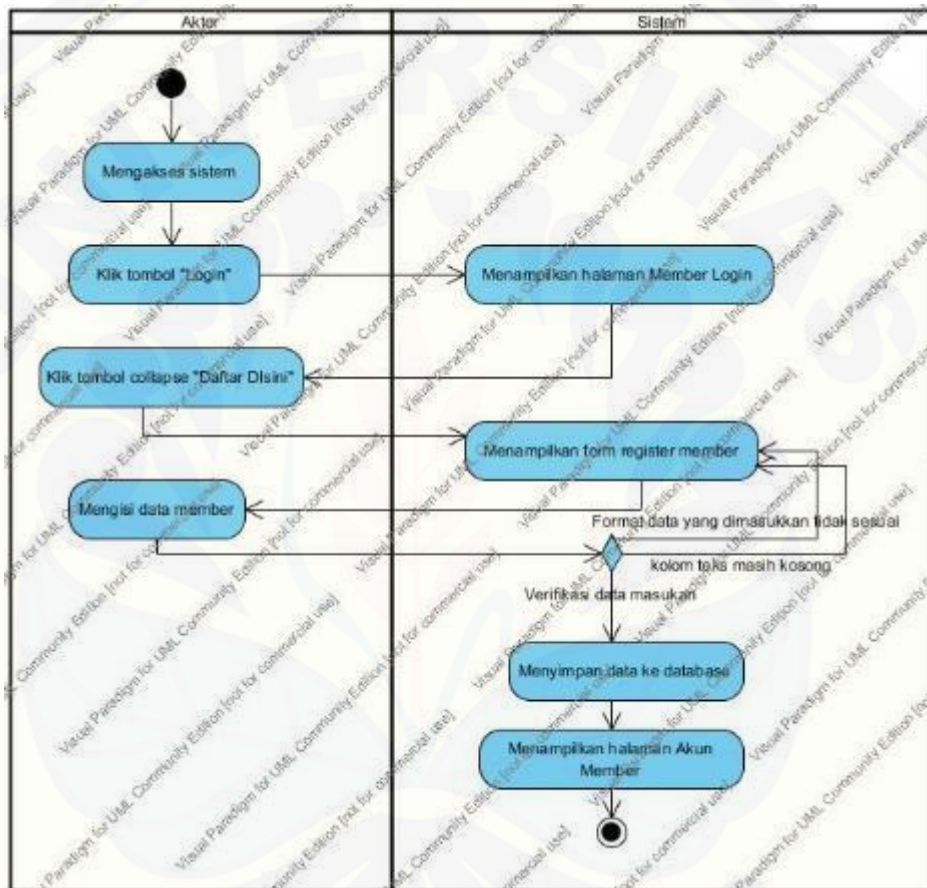
Activity diagram mengelola *admin* dapat dilakukan oleh *admin*. Activity diagram mengubah data profil admin menjelaskan tentang bagaimana admin mengubah data diri yang sebelumnya telah ditambahkan oleh CEO. Activity diagram mengubah data profil admin dapat dilihat pada Gambar 4.



Gambar 4. Activity Diagram Mengubah Data Profil Admin

B.5. Activity Diagram Register Member

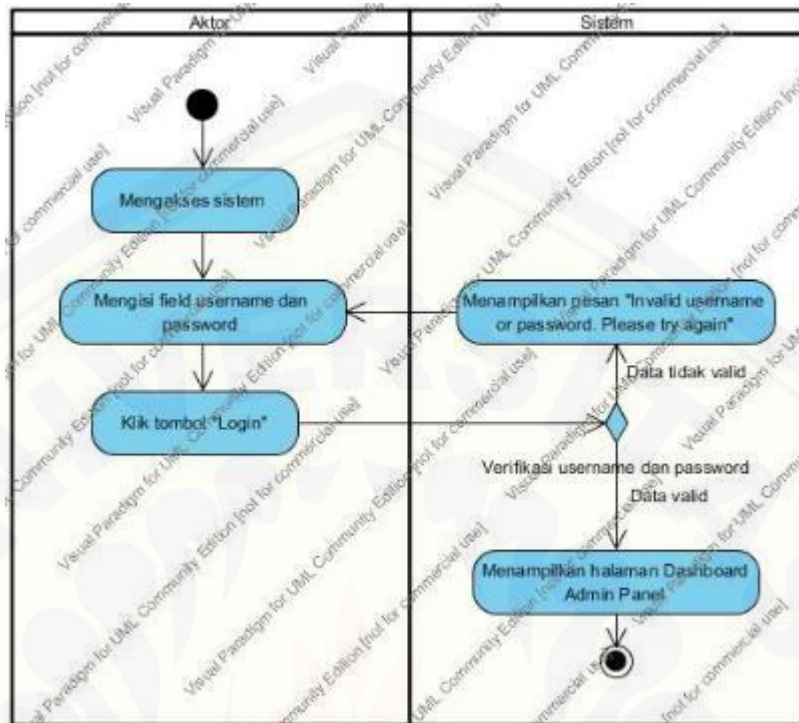
Activity diagram mengelola *admin* dapat dilakukan oleh *member*. Activity diagram register *member* menjelaskan tentang bagaimana *member* baru mendaftarkan data dirinya agar dapat melakukan transaksi pada sistem informasi penjualan GFO. Activity diagram register *member* dapat dilihat pada Gambar 5.



Gambar 5. Activity Diagram Register Member

B.6. Activity Diagram Login Member

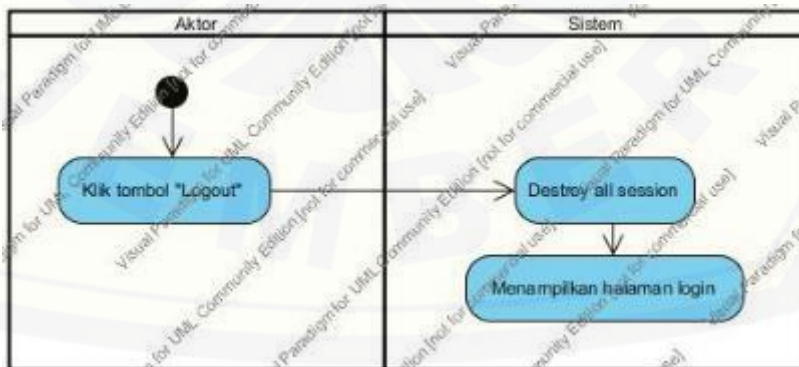
Activity diagram *login member* dapat dilakukan oleh *member*. Activity diagram *login member* menjelaskan tentang bagaimana *member* harus melakukan login terlebih dahulu sebelum dapat melakukan transaksi. Activity diagram *login member* dapat dilihat pada Gambar 6.



Gambar 6. Activity Diagram Login Member

B.7. Activity Diagram Logout Member

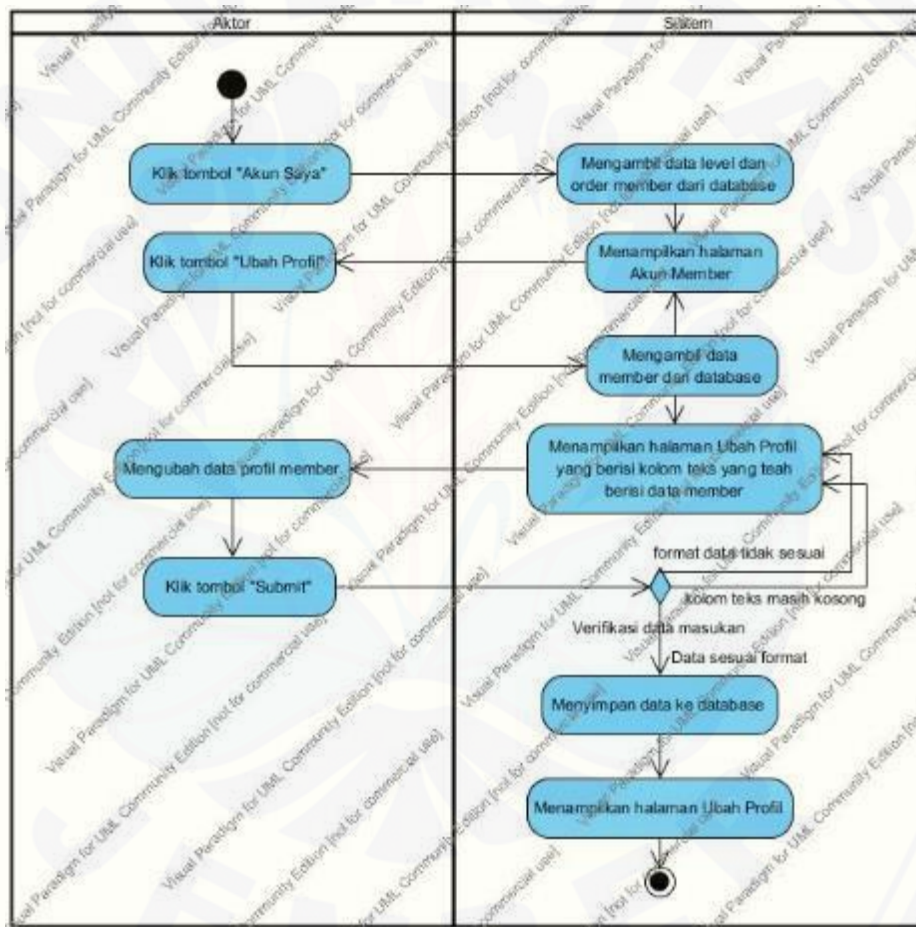
Activity diagram *logout member* dapat dilakukan oleh *member*. Activity diagram *logout member* menjelaskan tentang bagaimana *member* dapat keluar dari akun yang dimiliki namun masih bisa tetap berada di sistem. Activity diagram *logout member* dapat dilihat pada Gambar 7.



Gambar 7. Activity Diagram Logout Member

B.8. Activity Diagram Mengubah Data *Member*

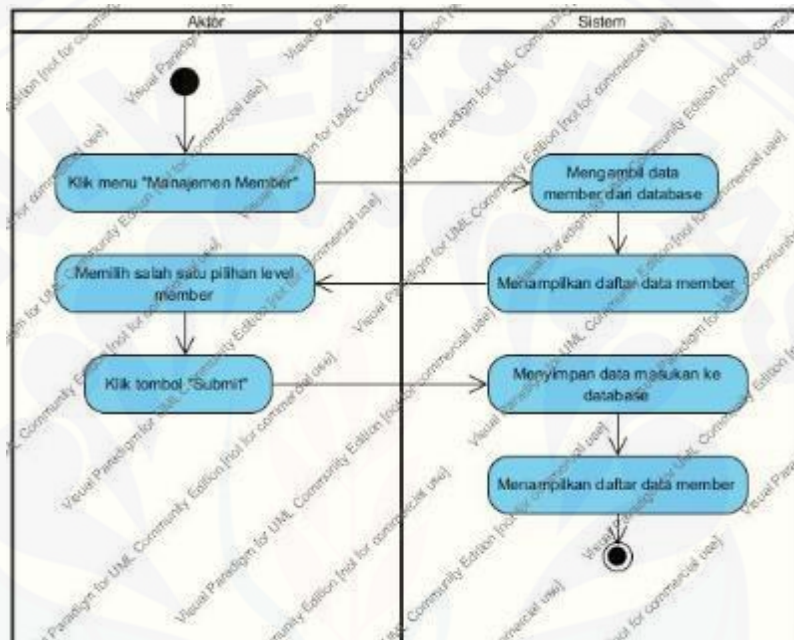
Activity diagram mengubah data *member* dapat dilakukan oleh *member*. Activity diagram mengubah data *member* menjelaskan tentang bagaimana *member* dapat mengubah data diri dari yang telah didaftarkan pada saat melakukan register. Activity diagram mengubah data *member* dapat dilihat pada Gambar 8.



Gambar 8. Activity Diagram Mengubah Data *Member*

B.9. Activity Diagram Mengubah Level *Member*

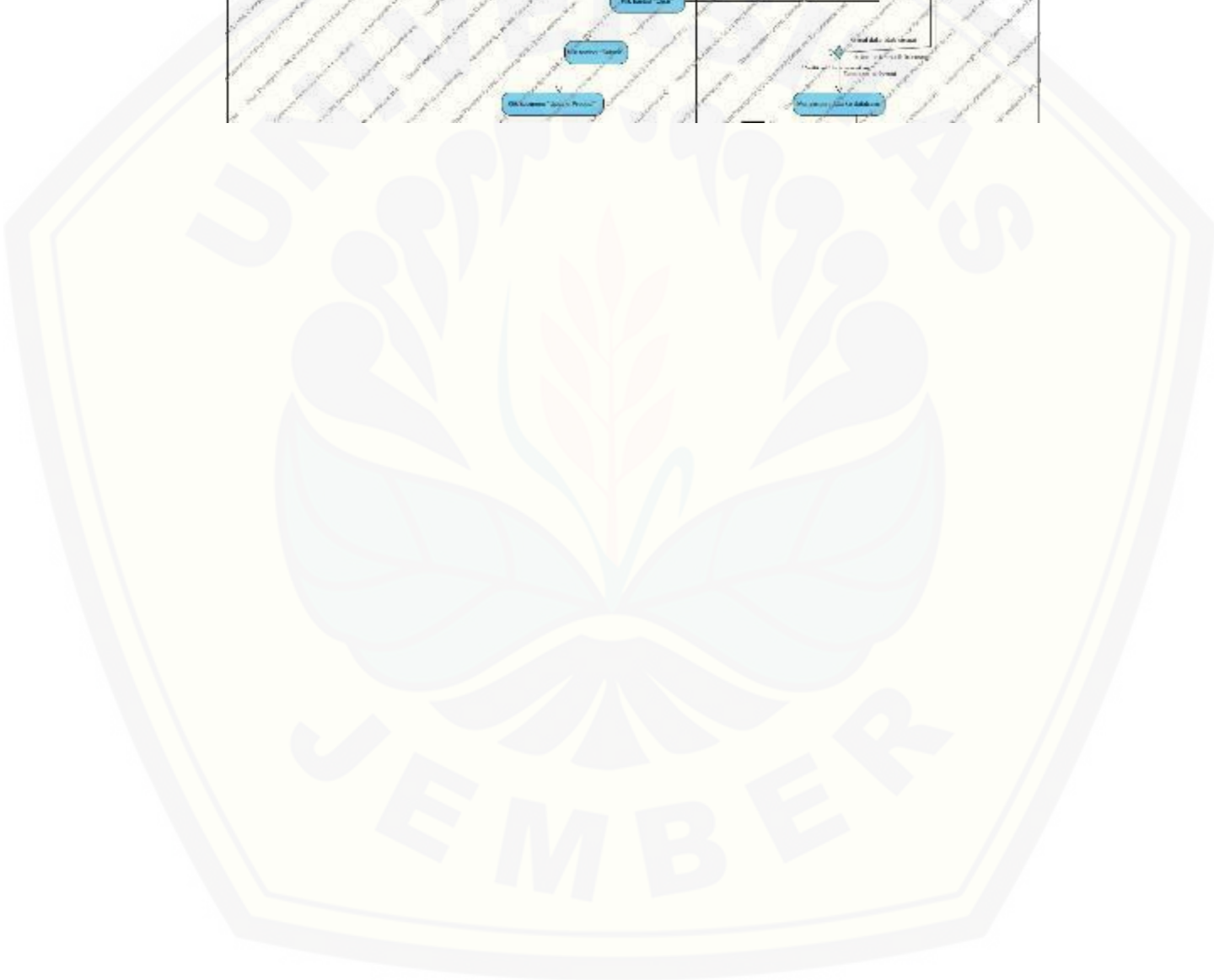
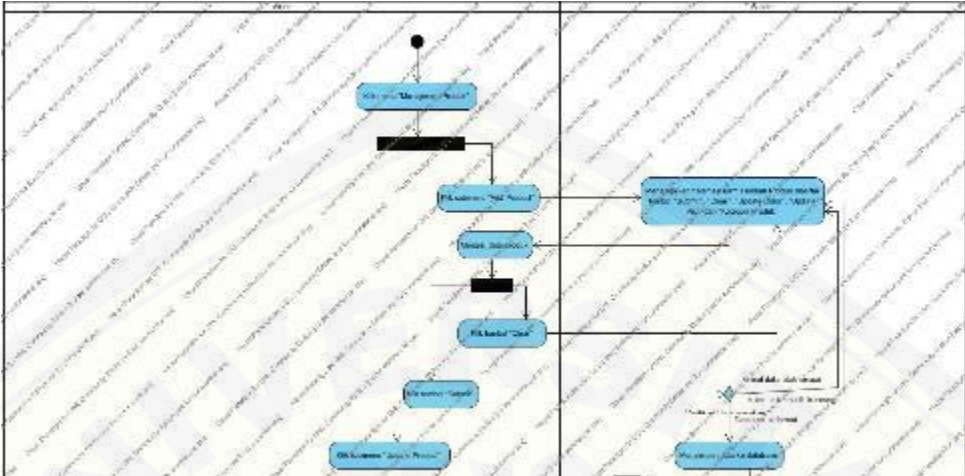
Activity diagram mengubah level member dapat dilakukan oleh admin. *Activity* diagram ini menjelaskan tentang bagaimana admin mengubah level member sesuai dengan poin yang didapat member dari akumulasi total belanja. *Activity* diagram mengubah level member dapat dilihat pada Gambar 9.

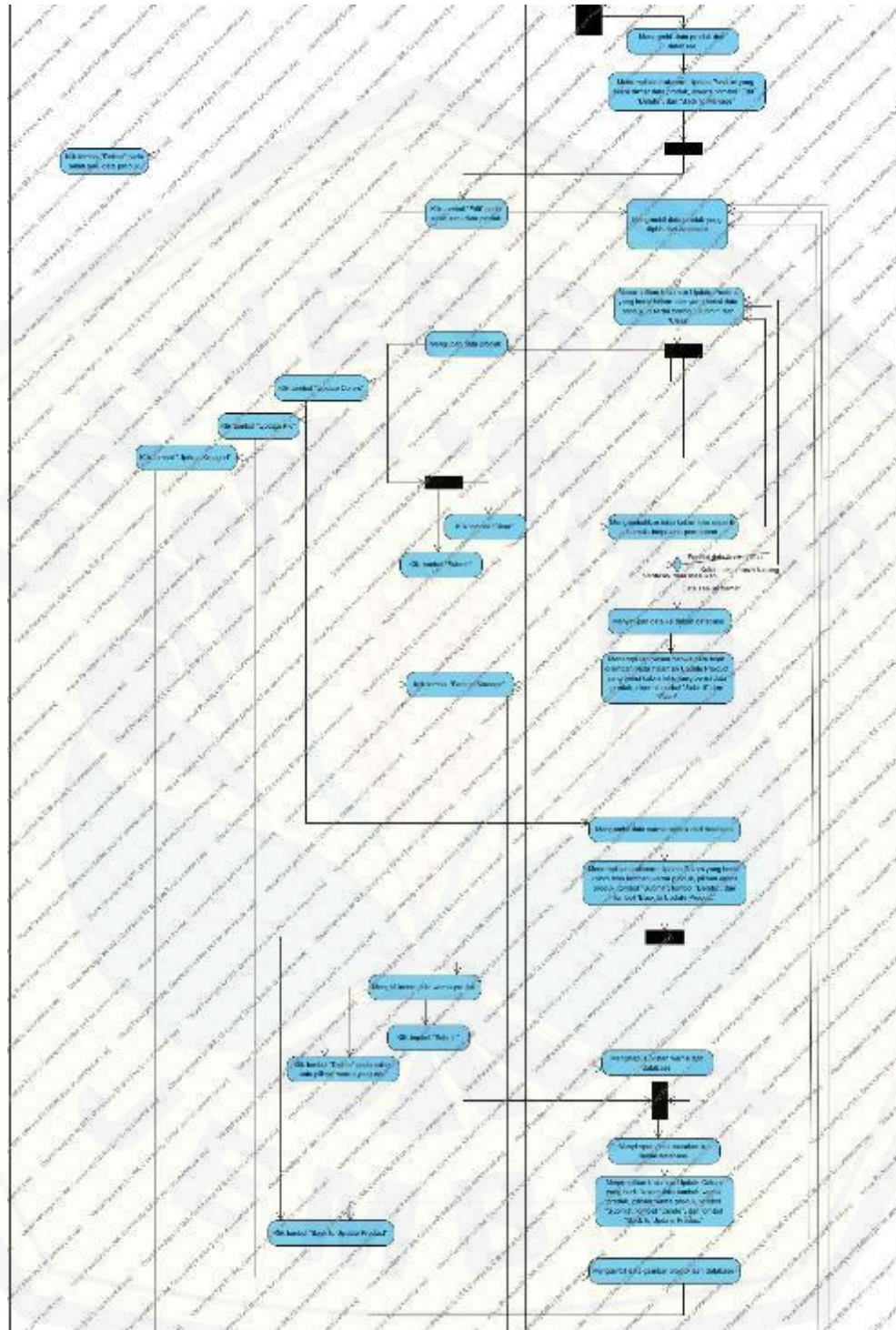


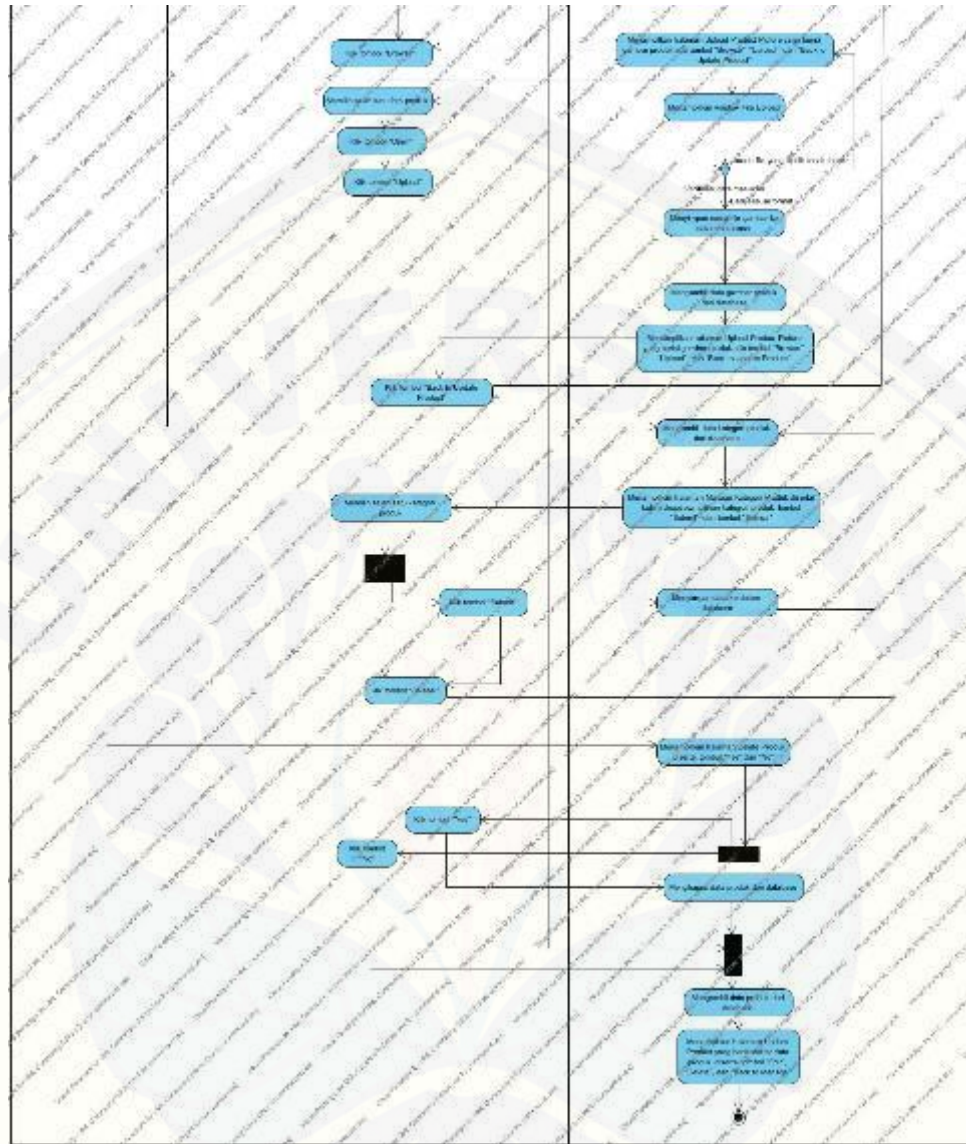
Gambar 9. Activity Diagram Mengubah Level *Member*

B.10. Activity Diagram Mengelola Data Produk

Activity diagram mengelola data produk dapat dilakukan oleh admin. *Activity* diagram ini menjelaskan tentang bagaimana admin dapat menambah, mengubah, dan menghapus data produk. *Activity* diagram mengelola data produk dapat dilihat pada Gambar 10.



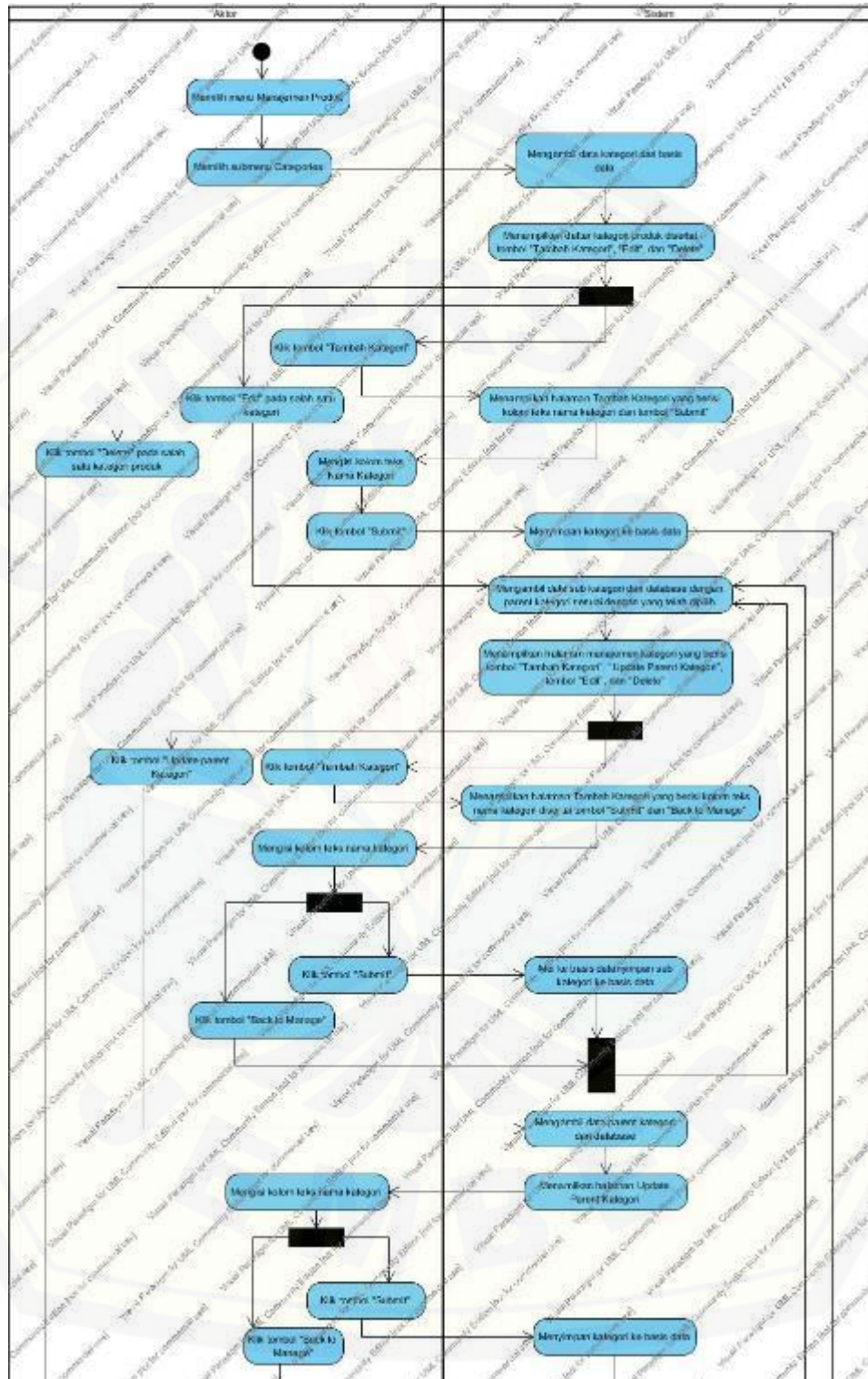


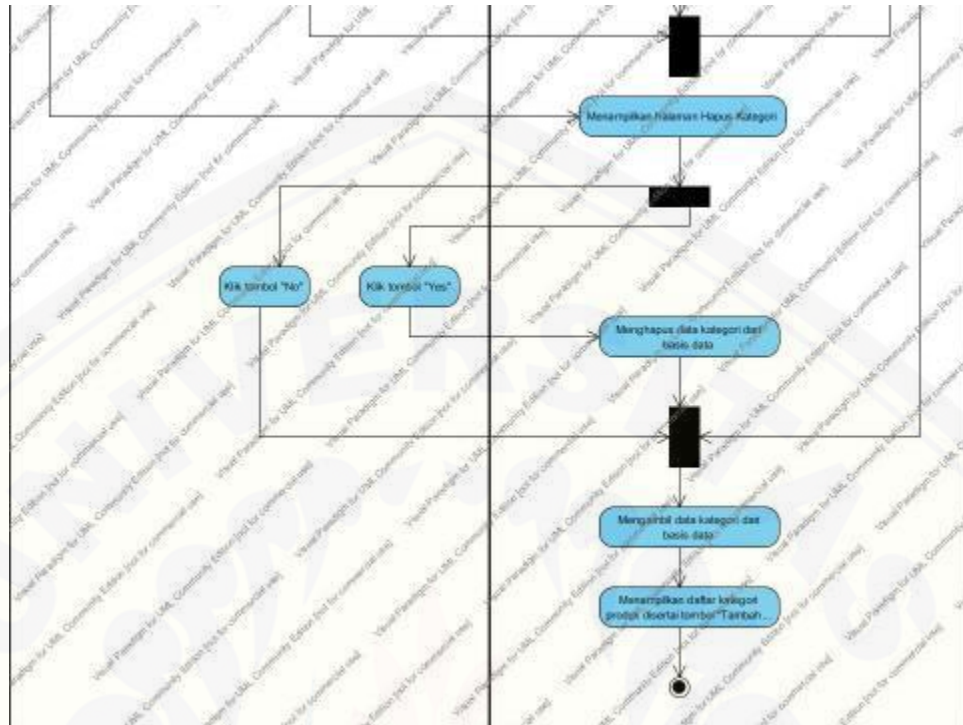


Gambar 10. Activity Diagram Mengelola Data Produk

B.11. Activity Diagram Mengelola Kategori

Activity diagram mengelola kategori dapat dilakukan oleh admin. Activity diagram ini menjelaskan tentang admin yang dapat menambah, mengubah, dan menghapus kategori produk yang dapat dijadikan pilihan pada saat menambah produk. Activity diagram ini dapat dilihat pada Gambar 11.



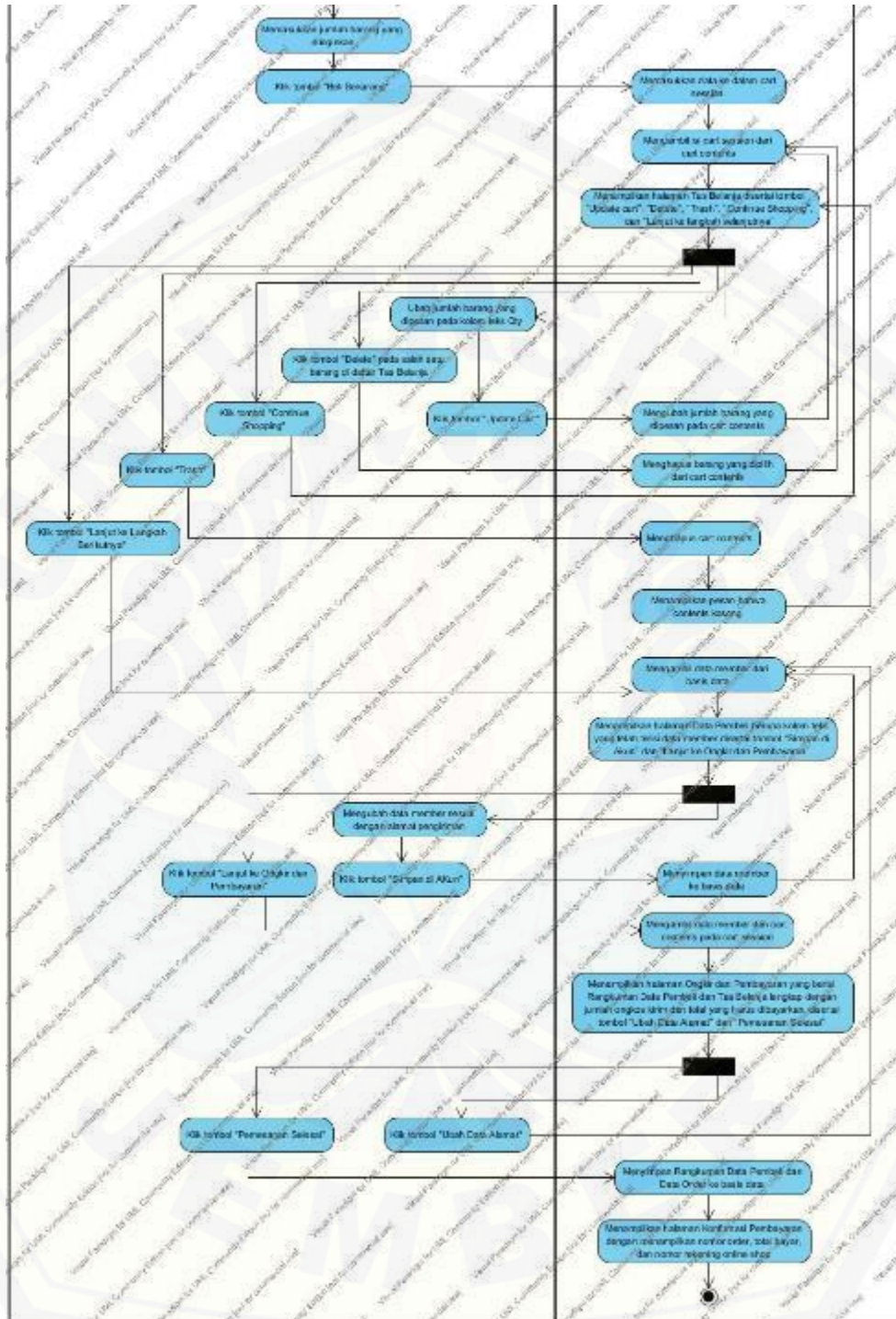


Gambar 11. Activity Diagram Mengelola Kategori Produk

B.12. Activity Diagram Mengelola Data Order

Activity Diagram mengelola data *order* dapat dilakukan oleh member. Activity Diagram ini menjelaskan tentang bagaimana member memulai hingga mengakhiri proses pemesanan barang pada sistem informasi penjualan GFO. Activity Diagram mengelola data *order* dapat dilihat pada Gambar 12.

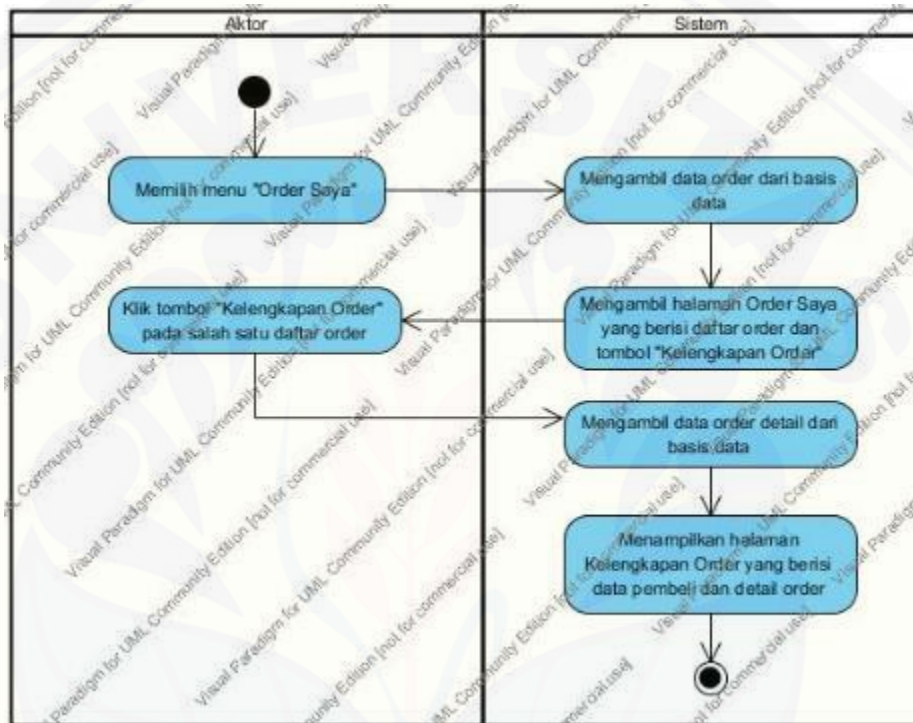




Gambar 12. Activity Diagram Mengelola Data Order

B.13. Activity Diagram Melihat Daftar Order

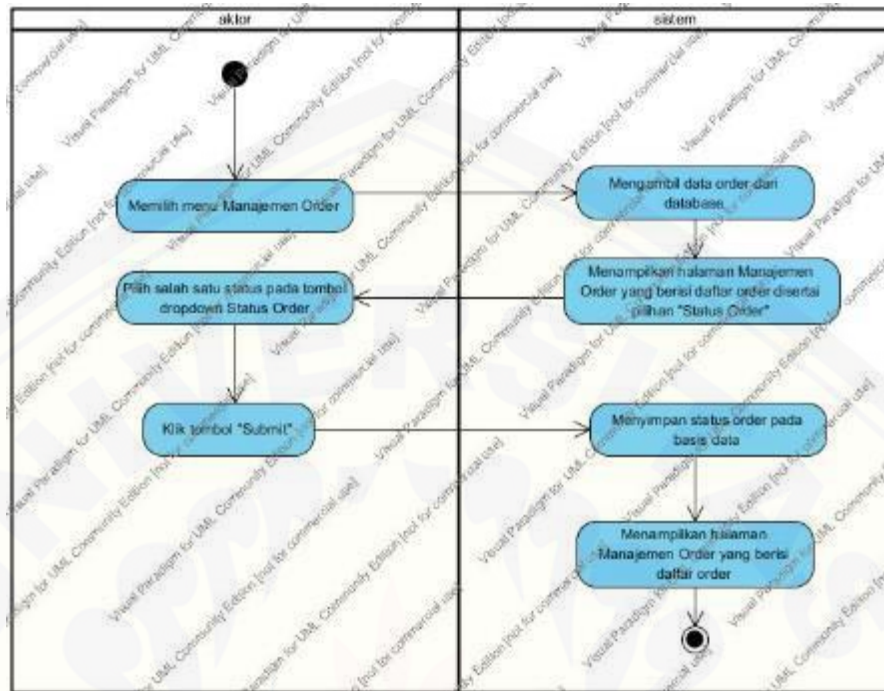
Activity diagram melihat daftar *order* dapat dilakukan oleh member. Activity Diagram ini menjelaskan tentang member yang dapat melihat status pesannya sekaligus dapat melihat detail pesanan yang pernah dilakukan. Activity Diagram melihat daftar *order* dapat dilihat pada Gambar 13.



Gambar 13. Activity Diagram Melihat Daftar Order

B.14. Activity Diagram Mengubah Status Order

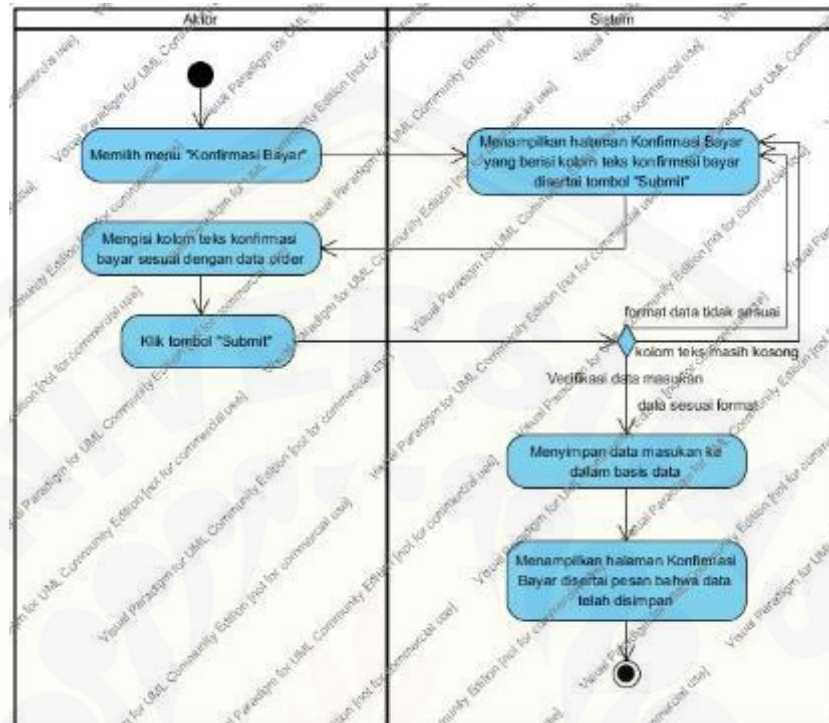
Activity diagram mengelola status *order* dapat dilakukan oleh admin. Activity diagram ini menjelaskan tentang bagaimana admin dapat mengubah status *order* sehingga member yang melakukan pemesanan dapat memantau proses barang pesannya. Activity diagram mengelola data *order* dapat dilihat pada Gambar 14.



Gambar 14. Activity Diagram Mengubah Status *Order*

B.15. Activity Diagram Menambah Data Konfirmasi Bayar

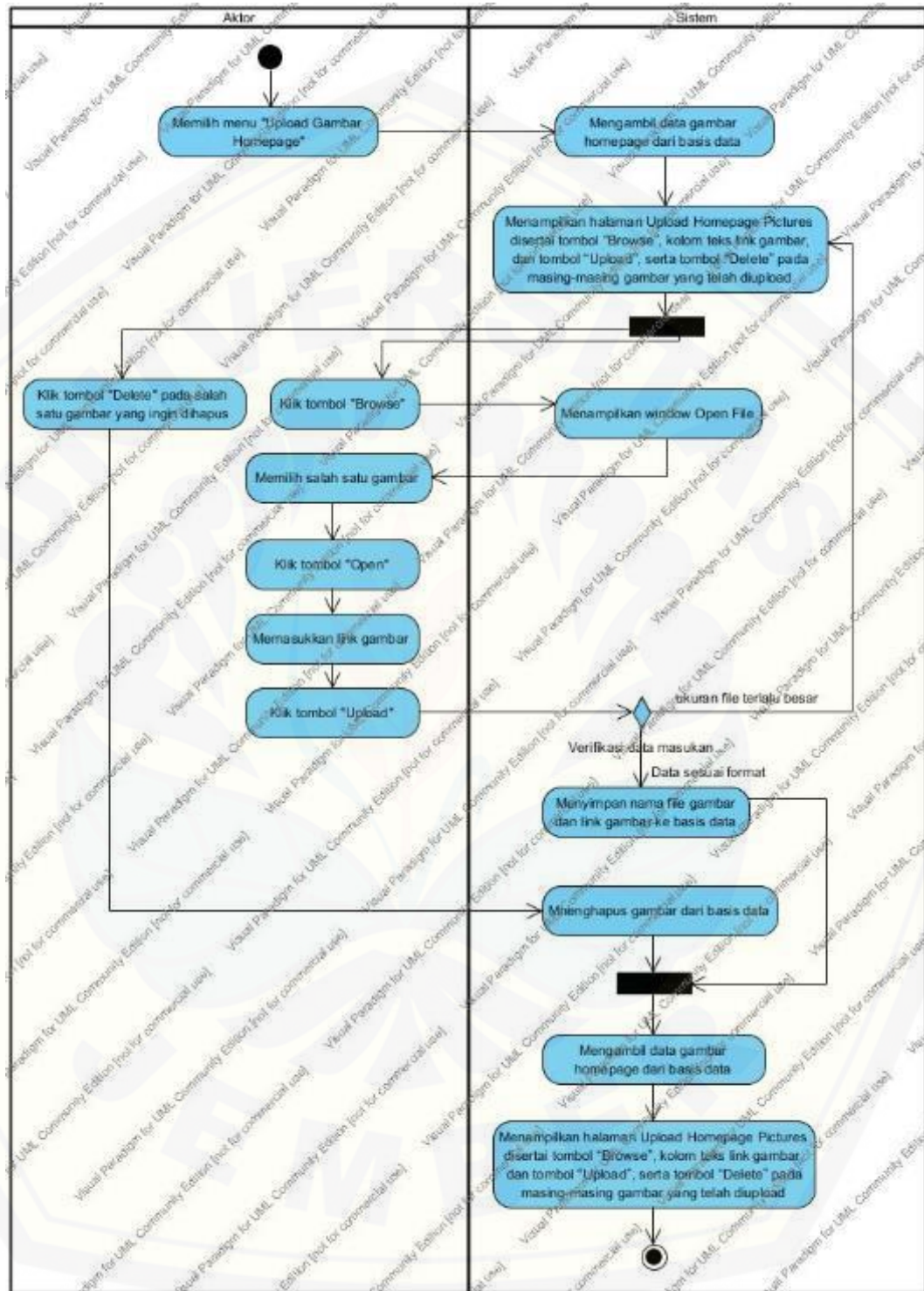
Activity Diagram menambah data konfirmasi bayar dilakukan oleh member. *Activity Diagram* ini menjelaskan tentang member menambahkan data konfirmasi bayar setelah melakukan pembayaran terhadap pesannya. *Activity Diagram* ini dapat dilihat pada Gambar 15.



Gambar 15. Activity Diagram Menambah Data Konfirmasi Bayar

B.16. Activity Diagram Mengelola Gambar *Landing Page* Member

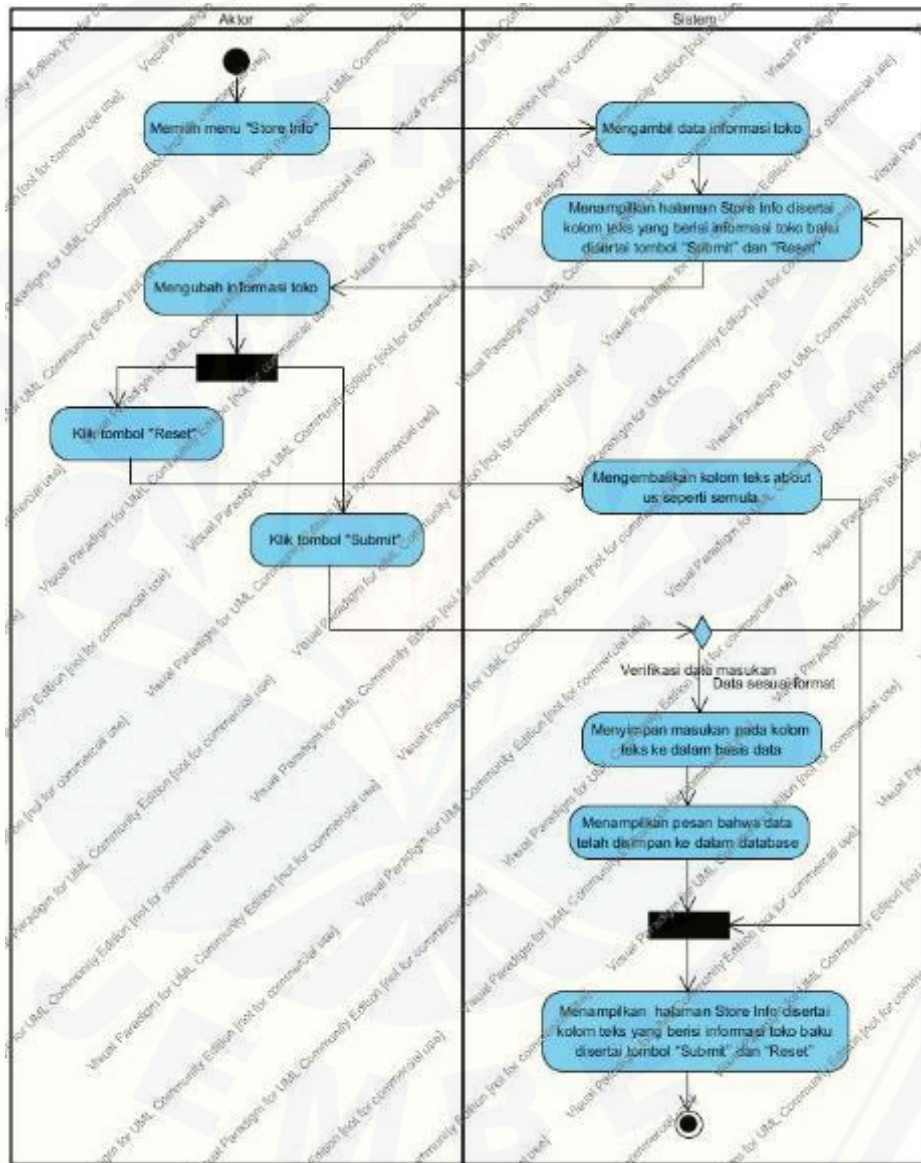
Activity Diagram mengelola gambar landing page member dilakukan oleh admin. *Activity* Diagram ini menjelaskan tentang admin yang dapat menambah dan menghapus gambar untuk halaman *landing page* member. *Activity* Diagram mengelola gambar landing page member dapat dilihat pada Gambar 16.



Gambar 16. Activity Diagram Mengelola Gambar Landing Page

B.17. Activity Diagram Mengubah Informasi Toko

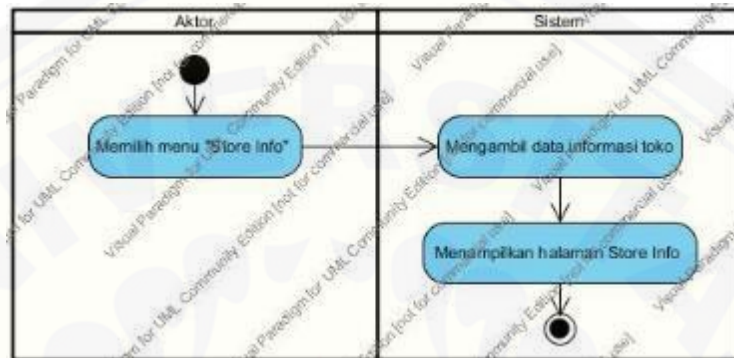
Activity Diagram mengubah informasi toko dilakukan oleh admin. Activity Diagram ini menjelaskan proses admin mengubah informasi yang berkaitan dengan GFO. Activity Diagram mengubah informasi toko dapat dilihat pada Gambar 17.



Gambar 17. Activity Diagram Mengubah Informasi Toko

B.18. Activity Diagram Melihat Informasi Toko

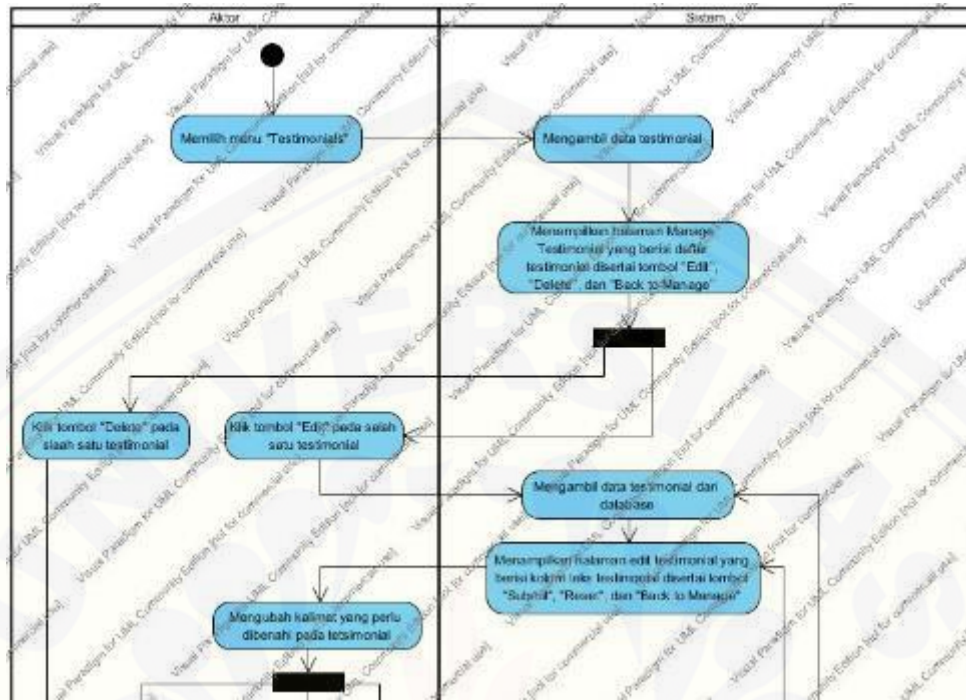
Activity Diagram melihat informasi toko dapat dilakukan oleh member. Activity Diagram ini menjelaskan tentang bagaimana member melihat halaman Store Info. Activity Diagram ini dapat dilihat pada Gambar 18.

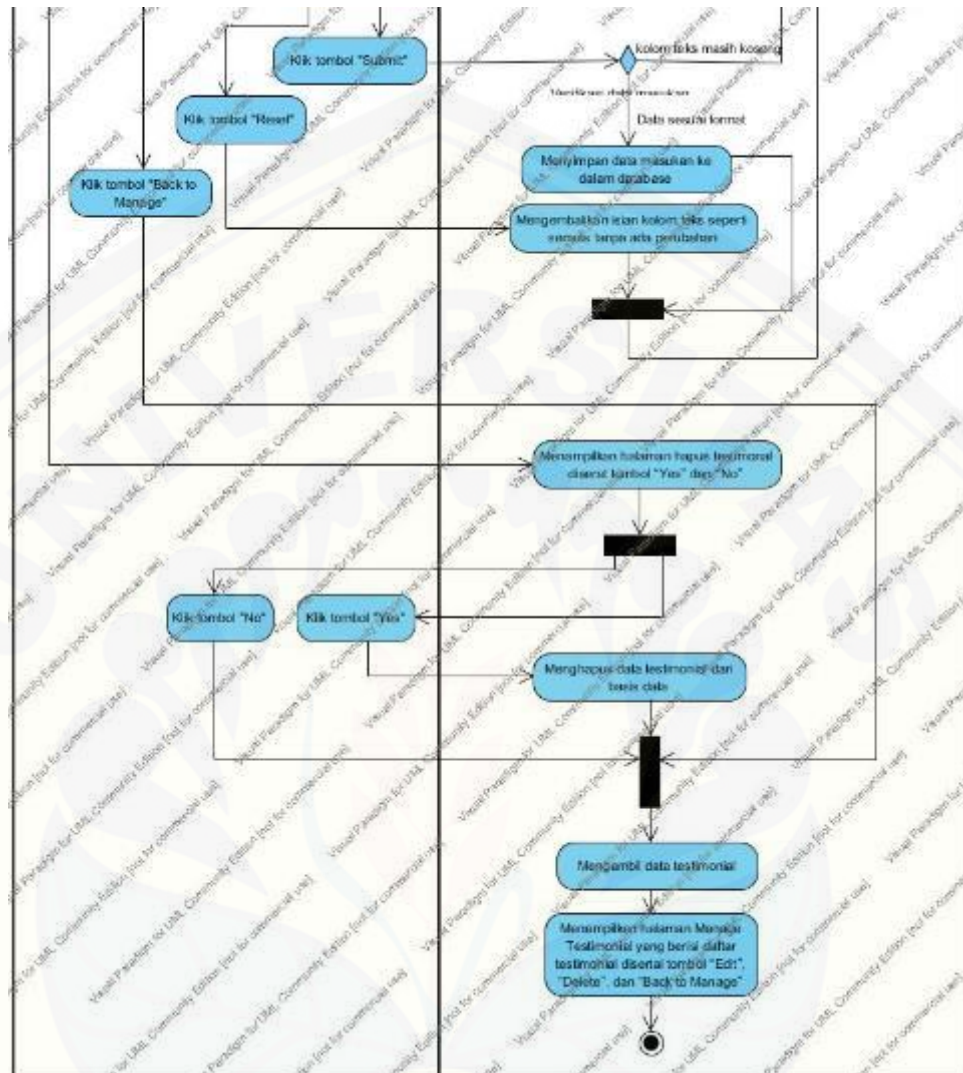


Gambar 18. Activity Diagram Melihat Informasi Toko

B.19. Activity Diagram Mengubah dan Menghapus Testimonial

Activity Diagram mengubah dan menghapus testimonial dapat dilakukan oleh admin. Activity Diagram ini menjelaskan bagaimana admin mengubah beberapa kalimat testimonial yang pernah ditambahkan member. Activity Diagram mengubah dan menghapus testimonial dapat dilihat pada Gambar 19.

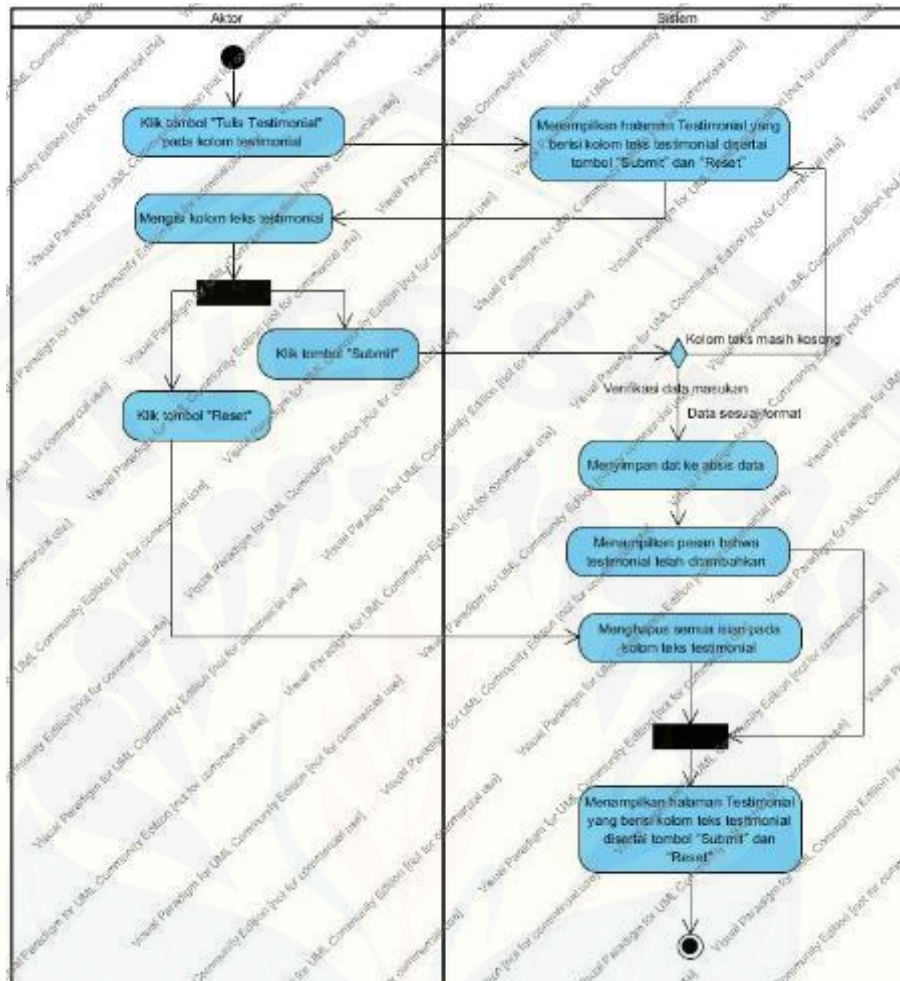




Gambar 19. Activity Diagram Mengubah dan Menghapus Testimonial

B.20. Activity Diagram Menambah Testimonial

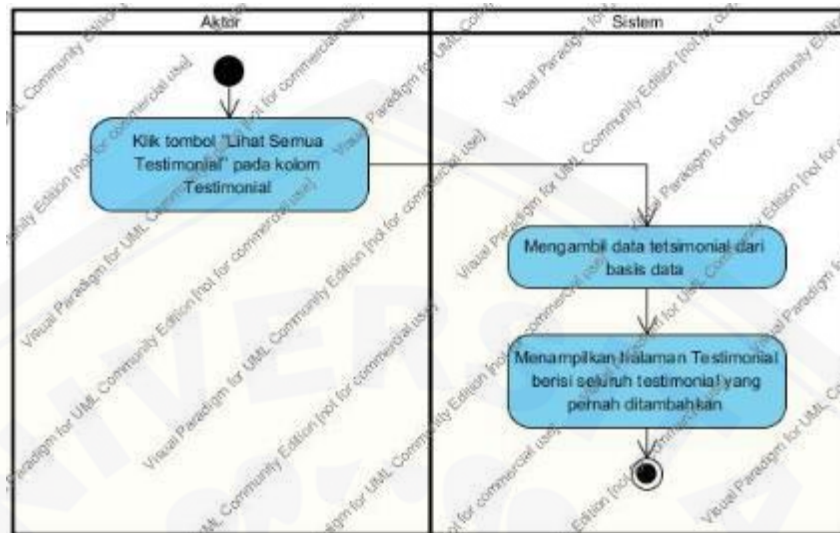
Activity Diagram menambah testimonial dapat dilakukan oleh member. Activity Diagram ini menjelaskan bagaimana member menuliskan testimonial pada halaman Tambah Testimonial dan menyimpannya. Activity Diagram menambah testimonial dapat dilihat pada Gambar 20.



Gambar 20. Activity Diagram Menambah Testimonial

B.21. Activity Diagram Melihat Testimonial

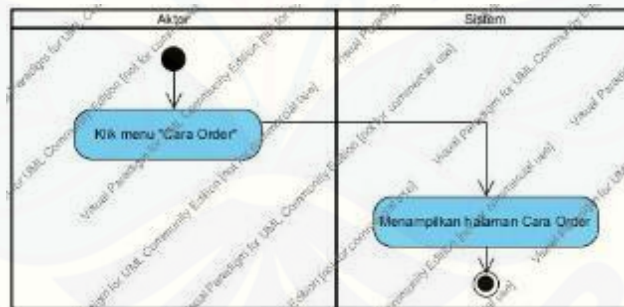
Activity Diagram melihat testimonial dapat dilakukan oleh member. Activity Diagram ini menjelaskan tentang member yang dapat melihat seluruh testimonial yang pernah ditambahkan oleh member lain. Activity Diagram melihat testimonial dapat dilihat pada Gambar 21.



Gambar 21. Activity Diagram Melihat Testimonial

B.22. Activity Diagram Melihat Cara Order

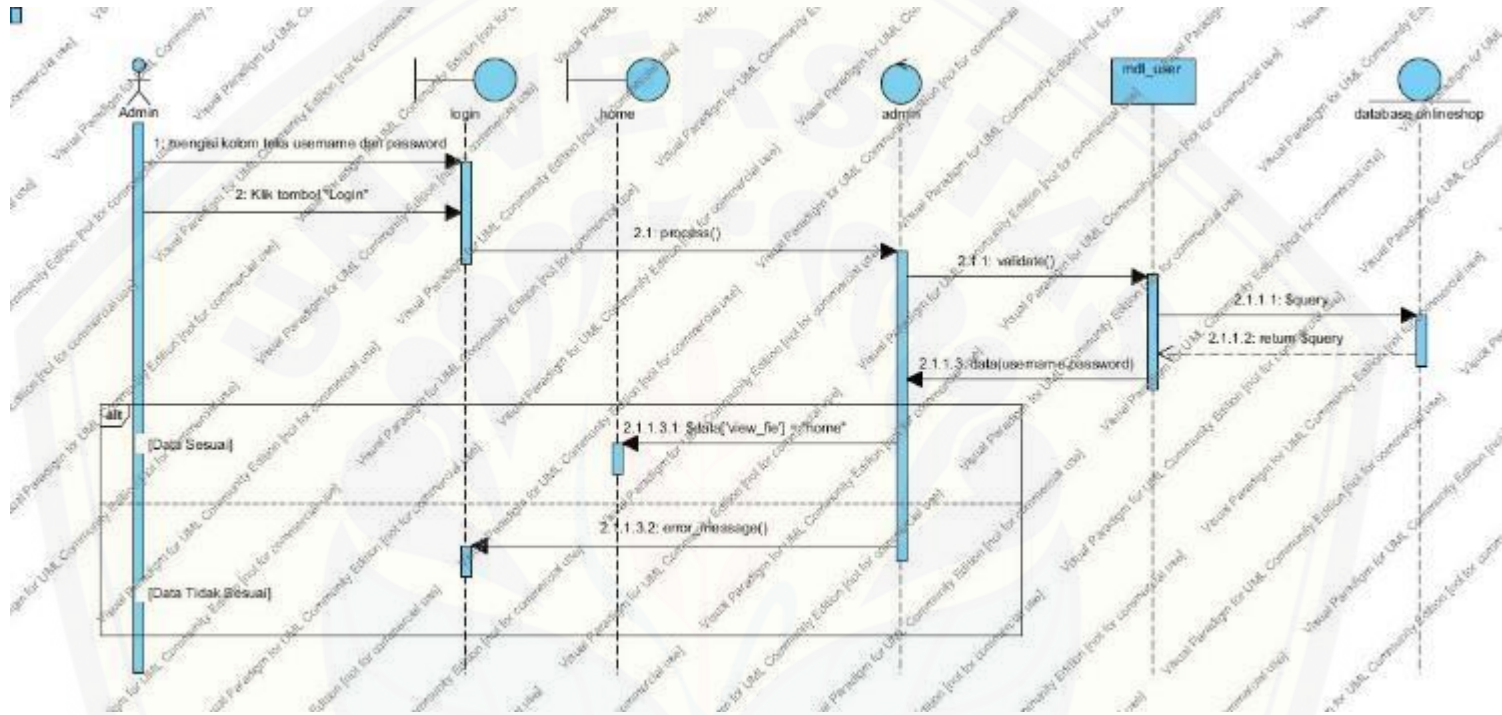
Activity Diagram melihat cara *order* dapat dilakukan oleh member. Activity Diagram ini menjelaskan tentang member yang dapat membuka halaman Cara Order pada sistem informasi penjualan GFO. Activity Diagram dapat dilihat pada Gambar 22.



Gambar 22. Activity Diagram Melihat Cara Order

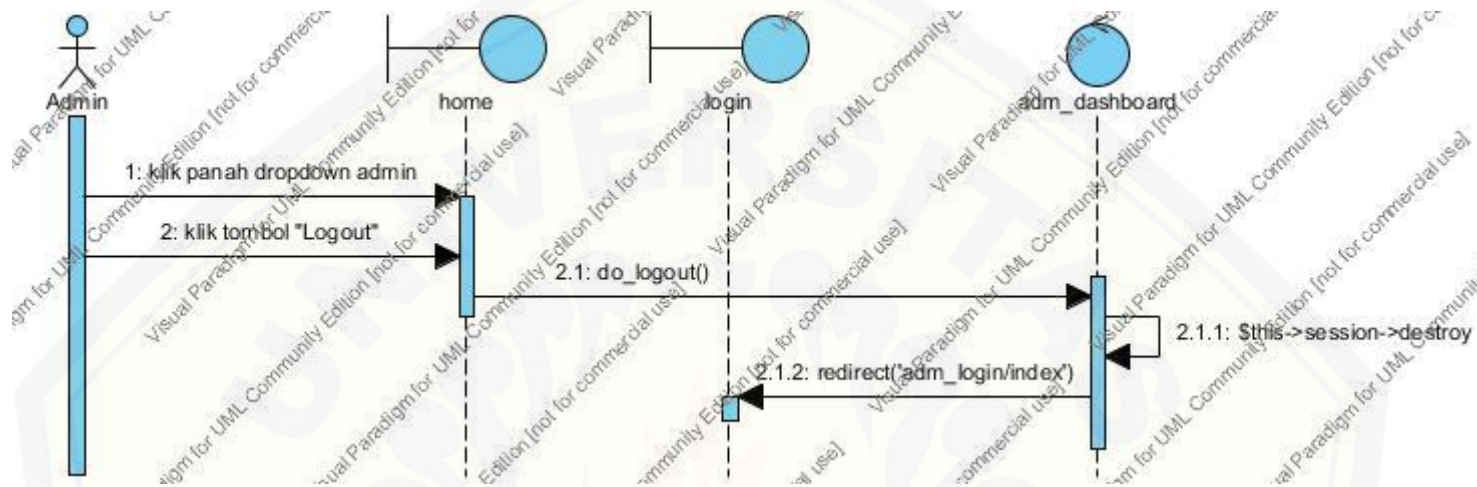
C. Sequence Diagram

C.1. Sequence Diagram Login Admin dan CEO



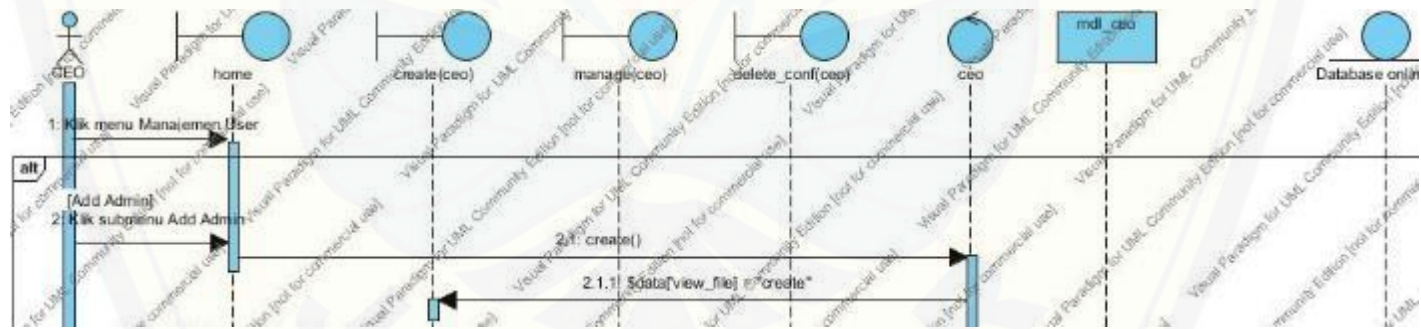
Gambar 23. Sequence Diagram Login Admin dan CEO

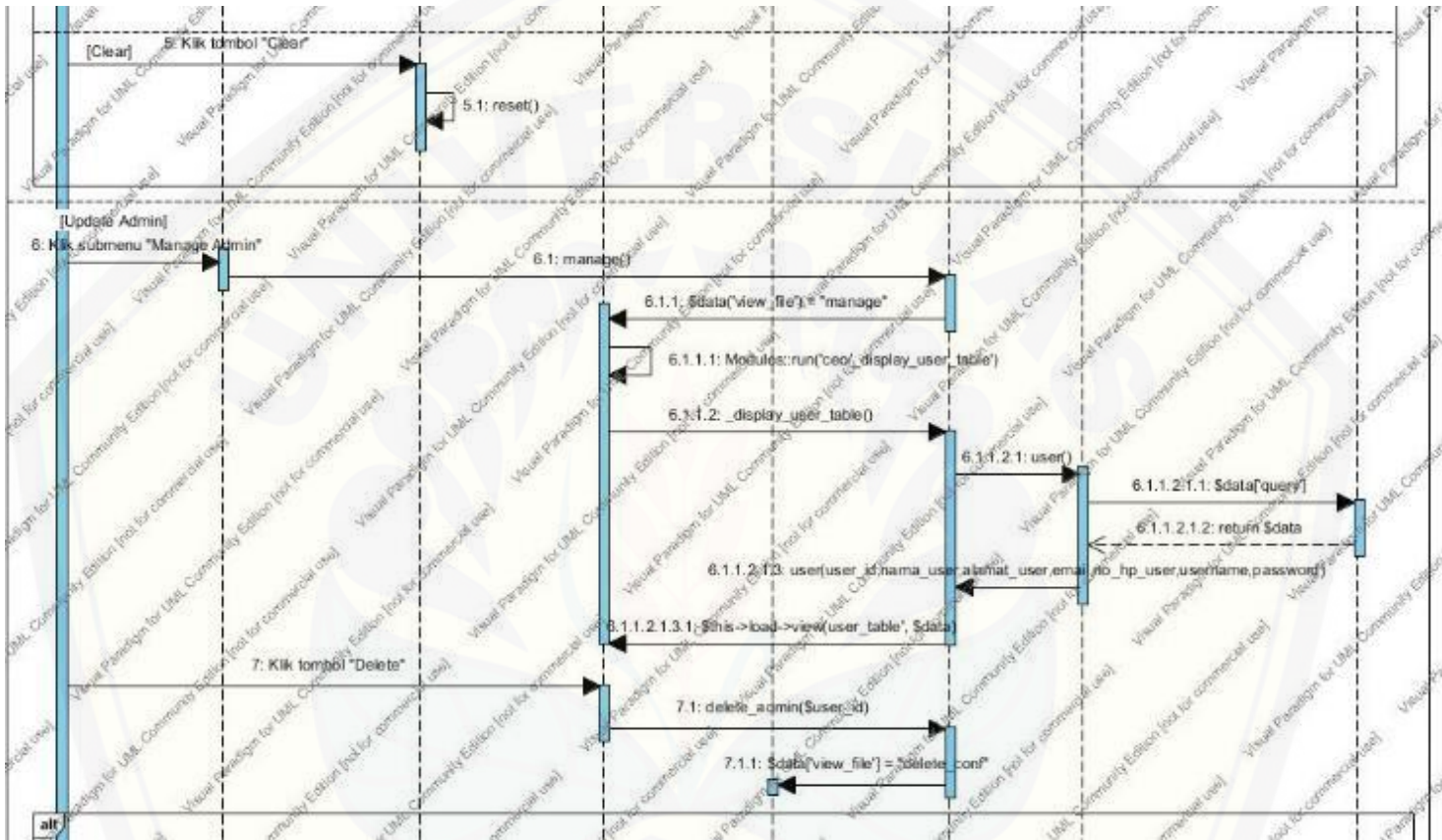
C.2. Sequence Diagram Logout Admin dan CEO

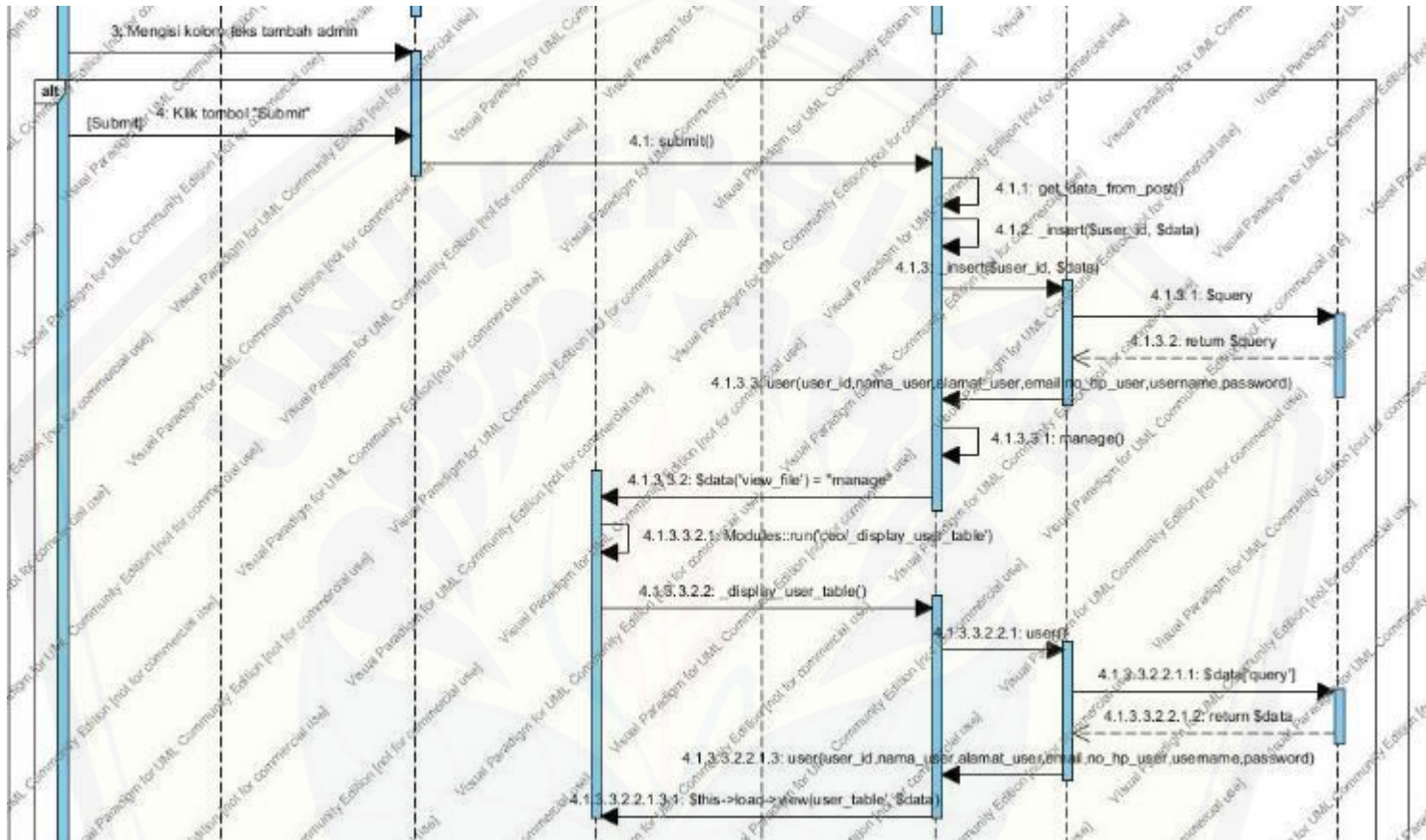


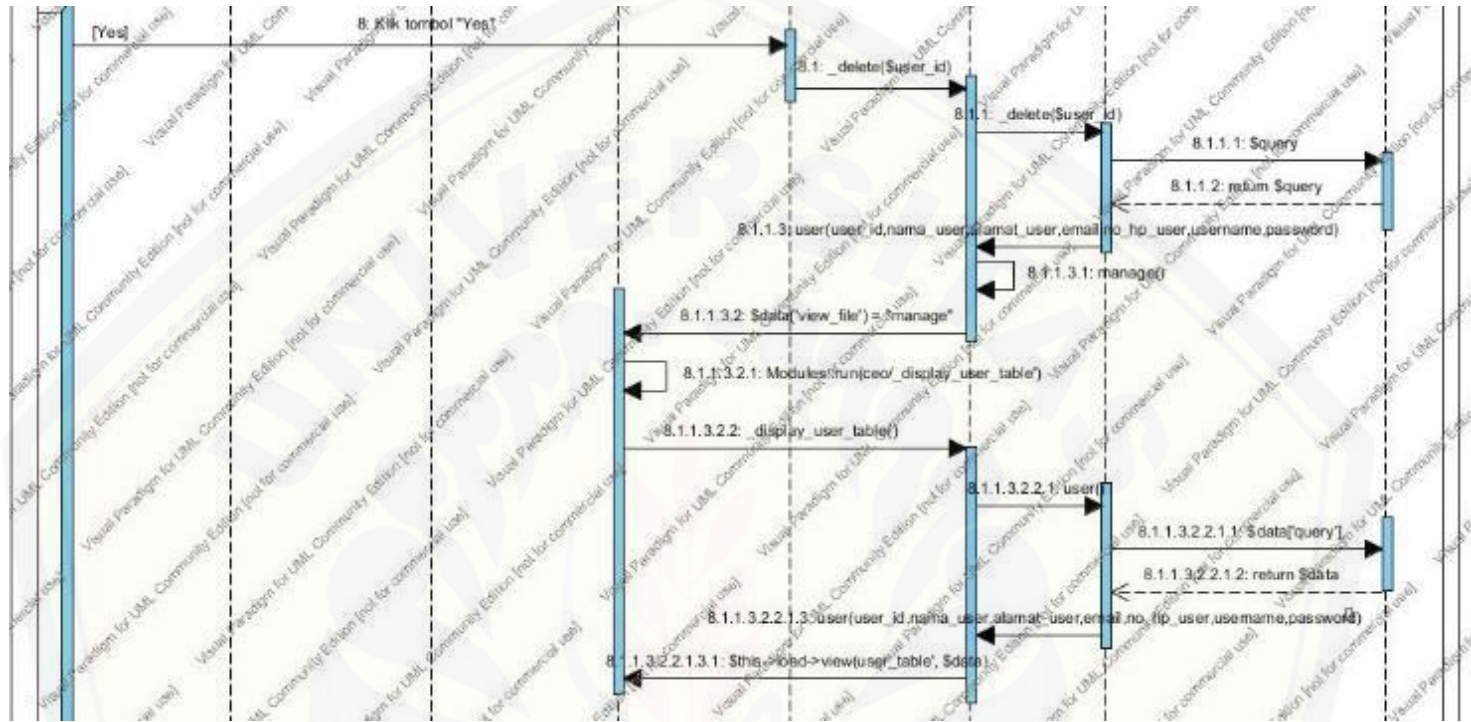
Gambar 24. Sequence Diagram Logout Admin dan CEO

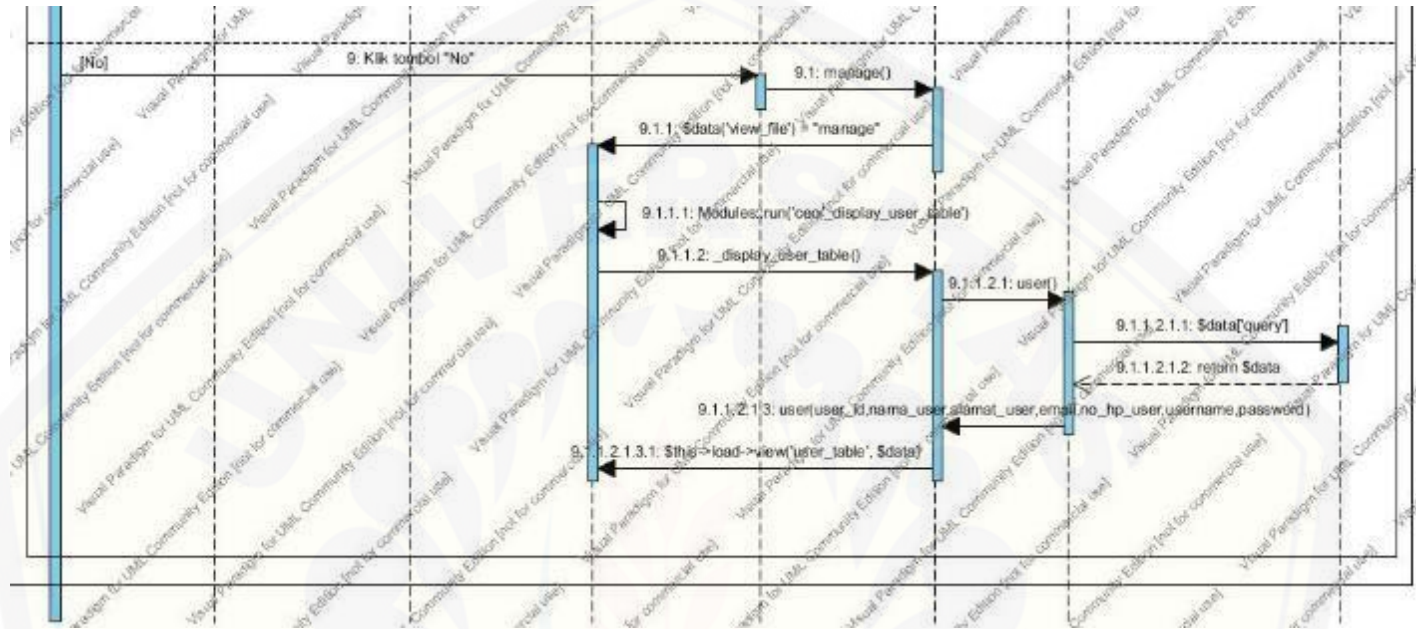
C.3. Sequence Diagram Mengelola User Admin





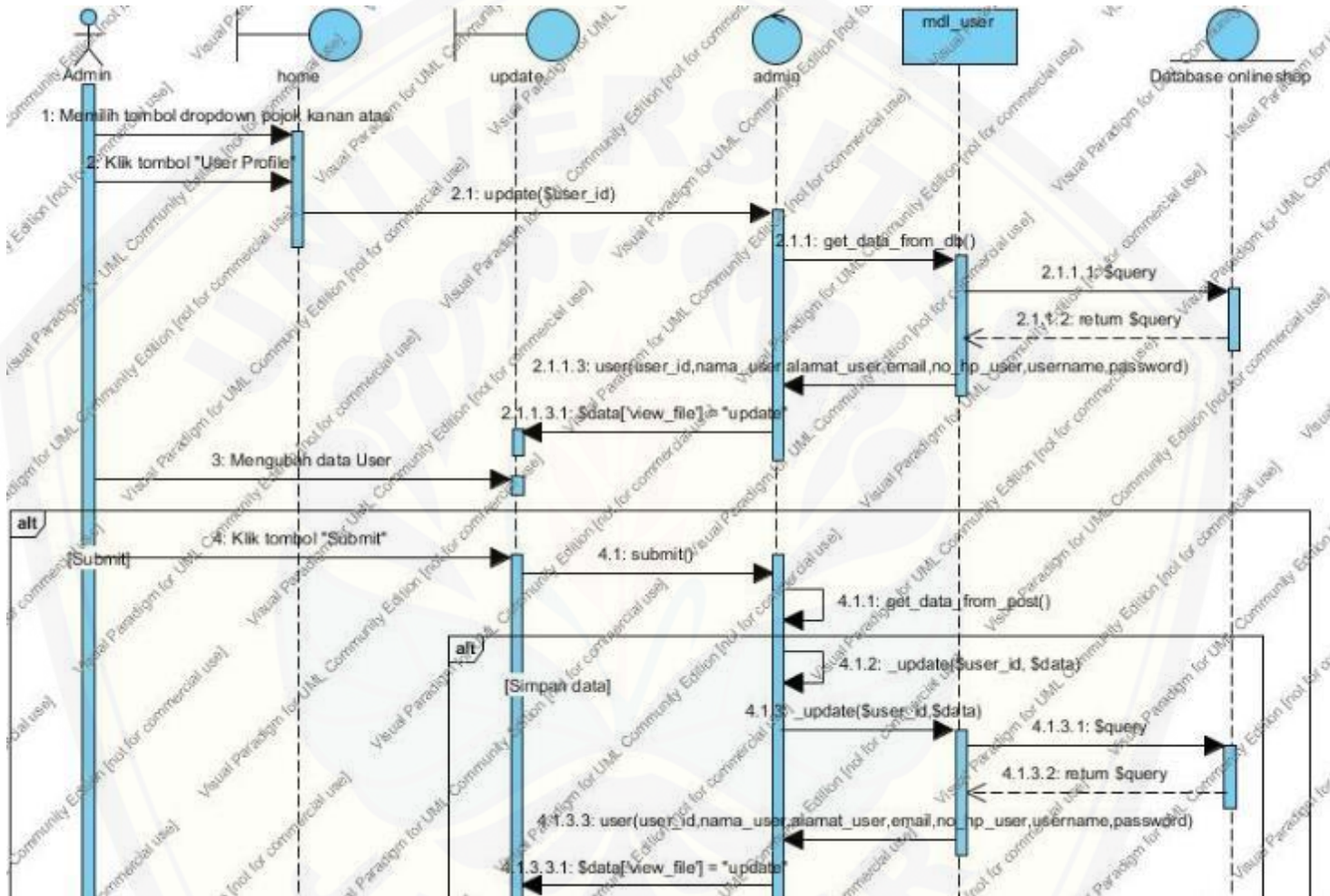


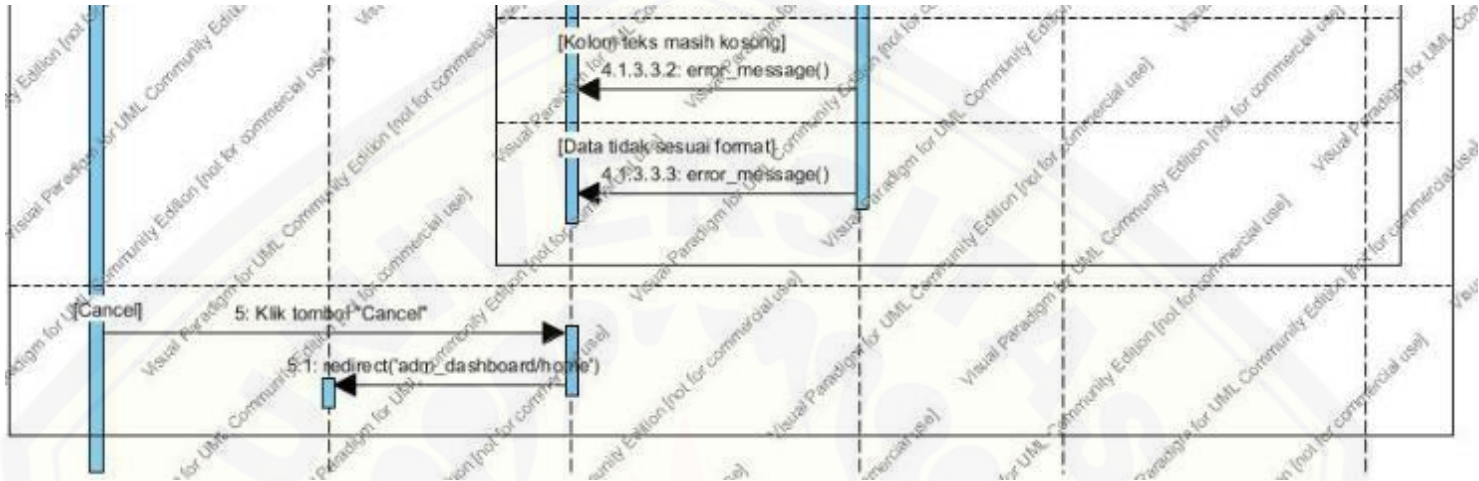




Gambar 25. Sequence Diagram Mengelola Data Admin

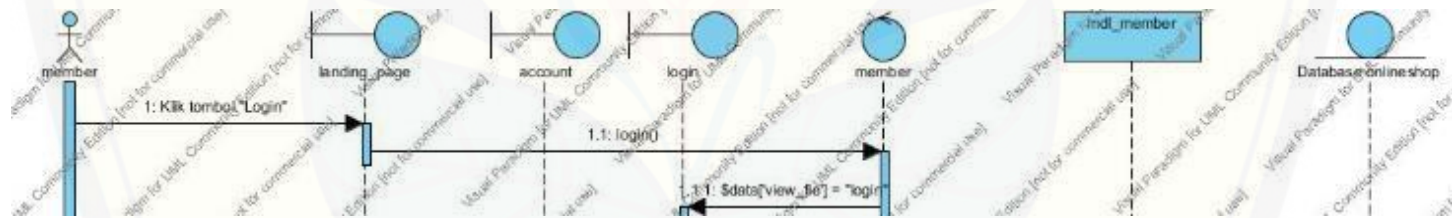
C.4. Sequence Diagram Mengubah Data Profil Admin

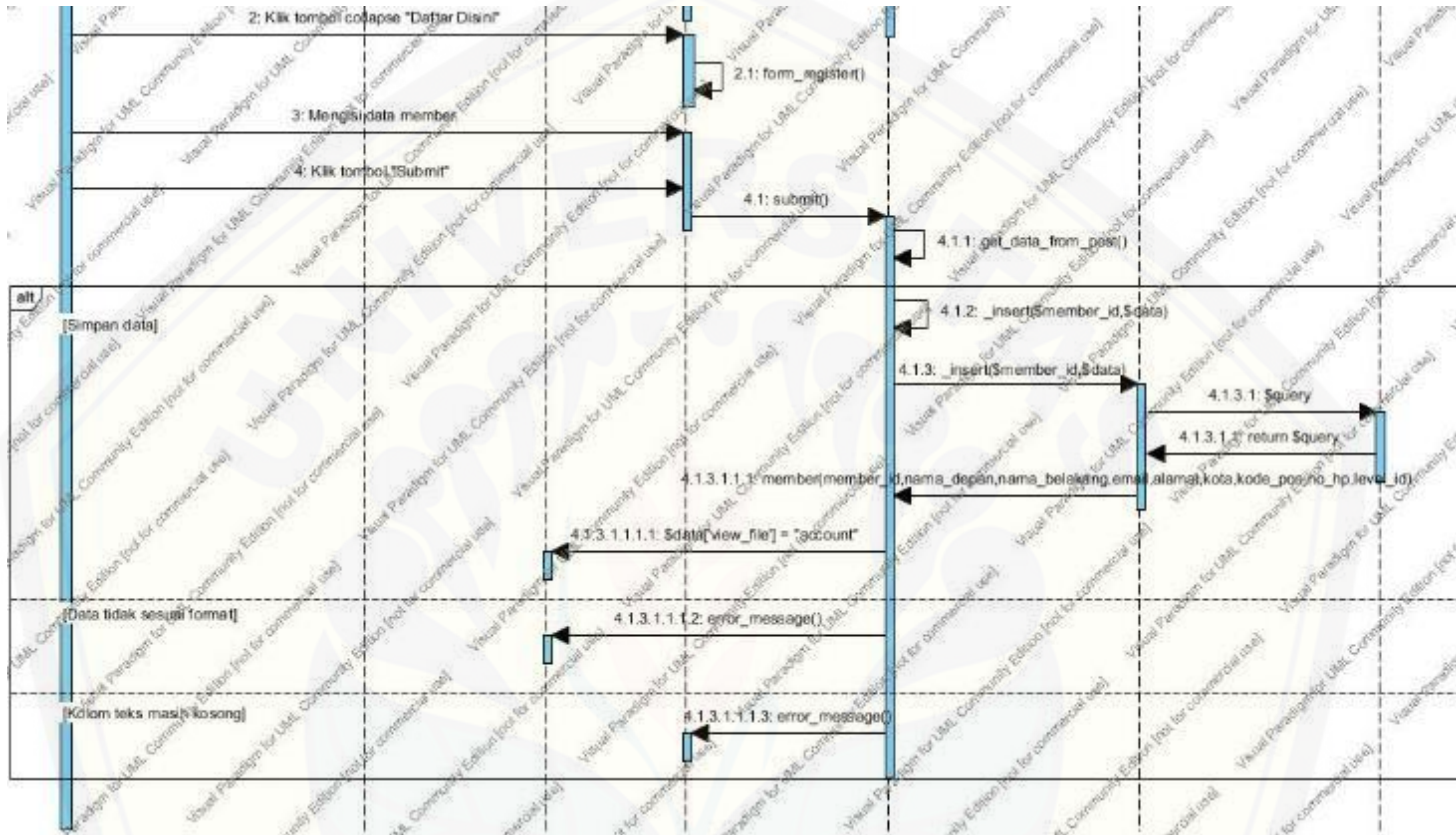




Gambar 26. Sequence Diagram Mengubah Data Profil Admin

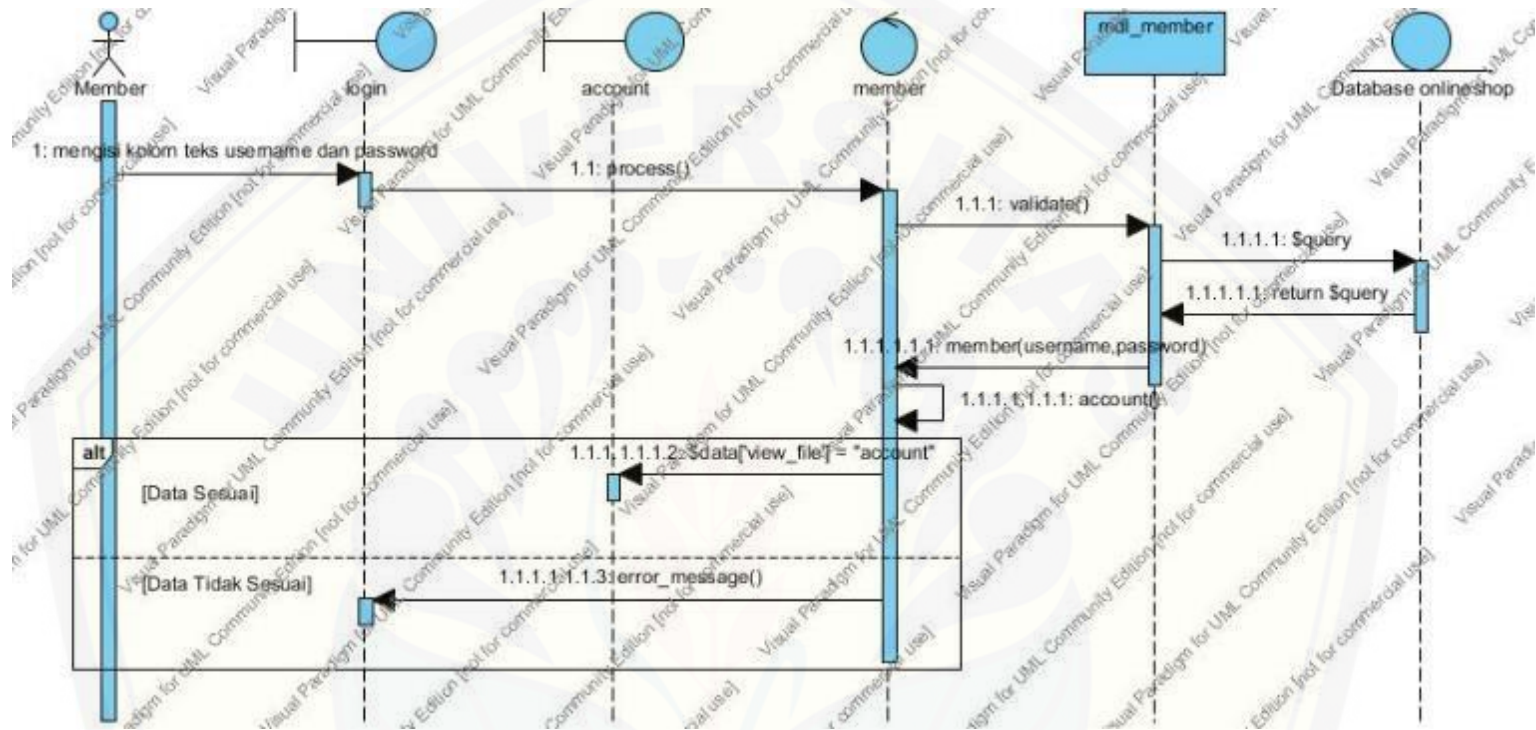
C.5. Sequence Diagram Register Member





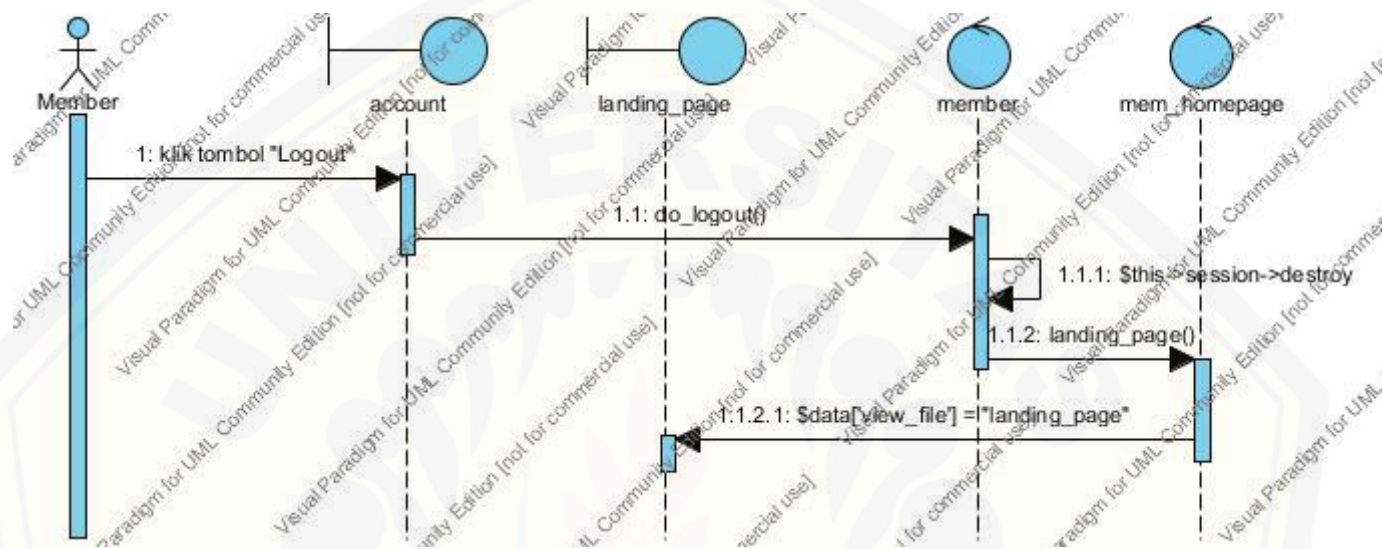
Gambar 27. Sequence Diagram Register Member

C.6. Sequence Diagram Login Member



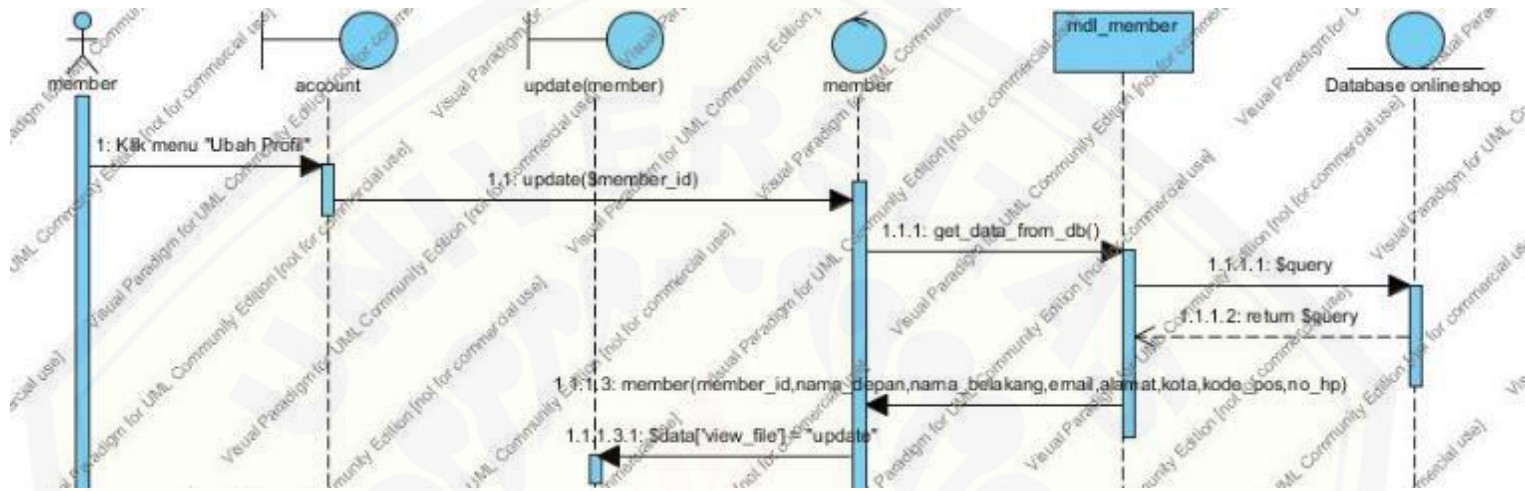
Gambar 28. Sequence Diagram Login Member

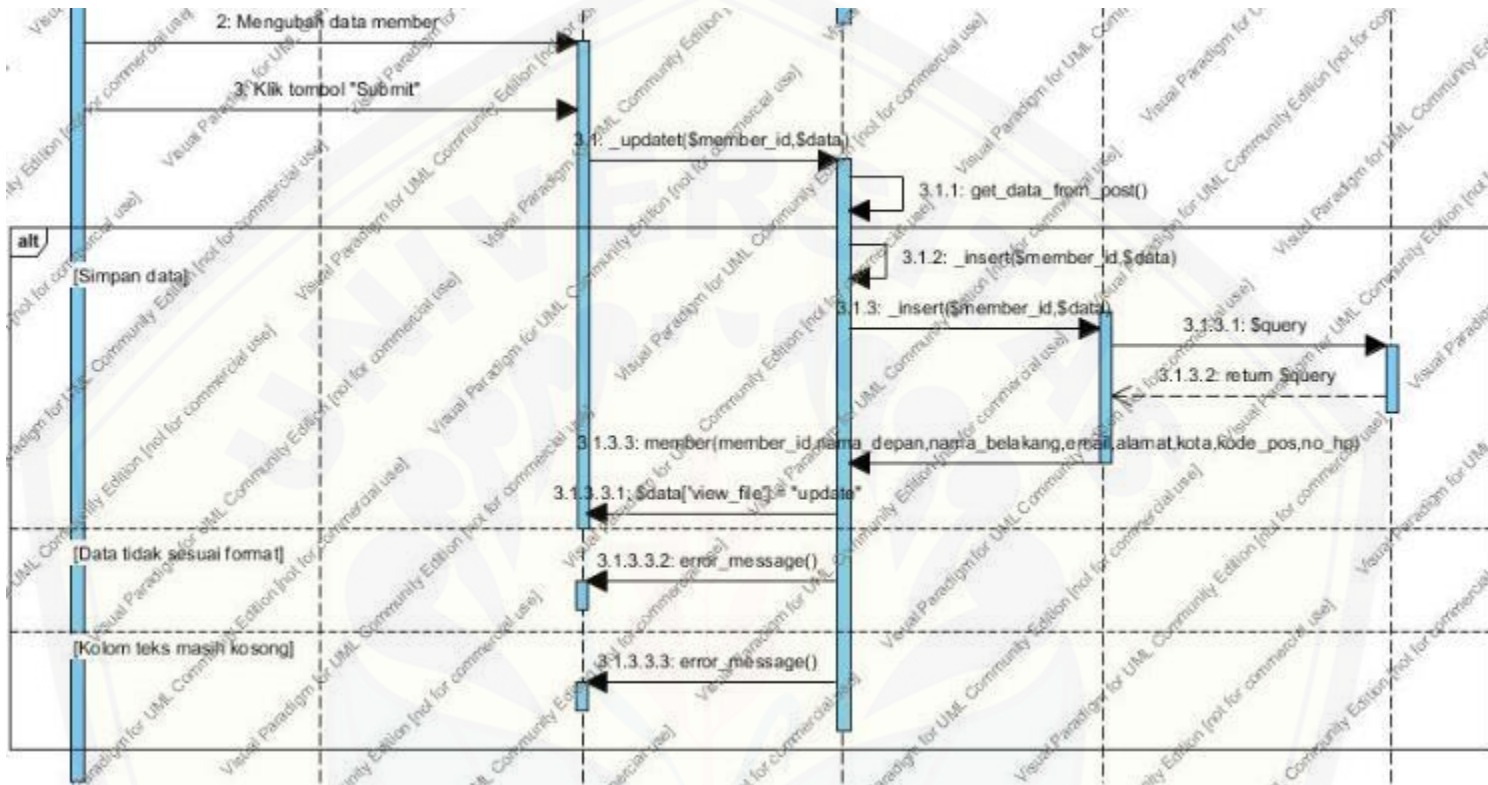
C.7. Sequence Diagram Logout Member



Gambar 29. Sequence Diagram Logout Member

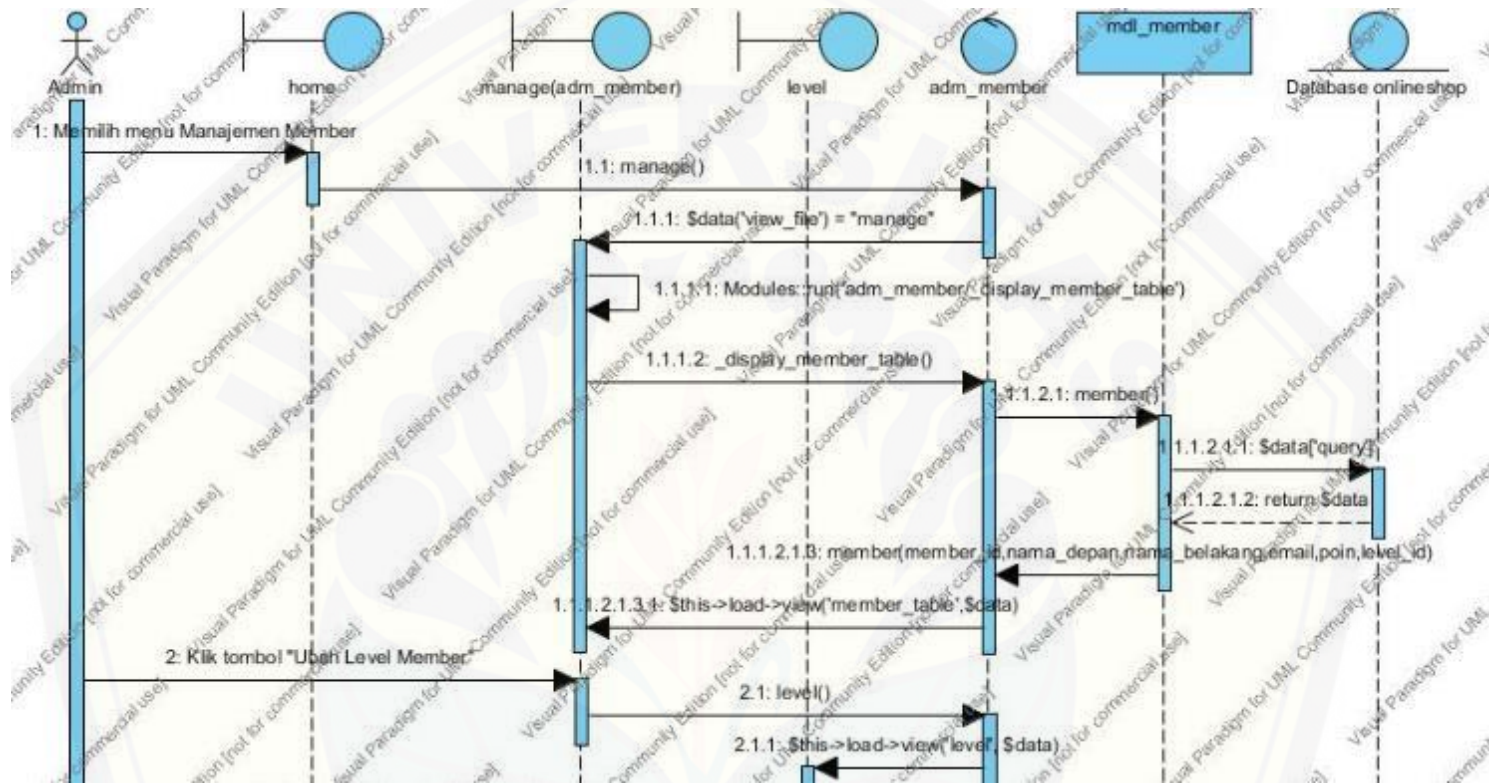
C.8. Sequence Diagram Mengubah Data Member

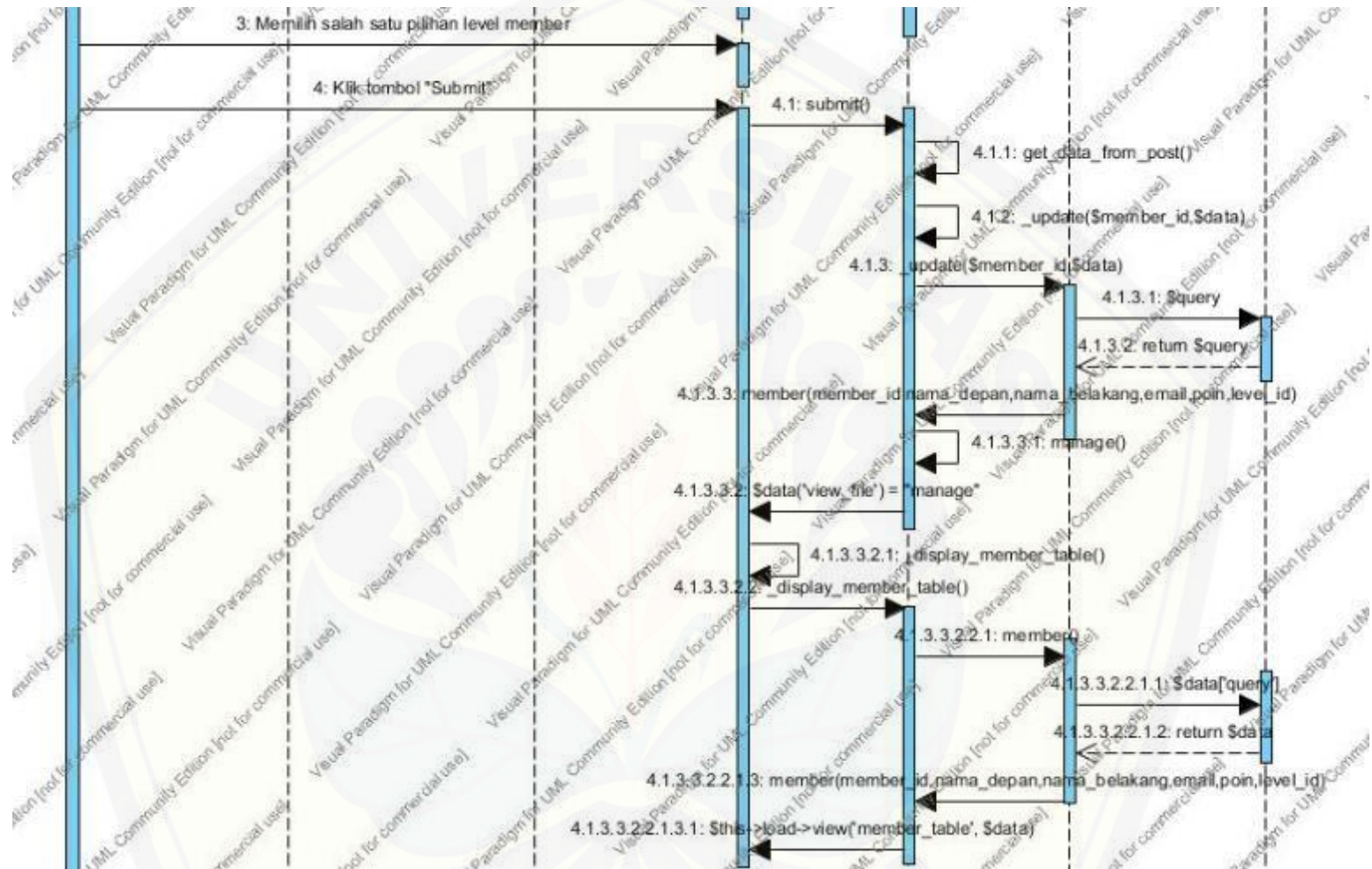




Gambar 30. Sequence Diagram Mengubah Data Member

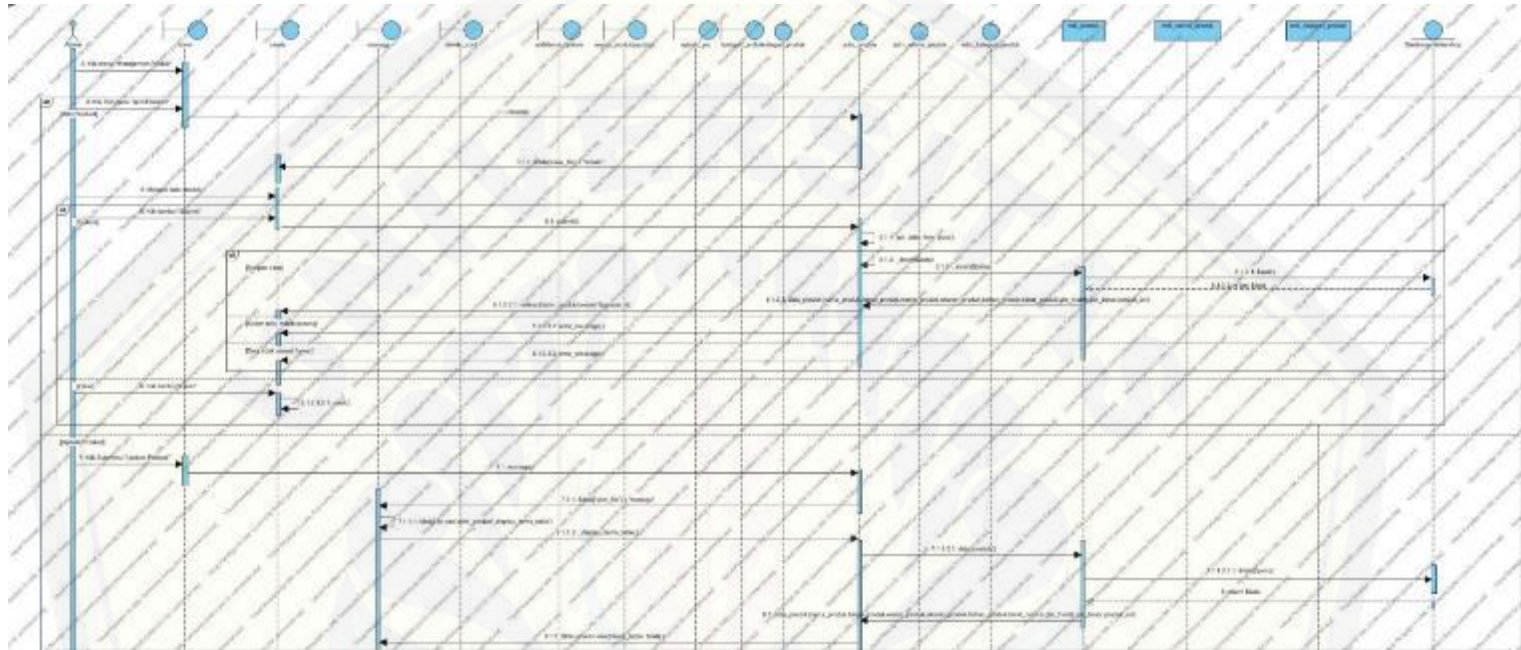
C.9. Sequence Diagram Mengubah Level Member

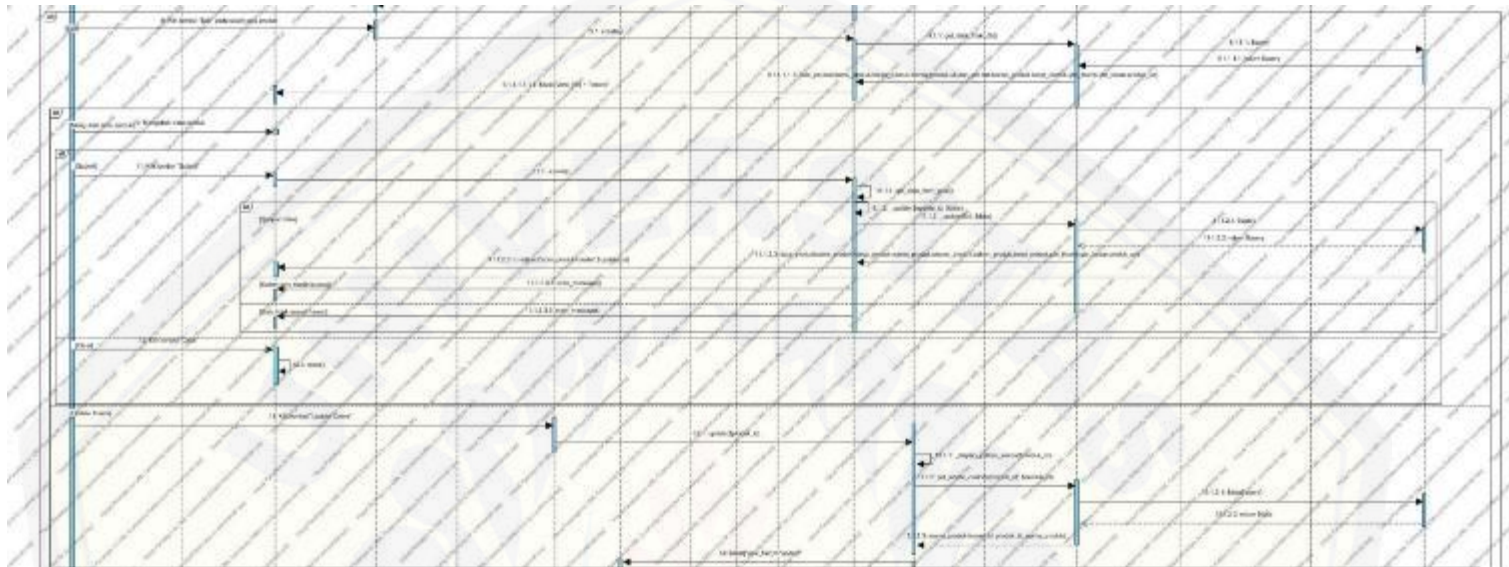


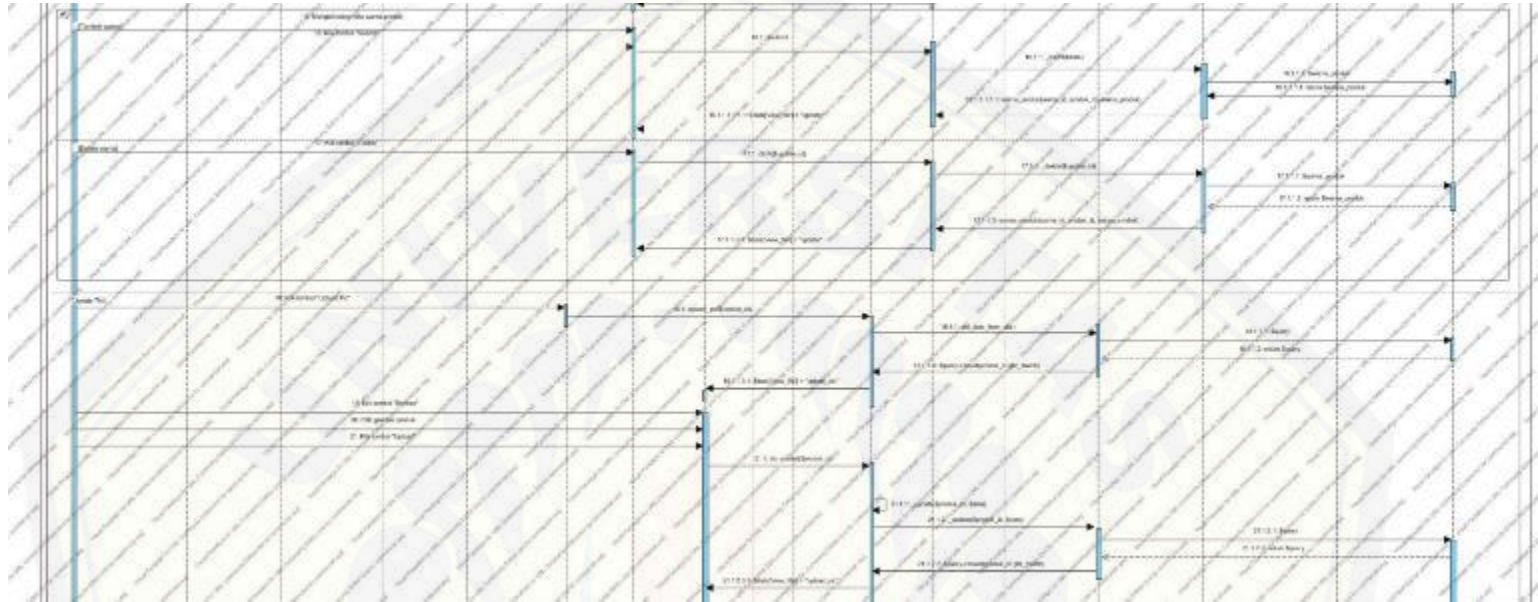


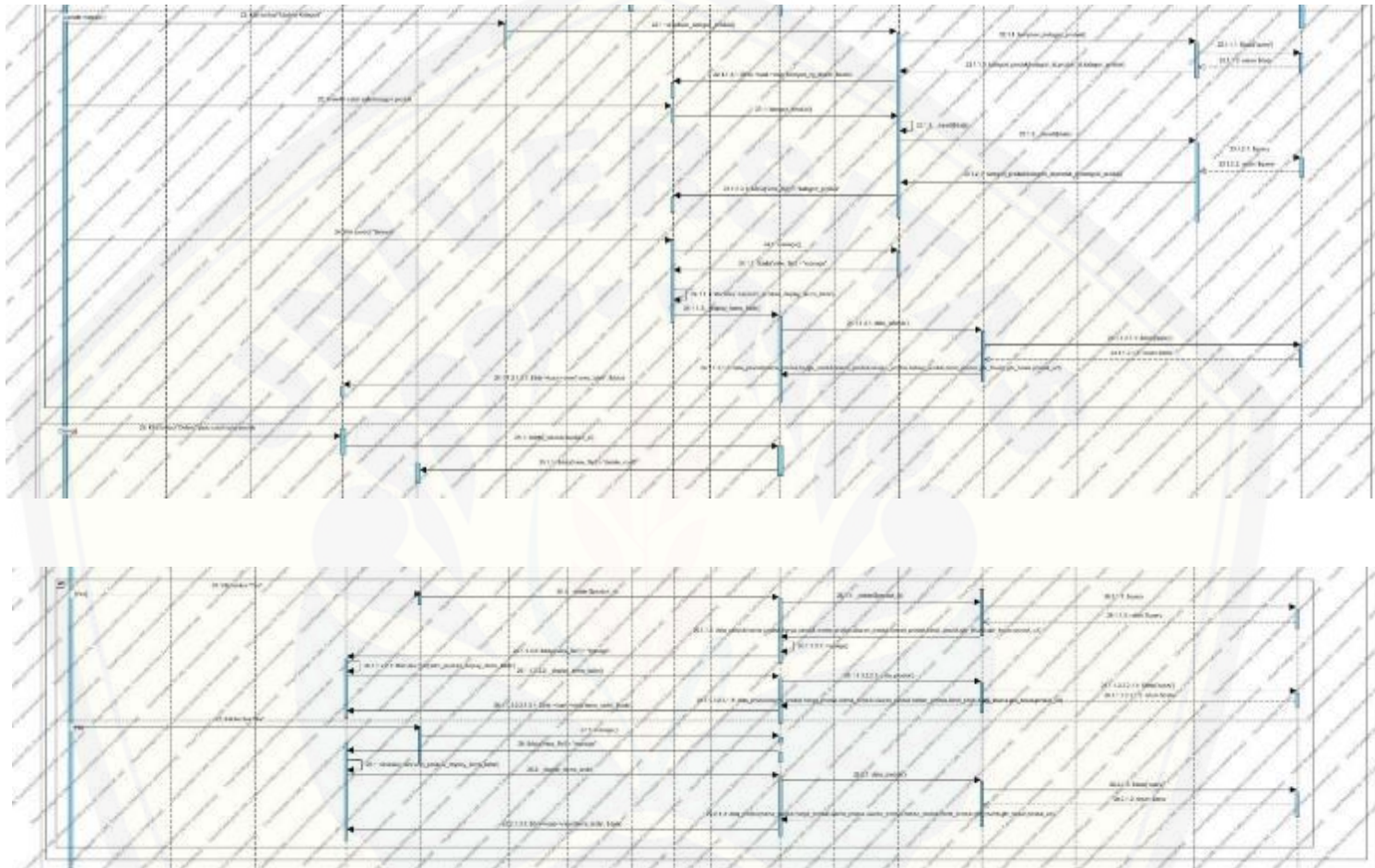
Gambar 31. Sequence Diagram Mengubah Level Member

C.10. Sequence Diagram Mengelola Data Produk



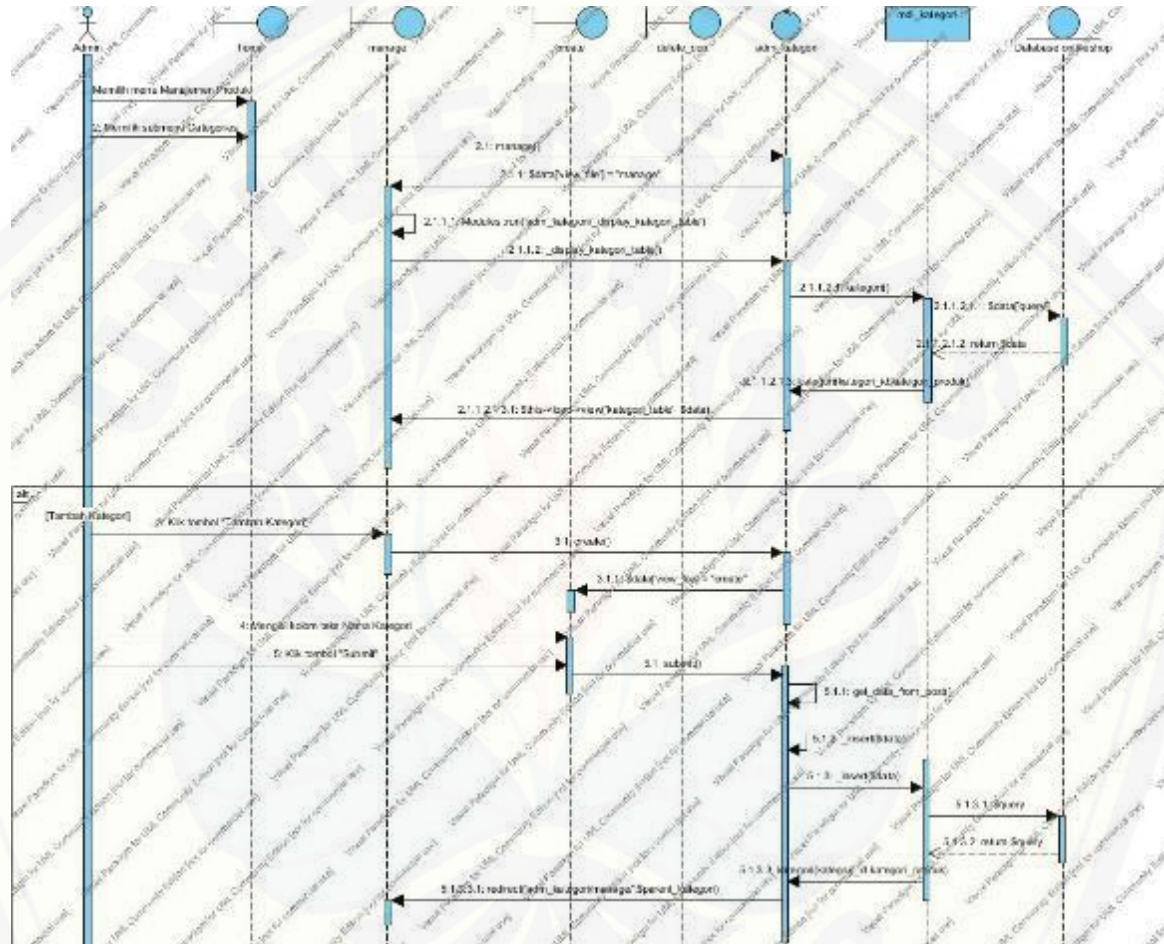


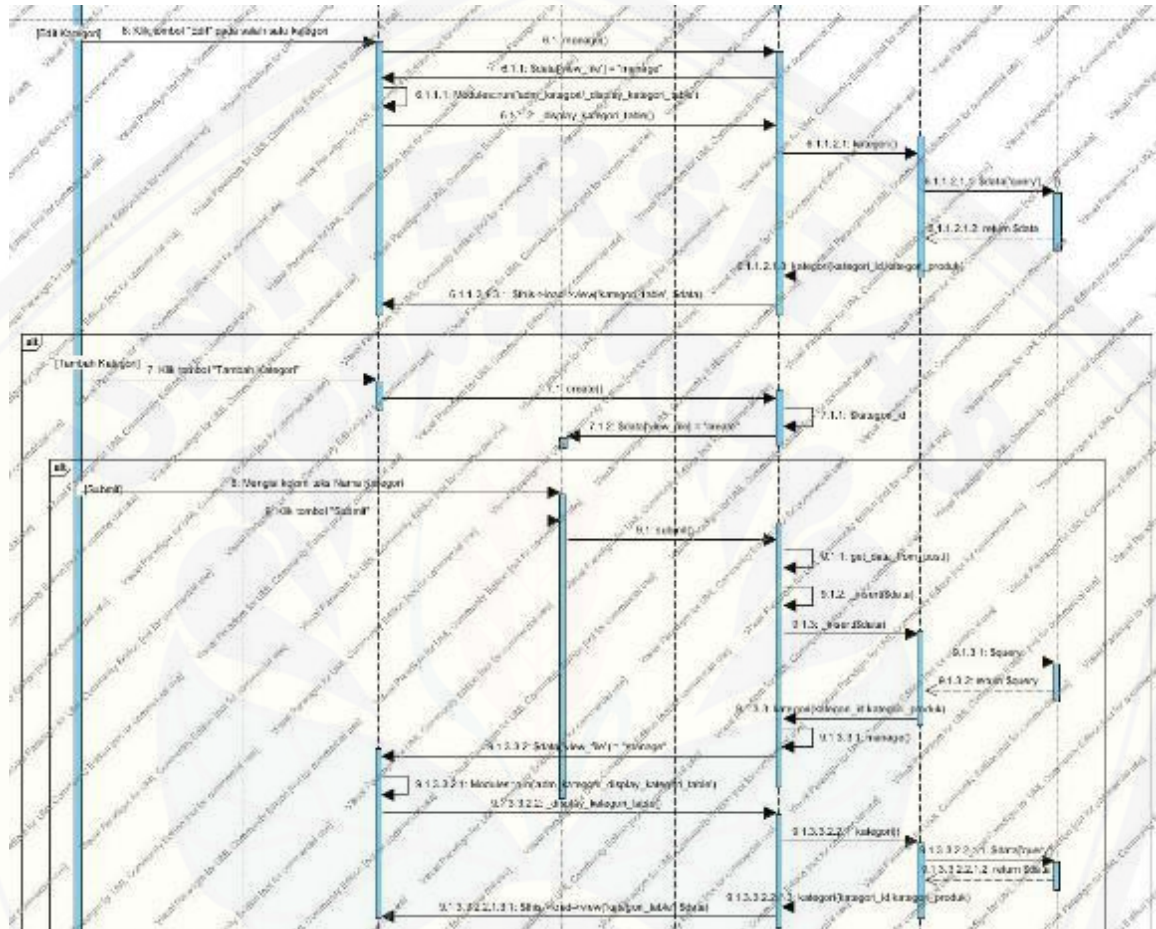




Gambar 32. Sequence Diagram Mengelola Data Produk

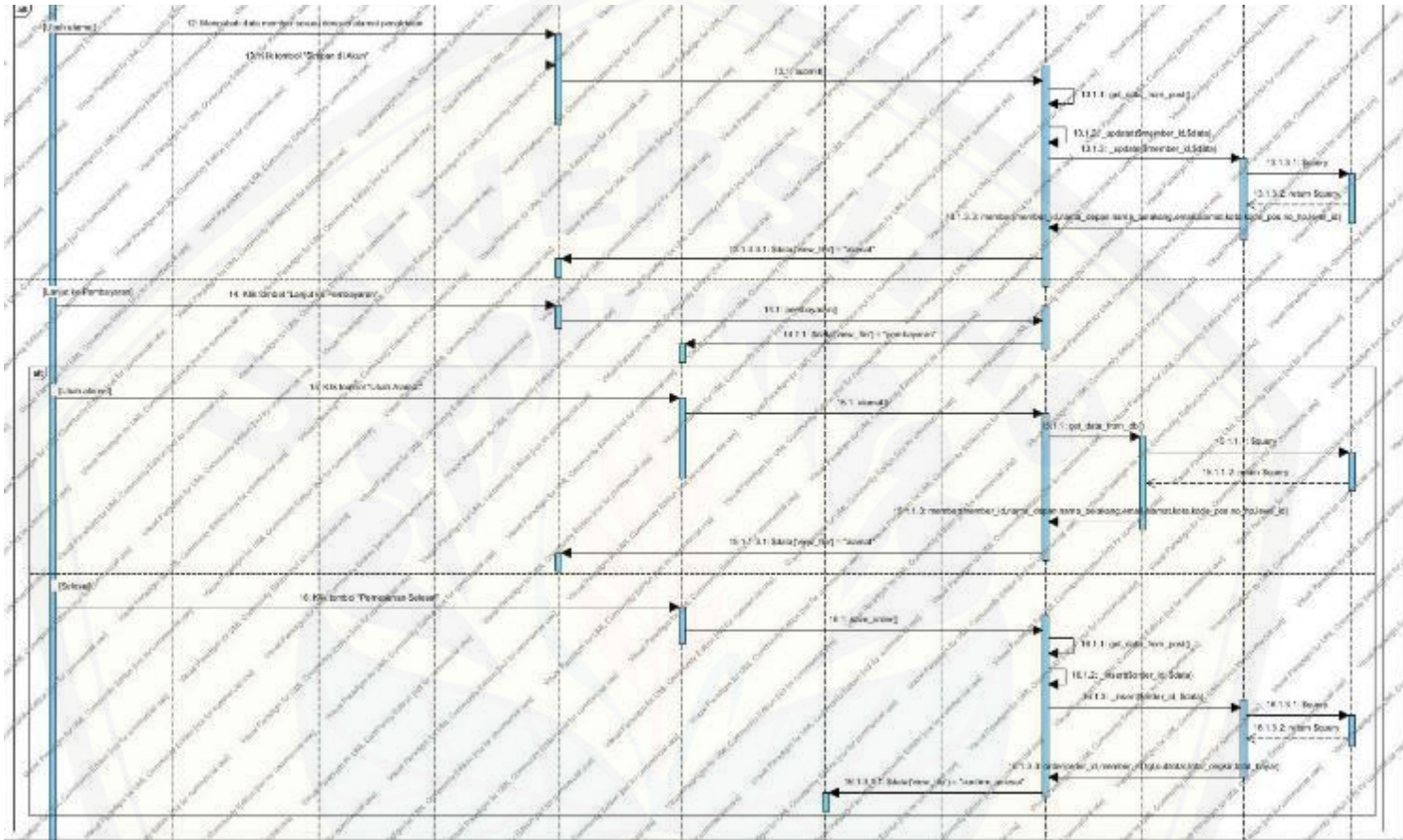
C.11. Sequence Diagram Mengelola Kategori Produk





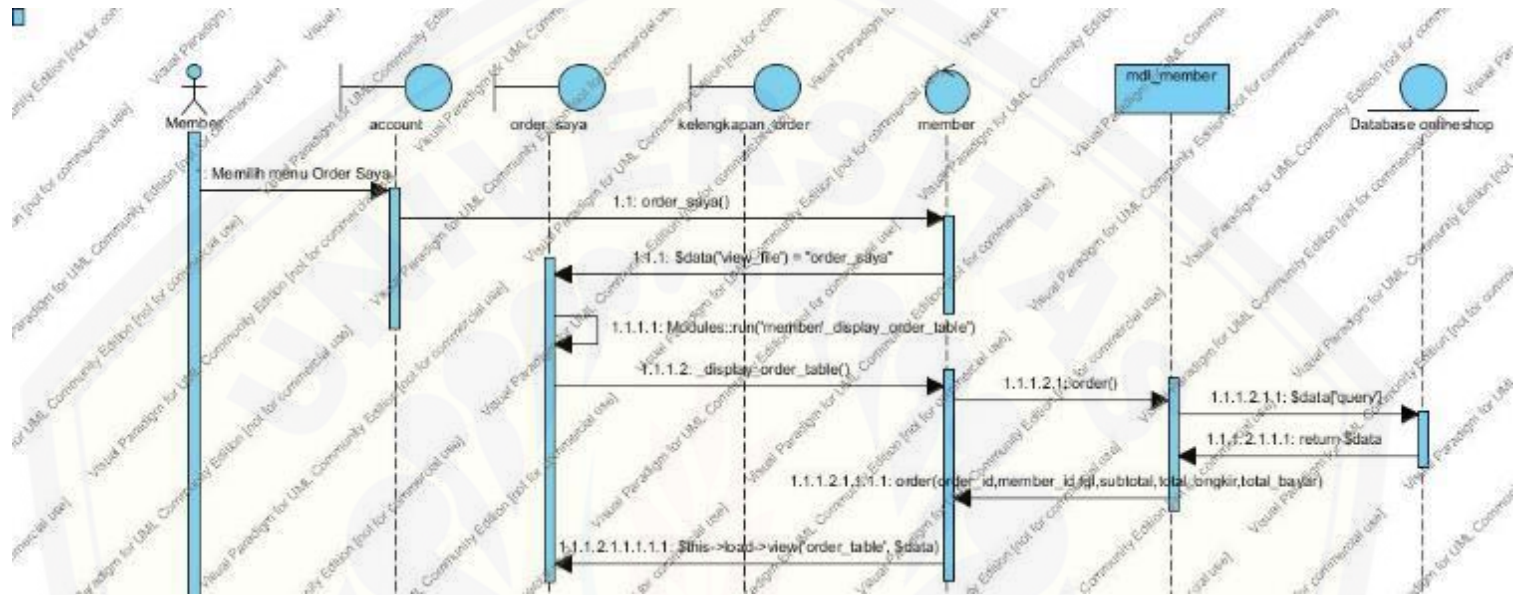


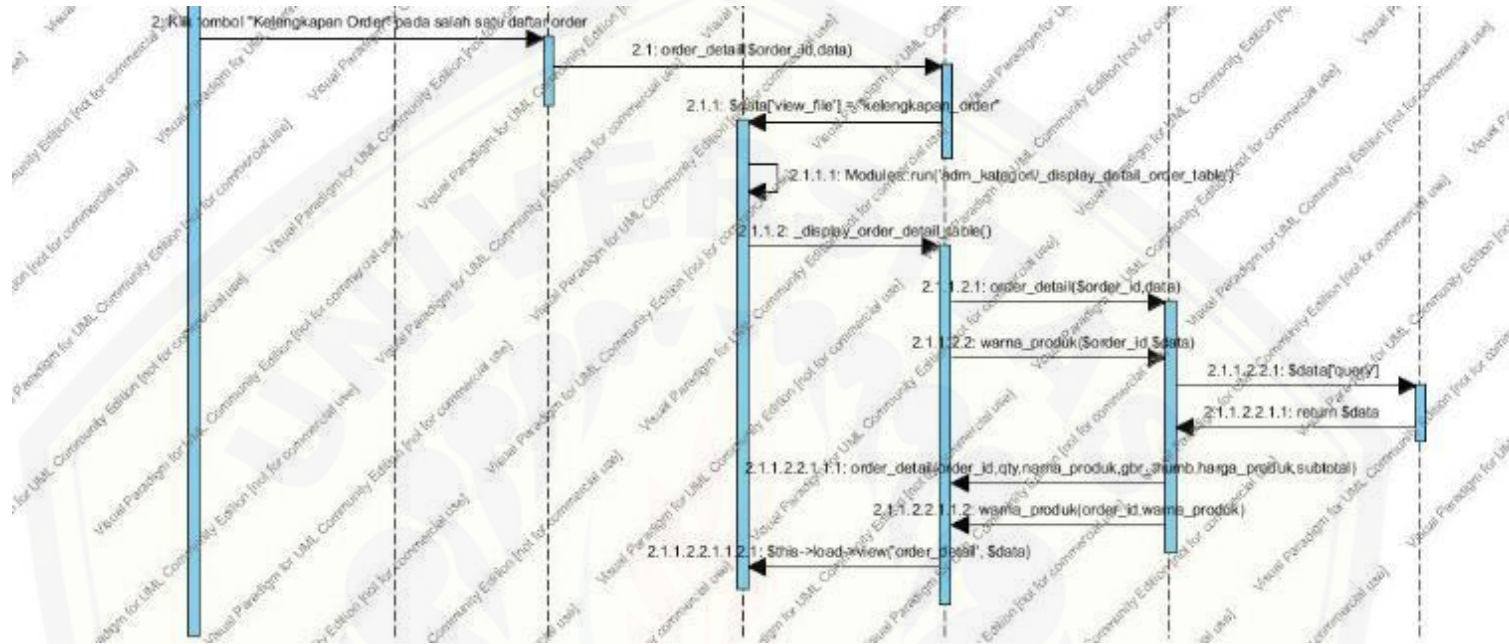
Gambar 33. Sequence Diagram Mengelola Kategori Produk



Gambar 34. Sequence Diagram Mengelola Data Order

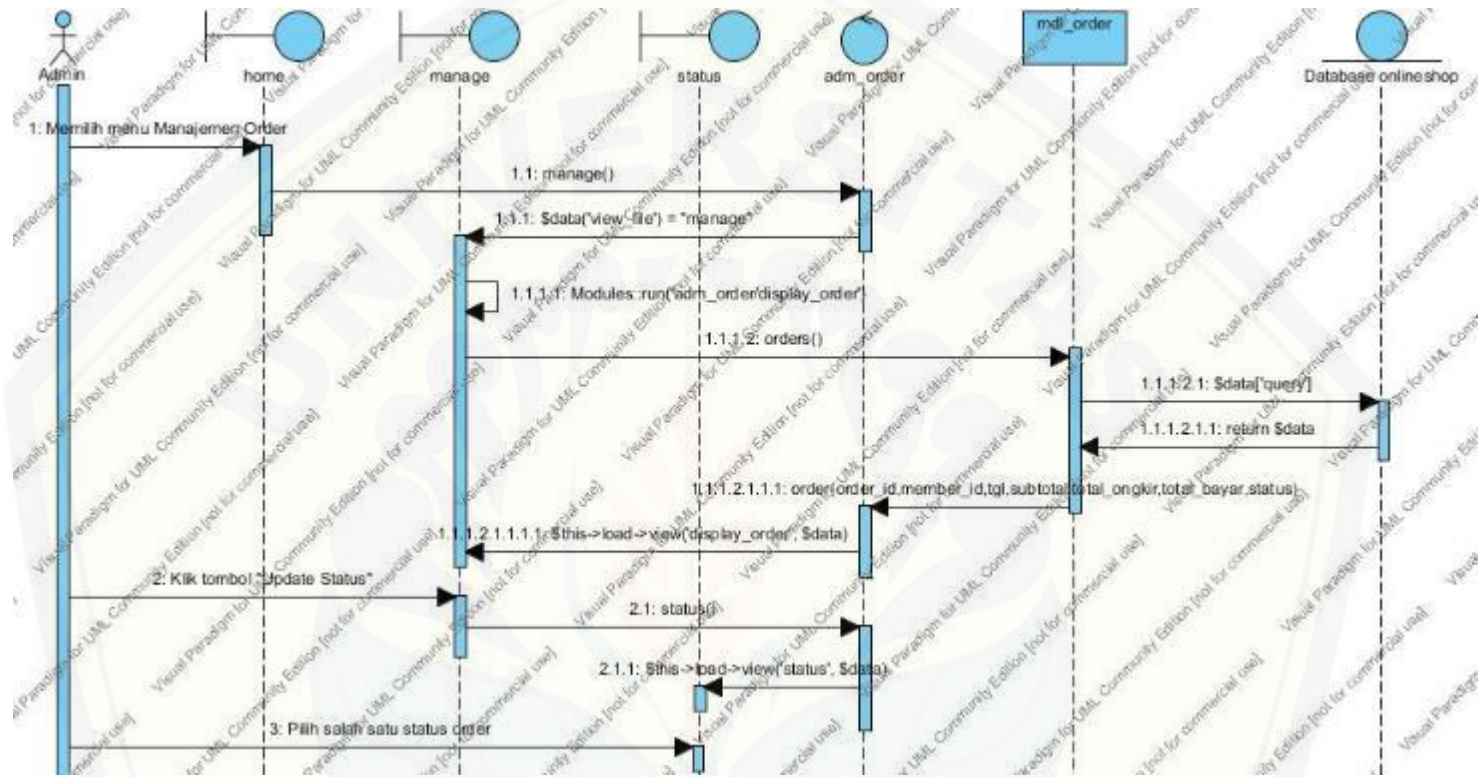
C.13. Sequence Diagram Melihat Daftar Order

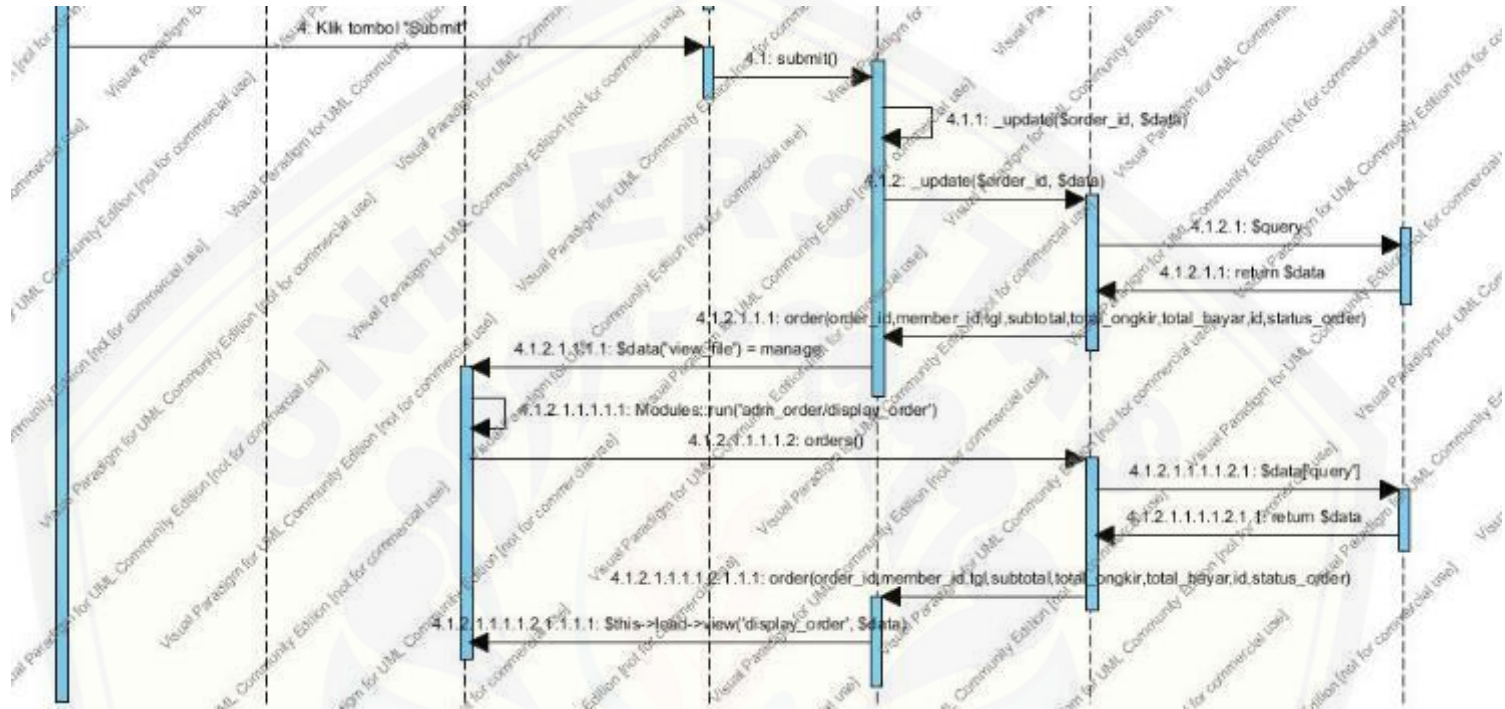




Gambar 35. Sequence Diagram Melihat Daftar Order

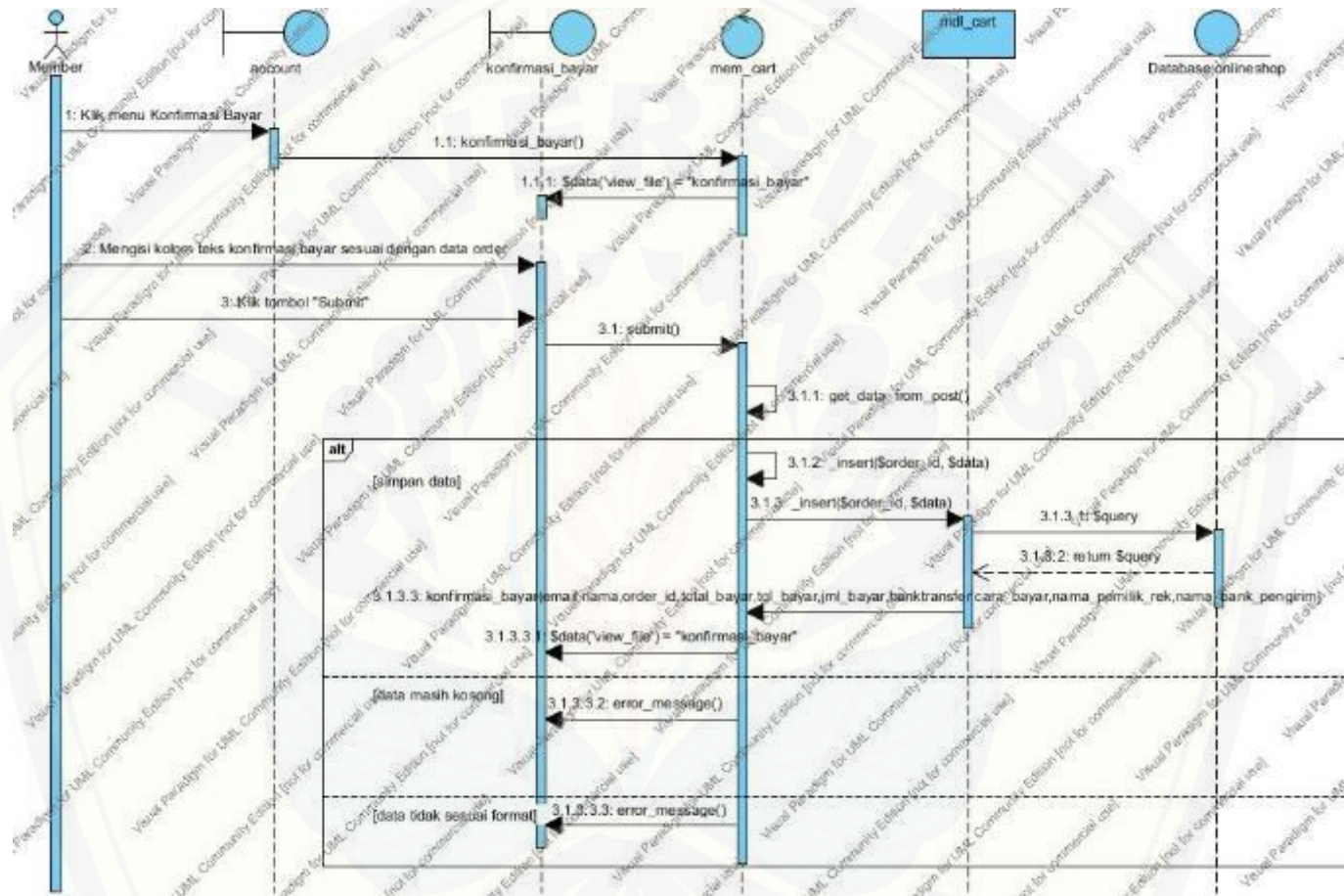
C.14. Sequence Diagram Mengubah Status Order





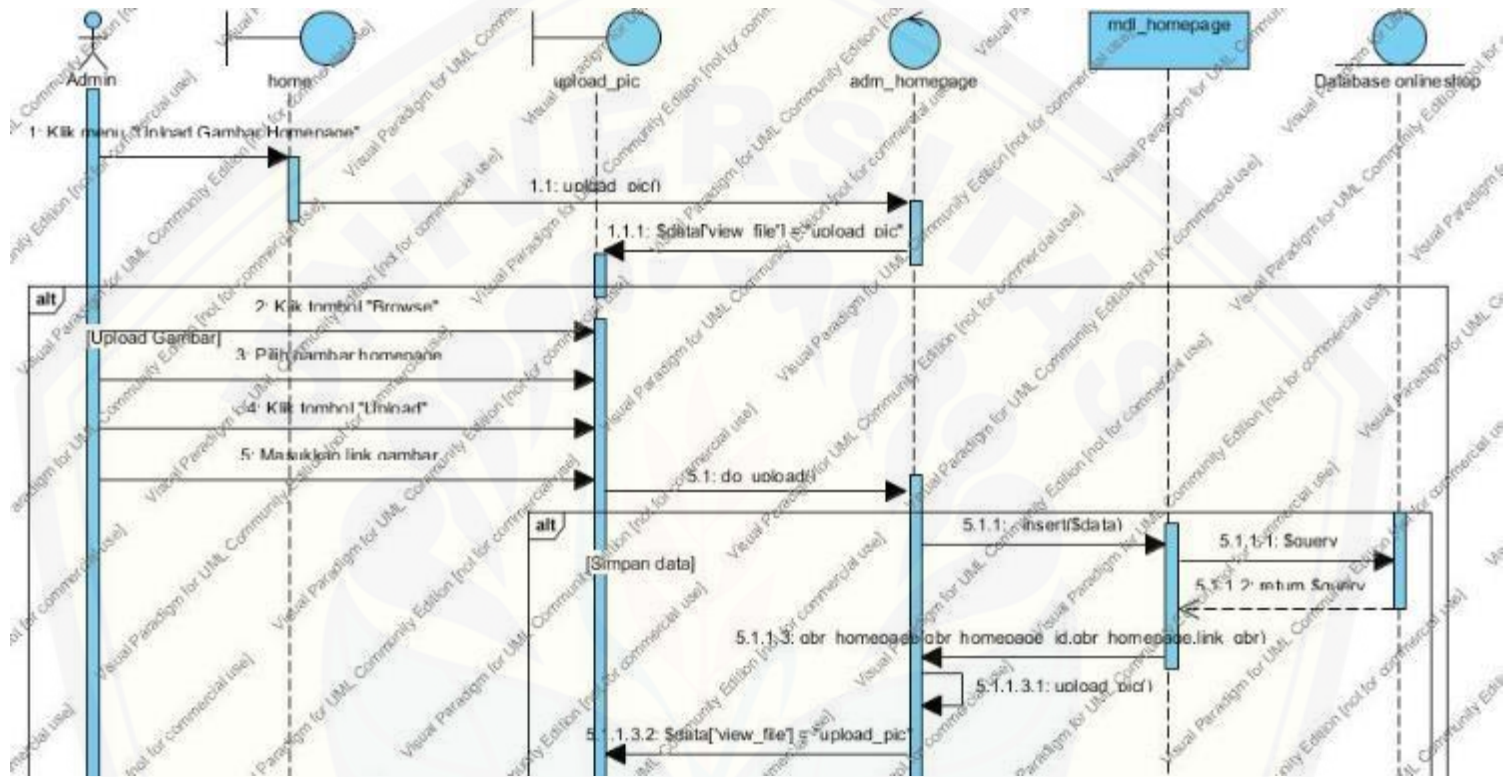
Gambar 36. Sequence Diagram Mengubah Status Order

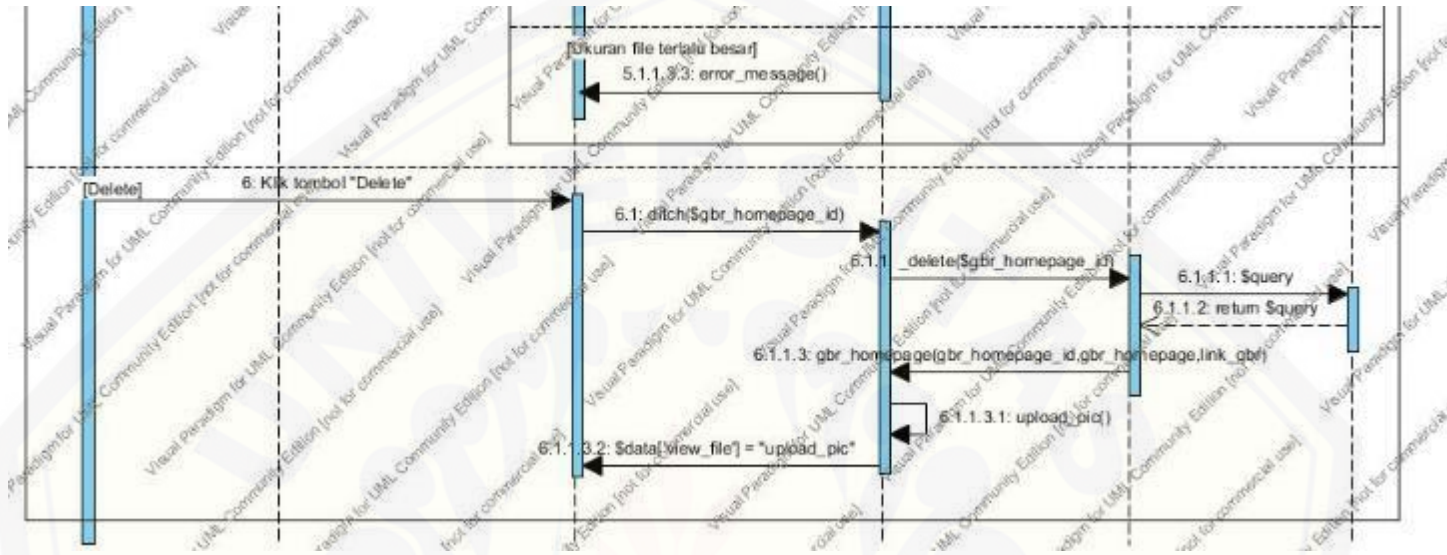
C.15. Sequence Diagram Menambah Data Konfirmasi Bayar



Gambar 37. Sequence Diagram Menambah Data Konfirmasi Bayar

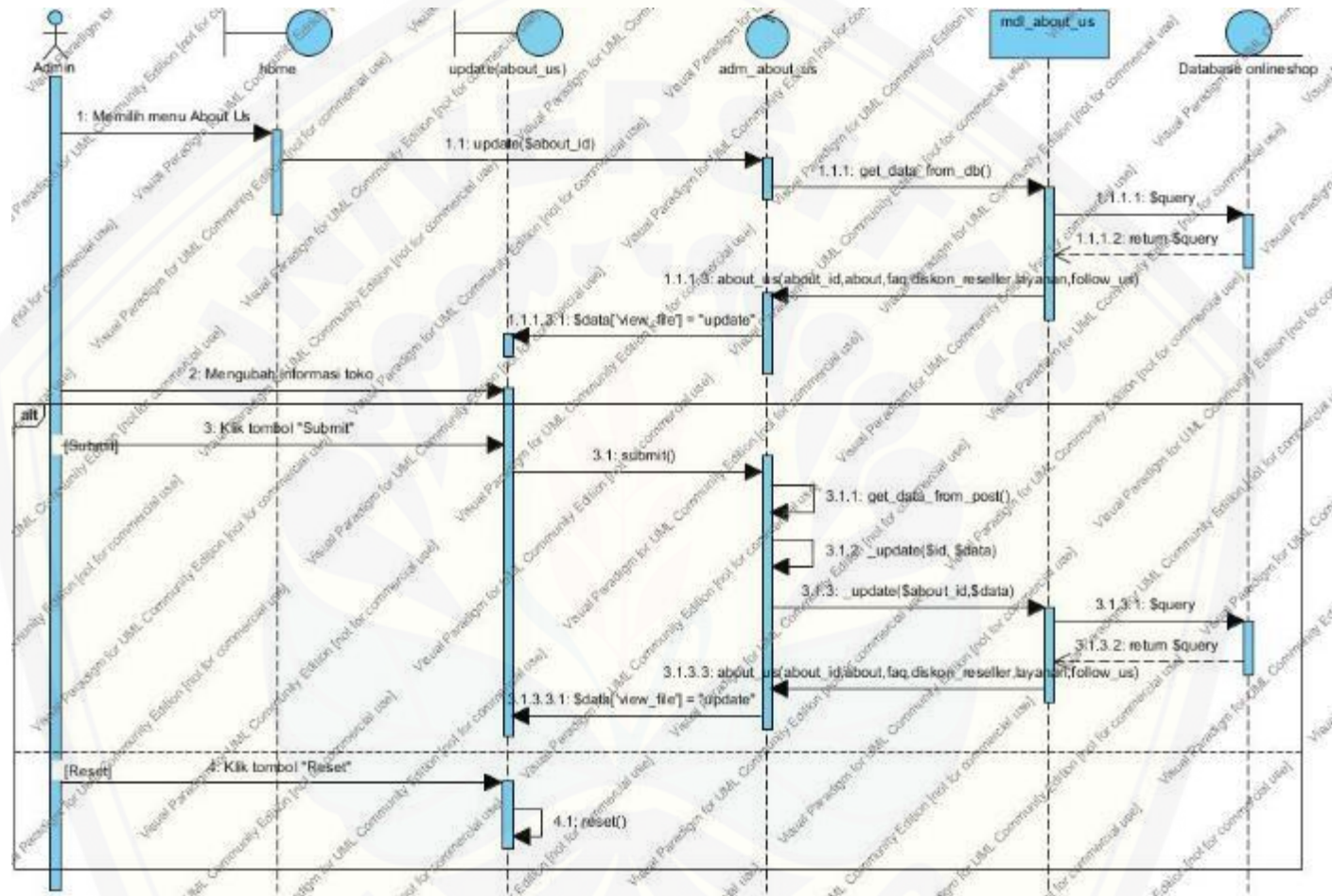
C.16. Sequence Diagram Mengelola Gambar Landing Page Member





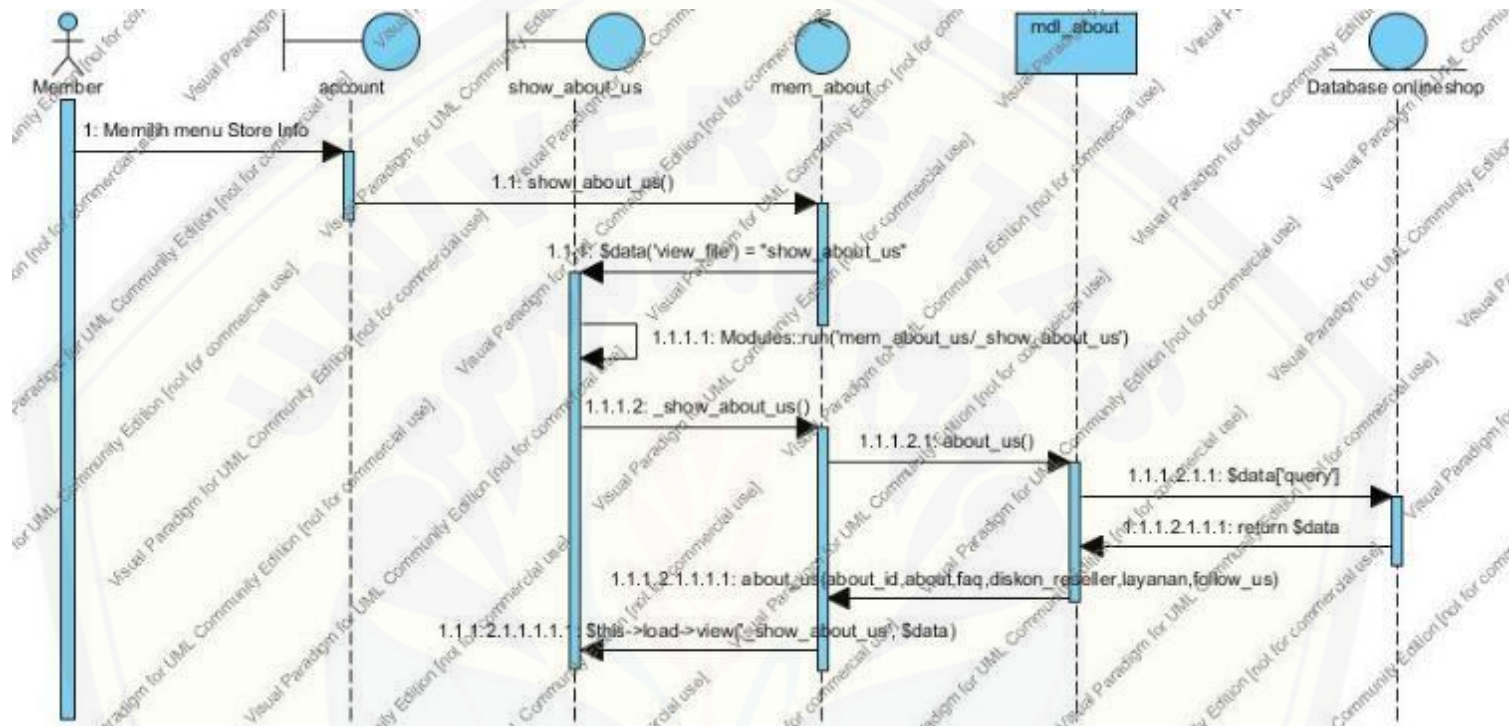
Gambar 38. Sequence Diagram Mengelola Gambar Landing Page Member

C.17. Sequence Diagram Mengubah Informasi Toko



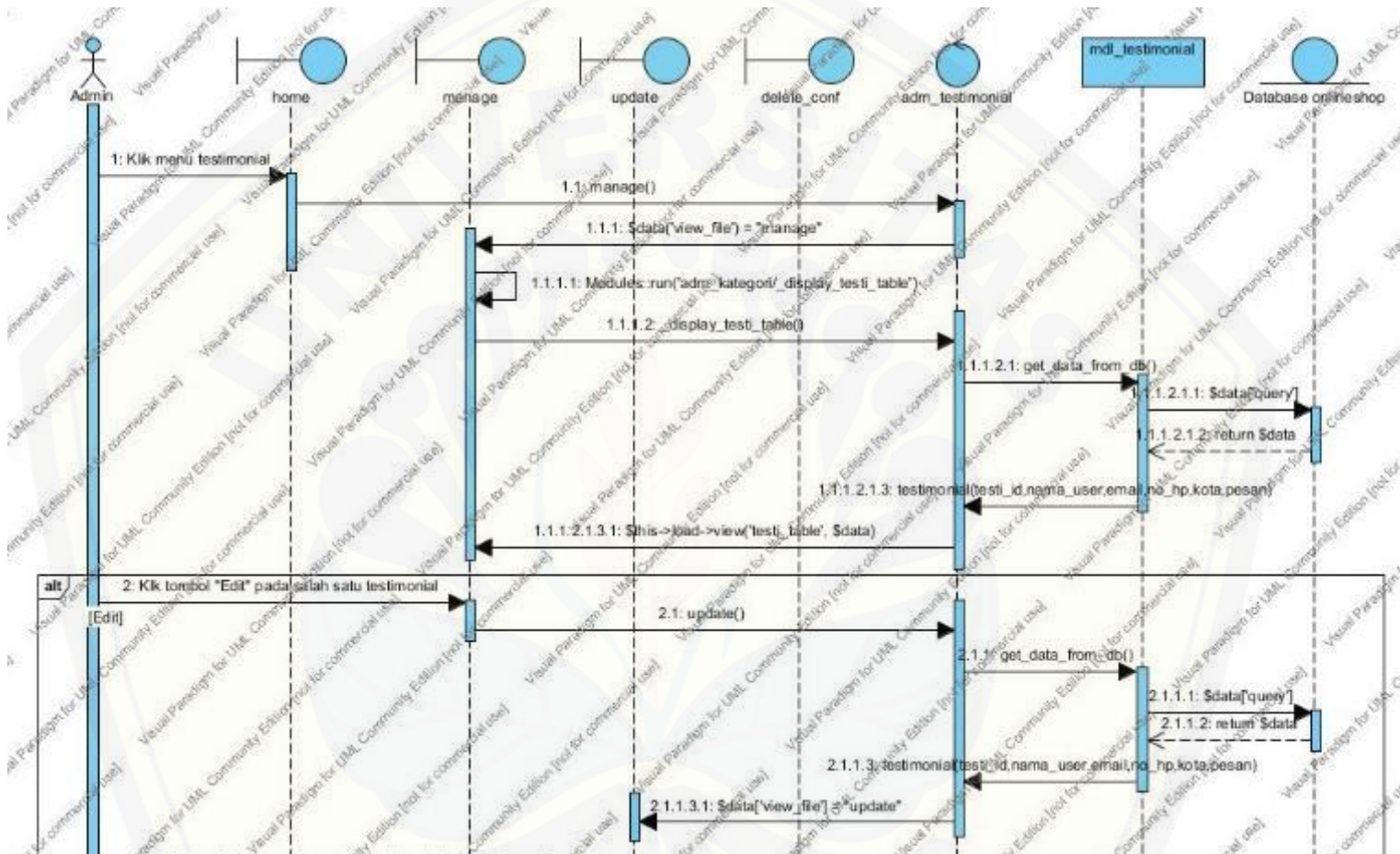
Gambar 39. Sequence Diagram Mengubah Informasi Toko

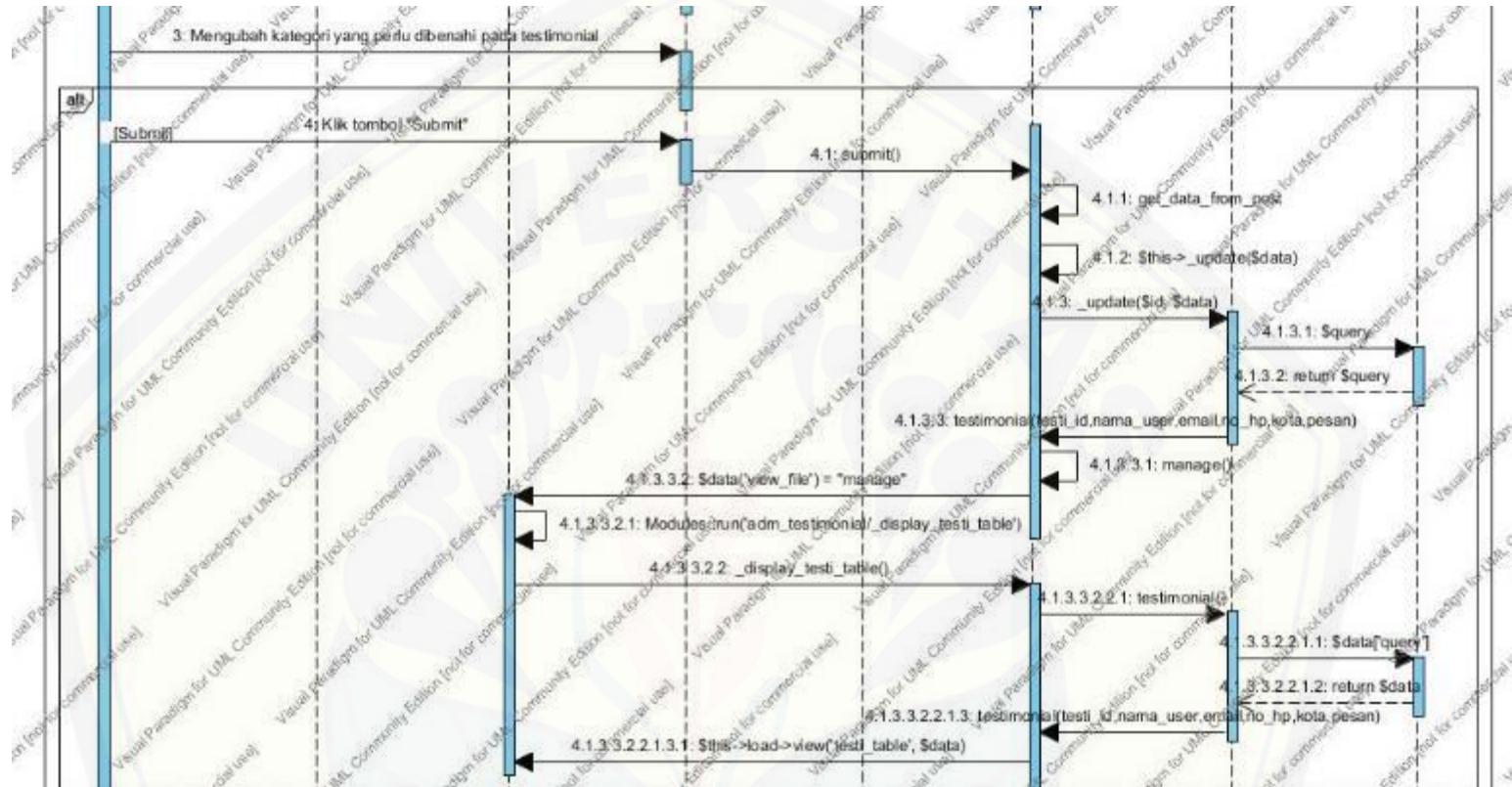
C.18. Sequence Diagram Melihat Informasi Toko

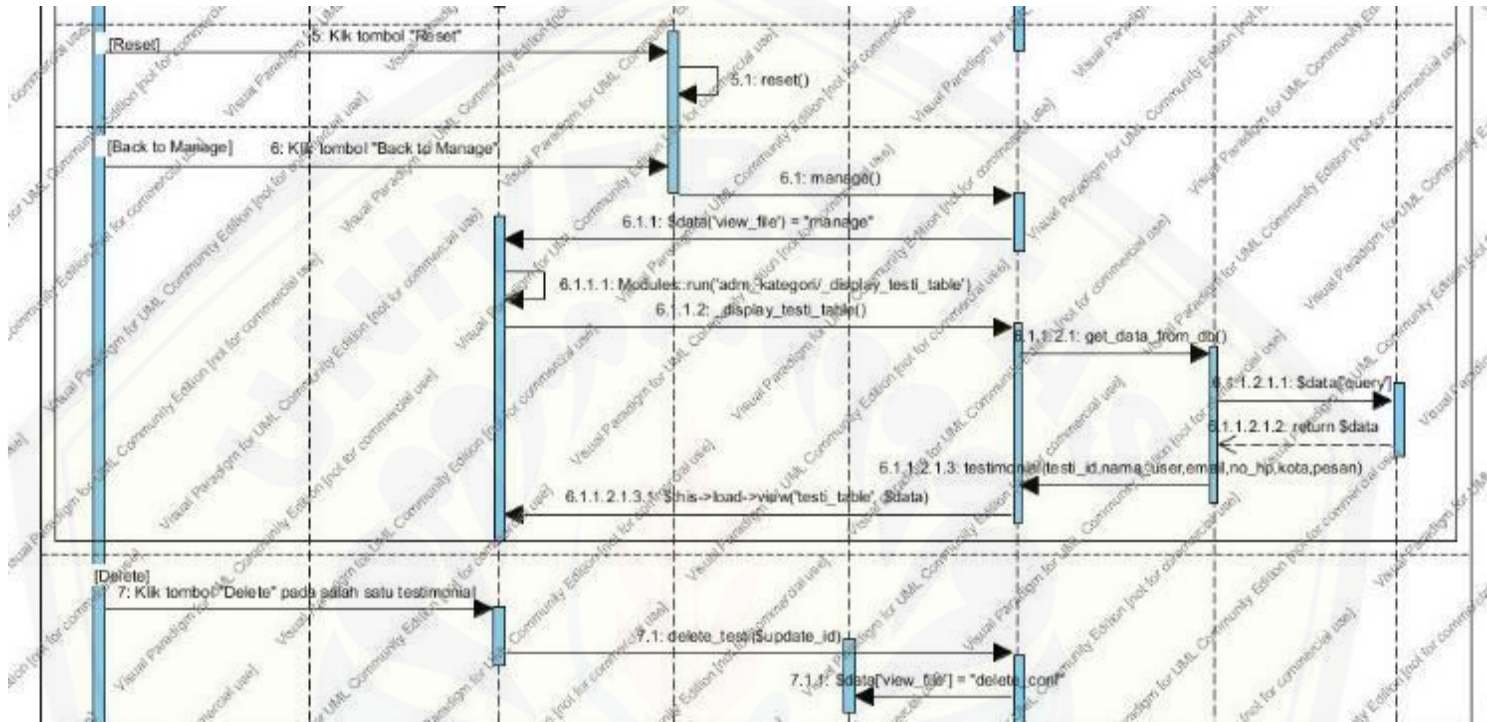


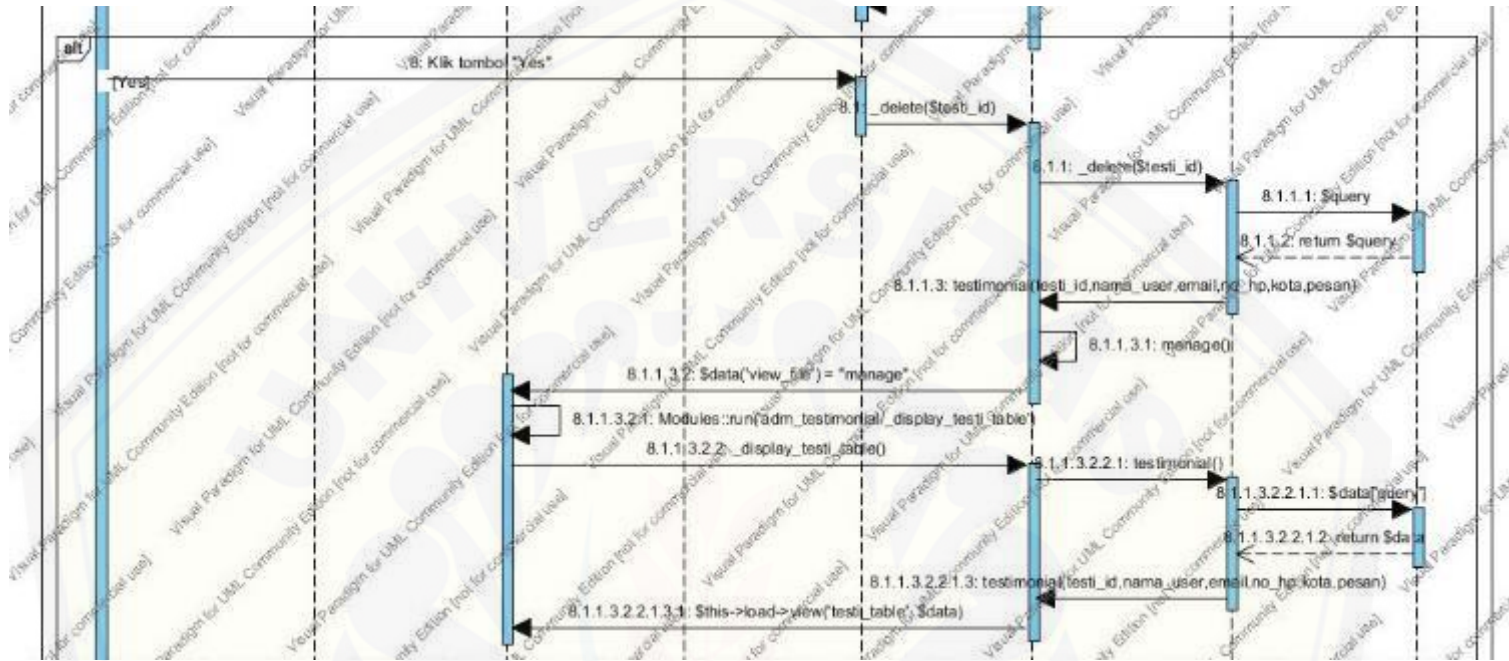
Gambar 40. Sequence Diagram Melihat Informasi Toko

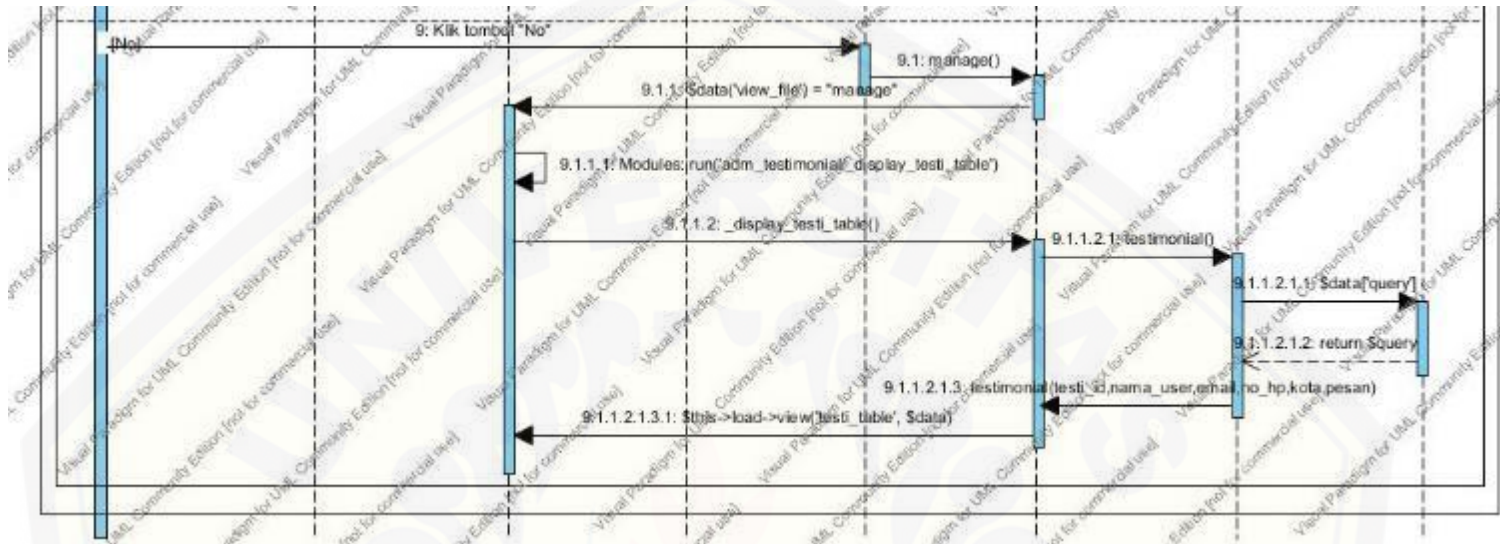
C.19. Sequence Diagram Mengubah dan Menghapus Testimonial





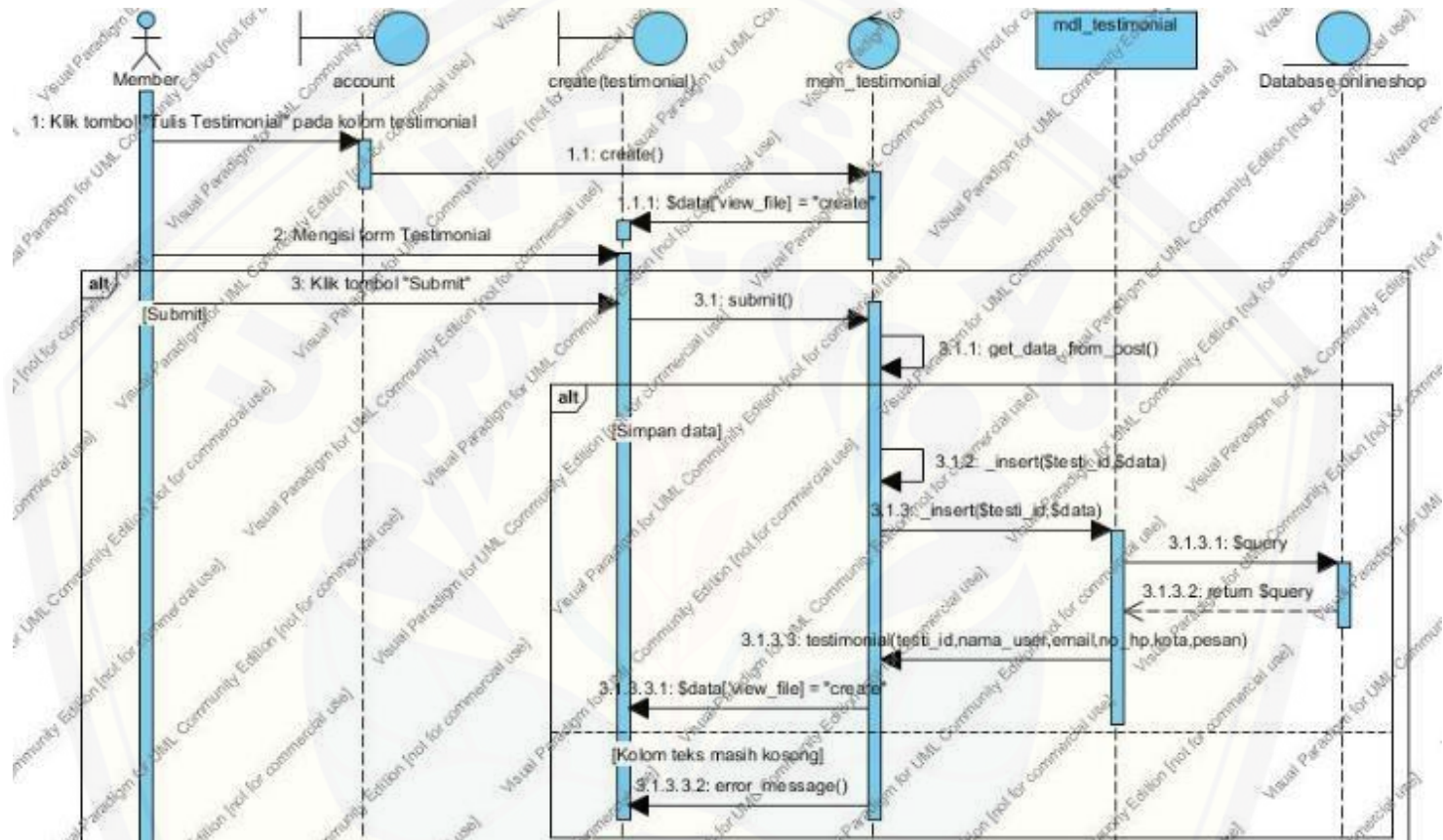


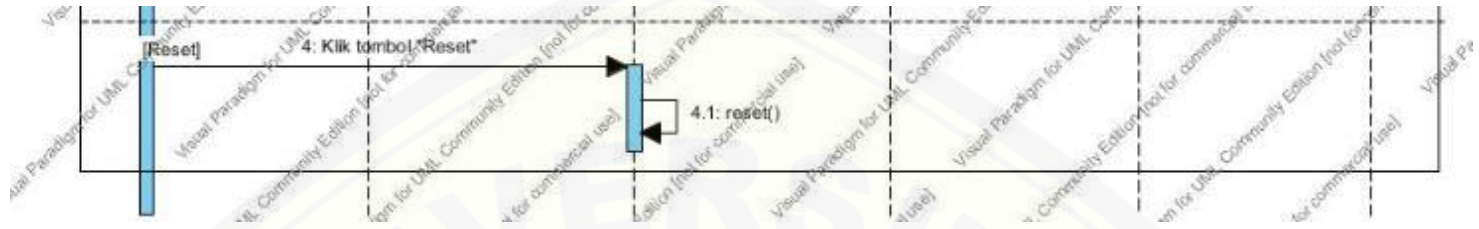




Gambar 41. Sequence Diagram Mengubah dan Menghapus Testimonial

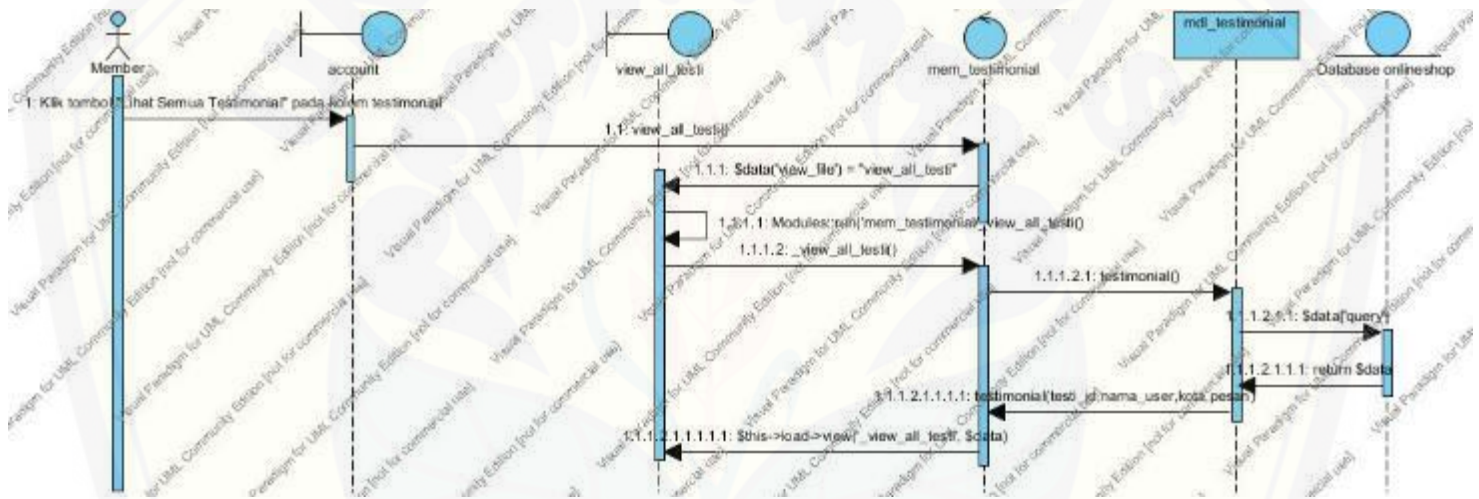
C.20. Sequence Diagram Menambah Testimonial





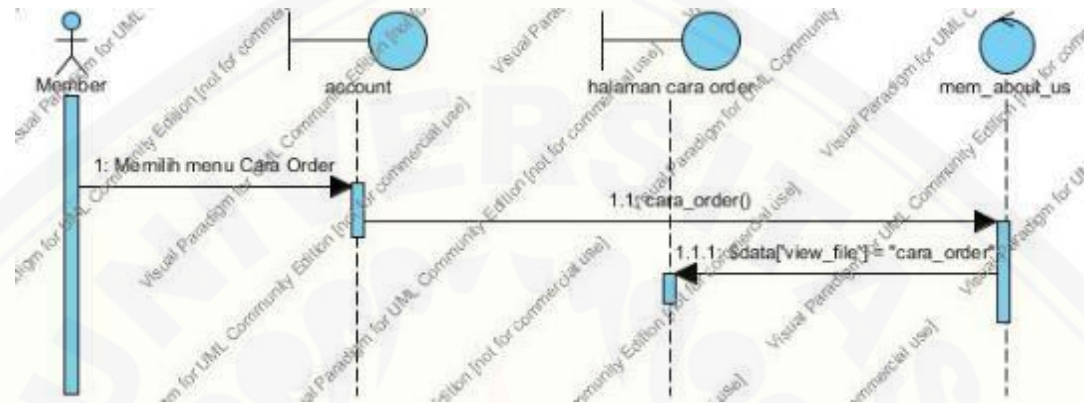
Gambar 42. Sequence Diagram Menambah Testimonial

C.21. Sequence Diagram Melihat Testimonial



Gambar 43. Sequence Diagram Melihat Testimonial

C.22. *Sequence Diagram Melihat Cara Order*



Gambar 44. *Sequence Diagram Melihat Cara Order*

D. Kode Program

D.1. Package *ceo*

D.1.1. Package *controllers*

1. Class *ceo.php*

```

<?php

if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class Ceo extends MX_Controller {

    function __construct() {
        parent::__construct();
        session_start();
        $this->load->model('mdl_ceo', '', TRUE);
    }

    function index($msg = NULL) {
        // Load our view to be displayed
        // to the user
        $data['msg'] = $msg;
        $this->load->view('login', $data);
    }

    public function do_logout() {
        $this->session->sess_destroy();
        redirect('ceo/index');
    }

    function process() {
        // Load the model
        $this->load->model('mdl_ceo');
        // Validate the user can login
        $result = $this->mdl_ceo->validate();
        // Now we verify the result
        if (!$result) {
            // If user did not validate, then show them login page again
            $msg = "<div class='alert alert-danger alert-dismissible'>
                <button type='button' class='close' data-dismiss='alert' aria-
                hidden='true'>&times;</button>
                Invalid username or password. Please try again.</div>";
            $this->index($msg);
        } else {
            // If user did validate,
            // Send them to members area
            $_SESSION['username'] = $username['username'];
            redirect('ceo/create');
        }
    }
}

```



```
}

function create() {
    $admin_id = $this->uri->segment(3);
    $data = $this->get_data_from_post();
    $submit = $this->input->post('submit', TRUE);

    $data['headline'] = "Register Admin";

    $current_url = current_url();
    $data['form_location'] = str_replace('/create', '/submit', $current_url);

    $flash = $this->session->flashdata('item');
    if ($flash != "") {
        $data['flash'] = $flash;
    }

    $data['admin_id'] = $admin_id;
    $template = "ceo";
    $data['view_file'] = "create";
    $this->load->module('template');
    $this->template->$template($data);
}

function submit() {
    $this->load->helper(array('form', 'url'));

    $this->load->library('form_validation');
    $this->form_validation->set_rules('nama_user', 'Nama User', 'required');

    if ($this->form_validation->run() == FALSE) {
        $this->create();
    } else {
        $update_id = $this->uri->segment(3);

        //tambah admin
        $data = $this->get_data_from_post();
        $this->_insert($data);
        $value = "<div class='alert alert-success alert-dismissible'>
            <button type='button' class='close' data-dismiss='alert'
            hidden='true'>&times;</button>
            User berhasil ditambahkan.
            </div>";

        $this->session->set_flashdata('item', $value);
        $update_id = $this->get_max();
        redirect('ceo/create/');
    }
}
```

```

function manage() {
    if (!$this->session->userdata('username')) {
        redirect(base_url() . 'admin/index');
    } else {
        $template = "ceo";
        $data['view_file'] = "manage";
        $this->load->module('template');
        $this->load->template->$template($data);
    }
}

function _display_user_table() {
    $this->load->library('pagination');

    $config['base_url'] = base_url() . 'ceo/manage';
    $config['total_rows'] = $this->db->get('user')->num_rows();
    $config['per_page'] = 3;
    $config['num_links'] = 5;

    $this->pagination->initialize($config);
    $data['query'] = $this->db->get('user', $config['per_page'], $this->uri->segment(3));

    // $data['query'] = $this->get('user_id');
    // $this->db->order_by("user_id", "desc");
    $this->load->view('user_table', $data);
}

function delete_admin($user_id) {
    $submit = $this->input->post('submit', TRUE);
    if ($submit == "No") {
        redirect('ceo/manage');
    }
    if ($submit == "Yes") {
        //delete produk
        $this->_delete($user_id);

        //add flashdata
        $value = "<div class='alert alert-danger alert-dismissible'>
            <button type='button' class='close' data-dismiss='alert' aria-
hidden='true'>&times;</button>
            Data Admin telah berhasil dihapus.
            </div>";
        $this->session->set_flashdata('item', $value);

        redirect('ceo/manage');
    }

    $data['user_id'] = $user_id;
    $template = "ceo";
    $current_url = current_url();
    $data['form_location'] = $current_url;
}

```

```

    $data['view_file'] = "delete_conf";
    $this->load->module('template');
    $this->template->$template($data);
}

function get_data_from_post() {
    $data['nama_user'] = $this->input->post('nama_user', TRUE);
    $data['alamat_user'] = $this->input->post('alamat_user', TRUE);
    $data['email'] = $this->input->post('email', TRUE);
    $data['no_hp_user'] = $this->input->post('no_hp_user', TRUE);
    $data['username'] = $this->input->post('username', TRUE);
    $data['password'] = $this->input->post('password', TRUE);
    return $data;
}

function get($order_by) {
    $this->load->model('mdl_ceo');
    $query = $this->mdl_ceo->get($order_by);
    return $query;
}

function _insert($data) {
    $this->load->model('mdl_ceo');
    $this->mdl_ceo->_insert($data);
}

function _delete($id) {
    $this->load->model('mdl_ceo');
    $this->mdl_ceo->_delete($id);
}

function get_max() {
    $this->load->model('mdl_ceo');
    $max_id = $this->mdl_ceo->get_max();
    return $max_id;
}
}

```

D.1.2. Package models

1. Class *mdl_ceo.php*

```

<?php

if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class Mdl_ceo extends CI_Model {

    function __construct() {

```

```
parent::__construct();
}

function get_table() {
    $table = "user";
    return $table;
}

function get_userID($username){
    $this->db->where('username',$username);
    $query = $this->db->get('user');
    foreach ($query->result() as $row)
    {
        $user_id = $row->user_id;
        $nama = $row->nama_user;
    }
    return $user_id;
}

public function validate() {
    // grab user input
    $username = $this->security->xss_clean($this->input->post('username'));
    $password = $this->security->xss_clean($this->input->post('password'));

    // Prep the query
    $this->db->where('username', $username);
    $this->db->where('password', $password);

    // Run the query
    $query = $this->db->get('ceo');
    // Let's check if there are any results
    if ($query->num_rows == 1) {
        // If there is a user, then create session data
        $rows = $query->row();
        $data = array(
            'ceo_id' => $rows->ceo_id,
            'username' => $rows->username,
            'password' => $rows->password,
            'validated' => true
        );
        $this->session->set_userdata($data);
        return true;
    }
    // If the previous process did not validate
    // then return false.
    return false;
}

function get($order_by) {
    $this->db->order_by($order_by);
    $query = $this->db->get('user');
```

```
    return $query;
}

function _insert($data) {
    $table = $this->get_table();
    $this->db->insert($table, $data);
}

function _update($user_id, $data) {
    $table = $this->get_table();
    $this->db->where('user_id', $user_id);
    $this->db->update($table, $data);
}

function _delete($user_id) {
    $table = $this->get_table();
    $this->db->where('user_id', $user_id);
    $this->db->delete($table);
}

function get_max() {
    $table = $this->get_table();
    $this->db->select_max('user_id');
    $query = $this->db->get($table);
    $row = $query->row();
    $user_id = $row->user_id;
    return $user_id;
}

function _custom_query($mysql_query) {
    $query = $this->db->query($mysql_query);
    return $query;
}
}
```

D.1.3. Package views

1. Class *create.php*

```
<!DOCTYPE html>
<div id="page-wrapper">
  <div class="row">
    <div class="col-lg-13">
      <p></p>
    </div>
  </div>
</div>
```

```

<div class="col-lg-13">
  <div class="panel panel-yellow">
    <div class="panel-heading">
      <div class="row">
        <div class="col-xs-12">
          <h2>
            <span style="color: #666666"><i class="fa fa-comments-o fa-fw"></i> Manage
Admin</span>
          </h2>
        </div>
      </div>
    </div>
  </div>
  <div class="panel-footer">
    <div class="margin-bottom-40" style="color: #666666">
      <div class="panel-body">
        </div>
        <!--panel body untuk form-->
        <div class="panel panel-green margin-bottom-40">

          <div class="panel-body">
            <?php
            if (isset($flash)) {
              echo $flash;
            }
            ?>
            <?php echo form_open($form_location); ?>
            <form method="POST" action="<?php echo base_url(); ?>admin/submit">
              <div class="clearfix"></div>
              <div class="form-group">
                <label for="nama_user" class="control-label">Nama User</label>
                <input type="text" name="nama_user" value="<?php echo $nama_user;
?>" class="form-control" id="nama_produk" required />
              </div>
              <div class="form-group">
                <label for="alamat_user" class="control-label">Alamat</label>
                <input type="text" name="alamat_user" value="<?php echo $alamat_user;
?>" class="form-control" id="harga_produk" required />
              </div>
              <div class="form-group">
                <label for="email" class="control-label">Email</label>
                <input type="email" name="email" value="<?php echo $email; ?>"
class="form-control" id="ukuran_produk" required />
              </div>
              <div class="form-group">
                <label for="no_hp_user" class="control-label">No. HP</label>
                <input type="text" name="no_hp_user" value="<?php echo $no_hp_user;
?>" class="form-control" id="ukuran_produk" required />
              </div>
              <div class="form-group">
                <label for="username" class="control-label">Username</label>

```

```

        <input type="text" name="username" value="<?php echo $username; ?>"
class="form-control" id="bahan_produk">
    </div>
    <div class="form-group">
        <label for="password" class="control-label">Password</label>
        <input type="password" name="password" value="<?php echo $password;
?>" class="form-control" id="berat_produk">
    </div>
    <div class="clearfix"></div>
    <button class="btn btn-sm btn-warning" type="reset" value="Clear"><span
class="glyphicon glyphicon-cloud"></span> Clear</button>
    <button class="btn btn-sm btn-success" type="submit"><span
class="glyphicon glyphicon-check"></span> Submit</button>
</form>

<?php
echo form_close();
?>
</div>
</div>
</div>
</div>
</div>
</div>
</div>

```

2. Class delete_conf.php

```

<div id="page-wrapper">
    <div class="row">
        <div class="col-lg-13">
            <p></p>
        </div>
    </div>
    <div class="col-lg-13">
        <div class="panel panel-yellow">
            <div class="panel-heading">
                <div class="row">
                    <div class="col-xs-12">
                        <!--kalo pake <h2 class="page-header"> ada garis bawah-->
                        <h2>
                            <i class="fa fa-wrench fa-fw" style="color: #666666"></i> <span style="color:
#666666">Delete</span>
                        </h2>
                    </div>
                </div>
            </div>
            <div class="panel-footer" style="background-color: #E6A8D7">
                <div class="margin-bottom-40" style="color: #666666">
                    <!--panel body untuk form-->
                </div>
            </div>
        </div>
    </div>

```

```

<div class="panel panel-green margin-bottom-40">
  <div class="panel-body">
    <p>Apakah Anda yakin ingin menghapus data?</p>
    <?php echo form_open($form_location); ?>
    <button class="btn btn-sm btn-info" href='<?php echo form_submit("submit",
    "Yes"); ?>'> <span class="glyphicon glyphicon-ok"></span> Yes</button>
    <button class="btn btn-sm btn-danger" href='<?php echo
    form_submit("submit", "No"); ?>'> <span class="glyphicon glyphicon-remove"></span>
    No</button>
    <?php echo form_close(); ?>
  </div>
</div>
</div>
</div>
</div>
</div>
</div>

```

3. Class login.php

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <?php $this->load->view('template/css_admin'); ?>
    <title>Login</title>
  </head>
  <body>
    <div class="container" style="margin-top: 100px; width: 500px">
      <div class="panel panel-primary">
        <div class="panel-heading">
          <?php if (isset($account_created)) { ?>
            <h1><?php echo $account_created; ?></h1>
          <?php } else { ?>
            <h1>Please Log In</h1>
          <?php } ?>
        </div>
        <div class="panel-body">
          <?php if (!is_null($msg)) echo $msg; ?>
          <?php echo form_open('ceo/process'); ?>
          <br />
          <div class="input-group">
            <span class="input-group-addon"><i class="fa fa-user"></i></span>
            <input type="text" value="" name="username" id="username" class="form-
            control" placeholder="username" required autofocus>
          </div>
          <br />
          <div class="input-group">
            <span class="input-group-addon"><i class="fa fa-lock"></i></span>
            <input type="password" value="" name="password" id="password" class="form-
            control" placeholder="password" required>
          </div>

```



```

        </div>
        <br />
        <button class="btn btn-lg btn-primary btn-block" name="submit"
type="Submit">Log in</button>
        <?php echo form_close(); ?>
    </div>
</div>
</div> <!-- /container -->
<?php $this->load->view('template/js'); ?>
</body>
</html>

```

4. *Class manage.php*

```

<div id="page-wrapper">
    <div class="row">
        <div class="col-lg-13">
            <p></p>
        </div>
    </div>
    <div class="col-lg-13">
        <div class="panel panel-yellow">
            <div class="panel-heading">
                <div class="row">
                    <div class="col-xs-12">
                        <h2>
                            <span style="color: #666666"><i class="fa fa-comments-o fa-fw"></i> Manage
Admin</span>
                        </h2>
                    </div>
                </div>
            </div>
            <div class="panel-footer">
                <div class="margin-bottom-40" style="color: #666666">
                    <div class="panel-body">
                        <div class="col-xs-12 text-right">
                            <a class="btn btn-sm btn-success" title="Tambah Admin" href="<?php echo
base_url(); ?>ceo/create">
                                <span class="glyphicon glyphicon-plus"></span> Tambah Admin
                            </a>
                        </div>
                    </div>
                </div>
            </div>
            <!--panel body untuk form-->
            <div class="panel panel-green margin-bottom-40">
                <div class="panel-body">
                    <center><?php
                        if (isset($flash)) {
                            echo $flash;
                        }
                    ?></center>
                </div>
            </div>
        </div>
    </div>

```

```

        <div class="panel-body">
            <?php echo Modules::run('ceo/_display_user_table'); ?> </div>
        </div>
    </div>
</div>
<div>
</div>
<center>
    <?php
    echo "<span style='font-size: 17px'>" . $this->pagination->create_links() . "</span>";
    ?>
</center>
</div>
</div>

```

5. Class *user_table.php*

```

<div class="dataTable_wrapper">
<table class="table table-responsive table-hover" style="font-family: tahoma; font-size: 12px">
<tr>
<th><center>No.</center></th>
<th><center>Nama User</center></th>
<th><center>Alamat</center></th>
<th><center>Email</center></th>
<th><center>No. HP</center></th>
<th><center>Action</center></th>
</tr>
<?php
$count = 0;
foreach ($query->result() as $row) {
    $count++;
    ?>
<tr>
<td><center><?php echo $count; ?></center></td>
<td><?php echo $row->nama_user; ?></td>
<td style="width: 200px"><?php echo $row->alamat_user; ?></td>
<td><center><?php echo $row->email; ?></center></td>
<td><center><?php echo $row->no_hp_user; ?></center></td>
<td>
<center>
<div class="btn-group">
<a class="btn btn-xs btn-danger" title="Delete" href="delete_admin/<?php echo
$row->user_id ?>"><span class="fa fa-trash"></span></a>
</div>
</center>
</td>
</tr>
<?php
}
?>

```

```
</table>
</div>
```

D.2. *Package admin*

D.2.1. *Package controllers*

1. *Class admin.php*

```
<?php
if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class Admin extends MX_Controller {

    function __construct() {
        parent::__construct();
        session_start();
        $this->load->model('mdl_admin', '', TRUE);
    }

    function index($msg = NULL) {
        // Load our view to be displayed
        // to the user
        $data['msg'] = $msg;
        $this->load->view('login', $data);
    }

    function process() {
        // Load the model
        $this->load->model('mdl_admin');
        // Validate the user can login
        $result = $this->mdl_admin->validate();
        // Now we verify the result
        if (!$result) {
            // If user did not validate, then show them login page again
            $msg = "<div class='alert alert-danger alert-dismissible'>
                <button type='button' class='close' data-dismiss='alert' aria-
                hidden='true'>&times;</button>
                Invalid username or password. Please try again.</div>";
            $this->index($msg);
        } else {
            // If user did validate,
            // Send them to members area
            $_SESSION['username'] = $username['username'];
            redirect('adm_dashboard/home');
        }
    }
}
```

```
function create() {
    $user_id = $this->uri->segment(3);
    $data = $this->get_data_from_post();
    $submit = $this->input->post('submit', TRUE);

    if ($user_id > 0) {

        if ($submit != "Submit") {
            //ambil data dr db
            $data = $this->get_data_from_db($user_id);
        }

        $data['headline'] = "Update Admin Profile";
    } else {
        $data['headline'] = "Sign Up";
    }

    $current_url = current_url();
    $data['form_location'] = str_replace('/create', '/submit', $current_url);

    $flash = $this->session->flashdata('item');
    if ($flash != "") {
        $data['flash'] = $flash;
    }

    $data['user_id'] = $user_id;
    $template = "index";
    $data['view_file'] = "create";
    $this->load->module('template');
    $this->template->$template($data);
}

function update() {
    $user_id = $this->uri->segment(3);
    $user_id = $this->mdl_admin->get_userID($this->input->post('username'));

    $data = $this->get_data_from_post();
    $submit = $this->input->post('submit', TRUE);

    if ($submit != "Submit") {
        $data = $this->get_data_from_db($user_id);
    }
    $data['headline'] = "Ubah Profil";

    $current_url = current_url();
    $data['form_location'] = str_replace('/update', '/submit', $current_url);

    $template = "admin";
    $data['view_file'] = "update";
```

```

$this->load->module('template');
$this->load->template->$template($data);
}

function submit() {
    $this->load->helper(array('form', 'url'));

    $this->load->library('form_validation');
    $this->form_validation->set_rules('nama_user', 'Nama User', 'required');

    if ($this->form_validation->run() == FALSE) {
        $this->create();
    } else {
        $update_id = $this->uri->segment(3);

        if ($update_id > 0) {
            //untuk edit, bukan create
            $data = $this->get_data_from_post();
            $this->_update($update_id, $data);
            $value = "<div class='alert alert-success alert-dismissable'>
                <button type='button' class='close' data-dismiss='alert' aria-
hidden='true'>&times;</button>
                Data user berhasil diubah.
            </div>";

            $this->session->set_flashdata('item', $value);
            $update_id = $this->get_max();
            redirect('adm_login/update/' . $update_id);
        } else {
            //tambah admin
            $data = $this->get_data_from_post();
            $this->_insert($data);
            $value = "<div class='alert alert-success alert-dismissable'>
                <button type='button' class='close' data-dismiss='alert' aria-
hidden='true'>&times;</button>
                User berhasil ditambahkan.
            </div>";

            $this->session->set_flashdata('item', $value);
            $update_id = $this->get_max();
            redirect('adm_login/create/');
        }
    }
}

function get_data_from_post() {
    $data['nama_user'] = $this->input->post('nama_user', TRUE);
    $data['alamat_user'] = $this->input->post('alamat_user', TRUE);
    $data['email'] = $this->input->post('email', TRUE);
    $data['no_hp_user'] = $this->input->post('no_hp_user', TRUE);
    $data['username'] = $this->input->post('username', TRUE);
}

```

```
$data['password'] = $this->input->post('password', TRUE);
// $data['level'] = $this->input->post('level', TRUE);
return $data;
}

function get_data_from_db($update_id) {
    $query = $this->get_where($update_id);
    foreach ($query->result() as $row) {
        $data['user_id'] = $row->user_id;
        $data['nama_user'] = $row->nama_user;
        $data['alamat_user'] = $row->alamat_user;
        $data['email'] = $row->email;
        $data['no_hp_user'] = $row->no_hp_user;
        $data['username'] = $row->username;
        $data['password'] = $row->password;
    }
    if (!isset($data)) {
        $data = "";
    }

    return $data;
}

function get($order_by) {
    $this->load->model('mdl_admin');
    $query = $this->mdl_admin->get($order_by);
    return $query;
}

function get_where($id) {
    $this->load->model('mdl_admin');
    $query = $this->mdl_admin->get_where($id);
    return $query;
}

function _insert($data) {
    $this->load->model('mdl_admin');
    $this->mdl_admin->_insert($data);
}

function _update($id, $data) {
    $this->load->model('mdl_admin');
    $this->mdl_admin->_update($id, $data);
}

function _delete($id) {
    $this->load->model('mdl_admin');
    $this->mdl_admin->_delete($id);
}

function get_max() {
```

```
$this->load->model('mdl_admin');
$max_id = $this->mdl_admin->get_max();
return $max_id;
}
}
```

D.2.2. Package models

2. Class mdl_admin.php

```
<?php
if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class Mdl_admin extends CI_Model {

    function __construct() {
        parent::__construct();
    }

    function get_table() {
        $table = "user";
        return $table;
    }

    function get_userID($username) {
        $this->db->where('username', $username);
        $query = $this->db->get('user');
        foreach ($query->result() as $row) {
            $user_id = $row->user_id;
            $nama = $row->nama_user;
        }
        return $user_id;
    }

    public function validate() {
        // grab user input
        $username = $this->security->xss_clean($this->input->post('username'));
        $password = $this->security->xss_clean($this->input->post('password'));

        // Prep the query
        $this->db->where('username', $username);
        $this->db->where('password', $password);

        // Run the query
        $query = $this->db->get('user');
        // Let's check if there are any results
        if ($query->num_rows == 1) {
            // If there is a user, then create session data
```

```
$rows = $query->row();
$data = array(
    'user_id' => $rows->user_id,
    'nama_user' => $rows->nama_user,
    'alamat_user' => $rows->alamat_user,
    'email' => $rows->email,
    'no_hp_user' => $rows->no_hp_user,
    'username' => $rows->username,
    'password' => $rows->password,
    'validated' => true
);
$this->session->set_userdata($data);
return true;
}
// If the previous process did not validate
// then return false.
return false;
}

function get($order_by) {
    $table = $this->get_table();
    $this->db->order_by($order_by);
    $query = $this->db->get($table);
    return $query;
}

function get_with_limit($limit, $offset, $order_by) {
    $table = $this->get_table();
    $this->db->limit($limit, $offset);
    $this->db->order_by($order_by);
    $query = $this->db->get($table);
    return $query;
}

function get_where($user_id) {
    $table = $this->get_table();
    $this->db->where('user_id', $user_id);
    $query = $this->db->get($table);
    return $query;
}

function get_where_custom($col, $value) {
    $table = $this->get_table();
    $this->db->where($col, $value);
    $query = $this->db->get($table);
    return $query;
}

function _insert($data) {
    $table = $this->get_table();
    $this->db->insert($table, $data);
}
```



```

}

function _update($user_id, $data) {
    $table = $this->get_table();
    $this->db->where('user_id', $user_id);
    $this->db->update($table, $data);
}

function _delete($user_id) {
    $table = $this->get_table();
    $this->db->where('user_id', $user_id);
    $this->db->delete($table);
}

function get_max() {
    $table = $this->get_table();
    $this->db->select_max('user_id');
    $query = $this->db->get($table);
    $row = $query->row();
    $user_id = $row->user_id;
    return $user_id;
}
}

```

D.2.3. Package views

1. Class login.php

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <?php $this->load->view('template/css_admin'); ?>
    <title>Login</title>
  </head>
  <body>
    <div class="container" style="margin-top: 100px; width: 500px">
      <div class="panel panel-primary">
        <div class="panel-heading">
          <?php if (isset($account_created)) { ?>
            <h1><?php echo $account_created; ?></h1>
          <?php } else { ?>
            <h1>Please Log In</h1>
          <?php } ?>
        </div>
        <div class="panel-body">
          <?php if (!is_null($msg)) echo $msg; ?>
          <?php echo form_open('admin/process'); ?>
          <br />
          <div class="input-group">
            <span class="input-group-addon"><i class="fa fa-user"></i></span>

```

```

        <input type="text" value="" name="username" id="username" class="form-
control" placeholder="username" required autofocus>
    </div>
    <br />
    <div class="input-group">
        <span class="input-group-addon"><i class="fa fa-lock"></i></span>
        <input type="password" value="" name="password" id="password"
class="form-control" placeholder="password" required>
    </div>
    <br />
    <button class="btn btn-lg btn-primary btn-block" name="submit"
type="Submit">Log in</button>
    <?php echo form_close(); ?>
</div>
</div>
</div> <!-- /container -->
<?php $this->load->view('template/js'); ?>
</body>
</html>

```

2. *Class update.php*

```

<div id="page-wrapper">
    <div class="row">
        <div class="col-lg-13">
            <p></p>
        </div>
    </div>
    <div class="col-lg-13">
        <div class="panel panel-green">
            <div class="panel-heading">
                <div class="row">
                    <div class="col-xs-12">
                        <h2>
                            <span style="color: #666666"><i class="fa fa-briefcase fa-fw"></i></span><?php echo
$headline; ?></span>
                        </h2>
                    </div>
                </div>
            </div>
            <div class="panel-footer">
                <div class="margin-bottom-40" style="color: #666666">
                    <div class="panel-body">
                        <div class="col-xs-12 text-right">
                            <center>
                                <?php
                                if (isset($flash)) {
                                    echo $flash;
                                }
                            </center>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

```

        echo validation_errors("<p style='color: red; background-color: white'>",
"</p>");
        ?>
        </center>
        <a class="btn btn-sm btn-warning" title="Kembali" href="<?php echo
base_url(); ?>adm_dashboard/home">
        <span class="glyphicon glyphicon-chevron-left"></span> Back to
Dashboard
        </a>
    </div>
</div>
<div class="panel panel-green margin-bottom-40">
    <div class="panel-body">
        <h2><i class="fa fa-edit"></i> <?php echo $headline; ?></h2>
        <?php
        if (isset($flash)) {
            echo $flash;
        }
        echo form_open($form_location);
        $user_id = $this->session->userdata('user_id');
        $fname = ucfirst($this->session->userdata('nama_user'));
        $email = $this->session->userdata('email');
        $uname = $this->session->userdata('username');
        $pass = $this->session->userdata('password');
        $alamat = $this->session->userdata('alamat_user');
        $no_hp = $this->session->userdata('no_hp_user');
        ?>
        <form method="POST" action="<?php echo base_url();
?>mem_member/submit">
            <?php echo form_hidden('user_id', $user_id); ?>
            <div class="form-group">
                <label for="nama_depan" class="control-label">Nama Admin</label>
                <input type="text" name="nama_user" value="<?php echo $fname; ?>"
class="form-control" id="nama_depan" required />
            </div>
            <div class="form-group">
                <label for="email" class="control-label">Email</label>
                <input type="email" name="email_user" value="<?php echo $email; ?>"
class="form-control" id="email" required />
            </div>
            <div class="form-group">
                <label for="alamat" class="control-label">Alamat</label>
                <input type="text" name="alamat_user" value="<?php echo $alamat; ?>"
class="form-control" id="alamat" required />
            </div>
            <div class="form-group">
                <label for="no_hp" class="control-label">No. HP</label>
                <input type="text" name="no_hp_user" value="<?php echo $no_hp; ?>"
class="form-control" id="no_hp" required />
            </div>
            <div class="form-group">

```

```

        <label for="username" class="control-label">Username</label>
        <input type="text" name="username" value="<?php echo $uname; ?>"
class="form-control" id="email" required />
    </div>
    <div class="form-group">
        <label for="password" class="control-label">Password</label>
        <input type="password" name="password" value="<?php echo $pass; ?>"
class="form-control" id="password" required />
    </div>
    <button class="btn btn-sm btn-success" type="submit"><span
class="glyphicon glyphicon-check"></span> Submit</button>
</form>
<?php
echo form_close();
?>
</div>
</div>
</div>
</div>
</div>
</div>
</div>

```

D.3. *Package adm_kategori*

D.3.1. *Package controllers*

```

<?php
if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class Adm_kategori extends MX_Controller {

    function __construct() {
        parent::__construct();
    }

    function get_breadcrumb($kategori_id) {
        $breadcrumb = "";

        do {
            if (!isset($parent_kategori)) {
                $parent_kategori = $kategori_id;
            }

            $parent_kategori = $this->get_parent_kategori($parent_kategori);

            if ($parent_kategori > 0) {
                $parents[] = $parent_kategori;
            }
        } while ($parent_kategori != "");
    }
}

```

```
if (isset($parents)) {
    $parents = array_reverse($parents);
    foreach ($parents as $parent) {
        $nama_kategori = $this->get_nama_kategori($parent);
        $breadcrumb.= $nama_kategori . " > ";
    }
}
return $breadcrumb;
}

function get_end_of_line_kategori() {
    $max_depth = 2;

    $query = $this->get('nama_kategori');
    foreach ($query->result() as $row) {
        $kategori_id = $row->kategori_id;
        $parent_kategori = $row->parent_kategori;
        $kategori_depth = $this->get_kategori_depth($parent_kategori);

        if ($kategori_depth == $max_depth) {
            //ini jadi 'end of line' kategori
            $categories[] /* $categories[] */ = $kategori_id;
        }
    }

    if (!isset($categories)) {
        $categories = " ";
    }

    return $categories;
}

function _kategori_baru_allowed($parent_kategori) {
    //karena di modul web_setting >> max_kategori_depth disetting 3,
    //jika level kategori sudah 3 maka ga bisa nambah kategori baru
    //return TRUE or FALSE
    //get max allowed depth
    $max_depth = $this->get_max_kategori_depth();

    //get current category depth
    $current_depth = $this->get_kategori_depth($parent_kategori);

    //utk mengetahui jika kategori baru allowed
    if ($current_depth < $max_depth) {
        return TRUE;
    } else {
        return FALSE;
    }
}
```

```
function get_max_kategori_depth() {
    $max_depth = 2;
    return $max_depth;
}

function get_kategori_depth($parent_kategori) {
    //bisa tau level kategori dilihat dr parent kategori-nya
    $depth = 0;

    do {
        $depth++;

        $parent_kategori = $this->get_parent_kategori($parent_kategori);
    } while ($parent_kategori != "");

    return $depth;
}

function get_parent_kategori($id) {
    $query = $this->get_where($id);
    foreach ($query->result() as $row) {
        $parent_kategori = $row->parent_kategori;
    }

    if (!isset($parent_kategori)) {
        $parent_kategori = "";
    }

    return $parent_kategori;
}

function delete_kategori($kategori_id) {
    $submit = $this->input->post('submit', TRUE);
    if ($submit == "No") {
        redirect('adm_kategori/manage');
    }
    if ($submit == "Yes") {
        //delete produk
        $this->_delete($kategori_id);

        //add flashdata
        $value = "<div class='alert alert-danger alert-dismissible'>
            <button type='button' class='close' data-dismiss='alert' aria-
hidden='true'>&times;</button>
            Kategori berhasil dihapus.
            </div>";
        $this->session->set_flashdata('item', $value);

        redirect('adm_kategori/manage');
    }
}
```

```
$data['kategori_id'] = $kategori_id;
$template = "admin";
$current_url = current_url();
$data['form_location'] = $current_url;
$data['view_file'] = "delete_conf";
$this->load->module('template');
$this->template->$template($data);
}

function _display_kategori_table($parent_kategori) {
    $data['query'] = $this->get_where_custom('parent_kategori', $parent_kategori);
    $this->load->view('kategori_table', $data);
}

function submit() {
    //untuk data yang telah disubmit
    $parent_kategori = $this->uri->segment(4);

    if (!is_numeric($parent_kategori)) {
        $parent_kategori = 0;
    }

    $this->load->helper(array('form', 'url'));

    $this->load->library('form_validation');
    $this->form_validation->set_rules('nama_kategori', 'Nama Kategori', 'required');

    if ($this->form_validation->run() == FALSE) {
        $this->create();
    } else {
        $update_id = $this->uri->segment(3);

        if ($update_id > 0) {
            //untuk edit, bukan create
            $data = $this->get_data_from_post();
            $data['kategori_url'] = url_title($data['nama_kategori']);
            $this->_update($update_id, $data);
            $value = "<div class='alert alert-success alert-dismissable'>
                <button type='button' class='close' data-dismiss='alert' aria-
                hidden='true'>&times;</button>
                Kategori berhasil diubah.
                </div>";
            $parent_kategori = $update_id;
        } else {
            //tambah produk
            $data = $this->get_data_from_post();
            $data['kategori_url'] = url_title($data['nama_kategori']);
            $data['parent_kategori'] = $parent_kategori;
            $this->_insert($data);
            $value = "<div class='alert alert-success alert-dismissable'>
```

```
<button type='button' class='close' data-dismiss='alert' aria-
hidden='true'>&times;</button>
    Kategori berhasil ditambahkan.
</div>";
    $update_id = $this->get_max();
}

//flashdata
//session data yg hanya untuk server request itu sendiri,
//selebihnya akan hilang jika page di refresh
//ini untuk info/status messages

$this->session->set_flashdata('item', $value);

redirect('adm_kategori/manage/' . $parent_kategori);
}
}

function get_data_from_post() {
    $data['nama_kategori'] = $this->input->post('nama_kategori', TRUE);
    return $data;
}

function get_data_from_db($update_id) {
    $query = $this->get_where($update_id);
    foreach ($query->result() as $row) {
        $data['nama_kategori'] = $row->nama_kategori;
    }
    if (!isset($data)) {
        $data = "";
    }

    return $data;
}

function create() {
    $kategori_id = $this->uri->segment(3);
    $data = $this->get_data_from_post();
    $submit = $this->input->post('submit', TRUE);

    if ($kategori_id > 0) {

        if ($submit != "Submit") {
            //ambil data dr db
            $data = $this->get_data_from_db($kategori_id);
        }

        $data['headline'] = "Ubah Kategori";
    } else {
```



```
$data['headline'] = "Tambah Kategori";
}

$current_url = current_url();
$data['form_location'] = str_replace('/create', '/submit', $current_url);

$flash = $this->session->flashdata('item');
if ($flash != "") {
    $data['flash'] = $flash;
}

$data['kategori_id'] = $kategori_id;
$template = "admin";
$data['view_file'] = "create";
$this->load->module('template');
$this->template->$template($data);
}

function get_nama_kategori($kategori_id) {
    $data = $this->get_data_from_db($kategori_id);
    $nama_kategori = $data['nama_kategori'];
    return $nama_kategori;
}

function manage() {
    if (!$this->session->userdata('username')) {
        redirect(base_url() . 'adm_login/index');
    } else {
        $template = "admin";
        $parent_kategori = $this->uri->segment(3);

        if (($parent_kategori < 1) || (!is_numeric($parent_kategori))) {
            $parent_kategori = 0;
        }

        $data['parent_kategori'] = $parent_kategori;

        if ($parent_kategori > 0) {
            $data['headline'] = "Manage " . $this->get_nama_kategori($parent_kategori);
        } else {
            //untuk kategori utama, jd pake headline default
            $data['headline'] = "Manage Kategori Produk";
        }

        $flash = $this->session->flashdata('item');
        if ($flash != "") {
            $data['flash'] = $flash;
        }

        $data['kategori_baru_allowed'] = $this->_kategori_baru_allowed($parent_kategori);
    }
}
```

```
$data['view_file'] = "manage";
$this->load->module('template');
$this->template->$template($data);
}
}

function get($order_by) {
    $this->load->model('mdl_kategori');
    $query = $this->mdl_kategori->get($order_by);
    return $query;
}

function get_where($kategori_id) {
    $this->load->model('mdl_kategori');
    $query = $this->mdl_kategori->get_where($kategori_id);
    return $query;
}

function get_where_custom($col, $value) {
    $this->load->model('mdl_kategori');
    $query = $this->mdl_kategori->get_where_custom($col, $value);
    return $query;
}

function _insert($data) {
    $this->load->model('mdl_kategori');
    $this->mdl_kategori->_insert($data);
}

function _update($kategori_id, $data) {
    $this->load->model('mdl_kategori');
    $this->mdl_kategori->_update($kategori_id, $data);
}

function _delete($kategori_id) {
    $this->load->model('mdl_kategori');
    $this->mdl_kategori->_delete($kategori_id);
}

function get_max() {
    $this->load->model('mdl_kategori');
    $max_id = $this->mdl_kategori->get_max();
    return $max_id;
}
}
```

D.3.2. Package models

```
<?php
if (!defined('BASEPATH'))
```

```
exit('No direct script access allowed');

class Mdl_kategori extends CI_Model {

    function __construct() {
        parent::__construct();
    }

    function get_table() {
        $table = "kategori";
        return $table;
    }

    function get($order_by) {
        $table = $this->get_table();
        $this->db->order_by($order_by);
        $query = $this->db->get($table);
        return $query;
    }

    function get_where($kategori_id) {
        $table = $this->get_table();
        $this->db->where('kategori_id', $kategori_id);
        $query = $this->db->get($table);
        return $query;
    }

    function get_where_custom($col, $value) {
        $table = $this->get_table();
        $this->db->where($col, $value);
        $query = $this->db->get($table);
        return $query;
    }

    function _insert($data) {
        $table = $this->get_table();
        $this->db->insert($table, $data);
    }

    function _update($kategori_id, $data) {
        $table = $this->get_table();
        $this->db->where('kategori_id', $kategori_id);
        $this->db->update($table, $data);
    }

    function _delete($kategori_id) {
        $table = $this->get_table();
        $this->db->where('kategori_id', $kategori_id);
        $this->db->delete($table);
    }
}
```

```

function get_max() {
    $table = $this->get_table();
    $this->db->select_max('kategori_id');
    $query = $this->db->get($table);
    $row = $query->row();
    $kategori_id = $row->kategori_id;
    return $kategori_id;
}
}

```

D.3.3. Package views

1. Class create.php

```

<div id="page-wrapper">
  <div class="row">
    <div class="col-lg-13">
      <p></p>
    </div>
  </div>
  <div class="col-lg-13">
    <div class="panel panel-green">
      <div class="panel-heading">
        <div class="row">
          <div class="col-xs-12">
            <h2>
              <span style="color: #666666"><i class="fa fa-list fa-fw"></i> <?php echo
$headline; ?></span>
            </h2>
          </div>
        </div>
      </div>
      <div class="panel-footer" style="background-color: #CDCDCD">
        <div class="margin-bottom-40" style="color: #666666">
          <div class="panel-body">
            <div class="col-xs-12 text-right">
              <center>
                <?php
                if (isset($flash)) {
                  echo $flash;
                }
                echo validation_errors("<p style='color: red; background-color: white'>",
"");
              </p>");
              ?>
            </center>
            <a class="btn btn-sm btn-warning" title="Kembali" href="<?php echo
base_url(); ?>adm_kategori/manage">
              <span class="glyphicon glyphicon-chevron-left"></span> Back to Manage
            </a>
            <?php
            if ($kategori_id > 0) {

```

```

        include('additional_options.php');
    }
    ?>
</div>
</div>
<div class="panel panel-green margin-bottom-40">
    <div class="panel-body">
        <?php
        echo form_open($form_location);
        ?>
        <form method="POST" action="<?php echo base_url();
?>adm_kategori/submit">
            <div class="form-group">
                <label for="nama_kategori">Nama Kategori</label>
                <input type="text" name="nama_kategori" value="<?php echo
$nama_kategori; ?>" class="form-control" id="nama_kategori">
            </div>
            <div class="clearfix"></div>
            <button type="submit" class="btn btn-default">Submit</button>
        </form>
        <?php
        echo form_close();
        ?>
    </div>
</div>
</div>
</div>
</div>
</div>

```

2. *Class delete_conf.php*

```

<div id="page-wrapper">
    <div class="row">
        <div class="col-lg-13">
            <p></p>
        </div>
    </div>
    <div class="col-lg-13">
        <div class="panel panel-yellow">
            <div class="panel-heading">
                <div class="row">
                    <div class="col-xs-12">
                        <h2>
                            <i class="fa fa-wrench fa-fw" style="color: #666666"></i> <span style="color:
#666666">Delete</span>
                        </h2>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

```

</div>
<div class="panel-footer" style="background-color: #E6A8D7">
  <div class="margin-bottom-40" style="color: #666666">
    <!--panel body untuk form-->
    <div class="panel panel-green margin-bottom-40">
      <div class="panel-body">
        <p>Apakah Anda yakin ingin menghapus data?</p>
        <?php echo form_open($form_location); ?>
        <button class="btn btn-sm btn-default" href="<?php echo
form_submit('submit', 'Yes'); ?> <span class="glyphicon glyphicon-ok"></span> Yes</button>
        <button class="btn btn-sm btn-danger" href="<?php echo
form_submit('submit', 'No'); ?> <span class="glyphicon glyphicon-remove"></span>
No</button>
        <?php echo form_close(); ?>
      </div>
    </div>
  </div>
</div>
</div>
</div>
</div>
</div>
</div>

```

3. *Class kategori_table.php*

```

<?php
$num_rows = $query->num_rows();
if ($num_rows > 0) {
  ?>
  <table class="table table-responsive table-hover" style="font-family: tahoma; font-size:
12px">
    <tr>
      <th><center>No.</center></th>
      <th><center>Nama Kategori</center></th>
      <th><center>Action</center></th>
    </tr>
    <?php
    $count = 0;
    foreach ($query->result() as $row) {
      $count++;
      ?>
      <tr>
        <td><center><?php echo $count; ?></center></td>
        <td><?php echo $row->nama_kategori; ?></td>
        <td>
          <center>
            <div class="btn-group">
              <a class="btn btn-xs btn-primary" title="Edit" href="<?php echo base_url();
?>adm_kategori/manage/<?php echo $row->kategori_id ?>"><span class="glyphicon glyphicon-
edit"></span></a>

```

```

        <a class="btn btn-xs btn-danger" title="Delete" href="<?php echo base_url();
?>adm_kategori/delete_kategori/<?php echo $row->kategori_id ?>"><span class="glyphicon
glyphicon-trash"></span></a>
    </div>
</center>
</td>
</tr>
<?php
}
?>
</table>
<?php
}
?>

```

4. *Class manage.php*

```

<div id="page-wrapper">
<div class="row">
<div class="col-lg-13">
<p></p>
</div>
</div>
<div class="col-lg-13">
<div class="panel panel-red">
<div class="panel-heading">
<div class="row">
<div class="col-xs-12">
<h2>
<span style="color: #666666"><i class="fa fa-list-alt fa-fw"></i><?php echo
$headline; ?></span>
</h2>
</div>
</div>
</div>
<div class="panel-footer">
<div class="margin-bottom-40" style="color: #666666">
<div class="panel-body">
<div class="col-xs-12 text-right">
<center>
<?php
if (isset($flash)) {
    echo $flash;
}
?></center>
<a class="btn btn-sm btn-success" title="Tambah Kategori (pada level ini)"
href="<?php echo base_url(); ?><?php echo 'adm_kategori/create/sub/' . $parent_kategori; ?>">
<span class="glyphicon glyphicon-plus"></span> Tambah Kategori
</a>
<?php

```

```

        if ($parent_kategori > 0) {
            ?>
            <a class="btn btn-sm btn-info" title="Ubah Parent Kategori" href="<?php
echo base_url(); ?><?php echo 'adm_kategori/create/' . $parent_kategori; ?>">
            <span class="glyphicon glyphicon-list"></span> Update Parent Kategori
            </a>
            <?php
        }
        ?>
    </div>
</div>
<div class="panel panel-green margin-bottom-40">
    <div class="panel-body">
        <?php
        echo Modules::run('adm_kategori/_display_kategori_table', $parent_kategori);
        ?>
    </div>
</div>
</div>
</div>
</div>
</div>
</div>

```

D.4. *Package adm_member*

D.4.1. *Package controllers*

1. Class adm_member.php

```

<?php

if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class Adm_member extends MX_Controller {

    function __construct() {
        parent::__construct();
    }

    function get_data_from_db($update_id) {
        $query = $this->get_where($update_id);
        foreach ($query->result() as $row) {
            $data['member_id'] = $row->member_id;
            $data['nama_user'] = $row->nama_user;
            $data['kota'] = $row->kota;
            $data['pesan'] = $row->pesan;
        }

        if (!isset($data)) {

```



```
$data = "";
}

return $data;
}

function level() {
    $member_id = $this->uri->segment(3);
    $submit = $this->input->post('submit', TRUE);

    if ($submit == "Submit") {
        $data['member_id'] = $member_id;
        $data['level_id'] = $this->input->post('level_id', TRUE);
        $this->update($member_id, $data);
        $value = "<div class='alert alert-success alert-dismissible'>
            <button type='button' class='close' data-dismiss='alert' aria-
hidden='true'>&times;</button>
            Level Member pada Member ID " . $member_id . " berhasil diubah.
        </div>";
        $this->session->set_flashdata('item', $value);
        redirect('adm_member/manage');
    }
    $data['member_id'] = $member_id;
    $template = "admin";
    $data['view_file'] = "level";
    $current_url = current_url();
    $data['form_location'] = $current_url;
    $this->load->module('template');
    $this->template->$template($data);
}

function manage() {
    if (!$this->session->userdata('username')) {
        redirect(base_url() . 'admin/index');
    } else {
        $template = "admin";
        $data['view_file'] = "manage";
        $this->load->module('template');
        $this->load->template->$template($data);
    }
}

function _display_member_table() {
    $this->load->library('pagination');

    $config['base_url'] = base_url() . 'admin/manage';
    $config['total_rows'] = $this->db->get('member')->num_rows();
    $config['per_page'] = 10;
    $config['num_links'] = 5;

    $this->pagination->initialize($config);
```

```
$data['query'] = $this->db->get('member', $config['per_page'], $this->uri->segment(3));

$data['query'] = $this->get('member_id');

$member_id = $this->uri->segment(3);

$this->load->view('member_table', $data);
}

function get_data_from_post() {
    $data['nama_user'] = $this->input->post('nama_user', TRUE);
    $data['email'] = $this->input->post('email', TRUE);
    return $data;
}

function get($order_by) {
    $this->load->model('mdl_member');
    $query = $this->mdl_member->get($order_by);
    return $query;
}

function get_where($id) {
    $this->load->model('mdl_member');
    $query = $this->mdl_member->get_where($id);
    return $query;
}

function _insert($data) {
    $this->load->model('mdl_member');
    $this->mdl_member->_insert($data);
}

function _update($id, $data) {
    $this->load->model('mdl_member');
    $this->mdl_member->_update($id, $data);
}

function _delete($id) {
    $this->load->model('mdl_member');
    $this->mdl_member->_delete($id);
}

function get_max() {
    $this->load->model('mdl_member');
    $max_id = $this->mdl_member->get_max();
    return $max_id;
}
}
```

D.4.2. *Package models*

1. Class mdl_member.php

```
<?php

if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class Mdl_member extends CI_Model {

    function __construct() {
        parent::__construct();
    }

    function get_table() {
        $table = "member";
        return $table;
    }

    function get($order_by) {
        $this->db->order_by($order_by);

        $query = $this->db->get('member');
        return $query;
    }

    function get_id() {
        $data['query'] = $this->get($member_id);
        foreach ($query->result() as $row) {
            $row->member_id;
        }
    }

    function get_total() {
        $this->db->select_sum('total_bayar');
        $this->db->from('orders');
        $this->db->join('member', 'member.member_id = orders.member_id');
        $this->db->where('member.member_id = ' . $this->get_id());

        $hasil = $this->db->get_total();
        return $hasil;
    }

    function get_where($member_id) {
        $table = $this->get_table();
        $this->db->where('member_id', $member_id);
        $query = $this->db->get($table);
        return $query;
    }
}
```

```

function _insert($data) {
    $table = $this->get_table();
    $this->db->insert($table, $data);
}

function _update($member_id, $data) {
    $table = $this->get_table();
    $this->db->where('member_id', $member_id);
    $this->db->update($table, $data);
}

function _delete($member_id) {
    $table = $this->get_table();
    $this->db->where('member_id', $member_id);
    $this->db->delete($table);
}

function get_max() {
    $table = $this->get_table();
    $this->db->select_max('member_id');
    $query = $this->db->get($table);
    $row = $query->row();
    $member_id = $row->member_id;
    return $member_id;
}
}

```

D.4.3. Package views

1. Class level.php

```

<div id="page-wrapper">
  <div class="row">
    <div class="col-lg-13">
      <p></p>
    </div>
  </div>
  <div class="col-lg-13">
    <div class="panel panel-purple">
      <div class="panel-heading">
        <div class="row">
          <div class="col-xs-12">
            <h2>
              <span style="color: #666666"><i class="fa fa-wrench fa-fw"></i> Manage
Level Member</span>
            </h2>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```

<div class="panel-footer">
  <div class="margin-bottom-40" style="color: #666666">
    <div class="panel-body">
      <div class="col-xs-12 text-right">
        <a class="btn btn-sm btn-warning" title="Kembali" href="<?php echo
base_url(); ?>adm_member/manage">
          <span class="glyphicon glyphicon-chevron-left"></span> Back to Manage
        </a>
      </div>
    </div>
  </div>
  <div class="panel panel-green margin-bottom-40">
    <div class="panel-body">
      <center><?php
        if (isset($flash)) {
          echo $flash;
        }
        echo form_open();
      ?>
    </center>
    <?php
      $member_id = $this->uri->segment(3);
      $this->db->select_sum('total_bayar');
      $this->db->from('member');
      $this->db->join('orders', 'member.member_id = orders.member_id');
      $this->db->where('member.member_id = ' . $member_id);
      $this->db->order_by('member.member_id');

      $hasil = $this->db->get();

      $poin_array = array();
      foreach ($hasil->result_array() as $hsl) {
        $poin_array[] = intval($hsl['total_bayar']);
      }
      $total = array_sum($poin_array);
      $poin = $total / 100000;
    ?>
    <?php
      foreach ($hasil->result() as $row) {
        echo "Total Poin dengan Member ID " . $member_id . " ini adalah = <span
style='font-size: 30px'> " . round($poin) . "</span>";
      }
    ?>
    <div class="panel-body">
      <div class="dropdown">
        <select class="form-control col-lg-5" name="level_id">
          <option>-- Pilih Level Member --</option>
          <option value="0">Level 0 -- kurang dari 10 poin</option>
          <option value="1">Silver VIP -- setara 10 poin</option>
          <option value="2">Gold VIP -- setara 30 poin</option>
          <option value="3">Platinum VIP -- setara 90 poin</option>
          <option value="4">Diamond VIP -- setara 270 poin</option>
        </select>
      </div>
    </div>
  </div>
</div>

```

```

        <option value="5">Super VIP -- setara 500 poin</option>
    </select>
</div>
</div>
<button class="btn btn-sm btn-success" href="<?php echo
form_submit('submit', 'Submit'); ?><span class="fa fa-check"></span> Submit</button>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>

```

2. Class manage.php

```

<div id="page-wrapper">
<div class="row">
<div class="col-lg-13">
<p></p>
</div>
</div>
<div class="col-lg-13">
<div class="panel panel-yellow">
<div class="panel-heading">
<div class="row">
<div class="col-xs-12">
<h2>
<span style="color: #666666"><i class="fa fa-comments-o fa-fw"></i>
Manage Member</span>
</h2>
</div>
</div>
</div>
</div>
<div class="panel-footer">
<div class="margin-bottom-40" style="color: #666666">
<div class="panel-body">
</div>
<!--panel body untuk form-->
<div class="panel panel-green margin-bottom-40">
<div class="panel-body">
<center><?php
if (isset($flash)) {
echo $flash;
}
?></center>
<div class="panel-body">
<?php echo Modules::run('adm_member/_display_member_table'); ?>
</div>

```

```

        </div>
    </div>
</div>
<div>
<center>
<?php
    echo "<span style='font-size: 17px'>" . $this->pagination->create_links() . "</span>";
?>
</center>
</div>
</div>

```

3. Class member_table.php

```

<div class="dataTable_wrapper">
    <table class="table table-responsive table-hover" style="font-family: tahoma; font-size:
    12px">
        <tr>
            <th><center>Member ID</center></th>
            <th><center>Nama</center></th>
            <th><center>Email</center></th>
            <th><center>Poin</center></th>
            <th><center>Level Member</center></th>
        </tr>

        <?php
        foreach ($query->result() as $row) {
            ?>
            <tr>
                <td><center><?php echo $row->member_id; ?></center></td>
                <td style="width: 50px"><?php echo $row->nama_depan . " " . $row-
                >nama_belakang; ?></td>
                <td><center><?php echo $row->email; ?></center></td>
            <?php
            $this->db->select_sum('total_bayar');
            $this->db->from('member');
            $this->db->join('orders', 'member.member_id = orders.member_id');
            $this->db->where('member.member_id = ' . $row->member_id);
            $this->db->order_by('member.member_id');

            $hasil = $this->db->get();

            $poin_array = array();
            foreach ($hasil->result_array() as $hsl) {
                $poin_array[] = intval($hsl['total_bayar']);
            }
            $total = array_sum($poin_array);
            $poin = $total / 100000;

```

```

?>
<td><center><?php echo round($poin); ?></center></td>
<td>
<center>
<?php
if ($row->level_id == "1") {
    echo "SILVER VIP";
} else if ($row->level_id == "2") {
    echo "GOLD VIP";
} else if ($row->level_id == "3") {
    echo "PLATINUM VIP";
} else if ($row->level_id == "4") {
    echo "DIAMOND VIP";
} else if ($row->level_id == "5") {
    echo "SUPER VIP";
} else {
    echo "Level 0";
}
?>
<div class="btn-group">
    <a class="btn btn-xs btn-primary" title="Ubah Level Member" href="level/<?php
echo $row->member_id ?>"><span class="fa fa-edit"></span></a>
</div>
</center>
</td>
</tr>

<?php
}
?>
</table>
</div>

```

D.5. Package *adm_order*

D.5.1. Package *controllers*

1. Class *adm_order.php*

```

<?php if (!defined('BASEPATH')) exit('No direct script access allowed');
class Adm_order extends MX_Controller
{

function __construct() {
parent::__construct();
$this->load->model('mdl_order');
}
}

```



```
function konfirmasi($serial){
    $data = $this->get_konfirm_from_db($serial);
    $template = "admin";
    $data['serial'] = $serial;
    $data['view_file'] = "konfirmasi";
    $this->load->module('template');
    $this->template->$template($data);
}

function get_konfirm_from_db($serial){
    $query = $this->get_where_konfirm($serial);
    foreach($query->result() as $row){
        $data['id'] = $row->id;
        $data['email'] = $row->email;
        $data['nama_depan'] = $row->nama_depan;
        $data['serial'] = $row->serial;
        $data['tgl_bayar'] = $row->tgl_bayar;
        $data['jumlah_bayar'] = $row->jumlah_bayar;
        $data['bank_transfer'] = $row->bank_transfer;
        $data['cara_bayar'] = $row->cara_bayar;
        $data['nama_pemilik_rek'] = $row->nama_pemilik_rek;
        $data['bank'] = $row->bank;
    }
    if (!isset($data)){
        $data = "";
    }

    return $data;
}

function status(){
    $serial = $this->uri->segment(3);
    $submit = $this->input->post('submit', TRUE);

    if($submit=="Submit"){
        $data['serial'] = $serial;
        $data['status_order'] = $this->input->post('status_order', TRUE);
        $this->update_status($serial, $data);
        $value = "<div class='alert alert-success alert-dismissable'>
            <button type='button' class='close' data-dismiss='alert' aria-
            hidden='true'>&times;</button>
            Status Order pada nomor order " . $serial . " berhasil diubah.
            </div>";
        $this->session->set_flashdata('item', $value);
        redirect('adm_order/manage');
    }
    $data['serial'] = $serial;
    $template = "admin";
    $data['view_file'] = "status";
    $current_url = current_url();
}
```

```
$data['form_location'] = $current_url;
$this->load->module('template');
$this->template->$template($data);
}

function update_status($serial, $data) {
$this->load->model('mdl_order');
$this->mdl_order->update_status($serial, $data);
}

function edit_status(){
$this->load->model('mdl_order');
$serial = $this->input->post('serial');
$status = $this->input->post('status');
$this->mdl_order->update_status($serial, $status);
redirect('adm_order/manage');
}

function _display_items_table(){
$this->load->library('pagination');

$config['base_url'] = base_url().'mem_order/manage';
$config['total_rows'] = $this->db->get('orders')->num_rows();
$config['per_page'] = 10;
$config['num_links'] = 5;

$this->pagination->initialize($config);
$data['query'] = $this->db->get('orders', $config['per_page'], $this->uri->segment(3));

$data['query'] = $this->db->order_by('serial', 'desc');

$this->load->view('items_table', $data);
}

function manage(){
if(!$this->session->userdata('username')){
redirect(base_url().'adm_login/index');
}else{
$template = "admin";

$flash = $this->session->flashdata('item');
if ($flash!=""){
$data['flash'] = $flash;
}

$data['view_file'] = "manage";
$this->load->module('template');
$this->template->$template($data);
}
}
```

```
function get($order_by) {
    $this->load->model('mdl_order');
    $query = $this->mdl_order->get($order_by);
    return $query;
}

function get_where($serial) {
    $this->load->model('mdl_order');
    $query = $this->mdl_order->get_where($serial);
    return $query;
}

function get_where_konfirm($serial) {
    $this->load->model('mdl_order');
    $query = $this->mdl_order->get_where_konfirm($serial);
    return $query;
}

function get_where_custom($col, $value) {
    $this->load->model('mdl_order');
    $query = $this->mdl_order->get_where_custom($col, $value);
    return $query;
}

function _insert($data) {
    $this->load->model('mdl_order');
    $this->mdl_order->_insert($data);
}

function _update($serial, $data) {
    $this->load->model('mdl_order');
    $this->mdl_order->_update($serial, $data);
}

function _delete($serial) {
    $this->load->model('mdl_order');
    $this->mdl_order->_delete($serial);
}

function get_max() {
    $this->load->model('mdl_order');
    $max_serial = $this->mdl_order->get_max();
    return $max_serial;
}
}
```

D.5.2. Package models

1. Class *mdl_order.php*

```
<?php
```

```
if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class Mdl_order extends CI_Model {

    function __construct() {
        parent::__construct();
    }

    function get_table() {
        $table = "orders";
        return $table;
    }

    function update_status($serial, $data) {
        $this->db->where('serial', $serial);
        $this->db->update('orders', $data);
    }

    function get_member() {
        $query = $this->db->query
        (
            '
            SELECT
            *
            FROM
            orders o
            , member m
            WHERE
            o.member_id = m.member_id
            ');

        $hasil = $query->result_array();

        return $hasil;
    }

    function get($order_by) {
        $table = $this->get_table();
        $this->db->order_by($order_by);
        $query = $this->db->get($table);
        return $query;
    }

    function get_where($serial) {
        $table = $this->get_table();
        $this->db->where('serial', $serial);
        $query = $this->db->get($table);
        return $query;
    }
}
```

```

function get_where_konfirm($serial) {
    $this->db->where('serial', $serial);
    $query = $this->db->get('konfirmasi_bayar');
    return $query;
}

function get_where_custom($col, $value) {
    $table = $this->get_table();
    $this->db->where($col, $value);
    $query = $this->db->get($table);
    return $query;
}

function _insert($data) {
    $table = $this->get_table();
    $this->db->insert($table, $data);
}

function _delete($serial) {
    $table = $this->get_table();
    $this->db->where('serial', $serial);
    $this->db->delete($table);
}

function get_max() {
    $table = $this->get_table();
    $this->db->select_max('serial');
    $query = $this->db->get($table);
    $row = $query->row();
    $serial = $row->serial;
    return $serial;
}
}

```

D.5.3. Package views

1. Class *items_table.php*

```

<div class="dataTable_wrapper">
  <table border="0" class="table table-responsive table-hover" style="font-family: tahoma;
font-size: 10px">
    <tr>
      <th><center>No.</center></th>
      <th><center>No. Order</center></th>
      <th><center>Tanggal</center></th>
      <th><center>Nama Member</center></th>
      <th><center>Subtotal</center></th>
      <th><center>Ongkir</center></th>

```

```

<th><center>Total Bayar</center></th>
<th><center>Konfirmasi Bayar</center></th>
<th colspan="2"><center>Status Order</center></th>
</tr>
<?php
$this->db->select('*');
$this->db->from('orders');
$this->db->join('member', 'orders.member_id = member.member_id');
$query = $this->db->get();
$count = 0;
foreach ($query->result() as $row) {
    $count++;
    ?>
    <tr>
        <td><center><?php echo $count; ?></center></td>
        <?php echo form_hidden('serial', $row->serial); ?>
        <td><center><?php echo $row->serial; ?></center></td>
        <td><center><?php echo $row->tgl; ?></center></td>
        <td style="width: 200px;"><center><?php echo $row->nama_depan; ?> <?php echo
$row->nama_belakang; ?></center></td>
        <td><center><?php echo $row->subtotal; ?></center></td>
        <td><center><?php echo $row->total_ongkir; ?></center></td>
        <td><center><?php echo $row->total_bayar; ?></center></td>
        <td><center>
            <?php
            $this->db->select('*');
            $this->db->from('konfirmasi_bayar');
            $this->db->join('orders', 'orders.serial = konfirmasi_bayar.serial');
            $this->db->where('serial', $row->serial);

            if ($row->id == 0) {
                echo "Menunggu Konfirmasi Bayar";
            } else {
                ?>
                <a class="btn btn-sm btn-info" href="<?php echo base_url(); ?><?php echo
'adm_order/konfirmasi/' . $row->serial; ?>">Konfirmasi Bayar</a>
                <?php
                }
                ?>
            </center></td>
        <td style="text-align: right">
        <center><?php echo $row->status_order; ?></center>
        </td>
        <td style="text-align: left">
        <center><a class="btn btn-sm btn-primary" href="<?php echo base_url(); ?><?php echo
'adm_order/status/' . $row->serial; ?>">Update Status</a></center>
        </td>
    </tr>
    <?php
}
?>

```

```
</table>
</div>
```

2. Class konfirmasi.php

```
<div id="page-wrapper">
  <div class="row">
    <div class="col-lg-13">
      <p></p>
    </div>
  </div>
  <div class="col-lg-13">
    <div class="panel panel-purple">
      <div class="panel-heading">
        <div class="row">
          <div class="col-xs-12">
            <h2>
              <span style="color: #666666"><i class="fa fa-wrench fa-fw"></i> Manage
Orders</span>
            </h2>
          </div>
        </div>
      </div>
      <div class="panel-footer">
        <div class="margin-bottom-40" style="color: #666666">
          <div class="panel-body">
            <div class="col-xs-12 text-right">
              <a class="btn btn-sm btn-warning" title="Kembali" href="<?php echo
base_url(); ?>adm_order/manage">
                <span class="fa fa-arrow-left"></span> Back to Manage
              </a>
            </div>
          </div>
        </div>
      </div>
      <div class="panel panel-green margin-bottom-40">
        <div class="panel-body">
          <center><?php
            if (isset($flash)) {
              echo $flash;
            }
          ?></center>
          <div class="panel-body">
            <table>
              <tr>
                <td style="width: 300px"><b>Email</b></td>
                <td><?php echo $email; ?></td>
              </tr>
              <tr>
                <td style="width: 300px"><b>Nama</b></td>
                <td><?php echo $nama_depan; ?></td>
              </tr>
            </table>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```

<tr>
  <td style="width: 300px"><b>No. Order</b></td>
  <td><?php echo $serial; ?></td>
</tr>
<tr>
<tr>
  <td style="width: 300px"><b>Tanggal Pembayaran</b></td>
  <td><?php echo $tgl_bayar; ?></td>
</tr>
<tr>
<tr>
  <td style="width: 300px"><b>Jumlah yang Dibayarkan</b></td>
  <td><?php echo $jumlah_bayar; ?></td>
</tr>
<tr>
<tr>
  <td style="width: 300px"><b>Dana Ditransfer Ke</b></td>
  <td><?php echo $bank_transfer; ?></td>
</tr>
<tr>
<tr>
  <td style="width: 300px"><b>Cara Pembayaran</b></td>
  <td><?php echo $cara_bayar; ?></td>
</tr>
<tr>
<tr>
  <td style="width: 300px"><b>Nama Pemilik Rekening</b></td>
  <td><?php echo $nama_pemilik_rek; ?></td>
</tr>
<tr>
<tr>
  <td style="width: 300px"><b>Nama Bank Pengirim</b></td>
  <td><?php echo $bank; ?></td>
</tr>
</table>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>

```

3. *Class manage.php*

```

<div id="page-wrapper">
  <div class="row">
    <div class="col-lg-13">
      <p></p>
    </div>
  </div>
  <div class="col-lg-13">
    <div class="panel panel-purple">
      <div class="panel-heading">
        <div class="row">

```



```

        <div class="col-xs-12">
            <h2>
                <span style="color: #666666"><i class="fa fa-wrench fa-fw"></i> Manage
Orders</span>
            </h2>
        </div>
    </div>
</div>
<div class="panel-footer">
    <div class="margin-bottom-40" style="color: #666666">
        <div class="panel panel-green margin-bottom-40">
            <div class="panel-body">
                <center><?php
                    if (isset($flash)) {
                        echo $flash;
                    }
                ?></center>

                <div class="panel-body">
                    <?php echo Modules::run('adm_order/_display_items_table'); ?>
                </div>
            </div>
        </div>
    </div>
</div>
<div>
    <center>
        <?php
            echo "<span style='font-size: 17px'" . $this->pagination->create_links() . "</span>";
        ?>
    </center>
</div>
</div>

```

4. *Class status.php*

```

<div id="page-wrapper">
    <div class="row">
        <div class="col-lg-13">
            <p></p>
        </div>
    </div>
    <div class="col-lg-13">
        <div class="panel panel-purple">
            <div class="panel-heading">
                <div class="row">
                    <div class="col-xs-12">
                        <h2>

```

```

        <span style="color: #666666"><i class="fa fa-wrench fa-fw"></i> Manage
Orders</span>
    </h2>
</div>
</div>
</div>
<div class="panel-footer">
    <div class="margin-bottom-40" style="color: #666666">
        <div class="panel panel-green margin-bottom-40">
            <div class="panel-body">
                <center><?php
                    if (isset($flash)) {
                        echo $flash;
                    }
                    echo form_open();
                ?></center>
                <div class="panel-body">
                    <div class="dropdown">
                        <select class="form-control col-lg-5" name="status_order">
                            <option>-- Pilih Status Order --</option>
                            <option value="Pembayaran diterima">Pembayaran diterima</option>
                            <option value="Proses Packing">Proses packing</option>
                            <option value="Proses pengiriman">Proses pengiriman</option>
                        </select>
                    </div>
                </div>
                <button class="btn btn-sm btn-success" href="<?php echo
form_submit('submit', 'Submit'); ?><span class="fa fa-check"></span> Submit</button>
            </div>
        </div>
    </div>
</div>
</div>
</div>
</div>
</div>

```

D.6. *Package adm_produk*

D.6.1. *Package controllers*

1. *Class adm_produk.php*

```

<?php
if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class Adm_produk extends MX_Controller {

    function __construct() {
        parent::__construct();
    }
}

```

```
}

function do_upload($produk_id) {
    Modules::run('web_security/check_admin');

    $config['upload_path'] = './img/imgproduk/';
    $config['allowed_types'] = 'gif|jpg|png';
    $config['max_size'] = '1000';
    $config['max_width'] = '2024';
    $config['max_height'] = '2768';

    $this->load->library('upload', $config);

    if (!$this->upload->do_upload()) {
        $data['error'] = array('error' => $this->upload->display_errors("<div class='alert alert-
danger alert-dismissible'>
        <button type='button' class='close' data-dismiss='alert' aria-
hidden='true'>&times;</button>", "</div>"));
        $data['produk_id'] = $produk_id;
        $template = "admin";
        $data['view_file'] = "upload_pic";
        $this->load->module('template');
        $this->template->$template($data);
    } else {
        $data = $this->upload->data();
        $file_name = $data['file_name']; //nama file yg sudah diupload
        //thumbnail gambar
        $config['image_library'] = 'gd2';
        $config['source_image'] = './img/imgproduk/' . $file_name;
        $config['create_thumb'] = TRUE;
        $config['maintain_ratio'] = TRUE;
        $config['width'] = 200;
        $config['height'] = 200;

        $this->load->library('image_lib', $config);
        $this->image_lib->resize();

        //resize gambar (350px x 350px)
        $new_width = 350;
        $new_height = 350;
        $this->_resize_pic($file_name, $new_width, $new_height);

        //update the database
        $raw_file_name = $data['raw_name'];
        $file_ext = $data['file_ext'];

        unset($data);
        $data['gbr_thumb'] = $raw_file_name . "_thumb" . $file_ext;
        $data['gbr_besar'] = $file_name;
        $this->_update($produk_id, $data);
    }
}
```

```
//divert to a success page
redirect("adm_produk/upload_pic/" . $produk_id);
}
}

function _resize_pic($file_name, $new_width, $new_height) {
    Modules::run('web_security/check_admin');

    $config['image_library'] = 'gd2';
    $config['source_image'] = './img/imgproduk/' . $file_name;
    $config['create_thumb'] = FALSE;
    $config['maintain_ratio'] = TRUE;
    $config['width'] = $new_width;
    $config['height'] = $new_height;

    $this->image_lib->initialize($config);

    $this->load->library('image_lib', $config);
    $this->image_lib->resize();
}

function upload_pic($produk_id) {
    if (!$this->session->userdata('username')) {
        redirect(base_url() . 'admin/index');
    } else {
        $query = $this->get_where($produk_id);
        foreach ($query->result() as $row) {
            $data['produk_id'] = $produk_id;
            $data['gbr_thumb'] = $row->gbr_thumb;
        }

        $data['produk_id'] = $produk_id;
        $template = "admin";
        $data['view_file'] = "upload_pic";
        $this->load->module('template');
        $this->template->$template($data);
    }
}

function delete_produk($update_id) {
    if (!$this->session->userdata('username')) {
        redirect(base_url() . 'admin/index');
    } else {
        $submit = $this->input->post('submit', TRUE);
        if ($submit == "No") {
            redirect('adm_produk/manage');
        }
        if ($submit == "Yes") {
            //delete produk
            $this->_delete($update_id);
        }
    }
}
```

```
//add flashdata
$value = "<div class='alert alert-danger alert-dismissible'>
    <button type='button' class='close' data-dismiss='alert' aria-
hidden='true'>&times;</button>
    Data produk berhasil dihapus.
</div>";
$this->session->set_flashdata('item', $value);

    redirect('adm_produk/manage');
}

$data['produk_id'] = $update_id;
$template = "admin";
$current_url = current_url();
$data['form_location'] = $current_url;
$data['view_file'] = "delete_conf";
$this->load->module('template');
$this->template->$template($data);
}
}

function get_data_from_post() {
    $data['nama_produk'] = $this->input->post('nama_produk', TRUE);
    $data['harga_produk'] = $this->input->post('harga_produk', TRUE);
    $data['ukuran_produk'] = $this->input->post('ukuran_produk', TRUE);
    $data['bahan_produk'] = $this->input->post('bahan_produk', TRUE);
    $data['berat_produk'] = $this->input->post('berat_produk', TRUE);
    $data['gbr_thumb'] = $this->input->post('gbr_thumb', TRUE);
    $data['gbr_besar'] = $this->input->post('gbr_besar', TRUE);
    $data['produk_url'] = $this->input->post('produk_url', TRUE);
    return $data;
}

function get_data_from_db($update_id) {
    $query = $this->get_where($update_id);
    foreach ($query->result() as $row) {
        $data['produk_id'] = $row->produk_id;
        $data['nama_produk'] = $row->nama_produk;
        $data['harga_produk'] = $row->harga_produk;
        $data['ukuran_produk'] = $row->ukuran_produk;
        $data['bahan_produk'] = $row->bahan_produk;
        $data['berat_produk'] = $row->berat_produk;
        $data['gbr_thumb'] = $row->gbr_thumb;
        $data['gbr_besar'] = $row->gbr_besar;
        $data['produk_url'] = $row->produk_url;
    }
    if (!isset($data)) {
        $data = "";
    }

    return $data;
}
```

```
}

function create() {
    if (!$this->session->userdata('username')) {
        redirect(base_url() . 'admin/index');
    } else {
        $produk_id = $this->uri->segment(3);
        $data = $this->get_data_from_post();
        $submit = $this->input->post('submit', TRUE);

        if ($produk_id > 0) {

            if ($submit != "Submit") {
                //ambil data dr db
                $data = $this->get_data_from_db($produk_id);
            }

            $data['headline'] = "Update Product";
        } else {
            $data['headline'] = "Add Product";
        }

        $current_url = current_url();
        $data['form_location'] = str_replace('/create', '/submit', $current_url);

        $flash = $this->session->flashdata('item');
        if ($flash != "") {
            $data['flash'] = $flash;
        }

        $data['produk_id'] = $produk_id;
        $template = "admin";
        $data['view_file'] = "create";
        $this->load->module('template');
        $this->template->$template($data);
    }
}

function submit() {
    //untuk data yang telah disubmit
    $this->load->helper(array('form', 'url'));

    $this->load->library('form_validation');
    $this->form_validation->set_rules('nama_produk', 'Nama Produk', 'required');
    $this->form_validation->set_rules('harga_produk', 'Harga Produk', 'is_numeric|required');

    if ($this->form_validation->run() == FALSE) {
        $this->create();
    } else {
        $update_id = $this->uri->segment(3);
    }
}
```

```
if ($update_id > 0) {
    //untuk edit, bukan create
    $data = $this->get_data_from_post();
    $data['produk_url'] = url_title($data['nama_produk']);
    $this->_update($update_id, $data);
    $value = "<div class='alert alert-success alert-dismissable'>
        <button type='button' class='close' data-dismiss='alert' aria-
hidden='true'>&times;</button>
        Data produk berhasil diubah.
    </div>";
} else {
    //tambah produk
    $data = $this->get_data_from_post();
    $data['produk_url'] = url_title($data['nama_produk']);
    $this->_insert($data);
    $value = "<div class='alert alert-success alert-dismissable'>
        <button type='button' class='close' data-dismiss='alert' aria-
hidden='true'>&times;</button>
        Data produk berhasil ditambahkan.
    </div>";
    $update_id = $this->get_max();
}

//flashdata
//session data yg hanya untuk server request itu sendiri,
//selebihnya akan hilang jika page di refresh
//ini untuk info/status messages
$this->session->set_flashdata('item', $value);

redirect('adm_produk/create/" . $update_id);
}
}

function _display_items_table() {
    $this->load->library('pagination');

    $config['base_url'] = base_url() . 'adm_produk/manage';
    $config['total_rows'] = $this->db->get('data_produk')->num_rows();
    $config['per_page'] = 10;
    $config['num_links'] = 5;

    $this->pagination->initialize($config);
    $data['query'] = $this->db->get('data_produk', $config['per_page'], $this->uri->segment(3));

    $this->load->view('items_table', $data);
}

function manage() {
    if (!$this->session->userdata('username')) {
```

```
        redirect(base_url() . 'admin/index');
    } else {
        $template = "admin";

        $flash = $this->session->flashdata('item');
        if ($flash != "") {
            $data['flash'] = $flash;
        }

        $data['view_file'] = "manage";
        $this->load->module('template');
        $this->template->$template($data);
    }
}

function get($order_by) {
    $this->load->model('mdl_produk');
    $query = $this->mdl_produk->get($order_by);
    return $query;
}

function get_with_limit($limit, $offset, $order_by) {
    $this->load->model('mdl_produk');
    $query = $this->mdl_produk->get_with_limit($limit, $offset, $order_by);
    return $query;
}

function get_where($id) {
    $this->load->model('mdl_produk');
    $query = $this->mdl_produk->get_where($id);
    return $query;
}

function get_where_level($id) {
    $this->load->model('mdl_produk');
    $query = $this->mdl_produk->get_where_level($id);
    return $query;
}

function get_where_custom($col, $value) {
    $this->load->model('mdl_produk');
    $query = $this->mdl_produk->get_where_custom($col, $value);
    return $query;
}

function _insert($data) {
    $this->load->model('mdl_produk');
    $this->mdl_produk->_insert($data);
}

function _update($id, $data) {
```



```

$this->load->model('mdl_produk');
$this->mdl_produk->_update($id, $data);
}

function _delete($id) {
    $this->load->model('mdl_produk');
    $this->mdl_produk->_delete($id);
}

function count_where($column, $value) {
    $this->load->model('mdl_produk');
    $count = $this->mdl_produk->count_where($column, $value);
    return $count;
}

function get_max() {
    $this->load->model('mdl_produk');
    $max_id = $this->mdl_produk->get_max();
    return $max_id;
}

function _custom_query($mysql_query) {
    $this->load->model('mdl_produk');
    $query = $this->mdl_produk->_custom_query($mysql_query);
    return $query;
}
}

```

D.6.2. Package models

1. Class *mdl_produk.php*

```

<?php

if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class Mdl_produk extends CI_Model {

    function __construct() {
        parent::__construct();
    }

    function get_by_kategori() {
        $this->db->distinct();
        $this->db->select('kategori.nama_kategori');
        $this->db->from('kategori_produk');
        $this->db->join('data_produk', 'kategori_produk.produk_id=data_produk.produk_id');
    }
}

```

```
$this->db->join('kategori', 'kategori_produk.kategori_id=kategori.kategori_id');

$query = $this->db->get();
return $query->result();
}

function get($order_by) {
    $this->db->order_by($order_by);
    $query = $this->db->get('data_produk');

    return $query;
}

function get_with_limit($limit, $offset, $order_by) {
    $this->db->limit($limit, $offset);
    $this->db->order_by($order_by);
    $query = $this->db->get('data_produk');
    return $query;
}

function get_where_level($level_id) {
    $this->db->where('level_id', $level_id);
    $query = $this->db->get('level_member');

    return $query;
}

function get_where($produk_id) {
    $this->db->where('produk_id', $produk_id);
    $query = $this->db->get('data_produk');

    return $query;
}

function get_where_custom($col, $value) {
    $this->db->where($col, $value);
    $query = $this->db->get('data_produk');
    return $query;
}

function _insert($data) {
    $this->db->insert('data_produk', $data);
}

function _update($produk_id, $data) {
    $this->db->where('produk_id', $produk_id);
    $this->db->update('data_produk', $data);
}

function _delete($produk_id) {
    $this->db->where('produk_id', $produk_id);
```

```

$this->db->delete('data_produk');
}

function count_where($column, $value) {
    $this->db->where($column, $value);
    $query = $this->db->get('data_produk');
    $num_rows = $query->num_rows();
    return $num_rows;
}

function count_all() {
    $query = $this->db->get('data_produk');
    $num_rows = $query->num_rows();
    return $num_rows;
}

function get_max() {
    $this->db->select_max('produk_id');
    $query = $this->db->get('data_produk');
    $row = $query->row();
    $produk_id = $row->produk_id;
    return $produk_id;
}

function _custom_query($mysql_query) {
    $query = $this->db->query($mysql_query);
    return $query;
}
}

```

D.6.3. Package views

1. Class *additional_options.php*

```

<a class="btn btn-sm btn-info" title="Ubah Warna" href="<?php echo base_url();?><?php echo
'adm_warna_produk/update/'.$produk_id; ?>">
    <span class="glyphicon glyphicon-tint"></span> Update Colors
</a>
<a class="btn btn-sm btn-default" title="Gambar Produk" href="<?php echo base_url();?><?php
echo 'adm_produk/upload_pic/'.$produk_id; ?>">
    <span class="glyphicon glyphicon-picture"></span> Upload Pic
</a>
<a class="btn btn-sm btn-primary" title="Kategori Produk" href="<?php echo base_url();?><?php
echo 'adm_kategori_produk/v_kategori_produk/'.$produk_id; ?>">
    <span class="glyphicon glyphicon-list"></span> Kategori Produk
</a>

```

2. Class *create.php*

```

<head>

```

```

<title>Admin Panel</title>
</head>
<div id="page-wrapper">
  <div class="row">
    <div class="col-lg-13">
      <p></p>
    </div>
  </div>
  <div class="col-lg-13">
    <div class="panel panel-green">
      <div class="panel-heading">
        <div class="row">
          <div class="col-xs-12">
            <h2>
              <span style="color: #666666"><i class="fa fa-briefcase fa-fw"></i><?php
echo $headline; ?></span>
            </h2>
          </div>
        </div>
      </div>
      <div class="panel-footer">
        <div class="margin-bottom-40" style="color: #666666">
          <div class="panel-body">
            <div class="col-xs-12 text-right">
              <center>
                <?php
                if (isset($flash)) {
                  echo $flash;
                }
                echo validation_errors("<p style='color: red; background-color: white'>",
"</p>");
              </center>
              <a class="btn btn-sm btn-warning" title="Kembali" href="<?php echo
base_url(); ?>adm_produk/manage">
                <span class="glyphicon glyphicon-chevron-left"></span> Back to Manage
              </a>
              <a class="btn btn-sm btn-success" title="Kembali" href="<?php echo
base_url(); ?>adm_produk/create">
                <span class="glyphicon glyphicon-plus"></span> Tambah Produk
              </a>
              <?php
              if ($produk_id > 0) {
                include('additional_options.php');
              }
            </div>
          </div>
        </div>
      </div>
      <!--panel body untuk form-->
      <div class="panel panel-green margin-bottom-40">
        <div class="panel-body">

```

```

<?php
echo form_open($form_location);
?>
<form method="POST" action="<?php echo base_url();
?>adm_produk/submit">
  <div class="clearfix"></div>
  <?php
  if($produk_id>0){
    ?>
    <div class="form-group">
      <label for="produk_id" class="col-lg-2 control-label">ID Produk</label>
      <span style="font-size: 20px"><?php echo $produk_id; ?></span>
    </div>
    <?php
  }else{
    echo form_hidden('produk_id', $produk_id);
  }
  ?>
  <div class="clearfix"></div>
  <div class="form-group">
    <label for="nama_produk" class="control-label">Nama Produk</label>
    <input type="text" name="nama_produk" value="<?php echo
$nama_produk; ?>" class="form-control" id="nama_produk" required />
  </div>
  <div class="form-group">
    <label for="harga_produk" class="control-label">Harga Produk
(Rp)</label>
    <input type="text" name="harga_produk" value="<?php echo
$harga_produk; ?>" class="form-control" id="harga_produk" required />
  </div>
  <div class="form-group">
    <label for="ukuran_produk" class="control-label">Ukuran Produk
(mm)</label>
    <input type="text" name="ukuran_produk" value="<?php echo
$ukuran_produk; ?>" class="form-control" id="ukuran_produk" placeholder="ex.: 40x50x30"
required />
  </div>
  <div class="form-group">
    <label for="bahan_produk" class="control-label">Bahan Produk</label>
    <input type="text" name="bahan_produk" value="<?php echo
$bahan_produk; ?>" class="form-control" id="bahan_produk">
  </div>
  <div class="form-group">
    <label for="berat_produk" class="control-label">Berat Produk
(gr)</label>
    <input type="text" name="berat_produk" value="<?php echo
$berat_produk; ?>" class="form-control" id="berat_produk">
  </div>
  <div class="clearfix"></div>
  <button class="btn btn-sm btn-warning" type="reset" value="Clear"><span
class="glyphicon glyphicon-cloud"></span> Reset</button>

```

```

        <button class="btn btn-sm btn-success" type="submit"><span
class="glyphicon glyphicon-check"></span> Submit</button>
    </form>
</div>
</div>
<?php
echo form_close();
?>
</div>
</div>
</div>
</div>
</div>
</div>

```

3. *Class delete_conf.php*

```

<div id="page-wrapper">
<div class="row">
<div class="col-lg-13">
<p></p>
</div>
</div>

<div class="col-lg-13">
<div class="panel panel-yellow">
<div class="panel-heading">
<div class="row">
<div class="col-xs-12">
<h2>
<i class="fa fa-wrench fa-fw" style="color: #666666"></i> <span style="color:
#666666">Delete</span>
</h2>
</div>
</div>
</div>
<div class="panel-footer" style="background-color: #E6A8D7">
<div class="margin-bottom-40" style="color: #666666">
<!--panel body untuk form-->
<div class="panel panel-green margin-bottom-40">
<div class="panel-body">
<p>Apakah Anda yakin ingin menghapus data?</p>
<?php echo form_open($form_location); ?>
<button class="btn btn-sm btn-info" href="<?php echo form_submit('submit',
'Yes'); ?> <span class="glyphicon glyphicon-ok"></span> Yes</button>
<button class="btn btn-sm btn-danger" href="<?php echo
form_submit('submit', 'No'); ?> <span class="glyphicon glyphicon-remove"></span>
No</button>
<?php echo form_close(); ?>
</div>

```

```

        </div>
    </div>
</div>
</div>
</div>

```

4. *Class items_table.php*

```

<div class="dataTable_wrapper">
  <table class="table table-responsive table-hover" style="font-family: tahoma; font-size:
  10px">
    <tr>
      <th><center>No.</center></th>
      <th><center>Gambar Utama</center></th>
      <th><center>Nama Produk</center></th>
      <th><center>Harga Satuan</center></th>
      <th><center>Ukuran</center></th>
      <th><center>Warna</center></th>
      <th><center>Bahan</center></th>
      <th><center>Berat</center></th>
      <th><center>Action</center></th>
    </tr>
    <?php
    $count = 0;
    foreach ($query->result() as $row) {
      $count++;
      ?>
      <tr>
        <td><center><?php echo $count; ?></center></td>
        <td>
          <center>
            
          </center>
        </td>
        <td style="width: 200px;"><?php echo $row->nama_produk; ?></td>
        <td><center><?php echo $row->harga_produk; ?></center></td>
        <td><center><?php echo $row->ukuran_produk; ?></center></td>
        <td>
          <center>
            <a class="btn btn-xs btn-info" href="<?php echo base_url(); ?><?php echo
            'adm_warna_produk/update/' . $row->produk_id; ?>">Pilihan Warna</a>
          </center>
        </td>
        <td><?php echo $row->bahan_produk; ?></td>
        <td><center><?php echo $row->berat_produk; ?></center></td>
        <td>

```

```

<center>
  <div class="btn-group">
    <a class="btn btn-xs btn-primary" title="Edit" href="<?php echo base_url();
?>adm_produk/create/<?php echo $row->produk_id ?>"><span class="glyphicon glyphicon-
edit"></span></a>
    <a class="btn btn-xs btn-danger" title="Delete" href="<?php echo base_url();
?>adm_produk/delete_produk/<?php echo $row->produk_id ?>"><span class="glyphicon
glyphicon-trash"></span></a>
  </div>
</center>
</td>
</tr>

<?php
}
?>
</table>
</div>

```

5. *Class manage.php*

```

<div id="page-wrapper">
  <div class="row">
    <div class="col-lg-13">
      <p></p>
    </div>
  </div>
  <div class="col-lg-13">
    <div class="panel panel-yellow">
      <div class="panel-heading">
        <div class="row">
          <div class="col-xs-12">
            <h2>
              <span style="color: #666666"><i class="fa fa-wrench fa-fw"></i> Manage
Products</span>
            </h2>
          </div>
        </div>
      </div>
      <div class="panel-footer">
        <div class="margin-bottom-40" style="color: #666666">
          <div class="panel-body">
            <div class="col-xs-12 text-right">
              <a class="btn btn-sm btn-success" title="Kembali" href="<?php echo
base_url(); ?>adm_produk/create">
                <span class="glyphicon glyphicon-plus"></span> Tambah Produk
              </a>
            </div>
          </div>
        </div>
      </div>
    <!--panel body untuk form-->

```



```

<div class="panel panel-green margin-bottom-40">
  <div class="panel-body">
    <center><?php
      if (isset($flash)) {
        echo $flash;
      }
    ?>
    </center>
    <div class="panel-body">
      <?php echo Modules::run('adm_produk/_display_items_table'); ?>
    </div>
  </div>
</div>
</div>
</div>
</div>
</div>
<center>
  <?php
    echo "<span style='font-size: 17px'>" . $this->pagination->create_links() . "</span>";
  ?>
</center>
</div>
</div>

```

6. *Class upload_pic.php*

```

<div id="page-wrapper">
  <div class="row">
    <div class="col-lg-13">
      <p></p>
    </div>
  </div>
  <div class="col-lg-13">
    <div class="panel panel-red">
      <div class="panel-heading">
        <div class="row">
          <div class="col-xs-12">
            <h2>
              <span style="color: #666666"><i class="fa fa-picture-o fa-fw"></i> Upload
              Product Picture</span>
            </h2>
          </div>
        </div>
      </div>
      <div class="panel-footer">
        <div class="margin-bottom-40" style="color: #666666">
          <div class="panel-body">
            <div class="col-xs-12 text-right">
              <center><?php
                if (isset($error)) {

```

```

        foreach ($error as $fault) {
            echo $fault;
        }
    }
    ?></center>
    <a class="btn btn-sm btn-warning" title="Kembali" href="<?php echo
base_url(); ?><?php echo 'adm_produk/create/' . $produk_id; ?>">
    <span class="glyphicon glyphicon-chevron-left"></span> Back to Update
    Produk
    </a>
</div>
</div>
<div class="panel panel-yellow margin-bottom-40">
    <div class="panel-body">
        <table border="0" width="970">
            <tr valign="top" >
                <td width="400">
                    <span style="color: #666666">
                        <b>Pilih file gambar:</b>
                    </span>
                    <br/><br/>
                    <div class="form-inline">
                        <div class="form-group">
                            <?php echo form_open_multipart('adm_produk/do_upload/' .
$produk_id); ?>
                            <input type="file" class="btn btn-sm btn-info" name="userfile"
size="25" />
                            </div>
                            <button class="btn btn-sm btn-success" type="submit"><span
class="glyphicon glyphicon-upload"></span> Upload</button>
                        </div>
                        <h6 style="color: #eb6864">size gambar minimal 400 x 400 px</h6>
                        <br />
                    </td>
                    <td>
                        <center><h4 style="color: #666666"><u>Gambar Produk</u></h4>
                        <br />
                        <?php
                        if (isset($gbr_thumb)) {
                            $pic_path = base_url() . "img/imgproduk/" . $gbr_thumb;
                            echo "<p>";
                            echo "<img src=\"" . $pic_path . "\">";
                            echo "<p>";
                        }
                    <?>
                    </center>
                </td>
            </tr>
        </table>
    </div>
</div>

```

```
</div>
</div>
</div>
</div>
</div>
```

D.7. Package member

D.7.1. Package controllers

1. Class member.php

```
<?php
if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class Member extends MX_Controller {

    function __construct() {
        parent::__construct();
        session_start();
        $this->load->model('mdl_member', '', TRUE);
    }

    function login() {
        $data = $this->get_data_from_post();
        $submit = $this->input->post('submit', TRUE);
        $current_url = current_url();
        $data['form_location'] = str_replace('/login', '/submit', $current_url);

        $template = "public_one_col";
        $data['view'] = "login";
        $this->load->module('template');
        $this->load->template->$template($data);
    }

    function register() {
        $member_id = $this->uri->segment(3);
        $data = $this->get_data_from_post();
        $submit = $this->input->post('submit', TRUE);
        $data['kota'] = $this->mdl_member->get_kota($this->uri->segment(3));

        $this->load->view('register', $data);
    }

    function process() {
        $this->load->model('mdl_member');
        $query = $this->mdl_member->validate();

        if ($query) {
```

```
$member_id = $this->mdl_member->get_userID($this->input->post('username'));
$level_id = $this->mdl_member->get_level($this->input->post('username'));

$data = array(
    'username' => $this->input->post('username'),
    'member_id' => $member_id
);

$this->session->set_userdata($data);
$_SESSION['username'] = $username['username'];
redirect('member/account');
} else {
    $msg = "<div class='alert alert-danger alert-dismissible'>
        <button type='button' class='close' data-dismiss='alert' aria-
hidden='true'>&times;</button>
        Invalid username or password. Please try again.</div>";
    $this->login();
}
}

public function do_logout() {
// berhasil destroy username & member_id tanpa destroy cart,
// tp cart ttp ada meski user lain yg login

    $this->session->sess_destroy();
    redirect('mem_homepage/landing_page');
}

function order_saya() {
    $member_id = $this->session->userdata('member_id');
    $this->load->library('pagination');

    $config['base_url'] = base_url() . 'member/order_saya';
    $config['total_rows'] = $this->db->get('data_produk')->num_rows();
    $config['per_page'] = 10;
    $config['num_links'] = 5;

    $this->pagination->initialize($config);
    $data['query'] = $this->db->get('orders', $config['per_page'], $this->uri->segment(3));

    $template = "member_account";
    $data['view_file'] = "order_saya";
    $this->load->module('template');
    $this->template->$template($data);
}

function order_detail() {
    $order_id = $this->uri->segment(3);

    $this->get_data_order($order_id);
```

```
$current_url = current_url();
$data['form_location'] = str_replace('/update', '/submit', $current_url);

$data['order_id'] = $order_id;
$template = "member_account";
$data['view_file'] = "order_details";
$this->load->module('template');
$this->load->template->$template($data);
}

function order_details() {
    $order_id = $this->uri->segment(3);
    $data = $this->get_data_from_post();
    $submit = $this->input->post('submit', TRUE);

    if ($submit != "Submit") {
        //ambil data dr db
        $data = $this->get_data_order($order_id);
    }

    $current_url = current_url();
    $data['form_location'] = str_replace('/create', '/submit', $current_url);

    $flash = $this->session->flashdata('item');
    if ($flash != "") {
        $data['flash'] = $flash;
    }

    $data['order_id'] = $order_id;
    $template = "member_account";
    $data['view_file'] = "order_details";
    $this->load->module('template');
    $this->template->$template($data);
}

function account() {
    if (!$this->session->userdata('username')) {
        redirect(base_url() . 'member/login');
    } else {
        $member_id = $this->session->userdata('member_id');

        $template = "member_account";
        $data['view_file'] = "account";
        $this->load->module('template');
        $this->template->$template($data);
    }
}

function _display_items_table() {
    $this->load->library('pagination');
```

```
$config['base_url'] = base_url() . 'adm_member/manage';
$config['total_rows'] = $this->db->get('member')->num_rows();
$config['per_page'] = 10;
$config['num_links'] = 5;

$this->pagination->initialize($config);
$data['query'] = $this->db->get('member', $config['per_page'], $this->uri->segment(3));

$this->load->view('items_table', $data);
}

function manage() {
    $template = "admin";

    $flash = $this->session->flashdata('item');
    if ($flash != "") {
        $data['flash'] = $flash;
    }

    $data['view_file'] = "manage";
    $this->load->module('template');
    $this->template->$template($data);
}

function update() {
    $member_id = $this->uri->segment(3);
    $member_id = $this->mdl_member->get_userID($this->input->post('username'));

    $data = $this->get_data_from_post();
    $submit = $this->input->post('submit', TRUE);

    if ($submit != "Submit") {
        $data = $this->get_data_from_db($member_id);
    }
    $data['headline'] = "Ubah Profil";

    $current_url = current_url();
    $data['form_location'] = str_replace('/update', '/submit', $current_url);

    $data['kota'] = $this->mdl_member->get_kota($this->uri->segment(3));

    $template = "member_account";
    $data['view_file'] = "update";
    $this->load->module('template');
    $this->load->template->$template($data);
}

function submit() {
    $this->load->helper(array('form', 'url'));

    $member_id = $this->mdl_member->get_userID($this->input->post('username'));
```

```
if ($member_id > 0) {
    //untuk edit, bukan create
    $data = $this->get_data_from_post();
    $this->_update($member_id, $data);
    $value = "<div class='alert alert-success alert-dismissable'>
        <button type='button' class='close' data-dismiss='alert' aria-
hidden='true'>&times;</button>
        Data member berhasil diubah.
    </div>";

    $query = $this->mdl_member->validate();
    $member_id = $this->mdl_member->get_userID($this->input->post('username'));

    $data = array(
        'username' => $this->input->post('username'),
        'member_id' => $member_id
    );
    $this->session->set_userdata($data);
    redirect('member/update/');
} else { //tambah
    $data = $this->get_data_from_post();
    $this->_insert($data);
    $value = "<div class='alert alert-success alert-dismissable'>
        <button type='button' class='close' data-dismiss='alert' aria-
hidden='true'>&times;</button>
        Selamat, Anda berhasil menjadi member Grosir Fashion Online.
    </div>";
    $query = $this->mdl_member->validate();
    $member_id = $this->mdl_member->get_userID($this->input->post('username'));

    $data = array(
        'username' => $this->input->post('username'),
        'member_id' => $member_id
    );

    $this->session->set_userdata($data);
    $_SESSION['username'] = $username['username'];
    redirect('member/account');
}

$this->session->set_flashdata('item', $value);
}

function get_data_from_post() {
    $data['nama_depan'] = $this->input->post('nama_depan', TRUE);
    $data['nama_belakang'] = $this->input->post('nama_belakang', TRUE);
    $data['email'] = $this->input->post('email', TRUE);
    $data['username'] = $this->input->post('username', TRUE);
    $data['password'] = $this->input->post('password', TRUE);
    $data['alamat'] = $this->input->post('alamat', TRUE);
}
```

```
$data['kode_pos'] = $this->input->post('kode_pos', TRUE);
$data['no_hp'] = $this->input->post('no_hp', TRUE);
$data['id_kota'] = $this->input->post('kota_id', TRUE);
return $data;
}

function get_data_from_db($update_id) {
    $query = $this->get_where($update_id);
    foreach ($query->result() as $row) {
        $data['member_id'] = $row->member_id;
        $data['nama_depan'] = $row->nama_depan;
        $data['nama_belakang'] = $row->nama_belakang;
        $data['email'] = $row->email;
        $data['username'] = $row->username;
        $data['password'] = $row->password;
        $data['alamat'] = $row->alamat;
        $data['kode_pos'] = $row->kode_pos;
        $data['no_hp'] = $row->no_hp;
    }
    if (!isset($data)) {
        $data = "";
    }

    return $data;
}

function get_data_order($order_id) {
    $query = $this->get_where_order($order_id);
    foreach ($query->result() as $row) {
        $data['order_id'] = $row->order_id;
        $data['id'] = $row->id;
        $data['qty'] = $row->qty;
        $data['price'] = $row->price;
        $data['name'] = $row->name;
        $data['image'] = $row->image;
        $data['weight'] = $row->weight;
        $data['warna'] = $row->warna;
        $data['total_weight'] = $row->total_weight;
        $data['subtotal'] = $row->subtotal;
        $data['total_ongkir'] = $row->total_ongkir;
        $data['total_bayar'] = $row->total_bayar;
    }
    if (!isset($data)) {
        $data = "";
    }

    return $data;
}

function get_data_member_db($member_id) {
    $query = $this->get_where_level_member($member_id);
```



```
foreach ($query->result() as $row) {  
  
    $data['level'] = $row->level;  
    $data['diskon'] = $row->diskon;  
}  
if (!isset($data)) {  
    $data = "";  
}  
  
return $data;  
}  
  
function get_data_member($order_by) {  
    $this->load->model('mdl_member');  
    $query = $this->mdl_member->get_data_member($order_by);  
    return $query;  
}  
  
function get($order_by) {  
    $this->load->model('mdl_member');  
    $query = $this->mdl_member->get($order_by);  
    return $query;  
}  
  
function get_where_level_member($id) {  
    $this->load->model('mdl_member');  
    $query = $this->mdl_member->get_where_level_member($id);  
    return $query;  
}  
  
function get_where($id) {  
    $this->load->model('mdl_member');  
    $query = $this->mdl_member->get_where($id);  
    return $query;  
}  
  
function get_where_order($id) {  
    $this->load->model('mdl_member');  
    $query = $this->mdl_member->get_where_order($id);  
    return $query;  
}  
  
function get_where_custom($col, $value) {  
    $this->load->model('mdl_member');  
    $query = $this->mdl_member->get_where_custom($col, $value);  
    return $query;  
}  
  
function _insert($data) {  
    $this->load->model('mdl_member');  
    $this->mdl_member->_insert($data);  
}
```

```
}

function _update($id, $data) {
    $this->load->model('mdl_member');
    $this->mdl_member->_update($id, $data);
}

function _delete($id) {
    $this->load->model('mdl_member');
    $this->mdl_member->_delete($id);
}

function count_where($column, $value) {
    $this->load->model('mdl_member');
    $count = $this->mdl_member->count_where($column, $value);
    return $count;
}

function get_max() {
    $this->load->model('mdl_member');
    $max_id = $this->mdl_member->get_max();
    return $max_id;
}
}
```

D.7.2. Package models

1. Class *mdl_member.php*

```
<?php

if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class Mdl_member extends CI_Model {

    var $tabel_provinsi = 'provinsi';
    var $tabel_kabupaten = 'kabupaten';
    var $tabel_kecamatan = 'kecamatan';
    var $tabel_kota = 'kota';

    function __construct() {
        parent::__construct();
    }

    function get_table() {
        $table = "member";
        return $table;
    }
}
```

```
function get_userID($username) {
    $this->db->where('username', $username);
    $query = $this->db->get('member');
    foreach ($query->result() as $row) {
        $member_id = $row->member_id;
        $fname = $row->nama_depan;
        $lname = $row->nama_belakang;
    }
    return $member_id;
}

function get_level($username) {
    $this->db->select('l.level, l.diskon');
    $this->db->from('member m');
    $this->db->join('level_member l', 'l.level_id = m.level_id');
    $this->db->where('m.username', $username);

    $query = $this->db->get();

    foreach ($query->result() as $row) {
        $level_id = $row->level_id;
        $level = $row->level;
        $diskon = $row->diskon;
    }
    return $level_id;
}

public function validate() {
    // grab user input
    $username = $this->security->xss_clean($this->input->post('username'));
    $password = $this->security->xss_clean($this->input->post('password'));

    // Prep the query
    $this->db->where('username', $username);
    $this->db->where('password', $password);

    // Run the query
    $query = $this->db->get('member');
    // Let's check if there are any results
    if ($query->num_rows == 1) {
        // If there is a user, then create session data
        $rows = $query->row();
        $data = array(
            'member_id' => $rows->member_id,
            'nama_depan' => $rows->nama_depan,
            'nama_belakang' => $rows->nama_belakang,
            'email' => $rows->email,
            'username' => $rows->username,
            'password' => $rows->password,
            'alamat' => $rows->alamat,
            'no_hp' => $rows->no_hp,
        );
    }
}
```

```
'id_kota' => $rows->id_kota,
'kode_pos' => $rows->kode_pos,
'poin' => $rows->poin,
'level_id' => $rows->level_id,
'validated' => true
);
$this->session->set_userdata($data);
return true;
}
// If the previous process did not validate
// then return false.
return false;
}

function get_kota() {
    $sql_kota = $this->db->get($this->tabel_kota);
    if ($sql_kota->num_rows() > 0) {
        foreach ($sql_kota->result_array() as $row) {
            $result['-'] = '- Pilih Kota -';
            $result[$row['id_kota']] = ucwords(strtolower($row['nama_kota']));
        }
        return $result;
    }
}

function get_data_member($order_by) {
    $this->db->select('m.nama_depan, m.nama_belakang, m.email, m.alamat, m.no_hp,
m.kode_pos, m.poin, kec.kecamatan, k.kabupaten, p.provinsi, l.level, l.diskon');
    $this->db->from('member m');
    $this->db->join('kecamatan kec', 'kec.id_kecamatan = m.id_kecamatan');
    $this->db->join('kabupaten k', 'k.id_kabupaten = m.id_kabupaten');
    $this->db->join('provinsi p', 'p.id_provinsi = m.id_provinsi');
    $this->db->join('level_member l', 'l.level_id = m.level_id');

    $query = $this->db->get();
    return $query;
}

function get($order_by) {
    $table = $this->get_table();
    $this->db->order_by($order_by);
    $query = $this->db->get($table);
    return $query;
}

function get_with_limit($limit, $offset, $order_by) {
    $table = $this->get_table();
    $this->db->limit($limit, $offset);
    $this->db->order_by($order_by);
    $query = $this->db->get($table);
    return $query;
}
```

```
}  
  
function get_where($member_id) {  
    $table = $this->get_table();  
    $this->db->where('member_id', $member_id);  
    $query = $this->db->get($table);  
    return $query;  
}  
  
function get_where_order($order_id) {  
    $this->db->select('*');  
    $this->db->from('orders');  
    $this->db->join('order_detail', 'orders.serial = order_detail.order_id');  
    $this->db->where('order_detail.order_id', $order_id);  
    $query = $this->db->get();  
    return $query;  
}  
  
function get_where_custom($col, $value) {  
    $this->db->where($col, $value);  
    $query = $this->db->get('provinsi');  
    return $query;  
}  
  
function _insert($data) {  
    $table = $this->get_table();  
    $this->db->insert($table, $data);  
}  
  
function _update($member_id, $data) {  
    $table = $this->get_table();  
    $this->db->where('member_id', $member_id);  
    $this->db->update($table, $data);  
}  
  
function _delete($member_id) {  
    $table = $this->get_table();  
    $this->db->where('member_id', $member_id);  
    $this->db->delete($table);  
}  
  
function count_where($column, $value) {  
    $table = $this->get_table();  
    $this->db->where($column, $value);  
    $query = $this->db->get($table);  
    $num_rows = $query->num_rows();  
    return $num_rows;  
}  
  
function count_all() {  
    $table = $this->get_table();
```

```

$query = $this->db->get($table);
$num_rows = $query->num_rows();
return $num_rows;
}

function get_max() {
    $table = $this->get_table();
    $this->db->select_max('member_id');
    $query = $this->db->get($table);
    $row = $query->row();
    $member_id = $row->member_id;
    return $member_id;
}

function _custom_query($mysql_query) {
    $query = $this->db->query($mysql_query);
    return $query;
}
}

```

D.7.3. Package views

1. *Class acoount.php*

```

<!--breadcrumb-->
<i class="fa fa-home"></i> Home / Akun

<div class="panel-body">
<h3>Dear Valued Customer,</h3>

<div style="margin-right: 15px; float: left; width: 400px; height: 270px; text-align: left">
  <div class="col-lg-13">
    <div class="well well-sm">
      <h4>Per Agustus 2014</h4>
      <p><i class="fa fa-ban"></i> Kami telah menghapus Bea Import<br>
      <p><i class="fa fa-tags"></i> Melakukan penurunan Harga Ecer sebesar 20% - 25% dari
      harga ecer sebelumnya.</p>
      <p><i class="fa fa-heart"></i> Diskon kini langsung berlaku dari Harga Ecer tertera</p>
      <p><i class="fa fa-truck"></i> Subsidi Ongkir Rp 8000/kg jika pembelian mencapai Rp
      300ribu setelah diskon</p>
    </div>
  </div>
</div>

<div style="margin-top: 300px; margin-right: -20px; width: 870px">
  <hr>
  <table border="0" style="width: 870px">
    <tr>

```

```

<td style="width: 20px">
  <h5>Level Member Anda <br>sekarang ini:</h5>
</td>
<td style="width: 20px">
  <h5>Anda memiliki <br> poin yg terkumpul:</h5>
</td>
<td style="width: 20px">
  <h5>Diskon tanpa minimum Order yg berhak Anda dapatkan saat ini:</h5>
</td>
</tr>
<tr>
<td>
  <h3>
    <?php
      $level_id = $this->session->userdata('level_id');
      if($level_id == "1"){
        echo "SILVER VIP";
      }else if($level_id == "2"){
        echo "GOLD VIP";
      }else if($level_id == "3"){
        echo "PLATINUM VIP";
      }else if($level_id == "4"){
        echo "DIAMOND VIP";
      }else if($level_id == "5"){
        echo "SUPER VIP";
      }else{
        echo "Level 0";
      }
    ?>
  </h3>
</td>
<?php $poin = $this->session->userdata('poin');?>
<td><h3><?php echo $poin;?></h3></td>
<td>
  <h3>
    <?php
      $level_id = $this->session->userdata('level_id');
      if($level_id == "1"){
        echo "5%";
      }else if($level_id == "2"){
        echo "9%";
      }else if($level_id == "3"){
        echo "14%";
      }else if($level_id == "4"){
        echo "25%";
      }else if($level_id == "5"){
        echo "28%";
      }else{
        echo "0%";
      }
    ?>
  </h3>
</td>

```

```

        </h3>
      </td>
    </tr>
  <tr>
    <td><h3></h3></td>
    <td>* 1 poin = Rp 100rb</td>
    <td><h3></h3></td>
  </table>
  <hr>
</div>

<h4>DISKON KHUSUS MEMBER VIP</h4>
<p>Kami menyediakan diskon bagi para member kami berdasarkan akumulasi total dari
pembelanjaan mereka sebelumnya.</p>
<p>Lihat detail di bawah untuk kebijakan diskon kami.</p>
<div class="table-responsive">
  <table class="table">
    <thead>
      <tr>
        <th><center>Level</center></th>
        <th><center>Diskon</center></th>
        <th colspan="2"><center>Akumulasi total pembelian Anda
sebelumnya</center></th>
      </tr>
    </thead>
    <tbody>
      <tr class="success">
        <td><center>Silver VIP</td>
        <td><center>5%</td>
        <td><center>di atas Rp 1.000.000</td>
        <td><center>setara 10 poin</td>
      </tr>
      <tr class="info">
        <td><center>Gold VIP</td>
        <td><center>9%</td>
        <td><center>di atas Rp 3.000.000</td>
        <td><center>setara 30 poin</td>
      </tr>
      <tr class="warning">
        <td><center>Platinum VIP</td>
        <td><center>14%</td>
        <td><center>di atas Rp 9.000.000</td>
        <td><center>setara 90 poin</td>
      </tr>
      <tr class="danger">
        <td><center>Diamond VIP</td>
        <td><center>25%</td>
        <td><center>di atas Rp 27.000.000</td>
        <td><center>setara 270 poin</td>
      </tr>
      <tr class="info">

```



```

        <td><center>Super VIP</td>
        <td><center>28%</td>
        <td><center>di atas Rp 50.000.000</td>
        <td><center>setara 500 poin</td>
    </tr>
</tbody>
</table>
</div>
<!-- /.table-responsive -->

<div style="float: right">
    <div class="col-lg-13">
        <div class="jumbotron">
            <ul>1. Diskon hanya berlaku untuk Total Belanja Anda, tidak untuk Ongkir JNE.</ul>
            <ul>2. Jika level Diskon Reseller lebih besar daripada Diskon Member VIP Anda, maka kami akan menggunakan Diskon Reseller di Order Anda.</ul>
            <ul>3. Bagi Member yang vakum atau tidak melakukan pembelian dalam kurun waktu 8 bulan berturut-turut, maka Membership dan semua akumulasi poin akan dihapus dari sistem.</ul>
        </div>
    </div>
</div>
</div>
</div>

```

2. Class login.php

```

<div class="container" style="width: 920px">
    <ol class="breadcrumb">
        <li><a href="http://localhost/onlineshop/mem_homepage/landing_page">Home</a></li>
        <li class="active">Member Login</li>
    </ol>
    <h2><i class="fa fa-sign-in"></i> Member Login</h2>
    <div class="panel-body" style="margin-left: 250px; width: 500px">
        <div class="panel-group" id="accordion">
            <div class="panel panel-default">
                <div class="panel-heading">
                    <h4 class="panel-title">
                        <a data-toggle="collapse" data-parent="#accordion" href="#collapseOne">Login
                        untuk proses belanja</a>
                    </h4>
                </div>
                <div id="collapseOne" class="panel-collapse collapse in">
                    <div class="panel-body" style="height: 230px">
                        <?php
                            if (isset($flash)) {
                                echo $flash;
                            }
                        <?>
                        <?php echo form_open('member/process'); ?>
                    <br />
                </div>
            </div>
        </div>
    </div>
</div>

```

```

<div class="input-group">
  <span class="input-group-addon"><i class="fa fa-user"></i></span>
  <input type="text" value="" name="username" id="username" class="form-
control" placeholder="username" required autofocus>
</div>
<br />
<div class="input-group">
  <span class="input-group-addon"><i class="fa fa-lock"></i></span>
  <input type="password" value="" name="password" id="password"
class="form-control" placeholder="password" required>
</div>
<br />
<button class="btn btn-lg btn-primary btn-block" name="submit"
type="Submit">Log in</button>
<?php echo form_close(); ?>
</div>
</div>
<div class="panel panel-default">
  <div class="panel-heading">
    <h4 class="panel-title">
      <a data-toggle="collapse" data-parent="#accordion"
href="#collapseTwo">Belum menjadi member? Daftar disini</a>
    </h4>
  </div>
  <div id="collapseTwo" class="panel-collapse collapse">
    <div class="panel-body">
      <?php echo Modules::run('member/register'); ?>
    </div>
  </div>
</div>
</div>
<div class="panel-footer" style="margin-left: -270px; width: 900px">
  <center><h4>Untungnya jadi member</h4></center>
  <table border="0" style="font-family: arial">
    <tr>
      <td style="width: 30px">
        <i class="fa fa-star-o"></i>
      </td>
      <td style="width: 400px">
        Akumulasi point dari setiap transaksi Anda yang menentukan level Member
        Anda, dimana setiap levelnya akan menikmati tambahan diskon khusus.
      </td>
      <td style="width: 30px">
      </td>
      <td style="width: 30px">
        <i class="fa fa-heart-o"></i>
      </td>
      <td style="width: 400px">

```

Membuat My Favorite, yaitu koleksi produk-produk yang Anda sukai, sehingga ketika Anda sudah memutuskan untuk belanja, tinggal lihat dari daftar produk yang sudah Anda masukkan ke My Favorite.


```
</td>
</tr>
<tr>
  <td style="width: 30px">
    <i class="fa fa-tags"></i>
  </td>
```

<td style="width: 400px">
 Menikmati promo spesial khusus member, mulai dari potongan belanja hingga hadiah langsung.

```
</td>
<td style="width: 30px">
</td>
<td style="width: 30px">
  <i class="fa fa-leaf"></i>
</td>
```

<td style="width: 400px">
 Bebas repot karena Anda tidak perlu lagi mengisi informasi pengiriman dan penagihan setiap kali Anda belanja.

```
</td>
</tr>
</table>
</div>
</div>
<!-- .panel-body -->
</div> <!-- /container -->
```

3. *Class order_details.php*

```
<i class="fa fa-home"></i> Home / Akun Saya / Order Saya / Kelengkapan Order
<div class="panel-body">
  <h3>Kelengkapan Order</h3>
  <div class="breadcrumb">
    <h4>Rangkuman Data Pembeli</h4>
  </div>
  <div class="panel-heading">
    <?php
    echo form_open($form_location);
    $member_id = $this->session->userdata('member_id');
    $fname = ucfirst($this->session->userdata('nama_depan'));
    $lname = ucfirst($this->session->userdata('nama_belakang'));
    $email = $this->session->userdata('email');
    $uname = $this->session->userdata('username');
    $pass = $this->session->userdata('password');
    $alamat = $this->session->userdata('alamat');
    $no_hp = $this->session->userdata('no_hp');
    $kode_pos = $this->session->userdata('kode_pos');
    $id_kota = $this->session->userdata('id_kota');
```



```

$this->db->from('orders');
$this->db->join('order_detail', 'orders.serial = order_detail.order_id');
$this->db->where('order_detail.order_id = ' . $order_id);
$query = $this->db->get();
$total_items = array();
foreach ($query->result_array() as $row) {
    $items_array[] = intval($row['qty']);
}
$items = array_sum($items_array);
?>

Total Belanja <?php echo $items; ?> buah
(
<?php
if ($level_id == "1") {
    echo "5%";
} else if ($level_id == "2") {
    echo "9%";
} else if ($level_id == "3") {
    echo "14%";
} else if ($level_id == "4") {
    echo "25%";
} else if ($level_id == "5") {
    echo "28%";
} else {
    echo "0%";
}
?>
)
</h5>
</td>
<?php
$this->db->select_sum('subtotal');
$this->db->from('orders');
$this->db->join('order_detail', 'orders.serial = order_detail.order_id');
$this->db->where('order_detail.order_id = ' . $order_id);
$query = $this->db->get();
$subtot = array();
foreach ($query->result_array() as $row) {
    $subtot_array[] = intval($row['subtotal']); //can it be float also?
}
$total = array_sum($subtot_array);
?>
<td style="text-align:right"><h5>Rp <?php echo number_format(($total), 0, ',', '.');
?></h5></td>
<td style="text-align: center"><h5><i class="fa fa-shopping-cart" style="font-size: 20px;
color: gray"></i></h5></td>
</tr>
<tr>
<td style="text-align:right" colspan="3">
<h5>

```

```

Ongkir dari Jakarta via JNE Reguler
(
<?php
$this->db->select_sum('total_weight');
$this->db->from('orders');
$this->db->join('order_detail', 'orders.serial = order_detail.order_id');
$this->db->where('order_detail.order_id = ' . $order_id);
$query = $this->db->get();
$weight_array = array();
foreach ($query->result_array() as $row) {
    $weight_array[] = intval($row['total_weight']); //can it be float also?
}
$total = array_sum($weight_array);
$berat = $total / 1000;
echo $berat;
?>
kg
)
</h5>
</td>
<td style="text-align:right">
<h5>Rp <?php echo $total_ongkir; ?>
</h5>
</td>
<td style="text-align: center"></td>
</tr>
<tr>
<td style="text-align:right" colspan="3"><h4 style="color: #FB4961">Total yang harus
dibayar</h4></td>
<td style="text-align:right; color: pink">
<h4 style="color: #FB4961">Rp <?php echo number_format($total_bayar, 0, ',', '.');
?></h4>
</td>
<td style="text-align: center"></td>
</tr>
</table>
</div>

```

4. *Class order_saya.php*

```

<i class="fa fa-home"></i> Home / Akun Saya / Order Saya
<div class="panel-body">
<h3>Order Saya</h3>

<p>Kemungkinan out of stock (OOS) dari beberapa produk pesanan Anda dapat terjadi
dikarenakan transaksi pada supplier kami tetap berjalan.</p>
<p>1. Silahkan klik Detail pada tabel dibawah ini untuk melihat kelengkapan Order
Anda</p>

```

<p>2. Seluruh barang yang OOS secara otomatis sudah menjadi deposit yang dapat digunakan untuk *order* Anda selanjutnya. silahkan lihat halaman Deposit Saya di menu kiri Anda</p>
 <p>3. Anda dapat meminta kami untuk me-refund nilai OOS / deposit dengan cara menekan tombol Refund yang ada di halaman Deposit Saya di menu kiri Anda. Refund akan kami transfer dalam tempo 4 hari kerja (diluar Sabtu Minggu), Silahkan klik menu Cek Status *Order* untuk mengetahui apakah refund Anda sudah ditransfer atau belum</p>

```

<div style="float: right">
  <button class="btn btn-default" href="<?php // echo base_url();
?>mem_order/cek_status_order"> Cek Status Order</button>
</div>

<br><br><br>
<h5>DISKON KHUSUS MEMBER VIP</h5>
<p>Kami menyediakan diskon bagi para member kami berdasarkan akumulasi total dari
pembelajaan mereka sebelumnya. Lihat detail di bawah untuk kebijakan diskon kami.</p>

<table class="table table-responsive table-striped" style="font-family: tahoma; font-size:
12px">
  <thead>
    <th><center>No. Order</center></th>
    <th><center>Detail Order</center></th>
    <th><center>Tanggal Pesan</center></th>
    <th><center>Status</center></th>
  </thead>
  <?php
  $member_id = $this->session->userdata('member_id');
  $this->db->select('*');
  $this->db->from('orders');
  $this->db->join('member', 'orders.member_id = member.member_id');
  $this->db->where('member.member_id = ' . $member_id);
  $query = $this->db->get();
  $count = 0;
  foreach ($query->result() as $row) {
    $count++;
    ?>
    <tr>
      <td><center><?php echo $row->serial; ?></center></td>
      <td><center><a class="btn btn-sm btn-info" href="<?php echo base_url(); ?><?php echo
'member/order_details/' . $row->serial; ?>">Kelengkapan Order</a></center></td>
      <td><center><?php echo $row->tgl; ?></center></td>
      <td><center><?php echo $row->status_order; ?></center></td>
    </tr>
    <?php
  }
  ?>
</table>
</div>

```

5. *Class register.php*

```
<?php
if (isset($flash)) {
    echo $flash;
}
?>
<?php echo form_open('member/submit'); ?>
<div class="form-group">
    <label for="nama_depan" class="control-label">Nama Depan</label>
    <input type="text" name="nama_depan" value="<?php echo $nama_depan; ?>" class="form-control" id="nama_depan" required />
</div>
<div class="form-group">
    <label for="nama_belakang" class="control-label">Nama Belakang</label>
    <input type="text" name="nama_belakang" value="<?php echo $nama_belakang; ?>" class="form-control" id="nama_belakang" required />
</div>
<div class="form-group">
    <label for="email" class="control-label">Email</label>
    <input type="email" name="email" value="<?php echo $email; ?>" class="form-control" id="email" required />
</div>
<div class="form-group">
    <label for="username" class="control-label">Username</label>
    <input type="text" name="username" value="<?php echo $username; ?>" class="form-control" id="username" required />
</div>
<div class="form-group">
    <label for="password" class="control-label">Password</label>
    <input type="password" name="password" value="<?php echo $password; ?>" class="form-control" id="password" required />
</div>
<div class="form-group">
    <label for="alamat" class="control-label">Alamat</label>
    <input type="text" name="alamat" value="<?php echo $alamat; ?>" class="form-control" id="alamat" required />
</div>
<div class="form-group">
    <label for="no_hp" class="control-label">No. HP</label>
    <input type="text" name="no_hp" value="<?php echo $no_hp; ?>" class="form-control" id="no_hp" required />
</div>
<div class="form-group">
    <label for="kode_pos" class="control-label">Kode Pos</label>
    <input type="text" name="kode_pos" value="<?php echo $kode_pos; ?>" class="form-control" id="kode_pos" required />
</div>
<div class="form-group">
    <label for="kota" class="control-label">Kota</label>
    <?php
```



```

    $style_kota = 'class="form-control input-sm" id="kota_id"';
    echo form_dropdown('kota_id', $kota, $style_kota);
    ?>
</div>
<button class="btn btn-sm btn-success" type="submit"><span class="glyphicon glyphicon-check"></span> Submit</button>
<?php
echo form_close();
?>

```

6. *Class update.php*

```

<!--breadcrumb-->
<i class="fa fa-home"></i> Home / Akun / Ubah Profil

<div class="panel-body">
  <h2><i class="fa fa-edit"></i> <?php echo $headline; ?></h2>
  <?php
  if (isset($flash)) {
    echo $flash;
  }
  echo form_open($form_location);
  $member_id = $this->session->userdata('member_id');
  $fname = ucfirst($this->session->userdata('nama_depan'));
  $lname = ucfirst($this->session->userdata('nama_belakang'));
  $email = $this->session->userdata('email');
  $uname = $this->session->userdata('username');
  $pass = $this->session->userdata('password');
  $alamat = $this->session->userdata('alamat');
  $no_hp = $this->session->userdata('no_hp');
  $kode_pos = $this->session->userdata('kode_pos');
  ?>

  <form method="POST" action="<?php echo base_url(); ?>mem_member/submit">
    <?php echo form_hidden('member_id', $member_id); ?>
    <div class="form-group">
      <label for="nama_depan" class="control-label">Nama Depan</label>
      <input type="text" name="nama_depan" value="<?php echo $fname; ?>" class="form-control" id="nama_depan" required />
    </div>
    <div class="form-group">
      <label for="nama_belakang" class="control-label">Nama Belakang</label>
      <input type="text" name="nama_belakang" value="<?php echo $lname; ?>" class="form-control" id="nama_belakang" required />
    </div>
    <div class="form-group">
      <label for="email" class="control-label">Email</label>
      <input type="email" name="email" value="<?php echo $email; ?>" class="form-control" id="email" required />
  </form>

```

```

</div>
<div class="form-group">
  <label for="username" class="control-label">Username</label>
  <input type="text" name="username" value="<?php echo $uname; ?>" class="form-
control" id="email" required />
</div>
<div class="form-group">
  <label for="password" class="control-label">Password</label>
  <input type="password" name="password" value="<?php echo $pass; ?>" class="form-
control" id="password" required />
</div>
<div class="form-group">
  <label for="alamat" class="control-label">Alamat</label>
  <input type="text" name="alamat" value="<?php echo $alamat; ?>" class="form-
control" id="alamat" required />
</div>
<div class="form-group">
  <label for="no_hp" class="control-label">No. HP</label>
  <input type="text" name="no_hp" value="<?php echo $no_hp; ?>" class="form-control"
id="no_hp" required />
</div>
<div class="form-group">
  <label for="kode_pos" class="control-label">Kode Pos</label>
  <input type="text" name="kode_pos" value="<?php echo $kode_pos; ?>" class="form-
control" id="kode_pos" required />
</div>
<div class="form-group">
  <label for="kota" class="control-label">Kota</label>
  <?php
  $style_kota = 'class="form-control input-sm" id="kota_id"';
  echo form_dropdown('kota_id', $kota, "", $style_kota);
  ?>
</div>
<button class="btn btn-sm btn-success" type="submit"><span class="glyphicon glyphicon-
check"></span> Submit</button>
</form>
<?php
echo form_close();
?>
<br>
<div class="panel-footer">
  <center><h4>Untungnya jadi member</h4></center>
  <table border="0" style="font-family: arial">
    <tr>
      <td style="width: 30px">
        <i class="fa fa-star-o"></i>
      </td>
      <td style="width: 400px">
        Akumulasi point dari setiap transaksi Anda yang menentukan level Member Anda,
        dimana setiap levelnya akan menikmati tambahan diskon khusus.
      </td>
    </tr>
  </table>
</div>

```

```

<td style="width: 30px">
</td>
<td style="width: 30px">
  <i class="fa fa-heart-o"></i>
</td>
<td style="width: 400px">
  Membuat My Favorite, yaitu koleksi produk-produk yang Anda sukai, sehingga
  ketika Anda sudah memutuskan untuk belanja, tinggal lihat dari daftar produk yang sudah Anda
  masukkan ke My Favorite.<br><br>
</td>
</tr>
<tr>
<td style="width: 30px">
  <i class="fa fa-tags"></i>
</td>
<td style="width: 400px">
  Menikmati promo spesial khusus member, mulai dari potongan belanja hingga
  hadiah langsung.
</td>
<td style="width: 30px">
</td>
<td style="width: 30px">
  <i class="fa fa-leaf"></i>
</td>
<td style="width: 400px">
  Bebas repot karena Anda tidak perlu lagi mengisi informasi pengiriman dan
  penagihan setiap kali Anda belanja.
</td>
</tr>
</table>
</div>
</div>

```

D.8. *Package konfirmasi_bayar*

D.8.1. *Package controllers*

1. *Class konfirmasi_bayar.php*

```

<?php
if (!defined('BASEPATH'))
  exit('No direct script access allowed');

class Mem_testimonial extends MX_Controller {

  function __construct() {
    parent::__construct();
  }

  function get_data_from_post() {

```

```
$data['nama_user'] = $this->input->post('nama_user', TRUE);
$data['email'] = $this->input->post('email', TRUE);
$data['no_hp'] = $this->input->post('no_hp', TRUE);
$data['kota'] = $this->input->post('kota', TRUE);
$data['pesan'] = $this->input->post('pesan', TRUE);
return $data;
}

function create() {
    $testi_id = $this->uri->segment(3);

    //ngambil atribut dr db
    $data = $this->get_data_from_post();

    //merubah url
    $current_url = current_url();
    $data['form_location'] = str_replace('/create', '/submit', $current_url);

    $flash = $this->session->flashdata('item');
    if ($flash != "") {
        $data['flash'] = $flash;
    }

    $template = "public_one_col";
    $data['view_file'] = "create";
    $this->load->module('template');
    $this->load->template->$template($data);
}

function submit() {
    $this->load->helper(array('form', 'url'));

    $this->load->library('form_validation');
    $this->form_validation->set_rules('nama_user', 'Nama User', 'required');

    if ($this->form_validation->run() == FALSE) {
        $this->create();
    } else {
        $update_id = $this->uri->segment(3);

        if ($update_id > 0) {
            //untuk edit, bukan create
            $data = $this->get_data_from_post();
            $this->_update($update_id, $data);
            $value = "<div class='alert alert-success alert-dismissable'>
                <button type='button' class='close' data-dismiss='alert' aria-
hidden='true'>&times;</button>
                Testimonial telah berhasil diubah.
            </div>";

            $this->session->set_flashdata('item', $value);
```

```
$update_id = $this->get_max();
redirect('mem_testimonial/update/'.$update_id);
} else {
    //tambah testi
    $data = $this->get_data_from_post();
    $this->_insert($data);
    $value = "<div class='alert alert-success alert-dismissable'>
        <button type='button' class='close' data-dismiss='alert' aria-
hidden='true'>&times;</button>
        Terimakasih, testimonial Anda telah ditambahkan.
    </div>";

    $this->session->set_flashdata('item', $value);
    $update_id = $this->get_max();
    redirect('mem_testimonial/create');
}
}
}

function show_testi() {
    $data['query'] = $this->get('testi_id');
    $this->db->order_by("testi_id", "desc");
    $this->load->view('show_testi', $data);
}

function view_all_testi() {
    $template = "public_one_col";
    $data['view_file'] = "view_all_testi";
    $this->load->module('template');
    $this->load->template->$template($data);
}

function _view_all_testi() {
    $this->load->library('pagination');

    $config['base_url'] = base_url(). 'mem_testimonial/view_all_testi';
    $config['total_rows'] = $this->db->get('testimonial')->num_rows();
    $config['per_page'] = 3;
    $config['num_links'] = 5;

    $this->pagination->initialize($config);
    $data['query'] = $this->db->get('testimonial', $config['per_page'], $this->uri->segment(3));

    $this->db->order_by("testi_id", "desc");
    $this->load->view('_view_all_testi', $data);
}

function get($order_by) {
    $this->load->model('mdl_testi');
    $query = $this->mdl_testi->get($order_by);
    return $query;
}
```

```
}

function get_where($id) {
    $this->load->model('mdl_testi');
    $query = $this->mdl_testi->get_where($id);
    return $query;
}

function _insert($data) {
    $this->load->model('mdl_testi');
    $this->mdl_testi->_insert($data);
}

function _update($id, $data) {
    $this->load->model('mdl_testi');
    $this->mdl_testi->_update($id, $data);
}

function _delete($id) {
    $this->load->model('mdl_testi');
    $this->mdl_testi->_delete($id);
}

function get_max() {
    $this->load->model('mdl_testi');
    $max_id = $this->mdl_testi->get_max();
    return $max_id;
}
}
```

D.8.2. Package models

1. Class *mdl_konfirm.php*

```
<?php if (!defined('BASEPATH')) exit('No direct script access allowed');
class Mdl_testi extends CI_Model {

    function __construct() {
        parent::__construct();
    }

    function get_table() {
        $table = "testimonial";
        return $table;
    }

    function get($order_by) {
        $table = $this->get_table();
        $this->db->order_by($order_by);
        $query=$this->db->get($table);
        return $query;
    }
}
```

```

}

function get_where($testi_id) {
    $table = $this->get_table();
    $this->db->where('testi_id', $testi_id);
    $query=$this->db->get($table);
    return $query;
}

function _insert($data) {
    $table = $this->get_table();
    $this->db->insert($table, $data);
}

function _update($testi_id, $data) {
    $table = $this->get_table();
    $this->db->where('testi_id', $testi_id);
    $this->db->update($table, $data);
}

function _delete($testi_id) {
    $table = $this->get_table();
    $this->db->where('testi_id', $testi_id);
    $this->db->delete($table);
}

function get_max() {
    $table = $this->get_table();
    $this->db->select_max('testi_id');
    $query = $this->db->get($table);
    $row=$query->row();
    $testi_id=$row->testi_id;
    return $testi_id;
}
}

```

D.8.3. Package views

1. Class *_view_all_testi.php*

```

<?php
//5 untuk limit, 0 utk offset = data yg ditampilkan mulai dr stlh 0
$query = $this->db->get('testimonial', 5, 0);
$count = 0;
foreach ($query->result() as $row) {
    $count++;
    ?>
    <ul class="timeline">
        <li style="height: 100px">
            <div class="timeline-badge primary"><i class="fa fa-user"></i>

```

```

    </div>
    <div class="timeline-panel">
      <div class="timeline-heading">
        <font style="font-family: monospace; font-size: 12px"><b><?php echo $row-
        >nama_user ?> - <?php echo $row->kota ?></b></font>
      </div>
    </div>
  </li>
  <li class="timeline-inverted" style="height: 100px">
    <div class="timeline-badge danger"><i class="fa fa-comment"></i>
  </div>
  <div class="timeline-panel">
    <div class="timeline-body">
      <font style="font-family: sans-serif; font-size: 10px"><?php echo $row->pesan
    ?></font>
    </div>
  </div>
  </li>
</ul>
<?php
}
?>

```

2. *Class create.php*

```

<div id="create_testi_container">
  <div class="panel-body">
    <?php
    if (isset($flash)) {
      echo $flash;
    }
    ?>
    <ul class="timeline">
      <li class="timeline" style="height: 200px" >
        <div class="timeline-badge info">
          <i class="fa fa-user"></i>
        </div>
        <div class="timeline-panel">
          <div class="timeline-heading"><h4 class="timeline-title">Testimonial</h4>
          <p>Kirim testimonial anda dengan mengisi form dibawah ini.</p>
        </div>
        <div class="timeline-body">
          <?php
          echo validation_errors(" ");
          echo form_open($form_location);
          ?>
          <form method="POST" action="<?php echo base_url();
          ?>mem_testimonial/submit">
            <div class="form-group" style="font-family: sans-serif; font-size: 11px">
              <label for="nama_user" class="control-label">Nama</label>

```



```

        <input type="text" name="nama_user" value="<?php echo $nama_user; ?>"
class="form-control" id="nama_user" required />
    </div>
    <div class="form-group" style="font-family: sans-serif; font-size: 11px">
        <label for="email" class="control-label">Email</label>
        <input type="email" name="email" value="<?php echo $email; ?>"
class="form-control" id="email" required />
    </div>
    <div class="form-group" style="font-family: sans-serif; font-size: 11px">
        <label for="no_hp" class="control-label">No. HP</label>
        <input type="text" name="no_hp" value="<?php echo $no_hp; ?>"
class="form-control" id="no_hp" required />
    </div>
    <div class="form-group" style="font-family: sans-serif; font-size: 11px">
        <label for="kota" class="control-label">Kota</label>
        <input type="text" name="kota" value="<?php echo $kota; ?>" class="form-
control" id="kota" required />
    </div>
</div>
</div>
</li>

<li class="timeline-inverted">
    <div class="timeline-badge warning"><i class="fa fa-pencil-square-o"></i>
    </div>
    <div class="timeline-panel">
        <div class="timeline-heading">
            <h4 class="timeline-title">Pesan</h4>
        </div>
        <div class="timeline-body">
            <form method="POST" action="<?php echo base_url();
?>mem_testimonial/submit">
                <div class="form-group" style="font-family: sans-serif; font-size: 11px">
                    <textarea class="form-control" rows="3" value="<?php echo $pesan; ?>"
name="pesan" id="pesan"></textarea>
                </div>
                <center>
                    <button class="btn btn-sm btn-warning" type="reset" value="Reset"><span
class="glyphicon glyphicon-cloud"></span> Reset</button>
                    <button class="btn btn-sm btn-success" type="submit"><span
class="glyphicon glyphicon-check"></span> Submit</button>
                </center>
            </form>
            <?php
echo form_close();
?>
        </div>
    </div>
</li>
</ul>
</div>

```

```
</div>
```

3. *Class show_testi.php*

```
<div id="page-wrapper">
  <div class="panel panel-success">
    <div class="panel-heading">
      <center>
        <h4>TESTIMONIAL</h4>
      </center>
    </div>
    <div class="panel-body">
      <center>
        <table border="0" width="190px">
          <?php
            //2 untuk limit, 0 utk offset = data yg ditampilkan mulai dr stlh 0
            $query = $this->db->get('testimonial', 1, 0);
            $count = 0;
            foreach ($query->result() as $row) {
              $count++;
              ?>
              <tr>
                <td>
                  <i class="fa fa-quote-left"></i>
                  <font style="font-family: sans-serif; font-size: 10px">
                    <?php echo $row->pesan; ?>
                  </font>
                  <i class="fa fa-quote-right"></i>
                </td>
                <tr class="pull-right">
                  <td>
                    <font style="font-family: monospace; font-size: 12px"><b>
                      <?php echo $row->nama_user; ?> - <?php echo $row->kota; ?>
                    </b></font>
                  </td>
                </tr>
              <?php
            }
            ?>
          </table>
        </center>
      </div>
      <center>
        <div class="panel-footer">
          <a class="btn btn-xs btn-default" style="width: 180px; margin-bottom: 2px"
            title="Lihat Semua Testimonial" href="<?php echo base_url(); ?><?php echo
            'mem_testimonial/view_all_testi'; ?>">
            <span class="fa fa-comments-o"> Lihat semua testimonial</span>
```

```

    </a>
    <a class="btn btn-xs btn-primary" style="width: 180px" title="Tulis Testimonial
Anda" href="<?php echo base_url(); ?><?php echo 'mem_testimonial/create'; ?>">
    <span class="fa fa-pencil"> Tulis testimonial Anda</span>
    </a>
  </div>
</center>
</div>
</div>

```

4. *Class view_all_testi.php*

```

<div class="container">
  <h2>
    <i class="fa fa-comments-o"></i> Testimonial
  </h2>
  <div class="panel-body" style="width: 900px">
    <?php echo Modules::run('mem_testimonial/_view_all_testi'); ?>
    <center>
      <?php
        echo "<span style='font-size: 17px'" . $this->pagination->create_links() . "</span">";
      ?>
    </center>
  </div>
</div>

```

D.9. *Package mem_cart*

D.9.1. *Package controllers*

D.9.2. *Package models*

D.9.3. *Package views*

D.10. *Package mem_produk*

D.10.1. *Package controllers*

1. *Class mem_produk.php*

```

<?php
if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class Mem_produk extends MX_Controller {

    function __construct() {
        parent::__construct();
    }
}

```

```
}

function harga_diskon($level_id) {
    $data = $this->get_data_level($level_id);
    $data['level_id'] = $level_id;
    $data['view_file'] = "harga_diskon";

    $template = "public_one_col";
    $this->load->module('template');
    $this->load->template->$template($data);
}

function get_data_level($level_id) {
    $query = $this->get_where_level($level_id);
    foreach ($query->result() as $row) {
        $data['level_id'] = $row->level_id;
        $data['level'] = $row->level;
        $data['diskon'] = $row->diskon;
    }
    if (!isset($data)) {
        $data = "";
    }

    return $data;
}

function landing_page() {
    $template = "public_one_col";
    $data['view_file'] = "landing_page";
    $this->load->module('template');
    $this->load->template->$template($data);
}

function products_landing_page() {
    $data['query'] = $this->get('produk_id');
    $this->db->order_by("produk_id", "desc");
    $this->load->view('products_landing_page', $data);
}

function cincin() {
    $this->load->library('pagination');

    $config['base_url'] = base_url() . 'mem_produk/cincin';
    $config['total_rows'] = $this->db->get('kategori_produk')->num_rows();
    $config['per_page'] = 10;
    $config['num_links'] = 5;

    $this->pagination->initialize($config);
    $data['query'] = $this->db->get('kategori_produk', $config['per_page'], $this->uri->segment(3));
    $this->get_where('nama_kategori');
```

```
$template = "public_one_col";
$data['view_file'] = "kat_cincin";
$this->load->module('template');
$this->load->template->$template($data);
}

function kalung() {
    $this->load->library('pagination');

    $config['base_url'] = base_url() . 'mem_produk/kalung';
    $config['total_rows'] = $this->db->get('kategori_produk')->num_rows();
    $config['per_page'] = 10;
    $config['num_links'] = 5;

    $this->pagination->initialize($config);
    $data['query'] = $this->db->get('kategori_produk', $config['per_page'], $this->uri->segment(3));
    $this->get_where('nama_kategori');

    $template = "public_one_col";
    $data['view_file'] = "kat_kalung";
    $this->load->module('template');
    $this->load->template->$template($data);
}

function gelang() {
    $this->load->library('pagination');

    $config['base_url'] = base_url() . 'mem_produk/gelang';
    $config['total_rows'] = $this->db->get('kategori_produk')->num_rows();
    $config['per_page'] = 10;
    $config['num_links'] = 5;

    $this->pagination->initialize($config);
    $data['query'] = $this->db->get('kategori_produk', $config['per_page'], $this->uri->segment(3));
    $this->get_where('nama_kategori');

    $template = "public_one_col";
    $data['view_file'] = "kat_gelang";
    $this->load->module('template');
    $this->load->template->$template($data);
}

function get_by_kategori() {
    $this->load->model('mdl_produk');
    $query = $this->mdl_produk->get_by_kategori();
    return $query;
}
```

```
function paket() {
    $data['query'] = $this->get('produk_id');
    $this->db->order_by('produk_id', 'random');
    $this->load->view('paket', $data);
}

function show($produk_id) {
    $data = $this->get_data_from_db($produk_id);
    $template = "public_one_col";
    $data['produk_id'] = $produk_id;
    $data['view_file'] = "showproduk";
    $this->load->module('template');
    $this->template->$template($data);
}

function get_data_from_db($update_id) {
    $query = $this->get_where($update_id);
    foreach ($query->result() as $row) {
        $data['produk_id'] = $row->produk_id;
        $data['nama_produk'] = $row->nama_produk;
        $data['harga_produk'] = $row->harga_produk;
        $data['ukuran_produk'] = $row->ukuran_produk;
        $data['bahan_produk'] = $row->bahan_produk;
        $data['berat_produk'] = $row->berat_produk;
        $data['gbr_thumb'] = $row->gbr_thumb;
        $data['gbr_besar'] = $row->gbr_besar;
        $data['produk_url'] = $row->produk_url;
    }
    if (!isset($data)) {
        $data = "";
    }

    return $data;
}

function get($order_by) {
    $this->load->model('mdl_produk');
    $query = $this->mdl_produk->get($order_by);
    return $query;
}

function get_where($id) {
    $this->load->model('mdl_produk');
    $query = $this->mdl_produk->get_where($id);
    return $query;
}

function get_where_level($id) {
    $this->load->model('mdl_produk');
    $query = $this->mdl_produk->get_where_level($id);
    return $query;
}
```

```

    }

    function _insert($data) {
        $this->load->model('mdl_produk');
        $this->mdl_produk->_insert($data);
    }

    function _update($id, $data) {
        $this->load->model('mdl_produk');
        $this->mdl_produk->_update($id, $data);
    }

    function _delete($id) {
        $this->load->model('mdl_produk');
        $this->mdl_produk->_delete($id);
    }

    function get_max() {
        $this->load->model('mdl_produk');
        $max_id = $this->mdl_produk->get_max();
        return $max_id;
    }
}

```

D.10.2. Package models

1. Class *mdl_produk.php*

```

<?php

if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class Mdl_produk extends CI_Model {

    function __construct() {
        parent::__construct();
    }

    function get_all() {
        $query = $this->db->get('data_produk')->result();
        foreach ($query as &$result) {
            if ($result->warna_produk) {
                $result->warna_produk = explode(',', $result->warna_produk);
            }
        }
        return $query;
    }

    function get_by_kategori() {

```

```
$this->db->distinct();
$this->db->select('kategori.nama_kategori');
$this->db->from('kategori_produk');
$this->db->join('data_produk', 'kategori_produk.produk_id=data_produk.produk_id');
$this->db->join('kategori', 'kategori_produk.kategori_id=kategori.kategori_id');

$query = $this->db->get();
return $query->result();
}

function get($order_by) {
    $this->db->order_by($order_by);
    $query = $this->db->get('data_produk');

    return $query;
}

function get_with_limit($limit, $offset, $order_by) {
    $this->db->limit($limit, $offset);
    $this->db->order_by($order_by);
    $query = $this->db->get('data_produk');
    return $query;
}

function get_where_level($level_id) {
    $this->db->where('level_id', $level_id);
    $query = $this->db->get('level_member');

    return $query;
}

function get_where($produk_id) {
    $this->db->where('produk_id', $produk_id);
    $query = $this->db->get('data_produk');

    return $query;
}

function get_where_custom($col, $value) {
    $this->db->where($col, $value);
    $query = $this->db->get('data_produk');
    return $query;
}

function _insert($data) {
    $this->db->insert('data_produk', $data);
}

function _update($produk_id, $data) {
    $this->db->where('produk_id', $produk_id);
    $this->db->update('data_produk', $data);
}
```



```

    }

    function _delete($produk_id) {
        $this->db->where('produk_id', $produk_id);
        $this->db->delete('data_produk');
    }

    function get_max() {
        $this->db->select_max('produk_id');
        $query = $this->db->get('data_produk');
        $row = $query->row();
        $produk_id = $row->produk_id;
        return $produk_id;
    }
}

```

D.10.3. Package views

1. Class *kat_kalung.php*

```

<center>
<?php
echo "<span style='font-size: 17px'>". $this->pagination->create_links(). "</span>";
?>
</center>
<?php
    $this->db->select('data_produk.produk_id, data_produk.nama_produk,
data_produk.harga_produk, data_produk.gbr_thumb');
    $this->db->from('kategori_produk');
    $this->db->join('data_produk', 'kategori_produk.produk_id=data_produk.produk_id');
    $this->db->join('kategori', 'kategori_produk.kategori_id=kategori.kategori_id');
    $this->db->where('kategori.nama_kategori = "kalung"');

    $query = $this->db->get();

    $count = 0;
    foreach ($query->result() as $row) {
        $count++;
        ?>
        <div class="col-sm-2 col-md-2 col-lg-2">
            <div class="thumbnail">
                <a href="<?php echo base_url(); ?>mem_produk/show/<?php echo $row->produk_id;
?>">
                    </a>
                    <div class="caption">
                        <h6><a href="<?php echo base_url(); ?>mem_produk/show/<?php echo $row-
>produk_id; ?>"><?php echo $row->nama_produk; ?></a></h6>
                        <?php echo "Rp " . number_format($row->harga_produk, 0, ',', '.'); ?>
                    </div>
            </div>
        </div>
    }
}

```

```

    </div>
  </div>
  <?php
}
?>

```

2. Class paket.php

```

<div id="page-wrapper">
  <div class="panel panel-warning">
    <div class="panel-heading">
      <center><h4>PAKET HEMAT</h4></center>
    </div>
    <div class="panel-body">
      <center>
        <table border="0">
          <?php
            $query = $this->db->get('data_produk', 2, 0);
            $count = 0;
            foreach ($query->result() as $row) {
              $count++;
              ?>
              <tr>
                <td rowspan="2">
                  <a href="<?php echo base_url(); ?>mem_produk/show/<?php echo $row->produk_id ?>"></a>
                </td>
                <td><?php echo nbs(3) ?></td>
                <td><a href="<?php echo base_url(); ?>mem_produk/show/<?php echo $row->produk_id ?>"><font style="font-family: tahoma; font-size: 10px"><?php echo $row->nama_produk; ?></font></a></td>
              </tr>
              <tr>
                <td></td>
                <td><a href="<?php echo base_url(); ?>mem_produk/show/<?php echo $row->produk_id ?>"><font style="font-family: tahoma; font-size: 10px">
                  Rp <?php echo $row->harga_produk; ?>
                </font></a></td>
              </tr>
              <tr>
                <td><?php echo nbs(3) ?></td>
              </tr>
            <?php
          }
          ?>
        </table>
      </center>

```

```

</div>
<div class="panel-footer">
  <center>
    <a class="btn btn-xs btn-default" style="width: 180px" title="Lihat Paket Produk"
href="<?php echo base_url(); ?><?php echo '#'; ?>">
      <span class="fa fa-suitcase"> Lihat Paket Lainnya</span>
    </a>
  </center>
</div>
</div>
</div>

```

3. *Class products_landing_page.php*

```

<div class="row">
  <div class="well" style="height: 50px; margin-left: 30px; margin-right: 30px">
    <div style="float: left; width: 200px; font-family: arial; color: #E68F8F;">
      <b>NEW ARRIVAL: Kalung</b>
    </div>
    <div style="float: right; width: 200px; text-align: right; color: #E68F8F;">
      <a href="<?php echo base_url(); ?><?php echo 'mem_produk/kalung'?>">View All
    >></a>
  </div>
</div>
</div>
<div class="clearfix"></div>
<?php
$this->db->select('data_produk.produk_id, data_produk.nama_produk,
data_produk.harga_produk, data_produk.gbr_thumb');
$this->db->from('kategori_produk');
$this->db->join('data_produk', 'kategori_produk.produk_id=data_produk.produk_id');
$this->db->join('kategori', 'kategori_produk.kategori_id=kategori.kategori_id');
$this->db->where('kategori.nama_kategori = "kalung"');

$query = $this->db->get("", 6, 0);

$count = 0;
foreach ($query->result() as $row) {
  $count++;
  ?>
  <div class="col-sm-2 col-md-2 col-lg-2">
    <div class="thumbnail">
      <a href="<?php echo base_url(); ?>mem_produk/show/<?php echo $row->produk_id;
?>">
        </a>
      <div>
        &nbsp;
        <?php echo "Rp " . number_format($row->harga_produk, 0, ',', '.'); ?>
      </div>
    </div>
  </div>
}
?>

```

```

</div>
</div>
<?php
}
?>
<div class="clearfix"></div>
<!--===== END KALUNG
=====-->

<div class="row">
  <div class="well" style="height: 50px; margin-left: 30px; margin-right: 30px">
    <div style="float: left; width: 200px; font-family: arial; color: #E68F8F;">
      <b>NEW ARRIVAL: Cincin</b>
    </div>
    <div style="float: right; width: 200px; text-align: right; color: #E68F8F;">
      <a href="<?php echo base_url();?><?php echo 'mem_produk/cincin'?>">View All
    >></a>
  </div>
</div>
</div>
<div class="clearfix"></div>
<?php
$this->db->select('data_produk.produk_id, data_produk.nama_produk,
data_produk.harga_produk, data_produk.gbr_thumb');
$this->db->from('kategori_produk');
$this->db->join('data_produk', 'kategori_produk.produk_id=data_produk.produk_id');
$this->db->join('kategori', 'kategori_produk.kategori_id=kategori.kategori_id');
$this->db->where('kategori.nama_kategori = "cincin"');

$query = $this->db->get("", 6, 0);

$count = 0;
foreach ($query->result() as $row) {
  $count++;
  ?>
  <div class="col-sm-2 col-md-2 col-lg-2">
    <div class="thumbnail">
      <a href="<?php echo base_url(); ?>mem_produk/show/<?php echo $row->produk_id;
?>">
        </a>
      <div>
        &nbsp;
        <?php echo "Rp " . number_format($row->harga_produk, 0, ",", '.'); ?>
      </div>
    </div>
  </div>
</div>
<?php
}
?>

```

4. *Class showproduk.php*

```

<div class="panel-body">
  <!--breadcrumb-->
  <ol class="breadcrumb">
    <li><a href="<?php echo base_url(); ?>mem_homepage/landing_page">Home</a></li>
    <li><a href="#">Produk</a></li>
    <li class="active">Item No. <?php echo $produk_id; ?></li>
  </ol>
  <div class="col-md-6">
    <center>
      <?php
        $pic_path = base_url() . "img/imgproduk/" . $gbr_besar;
        ?>
      
    </center>
  </div>
  <div class="col-md-5">
    <?php echo form_open('mem_cart/add'); ?>
    <h3 style="color: #46b8da">
      <?php
        echo $nama_produk;
        ?>
    </h3>
    <!--number format($angka, $jumlah_desimal, $pemisah_desimal, $pemisah_ribuan)-->
    <h4><?php echo "Rp " . number_format($harga_produk, 0, ',', '.'); ?></h4>
    <?php
      $level_id = $this->session->userdata('level_id');
      if (!$level_id == "0") {
        ?>
        Harga Anda saat ini: <br/>
        <?php
          $member_id = $this->session->userdata('member_id');
          if ($member_id < 1) {
            ?>
            <a class="btn btn-xs btn-warning" href="<?php echo base_url();
?>member/login">Login / Register</a>
            <?php
          } else {
            echo "<h3 style='color: violet'>";
            if ($level_id == "1") {
              $harga_diskon = $harga_produk - $harga_produk * 0.05;
            } else if ($level_id == "2") {
              $harga_diskon = $harga_produk - $harga_produk * 0.09;
            } else if ($level_id == "3") {
              $harga_diskon = $harga_produk - $harga_produk * 0.14;
            } else if ($level_id == "4") {
              $harga_diskon = $harga_produk - $harga_produk * 0.25;
            } else if ($level_id == "5") {
              $harga_diskon = $harga_produk - $harga_produk * 0.28;
            }
          }
        }
      }
    }
  }

```

```

        echo "Rp " . number_format($harga_diskon, 0, ",", '.') . "</h3>";
    }
} else {
    echo "";
}
?>
<hr>
<?php
echo Modules::run('mem_cart/_display_add_to_cart_box', $produk_id, $harga_produk);
?>
</div>
<div class="col-md-12">
    <br>
    <div class="panel panel-default">
        <!-- Default panel contents -->
        <div class="panel-heading" style="font-size: 11px">Deskripsi Produk</div>
        <div class="panel-body" style="font-size: 11px">
            <p>Mohon diperhatikan sebelum membeli.</p>
        </div>
        <!-- Table -->
        <table class="table" style="size: 900px; font-size: 11px">
            <tr>
                <td rowspan="7">
                    <?php
                    $pic_path = base_url() . "img/imgproduk/" . $gbr_thumb;
                    ?>
                    
                </td>
            </tr>
            <tr>
                <td>Berat</td>
                <td:</td>
                <td><center><?php echo $berat_produk; ?> gr</center></td>
            </tr>
            <tr>
                <td>Bahan</td>
                <td:</td>
                <td><center><?php echo $bahan_produk; ?></center></td>
            </tr>
            <tr>
                <td>Detail Ukuran Produk</td>
                <td:</td>
                <td><center><?php echo $ukuran_produk; ?> mm</center></td>
            </tr>
            <tr>
                <td>Detail Ukuran Produk</td>
                <td:</td>
                <td><center>Mohon perhatikan tabel ukuran ini dengan teliti. Beberapa ukuran
                produk dapat sedikit berbeda karena tergantung dari tanggal pembuatan pabrik</center></td>
            </tr>
            <tr>

```

```
<td>Kondisi</td>
<td>:</td>
<td><center>100% baru / tanpa tambahan aksesoris lainnya</center></td>
</tr>
<tr>
<td>Catatan</td>
<td>:</td>
<td><center>Warna dapat sedikit berbeda karena perbedaan pengaturan warna dan
cahaya layar monitor Anda, sehingga warna di foto belum tentu sama 100% dengan produk yang
nantinya Anda terima</center></td>
</tr>
</table>
</div>
</div>
</div>
```

E. Listing Program Fitur Menentukan Rekomendasi Paket Produk

E.1. Listing Program Fungsi Mengambil Masukan Nilai Kriteria

```

22 //mengambil nilai minsup minconf jumlah transaksi dari inputan
23 public function ambilinput() { }1
24     $minsup = $this->input->post('minsup');
25     $minconf = $this->input->post('minconf');
26     $jumlah = $this->input->post('jumlahtransaksi');
27     redirect("adm_paket/inputapriori/$minsup/$minconf/$jumlah");
28 } }3

```

Gambar E.1. Listing Program Fungsi Mengambil Masukan Nilai Kriteria

E.2. Listing Program Fungsi Menjalankan *Library* Apriori Sesuai Nilai Masukan

```

30 //APRIORI
31 public function inputapriori($minsup, $minconf, $jumlah) { }1
32     $ambilldb = $this->mdl_paket->ambilldb($jumlah);
33
34     $this->load->library('my_lib');
35     $this->my_lib->setData($jumlah, $minsup, $minconf);
36     $this->my_lib->process($ambilldb);
37
38     $data['pgetasosiasi'] = $this->my_lib->getAssociationRules();
39     $data['input'] = array('maxScan' => $jumlah, 'minSup' => $minsup, 'minConf' => $minconf);
40
41     $template = "admin";
42     $data['view file'] = "hasil";
43     $this->load->module('template');
44     $this->template->$template($data);
45 } }3

```

Gambar E.2. Listing Program Fungsi Menjalankan *Library* Apriori Sesuai Nilai Masukan

F. Listing Program *Library* Algoritma Apriori *My_lib*

F.1. Listing Program Fungsi Pembuatan Tabel Data Perhitungan *Support*

```

29 private function makeTable($db) { }1
30     $stable = $array = array(); }2
31     $counter = 1;
32     for ($i = 0, $num = count($db); $i < $num; $i++) { }3
33         $tmp = $db[$i]; }4
34         $x = array(); }4
35         for ($j = 0, $num1 = count($tmp); $j < $num1; $j++) { }5
36             $x = trim($tmp[$j]); }6
37             if ($x === '') { }7
38                 continue; }8
39             }
40
41             if (!isset($this->keys['v->k'][$x])) { }9
42                 $this->keys['v->k'][$x] = $x;
43                 $this->keys['k->v'][$counter] = $x; }10
44                 $counter++;
45             }
46
47             if (!isset($array[$this->keys['v->k'][$x]])) { }11
48                 $array[$this->keys['v->k'][$x]] = 1;
49                 $this->allsups[$this->keys['v->k'][$x]] = 1; }12
50             } else {
51                 $array[$this->keys['v->k'][$x]] ++;
52                 $this->allsups[$this->keys['v->k'][$x]] ++; }13
53             }
54
55             $stable[$i][$this->keys['v->k'][$x]] = 1; }14
56         }
57     }
58
59     $tmp = array(); }15
60     foreach ($array as $item => $sup) { }16
61         if ($sup >= $this->minSup) { }17
62             $tmp[] = array($item); }18
63         }
64     }
65
66     $this->allthings[$this->phase] = $tmp; }19
67     $this->table = $stable;
68 } }20

```

Gambar F.1. Pembuatan Tabel Data Perhitungan *Support*

F.2. Listing Program Fungsi Penggabungan Data Menjadi Array

```
private function scan($arr, $implodeArr = '') { }1
    $scr = 0; }2

    if ($implodeArr) { }3
        if (isset($this->allsups[$implodeArr])) { }4
            return $this->allsups[$implodeArr]; }5
        }
    } else {
        sort($arr);
        $implodeArr = implode($this->delimiter, $arr); }6
        if (isset($this->allsups[$implodeArr])) { }7
            return $this->allsups[$implodeArr]; }8
        }
    }

    for ($i = 0, $num = count($this->table); $i < $num; $i++) { }9
        $bool = true; }10
        for ($j = 0, $num1 = count($arr); $j < $num1; $j++) { }11
            if (!isset($this->table[$i][$arr[$j]])) { }12
                $bool = false; }13
                break;
            }
        }

        if ($bool) { }14
            $scr++; }15
        }

        $this->allsups[$implodeArr] = $scr; }16
    }
    return $scr;
} }17
```

Gambar F.2. Penggabungan Data Menjadi Array

F.3. Listing Program Fungsi Mengkombinasikan Array

```

104 private function combine($arr1, $arr2) { }1
105     $result = array(); }2
106
107     for ($i = 0, $num = count($arr1); $i < $num; $i++) { }3
108         if (!isset($result['k'][$arr1[$i]])) { }4
109             $result['v'][] = $arr1[$i]; }5
110             $result['k'][$arr1[$i]] = 1; }5
111         }
112     }
113
114     for ($i = 0, $num1 = count($arr2); $i < $num1; $i++) { }6
115         if (!isset($result['k'][$arr2[$i]])) { }7
116             $result['v'][] = $arr2[$i]; }8
117             $result['k'][$arr2[$i]] = 1; }8
118         }
119     }
120
121     return $result['v']; }9
122 } }10

```

Gambar F.3. Listing Program Fungsi Mengkombinasikan Array

F.4. Listing Program Fungsi Mengubah Nama Variabel Menjadi Nama Produk

```

124 private function realName($arr) { }1
125     $result = ''; }2
126
127     for ($j = 0, $num = count($arr); $j < $num; $j++) { }3
128         if ($j) { }4
129             $result .= $this->delimiter; }5
130         }
131
132         $result .= $this->mdl_paket->ambilnama($arr[$j]); }6
133     }
134
135     return $result; }7
136 } }8

```

Gambar F.4. Mengubah Nama Variabel Menjadi Nama Produk

F.5. *Listing* Program Fungsi Memeriksa Kembali Kesesuaian Data dengan Aturan Asosiasi

```

138 private function checkRule($a, $b) { }1
139     for ($i = 0, $a_num = count($a); $i < $a_num; $i++) { }2
140         for ($j = 0, $b_num = count($b); $j < $b_num; $j++) { }3
141             if ($a[$i] == $b[$j]) { }4
142                 return false; }5
143         }
144     }
145 }
146
147 return true; }6
148 } }7

```

Gambar F.5. Memeriksa Kembali Kesesuaian Data dengan Aturan Asosiasi

F.6. *Listing* Program Fungsi Menghitung Nilai *Confidence*

```

150 private function confidence($sup_a, $sup_ab) { }1
151     return round(($sup_ab / $sup_a) * 100, 2); }2
152 } }3

```

Gambar F.6. Menghitung Nilai *Confidence*

F.7. *Listing* Program Mengurutkan Data Sesuai Nilai *Support* dan *Confidence*

```

154 private function subsets($items) { }1
155     $result = array(); }2
156
157     for ($i = 0, $num = count($items), $members = pow(2, $num); $i < $members; $i++) { }3
158         $b = sprintf("%0" . $num . "b", $i); }4
159         $tmp = array();
160         for ($j = 0; $j < $num; $j++) { }5
161             if ($b[$j] == '1') { }6
162                 $tmp[] = $items[$j]; }7
163             }
164         }
165
166         if ($tmp) { }8
167             sort($tmp);
168             $result[] = $tmp; }9
169         }
170     }
171
172     return $result; }10
173 } }11

```

Gambar F.7. Mengurutkan Data Sesuai Nilai *Support* dan *Confidence*

F.8. Listing Program Fungsi Menentukan *Frequent Itemset*

```

176 private function FrequentItemset($db) { }1
177     $chis->makeTable($db); }2
178     while ($i) { }3
179         if ($chis->phase == $chis->maxPhase) { }4
180             break; }5
181     }
182     $or = 0;
183     for ($i = 0, $sum = count($chis->allthings[$chis->phase]); $i < $sum; $i++) { }6
184         for ($j = $i; $j < $sum; $j++) { }7
185             if ($i == $j) { }8
186                 continue; }9
187             }
188             $item = $chis->combine($chis->allthings[$chis->phase][$i], $chis->allthings[$chis->phase][$j]); }10
189             sort($item);
190             $implodeStr = implode($chis->delimiters, $item);
191             if (!isset($chis->freqItems[$implodeStr])) { }11
192                 $sup = $chis->count($item, $implodeStr); }12
193                 if ($sup > $chis->minSup) { }13
194                     $chis->addItems[$implodeStr] = $sup; }14
195                     $chis->freqItems[$implodeStr] = $sup;
196                 }
197             }
198         }
199     }
200     }
201     }
202     }
203     if ($or <= 1) { }16
204         break; }17
205     }
206     $chis->phase++; }18
207 }
208 }
209 }
210 }
211 foreach ($chis->freqItems as $k => $v) { }19
212     $arr = explode($chis->delimeter, $k); }20
213     $sum = count($arr);
214     if ($sum > 2) { }21
215         $subsets = $chis->subsets($arr); }22
216         $sum = count($subsets);
217         for ($i = 0; $i < $sum; $i++) { }23
218             if (count($subsets[$i]) < $sum) { }24
219                 unset($chis->freqItems[implode($chis->delimiters, $subsets[$i])]); }25
220             } else {
221                 break; }26
222             }
223         }
224     }
225 } }27

```

Gambar F.8. Menentukan *Frequent Itemset*

F.9. Listing Program Fungsi Proses Data Masukan

```

228 public function process($db) { }1
229     $checked = $result = array(); }2
230     $this->freqItemssets($db);
231
232     foreach ($this->freqItems as $k => $v) { }3
233         $subsets = $this->subsets(explode($this->delimiter, $k)); }4
234
235         for ($i = 0, $num = count($subsets); $i < $num; $i++) { }5
236             for ($j = 0; $j < $num; $j++) { }6
237                 if ($this->checkRule($subsets[$i], $subsets[$j])) { }7
238                     $n1 = $this->realName($subsets[$i]);
239                     $n2 = $this->realName($subsets[$j]);
240
241                     $scan = $this->scan($this->combine($subsets[$i], $subsets[$j])); }8
242                     $c1 = $this->confidence($this->scan($subsets[$i]), $scan);
243                     $c2 = $this->confidence($this->scan($subsets[$j]), $scan);
244
245                     if ($c1 >= $this->minConf) { }9
246                         $result[$n1][$n2] = $c1; }10
247                     }
248
249                     if ($c2 >= $this->minConf) { }12
250                         $result[$n2][$n1] = $c2; }13
251                     }
252
253                     $checked[$n1 . $this->delimiter . $n2] = 1; }14
254                     $checked[$n2 . $this->delimiter . $n1] = 1; }14
255                 }
256             }
257         }
258     }
259
260     return $this->rules = $result; }15
261 } }16

```

Gambar F.9. Proses Data Masukan

F.10. Listing Program Fungsi Mengambil Hasil *Frequent Itemset*

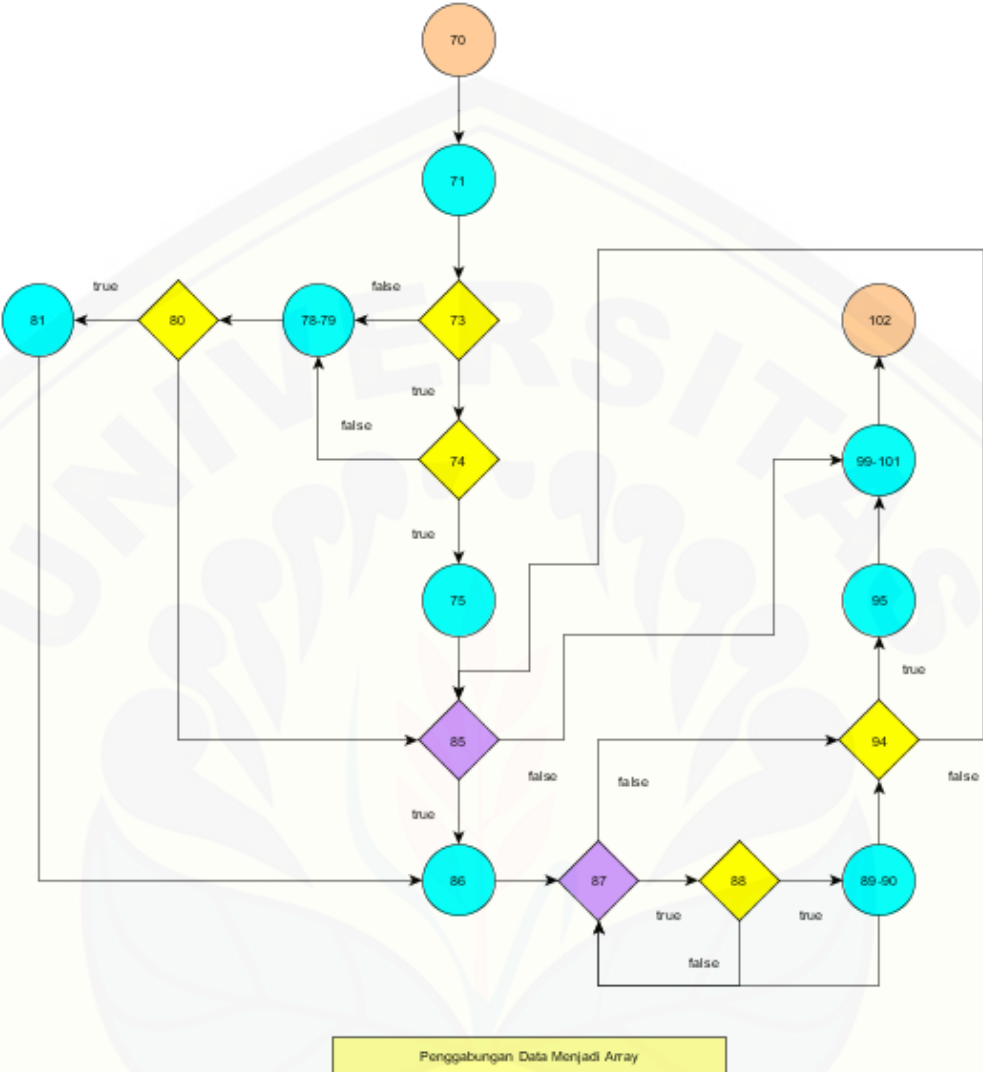
```
263 public function getFreqItemsets() ( }1
    $result = array(); }2
265
    foreach ($this->freqItemsets as $k => $v) { }3
267         $tmp = array();
268         $tmp['sup'] = $this->allsups[$k];
269         $k = explode($this->delimiter, $k); }4
270
271         for ($i = 0, $num = count($k); $i < $num; $i++) { }5
272             $tmp[] = $this->realName($k[$i]); }6
273         }
274
275         $result[] = $tmp; }7
276     }
277
278     return $result; }8
279 } }9
```

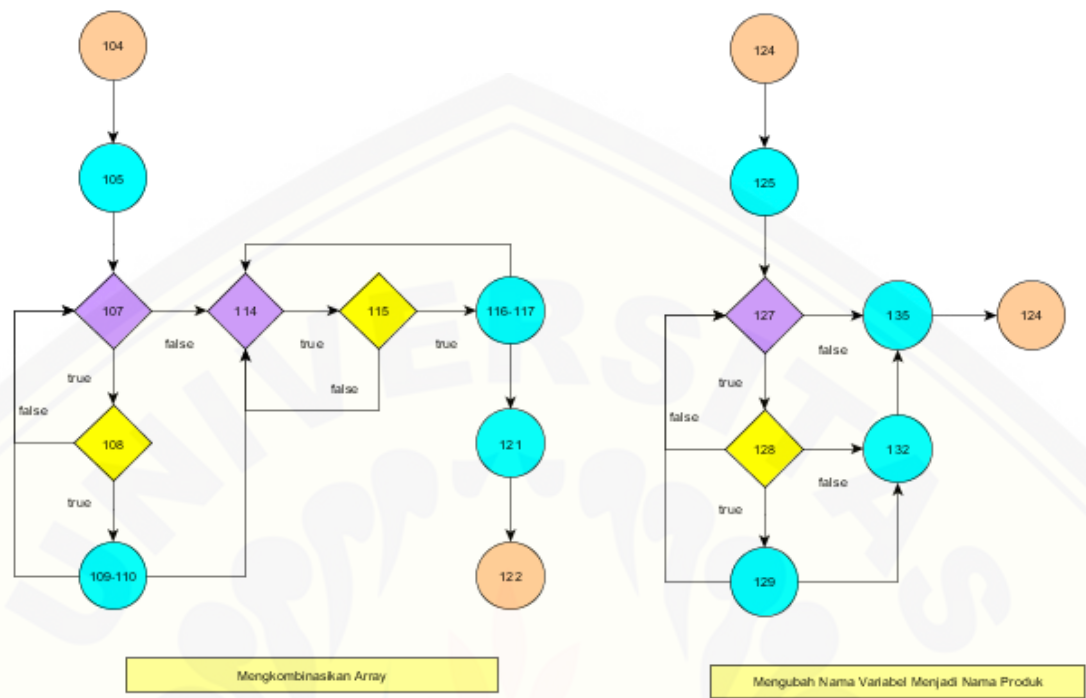
Gambar F.10. Mengambil Hasil *Frequent Itemset*

F.11. Listing Program Fungsi Mengambil Hasil Aturan Asosiasi Final

```
281 public function getAssociationRules() ( }1
282     return $this->rules; }2
283 } }3
```

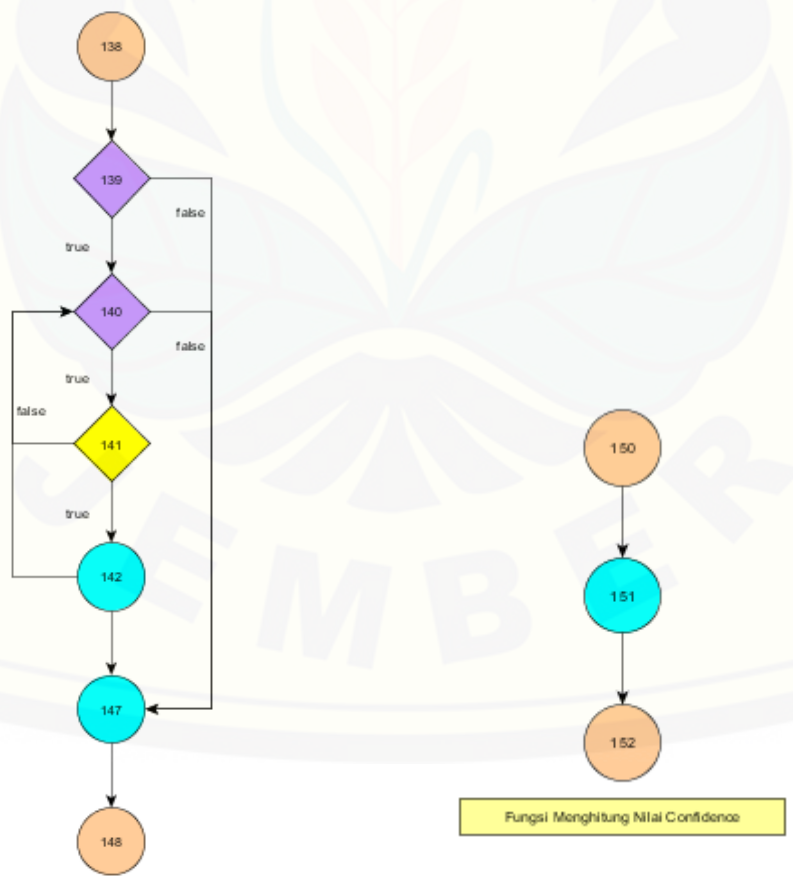
Gambar F.11. Mengambil Hasil Aturan Asosiasi Final





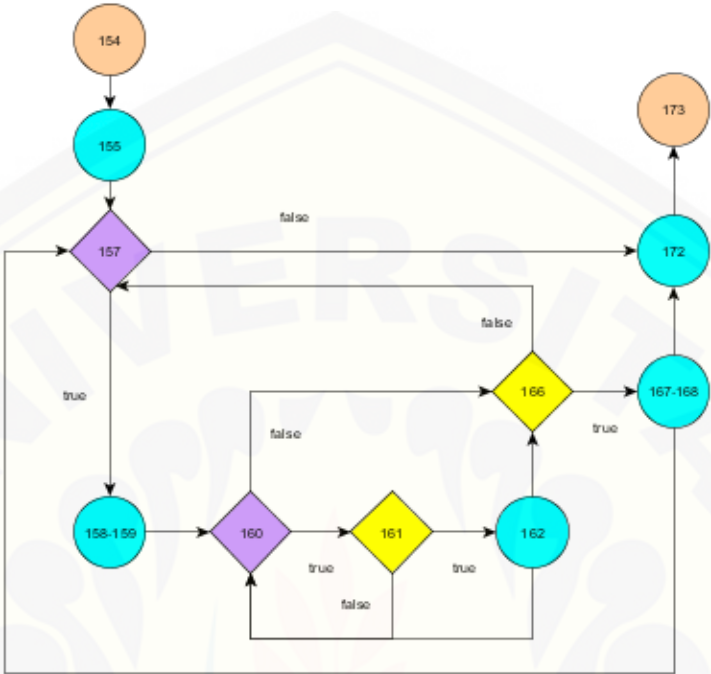
Mengkombinasikan Array

Mengubah Nama Variabel Menjadi Nama Produk

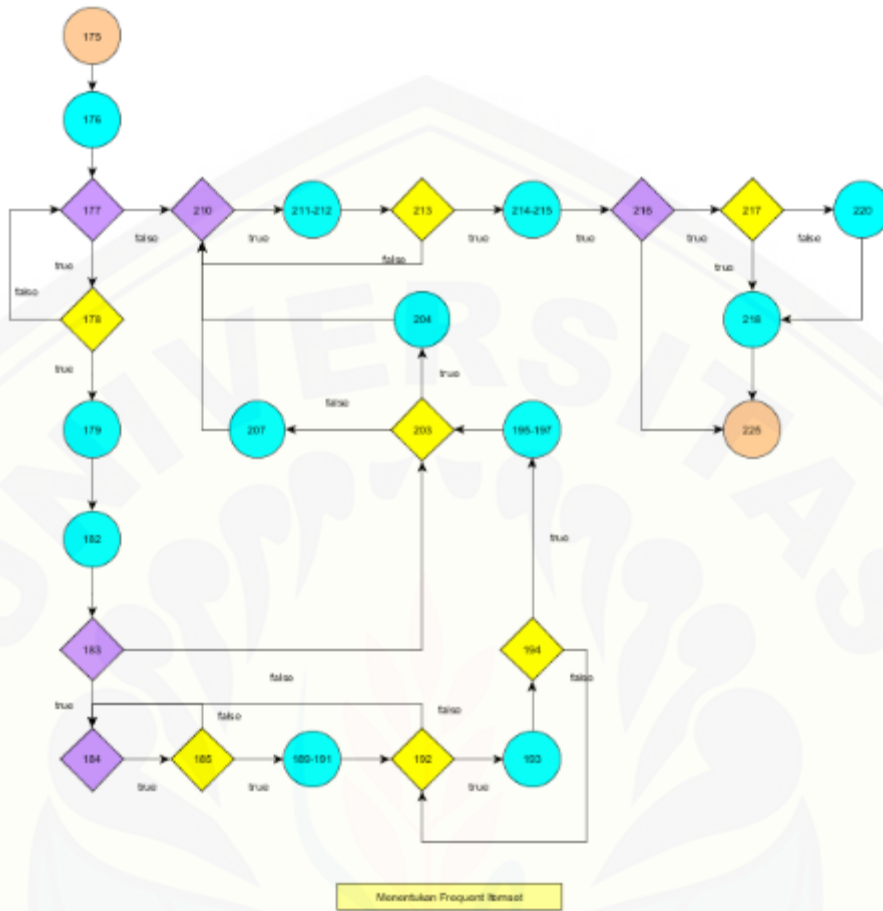


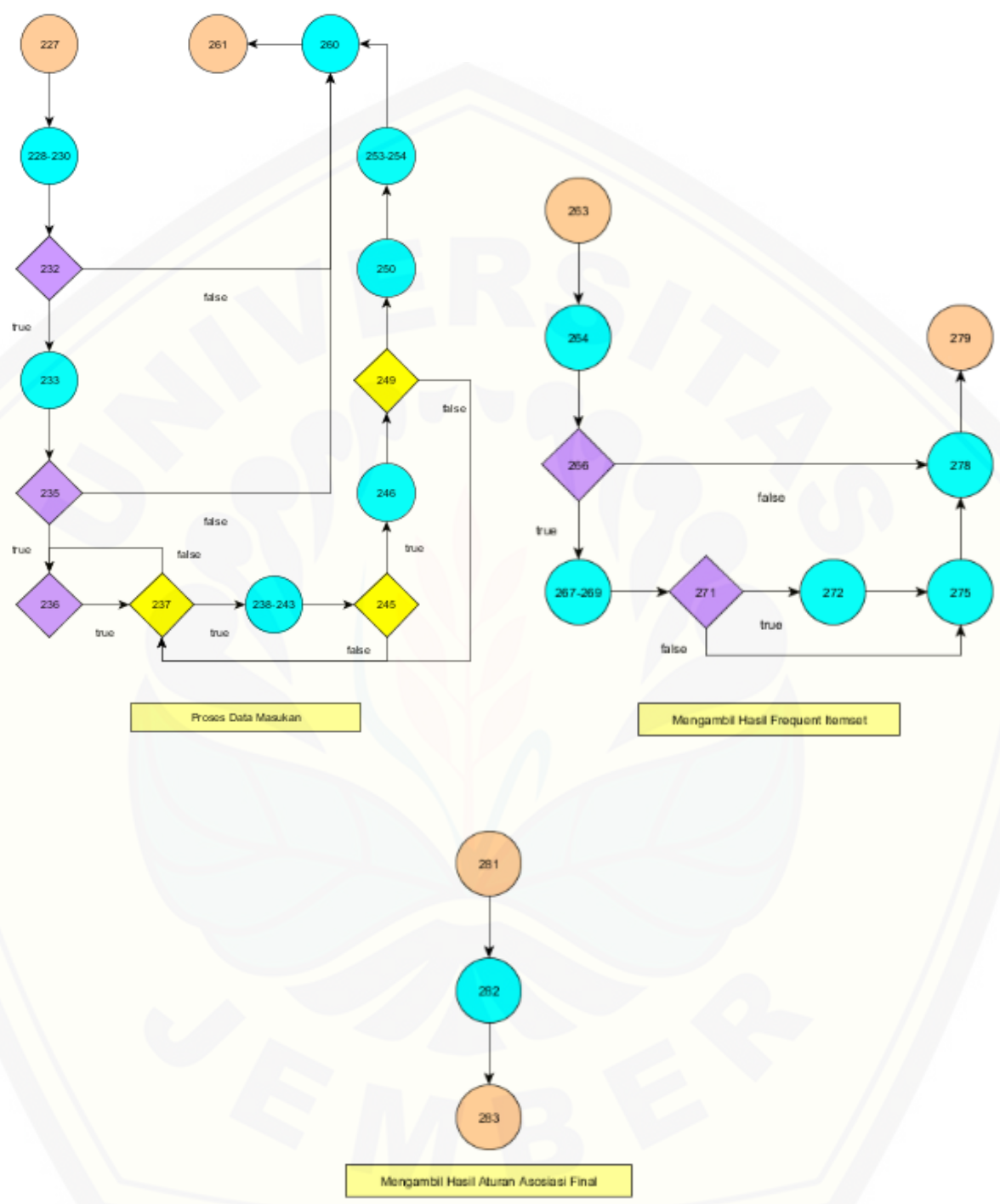
Memeriksa Kembali Kesesuaian Data dengan Aturan Asosiasi

Fungsi Menghitung Nilai Confidence



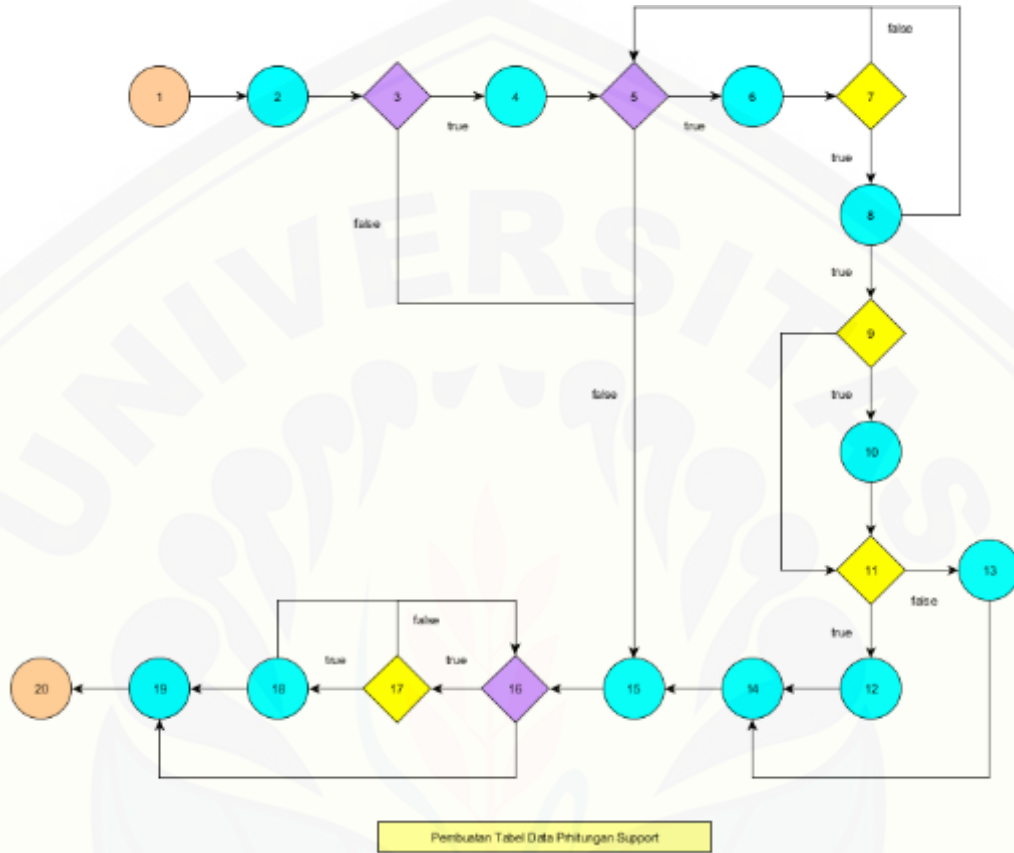
Mengurutkan Data Sesuai Nilai Support dan Confidence

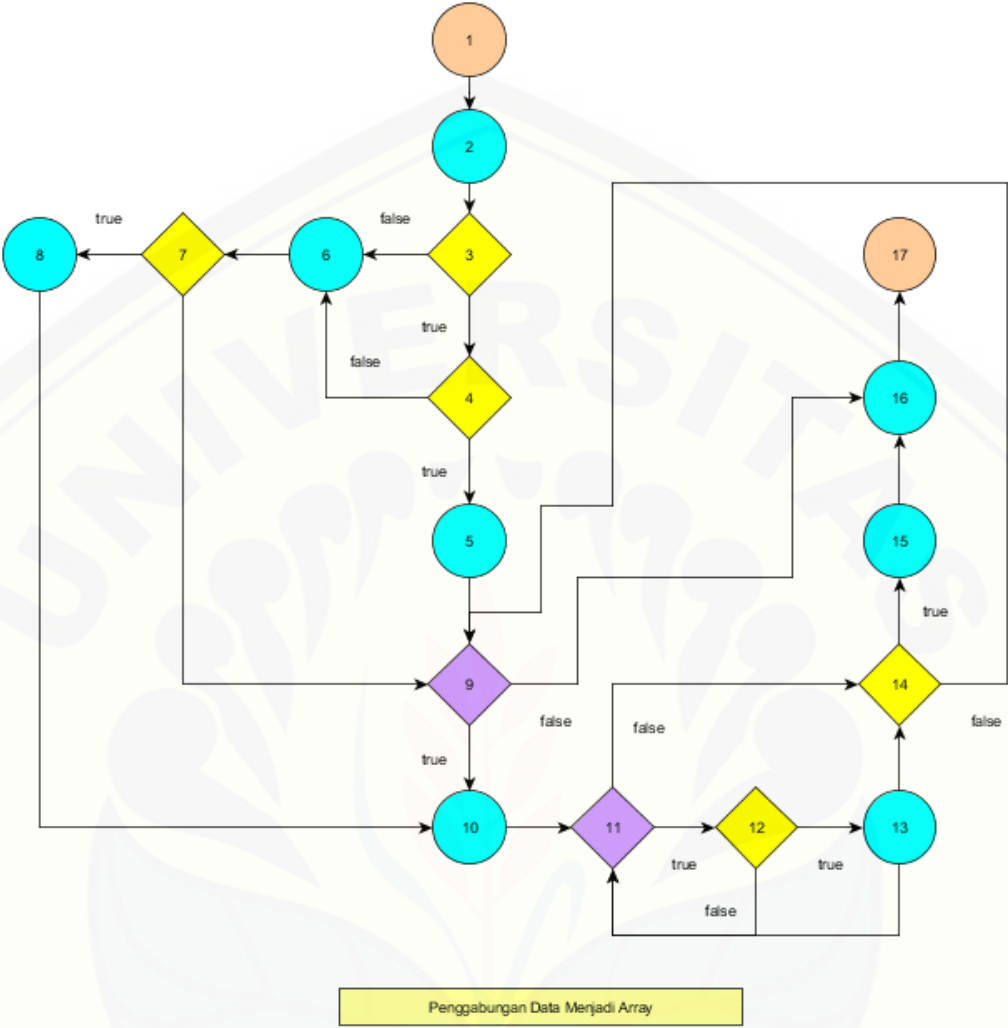


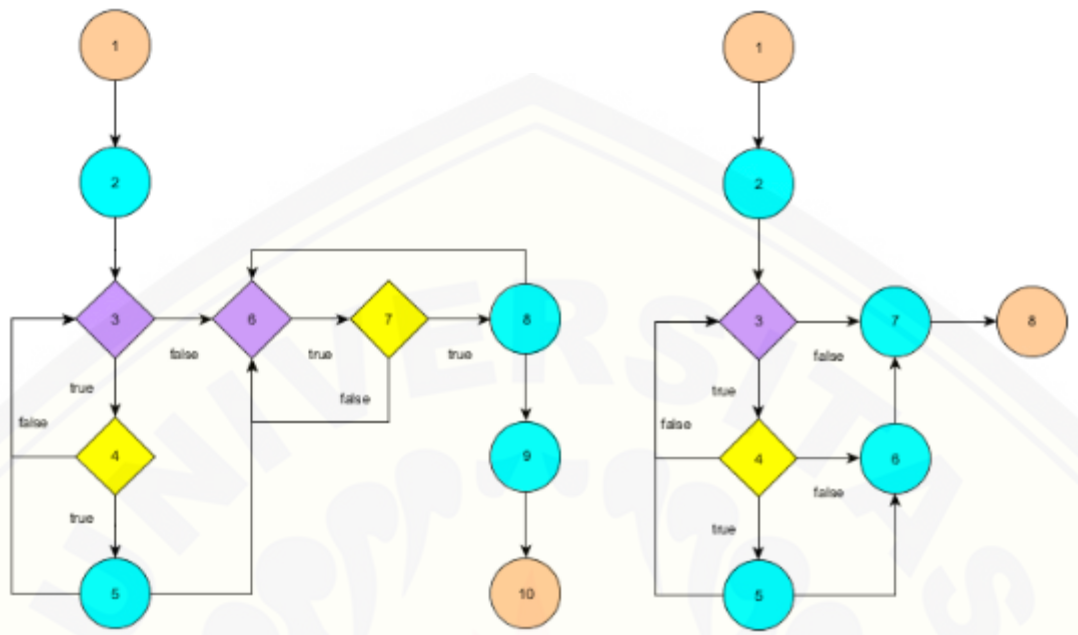


Gambar G.1. Diagram Alir *Library* Algoritma Apriori *My_lib*

H. Grafik Alir *Library* Algoritma Apriori *My_lib*

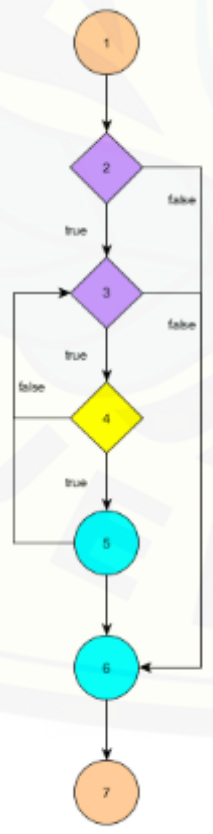




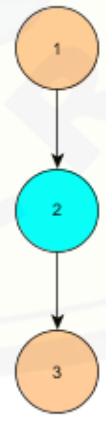


Mengkombinasikan Array

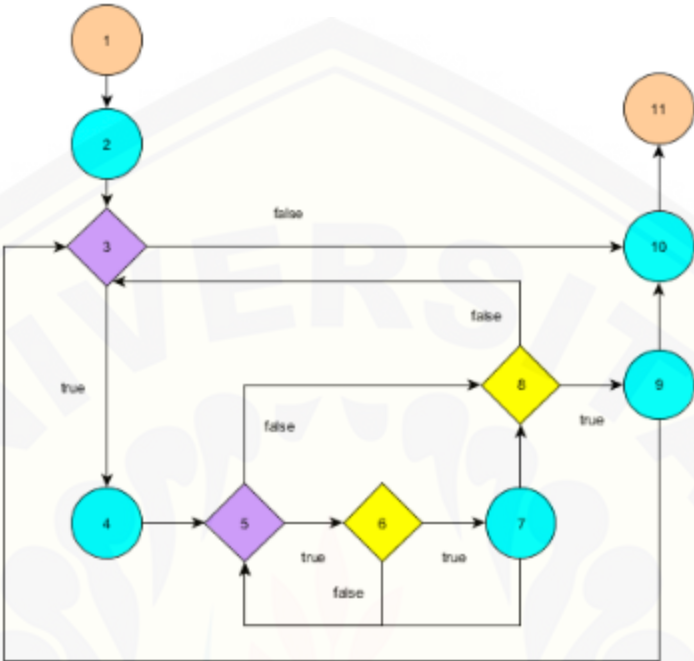
Mengubah Nama Variabel Menjadi Nama Produk



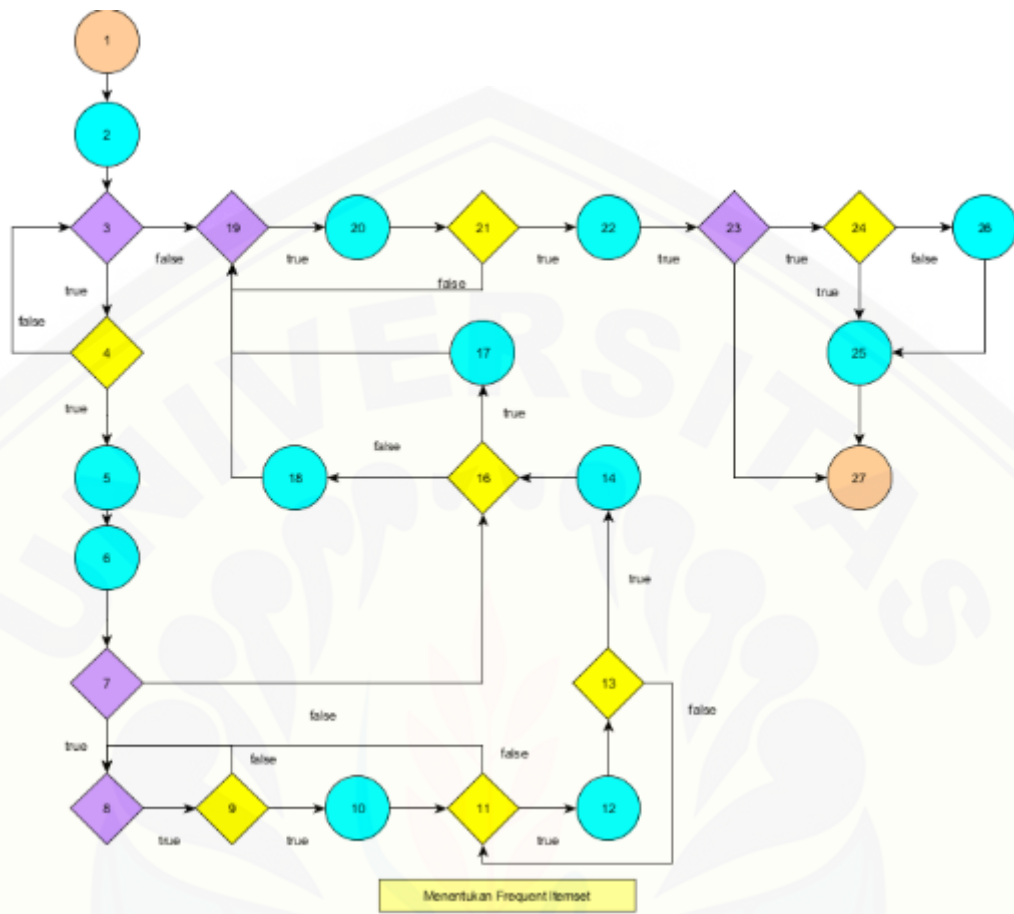
Memeriksa Kembali Kesesuaian Data dengan Aturan Asosiasi

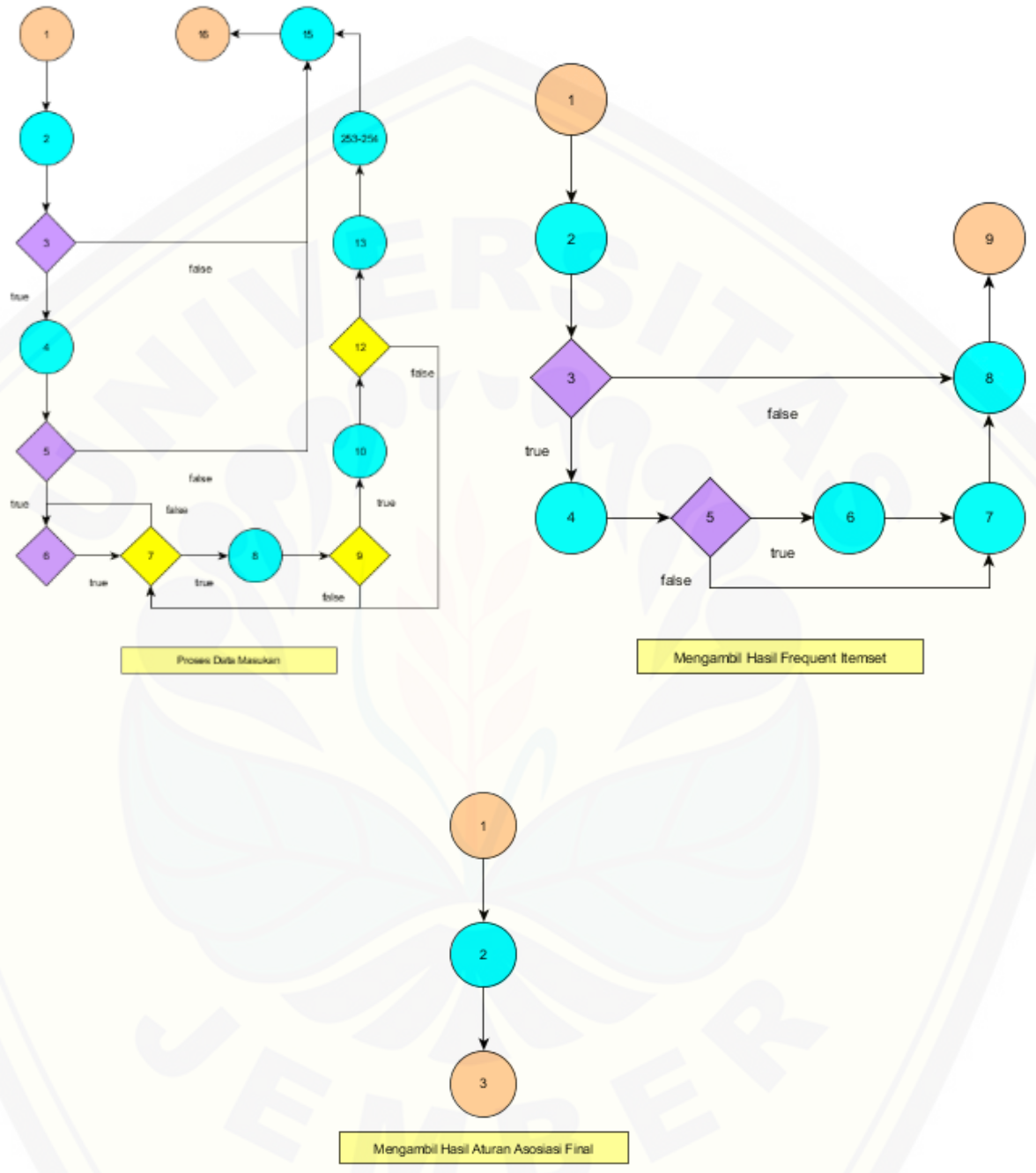


Menghitung Nilai Confidence



Mengurutkan Data Sesuai Nilai Support dan Confidence





Gambar H.1. Grafik Alir Library Algoritma Apriori My_lib

I. Pengujian *Black Box*

Tabel I.1. Pengujian Black Box

No.	Fitur	Aksi	Hasil	Kesimpulan	
				Berhasil	Gagal
1.	Login	Mengakses halaman sistem	Menampilkan halaman Login	✓	
		Mengisi kolom isian username dan password dan menekan tombol "Login"	Menampilkan halaman Dashboard Admin Panel	✓	
		Mengisi kolom isian username dan password yang tidak sesuai dengan database dan menekan tombol "Login"	Menampilkan pesan "Invalid username or password. Please try again."	✓	
		Tidak mengisi kolom isian username dan password dan menekan tombol "Login"	Menampilkan pesan "Please fill out this field" di bawah kolom isian.	✓	
2.	Logout	Klik panah dropdown pada pojok kanan atas kemudian menekan tombol "Logout"	Menampilkan halaman Login	✓	
3.	Menambah data produk	Mengisi data produk pada kolom isian data produk di halaman Add Product kemudian menekan tombol "Submit"	Menampilkan halaman Update Product yang berisi daftar data produk	✓	
4.	Menambah data produk	Memilih submenu "Add Product"	Menampilkan halaman form Tambah Produk disertai tombol "Submit", "Clear", "Update Color",	✓	

No.	Fitur	Aksi	Hasil	Kesimpulan	
				Berhasil	Gagal
		Mengisi data produk dan menekan tombol "Submit"	"Update Pic", dan "Kategori Produk" Menyimpan ke basis data dan menampilkan halaman Update Product yang berisi daftar data produk	✓	
		Format data isian tidak sesuai	Menampilkan pesan error	✓	
		Kolom teks masih kosong atau belum diisi	Menampilkan pesan error	✓	
5.	Mengubah Data Produk	Memilih submenu "Update Products"	Menampilkan halaman Update Product yang berisi daftar data produk, disertai tombol "Edit", "Delete", dan "Back to Manage"	✓	
		Klik tombol "Edit" pada salah satu data produk	Menampilkan halaman Update Product yang berisi kolom teks yang berisi data produk, disertai tombol "Submit" dan "Clear"	✓	
		Mengubah data produk pada halaman Update Product kemudian menekan tombol "Submit"	Menyimpan ke basis data dan menampilkan halaman Update Product	✓	
		Format data isian tidak sesuai	Menampilkan pesan error	✓	
		Kolom teks masih kosong atau belum diisi	Menampilkan pesan error	✓	

No.	Fitur	Aksi	Hasil	Kesimpulan	
				Berhasil	Gagal
a.	Menambah dan mengubah warna pilihan	Menekan tombol “Update Colors”	Menampilkan halaman Update Colors yang berisi kolom teks warna produk, pilihan warna yang ada, disertai tombol "Submit", "Delete", dan "Back to Manage"	✓	
		Mengisi kolom teks warna dan menekan tombol “Submit”	Menambahkan daftar warna produk	✓	
		Menekan tombol “Delete” pada salah satu pilihan warna	Menghapus pilihan warna dari basis data dan menampilkan daftar pilihan warna produk	✓	
b.	Menambah atau mengubah gambar produk	Menekan tombol “Upload Pic” dan memilih gambar, kemudian menekan tombol “Upload”	Menampilkan halaman Upload Product Picture disertai dengan gambar produk yang baru	✓	
		Ukuran file yang dipilih terlalu besar dari ukurna yang ditentukan	Menampilkan pesan error	✓	
c.	Menambah kategori produk	Menekan tombol “Kategori Produk”	Menampilkan halaman Manage Kategori Produk	✓	
		Memilih salah satu kategori produk kemudian menekan tombol “Submit”	Menampilkan daftar Kategori yang Dipilih	✓	
		Menekan tombol “Selesai”	Kembali ke halaman Update Product	✓	

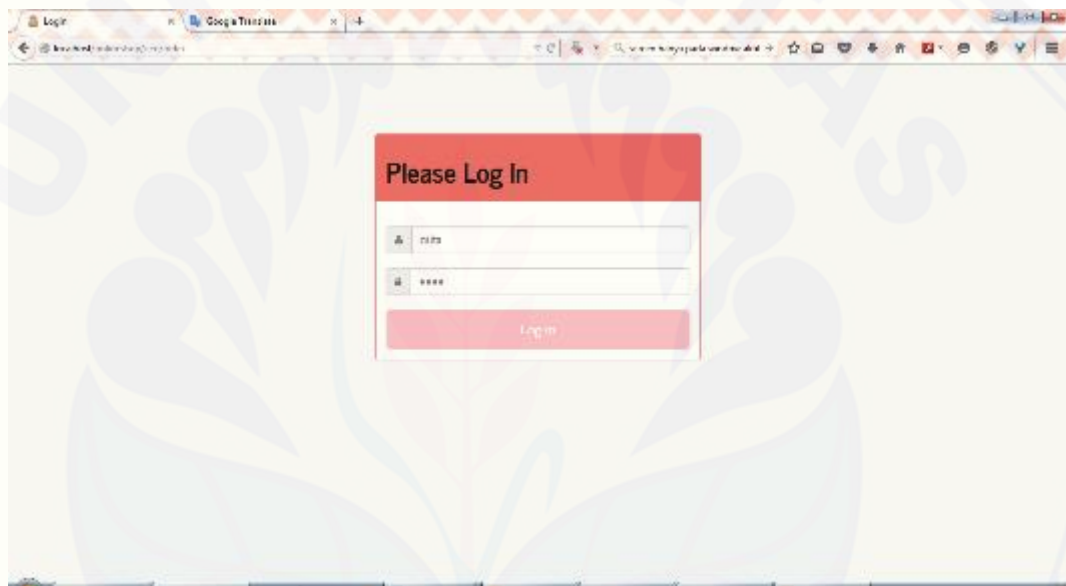
No.	Fitur	Aksi	Hasil	Kesimpulan		
				Berhasil	Gagal	
6.	Menghapus data produk	Klik tombol “Delete” pada salah satu produk	Menampilkan halaman Delete Produk disertai tombol “Yes” dan “No”	✓		
		Klik tombol “Yes”	Menghapus data dari basis data dan menampilkan halaman Update Product	✓		
		Klik tombol “No”	Kembali menampilkan halaman Update Product	✓		
7.	Menambah data order	Mengakses sistem	Menampilkan Landing Page berisi baris produk	✓		
		Memilih salah satu produk pada halaman baris produk	Menampilkan data produk yang dipilih	✓		
		Memilih warna, memasukkan jumlah barang, kemudian menekan tombol “Beli Sekarang”	Menambah produk pada cart contents kemudian menampilkan halaman Tas Belanja disertai tombol “Update cart”, “Delete”, “Trash”, “Continue Shopping”, dan “Lanjut ke langkah selanjutnya”	✓		
		Menekan tombol “Continue Shopping”	Kembali ke halaman baris produk	✓		
		a. Mengubah jumlah barang	Ubah jumlah barang pada kolom teks Qty kemudian menekan tombol “Update Cart”	Mengubah jumlah barang pada cart contents kemudian menampilkan halaman Cart Contents	✓	
		b. Menghapus barang pada Tas Belanja	Menekan tombol “Delete” pada salah satu barang di daftar Tas Belanja	Menghapus barang yang dipilih dari cart contents kemudian menampilkan halaman Cart Contents	✓	

No.	Fitur	Aksi	Hasil	Kesimpulan	
				Berhasil	Gagal
c.	Mengubah alamat pengiriman	Menekan tombol “Langkah Selanjutnya”	Mengambil data member dari basis data dan menampilkan halaman Data Pembeli	✓	
		Mengubah data pembeli sesuai dengan alamat pengiriman kemudian klik tombol “Simpan di Akun”	Menyimpan data member ke basis data kemudian menampilkan halaman Data Pembeli	✓	
d.	Melihat rangkuman data pembeli	Menekan tombol “Lanjut ke Ongkir dan Pembayaran”	Mengambil data member dari cart contents	✓	
			Menampilkan halaman Ongkir dan Pembayaran yang berisi Rangkuman Data Pembeli dan Tas Belanja lengkap dengan jumlah ongkos kirim dan total yang harus dibayarkan	✓	
		Menekan tombol “Ubah Data Alamat”	Mengambil data dari basis data dan menampilkan halaman Data Pembeli	✓	
		Menekan tombol “Pemesanan Selesai”	Menyimpan Rangkuman Data Pembeli dan Data <i>Order</i> ke basis data	✓	
		Menampilkan halaman Konfirmasi Pembayaran dengan menampilkan nomor <i>order</i> , total bayar, dan nomor rekening <i>online shop</i>	✓		

J. Hasil Pembuatan Sistem

J.1. Tampilan Fitur *Login*

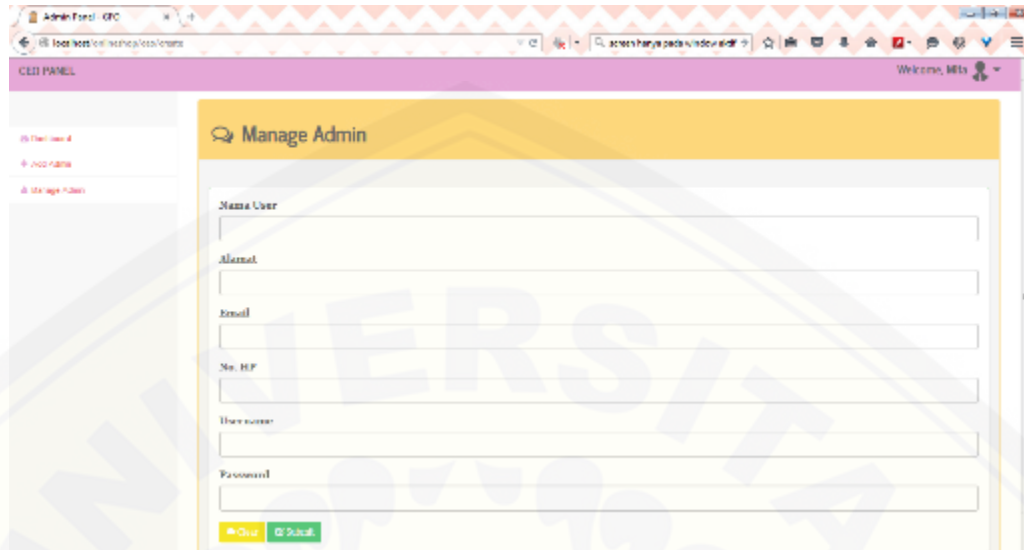
Sistem informasi penjualan GFO ini terdiri dari halaman CEO, *admin*, serta *member*. Perlu adanya fitur *login* sebagai pengaman sistem sehingga *user* harus melewati portal *login* sebelum dapat masuk ke halaman utama. Aktor yang akan dijelaskan adalah CEO yang melakukan *login* dengan memasukkan *username* mita dan *password* mita pada halaman *login* yang dapat dilihat pada Gambar J.1.



Gambar J.1. Halaman *Login* CEO

J.2. Tampilan Fitur Mengelola Data User Admin

Setelah melakukan proses *login*, kemudian CEO akan langsung dapat melihat tampilan halaman Tambah *Admin*, karena CEO hanya dapat mengelola user *admin* pada sistem. Di sebelah kiri nampak dua menu, yaitu menu *Add Admin* dan *Manage Admin*. Jika menu *Add Admin* dipilih, maka akan muncul halaman Tambah *Admin* seperti pada Gambar J.2.



The screenshot shows a web browser window with the URL 'localhost:8080/kelembukan/kelembukan/kelembukan'. The page title is 'Admin Panel' and the user is logged in as 'Welcome, Mba'. The main content area is titled 'Manage Admin' and contains a form with the following fields:

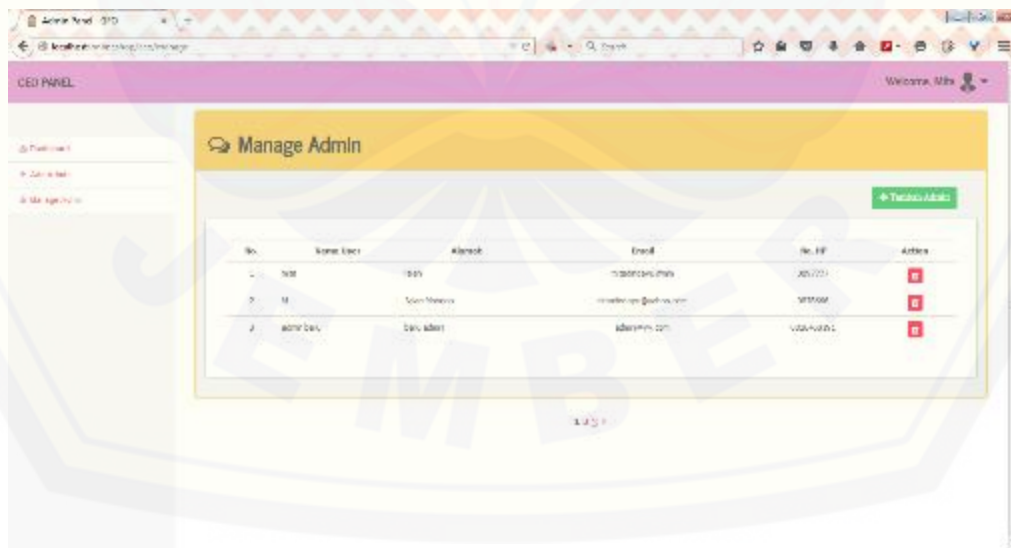
- Nama User
- Alamat
- Email
- No. HP
- User name
- Password

At the bottom of the form, there are two buttons: 'Add' (green) and 'Submit' (green).

Gambar J.2. Halaman Tambah *Admin*

J.3. Tampilan Fitur Mengelola *Admin*

Halaman *Manage Admin* akan muncul seperti pada Gambar J.3 setelah user memilih menu *Manage Admin* pada kolom menu. Pada halaman ini akan terlihat data *admin* yang sudah ditambahkan, yaitu nama *user*, alamat, *email*, nomor *handphone*, serta tombol hapus untuk menghapus data yang dipilih.



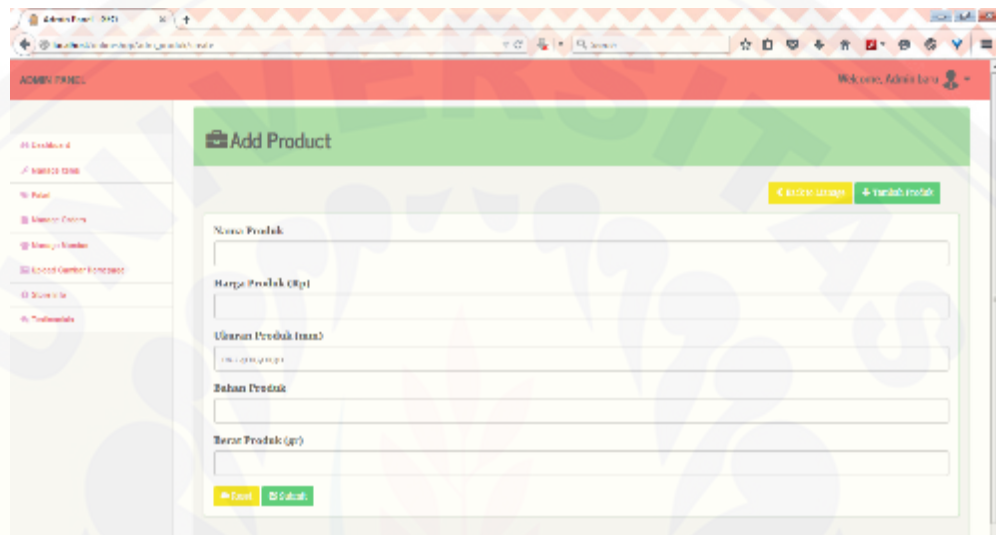
The screenshot shows the 'Manage Admin' page with a table of admin users. The table has the following columns: No., Nama User, Alamat, Email, No. HP, and Action. There are three rows of data and a 'Tambah Admin' button at the top right.

No.	Nama User	Alamat	Email	No. HP	Action
1	test	test	test@kelembukan.com	08123456789	[Delete]
2	test	test	test@kelembukan.com	08123456789	[Delete]
3	admin baru	Diku Admin	admin@kelembukan.com	08123456789	[Delete]

Gambar J.3. Halaman *Manage Admin*

Jika tombol “*delete*” pada kolom *Action* ditekan, maka akan muncul halaman konfirmasi hapus data, sedangkan jika tombol “*tambah admin*” ditekan, maka akan muncul halaman *Tambah Admin*.

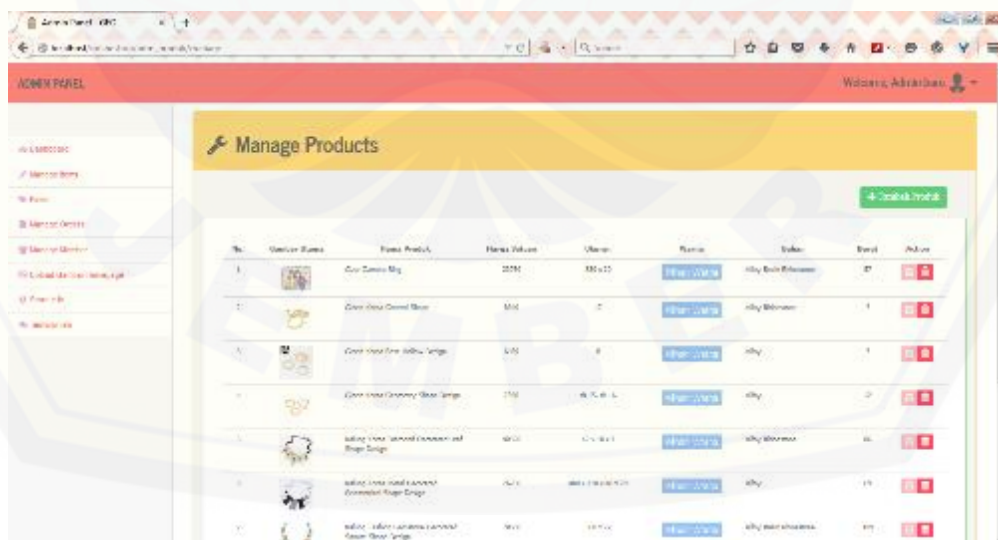
J.4. Tampilan Fitur Tambah Data Produk



The screenshot shows the 'Add Product' form in the Admin Panel. The form is titled 'Add Product' and has a green header. It contains several input fields: 'Nama Produk', 'Harga Produk (Rp)', 'Ukuran Produk (mm)', 'Bahan Produk', and 'Berat Produk (gr)'. There are also two buttons at the bottom: 'Simpan' (Save) and 'Batal' (Cancel). The left sidebar shows the 'ADMIN PANEL' navigation menu with options like Dashboard, Laporan, and Produk.

Gambar J.4. Tampilan Fitur Tambah Produk

J.5. Tampilan Fitur Ubah Data Produk

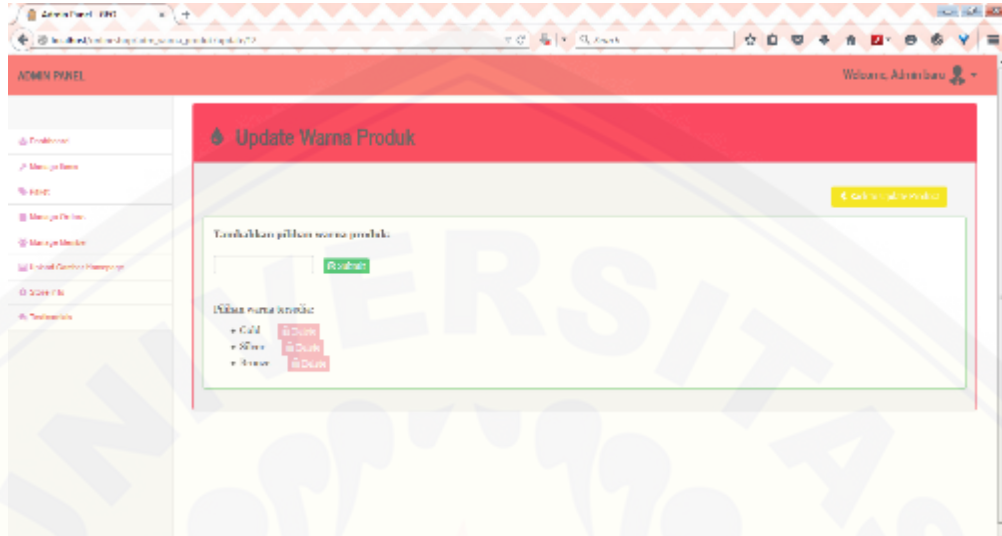


The screenshot shows the 'Manage Products' table in the Admin Panel. The table has a yellow header and contains a list of products. The columns are: No., Gambar, Nama Produk, Harga Pokok, Ukuran, Bahan, Satuan, Berat, and Aksi. The table contains 7 rows of product data. There is a '+ Tambah Produk' button in the top right corner of the table area. The left sidebar shows the 'ADMIN PANEL' navigation menu with options like Dashboard, Laporan, and Produk.

No.	Gambar	Nama Produk	Harga Pokok	Ukuran	Bahan	Satuan	Berat	Aksi
1		Cup Coklat 50g	2000	50x20	Plastik	kg	10	Ubah Data Hapus
2		Cup Coklat 100g	4000	100x40	Plastik	kg	10	Ubah Data Hapus
3		Cup Coklat 200g	8000	200x80	Plastik	kg	10	Ubah Data Hapus
4		Cup Coklat 500g	20000	500x200	Plastik	kg	10	Ubah Data Hapus
5		Cup Coklat 1kg	40000	1000x400	Plastik	kg	10	Ubah Data Hapus
6		Cup Coklat 2kg	80000	2000x800	Plastik	kg	10	Ubah Data Hapus
7		Cup Coklat 5kg	200000	5000x2000	Plastik	kg	10	Ubah Data Hapus

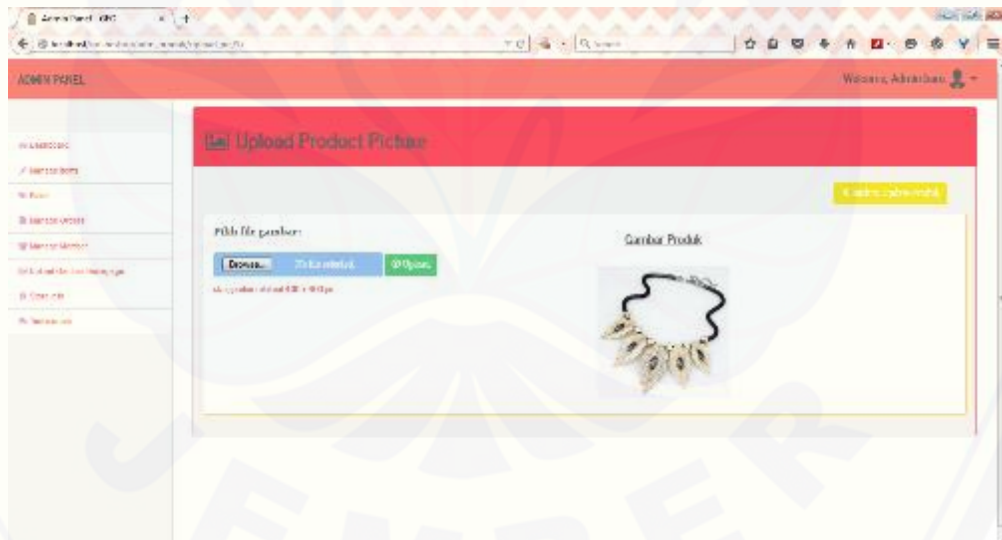
Gambar J.5. Tampilan Fitur Ubah Data Produk

J.6. Tampilan Fitur Tambah Warna Produk



Gambar J.6. Tampilan Fitur Tambah Warna produk

J.7. Tampilan Fitur Tambah Gambar Produk



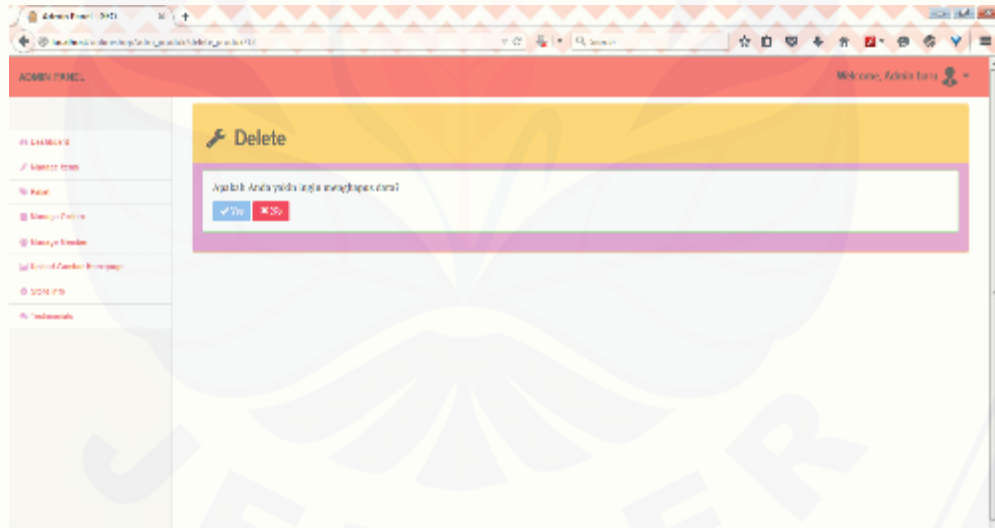
Gambar J.7. Tampilan Fitur Tambah Gambar Produk

J.8. Tampilan Ubah Kategori Produk

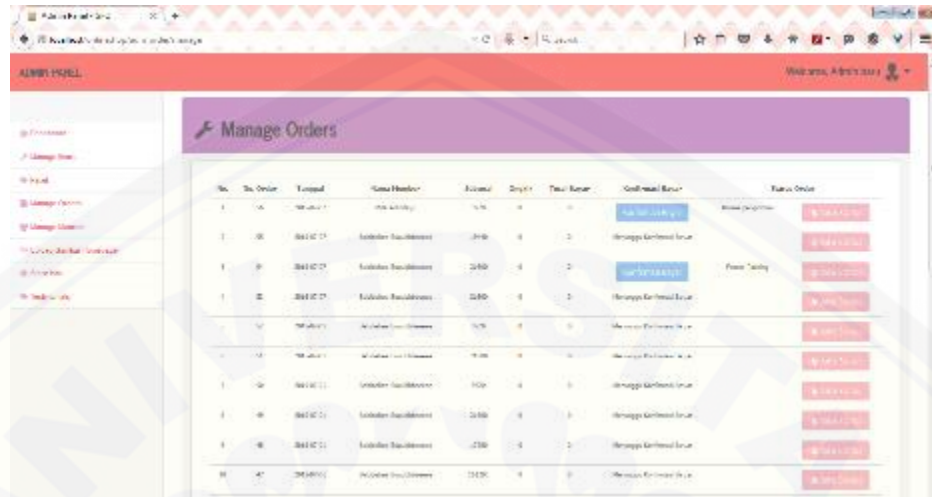


Gambar J.8. Tampilan Ubah Kategori Produk

J.9. Tampilan Fitur Hapus Data Produk



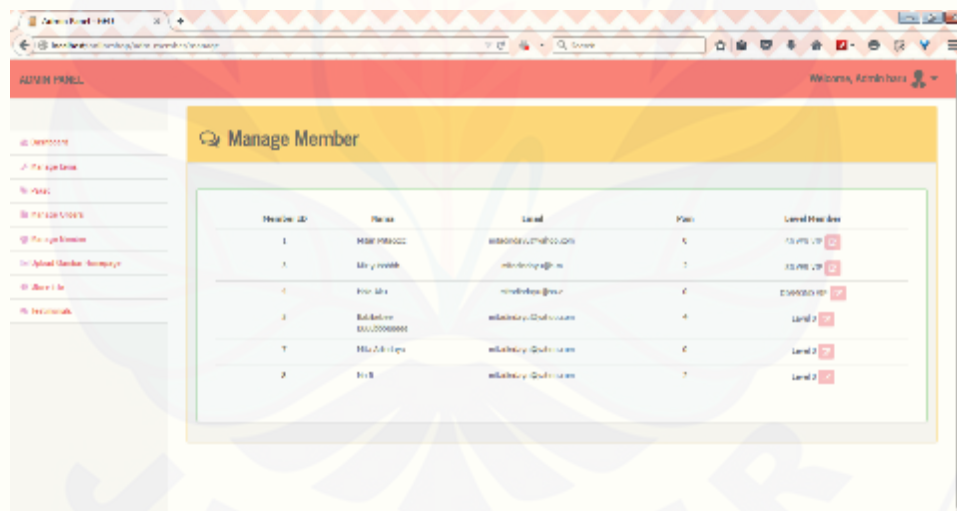
Gambar J.9. Tampilan Fitur Hapus Data Produk

J.10. Tampilan Fitur Mengelola Data *Order*

No.	No. Order	Tanggal	Nama Member	Alamat	Qty	Total Bayar	Detail Bayar	Status Order
1	00	2020-07-01	Wahid Wahid	Wahid Wahid	1	1000	1000	Batal Order
2	00	2020-07-01	Kelompok Mahasiswa	Jember	4	2000	2000	Selesai
3	00	2020-07-01	Kelompok Mahasiswa	Jember	4	2000	2000	Pembayaran
4	00	2020-07-01	Kelompok Mahasiswa	Jember	4	2000	2000	Pembayaran
5	00	2020-07-01	Kelompok Mahasiswa	Jember	4	2000	2000	Pembayaran
6	00	2020-07-01	Kelompok Mahasiswa	Jember	4	2000	2000	Pembayaran
7	00	2020-07-01	Kelompok Mahasiswa	Jember	4	2000	2000	Pembayaran
8	00	2020-07-01	Kelompok Mahasiswa	Jember	4	2000	2000	Pembayaran
9	00	2020-07-01	Kelompok Mahasiswa	Jember	4	2000	2000	Pembayaran
10	00	2020-07-01	Kelompok Mahasiswa	Jember	4	2000	2000	Pembayaran

Gambar J.10. Tampilan Fitur Mengelola Data *Order*

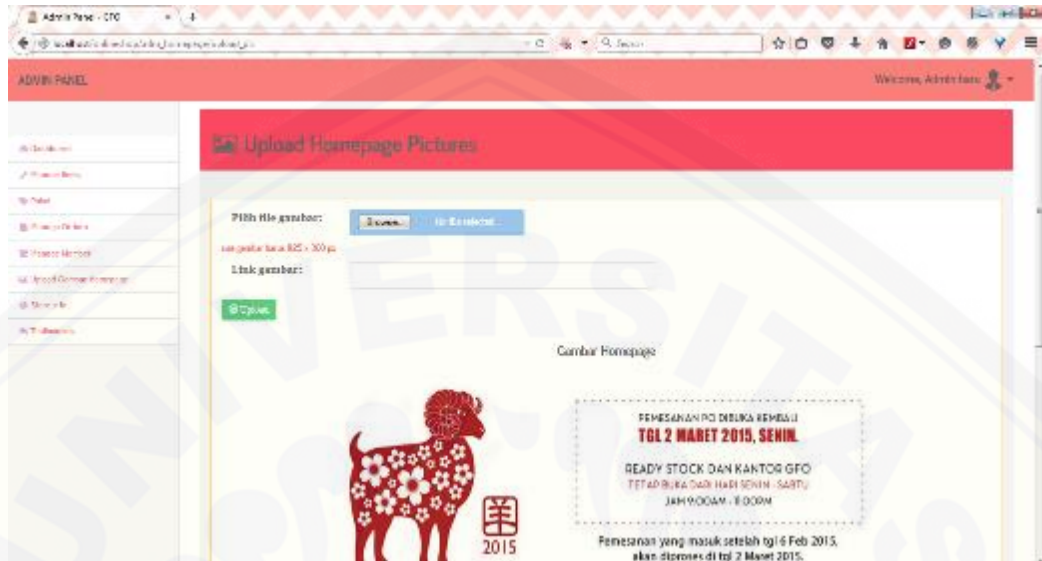
J.11. Tampilan Fitur Mengelola Data Member



Member ID	Nama	Email	Pass	Level Member
1	MAR 198000	mar198000@unsw.edu.au	5	Level 1
2	MAR 198000	mar198000@unsw.edu.au	7	Level 2
3	Fitri Sari	fitri.sari@unsw.edu.au	6	Level 3
4	Budi Santia	budi.santia@unsw.edu.au	4	Level 4
5	MAR 198000	mar198000@unsw.edu.au	6	Level 5
6	Fitri Sari	fitri.sari@unsw.edu.au	7	Level 6

Gambar J.11. Tampilan Fitur Mengelola Data Member

J.12. Tampilan Fitur Upload Gambar Homepage



Gambar J.12. Tampilan Fitur Upload Gambar Homepage

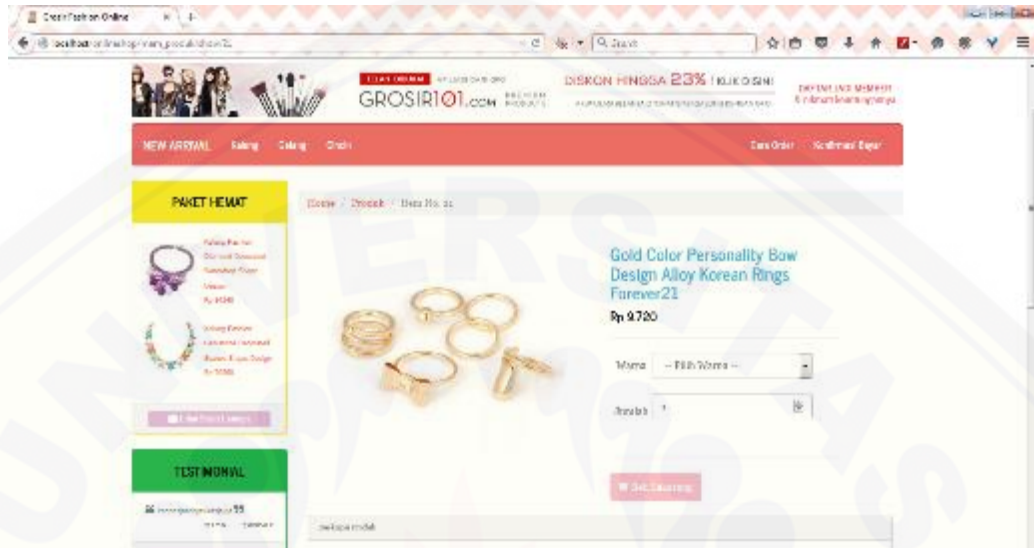
J.13. Tampilan Fitur Mengelola Testimonial



Gambar J.13. Tampilan Fitur Mengelola Testimonial

J.14. Fitur Menambah Data *Order*

J.14.1. Menampilkan Data Produk



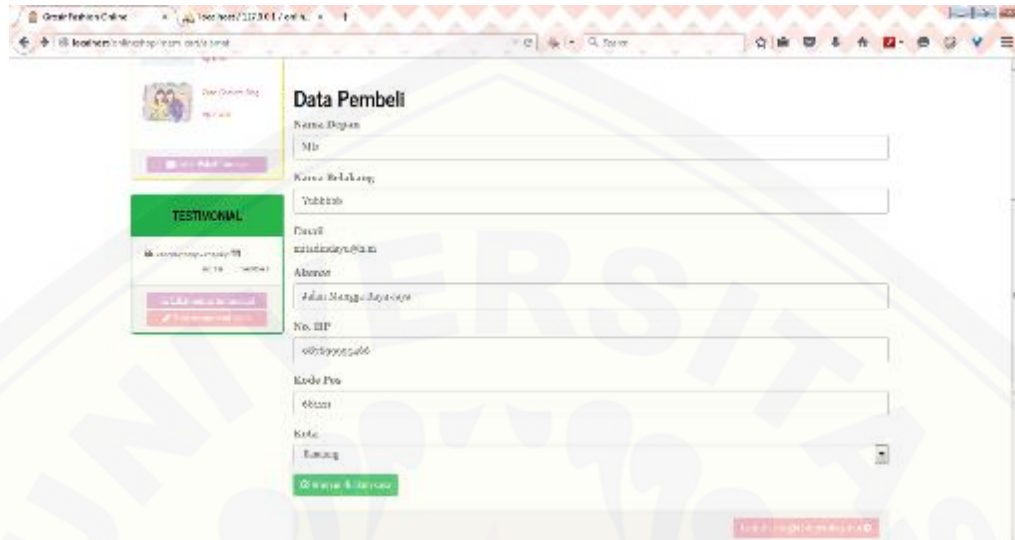
Gambar J.14. Menampilkan Data Produk

J.14.2. Fitur Mengubah Jumlah Barang



Gambar J.15. Fitur Mengubah Jumlah Barang

J.14.3. Fitur Mengubah data Pembeli



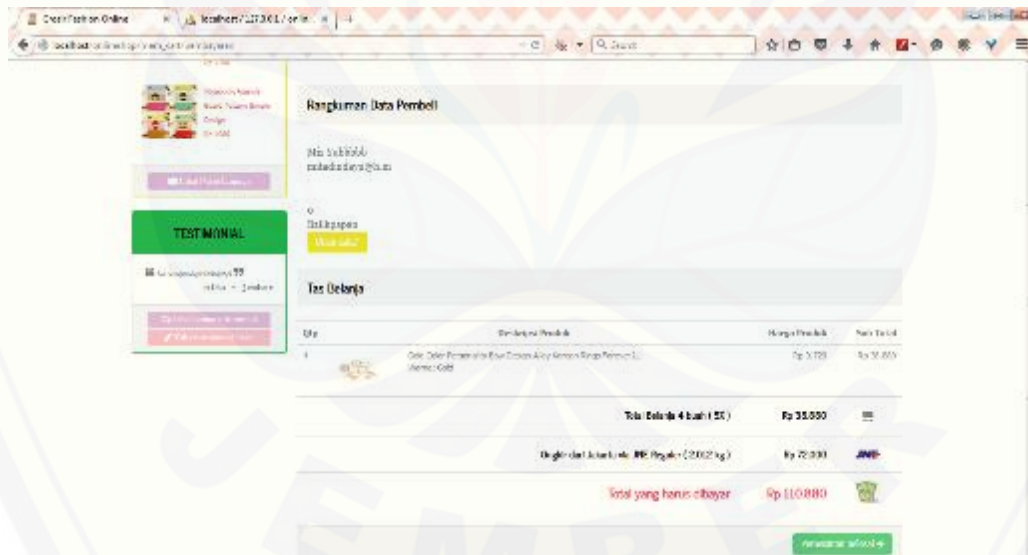
The screenshot shows a web browser window with the URL 'localhost/12301/online'. The page title is 'Galeri Fashion Online'. The main content area is titled 'Data Pembeli' and contains the following form fields:

- Nama Depan:
- Kantor:
- Yubelpost:
- Email:
- Alamat:
- No. HP:
- Kode Pos:
- Kota:

At the bottom of the form is a green button labeled 'Simpan & Kembali'.

Gambar J.16. Fitur Mengubah data Pembeli

J.15. Fitur Melihat Rangkuman Data Pembeli



The screenshot shows a web browser window with the URL 'localhost/12301/online'. The page title is 'Galeri Fashion Online'. The main content area is titled 'Rangkuman Data Pembeli' and displays the following information:

- Nama:
- Email:
- 0
- Tax Uebanya

Below the summary is a table of orders:

Id	Detail Pembeli	Harga Pembeli	Nomor Total
1	Order Pembeli dari Eran Doreen Ayu Sorens Ranga Permula... Jember, Kediri	Rp 31.723	Rp 31.880

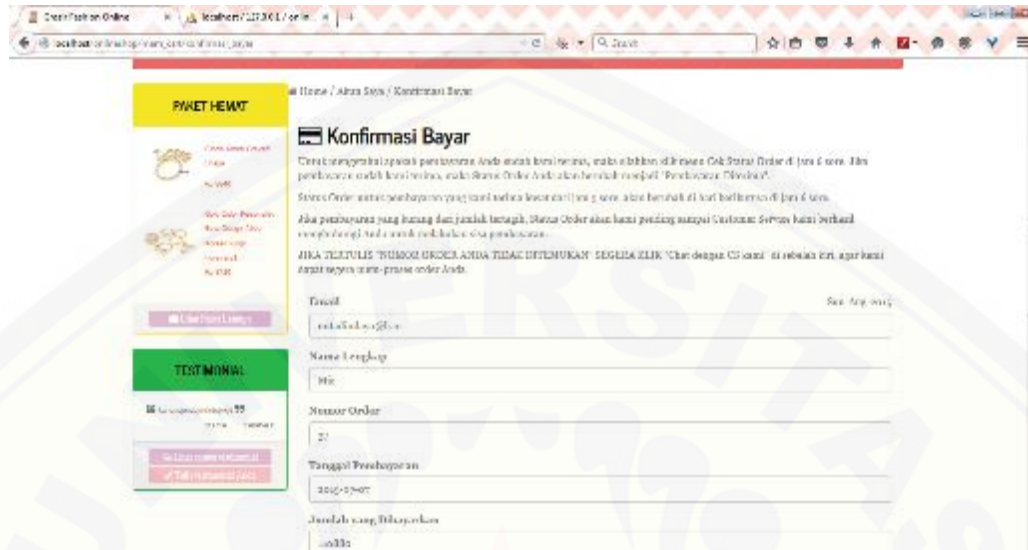
Below the table, there is a summary of the total amount to be paid:

- Total Belanja 4 buah (12%) Rp 33.550
- Unggahan & biaya lain: Rp 2.333
- Total yang harus dibayar Rp 110.880

At the bottom of the page is a green button labeled 'Perawatan & Info'.

Gambar J.17. Fitur Melihat Rangkuman Data Pembeli

J.16. Fitur Menambah Data Konfirmasi Bayar



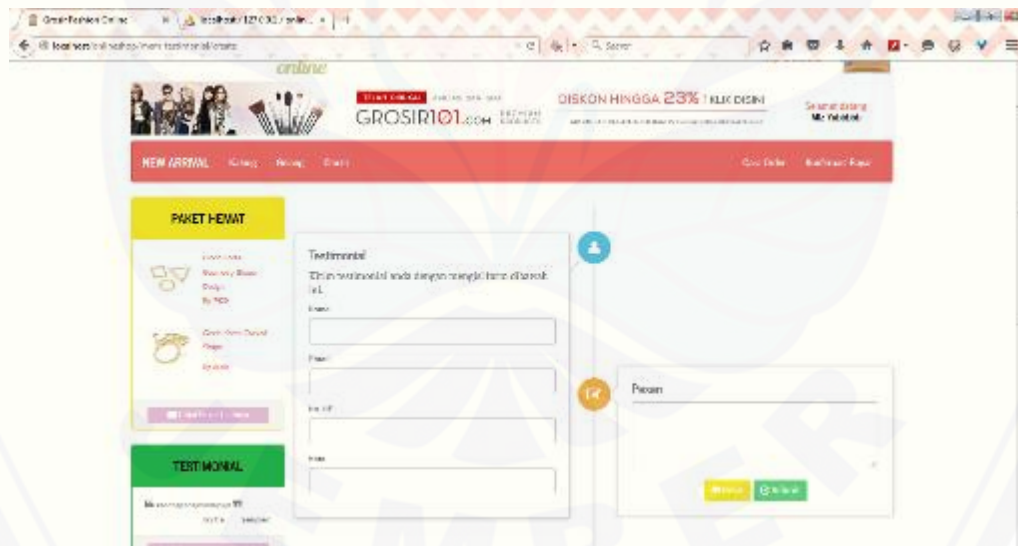
The screenshot shows a web browser window with the URL 'localhost:8080/gro/online'. The page title is 'Konfirmasi Bayar'. The main content area contains a form with the following fields:

- Tanggal:
- Waktu Pengiriman:
- Nama Lengkap:
- Alamat:
- Alamat Order:
- Tanggal Pengiriman:
- Jumlah barang dikembalikan:

On the left side, there are two sections: 'PAKET HEMAT' with a 'Lihat Detail Lainnya' button, and 'TESTIMONIAL' with a 'Lihat Testimoni Lainnya' button.

Gambar J.18. Fitur Menambah Data Konfirmasi Bayar

J.17. Fitur Menambah Testimonial



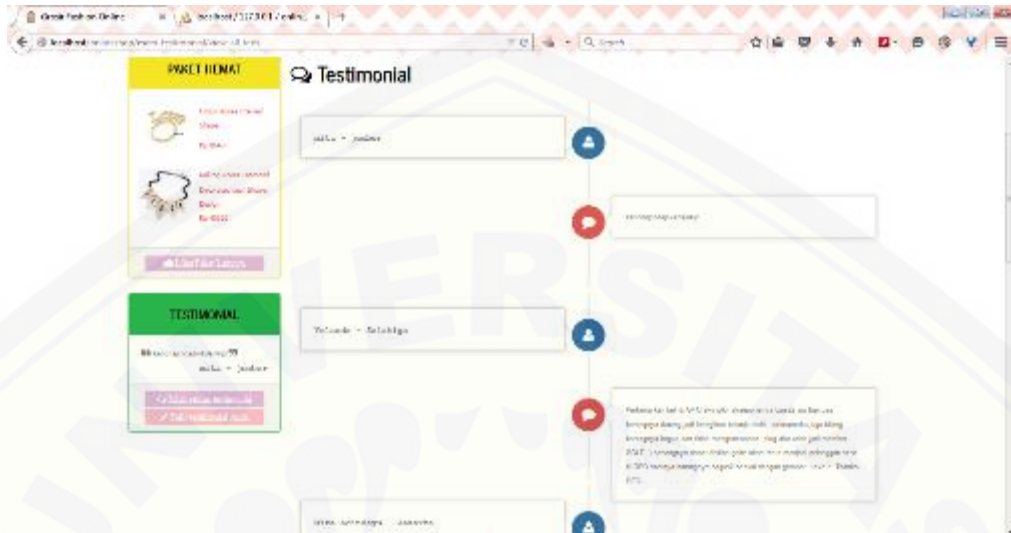
The screenshot shows a web browser window with the URL 'localhost:8080/gro/online'. The page title is 'Tambah Testimonial'. The main content area contains a form with the following fields:

- Nama:
- Alamat:
- Alamat Order:
- Tanggal Pengiriman:
- Jumlah barang dikembalikan:
- Review:
- Rating:
- File:
- Submit:
- Cancel:

On the left side, there are two sections: 'PAKET HEMAT' with a 'Lihat Detail Lainnya' button, and 'TESTIMONIAL' with a 'Lihat Testimoni Lainnya' button.

Gambar J.19. Fitur Menambah Testimonial

J.18. Fitur Melihat Semua Testimonial



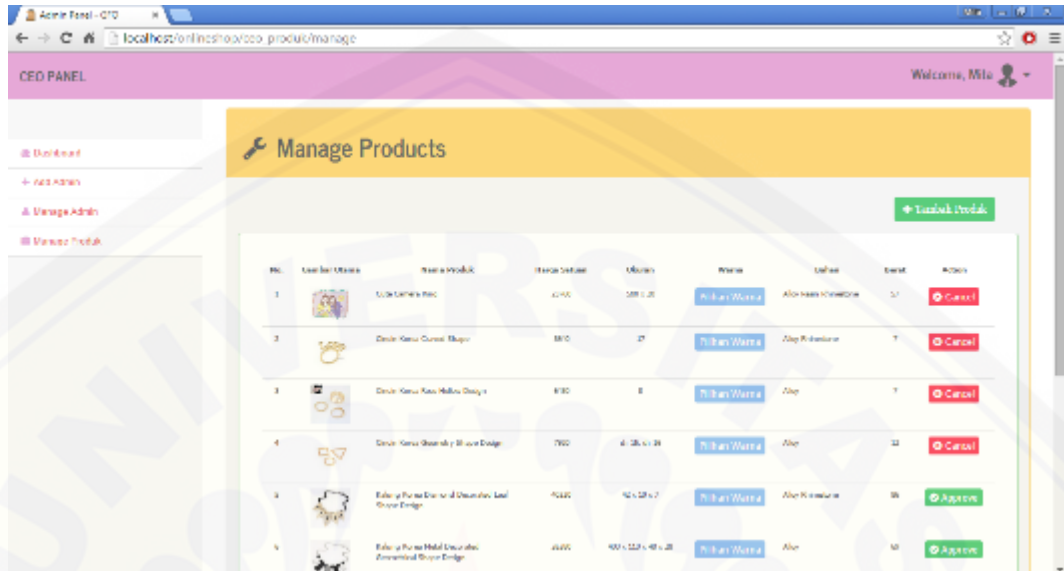
Gambar J.20. Fitur Melihat Semua Testimonial

J.19. Fitur Melihat Informasi Toko



Gambar J.21. Fitur Melihat Informasi Toko

J.20. Fitur Approval Data Produk

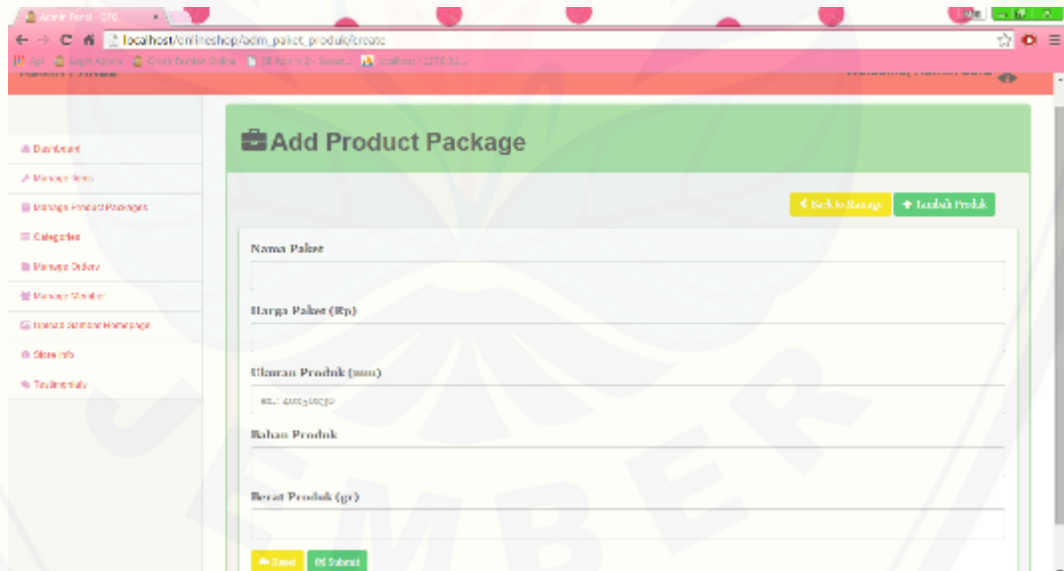


The screenshot displays the 'Manage Products' interface. A table lists the following products:

No.	User dan Status	Nama Produk	Harga Dasar	Stok	Status	Label	berat	Aksi
1	Admin	Produk Baru	1000	100	Disetujui	Produk Baru	100	Cancel
2	Admin	Produk Baru	2000	200	Ditolak	Produk Baru	200	Cancel
3	Admin	Produk Baru	3000	300	Ditolak	Produk Baru	300	Cancel
4	Admin	Produk Baru	4000	400	Ditolak	Produk Baru	400	Cancel
5	Admin	Produk Baru	5000	500	Ditolak	Produk Baru	500	Approve
6	Admin	Produk Baru	6000	600	Ditolak	Produk Baru	600	Approve

Gambar J.22. Fitur Approval Data Produk

J.21. Fitur Mengelola Paket Produk



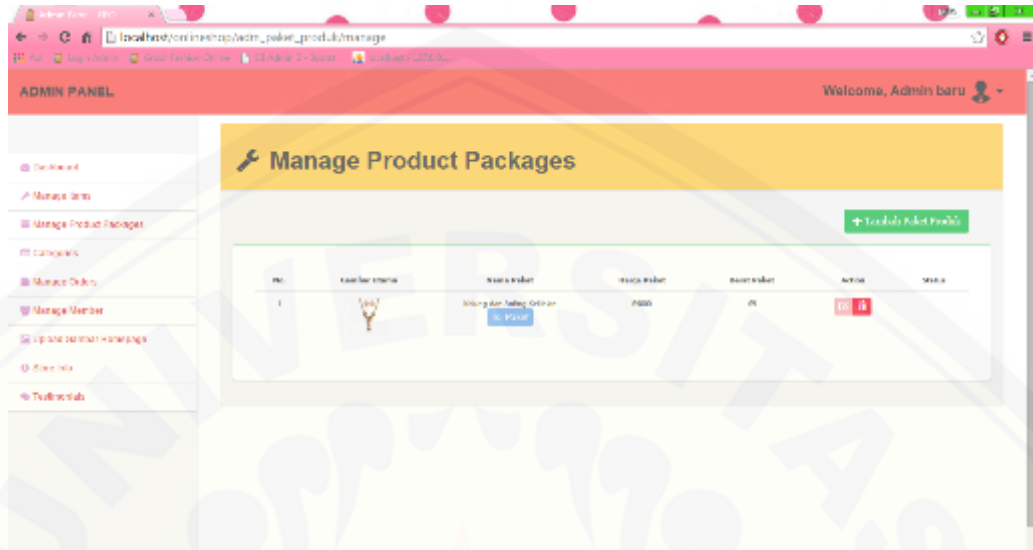
The screenshot shows the 'Add Product Package' form with the following fields:

- Nama Paket
- Harga Paket (Rp)
- Ulasan Produk (jika ada)
- Revisi Produk
- Berat Produk (gr)

Buttons: [Cancel](#), [Simpan](#)

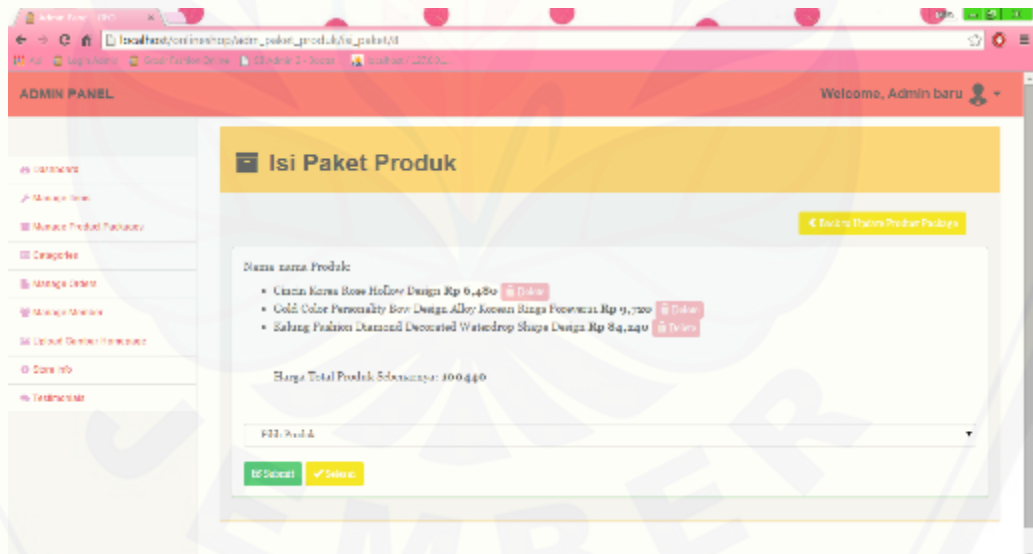
Gambar J.23. Menambah Paket Produk

J.22. Update Paket Produk



Gambar J.24 Update Paket Produk

J.23. Mengelola Isi Paket Produk



Gambar J.24 Mengelola Isi Paket Produk