



**SISTEM INFORMASI PENENTUAN *CROSS-SELLING* PRODUK PADA
SISTEM PENJUALAN KPRI UNIVERSITAS JEMBER MENGGUNAKAN
METODE APRIORI**

SKRIPSI

Oleh

Rizki Fernanda Agustin

NIM 132410101001

**PROGRAM STUDI SISTEM INFORMASI
UNIVERSITAS JEMBER**

2017



**SISTEM INFORMASI PENENTUAN *CROSS-SELLING* PRODUK PADA
SISTEM PENJUALAN KPRI UNIVERSITAS JEMBER MENGGUNAKAN
METODE APRIORI**

SKRIPSI

Diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat
untuk menyelesaikan pendidikan di Program Studi Sistem Informasi Universitas
Jember dan mendapat gelar Sarjana Sistem Informasi

Oleh

Rizki Fernanda Agustin

NIM 132410101001

PROGRAM STUDI SISTEM INFORMASI

UNIVERSITAS JEMBER

2017

PERSEMBAHAN

Skripsi ini saya persembahkan untuk :

1. Allah SWT yang senantiasa memberikan rahmat dan hidayah-Nya untuk mempermudah dan melancarkan dalam mengerjakan skripsi.
2. Ibunda tercinta Sutiah dan Ayahanda Joko Lelono.
3. Saudara laki-laki Indra Novian Ananto, Saudara perempuan Yulita Putri Nindya Ningrum dan Rizky Wulandari, Kakak Ipar Muhammad Hartono, serta keponakan Nazwa Khaira Salsabila.
4. Sahabatku Yuanita Fajrianty Afera, Isna Nadya Nur Islami, Putri Duwi Aprillia, Indri Setiyaning Sari dan Nining Syafi,atul Waqiah untuk dukungan beserta doanya.
5. Sahabat baper Uslifatul Ustaniah, Ainur Rohmah dan Ghaluh Hermawati Safitri.
6. Keluarga besar Mapala Balwana Program Studi Sistem Informasi Universitas Jember.
7. Guru – guru ku baik dari pendidikan formal maupun informal.
8. Almamater Program Studi Sistem Informasi Universitas Jember.

MOTO

“Tidak ada yang tidak mungkin selama kita berusaha”

“Hidup sederhana tetapi berharap akan selalu berguna untuk orang lain”



PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Rizki Fernanda Agustin

NIM : 132410101001

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Sistem Informasi Penentuan *Cross-Selling* Produk Pada Sistem Penjualan KPRI Universitas Jember Menggunakan Metode 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, 15 Juni 2017

Yang menyatakan,

Rizki Fernanda Agustin

NIM 132410101001

SKRIPSI

**SISTEM INFORMASI PENENTUAN *CROSS-SELLING* PRODUK PADA
SISTEM PENJUALAN KPRI UNIVERSITAS JEMBER MENGGUNAKAN
METODE APRIORI**

Oleh :

Rizki Fernanda Agustin

NIM 1322410101001

Pembimbing

Dosen Pembimbing Utama : Prof. Dr. Saiful Bukhori, S.T., M.Kom

Dosen Pembimbing Pendamping : Muhammad Arif Hidayat, S.Kom., M.Kom

PENGESAHAN PEMBIMBING

Skripsi berjudul “Sistem Informasi Penentuan *Cross-selling* Produk Pada Sistem Penjualan KPRI Universitas Jember Menggunakan Metode Apriori”, telah diuji dan disahkan pada:

hari, tanggal : 15 Juni 2017

tempat : Program Studi Sistem Informasi Universitas Jember

Disetujui oleh:

Pembimbing I,

Pembimbing II,

Prof. Dr. Saiful Bukhori, S.T., M.Kom
NIP. 196811131994121001

Muhamad Arief Hidayat, S.Kom., M.Kom
NIP 198101232010121003

PENGESAHAN PENGUJI

Skripsi berjudul “Sistem Informasi Penentuan *Cross-selling* Produk Pada Sistem Penjualan KPRI Universitas Jember Menggunakan Metode Apriori”, telah diuji dan disahkan pada:

hari, tanggal : 15 Juni 2017

tempat : Program Studi Sistem Informasi Universitas Jember

Tim Penguji :

Penguji I,

Penguji II,

Nelly Oktavia A, S.Si., MT.

Oktalia Juwita, S.Kom., M.MT

NIP.198410242009122008

NIP. 198110202014042001

Mengesahkan

Ketua Program Studi

Prof. Drs. Slamin, M.Comp.Sc.,Ph.D

NIP. 19670420 1992011001

RINGKASAN

Sistem Informasi Penentuan *Cross-selling* Produk Pada Sistem Penjualan KPRI Universitas Jember Menggunakan Metode Apriori; Rizki Fernanda Agustin, 132410101001; 2017, 203 HALAMAN; Program Studi Sistem Informasi Universitas Jember.

KPRI Universitas Jember merupakan salah satu usaha koperasi yang ada di Jember. Salah satu unit usaha yang dijalankan oleh KPRI Universitas Jember adalah Unit pertokoan. KPRI Universitas Jember sudah memiliki sebuah sistem informasi penjualan yang mengelola semua transaksi penjualan produknya. Menurut Priyana & Kardianawati (2015), data penjualan yang tersimpan pada database dapat diolah lebih lanjut untuk menerapkan suatu strategi pemasaran. Salah satu strategi pemasaran yang dapat diterapkan oleh KPRI dengan memanfaatkan data dalam database penjualan adalah strategi *cross-selling* produk.

Untuk membantu mengembangkan strategi pemasaran produk pada KPRI Universitas Jember, maka dirancang sebuah sistem untuk menentukan *cross selling* produk yang merupakan salah satu strategi pemasaran produk. Database yang tersimpan pada sistem informasi penjualan dapat dimanfaatkan untuk membuat sistem yang dapat membantu koperasi dalam menentukan produk *Cross-selling* yang sesuai dan lebih menarik minat konsumen. Sistem Informasi yang akan dibangun menggunakan metode *datamining market basket analysis* dengan menggunakan algoritma Apriori. Dengan menerapkan metode *market basket analysis* dengan menggunakan algoritma Apriori pada pembuatan sistem *Cross-selling* produk maka akan didapatkan hasil rekomendasi produk yang dapat ditawarkan kepada konsumen secara bersamaan dalam satu waktu yang dilihat dari perhitungan pola transaksi. Sehingga KPRI Universitas Jember dapat mengetahui kombinasi produk yang sering dibeli oleh konsumen dan dapat meningkatkan nilai penjualan.

PRAKATA

Puji syukur kehadirat Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul “Sistem Informasi Penentuan *Cross-Selling* Produk Pada Sistem Penjualan KPRI Universitas Jember Menggunakan Metode 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 dukungan berbagai pihak. Oleh karena itu penulis menyampaikan terima kasih kepada :

1. Prof. Drs. Slamin, M.Comp.Sc., Ph.D., selaku Ketua Program Studi Sistem Informasi Universitas Jember;
2. Prof. Dr. Saiful Bukhori, ST., M.Kom., selaku Dosen Pembimbing Utama dan M. Arief Hidayat, S.Kom., M.Kom., selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi;
3. Prof. Drs. Slamin, M.Comp.Sc., Ph.D., sebagai dosen pembimbing akademik, yang telah mendampingi penulis sebagai mahasiswa.
4. Seluruh Bapak dan Ibu dosen beserta staf karyawan di Program Studi Sistem Informasi Universitas Jember;
5. Ibunda tercinta Sutiah dan Ayahanda Joko Lelono yang selalu mendukung dan mendoakan.
6. Saudara laki-laki Indra Novian Ananto, saudara perempuan Yulita Putri Nindya Ningrum dan Rizky Wulandari, Kakak Ipar Muhammad Hartono.
7. Teman-teman seperjuanganku Intention angkatan 2013.
8. Yuanita Fajrianty Afera, Isna Nadya Nur Islami, Putri Duwi Aprillia, Indri Setiyaning Sari dan Nining Syafi'atul Waqiah yang telah mendampingi penulis dari awal semester 1 hingga hari ini serta telah menjadi sahabat yang setia mendengar keluh kesah tentang skripsi.

9. Sahabat baper Uslifatul Ustaniah, Ainur Rohmah Nur Faroka dan Ghaluh Hermawati Safitri
10. Segenap keluarga besar Mapala Balwana Program Studi Sistem Informasi Universitas Jember.
11. Segenap keluarga besar KKN 60 Wahyu Hidayat, S.Kg., Dwi Aguatyaningrum, S.E., Mamik Sumardyahningsih, S.H., Moh. Hasan Sabilli Maulana, Desy Anggreini, Ririn Yulianingtias, Nadia Oktalindyah Johan, Nila Mei Fajrin Triana, Agung Budi Cahyono.
12. Bagus Windy Prakoso dan Febrianto Rama Anji atas bantuan dan dukungannya.
13. KPRI Universitas Jember yang selalu memudahkan penulis dalam pengumpulan data.
14. Teman-Teman Program Studi Sistem Informasi di semua angkatan.
15. Semua pihak yang tidak dapat disebutkan satu persatu.

Dengan harapan bahwa penelitian ini nantinya akan terus berlanjut dan berkembang kelak, penulis juga menerima segala kritik dan saran dari semua pihak demi kesempurnaan skripsi ini. Akhirnya penulis berharap, semoga skripsi ini dapat bermanfaat.

Jember, 15 Juni 2017

Penulis

DAFTAR ISI

SKRIPSI.....	i
PERSEMBAHAN.....	ii
MOTO.....	iii
PERNYATAAN.....	iv
PENGESAHAN PEMBIMBING.....	vi
PENGESAHAN PENGUJI.....	vii
RINGKASAN.....	viii
PRAKATA.....	ix
DAFTAR ISI.....	xi
DAFTAR TABEL.....	xv
DAFTAR GAMBAR.....	xvii
BAB 1. PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan.....	3
1.4 Batasan Masalah.....	4
BAB 2. TINJAUAN PUSTAKA.....	5
2.1. Penetian Terdahulu.....	5
2.2. Data Mining.....	5
2.3. Market Basket Analisis.....	6

2.4. Association Rule	6
2.5. Algoritma Apriori.....	7
BAB 3. METODOLOGI PENELITIAN.....	10
3.1. Jenis Penelitian.....	10
3.2. Perancangan Sistem.....	10
3.2.1. Analisa Kebutuhan.....	12
3.2.2. Desain Sistem.....	12
3.2.3. <i>Implementation & Unit Testing</i>	13
3.2.4. <i>Integration & Testing</i>	13
3.2.5. <i>Operation & Maintenance</i>	14
3.3. Gambaran Umum Sistem	14
BAB 4. PERANCANGAN SISTEM	17
4.1. Deskripsi Umum Sistem.....	17
4.1.1 SOP (<i>statement of purpose</i>)	17
4.2. Analisis Kebutuhan Sistem	17
4.2.1. Kebutuhan Fungsional	17
4.2.2. Kebutuhan Non Fungsional.....	19
4.2.3. Antarmuka Perangkat Keras	19
4.2.4. Antarmuka Perangkat Lunak.....	20
4.3. Desain Sistem.....	20
4.3.1. Business Process	20
4.3.2. Use Case Diagram.....	21
4.3.3. Skenario Sistem.....	26

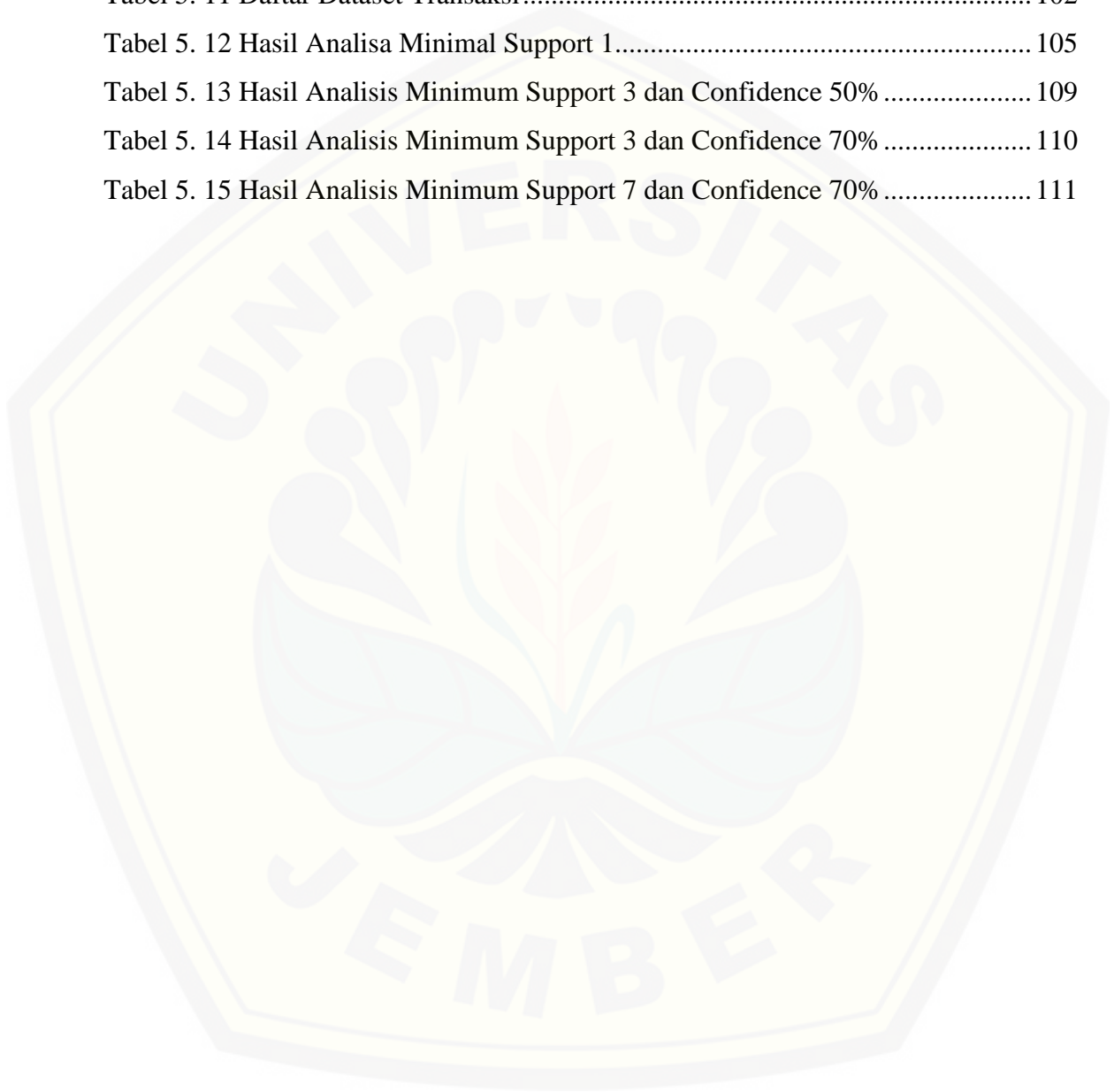
4.3.4.	Sequence Diagram	33
4.3.5.	Activity Diagram.....	38
4.3.6.	Class Diagram	43
4.3.7.	Entity Relationship Diagram.....	45
4.4.	Implementasi Perancangan.....	46
4.5.	Pengujian Sistem	56
4.5.1	Pengujian <i>White Box</i>	56
4.5.2	Pengujian <i>Black Box</i>	69
BAB 5. HASIL DAN PEMBAHASAN.....		70
5.1.	Hasil Implementasi <i>Coding</i> Pada Sistem Informasi <i>Cross-selling</i> Produk KPRI Universitas Jember	70
5.1.1.	Fitur Login	70
5.1.2.	Halaman <i>Dashboard</i>	71
5.1.3.	Fitur Laporan Penjualan.....	72
5.1.4.	Fitur Laporan Barang Masuk	73
5.1.5.	Fitur Laporan Transfer Barang (<i>Manager</i>)	73
5.1.6.	Fitur <i>Cross-selling</i> Produk	74
5.1.7.	Fitur <i>User</i>	76
5.1.8.	Fitur <i>Member</i>	78
5.1.9.	Fitur Supplier	80
5.1.10.	Fitur Kategori	81
5.1.11.	Fitur Daftar Barang	81
5.1.12.	Fitur Pembelian	83

5.1.13. Fitur Penjualan Barang.....	83
5.1.14. Fitur Transfer Barang.....	84
5.2. Pengujian Perhitungan Manual Metode Apriori dengan Perhitungan Aplikasik	85
5.3. Hasil Penerapan Algoritma Apriori dalam Penentuan Cross-selling Produk	95
5.4. Pembahasan Penerapan Algoritma Apriori dalam Penentuan Cross-Selling Produk	102
BAB 6. PENUTUP	113
6.1. Kesimpulan.....	113
6.2. Saran.....	114
DAFTAR PUSTAKA	115
LAMPIRAN	116
LAMPIRAN 1 (Skenario Sistem).....	116
LAMPIRAN 2 (<i>Squence</i> Diagram).....	144
LAMPIRAN 3 (<i>Activity</i> Diagram).....	158

DAFTAR TABEL

Tabel 3. 1 Tabel Uji Black Box	14
Tabel 4. 1 Kebutuhan Fungsional	18
Tabel 4. 2 Kebutuhan Non Fungsional	19
Tabel 4. 3 Definisi Aktor	23
Tabel 4. 4 Definisi Use Case.....	23
Tabel 4. 5 Skenario Mengelola Data Cross-selling Produk	26
Tabel 4. 6 Skenario Melihat Data Cross-selling Produk.....	30
Tabel 4. 7 Kode Program pada Class cross_selling	46
Tabel 4. 8 Kode Program M_cross-selling().....	50
Tabel 4. 9 Kode Program Class Penjualan function add_to_list.....	55
Tabel 4. 10 Test Case Function __construct()	65
Tabel 4. 11 Test Case Function index()	66
Tabel 4. 12 Test Case Function form()	66
Tabel 4. 13 Test Case Function process().....	66
Tabel 4. 14 Test Case Function delete().....	67
Tabel 4. 15 Test Case Function detailhasil().....	68
Tabel 4. 16 Test Case Function add_to_list()	68
Tabel 5. 1 Nama Produk.....	86
Tabel 5. 2 Transaksi Penjualan	86
Tabel 5. 3 Transaksi Penjualan	87
Tabel 5. 4 Daftar Dataset Transaksi	88
Tabel 5. 5 Daftar Hasil 1 itemset	89
Tabel 5. 6 Daftar Penentuan 2 itemset	89
Tabel 5. 7 Daftar Hasil 2 itemset	90
Tabel 5. 8 Daftar Penentuan 3 itemset	90

Tabel 5. 9 Hasil Perhitungan Nilai Confidence	91
Tabel 5. 10 Hasil Penentuan Cross-selling Produk.....	92
Tabel 5. 11 Daftar Dataset Transaksi.....	102
Tabel 5. 12 Hasil Analisa Minimal Support 1.....	105
Tabel 5. 13 Hasil Analisis Minimum Support 3 dan Confidence 50%	109
Tabel 5. 14 Hasil Analisis Minimum Support 3 dan Confidence 70%	110
Tabel 5. 15 Hasil Analisis Minimum Support 7 dan Confidence 70%	111



DAFTAR GAMBAR

Gambar 3.1 Model Waterfall	11
Gambar 3. 2. Flowchart Algoritma Apriori	16
Gambar 4. 1Business Procces (Sumber: Analisis Data, 2017).....	21
Gambar 4. 2 Use Case Diagram (Sumber: Analisis Data, 2017)	23
Gambar 4. 3 Squance diagram untuk mengelola data cross-selling produk (Sumber : Analisis Data, 2017).....	34
Gambar 4. 4 Squance diagram untuk melihat data cross-selling produk (Sumber : Analisis Data, 2017).....	35
Gambar 4. 5 Activity Mengelola Data Cross-selling Produk (Sumber: Analisis Data, 2017)	39
Gambar 4. 6 Activity Melihat Data Cross-selling Produk (Sumber: Analisis Data, 2017)	40
Gambar 4. 7 Class Diagram (Sumber: Analisis Data, 2017)	44
Gambar 4. 8 Entity Relation Diagram (Sumber: Analisis Data, 2017).....	45
Gambar 4. 9 Listing Program Function __construct()	57
Gambar 4. 10 Listing Program Function index()	57
Gambar 4. 11 Listing Program Function form()	57
Gambar 4. 12 Listing Program Function process().....	58
Gambar 4. 13 Listing Program Function delete()	58
Gambar 4. 14 Listing Program Function detailhasil()	58
Gambar 4. 15 Listing Program Function add_to_list()	59
Gambar 4. 16 Diagram Alir Function __construct().....	60
Gambar 4. 17 Diagram Alir Function index().....	60
Gambar 4. 18 Diagram Alir Function form().....	60
Gambar 4. 19 Diagram Alir Function process().....	61
Gambar 4. 20 Diagram Alir Function delete()	61
Gambar 4. 21 Diagram Alir Function detailhasil()	62

Gambar 4. 22 Diagram Alir Function add_to_list().....	63
Gambar 5. 1 Tampilan Fitur Login	71
Gambar 5. 2 Tampilan Halaman Dashboard Manager	71
Gambar 5. 3 Tampilan Halaman Dashboard Gudang	72
Gambar 5. 4 Tampilan Halaman Dashboard Kasir	72
Gambar 5. 5 Tampilan Halaman Laporan Penjualan	72
Gambar 5. 6 Tampilan Halaman Laporan Barang Masuk	73
Gambar 5. 7 Tampilan Halaman Laporan Transfer Barang.....	74
Gambar 5. 8 Tampilan menu cross-selling	74
Gambar 5. 9 Tampilan Halaman Tambah Cross-selling.....	75
Gambar 5. 10 Tampilan Halaman Detail Hasil Cross-selling.....	75
Gambar 5. 11 Tampilan Halaman Dataset Cross-selling	75
Gambar 5. 12 Tampilan Halaman Ubah Cross-selling	76
Gambar 5. 13 Tampilan Halaman Hapus Cross-selling.....	76
Gambar 5. 14 Tampilan Halaman Daftar User	77
Gambar 5. 15 Tampilan Halaman Tambah User	77
Gambar 5. 16 Tampilan Halaman Detail User.....	78
Gambar 5. 17 Tampilan Halaman Ubah User.....	78
Gambar 5. 18 Tampilan Halaman Daftar Member	79
Gambar 5. 19 Tampilan Halaman Tambah Member	79
Gambar 5. 20 Tampilan Halaman Ubah Member	80
Gambar 5. 21 Tampilan Halaman Daftar Supplier	80
Gambar 5. 22 Tampilan Halaman Daftar Barang Supplier.....	81
Gambar 5. 23 Tampilan Halaman Daftar Kategori.....	81
Gambar 5. 24 Tampilan Halaman Daftar Barang	82
Gambar 5. 25 Tampilan Halaman Ubah Daftar Barang.....	82
Gambar 5. 26 Tampilan Halaman Tambah Daftar Barang	83
Gambar 5. 27 Tampilan Halaman Pembelian Barang.....	83

Gambar 5. 28 Tampilan Halaman Tambah Penjualan	84
Gambar 5. 29 Tampilan Halaman Print Penjualan.....	84
Gambar 5. 30 Tampilan Halaman Transfer Barang	85
Gambar 5. 31 Halaman Pengaturan Cross-selling pada Sistem.....	93
Gambar 5. 32 Halaman Detail Hasil Cross-selling pada Sistem.....	94
Gambar 5. 33 Halaman Form Tambah Pengaturan Cross-selling Produk	95
Gambar 5. 34 Halaman Tambah Pengaturan Cross-selling Produk.....	97
Gambar 5. 35 Halaman Detail Hasil Pengaturan Cross-selling Produk.....	97
Gambar 5. 36 Halaman Penerapan Cross-selling Produk Pada Penjualan.....	98
Gambar 5. 37 Kode Program pada Function process()	98
Gambar 5. 38 Kode Program pada Function detailhasil().....	99
Gambar 5. 39 Kode Program pada Function add_to_list()	100
Gambar 5. 40 Kode Program pada Function get_detailhasil().....	101
Gambar 5. 41 Kode Program pada Function get_last().....	102

BAB 1. PENDAHULUAN

Bab ini merupakan langkah awal dari penulisan tugas akhir. Bab ini berisi latar belakang, rumusan masalah, tujuan dan manfaat, batasan masalah, metodologi penelitian, dan sistematika penulisan.

1.1 Latar Belakang

KPRI (Koperasi Pegawai Republik Indonesia) Universitas Jember merupakan salah satu usaha koperasi yang ada di Jember. Koperasi adalah organisasi bisnis yang dimiliki dan dioperasikan oleh orang-orang demi kepentingan bersama. Koperasi berlandaskan Pancasila dan Undang-Undang Dasar Negara Republik Indonesia Tahun 1945. Koperasi berdasar atas asas kekeluargaan, dan Koperasi bertujuan meningkatkan kesejahteraan Anggota pada khususnya dan masyarakat pada umumnya, sekaligus sebagai bagian bagian yang tidak terpisahkan dari tatanan perekonomian nasional demokratis dan berkeadilan (UU No 17 Tahun 2012). KPRI Universitas Jember sebagaimana bentuk koperasi lainnya memiliki tujuan utama mensejahterakan anggotanya dan memuaskan anggotannya (Sanda, 2015)

Salah satu unit usaha yang dijalankan oleh KPRI Universitas Jember adalah Unit pertokoan. Pada Unit pertokoan, produk – produk yang disediakan sangatlah beragam dan memiliki harga yang bersaing. Jenis produk yang dijual juga beragam meliputi kebutuhan pokok (konsumsi dan non-konsumsi), makanan ringan, peralatan elektronik, alat-alat listrik, peralatan rumah tangga dan tekstil. Konsumen dari KPRI Universitas Jember juga beragam. Mulai dari mahasiswa Unej hingga masyarakat sekitar kampus Universitas Jember. Menurut Sanda (2015), beberapa mahasiswa lebih memilih berbelanja di KPRI Universitas Jember dikarenakan barang yang dijual lebih lengkap dan harganya lebih murah dibandingkan dengan minimarket. Di KPRI Universitas Jember juga mengadakan promo bahan pokok, serta bisa melayani

pembelian voucher token dan pembayaran TELISA. Inilah beberapa alasan mengapa mahasiswa memilih berbelanja di KPRI Universitas Jember.

KPRI Universitas Jember memiliki sebuah sistem informasi penjualan yang mengelola semua transaksi penjualan. Data transaksi yang tersimpan dalam *database* yang dihasilkan pada sistem digunakan untuk mengetahui laba dan rugi penjualan. Menurut Priyana & Kardianawati (2015), data penjualan yang tersimpan pada database dapat diolah lebih lanjut untuk menerapkan suatu strategi pemasaran. Salah satu strategi pemasaran yang dapat diterapkan oleh KPRI dengan memanfaatkan data dalam *database* penjualan adalah strategi *cross-selling* produk.

Cross-selling adalah teknik yang digunakan oleh seorang penjual untuk meningkatkan ukuran pembelian barang dan mentransformasikan pembeli produk tunggal menjadi produk multi-produk. *Cross-selling* melibatkan penjualan item tambahan yang terkait dengan barang yang telah dibeli sebelumnya. Sebuah konsep penjualan yang memanfaatkan penjualan produk pertama sebagai *trigger* untuk memasuki penjualan produk selanjutnya (Kamakura, 2007). Dalam hal penjualan, *cross-selling* produk memberikan keuntungan dengan mengurangi kemungkinan konsumen beralih ke pengelola bisnis lain untuk mendapatkan produk terkait (Kumar, 2012). Hal ini tentunya dapat meningkatkan laba KPRI Universitas Jember.

Untuk membantu mengembangkan strategi pemasaran produk pada KPRI Universitas Jember, maka dirancang sebuah sistem untuk menentukan *cross selling* produk yang merupakan salah satu strategi pemasaran produk. Database yang tersimpan pada sistem informasi penjualan dapat dimanfaatkan untuk membuat sistem yang dapat membantu koperasi dalam menentukan produk *Cross-selling* yang sesuai dan lebih menarik minat konsumen. Sehingga diharapkan omset penjualan KPRI Universitas Jember juga semakin meningkat.

Sistem Informasi yang akan dibangun menggunakan metode data mining *market basket analysis* dengan menggunakan algoritma Apriori. *Market basket analysis* dapat menemukan pola yang berupa produk-produk yang sering dibeli

bersamaan atau cenderung muncul bersama dalam sebuah transaksi (Rahayu, 2014). Algoritma apriori dapat digunakan untuk menentukan kandidat-kandidat yang mungkin muncul bersamaan dengan memperhatikan dari syarat nilai minimum *support* dan syarat nilai minimum *confidence*. Algoritma apriori merupakan algoritma yang lebih sederhana dan dapat menangani data yang besar dibandingkan algoritma lainnya.

Dengan menerapkan metode *market basket analysis* menggunakan algoritma Apriori pada pembuatan sistem *Cross-selling* produk maka akan didapatkan hasil rekomendasi produk yang dapat ditawarkan kepada konsumen secara bersamaan dalam satu waktu yang dilihat dari perhitungan pola transaksi. Sehingga KPRI Universitas Jember dapat mengetahui kombinasi produk yang sering dibeli oleh konsumen yang nantinya diharapkan dapat meningkatkan nilai penjualan.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang diatas, permasalahan yang muncul adalah :

1. Bagaimana menerapkan metode Apriori untuk mengetahui kombinasi produk yang sering dibeli oleh konsumen dan membentuk rekomendasi *Cross-selling* produk dari data penjualan yang diperoleh?
2. Bagaimana merancang dan membangun sistem informasi penentuan *Cross-selling* produk pada sistem penjualan KPRI Universitas Jember menggunakan metode apriori?

1.3 Tujuan

Adapun tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut :

1. Dapat menerapkan metode Apriori untuk mengetahui kombinasi produk yang sering dibeli oleh konsumen dan membentuk rekomendasi *Cross-selling* produk dari data penjualan yang diperoleh.

2. Dapat merancang dan membangun sistem informasi penentuan *Cross-selling* produk pada sistem penjualan KPRI Universitas Jember menggunakan metode apriori.

1.4 Batasan Masalah

1. Perancangan dan pembangunan sistem ini digunakan untuk membuat sebuah paket *cross-selling* produk di KPRI Universitas Jember.
2. Metode yang digunakan adalah metode *market basket analysis* dengan algoritma Apriori.
3. Data transaksi yang digunakan hanya yang mengandung itemset dari 10 jenis dari produk makanan dan bahan pokok sebagai analisis penelitian yaitu susu, minuman kemasan, snack, bumbu masakan, beras, gula, roti, mie instan, kopi dan teh.
4. Data yang digunakan untuk item set adalah data penjualan dari KPRI Universitas Jember.

BAB 2. TINJAUAN PUSTAKA

Penelitian tentang sistem informasi penentuan *cross-selling* produk pada sistem penjualan KPRI Universitas Jember menggunakan metode apriori ini membutuhkan landasan teori yang digunakan untuk mengarahkan penelitian ini agar tidak keluar atau melenceng dari kaidah ilmu yang ada. Berikut merupakan teori-teori yang digunakan dan dibahas dalam penelitian :

2.1. Penelitian Terdahulu

Penelitian sebelumnya dengan judul “Implementasi Data Mining pada Penjualan Tiket Pesawat Menggunakan Algoritma Apriori (Studi Kasus : Jumbo Travel Medan)” (Siregar, 2014). Penelitian ini untuk mengetahui sejauh mana algoritma apriori dapat membantu pengembangan strategi pemasaran dalam penjualan tiket pesawat pada jumbo travel serta membantu perusahaan untuk mengetahui penjualan tiket pesawat yang paling banyak terjual.

Penelitian lain dengan judul “Implementasi Data Mining Pada Penjualan Produk Elektronik dengan Algoritma Apriori (Studi Kasus : Kreditplus)” (Pane, 2013) Penelitian ini untuk mengetahui sejauh mana algoritma apriori dapat membantu pengembangan strategi pemasaran serta untuk mengetahui produk elektronik yang paling banyak terjual. Hasil analisa menunjukkan bahwa merk produk yang paling banyak terjual adalah Acer dan Toshiba, dengan diketahuinya produk yang paling banyak terjual tersebut, maka perusahaan dapat menyusun strategi pemasaran untuk memasarkan produk dengan merk lain dengan meneliti kelebihan yang ada pada produk yang paling banyak terjual tersebut dengan produk lainnya.

2.2. Data Mining

Menurut Turban (Gunadi & Sensue, 2012) *data mining* adalah suatu istilah yang digunakan untuk menguraikan penemuan pengetahuan di dalam basis data. Data mining adalah proses yang menggunakan teknik *statistic*, matematika, kecerdasan

buatan, dan machine learning untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai basis data besar.

Data mining merupakan suatu proses pendukung pengambil keputusan dimana kita mencari pola informasi dalam data. Pencarian ini dapat dilakukan oleh pengguna, misalnya dengan menggunakan *query* atau dapat dibantu dengan suatu aplikasi yang secara otomatis mencari pola informasi pada basis data. Pencarian ini disebut *discovery*.

Data mining dapat dibagi menjadi beberapa kelompok berdasarkan tugas yang dapat dilakukan, antara lain : deskripsi, estimasi, klasifikasi, prediksi, pengklusteran, dan asosiasi (Syarifullah, 2010).

2.3. Market Basket Analisis

Market basket analysis adalah suatu metodologi untuk melakukan analisis *buying habit* konsumen dengan menemukan asosiasi antar beberapa item yang berbeda, yang diletakkan konsumen dalam *shopping basket* (keranjang belanja) yang dibeli pada suatu transaksi tertentu. Tujuan dari *market basket analysis* adalah untuk mengetahui produk-produk mana yang mungkin akan dibeli secara bersamaan (Gunadi & Sensue, 2012).

Market basket analysis umumnya dimanfaatkan sebagai titik awal pencarian pengetahuan dari suatu transaksi data ketika kita tidak mengetahui pola spesifik hal yang kita cari. Kebutuhan *market basket analysis* berawal dari keakuratan dan manfaat yang dihasilkannya dalam wujud aturan asosiasi (*association rules*). Yang dimaksud dengan *association rules* adalah pola-pola keterkaitan data dalam basis data.

2.4. Association Rule

Association rule merupakan salah satu teknik di dalam data mining untuk menentukan hubungan antar item dalam suatu dataset (sekumpulan data) yang telah ditentukan. Contoh aturan asosiasi dari analisis pembelian di suatu pasar swalayan

adalah dapat diketahuinya berapa besar kemungkinan seorang konsumen membeli roti bersama dengan susu (Priyana & Kardianawati, 2015).

Association rule (aturan asosiatif) akan menemukan pola tertentu yang mengasosiasikan data yang satu dengan data yang lain. Untuk mencari *association rule* dari suatu kumpulan data, tahap pertama yang harus dilakukan adalah mencari *frequent itemset* terlebih dahulu. *Frequent itemset* adalah sekumpulan item yang sering muncul secara bersamaan. Setelah semua pola *frequent itemset* ditemukan, barulah mencari aturan asosiatif atau aturan keterkaitan yang memenuhi syarat yang telah ditentukan.

2.5. Algoritma Apriori

Algoritma Apriori adalah suatu algoritma dasar yang diusulkan oleh Agrawal & Srikant pada tahun 1994 untuk penentuan *frequent itemsets* untuk aturan asosiasi *boolean*. Algoritma apriori termasuk jenis aturan asosiasi pada data mining. Aturan yang menyatakan asosiasi antara beberapa atribut sering disebut *affinity analysis* atau *market basket analysis*. (Siregar, 2014)

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* (nilai penunjang) adalah persentase kombinasi *item* tersebut dalam database, sedangkan *confidence* (nilai kepastian) adalah kuatnya hubungan antar-*item* dalam aturan asosiasi (Siregar, 2014).

Support (nilai penunjang) merupakan presentase kombinasi sebuah item dalam database. Nilai *support* sebuah item dapat diperoleh menggunakan persamaan 1.

$$\text{Support } A = \frac{\text{Jumlah Transaksi Mengandung } A}{\text{Total Transaksi}} \times 100\% \dots\dots\dots(1)$$

Sedangkan nilai *support* untuk 2 item dapat diperoleh menggunakan persamaan 2.

$$\text{Support } P(A \cap B) = \frac{\text{Jumlah Transaksi Mengandung A dan B}}{\text{Total Transaksi}} \times 100\% \dots\dots\dots(2)$$

Confidence (nilai kepastian) adalah kuatnya hubungan antar item dalam aturan asosiasi.

Nilai *Confidence* dapat dihitung menggunakan persamaan 3.

$$\text{Confidence } P(B|A) = \frac{\text{Total Transaksi Mengandung A dan B}}{\text{Transaksi Mengandung A}} \times 100\% \dots\dots\dots(3)$$

Pada Algoritma Apriori telah dibagi menjadi beberapa tahap yang disebut iterasi. Tiap iterasi dapat menghasilkan pola frekuensi tinggi dengan panjang yang sama. Dimulai dari iterasi pertama yang menghasilkan pola frekuensi tinggi dengan panjang satu. Pada Iterasi kedua menghasilkan 2-*itemset* yang tiap setnya memiliki dua item sampai iterasi selanjutnya. Pertama dibuat kandidat k-*itemset* dari kombinasi semua (k-1)-*itemset*. Lalu untuk tiap kandidat k-*itemset* ini dihitung nilai *support*nya dengan men-*scan* database. *Support* disini artinya jumlah transaksi dalam database yang mengandung semua item dalam kandidat k-*itemset*. Setelah *support* dari semua kandidat k-*itemset* didapatkan, kandidat k-*itemset* yang memenuhi syarat minimum *support* dapat ditetapkan sebagai k-*itemset* yang juga merupakan pola frekuensi tinggi dengan panjang k.

Proses perhitungan *association rule* terdiri dari beberapa tahap :

1. Sistem men-*scan* database untuk mendapatkan kandidat 1-*itemset* (himpunan item yang terdiri dari 1item) dan menghitung nilai *support*nya. Misalnya T adalah himpunan semua jenis ikan dalam database. *Support* dari suatu itemset(S) merupakan presentase dari transaksi dalam T yang mengandung S. Kemudian nilai *support* tersebut dibandingkan dengan minimum *support* yang telah ditentukan, jika nilainya lebih besar atau sama dengan minimum *support* maka *itemset* tersebut termasuk dalam *large itemset*.
2. *Itemset* yang tidak termasuk dalam *large itemset* tidak diikuti dalam iterasi selanjutnya.

3. Pada iterasi kedua sistem akan menggunakan hasil *large itemset* pada iterasi pertama (L1) untuk membentuk kandidat *itemset* kedua (L2). Pada iterasi selanjutnya sistem akan menggunakan hasil *large itemset* pada iterasi sebelumnya (Lk-1) untuk membentuk kandidat *itemset* berikutnya (Lk). Sistem akan menggabungkan (*join*) Lk-1 dengan Lk-1 untuk mendapatkan Lk, seperti pada iterasi sebelumnya sistem akan menghapus (*prune*) kombinasi *itemset* yang tidak termasuk dalam *large itemset*.
4. Setelah dilakukan operasi *join*, maka pasangan *itemset* baru dari hasil proses *join* tersebut juga dihitung *support*-nya.
5. Proses pembentukan kandidat yang terdiri dari proses *join* dan *prune* akan terus dilakukan hingga himpunan kandidat *itemset*-nya null, atau sudah tidak ada lagi kandidat yang akan dibentuk.
6. Setelah itu, dari hasil *frequent itemset* tersebut dibentuk *association rule* yang memenuhi nilai *support* dan *confidence* yang telah ditentukan.
7. Pada pembentukan *association rule*, nilai yang sama seperti "kerapu, kakap" = "kakap, kerapu", dianggap sebagai satu nilai.
8. *Association rule* yang terbentuk harus memenuhi nilai minimum yang telah ditentukan.
9. Untuk setiap *large itemset* L, kita cari himpunan bagian L yang tidak kosong. Untuk setiap himpunan bagian tersebut, dihasilkan *rule* dengan bentuk aP (L - a) jika *support* (L) dan *support* (a) lebih besar dari minimum *support*.

BAB 3. METODOLOGI PENELITIAN

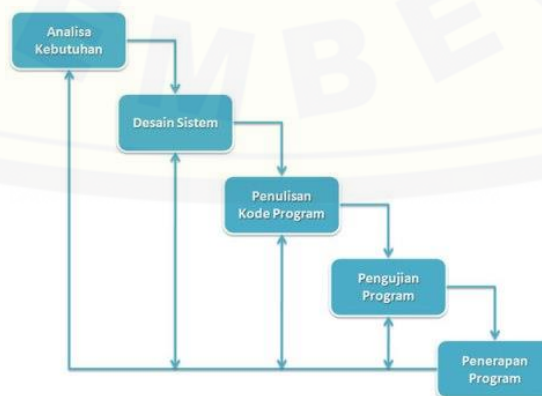
Bab ini menguraikan metodologi penelitian dalam pembuatan dan perancangan sistem informasi *cross-selling* produk pada KPRI Universitas Jember. Metodologi penelitian ini meliputi Jenis Penelitian dan Tahapan Perancangan sistem.

3.1. Jenis Penelitian

Penelitian ini menggunakan dua jenis penelitian yaitu penelitian kualitatif dan penelitian kuantitatif. Jenis penelitian kualitatif digunakan karena penelitian ini menganalisa studi literature dan melakukan *interview* untuk pengumpulan sampel data dan jenis penelitian kuantitatif digunakan karena data yang diolah dalam bentuk angka. Penelitian ini bertujuan untuk membuat suatu paket produk berdasarkan data transaksi penjualan.

3.2. Perancangan Sistem

Metode yang digunakan dalam penelitian ini yaitu *Sytem Development Life Cycle* (SDLC) dengan model *Waterfall*. Model ini dipakai karena umum digunakan dalam pengembangan *software* berskala kecil. Lima tahap dalam model waterfall adalah sebagai berikut, yaitu Analisis Kebutuhan (*Requirement Analysis*), Desain Sistem (*System Design*), Penulisan Kode Program (*Implementation & UnitTesting*), Pengujian Program (*Integration & Testing*), Penerapan Program (*Operations & Maintenance*). Lima tahap pada model waterfall dapat dijelaskan pada Gambar 3.1 dibawah ini.



Gambar 3.1 Model Waterfall

Tahapan tahapan dari metode waterfall adalah sebagai berikut :

1. Requirement Analysis

Seluruh kebutuhan software harus bisa didapatkan dalam fase ini, termasuk didalamnya kegunaan software yang diharapkan pengguna dan batasan software. Informasi ini biasanya dapat diperoleh melalui wawancara, survey atau diskusi. Informasi tersebut dianalisis untuk mendapatkan dokumentasi kebutuhan pengguna untuk digunakan pada tahap selanjutnya.

2. System Design

Tahap ini dilakukan sebelum melakukan coding. Tahap ini bertujuan untuk memberikan gambaran apa yang seharusnya dikerjakan dan bagaimana tampilannya. Tahap ini membantu dalam menspefikasikan kebutuhan hardware dan sistem serta mendefinisikan arsitektur sistem secara keseluruhan.

3. Implementation

Dalam tahap ini dilakukan pemrograman. Pembuatan software dipecah menjadi modul-modul kecil yang nantinya akan digabungkan dalam tahap berikutnya. Selain itu dalam tahap ini juga dilakukan pemeriksaan terhadap modul yang dibuat, apakah sudah memenuhi fungsi yang diinginkan atau belum.

4. Integration & Testing

Di tahap ini dilakukan penggabungan modul-modul yang sudah dibuat dan dilakukan pengujian. Ini dilakukan untuk mengetahui apakah software yang dibuat telah sesuai dengan desainnya dan masih terdapat kesalahan atau tidak.

5. Operation & Maintenance

Ini merupakan tahap terakhir dalam model waterfall. Software yang sudah jadi dijalankan serta dilakukan pemeliharaan . Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya.

3.2.1. Analisa Kebutuhan

Tahap pertama pada proses perancangan perangkat lunak adalah analisis kebutuhan. Pada tahap ini, peneliti mencari permasalahan yang ada untuk dapat dianalisis kebutuhan yang diperlukan, sebagai solusi dari permasalahan yang muncul. Data kebutuhan dibagi menjadi kebutuhan fungsional dan non-fungsional. Data yang sudah terkumpul akan menentukan bagaimana fitur yang akan dibangun pada sistem.

3.2.2. Desain Sistem

Pembuatan desain system pada penelitian ini menggunakan *Unified Modelling Language* (UML) yang dirancang dengan konsep *Object-Oriented Programming* (OOP). Pemodelan UML yang digunakan adalah sebagai berikut:

a. *Business Process*

Business process adalah diagram yang memperlihatkan inputan yang digunakan sistem, output yang dihasilkan sistem, dan tujuan pembuatan sistem.

b. *Use Case Diagram*

Use case diagram adalah gambaran fitur dari sistem yang dijalankan oleh aktor. Pada diagram ini dapat dilihat juga hak akses dari aktor.

c. *Scenario*

Scenario digunakan untuk menjelaskan fitur yang ada pada *use case diagram*

d. *Sequence Diagram*

Sequence diagram adalah diagram yang menggambarkan interaksi antara objek satu dengan yang lain di dalam sistem yang dibangun pada urutan waktu. Diagram juga menggambarkan interaksi antara aktor, fitur, serta data yang berjalan.

e. *Activity Diagram*

Activity diagram adalah penggambaran alir sistem yang akan dibangun, bagaimana sistem dari awal hingga sistem ditutup, serta bagaimana alir sistem ketika diimplementasikan dengan metode yang digunakan.

f. *Class Diagram*

Class diagram adalah diagram yang menggambarkan kelas-kelas dalam sebuah sistem dan hubungannya antara satu kelas dengan yang lain. Dalam kelas ini juga ditampilkan atribut dan operasi yang ada pada sistem.

g. *Entity Relation Diagram (ERD)*

Entity relation diagram (ERD) adalah diagram yang menggambarkan relasi objek-objek dasar data dalam sebuah basis data.

3.2.3. *Implementation & Unit Testing*

Desain yang sudah dibuat pada tahap sebelumnya digunakan sebagai acuan dalam pembangunan sistem. Sistem dibangun menggunakan bahasa pemrograman *Page Hypertext Pre-Processor (PHP)* dengan *tool* yang digunakan Notepad++, dan database yang digunakan adalah *MySQL*. Proses pengkodean menggunakan *framework Laravel* karena mendukung penggunaan konsep *Object Oriented Programming (OOP)*.

3.2.4. *Integration & Testing*

Tahap testing harus dilakukan sebelum sistem diserahkan kepada *user*. Tahap dilakukan agar programmer dapat mengetahui apakah sistem yang dibangun sesuai dengan kebutuhan yang telah dianalisis diawal. Serta agar mengetahui apakah terdapat kesalahan pada sistem yang dibangun. Tahap testing dilakukan guna menyempurnakan sistem sebelum diserahkan kepada *user*. Pada tahap testing ini dilakukan pengujian dengan metode *white-box* dan metode *black-box*.

White-box testing adalah metode desain *test case* yang menggunakan struktur kontrol desain prosedural untuk memperoleh *test case*. Pada penelitian ini metode yang digunakan dalam pengujian *white box* adalah *independent path* dengan menggunakan *cyclomatic complexity*.

Cyclomatic complexity yaitu metrik perangkat lunak yang menyajikan ukuran kuantitatif dari kekompleksan logikal suatu program. *Cyclomatic complexity* digunakan

untuk mencari jumlah *path* dalam suatu *flowgraph*. Rumus yang digunakan dalam menghitung *cyclomatic complexity* adalah sebagai berikut :

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

Keterangan :

E : Jumlah *edge* grafik alir

N : Jumlah *node* grafik alir

Black box testing adalah menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian *black box* seperti pada tabel 3.1 berikut:

Tabel 3. 1 Tabel Uji Black Box

Kelas Uji	Skenario Uji	Hal yang Diharapkan	Kesimpulan

3.2.5. Operation & Maintenance

Tahap *operations & maintenance* adalah tahap akhir dari model waterfall. Tahap ini dilakukan setelah sistem melalui tahap pengujian baik melalui pengujian *white-box* maupun pengujian *black-box*. Tahap *operations & maintenance* dilakukan guna pengembangan sistem atau mengatasi masalah (*bugs*) yang muncul setelah sistem diserahkan kepada *user*.

3.3. Gambaran Umum Sistem

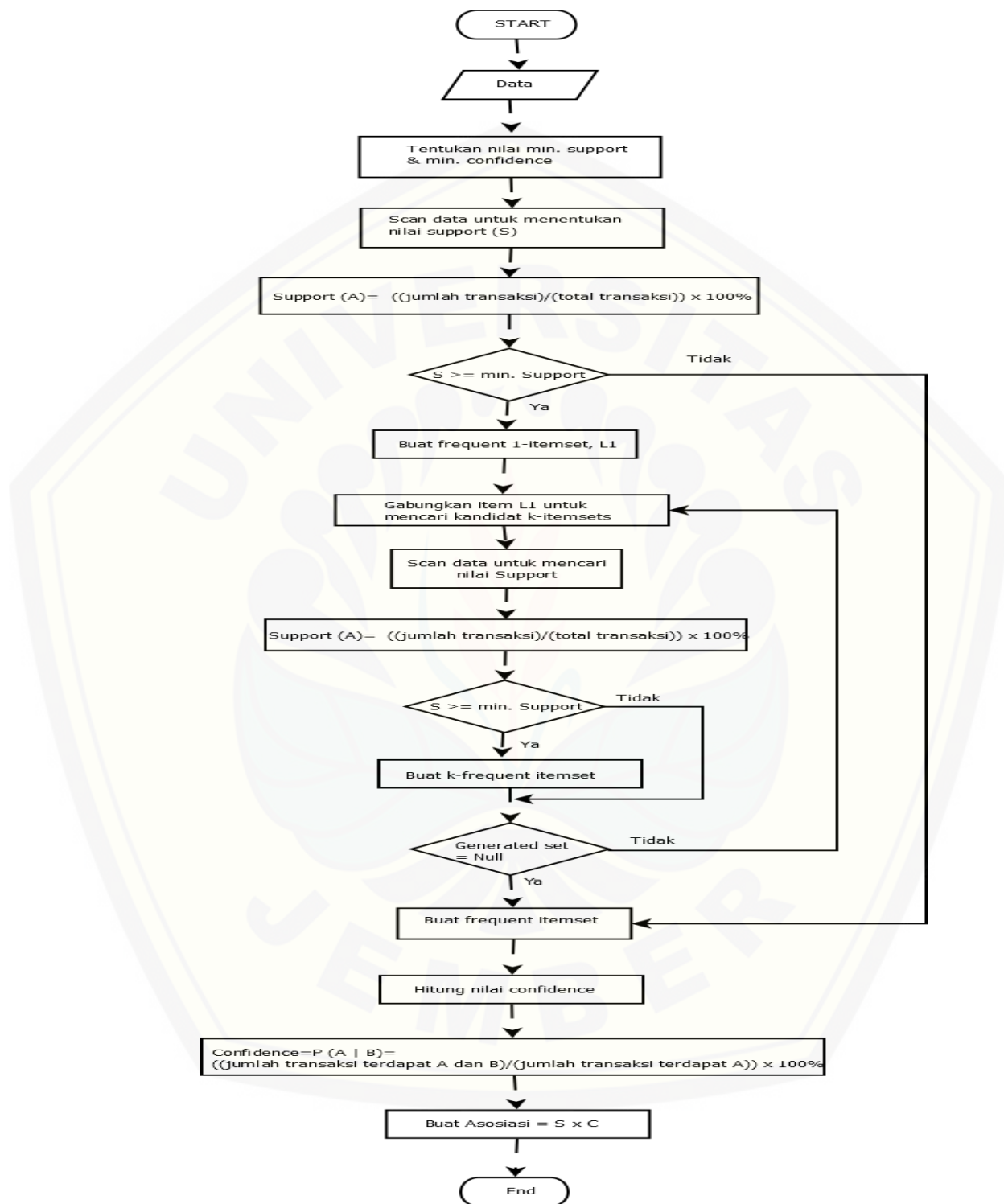
Sistem informasi yang akan dibangun adalah sistem informasi *Cross-selling* produk dengan menerapkan algoritma Apriori. Sistem yang dirancang nantinya dapat mengelola data penjualan dari sistem informasi penjualan KPRI Universitas Jember menjadi data untuk menentukan paket rekomendasi produk untuk melakukan *Cross-selling*.

Dari data transaksi penjualan yang telah didapat nantinya akan dibentuklah *itemset* dari masing-masing item produk, kombinasi *itemset* dapat dibentuk dari dua atau tiga

item set dengan nilai minimal *support* dan *confidence* yang telah ditentukan. Sehingga nantinya akan didapat sebuah data yang menunjukkan kelompok produk yang lebih banyak diminati konsumen yang nantinya akan menjadi produk yang dapat di rekomendasikan kepada konsumen. Proses perhitungan *algoritma apriori* terdiri dari beberapa tahap :

1. Menentukan nilai minimum *support* dan minimum *confidence*
2. Sistem men-scan database untuk mendapatkan kandidat 1-*itemset* (himpunan item yang terdiri dari 1item) dan menghitung nilai *support*nya. Kemudian nilai *support* tersebut dibandingkan dengan minimum *support* yang telah ditentukan, jika nilainya lebih besar atau sama dengan minimum *support* maka *itemset* tersebut akan menjadi kandidat dalam iterasi selanjutnya (*large itemset*). *Itemset* yang tidak termasuk dalam *large itemset* tidak diikuti dalam iterasi selanjutnya.
3. Pada iterasi kedua sistem akan menggunakan hasil *large itemset* pada iterasi pertama (L1) untuk membentuk kandidat *itemset* kedua (L2). Pada iterasi selanjutnya sistem akan menggunakan hasil *large itemset* pada iterasi sebelumnya (Lk-1) untuk membentuk kandidat *itemset* berikutnya (Lk). Sistem akan menggabungkan (join) Lk-1 dengan Lk-1 untuk mendapatkan Lk
4. Setelah didapat k-*itemset* lalu dihitung nilai *support*nya. Proses pembentukan kandidat yang terdiri dari proses *join* dan *prune* akan terus dilakukan hingga himpunan kandidat *itemset*nya null, atau sudah tidak ada lagi kandidat yang akan dibentuk.
5. Setelah itu, dari hasil *frequent itemset* tersebut dihitung nilai minimum *confidencenya* dan dibentuk *association rule* yang memenuhi nilai *support* dan *confidence* yang telah ditentukan.

Untuk menjelaskan alur metode apriori pada sistem dapat dilihat pada Gambar 3.2 dibawah ini.



Gambar 3. 2. Flowchart Algoritma Apriori

BAB 4. PERANCANGAN SISTEM

Bab ini akan menguraikan tentang perancangan desain sistem untuk implementasi metode Apriori pada sistem informasi *cross-selling* produk. Perancangan sistem dimulai dari analisis kebutuhan fungsional dan non-fungsional sistem, kemudian merancang *business process*, *usecase diagram*, *scenario*, *activity diagram*, *sequence diagram*, *class diagram* dan *entity relationship diagram (ERD)*.

4.1. Deskripsi Umum Sistem

Deskripsi umum dari Sistem Informasi Cross-selling Produk pada KPRI Universitas Jember Menggunakan Metode Apriori yang dibangun dalam penelitian ini akan dijelaskan lebih detail pada SOP (*statement of purpose*) sistem dan fungsi sistem.

4.1.1 SOP (*statement of purpose*)

Sistem Informasi *Cross-selling* Produk pada KPRI Universitas Jember Menggunakan Metode Apriori ini merupakan sebuah sistem informasi yang mampu melakukan proses untuk mengolah data dari transaksi penjualan menjadi data baru untuk menentukan *cross-selling* produk pada KPRI Universitas Jember. Data yang dibutuhkan untuk menjalankan sistem ini adalah data penjualan yang didapat dari transaksi penjualan. Data penjualan tersebut nantinya akan digunakan untuk itemset pada proses perhitungan algoritma apriori. Sistem tersebut nantinya akan menghasilkan kombinasi barang apa saja yang sering dibeli secara bersamaan pada transaksi.

4.2. Analisis Kebutuhan Sistem

Pada tahap pertama dilakukan pendefinisian kebutuhan untuk merancang dan membangun Sistem informasi *cross-selling* produk pada KPRI Universitas Jember menggunakan metode Apriori. Literatur-literatur di analisis sehingga kebutuhan system didefinisikan ke dalam kebutuhan fungsional dan kebutuhan non fungsional.

4.2.1. Kebutuhan Fungsional

Kebutuhan fungsional di definisikan sebagai proses yang dilakukan perangkat lunak supaya inputan atau masukan data diterima dan di proses untuk

menghasilkan keluaran sesuai dengan kebutuhan. Kebutuhan fungsional dalam Sistem informasi *cross-selling* produk pada KPRI Universitas Jember digambarkan pada Tabel 4.1.

Tabel 4. 1 Kebutuhan Fungsional

SRSF_ID	Identifikasi
SRSF 01	Sitem dapat mengidentifikasi user dan admin yang berhak atau tidak berhak mengakses sistem saat menggunakan fitur Login.
SRSF 02	Sistem dapat mengelola data barang meliputi menambah, menghapus, merubah.
SRSF 03	Sistem dapat mengelola transaksi pembelian barang yang dilakukan oleh konsumen.
SRSF 04	Sistem dapat menghitung dan menampilkan informasi kombinasi barang apa saja yang sering dibeli secara bersamaan pada transaksi menggunakan Algoritma apriori
SRSF 05	Sistem dapat menampilkan rekomendasi <i>cross-selling</i> produk untuk manager.
SRSF 06	Sistem dapat menampilkan laporan data transaksksi penjualan , transaksi transfer barang, dan barang masuk secara harian, bulanan, dan tahunan.
SRF 07	Sistem dapat mengelola transaksi pembelian barang yang dilakukan pihak Gudang dengan supplier.
SRF 08	Sistem dapat mengelola transaksi transfer barang dari pihak Gudang ke Unit toko.

4.2.2. Kebutuhan Non Fungsional

Kebutuhan non fungsional adalah kebutuhan sistem yang tidak secara langsung berhubungan dengan fungsi dan kebutuhan sistem. Kebutuhan non fungsional digambarkan pada Tabel 4.2

Tabel 4. 2 Kebutuhan Non Fungsional

SRSNF_ID	Parameter	Identifikasi
SRSNF_01	<i>Availability</i>	Sistem harus dapat diakses dari mana saja dan kapan saja selama 24 jam non-stop
SRSNF_02	<i>Respon Time</i>	Sistem harus dapat merespon permintaan pengguna paling lama 15 detik.
SRSNF_03	<i>User Friendly</i>	Tampilan sistem mudah di mengerti oleh pengguna agar pengguna tidak merasa kesulitan saat mengoperasikan sistemnya.
SRSNF_04	<i>Realibility</i>	Sistem dapat berjalan sesuai dengan kebutuhan pengguna.

4.2.3. Antarmuka Perangkat Keras

Kebutuhan yang diperlukan untuk mengakses sistem *cross-selling* produk ini adalah PC atau Laptop dengan spesifikasi, sebagai berikut:

1. Satu unit computer processor
2. RAM (Random Access Memory) minimal 1 GB
3. Monitor
4. Keyboard
5. Mouse

4.2.4. Antarmuka Perangkat Lunak

Perangkat lunak yang digunakan untuk memfasilitasi perkembangan sistem, yaitu:

1. Sistem Operasi Windows XP/7/8/Vista
2. NetBeans IDE versi 7.3.1 sebagai PHP editor atau Notepad++
3. Google Chrome / Mozilla Firefox sebagai Browsing
4. XAMPP untuk manajemen database.
5. Visual Paradigm versi 12.0

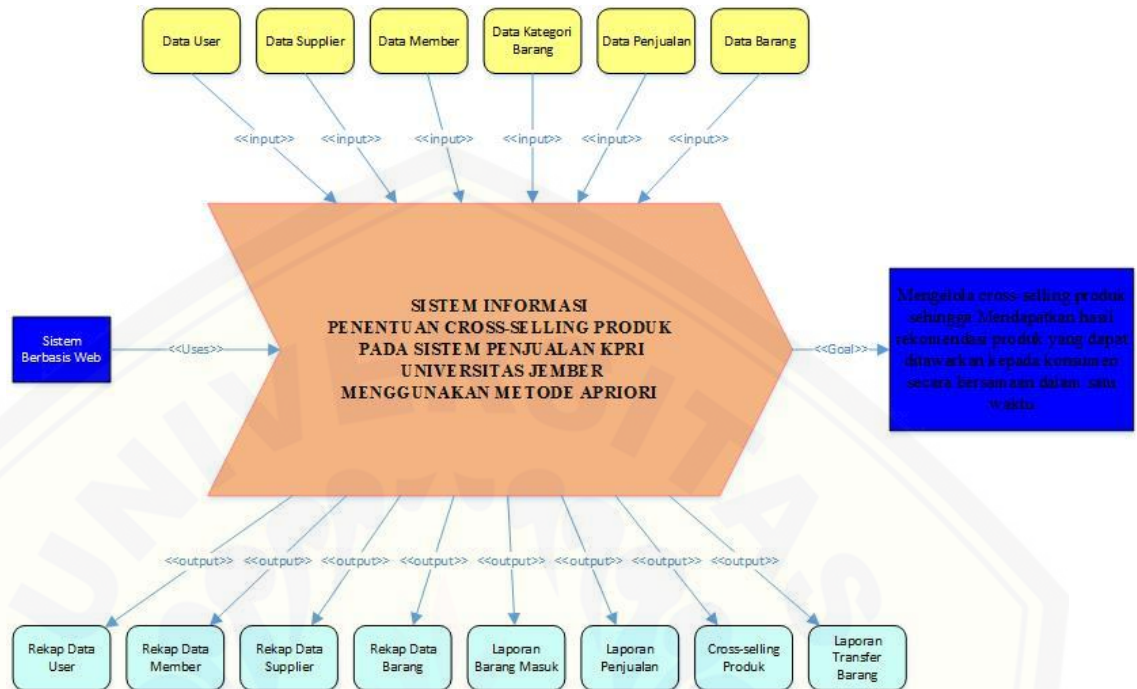
4.3. Desain Sistem

Tahapan ini menggambarkan sistem *cross-selling* produk dengan tahapan yakni *Business Process*, *Usecase Diagram*, *Skenario system*, *Activity Diagram*, *Sequence Diagram*, *Class Diagram* dan *Entity Relationship Diagram (ERD)*.

4.3.1. Business Process

Business process merupakan diagram yang menggambarkan kebutuhan data yang dibutuhkan oleh sistem. *Business process* didalamnya terdiri atas beberapa poin diantaranya:

1. *Input* : Data yang dimasukkan ke dalam aplikasi
2. *Output* : Data yang dihasilkan oleh aplikasi
3. *Goal* : Tujuan dibangun suatu aplikasi
4. *Used* : Platform yang menjadi basis aplikasi
5. *Process* : Sistem yang bekerja

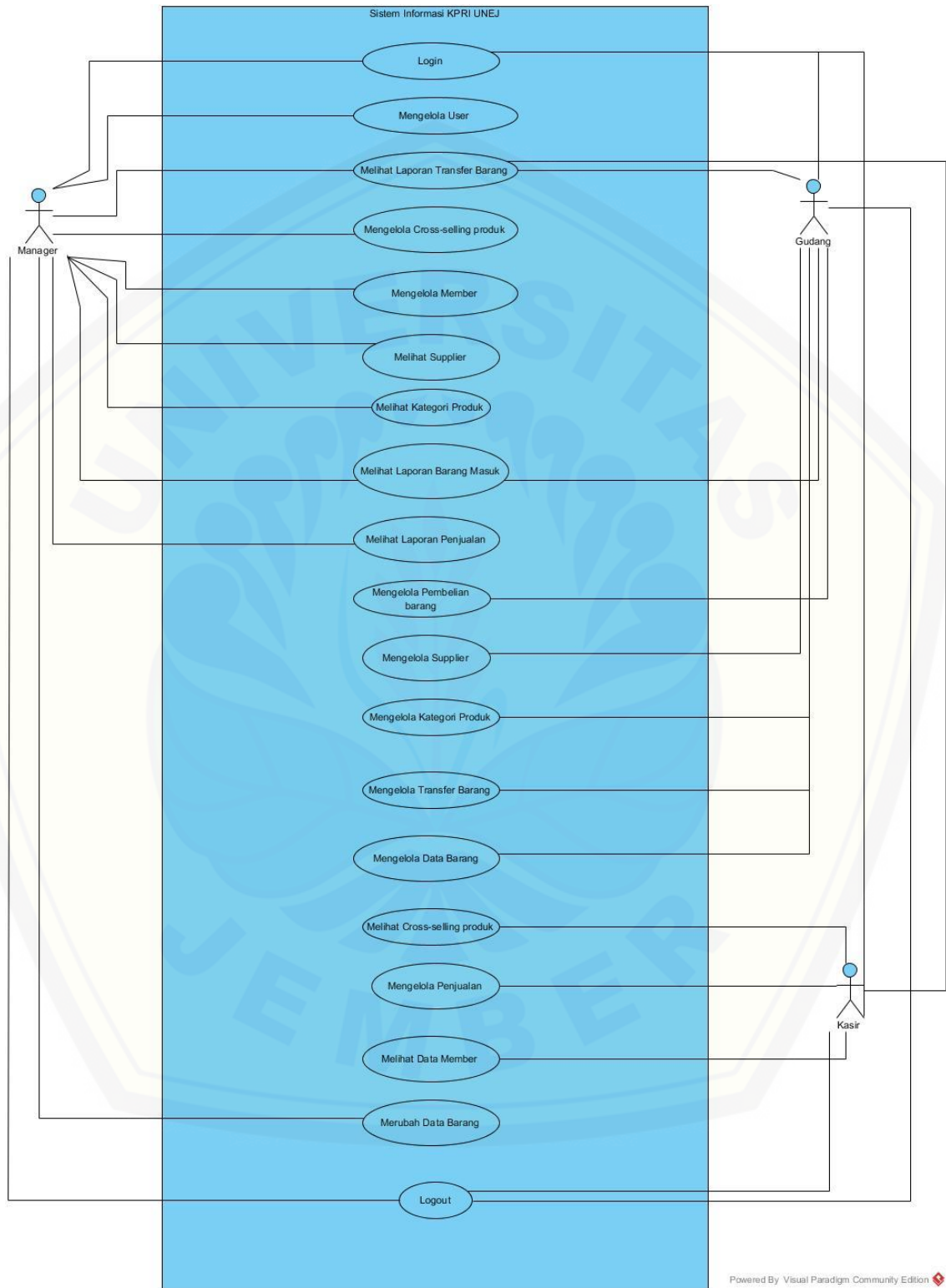


Gambar 4. 1 Bussiness Procces (Sumber: Analisis Data, 2017)

Gambar 4.1 merupakan *business process* dari sistem informasi penentuan cross-selling produk pada sistem penjualan KPRI Universitas Jember menggunakan metode apriori. *Business process* menjelaskan proses *input*, *output*, *goal* dan *uses* yang diaplikasikan ke dalam aplikasi.

4.3.2. Use Case Diagram

Use Case diagram merupakan pemodelan yang dibuat untuk dapat menggambarkan interaksi antara aktor dengan sistem yang akan dibangun. Melalui *use case* diagram dapat diketahui interaksi yang dapat dilakukan aktor terhadap sistem sesuai dengan hak akses yang dimiliki oleh masing-masing aktor atau pengguna. *Use case* diagram ditunjukkan pada Gambar 4.2



Gambar 4. 2 Use Case Diagram (Sumber: Analisis Data, 2017)

Penjelasan berupa tabel definisi aktor yang menggambarkan aktor siapa saja yang terdapat dalam sistem tersebut dapat dilihat pada tabel 4.3 Definisi Aktor. Penjelasan lainnya yaitu disebut definisi *use case* yang menggambarkan fungsionalitas dari setiap *use case* dapat dilihat pada tabel 4.4 Definisi *Use Case*.

Tabel 4. 3 Definisi Aktor

No	Aktor	Deskripsi
1	<i>Manager</i>	Aktor Manager memiliki hak akses yang dapat melakukan proses login, mengelola data user, mengelola data member, melihat data supplier, mengelola <i>cross-selling</i> produk, melihat kategori produk, mengubah daftar barang, melihat data barang masuk, melihat data laporan penjualan, melihat data laporan transfer barang.
2	Gudang	Aktor Gudang memiliki hak akses yang dapat melakukan proses login, mengelola data barang, mengelola data kategori produk, mengelola data transfer barang, mengelola pembelian barang, melihat laporan transfer barang dan melihat laporang barang masuk.
3	Kasir	Aktor Kasir memiliki hak akses yang dapat melakukan proses login, mengelola data penjualan, melihat data transfer barang, melihat data <i>member</i> dan melihat data <i>cross-selling</i> produk.

Tabel 4. 4 Definisi Use Case

No	<i>Usecase</i>	Penjelasan

1	<i>Login</i>	<i>Use case Login</i> merupakan usecase untuk dapat mengakses sistem sesuai hak akses masing-masing aktor.
2	Mengelola Data <i>User</i>	<i>Use case</i> mengelola <i>user</i> merupakan pengelolaan <i>user</i> atau pengguna untuk sistem meliputi tambah, hapus, ubah dan detail.
3	Melihat Laporan Transfer Barang	<i>Use case</i> melihat laporan transfer barang merupakan <i>use case</i> yang digunakan untuk melihat laporan dari transfer barang ke toko
4	Mengelola <i>cross-selling</i> produk	<i>Use case</i> mengelola <i>cross-selling</i> produk merupakan pengelolaan data penjualan menjadi data <i>cross-selling</i> produk meliputi tambah, hapus, ubah, detail, dan detail hasil.
5	Mengelola Data <i>Member</i>	<i>Use case</i> mengelola data <i>member</i> merupakan pengelolaan <i>member</i> di KPRI meliputi tambah, hapus, ubah dan detail.
6	Melihat Data <i>Supplier</i>	<i>Use case</i> melihat data <i>supplier</i> merupakan <i>use case</i> yang digunakan untuk melihat daftar <i>supplier</i> pada KPRI.
7	Melihat Kategori Produk	<i>Use case</i> melihat data kategori produk merupakan <i>use case</i> yang digunakan untuk melihat daftar kategori atau jenis barang yang dijual pada KPRI.
8	Melihat Laporan Barang Masuk	<i>Use case</i> melihat laporan barang masuk merupakan <i>use case</i> yang digunakan untuk melihat laporan

		dari barang yang sudah dibeli dari <i>supplier</i> pada transaksi pembelian.
9	Melihat Laporan Penjualan	<i>Use case</i> melihat laporan penjualan merupakan <i>use case</i> yang digunakan untuk melihat laporan dari barang yang sudah dijual ke konsumen pada transaksi penjualan
10	Mengelola Pembelian Barang	<i>Use case</i> mengelola data pembelian barang merupakan pengelolaan transaksi pembelian barang ke <i>supplier</i> di KPRI meliputi tambah, hapus, ubah dan detail.
11	Mengelola Data <i>Supplier</i>	<i>Use case</i> mengelola data <i>supplier</i> merupakan pengelolaan <i>supplier</i> di KPRI meliputi tambah, hapus, ubah dan detail.
12	Mengelola Kategori Produk	<i>Use case</i> mengelola data kategori produk merupakan pengelolaan kategori atau jenis barang yang dijual di KPRI meliputi tambah, hapus, ubah dan detail.
13	Mengelola Transfer Barang	<i>Use case</i> mengelola data transfer barang merupakan pengelolaan penransferan barang ke Unit Toko di KPRI meliputi tambah, hapus, ubah dan detail.
14	Mengelola Data Barang	<i>Use case</i> mengelola data barang merupakan pengelolaan barang atau produk yang dijual di KPRI meliputi tambah, hapus, ubah dan detail.

15	Melihat <i>Cross-selling</i> Produk	<i>Use case</i> mengelola data <i>cross-selling</i> produk merupakan pengelolaan <i>cross-selling</i> dari data penjualan di KPRI meliputi tambah, hapus, ubah dan detail.
16	Mengelola Penjualan	<i>Use case</i> mengelola data penjualan barang merupakan pengelolaan transaksi penjualan barang ke konsumen di KPRI meliputi tambah, hapus, ubah, detail dan print
17	Melihat Data <i>Member</i>	<i>Use case</i> melihat data member merupakan <i>use case</i> yang digunakan untuk melihat daftar <i>member</i> pada KPRI.
18	Mengubah Data Barang	<i>Use case</i> mengubah data barang merupakan <i>use case</i> yang digunakan untuk mengubah data barang
19	<i>Logout</i>	<i>Use case Logout</i> digunakan untuk keluar dari sistem

4.3.3. Skenario Sistem

Skenario sistem berfungsi untuk menjelaskan alur dari sebuah sistem serta alur alternatif yang dilakukan oleh aktor yang menggunakan sistem informasi *cross-selling* produk. Skenario sistem yang sesuai pada *use case* diagram dapat dilihat dibawah ini.

1. Skenario Mengelola Data *Cross-selling* Produk

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario utama dan skenario alternatif skenario pengelolaan data *cross-selling* produk ditunjukkan pada Tabel 4.5.

Tabel 4. 5 Skenario Mengelola Data *Cross-selling* Produk

No Usecase	UCS04
Nama Usecase	Mengelola data <i>Cross-selling</i>
Aktor	Manager

Deskripsi	Aktor dapat membuat data Cross-selling
Prekondisi	Manager sudah melakukan login
Prakondisik	Manager berhasil mengelola data Cross-selling
Flow of Events	
Skenario Normal : Buat Data <i>Cross-selling</i>	
Aktor	Sistem
1. Klik Menu Cross-selling	
	2. Menampilkan daftar Cross-selling
3. Klik tombol Tambah	
	4. Menampilkan form Cross-selling untuk menambahkan Cross-selling baru
5. Mengisi form Cross-selling yang ditampilkan sistem	
6. Klik tombol "Save"	
	7. Menampilkan notif "Selamat! Penambahan Cross-selling, SUKSES!"
	8. Menampilkan daftar Cross-selling
Skenario Alternatif : Batal Menambahkan Data Cross-selling	
Aktor	Sistem
5. Mengisi form Cross-selling yang ditampilkan sistem	
6. Klik tombol "Close"	
	7. Menampilkan daftar Cross-selling
Skenario Alternatif : Data Tidak Lengkap	
Aktor	Sistem
5. Mengisi form Cross-selling yang ditampilkan sistem	
6. Klik tombol "Save"	
	7. Menampilkan alert "This field is required" pada form Cross-selling
Skenario Normal : Lihat Dataset Cross-selling	

Aktor	Sistem
1. Klik Menu Cross-selling	
	2. Menampilkan daftar Cross-selling
3. Memasukkan data yang akan dicari pada <i>field</i> "Search"	
	4. Menampilkan baris data Cross-selling yang dicari
5. Klik tombol "Dataset"	
	6. Menampilkan dataset pada form Cross-selling
7. Klik tombol "Close"	
	8. Menampilkan daftar Cross-selling
Skenario Normal : Ubah Data Cross-selling	
Aktor	Sistem
1. Klik Menu Cross-selling	
	2. Menampilkan daftar Cross-selling
3. Memasukkan data yang akan dicari pada <i>field</i> "Search"	
	4. Menampilkan baris data Cross-selling yang dicari
5. Klik tombol "Ubah"	
	6. Menampilkan data form Cross-selling
7. Memperbarui data Cross-selling yang ditampilkan di form	
8. Klik "Save"	
	9. Menampilkan notif "Selamat! Perubahan Cross-selling, SUKSES!"
	10. Menampilkan daftar Cross-selling
Skenario Alternatif : Batal Menyimpan Data Cross-selling	
Aktor	Sistem
7. Memperbarui data Cross-selling yang ditampilkan di form	
8. Klik tombol "Close"	
	9. Menampilkan daftar Cross-selling
Skenario Alternatif : Data Tidak Lengkap	

Aktor	Sistem
7. Memperbarui data Cross-selling yang ditampilkan di form	
8. Klik tombol "Save"	
	9. Menampilkan alert "This field is required" pada form Cross-selling
Skenario Normal : Hapus Data Cross-selling	
Aktor	Sistem
1. Klik Menu Cross-selling	
	2. Menampilkan daftar Cross-selling
3. Memasukkan data yang akan dicari pada <i>field</i> "Search"	
	4. Menampilkan baris data Cross-selling yang dicari
5. Klik tombol "Hapus"	
	6. Menampilkan popup "Apakah Anda yakin menghapus data penting ini?"
7. Klik "Oke"	
	8. Menampilkan notif "Selamat! Cross-selling telah dihapus!"
	9. Menampilkan daftar Cross-selling
Skenario Alternatif : Batal Menghapus Data Cross-selling	
Aktor	Sistem
3. Memasukkan data yang akan dicari pada <i>field</i> "Search"	
	4. Menampilkan baris data Cross-selling yang dicari
5. Klik tombol "Hapus"	
	6. Menampilkan popup "Apakah Anda yakin menghapus data penting ini?"
7. Klik "Batal"	
	8. Menampilkan daftar Cross-selling
Skenario Normal : Lihat Data Detail Hasil	
Aktor	Sistem
1. Klik Menu Cross-selling	
	2. Menampilkan daftar Cross-selling

3. Klik tombol Detail Hasil	
	4. Menampilkan detail hasil
5. Klik tombol “Close”	
	6. Menampilkan daftar Cross-selling

2. Skenario Melihat Cross-selling Produk

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario utama dan skenario alternatif skenario melihat cross-selling produk ditunjukkan pada Tabel 4.6.

Tabel 4. 6 Skenario Melihat Data Cross-selling Produk

No Usecase	UCS 15
Nama Usecase	Melihat Data Cross-selling Produk
Aktor	Kasir
Deskripsi	Aktor dapat melihat data cross-selling
Prekondisi	Kasir telah melakukan login
Prakondisi	Kasir telah melihat data cross-selling produk
Flow of Events	
Skenario Normal : Lihat Data Cross-selling	
Aktor	Sistem
1. Klik Menu Penjualan	
	2. Menampilkan Form Penjualan dan Tabel Data Barang
3. Mengisi Form Penjualan yang ditampilkan sistem	
4. Klik tombol “Tambah Ke List”	
	5. Menampilkan alert cross-selling produk
6. Klik OK	
	7. Menampilkan daftar barang yang dibeli

3. Skenario *Login*

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif dari skenario *login* dapat dilihat pada Lampiran A (Skenario Sistem).

4. Skenario Mengelola Data *User*

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif dari skenario mengelola data *user* dapat dilihat pada Lampiran 1 (Skenario Sistem).

5. Skenario Melihat Laporan Transfer Barang

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif dari skenario melihat laporan transfer barang dapat dilihat pada Lampiran 1 (Skenario Sistem).

6. Skenario Mengelola Data *Member*

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif dari skenario mengelola data *member* dapat dilihat pada Lampiran 1 (Skenario Sistem).

7. Skenario Melihat Data *Supplier*

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif dari skenario melihat data *supplier* dapat dilihat pada Lampiran 1 (Skenario Sistem).

8. Skenario Melihat Data Kategori Produk

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif dari skenario melihat data kategori produk dapat dilihat pada Lampiran 1 (Skenario Sistem).

9. Skenario Melihat Laporan Barang Masuk

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif dari skenario melihat laporan barang masuk dapat dilihat pada Lampiran 1 (Skenario Sistem).

10. Skenario Melihat Laporan Penjualan

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif dari skenario melihat laporan penjualan dapat dilihat pada Lampiran 1 (Skenario Sistem).

11. Skenario Mengelola Pembelian Barang

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif dari skenario mengelola pembelian barang dapat dilihat pada Lampiran 1 (Skenario Sistem).

12. Mengelola Data *Supplier*

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif dari skenario mengelola data *supplier* dapat dilihat pada Lampiran 1 (Skenario Sistem).

13. Skenario Mengelola Data Kategori Produk

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif dari skenario mengelola data kategori produk dapat dilihat pada Lampiran 1 (Skenario Sistem).

14. Mengelola Transfer Barang

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif dari skenario mengelola transfer barang dapat dilihat pada Lampiran 1 (Skenario Sistem).

15. Skenario Mengelola Data Barang

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif dari skenario mengelola data barang dapat dilihat pada Lampiran 1 (Skenario Sistem).

16. Skenario Mengelola Penjualan

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif dari skenario mengelola penjualan dapat dilihat pada Lampiran 1 (Skenario Sistem).

17. Skenario Melihat Data *Member*

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif dari skenario melihat data *member* dapat dilihat pada Lampiran 1 (Skenario Sistem).

18. Mengubah Data Barang

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif dari skenario mengubah data barang dapat dilihat pada Lampiran 1 (Skenario Sistem).

19. Skenario *Logout*

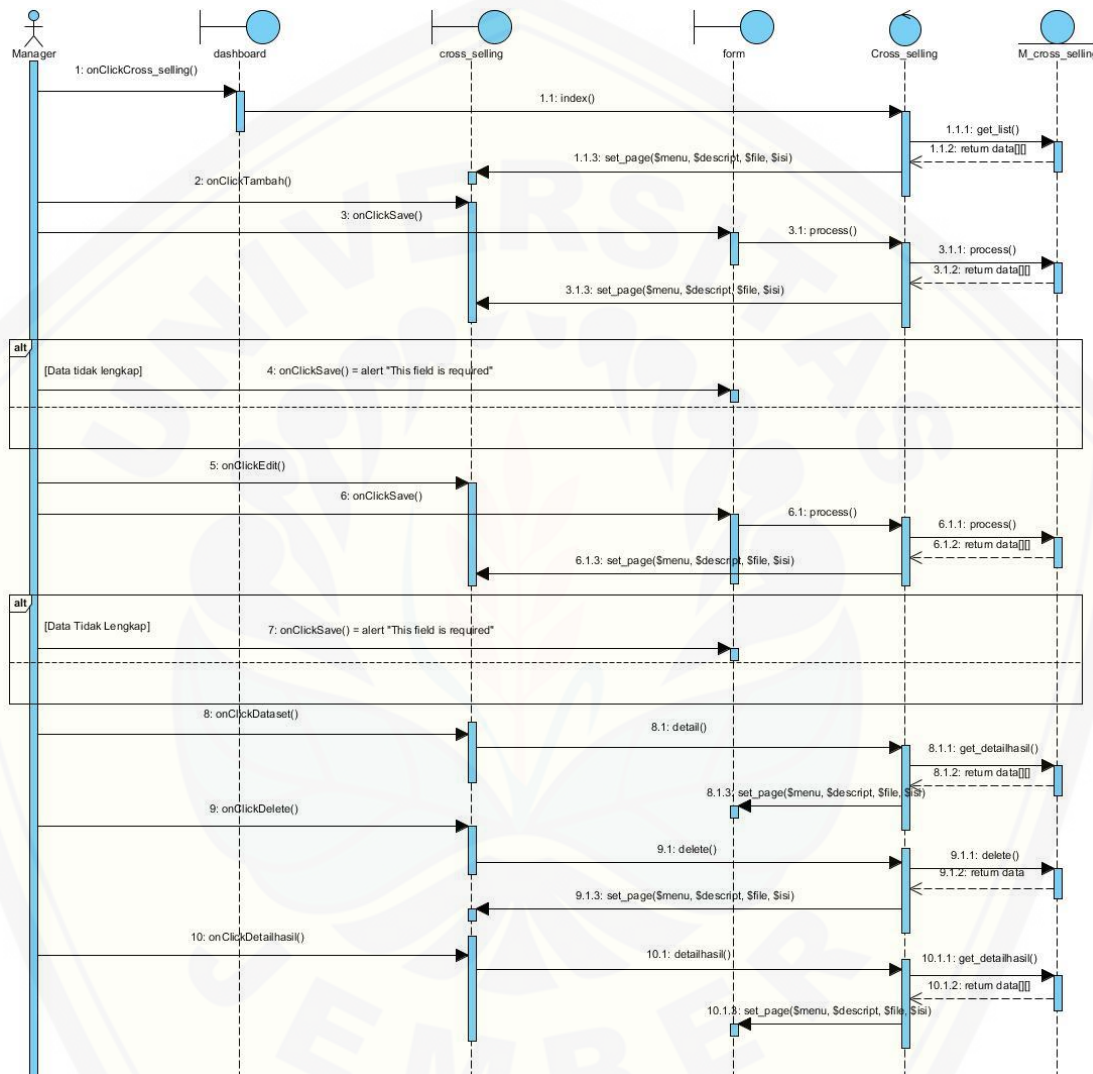
Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif dari skenario *logout* dapat dilihat pada Lampiran 1 (Skenario Sistem).

4.3.4. **Sequence Diagram**

Sequence diagram adalah tahapan dokumentasi suatu diagram terurut yang menampilkan interaksi – interaksi antar objek di dalam sistem, yang juga menggambarkan skenario dan memodelkan aliran logika dalam sistem dengan cara visual. *Sequence* diagram dari sistem informasi *cross-selling* produk adalah sebagai berikut :

1. *Sequence* Diagram Mengelola Data *Cross-selling* Produk

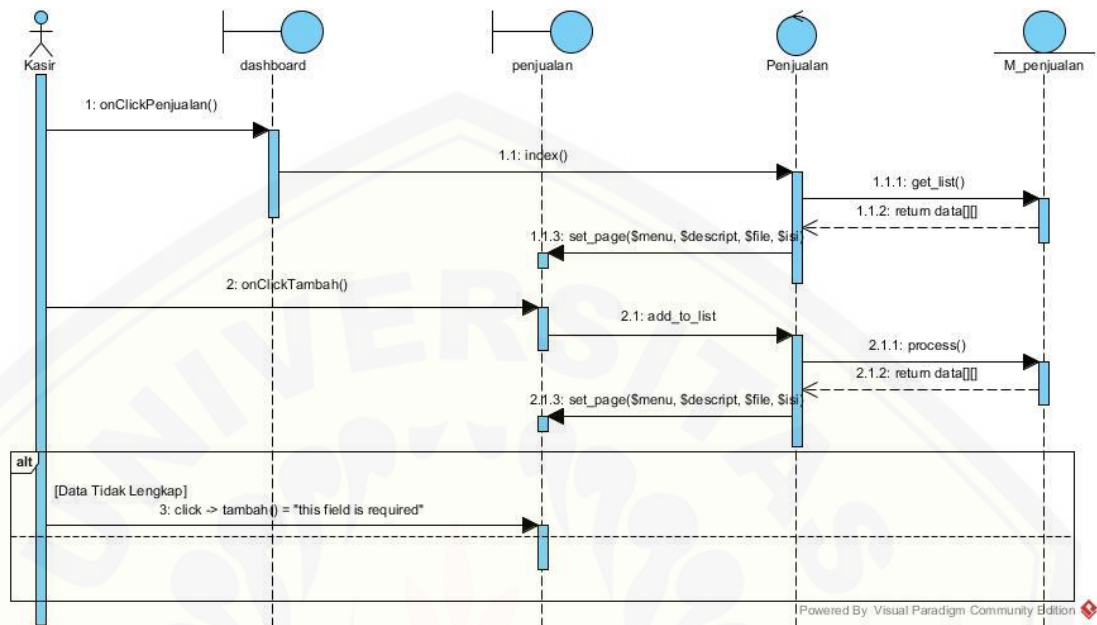
Penggambaran *sequence* diagram mengelola data *cross-selling* produk digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Gambar 4.3.



Gambar 4. 3 Squence diagram untuk mengelola data cross-selling produk (Sumber : Analisis Data, 2017)

2. Sequence Diagram Melihat Data Cross-selling Produk

Penggambaran *sequence* diagram Melihat Data *Cross-selling* Produk digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Gambar 4.4.



Gambar 4. 4 Squence diagram untuk melihat data cross-selling produk (Sumber : Analisis Data, 2017)

3. Sequence Diagram Login

Penggambaran *sequence* diagram *login* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat ditunjukkan pada Lampiran 2 (*Sequence* Diagram).

4. Sequence Diagram Mengelola Data User

Penggambaran *sequence* diagram mengelola data *user* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat ditunjukkan pada Lampiran 2 (*Sequence* Diagram).

5. Sequence Diagram Melihat Laporan Transfer Barang

Penggambaran *sequence* diagram melihat laporan transfer barang digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat ditunjukkan pada Lampiran 2 (*Sequence* Diagram).

6. Sequence Diagram Mengelola Data Member

Penggambaran *sequence* diagram mengelola data *member* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat ditunjukkan pada Lampiran 2 (*Sequence* Diagram).

7. *Sequence* Diagram Melihat Data *Supplier*

Penggambaran *sequence* diagram melihat data *supplier* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat ditunjukkan pada Lampiran 2 (*Sequence* Diagram).

8. *Sequence* Diagram Melihat Data Kategori Produk

Penggambaran *sequence* diagram melihat data kategori produk digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat ditunjukkan pada Lampiran 2 (*Sequence* Diagram).

9. *Sequence* Diagram Melihat Laporan Barang Masuk

Penggambaran *sequence* diagram melihat laporan transfer barang digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat ditunjukkan pada Lampiran 2 (*Sequence* Diagram).

10. *Sequence* Diagram Melihat Laporan Penjualan

Penggambaran *sequence* diagram melihat laporan penjualan digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat ditunjukkan pada Lampiran 2 (*Sequence* Diagram).

11. *Sequence* Diagram Mengelola Pembelian Barang

Penggambaran *sequence* diagram mengelola pembelian barang digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat ditunjukkan pada Lampiran 2 (*Sequence* Diagram).

12. *Sequence* Diagram Mengelola Data *Supplier*

Penggambaran *sequence* diagram mengelola data *supplier* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat ditunjukkan pada Lampiran 2 (*Sequence* Diagram).

13. *Sequence* Diagram Mengelola Data Kategori Produk

Penggambaran *sequence* diagram mengelola data kategori produk digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat ditunjukkan pada Lampiran 2 (*Sequence* Diagram).

14. *Sequence* Diagram Mengelola Transfer Barang

Penggambaran *sequence* diagram mengelola transfer barang digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat ditunjukkan pada Lampiran 2 (*Sequence* Diagram).

15. *Sequence* Diagram Mengelola Data Barang

Penggambaran *sequence* diagram mengelola data barang digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat ditunjukkan pada Lampiran 2 (*Sequence* Diagram).

16. *Sequence* Diagram Mengelola Penjualan

Penggambaran *sequence* diagram mengelola penjualan digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat ditunjukkan pada Lampiran 2 (*Sequence* Diagram).

17. *Sequence* Diagram Melihat Data *Member*

Penggambaran *sequence* diagram melihat data *member* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat ditunjukkan pada Lampiran 2 (*Sequence* Diagram).

18. *Sequence* Diagram Mengubah Data Barang

Penggambaran *sequence* diagram mengubah data barang digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat ditunjukkan pada Lampiran 2 (*Sequence* Diagram).

19. *Sequence* Diagram Logout

Penggambaran *sequence* diagram *logout* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat ditunjukkan pada Lampiran 2 (*Sequence* Diagram).

4.3.5. Activity Diagram

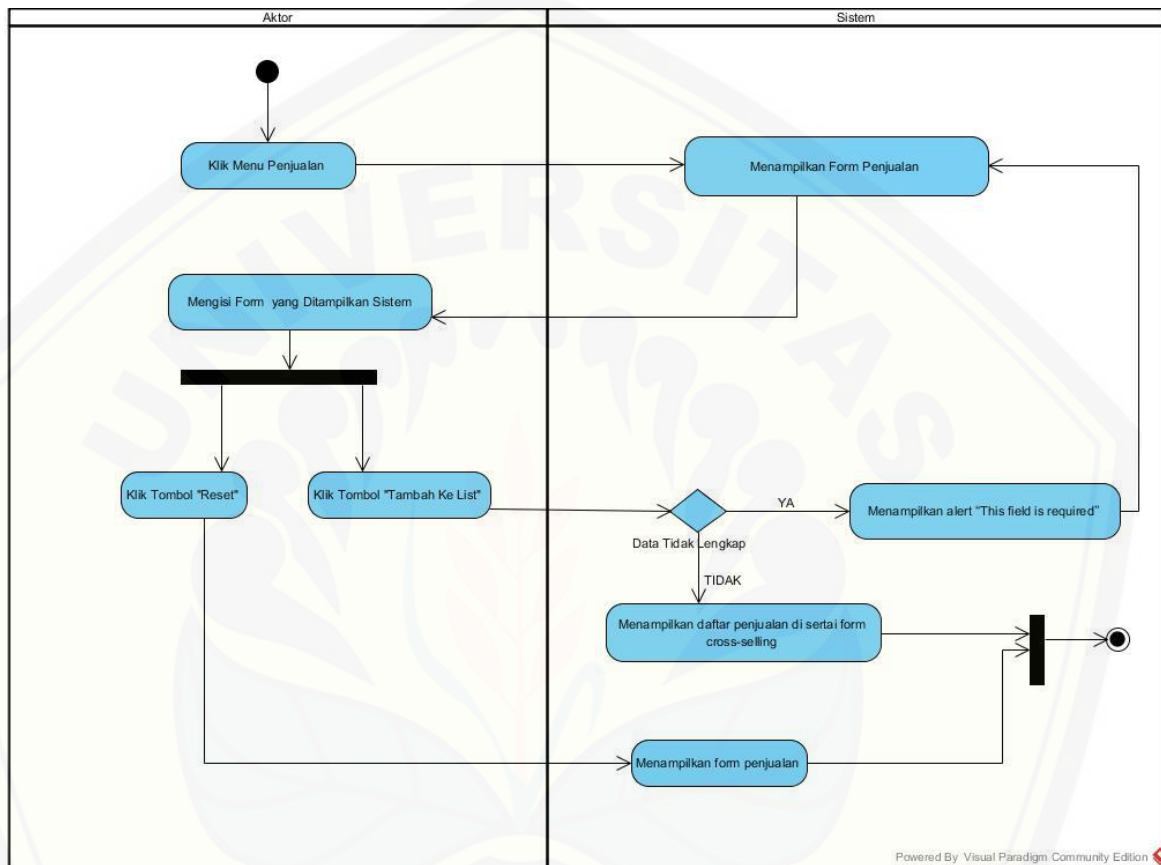
Activity diagram pada sistem *cross-selling* produk ini berfungsi untuk menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir.

1. Activity Diagram Mengelola Data *Cross-selling* Produk

Activity diagram ini menggambarkan alur aktivitas manager jika ingin mengakses dan mengelola data *cross-selling* produk seperti yang ditunjukkan pada Gambar 4.5.

2. Activity Diagram Melihat Data *Cross-selling* Produk

Activity diagram ini menggambarkan alur aktivitas kasir jika ingin melihat data *cross-selling* produk seperti yang ditunjukkan pada Gambar 4.6.



Gambar 4. 6 Activity Melihat Data *Cross-selling* Produk (Sumber: Analisis Data, 2017)

3. Activity Diagram *Login*

Activity diagram untuk *login* dapat digunakan oleh seluruh aktor. Activity diagram ini digunakan untuk melakukan login sebelum memasuki sistem dengan hak akses *manager*, *gudang* dan *kasir*, yang dapat dilihat pada Lampiran 3 (Activity Diagram).

4. Activity Diagram Mengelola Data *User*

Activity diagram untuk mengelola data *user* dapat digunakan oleh *manager*. Activity diagram ini digunakan untuk melakukan pengelolaan terhadap *user* dari sistem

ini, seperti menambah, mengubah, melihat detail, dan menghapus *user*, yang dapat dilihat pada Lampiran 3 (*Activity Diagram*).

5. *Activity Diagram* Melihat Laporan Transfer Barang

Activity diagram untuk melihat laporan transfer barang dapat digunakan oleh seluruh aktor. *Activity diagram* ini digunakan untuk melihat laporan dari transaksi transfer barang dari pihak gudang ke unit toko, yang dapat dilihat pada Lampiran 3 (*Activity Diagram*).

6. *Activity Diagram* Mengelola Data Member

Activity diagram untuk mengelola data *member* dapat digunakan oleh manager. *Activity diagram* ini digunakan untuk melakukan pengelolaan terhadap member yang mendaftar dan terdaftar pada KPRI, seperti menambah, mengubah, melihat detail, dan menghapus member yang dapat dilihat pada Lampiran 3 (*Activity Diagram*).

7. *Activity Diagram* Melihat Data *Supplier*

Activity diagram untuk melihat data *supplier* dapat digunakan oleh manager. *Activity diagram* ini digunakan untuk melihat data *supplier* yang bekerja sama dengan KPRI, yang dapat dilihat pada Lampiran 3 (*Activity Diagram*).

8. *Activity Diagram* Melihat Data Kategori Produk

Activity diagram untuk melihat data kategori produk dapat digunakan oleh manager. *Activity diagram* ini digunakan untuk melihat data kategori produk apa saja yang ada pada KPRI, yang dapat dilihat pada Lampiran 3 (*Activity Diagram*).

9. *Activity Diagram* Melihat Laporan Barang Masuk

Activity diagram untuk melihat laporan barang masuk dapat digunakan oleh manager dan gudang. *Activity diagram* ini digunakan untuk melihat laporan barang yang masuk atau yang dibeli oleh gudang dari *supplier*, yang dapat dilihat pada Lampiran 3 (*Activity Diagram*).

10. *Activity Diagram* Melihat Laporan Penjualan

Activity diagram untuk melihat laporan penjualan dapat digunakan oleh manager untuk melihat laporan dari transaksi penjualan di KPRI untuk harian, bulanan, atau tahunan, yang dapat dilihat pada Lampiran 3 (*Activity Diagram*).

11. *Activity Diagram* Mengelola Pembelian Barang

Activity diagram untuk mengelola pembelian barang dapat digunakan oleh gudang untuk melakukan pengelolaan terhadap transaksi pembelian barang dari pihak supplier dari sistem ini, seperti menambah, mengubah, melihat detail, dan menghapus, yang dapat dilihat pada Lampiran 3 (*Activity Diagram*).

12. *Activity Diagram* Mengelola Data *Supplier*

Activity diagram untuk mengelola data *supplier* dapat digunakan oleh gudang untuk melakukan pengelolaan terhadap *supplier* dari sistem ini, seperti menambah, mengubah, melihat detail, dan menghapus *supplier*, yang dapat dilihat pada Lampiran 3 (*Activity Diagram*).

13. *Activity Diagram* Mengelola Data Kategori Produk

Activity diagram untuk mengelola data kategori produk dapat digunakan oleh gudang. *Activity diagram* ini digunakan untuk melakukan pengelolaan terhadap kategori produk, seperti menambah, mengubah, melihat detail, dan menghapus kategori produk, yang dapat dilihat pada Lampiran 3 (*Activity Diagram*).

14. *Activity Diagram* Mengelola Transfer Barang

Activity diagram untuk mengelola transfer barang dapat digunakan oleh gudang. *Activity diagram* ini digunakan untuk melakukan pengelolaan terhadap transaksi transfer barang dari pihak gudang ke Unit toko, seperti menambah, mengubah, melihat detail, dan menghapus transfer, yang dapat dilihat pada Lampiran 3 (*Activity Diagram*).

15. *Activity Diagram Mengelola Data Barang*

Activity diagram untuk mengelola data barang dapat digunakan oleh gudang. *Activity diagram* ini digunakan untuk melakukan pengelolaan terhadap barang, seperti menambah, mengubah, melihat detail, dan menghapus barang, yang dapat dilihat pada Lampiran 3 (*Activity Diagram*).

16. *Activity Diagram Mengelola Penjualan*

Activity diagram untuk mengelola penjualan dapat digunakan oleh kasir. *Activity diagram* ini digunakan untuk melakukan pengelolaan terhadap transaksi penjualan barang ke konsumen, seperti menambah, mengubah, melihat detail, dan menghapus penjualan, yang dapat dilihat pada Lampiran 3 (*Activity Diagram*).

17. *Activity Diagram Melihat Data Member*

Activity diagram untuk melihat data *member* dapat digunakan oleh kasir. *Activity diagram* dapat dilihat pada Lampiran 3 (*Activity Diagram*).

18. *Activity Diagram Mengubah Data Barang*

Activity diagram untuk mengubah data barang dapat digunakan oleh manager. *Activity diagram* ini digunakan untuk melakukan mengubah dan melihat detail barang, yang dapat dilihat pada Lampiran 3 (*Activity Diagram*).

19. *Activity Diagram Logout*

Activity diagram untuk melihat *logout* dapat digunakan oleh semua aktor. *Activity diagram* ini digunakan untuk keluar dari sistem, yang dapat dilihat pada Lampiran 3 (*Activity Diagram*).

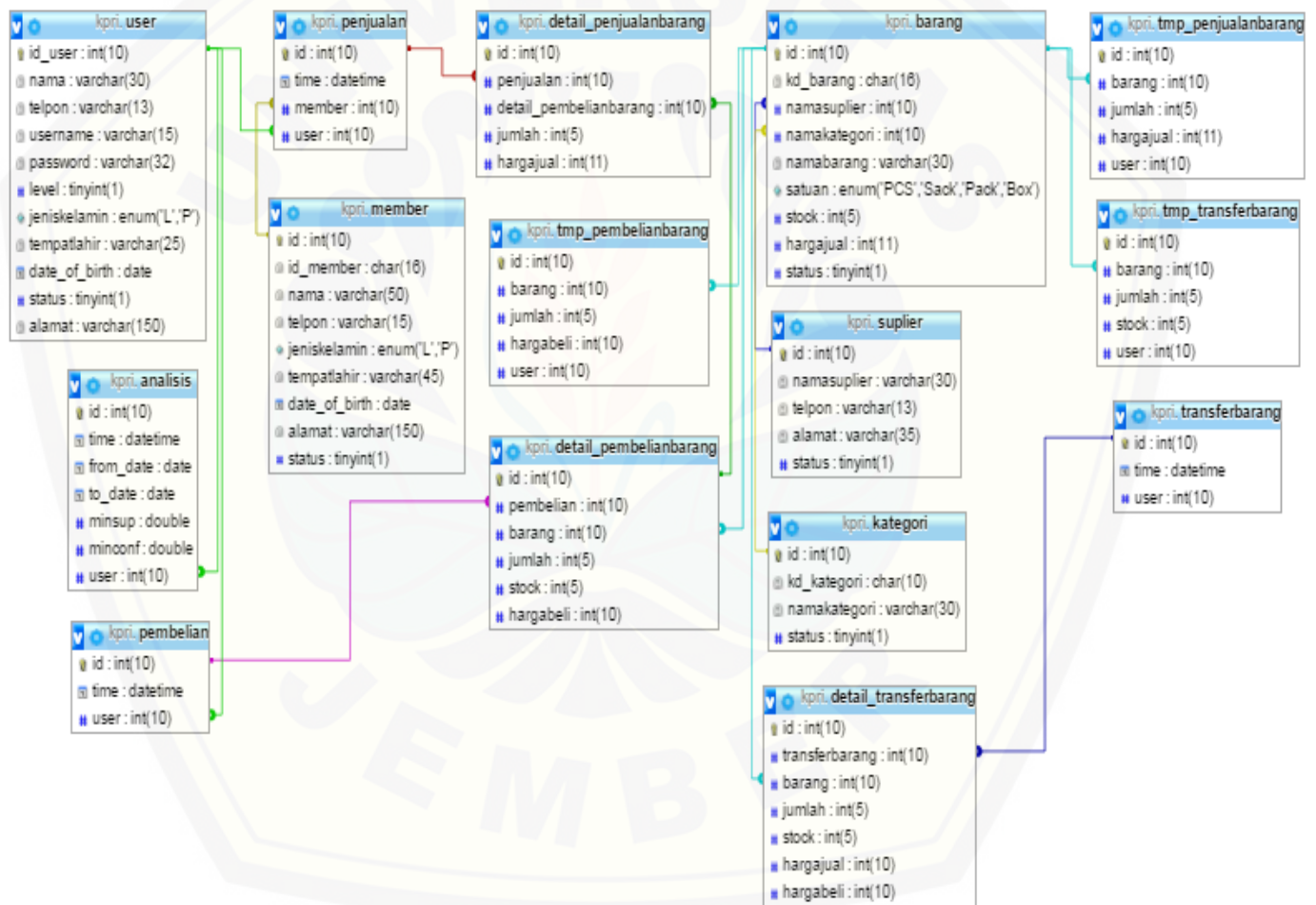
4.3.6. *Class Diagram*

Class Diagram menggambarkan kelas-kelas pada sistem yang dibangun dan hubungan antara kelas satu dan kelas lainnya, serta berisi atribut dan *method* apa saja yang ada di dalamnya. *Class diagram* sistem *cross-selling* produk dapat dilihat pada Gambar 4.7.

Gambar 4.7 menggambarkan tentang relasi antar *class* di dalam sistem. Relasi terjadi antar *controller* dan *model*. Berdasarkan gambar tersebut bisa dipahami keterkaitan dan ketergantungan antar *class* di dalam sistem.

4.3.7. Entity Relationship Diagram

Entity Relationship Diagram (ERD) menggambarkan komponen dan struktur dari database yang digunakan dalam membuat suatu sistem. ERD yang dihasilkan dari database sistem cross-selling produk dapat dilihat pada gambar 4.8.



Gambar 4. 8 Entity Relation Diagram (Sumber: Analisis Data, 2017)

4.4. Implementasi Perancangan

Setelah tahap desain perancangan selesai, tahap selanjutnya dalam penelitian ini yaitu tahap pengimplementasian desain perancangan ke dalam bahasa pemrograman. Bahasa pemrograman yang dipakai adalah bahasa pemrograman PHP dan menggunakan *DBMS MySQL*. Pada tahap implementasi perancangan ini menjelaskan tentang fitur – fitur yang terdapat pada sistem informasi *cross-selling* produk.

Kode program perhitungan metode *Apriori* terdapat pada *class cross_selling* dan *class penjualan* pada *package controllers*, *class M_cross_selling* pada *package models*, dan *class aprioriclass* pada *package libraries*. Halaman yang ditampilkan terdapat pada, *package views* dalam *class cross_selling* dan *class penjualan*. Penulisan kode program perhitungan metode tersebut terdapat pada tabel berikut :

Tabel 4. 7 Kode Program pada Class *cross_selling*

	Kode
1	<?php
2	
3	if (!defined('BASEPATH'))
4	exit('No direct script access allowed');
5	
6	class Cross_selling extends RAST_Control {
7	
8	private \$data;
9	
10	function __construct() {
11	parent::__construct();
12	
13	if (\$this->session->userdata('level') != 1) {
14	redirect('dashboard');
15	}
16	
17	\$this->load->model('M_cross_selling');

```
18     $this->load->library('Aprioriclass');
19     }
20
21     function index() {
22         $sisi = array(
23             'descript' => 'Daftar Produk Cross-selling'
24             , 'dataTable' => array()
25             , 'dataTable' => $this->M_cross_selling->get_list()
26             , 'add_button' => true
27             , 'back_button' => false
28         );
29
30         $this->set_page('cross_selling', $sisi['descript'], 'cross_selling', $sisi);
31     }
32
33     function form() {
34         $this->output->set_content_type('application/json');
35
36         $sid = $_POST['id'];
37         echo json_encode($this->M_cross_selling->get_isi($sid));
38     }
39
40     function process() {
41         $post = $_POST;
42         if ($this->M_cross_selling->process($post)) {
43             $this->session->set_userdata('pesan_sistem', 'Selamat! ' . (($post['id'] == '') ? 'Penambahan'
44             : $this->session->set_userdata('tipe_pesan', 'Sukses');
45             redirect('cross_selling');
46         } else {
47             $this->session->set_userdata('pesan_sistem', 'Maaf! ' . (($post['id'] == '') ? 'Penambahan' :
48             $this->session->set_userdata('tipe_pesan', 'Gagal');
49             redirect('cross_selling');
50         }
51     }
```

```
52
53 function delete($a) {
54     if ($this->M_cross_selling->delete($a)) {
55         $this->session->set_userdata('pesan_sistem', 'Selamat! Cross-selling telah dihapus!');
56         $this->session->set_userdata('tipe_pesan', 'Sukses');
57         redirect('cross_selling');
58     } else {
59         $this->session->set_userdata('pesan_sistem', 'Maaf! Cross-selling tidak terhapus!');
60         $this->session->set_userdata('tipe_pesan', 'Gagal');
61         redirect('cross_selling');
62     }
63 }
64
65 function detailhasil() {
66     $this->output->set_content_type('application/json');
67
68     $id = $_POST['id'];
69     $minsup = $_POST['minsup'];
70     $minconf = $_POST['minconf'];
71     $data = $this->M_cross_selling->get_detailhasil($id);
72
73     $Apriori = new Aprioriclass();
74     $Apriori->setMaxScan(3); //untuk scan maksimal kombinasi 2, 3, ...
75     $Apriori->setMinSup($minsup); //untuk inputan minimum support 1, 2, 3, ...
76     $Apriori->setMinConf($minconf); //untuk minimum confidence - Percent 1, 2, ..., 100
77     $Apriori->setDelimiter(','); //Delimiter
78
79     $Apriori->process($data); //proses
80
81     $d['pfi1'] = $Apriori->printFreqItemsets()[0];
82     $d['pfi2'] = $Apriori->printFreqItemsets()[1];
83     $d['pfi3'] = $Apriori->printFreqItemsets()[2]; //nama barang
84     $d['pfi4'] = $Apriori->printFreqItemsets()[3]; //percent
85
```

```
86     $d['pai1'] = $Apriori->printAssociationRules()[0];
87     $d['pai2'] = $Apriori->printAssociationRules()[1];
88     $d['pai3'] = $Apriori->printAssociationRules()[2];
89     $d['pai4'] = $Apriori->printAssociationRules()[3];
90     echo json_encode($d);
91 }
92
93 function detail() {
94
95     $this->output->set_content_type('application/json');
96
97     $id = $_POST['id'];
98     echo json_encode($this->M_cross_selling->get_detail($id));
99 }
100 function set_page($menu, $descript, $file, $isi) {
101     $data['menu'] = $menu;
102     $data['descript'] = $descript;
103
104     $data['content'] = $this->load->view($file, $isi, TRUE);
105     $this->load->view('template/template', $data);
106 }
107 }
```

Tabel 4.7 merupakan kode program yang terdapat pada *class* *cross_selling* yang berisi beberapa *method function* yang akan digunakan untuk proses penerapan pengaturan paket *cross-selling* produk untuk konsumen. Method *function* *__construct()* pada baris 10-19 digunakan untuk memanggil class lain yang dibutuhkan dalam menjalankan class *cross_selling*. Method *function* *index()* pada baris 21-31 digunakan untuk mengambil data pengaturan *cross-selling* dan menampilkan halaman data pengaturan *cross-selling* produk. Method *function* *form()* pada baris 33-38 digunakan untuk memanggil data pengaturan *cross-selling* produk yang nantinya akan digunakan untuk mengedit dan melihat detail *cross-selling* produk.

Method *function* `process()` pada baris 40-51 digunakan untuk menyimpan data pengaturan *cross-selling* produk. Method *function* `delete()` pada baris 53-63 digunakan untuk menghapus data pengaturan *cross-selling* produk. Method *function* `detailhasil()` pada baris 65-91 digunakan untuk mengolah dan menganalisa *dataset* berdasarkan interval tanggal yang telah dimasukkan serta pengaturan *min. support* dan *min. confidence* berdasarkan id yang telah dipilih pada halaman *cross-selling* produk. Data yang diambil akan di proses pada *Aprioriclass*. *Function* `detailhasil()` digunakan untuk mengambil nilai dan menyimpan dalam bentuk *array* yang digunakan pada *class view* `cross_selling` berdasarkan *query* di *function* `get_detailhasil()` dalam *class* `cross_selling` di package *models*. *Function* `get_detailhasil` memanggil *library* *aprioriclass* untuk menjalankan fungsi perhitungan metode algoritma *apriori*. Method *function* `detail()` pada baris 93-99 digunakan untuk memanggil data transaksi penjualan yang dijadikan *dataset* *cross-selling* produk yang nantinya akan digunakan untuk melihat detail *dataset* *cross-selling* produk.

Tabel 4. 8 Kode Program M_cross-selling()

No	Kode
1	<?php
2	
3	if (!defined('BASEPATH'))
4	exit('No direct script access allowed');
5	
6	class M_cross_selling extends RAST_Model {
7	
8	function get_list() {
9	\$query = \$this->db->query
10	('
11	SELECT
12	*
13	FROM
14	analisis

```
15     );
16
17     return $query->result_array();
18 }
19
20 function get_last() {
21     $query = $this->db->query
22     (
23         SELECT
24         *
25         FROM
26         analisis
27         ORDER BY
28         id DESC
29         LIMIT 1
30     );
31
32     return $query->result_array();
33 }
34
35 function get_isi($a) {
36     $query = $this->db->query
37     (
38         SELECT
39         *
40         FROM
41         analisis
42         WHERE
43         id = ' . $a
44     );
45
46     return $query->result_array();
47 }
48
```

```
49 function process($a) {
50     $query = FALSE;
51     $tgl = explode('/', $a['from_date']);
52     $tgl1 = explode('/', $a['to_date']);
53     if ($a['id'] == "") {
54         $query = $this->db->query
55             (
56             INSERT INTO analisis VALUES
57             (
58             ' . "" . "" . '
59             , NOW()
60             , ' . "" . $tgl[2] . '-' . $tgl[0] . '-' . $tgl[1] . "" . '
61             , ' . "" . $tgl1[2] . '-' . $tgl1[0] . '-' . $tgl1[1] . "" . '
62             , ' . "" . $a['minsup'] . "" . '
63             , ' . "" . $a['minconf'] . "" . '
64             , ' . $this->session->userdata('user_id') . '
65             )
66         );
67     } else {
68         $query = $this->db->query
69             (
70             UPDATE analisis SET
71             from_date = ' . "" . $tgl[2] . '-' . $tgl[0] . '-' . $tgl[1] . "" . '
72             , to_date = ' . "" . $tgl1[2] . '-' . $tgl1[0] . '-' . $tgl1[1] . "" . '
73             , minsup = ' . "" . $a['minsup'] . "" . '
74             , minconf = ' . "" . $a['minconf'] . "" . '
75             WHERE
76             id = ' . $a['id']
77             );
78     }
79 }
80
81 if ($query) {
82     return TRUE;
```



```
83     } else {
84         return FALSE;
85     }
86 }
87
88 function delete($a) {
89     $query = $this->db->query
90     (
91         DELETE FROM analisis
92         WHERE
93             id = ' . $a
94     );
95
96     if ($query) {
97         return TRUE;
98     } else {
99         return FALSE;
100     }
101 }
102
103 function get_detailhasil($a) {
104     $query = $this->db->query
105     (
106         SELECT
107         *
108         FROM
109             analisis
110         WHERE
111             id = ' . $a
112     );
113
114     $b = $query->result_array();
115     $tanggal = 'DATE(a.time) >= "' . $b[0]['from_date'] . "' AND DATE(a.time) <= "' .
116     $b[0]['to_date'] . "'";
```

```
117
118     $query = $this->db->query
119     (
120         SELECT
121             a.id
122             , group_concat(d.namabarang separator ",") AS namabarang
123         FROM
124             penjualan a
125             JOIN detail_penjualanbarang b ON (a.id = b.penjualan)
126             JOIN detail_pembelianbarang c ON (b.detail_pembelianbarang = c.id)
127             JOIN barang d ON (c.barang = d.id)
128         WHERE
129             '. $tanggal .'
130         GROUP BY
131             a.id
132     ');
133     return $query->result_array();
134 }
135 }
```

Tabel 4.8 merupakan kode program `M_cross_selling()` yang digunakan untuk mengelola data *cross-selling* produk menggunakan *query*. Method *function* `get_list()` pada baris 8-18 digunakan untuk mengambil seluruh data pada tabel analisis. Method *function* `get_last()` pada baris 20-33 digunakan untuk mengambil data dari analisis berdasarkan id terakhir. Method *function* `get_isi()` pada baris 35-47 digunakan untuk mengambil data dari tabel analisis berdasarkan id yang dipilih pada package views. Method *function* `process()` pada baris 49-86 digunakan untuk menyimpan dan mengubah data tabel analisis. Method *function* `delete()` pada baris 88-101 digunakan untuk menghapus data dari tabel analisis. Method *function* `get_detailhasil()` pada baris 103-134 digunakan untuk mengambil data tabel analisis sesuai id yang dipilih dan

untuk mengambil data transaksi detail_penjualanbarang berdasarkan tanggal transaksi (periode awal - periode akhir) yang dipilih dan dikelompokkan berdasarkan id.

Tabel 4. 9 Kode Program Class Penjualan function add_to_list

No	Kode
56	function add_to_list() {
57	\$this->output->set_content_type('application/json');
58	
59	\$this->load->model('M_cross_selling');
60	\$this->load->library('Aprioriclass');
61	\$post = \$_POST;
62	
63	if (\$this->M_penjualan->process(\$post)) {
64	\$wedhus = \$this->M_cross_selling->get_last();
65	
66	\$id = \$wedhus[0]['id'];
67	\$minsup = \$wedhus[0]['minsup'];
68	\$minconf = \$wedhus[0]['minconf'];
69	\$data = \$this->M_cross_selling->get_detailhasil(\$id);
70	
71	
72	\$Apriori = new Aprioriclass();
73	\$Apriori->setMaxScan(3); //Scan 2, 3, ...
74	\$Apriori->setMinSup(\$minsup); //Minimum support 1, 2, 3, ...
75	\$Apriori->setMinConf(\$minconf); //Minimum confidence - Percent 1, 2, ..., 100
76	\$Apriori->setDelimiter(','); //Delimiter
77	
78	\$Apriori->process(\$data); //override
79	
80	\$d['pfi1'] = \$Apriori->printFreqItemsets()[0];
81	\$d['pfi2'] = \$Apriori->printFreqItemsets()[1];
82	\$d['pfi3'] = \$Apriori->printFreqItemsets()[2]; //nama barang
83	\$d['pfi4'] = \$Apriori->printFreqItemsets()[3]; //percent
84	

85	\$d['pai1'] = \$Apriori->printAssociationRules()[0];
86	\$d['pai2'] = \$Apriori->printAssociationRules()[1];
87	\$d['pai3'] = \$Apriori->printAssociationRules()[2];
88	\$d['pai4'] = \$Apriori->printAssociationRules()[3];
89	echo json_encode(\$d);
90	}
91	}

Tabel 4.9 merupakan kode program yang terdapat pada package controllers yaitu class penjualan pada function `add_to_list`, yang digunakan untuk memanggil hasil dari perhitungan apriori pada class `M_cross_selling` untuk ditampilkan pada saat transaksi penjualan berlangsung.

4.5. Pengujian Sistem

Pengujian dilakukan untuk mengevaluasi aplikasi yang telah dibuat. Proses pengujian dilakukan dengan pengujian *whitebox* terlebih dahulu kemudian akan dilanjutkan dengan pengujian *blackbox*.

4.5.1 Pengujian White Box

Pengujian sistem dengan metode *white box* dilakukan untuk menguji sistem dari segi desain dan kode program. Hal tersebut bertujuan untuk mengevaluasi apakah sistem mampu menghasilkan fungsi-fungsi, inputan, dan keluaran yang sesuai dengan spesifikasi dari kebutuhan sistem itu sendiri. Pengujian dengan metode *White Box* dilakukan dengan cara menghitung independent path yaitu dengan menggunakan pengukuran kuantitatif *cyclomatic complexity*, *listing* program, penentuan jalur independen, dan *test case*. Tahapan pengujian *White Box* meliputi:

a. Listing Program

Listing program merupakan baris-baris kode program yang nantinya akan diuji. Setiap langkah dari kode-kode yang terdapat pada program akan diberi nomor baik menjalankan statement biasa atau penggunaan kondisi dalam program. *Listing* program yang digunakan untuk pengujian *White box* yaitu pada class `cross_selling` untuk function `__construct()` dapat dilihat pada Gambar 4.9, function `index()` dapat

dilihat pada Gambar 4.10, *function form ()* dapat dilihat pada Gambar 4.11, *function process()* dapat dilihat pada Gambar 4.12, *function delete()* dapat dilihat pada Gambar 4.13. *function detailhasil()* dapat dilihat pada Gambar 4.14, pada class penjualan untuk *function add_to_list()* dapat dilihat pada Gambar 4.15.

```
10     function __construct() {
11         parent::__construct();
12
13         if ($this->session->userdata('level') != 1) {
14             redirect('dashboard');
15         }
16
17         $this->load->model('M_cross_selling');
18         $this->load->library('Aprioriclass');
19     }
```

Gambar 4. 9 Listing Program Function __construct()

```
21     function index() {
22         $isi = array(
23             'descript' => 'Daftar Produk Cross-selling'
24             , 'dataTable' => array()
25             , 'dataTable' => $this->M_cross_selling->get_list()
26             , 'add_button' => true
27             , 'back_button' => false
28         );
29
30         $this->set_page('cross_selling', $isi['descript'], 'cross_selling', $isi);
31     }
```

Gambar 4. 10 Listing Program Function index()

```
33     function form() {
34         $this->output->set_content_type('application/json');
35
36         $id = $_POST['id'];
37         echo json_encode($this->M_cross_selling->get_isi($id));
38     }
```

Gambar 4. 11 Listing Program Function form()

```

40 function process() {
41     $post = $_POST;
42     if ($this->M_cross_selling->process($post)) {
43         $this->session->set_userdata('pesan_sistem', 'Selamat! ' . (($post['id'] == '') ? 'Penambahan' : 'Perubahan') . ' Cross-selling, GA
44         $this->session->set_userdata('tipe_pesan', 'Sukses');
45         redirect('cross_selling');
46     } else {
47         $this->session->set_userdata('pesan_sistem', 'Maaf! ' . (($post['id'] == '') ? 'Penambahan' : 'Perubahan') . ' Cross-selling, GA
48         $this->session->set_userdata('tipe_pesan', 'Gagal');
49         redirect('cross_selling');
50     }
51 }

```

Gambar 4. 12 Listing Program Function process()

```

53 function delete($a) {
54     if ($this->M_cross_selling->delete($a)) {
55         $this->session->set_userdata('pesan_sistem', 'Selamat! Cross-selling telah dihapus!');
56         $this->session->set_userdata('tipe_pesan', 'Sukses');
57         redirect('cross_selling');
58     } else {
59         $this->session->set_userdata('pesan_sistem', 'Maaf! Cross-selling tidak terhapus! Silahkan periksa dan coba kembali!');
60         $this->session->set_userdata('tipe_pesan', 'Gagal');
61         redirect('cross_selling');
62     }
63 }

```

Gambar 4. 13 Listing Program Function delete()

```

65 function detailhasil() {
66     $this->output->set_content_type('application/json');
67
68     $id = $_POST['id'];
69     $minsup = $_POST['minsup'];
70     $minconf = $_POST['minconf'];
71     $data = $this->M_cross_selling->get_detailhasil($id);
72
73     $Apriori = new Aprioriclass();
74     $Apriori->setMaxScan(3); //Scan 2, 3, ...
75     $Apriori->setMinSup($minsup); //Minimum support 1, 2, 3, ...
76     $Apriori->setMinConf($minconf); //Minimum confidence - Percent 1, 2, ...
77     $Apriori->setDelimiter(','); //Delimiter
78
79     $Apriori->process($data); //override
80
81     $d['pfi1'] = $Apriori->printFreqItemsets()[0];
82     $d['pfi2'] = $Apriori->printFreqItemsets()[1];
83     $d['pfi3'] = $Apriori->printFreqItemsets()[2]; //nama barang
84     $d['pfi4'] = $Apriori->printFreqItemsets()[3]; //jml
85
86     $d['pai1'] = $Apriori->printAssociationRules()[0];
87     $d['pai2'] = $Apriori->printAssociationRules()[1];
88     $d['pai3'] = $Apriori->printAssociationRules()[2];
89     $d['pai4'] = $Apriori->printAssociationRules()[3];
90     echo json_encode($d);
91 }
92

```

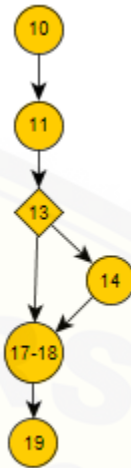
Gambar 4. 14 Listing Program Function detailhasil()

```
56 function add_to_list() {
57     $this->output->set_content_type('application/json');
58
59     $this->load->model('M_cross_selling');
60     $this->load->library('Aprioriclass');
61     $post = $_POST;
62
63     if ($this->M_penjualan->process($post)) {
64         $wedhus = $this->M_cross_selling->get_last();
65
66         $id = $wedhus[0]['id'];
67         $minsup = $wedhus[0]['minsup'];
68         $minconf = $wedhus[0]['minconf'];
69         $data = $this->M_cross_selling->get_detailhasil($id);
70
71
72         $Apriori = new Aprioriclass();
73         $Apriori->setMaxScan(3); //Scan 2, 3, ...
74         $Apriori->setMinSup($minsup); //Minimum support 1, 2, 3, ...
75         $Apriori->setMinConf($minconf); //Minimum confidence - Percent 1, 2, ..., 100
76         $Apriori->setDelimiter(','); //Delimiter
77
78         $Apriori->process($data); //override
79
80         $d['pfi1'] = $Apriori->printFreqItemsets()[0];
81         $d['pfi2'] = $Apriori->printFreqItemsets()[1];
82         $d['pfi3'] = $Apriori->printFreqItemsets()[2]; //nama barang
83         $d['pfi4'] = $Apriori->printFreqItemsets()[3]; //jml
84
85         $d['pai1'] = $Apriori->printAssociationRules()[0];
86         $d['pai2'] = $Apriori->printAssociationRules()[1];
87         $d['pai3'] = $Apriori->printAssociationRules()[2];
88         $d['pai4'] = $Apriori->printAssociationRules()[3];
89         echo json_encode($d);
90     }
91 }
92
```

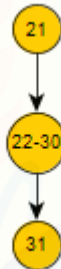
Gambar 4. 15 Listing Program Function add_to_list()

b. Diagram Alir

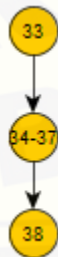
Diagram alir merupakan notasi sederhana yang digunakan untuk merepresentasikan aliran kontrol. Aliran kontrol yang digambarkan merupakan hasil penomoran dari listing program. Diagram alir digambarkan dengan *node-node* (simpul) yang dihubungkan dengan *edge-edge* (garis) yang menggambarkan alur jalannya program. Diagram alir yang digunakan untuk pengujian *White box* yaitu pada class *cross_selling* untuk *function__construct()* dapat dilihat pada Gambar 4.16, *function index()* dapat dilihat pada Gambar 4.17, *function form ()* dapat dilihat pada Gambar 4.18, *function process()* dapat dilihat pada Gambar 4.19, *function delete()* dapat dilihat pada Gambar 4.20. *function detailhasil()* dapat dilihat pada Gambar 4.21, pada class *penjualan* untuk *function add_to_list()* dapat dilihat pada Gambar 4.22.



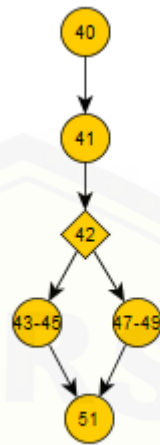
Gambar 4. 16 Diagram Alir Function `__construct()`



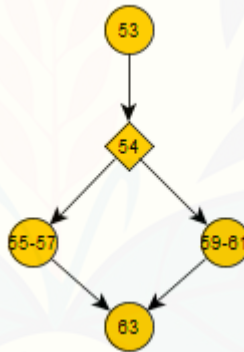
Gambar 4. 17 Diagram Alir Function `index()`



Gambar 4. 18 Diagram Alir Function `form()`



Gambar 4. 19 Diagram Alir Function process()



Gambar 4. 20 Diagram Alir Function delete()



Gambar 4. 21 Diagram Alir Function detailhasil()



Gambar 4. 22 Diagram Alir Function add_to_list()

c. *Cyclomatic complexity*

Cyclomatic complexity merupakan metrik perangkat lunak yang menyediakan ukuran kuantitatif dari kompleksitas logis suatu program. Perhitungan *Cyclomatic complexity* dari listing program dan diagram alir yang telah dibuat adalah sebagai berikut :

1. *Function__construct()*

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

$$V(G) = 6 - 6 + 2 = 2$$

2. *Function index()*

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

$$V(G) = 2 - 3 + 2 = 1$$

3. *Function* form()

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

$$V(G) = 2 - 3 + 2 = 1$$

4. *Function* process()

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

$$V(G) = 6 - 6 + 2 = 2$$

5. *Function* delete()

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

$$V(G) = 5 - 5 + 2 = 2$$

6. *Function* detailhasil()

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

$$V(G) = 9 - 10 + 2 = 1$$

7. *Function* add_to_list

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

$$V(G) = 12 - 12 + 2 = 2$$

d. Jalur Program Independen

Jalur independen adalah setiap jalur yang melalui program yang memperkenalkan setidaknya satu kumpulan pernyataan-pernyataan pemrosesan atau kondisi baru. Bila dinyatakan dalam grafik alir, jalur independen harus bergerak setidaknya sepanjang satu edge yang belum dilintasi sebelum jalur tersebut didefinisi. Jalur independen dari *function* yang digunakan untuk pengujian adalah sebagai berikut :

1. *Function* __construct()

$$\text{Jalur 1} = 10-11-13-14 -17-18-19$$

$$\text{Jalur 2} = 10, 11, 13, 17-18, 19$$

2. *Function* index()

$$\text{Jalur 1} = 21-22-30-31$$

3. *Function* form()

$$\text{Jalur 1} = 33-34-37-38$$

4. *Function* process()

Jalur 1 = 40-41-42-43-45-51

Jalur 2 = 40-41-42-47-49-51

5. *Function* delete()

Jalur 1 = 53-54-55-57-63

Jalur 2 = 53-54-59-61-63

6. *Function* detailhasil()

Jalur 1 = 65-66-73-74-75-76-77-79-82-83-84-90-91

7. *Function* add_to_list

Jalur 1 = 56-57-61-63-64-76-73-74-75-76-78-81-82-83-90-91

Jalur 2 = 56-57-61-63-91

e. Pengujian Basis Set (*Test Case*)

Tabel 4. 10 Test Case Function __construct()

<i>Test Case function</i> __construct()	
Jalur1	
<i>Test Case</i>	Jika <i>function</i> __construct() dijalankan pada baris 10-19
Target yang diharapkan	Berhasil menampilkan halaman dashboard
Hasil pengujian	Benar
Path/Jalur	10-11-13-14 -17-18-19
Jalur2	
<i>Test Case</i>	Jika <i>function</i> __construct() dijalankan pada baris 10-19
Target yang diharapkan	Berhasil memanggil class model <i>M_cross_selling</i> dan class library <i>aprioriclass</i>
Hasil pengujian	Benar
Path/ Jalur	10, 11, 13, 17-18, 19

Tabel 4. 11 Test Case Function index()

<i>Test Case function index()</i>	
Jalur1	
<i>Test Case</i>	Jika <i>function index()</i> dijalankan pada baris 21-31
Target yang diharapkan	Berhasil mengambil data dengan memanggil function <i>get_list()</i> pada class models <i>M_cross_selling</i> dalam bentuk array \$isi dan menampilkannya dalam class views data
Hasil pengujian	Benar
Path/ Jalur	21-22-30-31

Tabel 4. 12 Test Case Function form()

<i>Test Case function form()</i>	
Jalur1	
<i>Test Case</i>	Jika <i>function form()</i> dijalankan pada baris 33-38
Target yang diharapkan	Berhasil mengambil data dengan memanggil function <i>get_isi()</i> pada class models <i>M_cross_selling</i> dan menampilkannya dalam class views data
Hasil pengujian	Benar
Path/ Jalur	33-34-37-38

Tabel 4. 13 Test Case Function process()

<i>Test Case function process()</i>	
Jalur1	
<i>Test Case</i>	Jika <i>function process()</i> dijalankan pada baris 40-51
Target yang diharapkan	Berhasil menyimpan isian pada form tambah pengaturan <i>cross-selling</i> produk menggunakan function <i>save()</i> pada

	class <i>M_cross_selling</i> dan memanggil class controller <i>cross_selling</i> .
Hasil pengujian	Benar
Path/ Jalur	40-41-42-43-45-51
Jalur2	
<i>Test Case</i>	Jika <i>function</i> <i>process()</i> dijalankan pada baris 40-51
Target yang diharapkan	Berhasil menampilkan pesan kesalahan pada pengisian form tambah pengaturan <i>cross-selling</i> dan menampilkannya pada class views data.
Hasil pengujian	Benar
Path/ Jalur	40-41-42-47-49-51

Tabel 4. 14 Test Case Function delete()

<i>Test Case function delete()</i>	
Jalur1	
<i>Test Case</i>	Jika <i>function</i> <i>delete()</i> dijalankan pada baris 53-63
Target yang diharapkan	Berhasil menghapus data pengaturan paket produk sesuai id yang dipilih
Hasil pengujian	Benar
Path/ Jalur	53-54-55-57-63
Jalur2	
<i>Test Case</i>	Jika <i>function</i> <i>delete()</i> dijalankan pada baris 53-63
Target yang diharapkan	Berhasil menampilkan pesan kesalahan pada saat menghapus pengaturan <i>cross-selling</i> dan menampilkannya pada class views data.
Hasil pengujian	Benar
Path/ Jalur	53-54-59-61-63

Tabel 4. 15 Test Case Function detailhasil()

<i>Test Case function detailhasil()</i>	
Jalur1	
<i>Test Case</i>	Jika <i>function detailhasil()</i> dijalankan pada baris 65-91
Target yang diharapkan	Berhasil mengambil data pengaturan <i>cross-selling</i> produk sesuai dengan id yang dipilih dan mengambil data dari function <i>get_detailhasil()</i> pada class model <i>M_cross_selling</i> . Proses selanjutnya memanggil class library <i>aprioriclass</i> dengan <i>meng-override</i> sesuai pengaturan pola yang dipilih dan menyimpan dalam bentuk array <i>\$data</i> sehingga data hasil dapat dilihat pada class view <i>cross_selling</i>
Hasil pengujian	Benar
Path/ Jalur	65-66-73-74-75-76-77-79-82-83-84-90-91

Tabel 4. 16 Test Case Function *add_to_list()*

<i>Test Case function add_to_list()</i>	
Jalur1	
<i>Test Case</i>	Jika <i>function add_to_list()</i> dijalankan pada baris 56-91
Target yang diharapkan	Berhasil mengambil data pengaturan <i>cross-selling</i> produk sesuai dengan id terakhir dari pengaturan <i>cross-selling</i> yang sesuai dengan nama produk yang sedang di tambahkan di penjualan dan mengambil data dari function <i>get_last()</i> dan <i>get_detailhasil()</i> pada class model <i>M_cross_selling</i> . Proses selanjutnya memanggil class library <i>aprioriclass</i> dengan <i>meng-override</i> sesuai pengaturan pola yang dipilih dan menyimpan dalam bentuk array <i>\$data</i> sehingga data hasil dapat dilihat pada class view penjualan
Hasil pengujian	Benar
Path/ Jalur	56-57-61-63-64-76-73-74-75-76-78-81-82-83-90-91
Jalur2	

<i>Test Case</i>	Jika <i>function</i> <code>add_to_list()</code> dijalankan pada baris 56-91
Target yang diharapkan	Berhasil menambah barang pada form penjualan
Hasil pengujian	Benar
Path/ Jalur	56-57-61-63-91

4.5.2 Pengujian *Black Box*

Pengujian *black box* berfungsi untuk menguji sistem dari segi spesifikasi fungsional sistem dengan tujuan mengetahui apakah fungsi-fungsi, *inputan*, dan keluaran sistem sesuai dengan spesifikasi yang dibutuhkan oleh pengguna. Hasil pengujian dengan metode *black box* dapat dilihat pada Lampiran 4.

BAB 6. PENUTUP

Bab ini berisi mengenai 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 hasil penelitian yang telah dilakukan oleh peneliti adalah sebagai berikut:

1. Penerapan algoritma Apriori pada sistem informasi *cross-selling* produk dapat membantu mengetahui kombinasi produk yang sering dibeli oleh konsumen dan membentuk kombinasi *cross-selling* produk yang dilakukan oleh manager sesuai dengan periode yang diinginkan. Manager dapat menentukan nilai minimum *support* dan nilai minimal *confidence* serta tanggal yang akan menjadi acuan dalam perhitungan.
2. Penggunaan algoritma Apriori pada sistem informasi informasi *cross-selling* produk telah dapat membantu dalam pembentukan kombinasi *cross-selling* produk. Pada penelitian ini dataset yang digunakan sebanyak 10 transaksi yang terdiri dari 13 produk. Untuk satu transaksi terdiri dari 2 sampai 4 produk. Pada penentuan *cross-selling* produk ditentukan batas minimum *support*nya sebesar 2 dan minimal *confidencenya* sebesar 50%. Hasil rekomendasi *cross-selling* produk yang ditemukan dari 10 transaksi pada Tabel 5.10 yaitu Happy Tos dan Aqua dengan persentase sebesar 50%, Happy Tos dan Hidro Coco dengan persentase 50%, Happy Tos dan Mr Bread Roti Kasur dengan persentase 50%, Aqua dan Happy Tos dengan persentase 66.67%, Hidro Coco dan Happy Tos dengan persentase 100%, Mr Bread Roti Kasur dan Happy Tos dan nilai 66,67%, Indomilk dan Sari Roti dengan persentase 100%, Sari Roti dan Indomilk dengan persentase 100%.
3. Sistem informasi *cross-selling* produk pada KPRI Universitas Jember dibangun berbasis *website* dengan tiga hak akses yaitu manager, gudang dan kasir dengan

fitur utama penentuan kombinasi *cross-selling* produk. Sistem ini dikembangkan dengan menggunakan model *waterfall*, dimana kebutuhan sistem telah terdefiniskan diawal pembangunan system. Proses pengerjaan model *waterfall* dilakukan secara terurut, jika terdapat perubahan atau penambahan fitur maka proses analisa akan dilakukan kembali. Tahap selanjutnya adalah pengembangan yang meliputi analisis, desain, penulisan kode program dan pengujian sistem. Tahap desain dilakukan dengan melakukan perancangan sistem dengan membuat *bussines process, usecase diagram, scenario, activity diagram, sequence diagram, class diagram* dan ERD yang digunakan sebagai acuan dalam penulisan kode program. Tahap penulisan kode program menggunakan Bahasa pemrograman *PHP, framework CodeIgniter*, dan *database* manajemen *MySQL*. Tahap pengujian dilakukan dengan cara metode *black box* dan *white box*.

4. Pada penerapan algoritma Apriori pada sistem informasi *cross-selling* produk, semakin tinggi nilai minimum *support* dan nilai minimal *confidence* yang ditentukan, maka akan semakin sedikit hasil *cross-selling* produk yang ditemukan.

6.2. Saran

Saran penulis untuk pengembangan lebih lanjut untuk penelitian ini dapat dilakukan dengan menggunakan metode lain yang memungkinkan dalam penentuan *cross-selling* produk tetapi dengan objek yang sama. Diharapkan pula untuk peneliti selanjutnya memiliki fitur yang lebih lengkap.

DAFTAR PUSTAKA

- Gunadi, G., & Sensue, D. (2012). *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.*
- Kamakura, W. A. (2007). *Cross-selling : Offering The Right Product to The Right Customer at The Right Time.* Durham: Duke University.
- Kumar, M. A. (2012). *CROSS SELLING (With Special Reference to State Bank Of India).* Bareilly.
- Pane, D. K. (2013). *Implementasi Data Mining Pada Penjualan Produk Elektronik dengan Algoritma Apriori (Studi Kasus : Kreditplus).* Medan: Pelita Informatika Budi Darma.
- Priyana, F. A., & Kardianawati, A. (2015). *Data Mining Asosiasi Untuk Menentukan Cross-selling Produk Menggunakan Algoritma Frequent Pattern-Growth Pada Koperasi Karyawan PT. PHAROS Semarang.*
- Rahayu, R. (2014). *Implementasi Agoritma Apriori Dan FP-Growth Dalam Market Basket Analysis Pembelian Obat Pada Suatu Apotek.* Medan.
- Sanda, R. (2015). *Analisa Pengaruh Kualitas Layanan Terhadap Loyalitas Konsumen Mahasiswa Pada KPRI Universitas Jember.* Jember.
- Siregar, S. R. (2014). *Implementasi Data Mining pada Penjualan Tiket Pesawat Menggunakan Algoritma Apriori (Studi Kasus : Jumbo Travel Medan).* Medan: Pelita Informatika Budi Darma.
- Syaifullah, M. A. (2010). *Implementasi Data Mining Algoritma Apriori pada Sistem Penjualan.* Yogyakarta.

LAMPIRAN

LAMPIRAN 1 (Skenario Sistem)

1.1. Skenario Mengelola Data *Login*

No Usecase	UCS 001
Nama Usecase	Login
Aktor	All Aktor
Deskripsi	Aktor dapat mengakses menu jika berhasil login
Prekondisi	Aktor belum melakukan login
Prakondisi	Aktor telah berhasil login
Flow of Events	
Skenario Normal : Login	
Aktor	Sistem
1. Membuka Sistem <i>Cross-selling</i> Produk KPRI	
	2. Menampilkan halaman utama login Sistem <i>Cross-selling</i> Produk KPRI
3. Masukkan Username dan Password	
4. Klik Tombol Sign In	
	5. Menampilkan Halaman Dashboard
Skenario Alternative : Username atau Password Salah	
Aktor	Simpan
4. Klik Tombol Sign In	
	5. Menampilkan alert “Username atau Password SALAH! Silahkan periksa dan coba kembali”
	6. Menampilkan halaman utama Sistem <i>Cross-selling</i> Produk KPRI

1.2. Skenario Mengelola Data *User*

No Usecase	UCS 002
Nama Usecase	Mengelola data User
Aktor	Manager
Deskripsi	Aktor dapat membuat data User

Prekondisi	Manager telah melakukan login
Prakondisi	Manager berhasil mengelola data user
Flow of Events	
Skenario Normal : Buat Data User	
Aktor	Sistem
1. Klik Menu User	
	2. Menampilkan daftar User
3. Klik tombol Tambah	
	4. Menampilkan form User untuk menambahkan User baru
5. Mengisi form User yang ditampilkan sistem	
6. Klik tombol “Save”	
	7. Menampilkan notif “Selamat! Penambahan user, SUKSES!”
	8. Menampilkan daftar User
Skenario Alternatif : Batal Menambahkan Data User	
Aktor	Sistem
5. Mengisi form User yang ditampilkan sistem	
6. Klik tombol “Close”	
	7. Menampilkan daftar User
Skenario Alternatif : Data Tidak Lengkap	
Aktor	Sistem
5. Mengisi form User yang ditampilkan sistem	
6. Klik tombol “Save”	
	7. Menampilkan alert “This field is required” pada form User
Skenario Normal : Melihat Data User	
Aktor	Sistem
1. Klik Menu User	
	2. Menampilkan daftar User
3. Memasukkan data yang akan dicari pada <i>field</i> “Search”	

	4. Menampilkan baris data User yang dicari
5. Klik tombol “Detail”	
	6. Menampilkan data pada form User
7. Klik tombol “Close”	
	8. Menampilkan daftar User
Skenario Normal : Ubah Data User	
Aktor	Sistem
1. Klik Menu User	
	2. Menampilkan daftar User
3. Memasukkan data yang akan dicari pada <i>field</i> “Search”	
	4. Menampilkan baris data User yang dicari
5. Klik tombol “Ubah”	
	6. Menampilkan data form user, antara lain:
7. Memperbarui data User yang ditampilkan di form	
8. Klik “Save”	
	9. Menampilkan notif “Selamat! Perubahan user, SUKSES!”
	10. Menampilkan daftar User
Skenario Alternatif : Batal Menyimpan Data User	
Aktor	Sistem
7. Memperbarui data User yang ditampilkan di form	
8. Klik tombol “Close”	
	9. Menampilkan daftar User
Skenario Alternatif : Data Tidak Lengkap	
Aktor	Sistem
7. Memperbarui data User yang ditampilkan di form	
8. Klik tombol “Save”	
	9. Menampilkan alert “This field is required” pada form User

Skenario Normal : Delete Data User	
Aktor	Sistem
1. Klik Menu User	
	2. Menampilkan daftar User
3. Memasukkan data yang akan dicari pada <i>field</i> "Search"	
	4. Menampilkan baris data User yang dicari
5. Klik tombol "Hapus"	
	6. Menampilkan popup "Apakah Anda yakin menghapus data penting ini?"
7. Klik "Oke"	
	8. Menampilkan notif "Selamat! User telah dihapus!"
	9. Menampilkan daftar User
Skenario Alternatif : Batal Menghapus Data User	
Aktor	Sistem
3. Memasukkan data yang akan dicari pada <i>field</i> "Search"	
	4. Menampilkan baris data User yang dicari
5. Klik tombol "Hapus"	
	6. Menampilkan popup "Apakah Anda yakin menghapus data penting ini?"
7. Klik "Batal"	
	8. Menampilkan daftar User

1.3. Skenario Melihat Laporan Transfer Barang

No Usecase	UCS 003
Nama Usecase	Melihat Laporan Transfer Barang
Aktor	Manager
Deskripsi	Aktor dapat melihat Laporan Transfer Barang
Prekondisi	Manager telah melakukan login

Prakondisik	Manager telah melihat laporan transfer barang
Flow of Events	
Skenario Normal : Melihat Laporan Transfer Barang	
Aktor	Sistem
1. Klik Menu Laporan Transfer Barang	
	2. Menampilkan data Laporan Transfer Barang
3. Klik tombol “Show Form”	
	4. Menampilkan form tanggal
5. Mengisi form	
	6. Menampilkan data Laporan Transfer Barang

1.4. Skenario Mengelola Data Member

No Usecase	UCS 005
Nama Usecase	Mengelola data Member
Aktor	Manager
Deskripsi	Aktor dapat membuat data Member
Prekondisi	Manager telah melakukan login
Prakondisik	Manager telah berhasil mengelola data member
Flow of Events	
Skenario Normal : Buat Data Member	
Aktor	Sistem
1. Klik Menu Member	
	2. Menampilkan daftar Member
3. Klik tombol Tambah	
	4. Menampilkan form Member untuk menambahkan Member baru
5. Mengisi form Member yang ditampilkan sistem	
6. Klik tombol “Save”	

	7. Menampilkan notif “Selamat! Penambahan member, SUKSES!”
	8. Menampilkan daftar Member
Skenario Alternatif : Batal Menambahkan Data Member	
Aktor	Simpan
5. Mengisi form Member yang ditampilkan sistem	
6. Klik tombol “Close”	
	7. Menampilkan daftar Member
Skenario Alternatif : Data Tidak Lengkap	
Aktor	Sistem
5. Mengisi form Member yang ditampilkan sistem	
6. Klik tombol “Save”	
	7. Menampilkan alert “This field is required” pada form Member
Skenario Normal : Lihat Data Member	
Aktor	Sistem
1. Klik Menu Member	
	2. Menampilkan daftar Member
3. Memasukkan data yang akan dicari pada <i>field</i> “Search”	
	4. Menampilkan baris data Member yang dicari
5. Klik tombol “Detail”	
	6. Menampilkan data pada form Member
7. Klik tombol “Close”	
	8. Menampilkan daftar Member
Skenario Normal : Ubah Data Member	
Aktor	Sistem
1. Klik Menu Member	
	2. Menampilkan daftar Member
3. Memasukkan data yang akan dicari pada <i>field</i> “Search”	

	4. Menampilkan baris data Member yang dicari
5. Klik tombol “Ubah”	
	6. Menampilkan data form Member
7. Memperbarui data Member yang ditampilkan di form	
8. Klik “Save”	
	9. Menampilkan notif “Selamat! Perubahan member, SUKSES!”
	10. Menampilkan daftar Member
Skenario Alternatif : Batal Menyimpan Data Member	
Aktor	Sistem
7. Memperbarui data Member yang ditampilkan di form	
8. Klik tombol “Close”	
	9. Menampilkan daftar Member
Skenario Alternatif : Data Tidak Lengkap	
Aktor	Sistem
7. Memperbarui data Member yang ditampilkan di form	
8. Klik tombol “Save”	
	9. Menampilkan alert “This field is required” pada form Member
Skenario Normal : Hapus Data Member	
Aktor	Sistem
1. Klik Menu Member	
	2. Menampilkan daftar Member
3. Memasukkan data yang akan dicari pada <i>field</i> “Search”	
	4. Menampilkan baris data Member yang dicari
5. Klik tombol “Hapus”	
	6. Menampilkan popup “Apakah Anda yakin menghapus data penting ini?”
7. Klik “Oke”	

	8. Menampilkan notif “Selamat! Member telah dihapus!”
	9. Menampilkan daftar Member
Skenario Alternatif: Batal Menghapus Data Member	
Aktor	Sistem
3. Memasukkan data yang akan dicari pada <i>field</i> “Search”	
	4. Menampilkan baris data Member yang dicari
5. Klik tombol “Hapus”	
	6. Menampilkan popup “Apakah Anda yakin menghapus data penting ini?”
7. Klik “Batal”	
	8. Menampilkan daftar Member

1.5. Skenario Melihat Data Supplier

No Usecase	UCS 006
Nama Usecase	Melihat data Supplier
Aktor	Manager
Deskripsi	Aktor dapat melihat data Supplier
Prekondisi	Manager telah melakukan login
Prakondisik	Manager telah berhasil melihat data supplier
Flow of Events	
Skenario Normal : Lihat Data Supplier	
Aktor	Sistem
1. Klik Menu Supplier	
	2. Menampilkan daftar Supplier
3. Memasukkan data yang akan dicari pada <i>field</i> “Search”	
	4. Menampilkan baris data Supplier yang dicari
5. Klik tombol “Detail”	
	6. Menampilkan data pada form Supplier
7. Klik tombol “Close”	

	8. Menampilkan daftar Supplier
Skenario Normal : Lihat Data Daftar Barang	
Aktor	Sistem
1. Klik Menu Supplier	
	2. Menampilkan daftar Supplier
3. Klik tombol Daftar Barang	
	4. Menampilkan daftar barang yang dimiliki supplier
5. Klik tombol "Close"	
	6. Menampilkan daftar Supplier

1.6. Skenario Melihat Data Kategori Produk

No Usecase	UCS 007
Nama Usecase	Melihat data Kategori Produk
Aktor	Manager
Deskripsi	Aktor dapat melihat data Kategori
Prekondisi	Manager telah melakukan login
Prakondisik	Manager telah berhasil melihat data kategori produk
Flow of Events	
Skenario Normal : Lihat Data Kategori	
Aktor	Sistem
1. Klik Menu Kategori	
	2. Menampilkan daftar Kategori
3. Memasukkan data yang akan dicari pada <i>field</i> "Search"	
	4. Menampilkan baris data Kategori yang dicari
5. Klik tombol "Detail"	
	6. Menampilkan data pada form Kategori
7. Klik tombol "Close"	
	8. Menampilkan daftar Kategori

1.7. Skenario Melihat Laporan Barang Masuk

No Usecase	UCS 008
Nama Usecase	Melihat Laporan Barang Masuk
Aktor	Manager
Deskripsi	Aktor dapat melihat Laporan Barang Masuk
Prekondisi	Manager telah melakukan login
Prakondisik	Manager telah berhasil melihat Laporan Barang Masuk
Flow of Events	
Skenario Normal : Melihat Laporan Barang Masuk	
Aktor	Sistem
1. Klik Menu Laporan Barang Masuk	
	2. Menampilkan data Laporan Barang Masuk
3. Klik tombol “Show Form”	
	4. Menampilkan form tanggal
5. Mengisi form	
	6. Menampilkan data Laporan Barang Masuk

1.8. Skenario Melihat Laporan Penjualan

No Usecase	UCS 009
Nama Usecase	Melihat Laporan Penjualan
Aktor	Manager
Deskripsi	Aktor dapat melihat Laporan Penjualan
Prekondisi	Manager telah melakukan login
Prakondisik	Manager telah berhasil melihat Laporan Penjualan
Flow of Events	
Skenario Normal : Melihat Laporan Penjualan	
Aktor	Sistem
1. Klik Menu Laporan Penjualan	
	2. Menampilkan data Laporan Penjualan
3. Klik tombol “Show Form”	
	4. Menampilkan form tanggal

5. Mengisi form	
	6. Menampilkan data Laporan Penjualan

1.9. Skenario Mengelola Pembelian Barang

No Usecase	USC 010
Nama Usecase	Mengelola data Pembelian Barang
Aktor	Gudang
Deskripsi	Aktor dapat membuat data Pembelian Barang
Prekondisi	Gudang telah melakukan login
Prakondisik	Gudang telah berhasil mengelola Data Pembelian Barang
Flow of Events	
Skenario Normal : Buat Data Pembelian	
Aktor	Sistem
1. Klik Menu Pembelian	
	2. Menampilkan form Pembelian
3. Mengisi form Pembelian	
4. Klik tombol “Tambah Ke List”	
	5. Menampilkan daftar Pembelian
6. Klik tombol “Save”	
	7. Daftar Pembelian akan kosong
Skenario Alternatif: Batal Menambahkan Data Pembelian	
Aktor	Sistem
3. Mengisi form Pembelian	
4. Klik tombol “Reset”	
	5. Menampilkan daftar Pembelian
Skenario Alternatif : Data Tidak Lengkap	
Aktor	Sistem
3. Mengisi form Pembelian	
4. Klik tombol “Tambah Ke List”	
	5. Menampilkan form Pembelian

Skenario Normal : Ubah Data Pembelian	
Aktor	Sistem
1. Klik tombol “Ubah” pada data barang yang akan diubah	
	2. Form akan terisi dengan data barang yang akan diubah
3. Merubah data barang	
4. Klik tombol “Tambah Ke List”	
	5. Menampilkan daftar Pembelian
Skenario Alternatif : Batal Mengubah Data Pembelian	
Aktor	Sistem
3. Memperbarui data	
4. Klik tombol “Reset”	
	5. Menampilkan daftar Pembelian
Skenario Alternatif : Data Tidak Lengkap	
Aktor	Sistem
3. Memperbarui data	
4. Klik tombol “Tambah Ke List”	
	5. Menampilkan form Pembelian
Skenario Alternatif : Hapus Data Pembelian	
Aktor	Sistem
1. Klik tombol “Hapus” pada data barang yang akan di hapus	
	2. Menghapus data yang dipilih
	3. Menampilkan daftar Pembelian

1.10. Skenario Mengelola Data Supplier

No Usecase	UCS 011
Nama Usecase	Mengelola data Supplier
Aktor	Gudang
Deskripsi	Aktor dapat membuat data Supplier
Prekondisi	Gudang telah melakukan login
Prakondisik	Gudang telah berhasil mengelola Data Supplier

Flow of Events	
Skenario Normal : Buat Data Supplier	
Aktor	Sistem
1. Klik Menu Supplier	
	2. Menampilkan daftar Supplier
3. Klik tombol Tambah	
	4. Menampilkan form Supplier untuk menambahkan Supplier baru
5. Mengisi form Supplier yang ditampilkan sistem	
6. Klik tombol "Save"	
	7. Menampilkan notif "Selamat! Penambahan supplier, SUKSES!"
	8. Menampilkan daftar Supplier
Skenario Alternatif : Batal Menambahkan Data Supplier	
Aktor	Sistem
5. Mengisi form Supplier yang ditampilkan sistem	
6. Klik tombol "Close"	
	7. Menampilkan daftar Supplier
Skenario Alternatif : Data Tidak Lengkap	
Aktor	Sistem
5. Mengisi form Supplier yang ditampilkan sistem	
6. Klik tombol "Save"	
	7. Menampilkan alert "This field is required" pada form Supplier
Skenario Normal : Lihat Data Supplier	
Aktor	Sistem
1. Klik Menu Supplier	
	2. Menampilkan daftar Supplier
3. Memasukkan data yang akan dicari pada <i>field</i> "Search"	
	4. Menampilkan baris data Supplier yang dicari

5. Klik tombol “Detail”	
	6. Menampilkan data pada form Supplier
7. Klik tombol “Close”	
	8. Menampilkan daftar Supplier
Skenario Normal : Ubah Data Supplier	
Aktor	Sistem
1. Klik Menu Supplier	
	2. Menampilkan daftar Supplier
3. Memasukkan data yang akan dicari pada <i>field</i> “Search”	
	4. Menampilkan baris data Supplier yang dicari
5. Klik tombol “Ubah”	
	6. Menampilkan data form Supplier
7. Memperbarui data Supplier yang ditampilkan di form	
8. Klik “Save”	
	9. Menampilkan notif “Selamat! Perubahan Supplier, SUKSES!”
	10. Menampilkan daftar Supplier
Skenario Alternatif : Batal Menyimpan Data Supplier	
Aktor	Sistem
7. Memperbarui data Supplier yang ditampilkan di form	
8. Klik tombol “Close”	
	9. Menampilkan daftar Supplier
Skenario Alternatif : Data Tidak Lengkap	
Aktor	Sistem
7. Memperbarui data Supplier yang ditampilkan di form	
8. Klik tombol “Save”	
	9. Menampilkan alert “This field is required” pada form Supplier
Skenario Normal : Hapus Data Supplier	
Aktor	Sistem

1. Klik Menu Supplier	
	2. Menampilkan daftar Supplier
3. Memasukkan data yang akan dicari pada <i>field</i> "Search"	
	4. Menampilkan baris data Supplier yang dicari
5. Klik tombol "Hapus"	
	6. Menampilkan popup "Apakah Anda yakin menghapus data penting ini?"
7. Klik "Oke"	
	8. Menampilkan notif "Selamat! Supplier telah dihapus!"
	9. Menampilkan daftar Supplier
Skenario Alternatif : Batal Menghapus Data Supplier	
Aktor	Sistem
3. Memasukkan data yang akan dicari pada <i>field</i> "Search"	
	4. Menampilkan baris data Supplier yang dicari
5. Klik tombol "Hapus"	
	6. Menampilkan popup "Apakah Anda yakin menghapus data penting ini?"
7. Klik "Batal"	
	8. Menampilkan daftar Supplier
Skenario Normal : Lihat Data Daftar Barang	
Aktor	Sistem
1. Klik Menu Supplier	
	2. Menampilkan daftar Supplier
3. Klik tombol Daftar Barang	
	4. Menampilkan daftar barang yang dimiliki supplier
5. Klik tombol "Close"	
	6. Menampilkan daftar Supplier

1.11. Skenario Mengelola Data Kategori Produk

No Usecase	UCS 012
Nama Usecase	Mengelola data Kategori Produk
Aktor	Gudang
Deskripsi	Aktor dapat membuat data Kategori
Prekondisi	Gudang telah melakukan login
Prakondisik	Gudang telah berhasil mengelola Data Kategori
Flow of Events	
Skenario Normal : Buat Data Kategori	
Aktor	Sistem
1. Klik Menu Kategori	
	2. Menampilkan daftar Kategori
3. Klik tombol Tambah	
	4. Menampilkan form Kategori untuk menambahkan kategori baru
5. Mengisi form Kategori yang ditampilkan sistem	
6. Klik tombol "Save"	
	7. Menampilkan notif "Selamat! Penambahan Kategori, SUKSES!"
	8. Menampilkan daftar Kategori
Skenario Alternatif : Batal Menambahkan Data Kategori	
Aktor	Sistem
5. Mengisi form Kategori yang ditampilkan sistem	
6. Klik tombol "Close"	
	7. Menampilkan daftar Kategori
Skenario Alternatif : Data Tidak Lengkap	
Aktor	Sistem
5. Mengisi form Kategori yang ditampilkan sistem	
6. Klik tombol "Save"	
	7. Menampilkan alert "This field is required" pada form Kategori

Skenario Normal : Lihat Data Kategori	
Aktor	Sistem
1. Klik Menu Kategori	
	2. Menampilkan daftar Kategori
3. Memasukkan data yang akan dicari pada <i>field</i> "Search"	
	4. Menampilkan baris data Kategori yang dicari
5. Klik tombol "Detail"	
	6. Menampilkan data pada form Kategori
7. Klik tombol "Close"	
	8. Menampilkan daftar Kategori
Skenario Normal : Ubah Data Kategori	
Aktor	Sistem
1. Klik Menu Kategori	
	2. Menampilkan daftar Kategori
3. Memasukkan data yang akan dicari pada <i>field</i> "Search"	
	4. Menampilkan baris data Kategori yang dicari
5. Klik tombol "Ubah"	
	6. Menampilkan data form Kategori
7. Memperbarui data Kategori yang ditampilkan di form	
8. Klik "Save"	
	9. Menampilkan notif "Selamat! Perubahan Kategori, SUKSES!"
	10. Menampilkan daftar Kategori
Skenario Alternatif : Batal Menyimpan Data Kategori	
Aktor	Sistem
7. Memperbarui data Kategori yang ditampilkan di form	
8. Klik tombol "Close"	
	9. Menampilkan daftar Kategori
Skenario Alternatif : Data Tidak Lengkap	
Aktor	Sistem

7. Memperbarui data Kategori yang ditampilkan di form	
8. Klik tombol "Save"	
	9. Menampilkan alert "This field is required" pada form Kategori
Skenario Normal : Hapus Data Kategori	
Aktor	Sistem
1. Klik Menu Kategori	
	2. Menampilkan daftar Kategori
3. Memasukkan data yang akan dicari pada field "Search"	
	4. Menampilkan baris data Kategori yang dicari
5. Klik tombol "Hapus"	
	6. Menampilkan popup "Apakah Anda yakin menghapus data penting ini?"
7. Klik "Oke"	
	8. Menampilkan notif "Selamat! Kategori telah dihapus!"
	9. Menampilkan daftar Member
Skenario Alternatif : Batal Menghapus Data Kategori	
Aktor	Sistem
3. Memasukkan data yang akan dicari pada <i>field</i> "Search"	
	4. Menampilkan baris data Kategori yang dicari
5. Klik tombol "Hapus"	
	6. Menampilkan popup "Apakah Anda yakin menghapus data penting ini?"
7. Klik "Batal"	
	8. Menampilkan daftar Kategori

1.12. Skenario Mengelola Transfer Barang

No Usecase	USC 013
Nama Usecase	Mengelola data Transfer Barang
Aktor	Gudang
Deskripsi	Aktor dapat membuat data Transfer Barang
Prekondisi	Manager telah melakukan login
Prakondisik	Gudang telah berhasil mengelola Data Transfer Barang
Flow of Events	
Skenario Normal : Buat Data Transfer Barang	
Aktor	Sistem
1. Klik Menu Transfer Barang	
	2. Menampilkan form Transfer Barang
3. Mengisi form Transfer Barang	
4. Klik tombol “Tambah Ke List”	
	5. Menampilkan daftar Transfer Barang
6. Klik tombol “Save”	
	7. Daftar Transfer Barang akan kosong
Skenario Alternatif : Batal Menambahkan Data Transfer Barang	
Aktor	Sistem
3. Mengisi form Transfer Barang	
4. Klik tombol “Reset”	
	5. Menampilkan daftar Transfer Barang
Skenario Alternatif : Data Tidak Lengkap	
Aktor	Sistem
3. Mengisi form Transfer Barang	
4. Klik tombol “Tambah Ke List”	
	5. Menampilkan form Transfer Barang
Skenario Normal : Ubah Data Transfer Barang	
Aktor	Sistem

1. Klik tombol “Ubah” pada data barang yang akan diedit	
	2. Form akan terisi dengan data barang yang akan diubah
3. Merubah data barang	
4. Klik tombol “Tambah Ke List”	
	5. Menampilkan daftar Transfer Barang
Skenario Alternatif : Batal Mengubah Data Transfer Barang	
Aktor	Sistem
3. Memperbarui data	
4. Klik tombol “Reset”	
	5. Menampilkan daftar Transfer Barang
Skenario Alternatif : Data Tidak Lengkap	
Aktor	Sistem
3. Memperbarui data	
4. Klik tombol “Tambah Ke List”	
	5. Menampilkan form Transfer Barang
Skenario Normal : Hapus Data Transfer Barang	
Aktor	Sistem
1. Klik tombol “Hapus” pada data barang yang akan di hapus	
	2. Menghapus data yang dipilih
	3. Menampilkan daftar Transfer Barang

1.13. Skenario Mengelola Data Barang

No Usecase	UCS 014
Nama Usecase	Mengelola data Daftar Barang
Aktor	Gudang
Deskripsi	Aktor dapat membuat data Kategori
Prekondisi	Gudang telah melakukan login

Prakondisik	Gudang telah berhasil mengelola Data Daftar Barang
Flow of Events	
Skenario Normal : Buat Data Daftar Barang	
Aktor	Sistem
1. Klik Menu Daftar Barang	
	2. Menampilkan daftar Daftar Barang
3. Klik tombol Tambah	
	4. Menampilkan form Daftar Barang untuk menambahkan kategori baru
5. Mengisi form Daftar Barang yang ditampilkan sistem	
6. Klik tombol "Save"	
	7. Menampilkan notif "Selamat! Penambahan Daftar Barang, SUKSES!"
	8. Menampilkan daftar Daftar Barang
Skenario Alternatif : Batal Menambahkan Data Daftar Barang	
Aktor	Sistem
5. Mengisi form Daftar Barang yang ditampilkan sistem	
6. Klik tombol "Close"	
	7. Menampilkan daftar Daftar Barang
Skenario Alternatif : Data Tidak Lengkap	
Aktor	Sistem
5. Mengisi form Daftar Barang yang ditampilkan sistem	
6. Klik tombol "Save"	
	7. Menampilkan alert "This field is required" pada form Daftar Barang
Skenario Normal : Lihat Data Daftar Barang	
Aktor	Sistem
1. Klik Menu Daftar Barang	
	2. Menampilkan daftar Daftar Barang

3. Memasukkan data yang akan dicari pada <i>field</i> "Search"	
	4. Menampilkan baris data Daftar Barang yang dicari
5. Klik tombol "Detail"	
	6. Menampilkan data pada form Daftar Barang
7. Klik tombol "Close"	
	8. Menampilkan daftar Daftar Barang
Skenario Normal : Ubah Data Daftar Barang	
Aktor	Sistem
1. Klik Menu Daftar Barang	
	2. Menampilkan daftar Daftar Barang
3. Memasukkan data yang akan dicari pada <i>field</i> "Search"	
	4. Menampilkan baris data Daftar Barang yang dicari
5. Klik tombol "Ubah"	
	6. Menampilkan data form barang
7. Memperbarui data Daftar Barang yang ditampilkan di form	
8. Klik "Save"	
	9. Menampilkan notif "Selamat! Perubahan Daftar Barang, SUKSES!"
	10. Menampilkan daftar Daftar Barang
Skenario Alternatif : Batal Menyimpan Data Daftar Barang	
Aktor	Sistem
7. Memperbarui data Daftar Barang yang ditampilkan di form	
8. Klik tombol "Close"	
	9. Menampilkan daftar Daftar Barang
Skenario Alternatif : Data Tidak Lengkap	
Aktor	Sistem
7. Memperbarui data Daftar Barang yang ditampilkan di form	

8. Klik tombol “Save”	
	9. Menampilkan alert “This field is required” pada form Daftar Barang
Skenario Normal : Hapus Data Daftar Barang	
Aktor	Sistem
1. Klik Menu Daftar Barang	
	2. Menampilkan daftar Daftar Barang
3. Memasukkan data yang akan dicari pada <i>field</i> “Search”	
	4. Menampilkan baris data Daftar Barang yang dicari
5. Klik tombol “Hapus”	
	6. Menampilkan popup “Apakah Anda yakin menghapus data penting ini?”
7. Klik “Oke”	
	8. Menampilkan notif “Selamat! Daftar Barang telah dihapus!”
	9. Menampilkan daftar Daftar Barang
Skenario Alternatif: Batal Menghapus Data Daftar Barang	
Aktor	Sistem
3. Memasukkan data yang akan dicari pada <i>field</i> “Search”	
	4. Menampilkan baris data Daftar Barang yang dicari
5. Klik tombol “Hapus”	
	6. Menampilkan popup “Apakah Anda yakin menghapus data penting ini?”
7. Klik “Batal”	
	8. Menampilkan daftar Daftar Barang

1.14. Skenario Mengelola Penjualan

No Usecase	UCS 016
Nama Usecase	Mengelola Penjualan Barang
Aktor	Kasir

Deskripsi	Aktor dapat membuat Data Penjualan Barang
Prekondisi	Kasir telah melakukan login
Prakondisi	Kasir telah berhasil mengelola Data Penjualan Barang
Flow of Events	
Skenario Normal : Buat Data Penjualan	
Aktor	Sistem
1. Klik Menu Penjualan	
	2. Menampilkan Form Penjualan dan Tabel Data Barang
3. Mengisi Form Penjualan yang ditampilkan sistem	
4. Klik tombol “Tambah Ke List”	
	5. Menampilkan daftar Penjualan disertai form member dan diskon
6. Klik tombol “Print”	
	7. Menampilkan pop-up form “Uang Pembayaran :”
8. Mengisi pop-up form	
9. Klik tombol “OK”	
	10. Menampilkan pop-up “Uang kembalian = “
11. Klik tombol “OK”	
	12. Menampilkan halaman print sistem
13. Klik tombol “Print”	
	14. Menampilkan notif “Selamat ! Penjualan Barang, SUKSES!”
	15. Daftar Penjualan Barang akan kosong
Skenario Alternatif : Batal Menambahkan Data Penjualan	
Aktor	Sistem

3. Mengisi form Penjualan yang ditampilkan sistem	
4. Klik tombol “Reset”	
	5. Menampilkan daftar Penjualan pada Tabel
Skenario Alternatif : Data Tidak Lengkap	
Aktor	Sistem
3. Mengisi Form Penjualan barang yang ditampilkan sistem	
4. Klik tombol “Tambah Ke List”	
	5. Menampilkan alert “This field is required” pada form Penjualan
Skenario Normal : Batal Mencetak Penjualan	
Aktor	Sistem
6. Klik tombol “Print”	
	7. Menampilkan pop-up form “Uang Pembayaran : ”
8. Mengisi pop-up form	
9. Klik tombol “OK”	
	10. Menampilkan pop-up “Uang kembalikan = “
11. Klik tombol “OK”	
	12. Menampilkan halaman print sistem
13. Klik tombol “Cancel”	
	14. Menampilkan daftar penjualan barang
Skenario Normal : Ubah Data Penjualan	
Aktor	Sistem
1. Klik tombol “Ubah” pada data penjualan barang yang akan di edit	
	2. Form akan terisi dengan data penjualan barang yang akan diubah

3. Merubah data penjualan barang	
4. Klik tombol “Tambah Ke List”	
	5. Menampilkan daftar Penjualan
Skenario Alternatif : Batal Mengubah Data Penjualan	
Aktor	Sistem
3. Memperbarui data Penjualan	
4. Klik tombol “Reset”	
	5. Menampilkan daftar Penjualan
Skenario Alternatif : Data Tidak Lengkap	
Aktor	Sistem
3. Memperbarui data Penjualan Obat	
4. Klik tombol “Tambah Ke List”	
	5. Menampilkan alert “This field is required” pada form Penjualan
Skenario Normal : Hapus Data Penjualan	
Aktor	Sistem
1. Klik tombol “Hapus” pada Data Penjualan barang yang akan di hapus	
	2. Menghapus Data yang dipilih
	3. Menampilkan daftar Penjualan

1.15. Skenario Melihat Data Member

No Usecase	UCS 017
Nama Usecase	Melihat data Member
Aktor	Kasir
Deskripsi	Aktor dapat melihat data Member
Prekondisi	Manager telah melakukan login
Prakondisik	Kasir telah berhasil melihat Data Member
Flow of Events	
Skenario Normal : Lihat Data Member	
Aktor	Sistem
1. Klik Menu Member	
	2. Menampilkan daftar Member

3. Memasukkan data yang akan dicari pada field "Search"	
	4. Menampilkan baris data Member yang dicari
5. Klik tombol "Detail"	
	6. Menampilkan data pada form Member
7. Klik tombol "Close"	
	8. Menampilkan daftar Member

1.16. Skenario Mengubah Data Barang

No Usecase	UCS 018
Nama Usecase	Mengubah data Daftar Barang
Aktor	Gudang
Deskripsi	Aktor dapat mengubah data Kategori
Prekondisi	Gudang telah melakukan login
Prakondisik	Gudang telah berhasil mengubah Data Daftar Barang
Flow of Events	
Skenario Normal : Ubah Data Daftar Barang	
Aktor	Sistem
1. Klik Menu Daftar Barang	
	2. Menampilkan daftar Daftar Barang
3. Memasukkan data yang akan dicari pada <i>field</i> "Search"	
	4. Menampilkan baris data Daftar Barang yang dicari
5. Klik tombol "Ubah"	
	6. Menampilkan data form barang
7. Memperbarui data Daftar Barang pada <i>field</i> "Harga Jual" yang ditampilkan di form	
8. Klik "Save"	
	9. Menampilkan notif " Selamat! Perubahan Daftar Barang, SUKSES! "

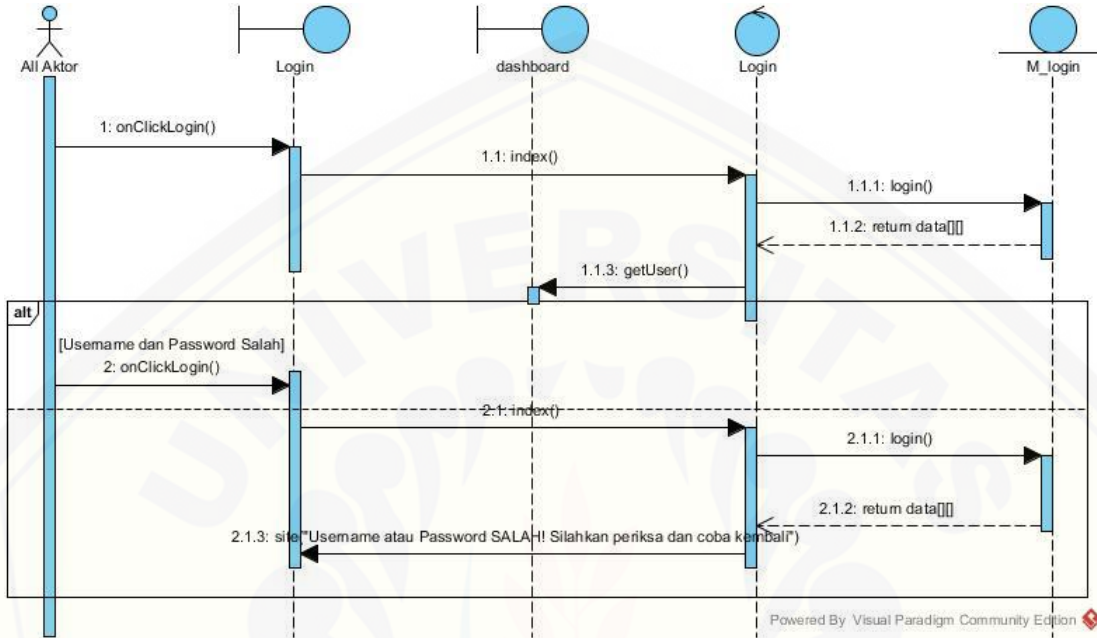
	10. Menampilkan daftar Daftar Barang
Skenario Alternatif : Batal Menyimpan Data Daftar Barang	
Aktor	Sistem
7. Memperbarui data Daftar Barang yang ditampilkan di form	
8. Klik tombol “Close”	
	9. Menampilkan daftar Daftar Barang
Skenario Alternatif : Data Tidak Lengkap	
Aktor	Sistem
7. Memperbarui data Daftar Barang yang ditampilkan di form	
8. Klik tombol “Save”	
	9. Menampilkan alert “This field is required” pada form Daftar Barang

1.17. Skenario Mengelola Logout

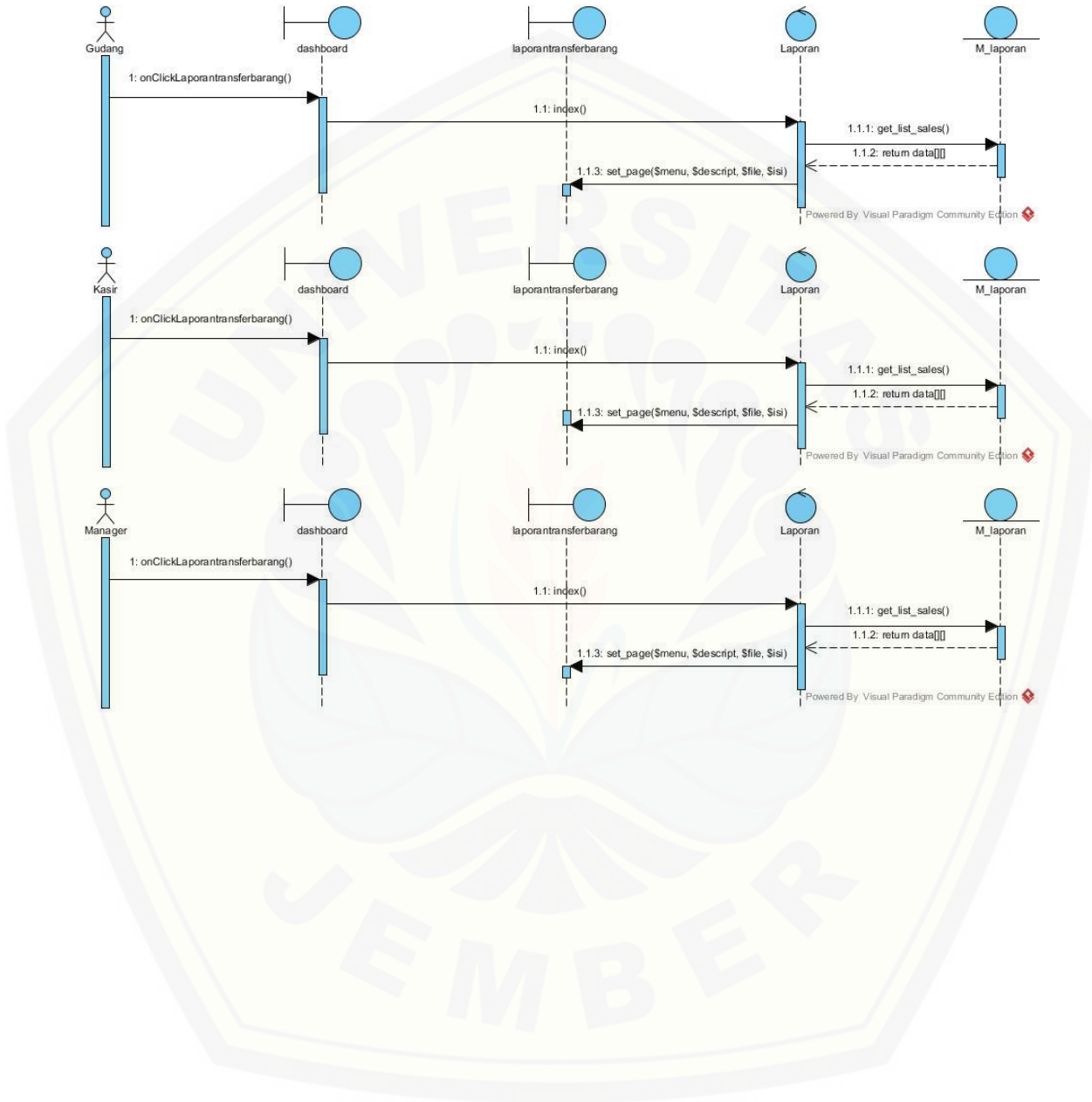
No Usecase	UCS 021
Nama Usecase	Log Out
Aktor	All actor
Deskripsi	Aktor Keluar dari Aplikasi
Prekondisi	Aktor telah melakukan login
Prakondisi	Aktor telah berhasil logout
Flow of Events	
Skenario Normal : Log Out	
Aktor	Sistem
1. Klik Tombol Log Out	
	2. Menampilkan halaman utama login Sistem <i>Cross-selling</i> Produk KPRI

LAMPIRAN 2 (Sequence Diagram)

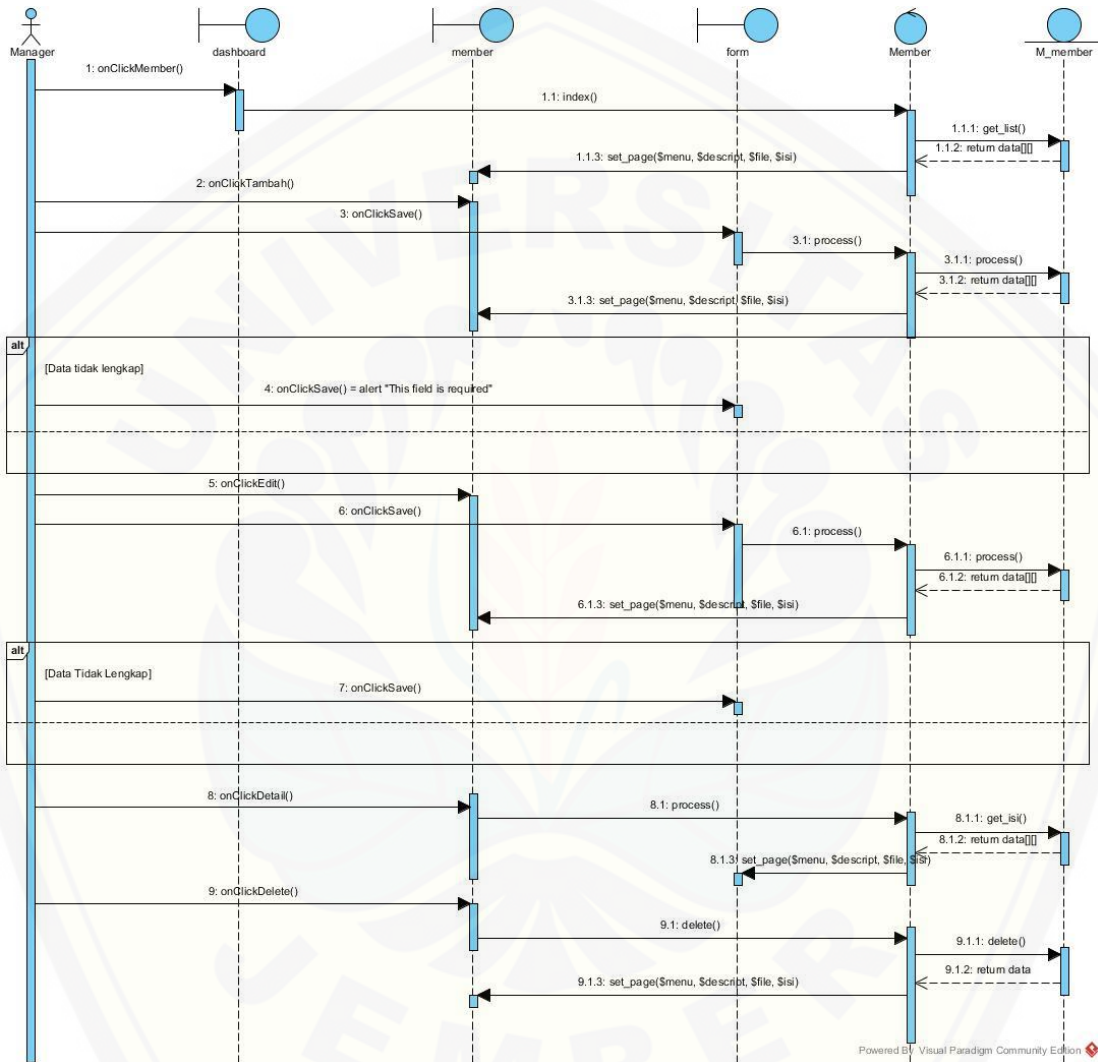
2.1 Sequence Diagram Mengelola Data Login



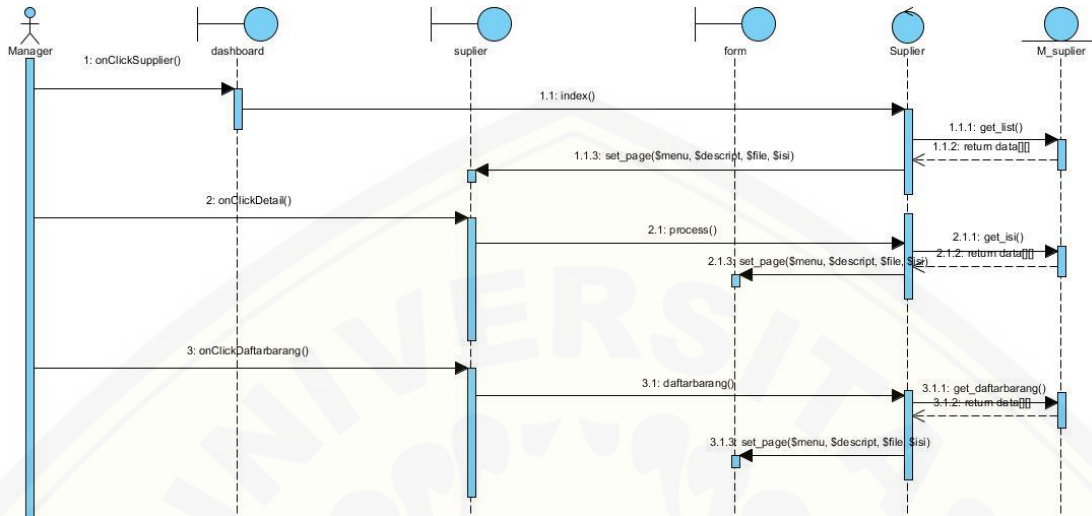
2.3 Sequence Diagram Melihat Laporan Transfer Barang



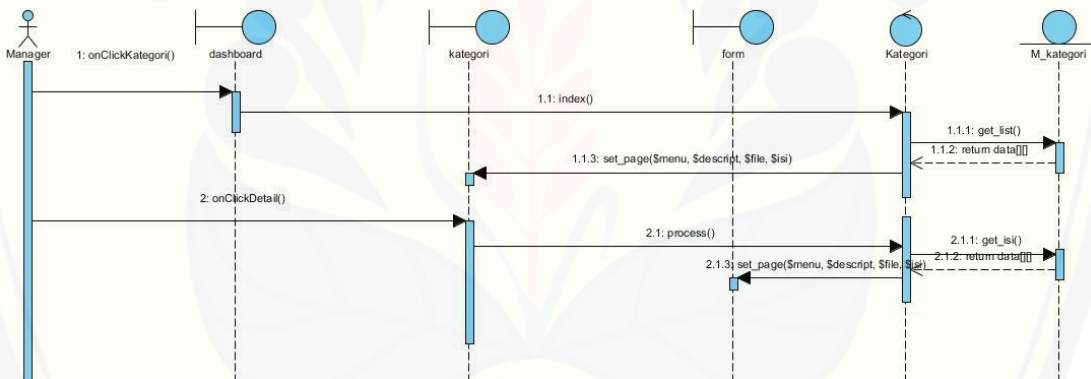
2.4 Sequence Diagram Mengelola Data Member



2.5 Sequence Diagram Melihat Data Supplier

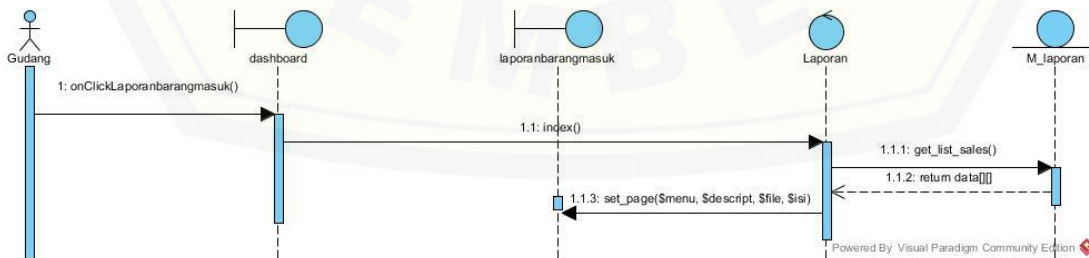


2.6 Sequence Diagram Melihat Data Kategori Produk

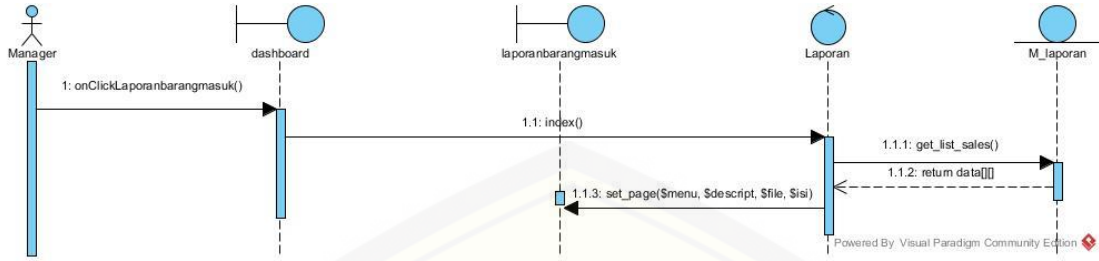


2.7 Sequence Diagram Melihat Laporan Barang Masuk

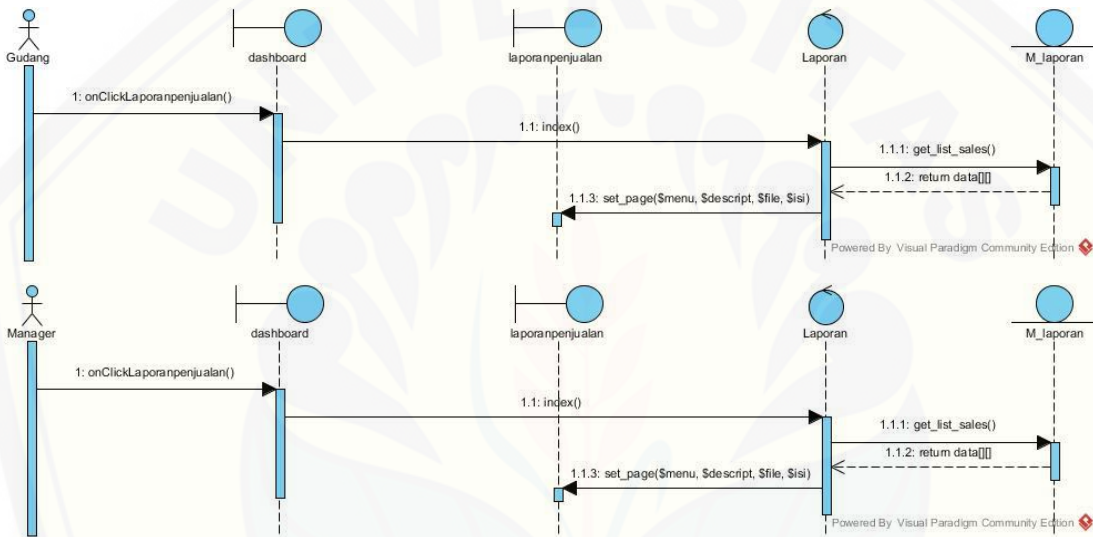
1. Gudang



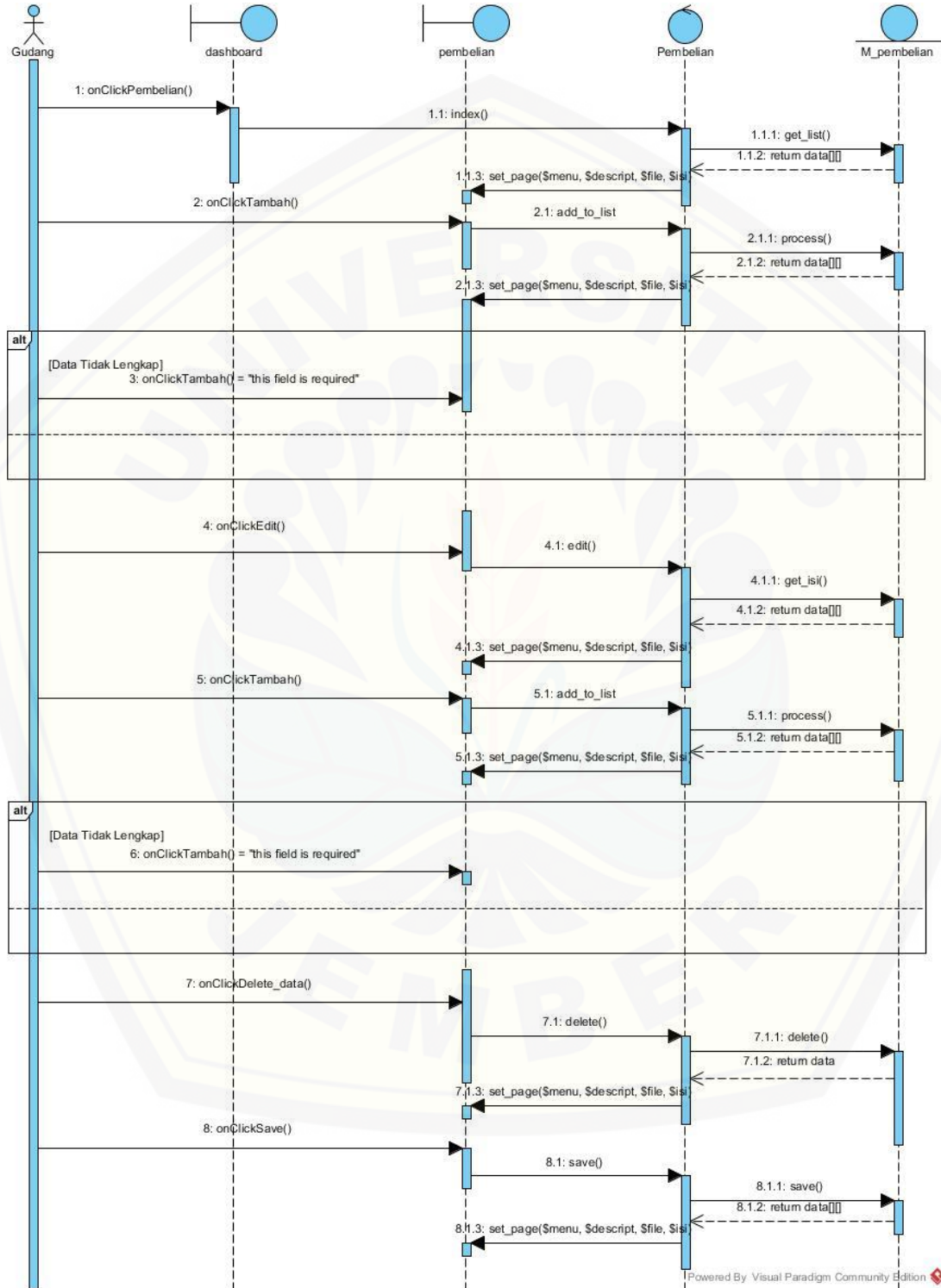
2. Manager



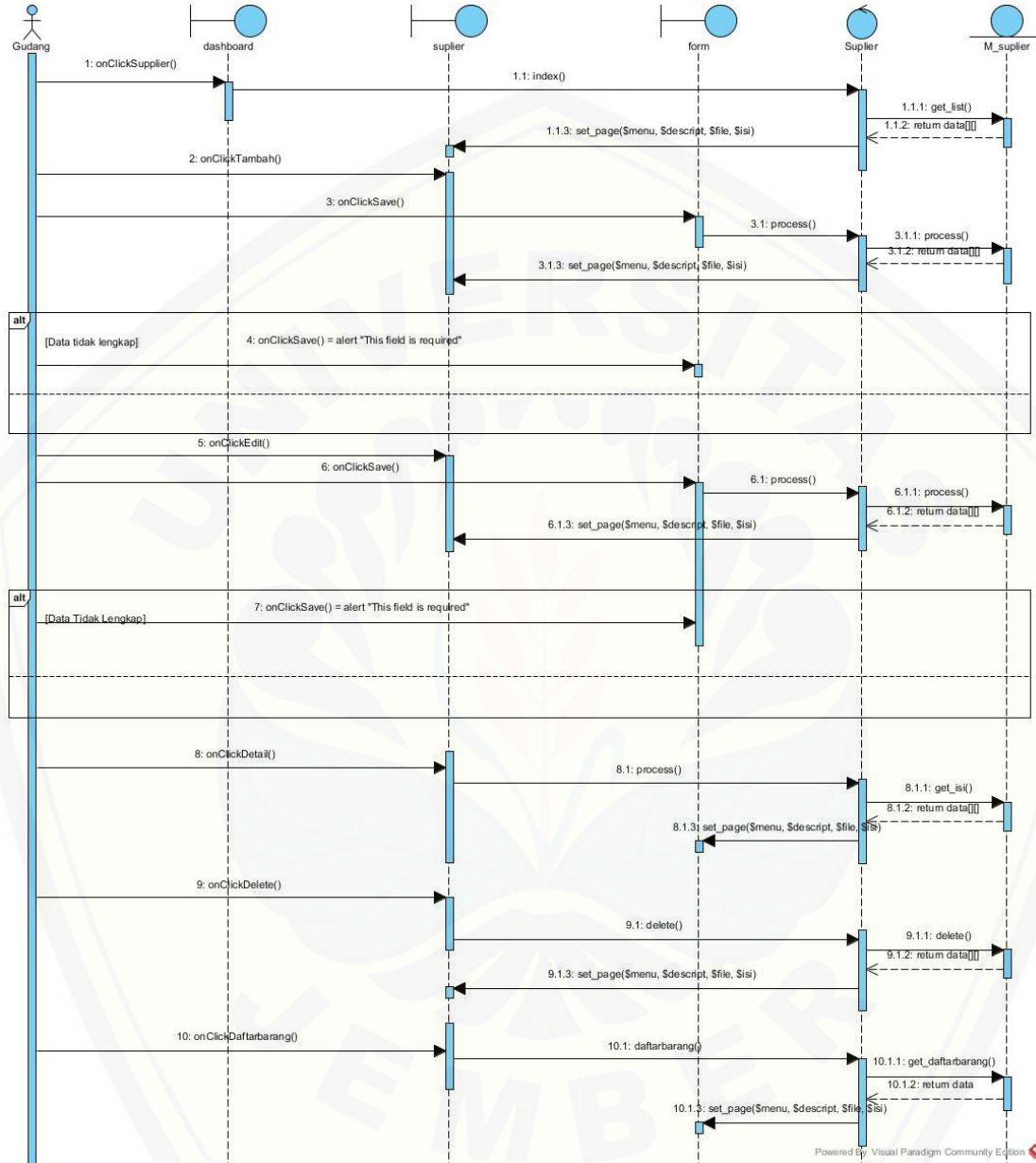
2.8 Sequence Diagram Melihat Laporan Penjualan



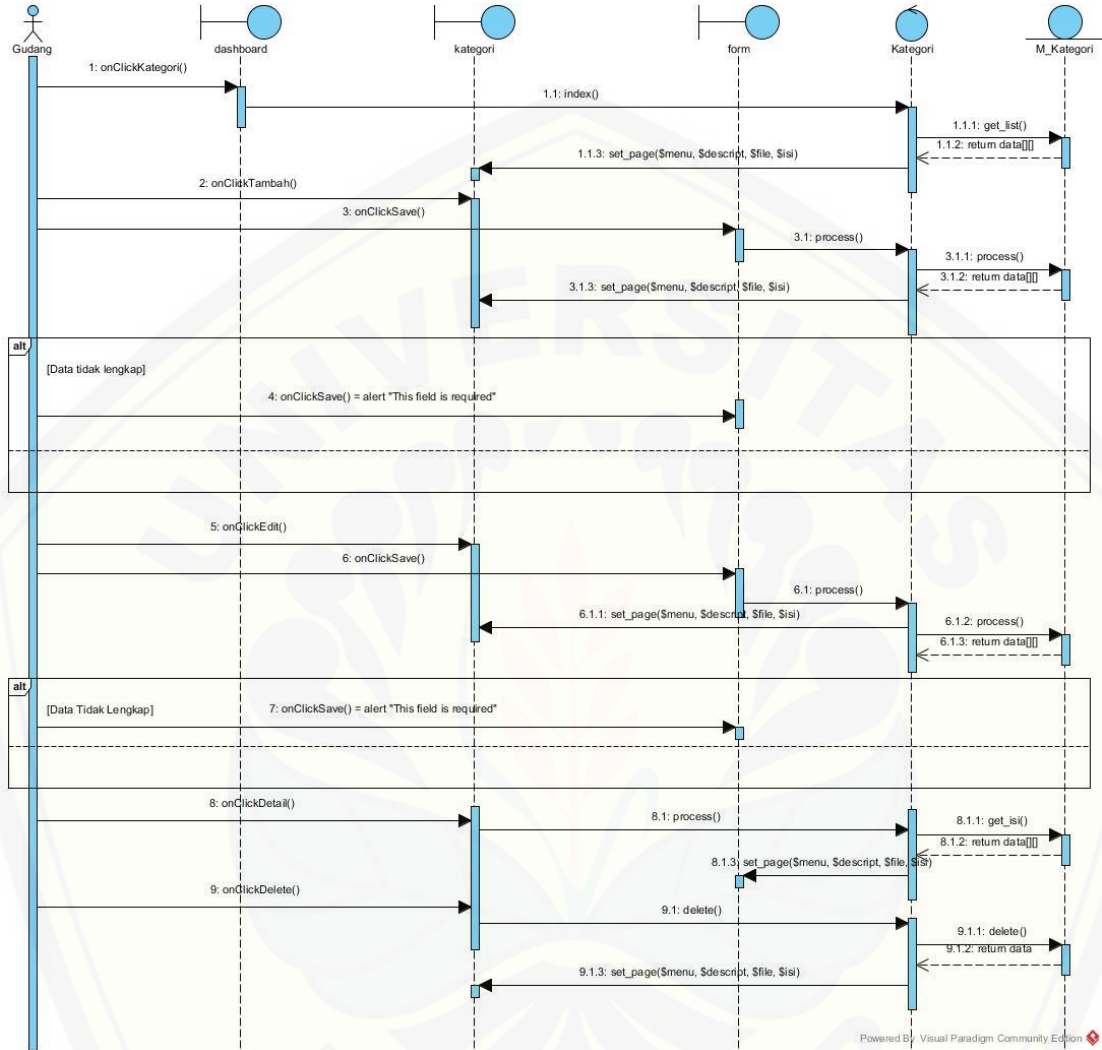
2.9 Sequence Diagram Mengelola Pembelian Barang



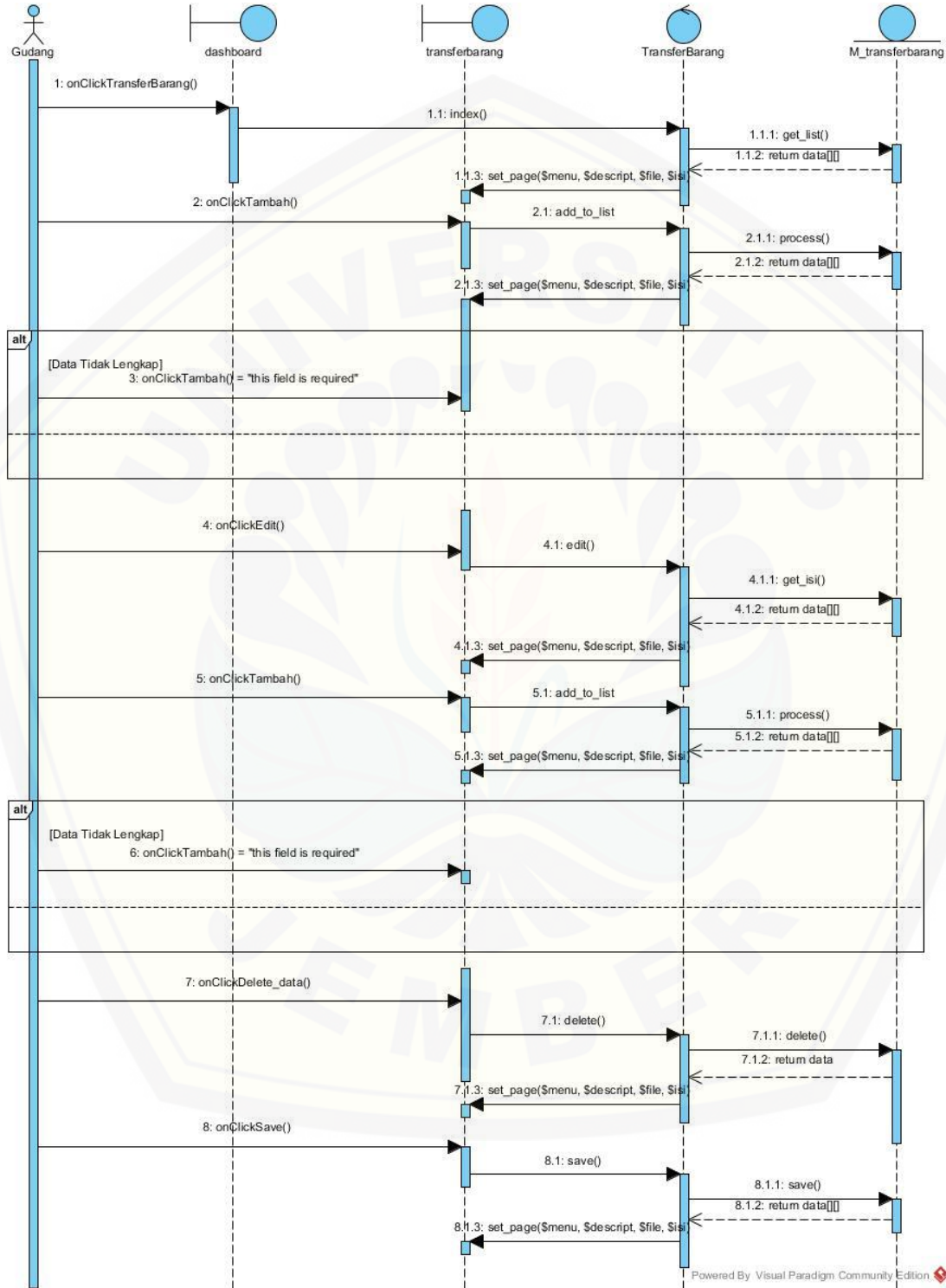
2.10 Sequence Diagram Mengelola Data Suplier



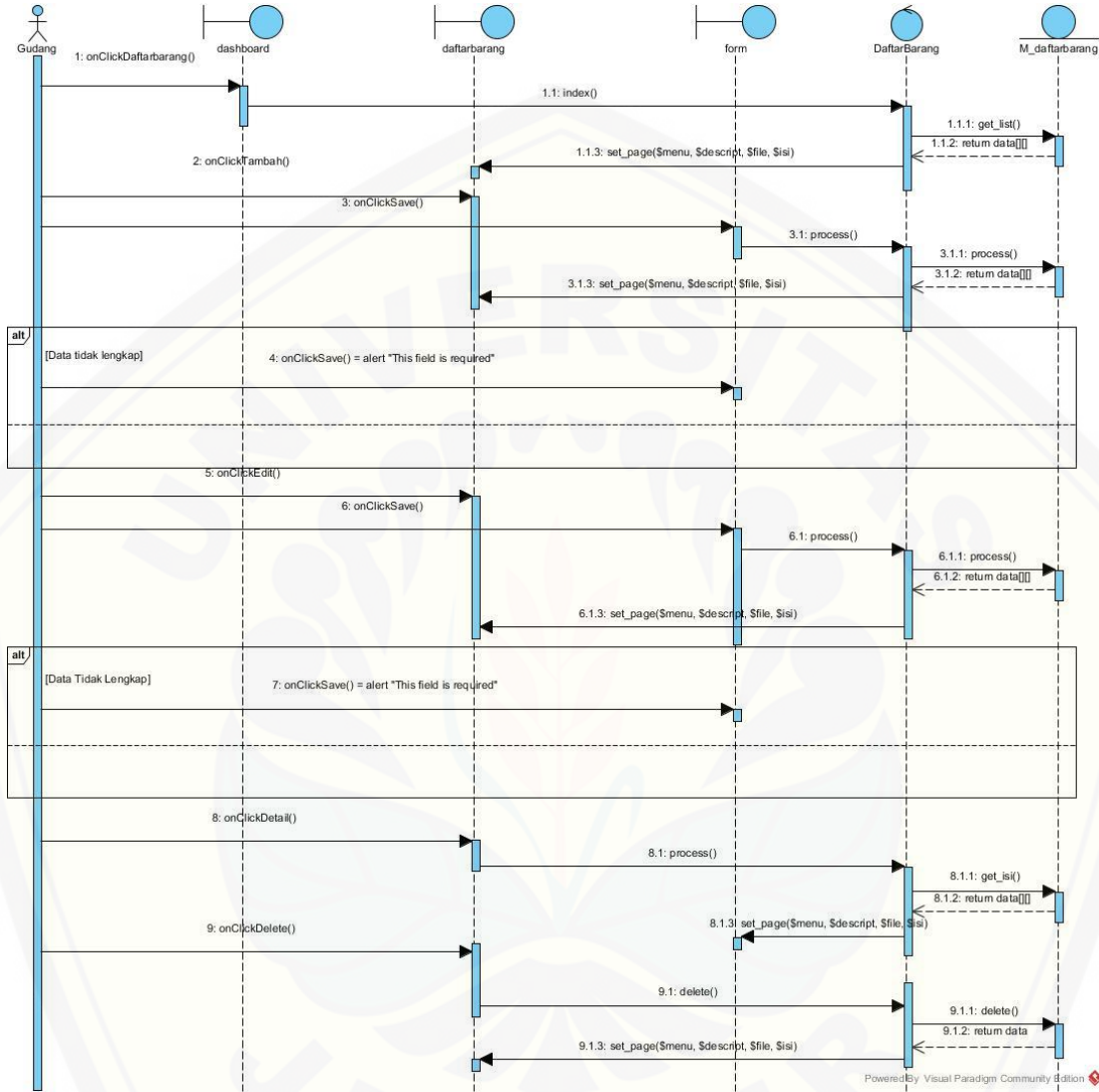
2.11 Sequence Diagram Mengelola Data Kategori Produk



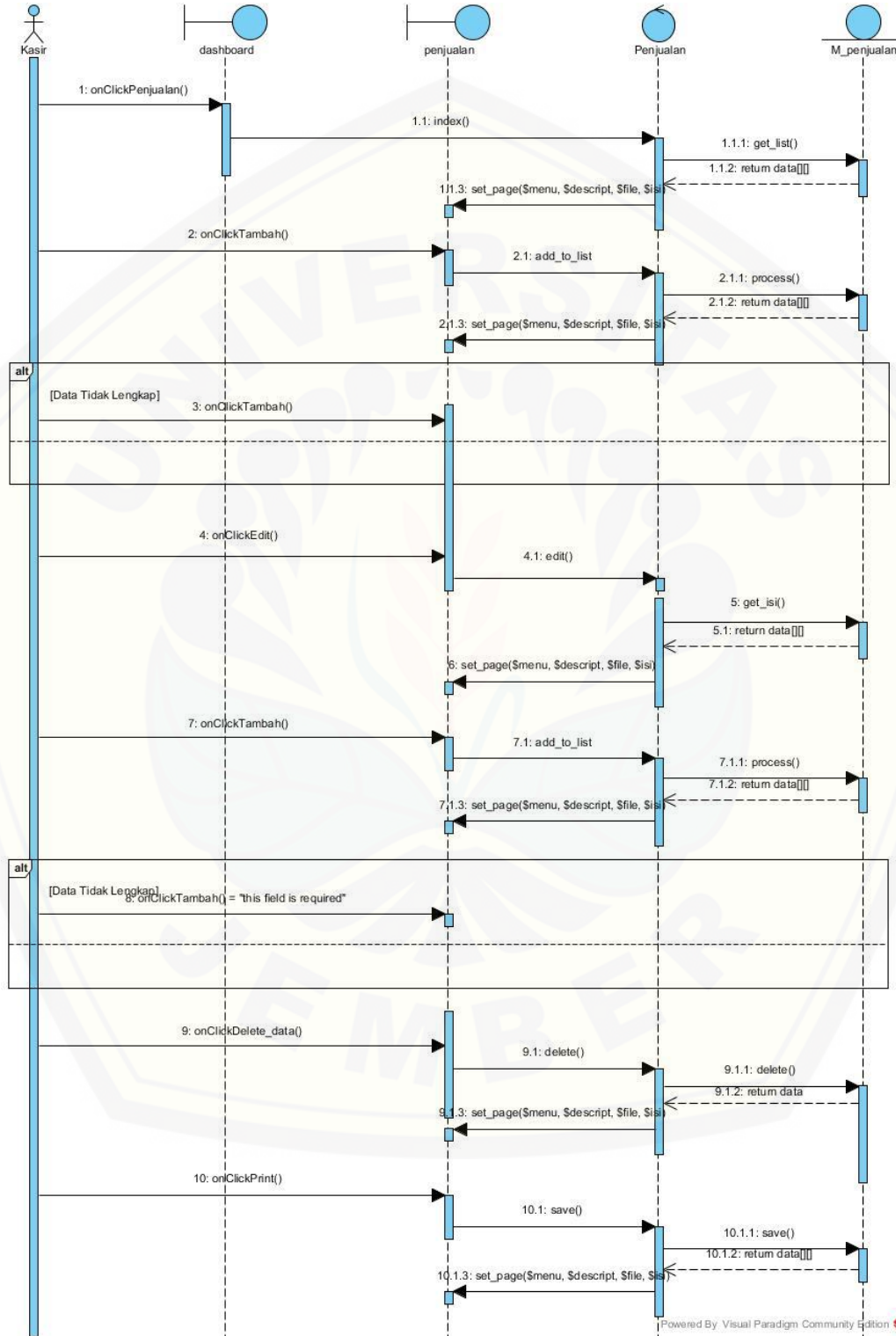
2.12 Sequence Diagram Mengelola Transfer Barang



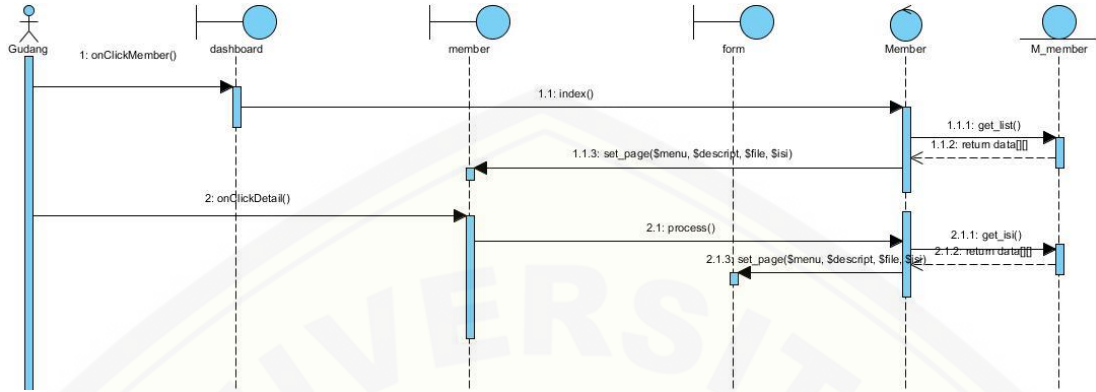
2.13 Sequence Diagram Mengelola Data Barang



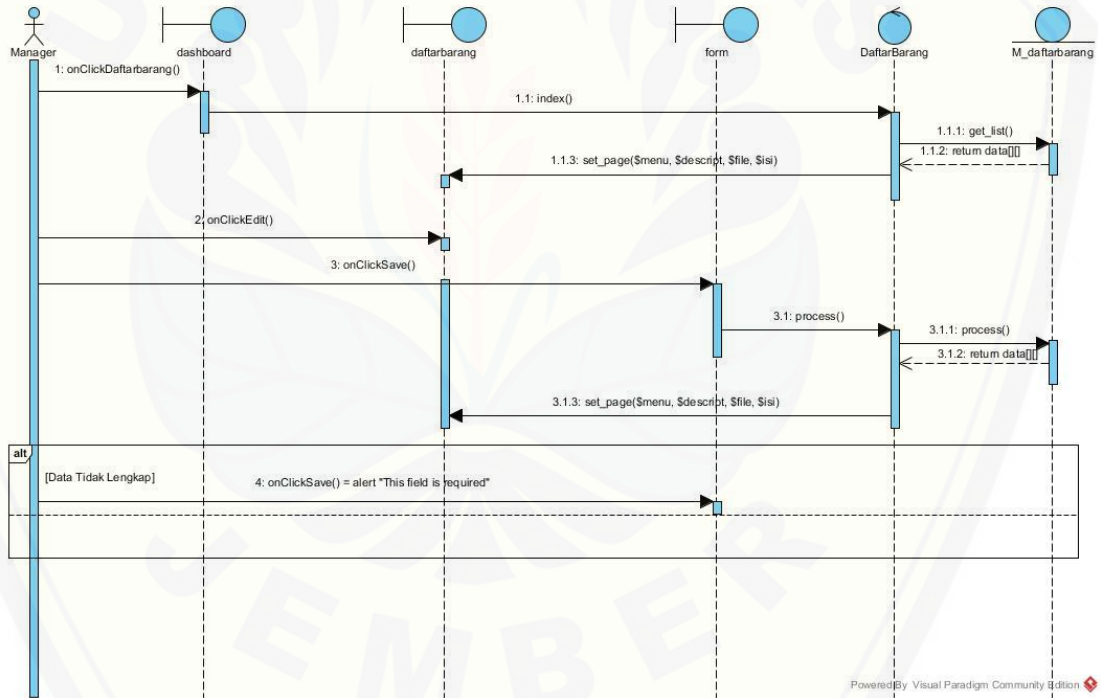
2.14 Sequence Diagram Mengelola Penjualan



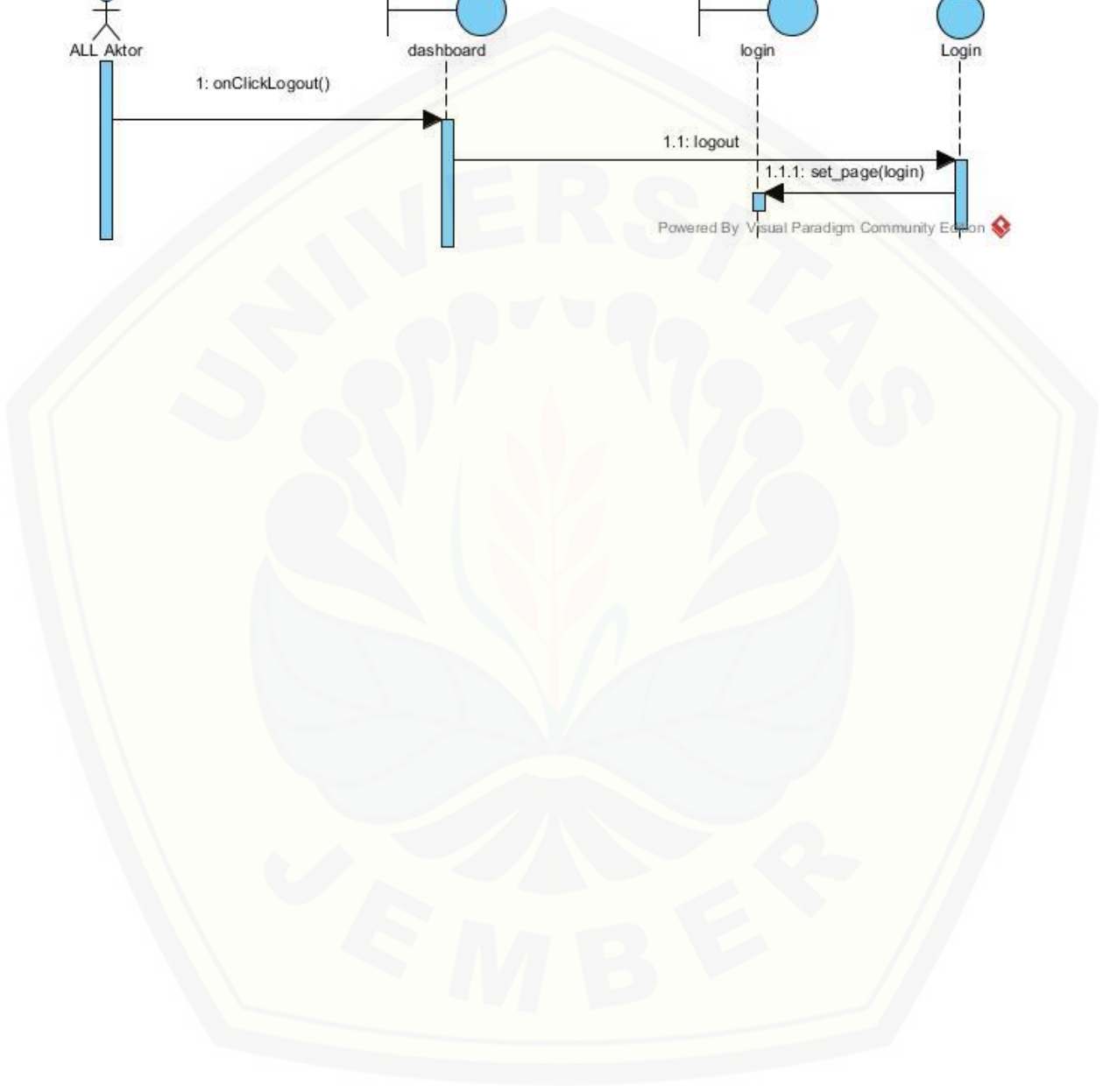
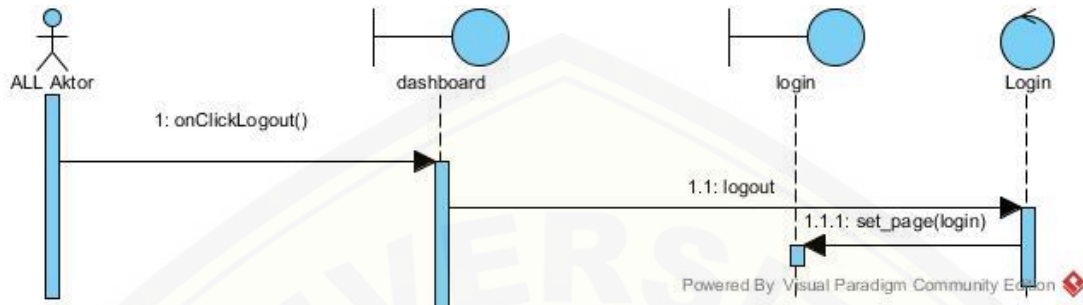
2.15 Sequence Diagram Melihat Data Member

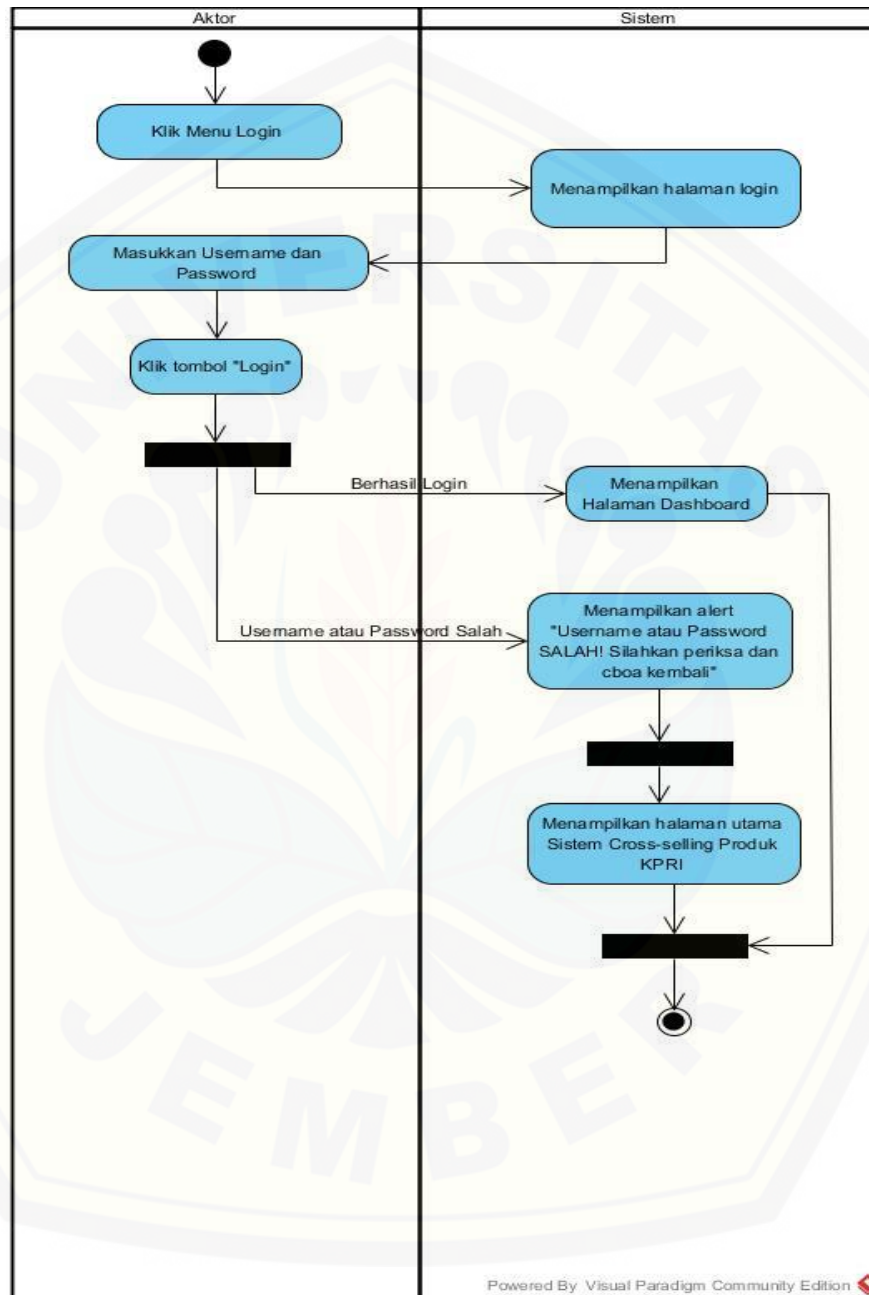


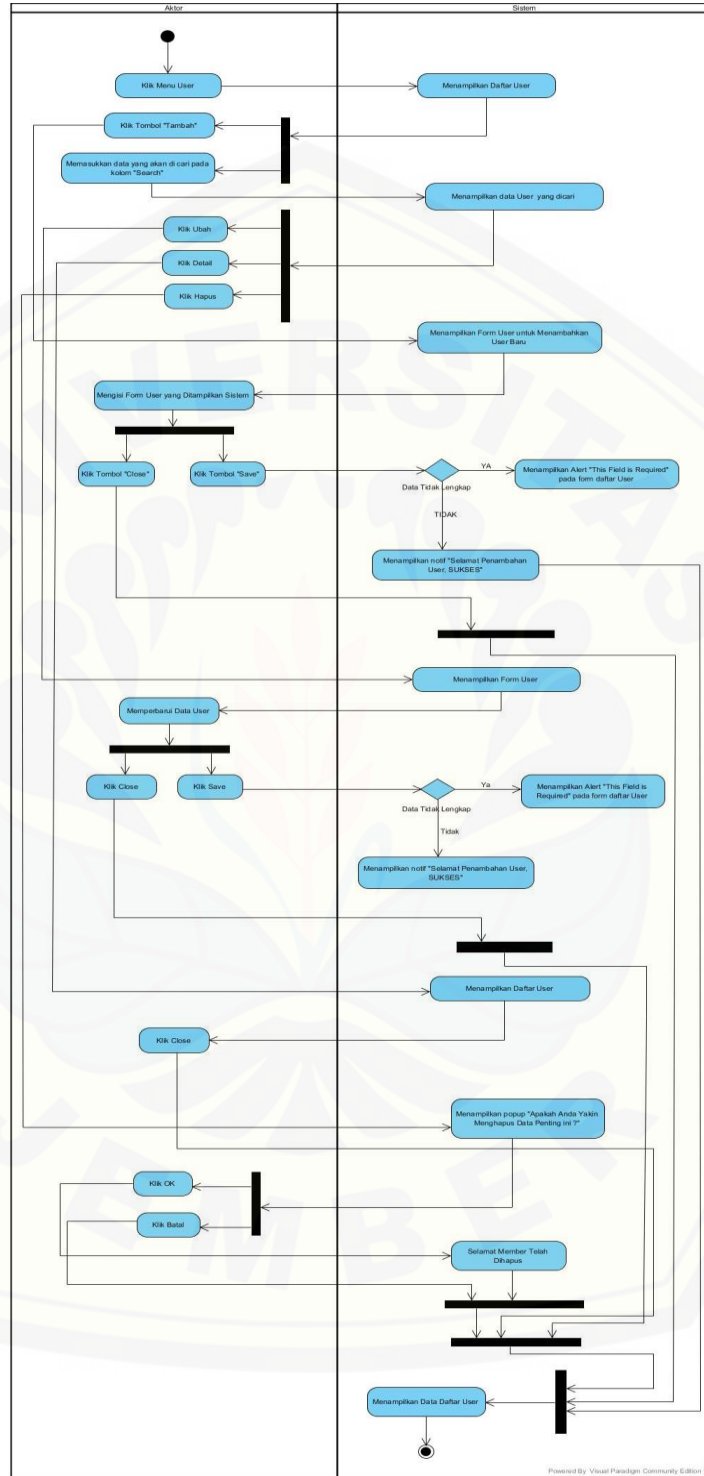
2.16 Sequence Diagram Mengubah Data Barang



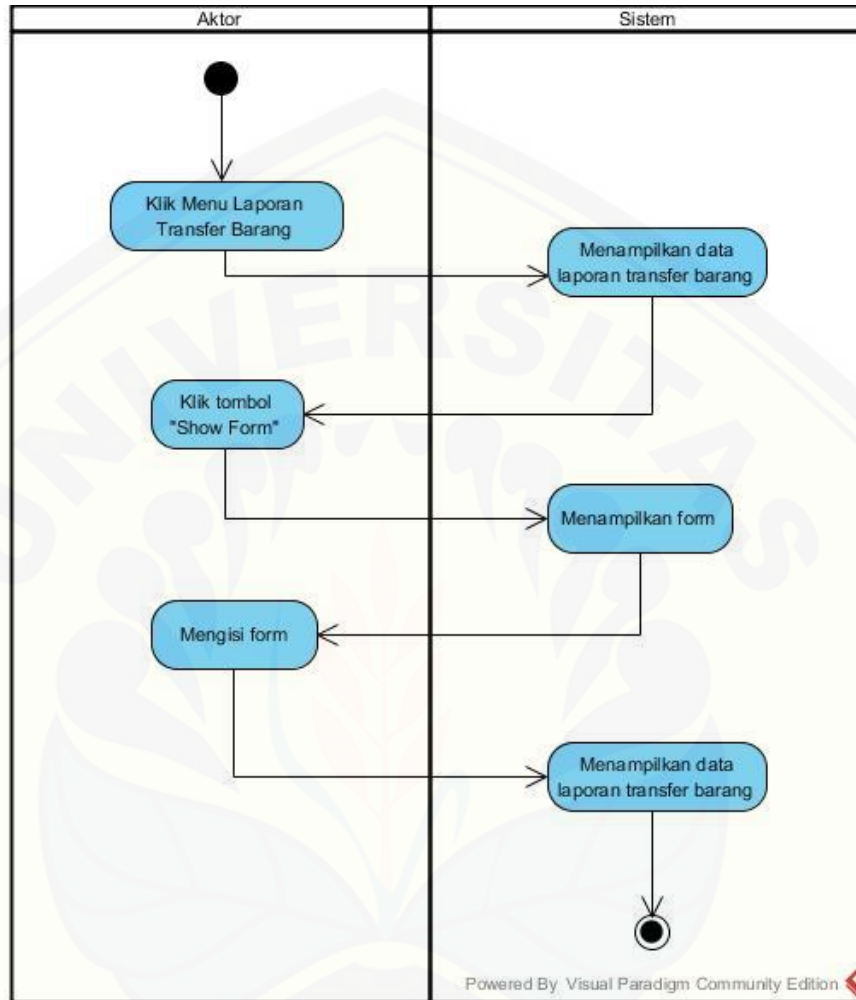
2.17 Sequence Diagram Mengelola Logout



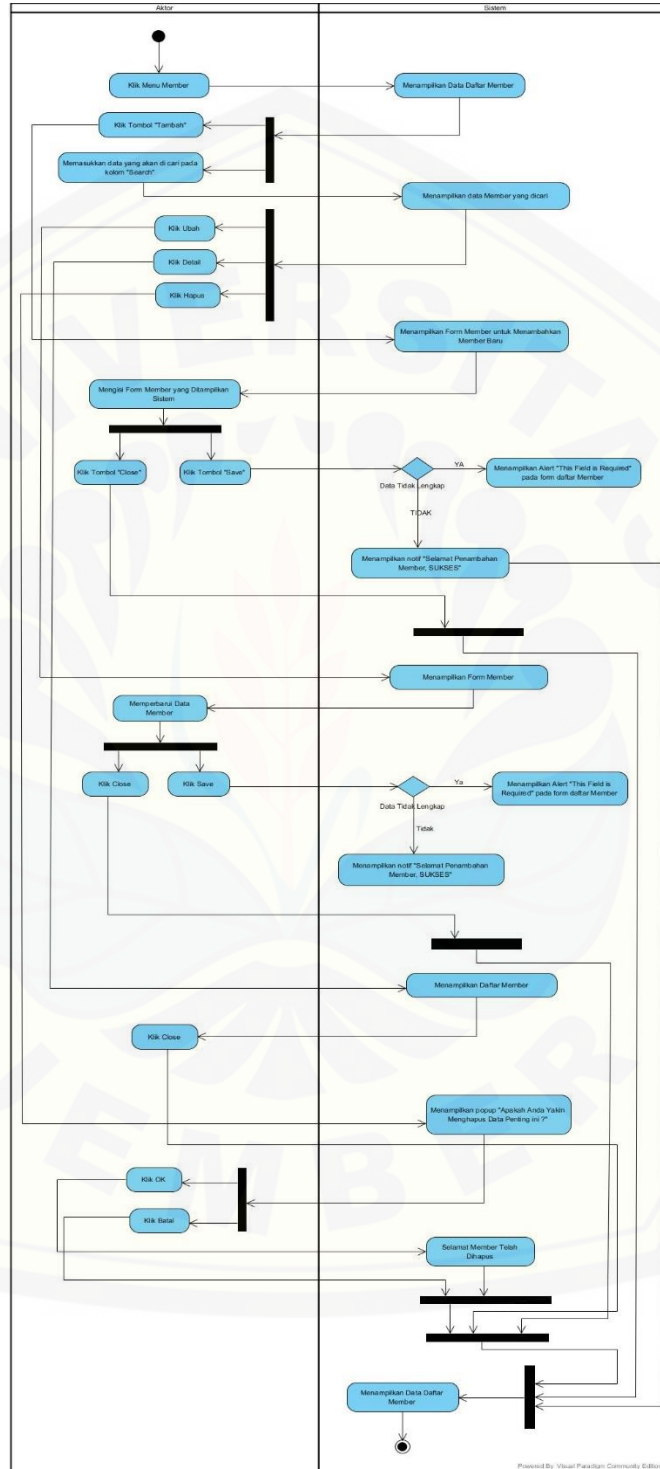
LAMPIRAN 3 (Activity Diagram)**3.1 Activity Diagram Mengelola Data Login****3.2 Activity Diagram Mengelola Data User**



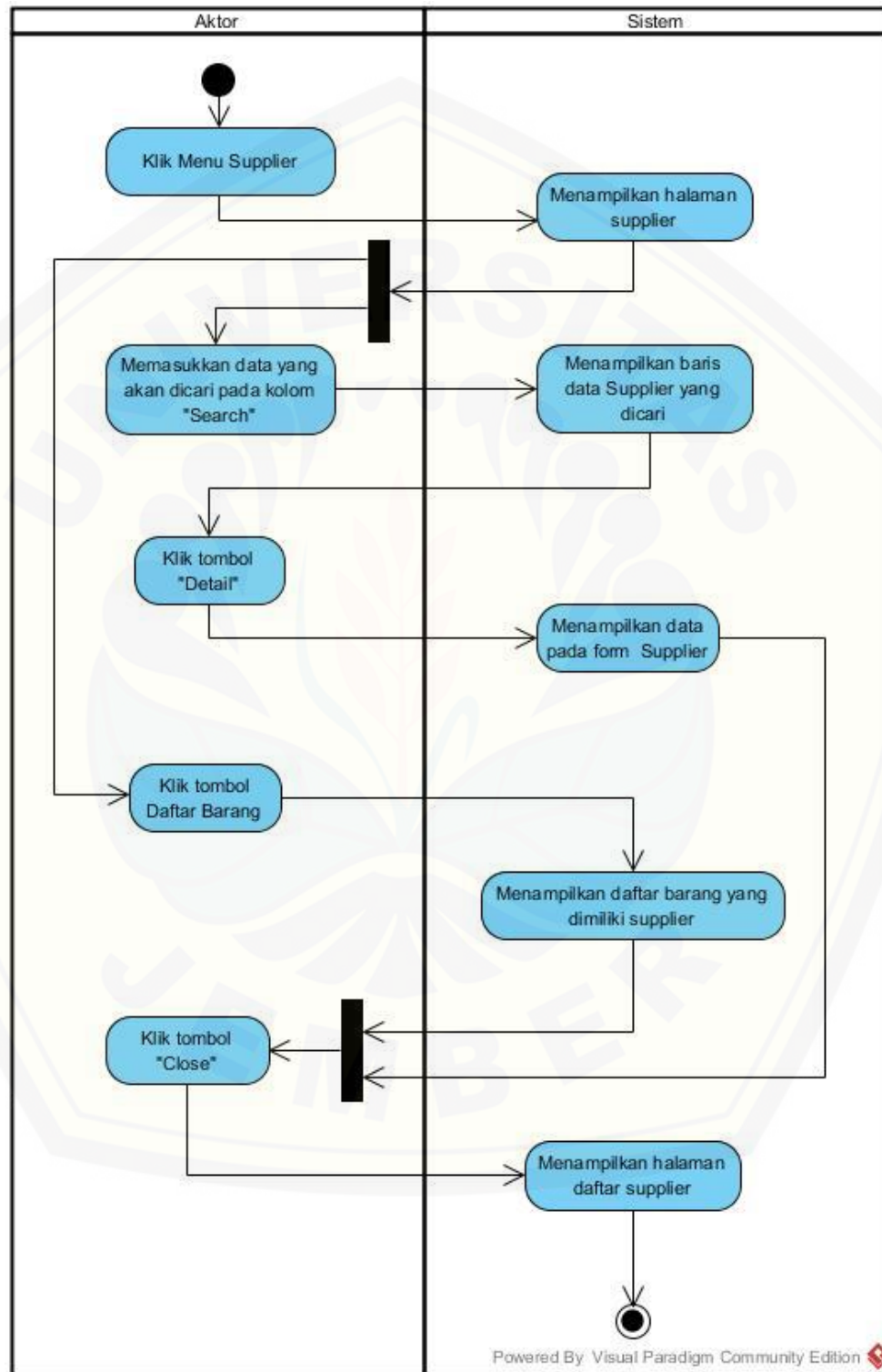
3.3 Activity Diagram Melihat Laporan Transfer Barang

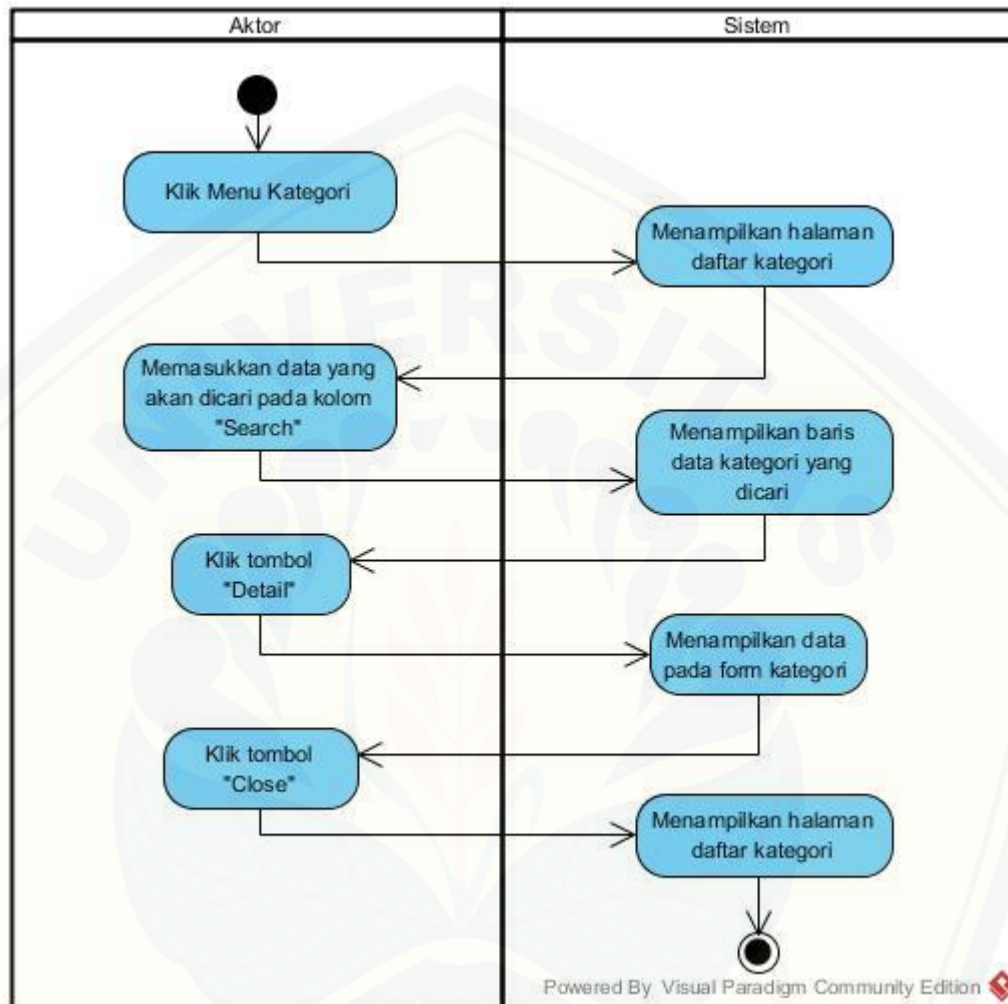


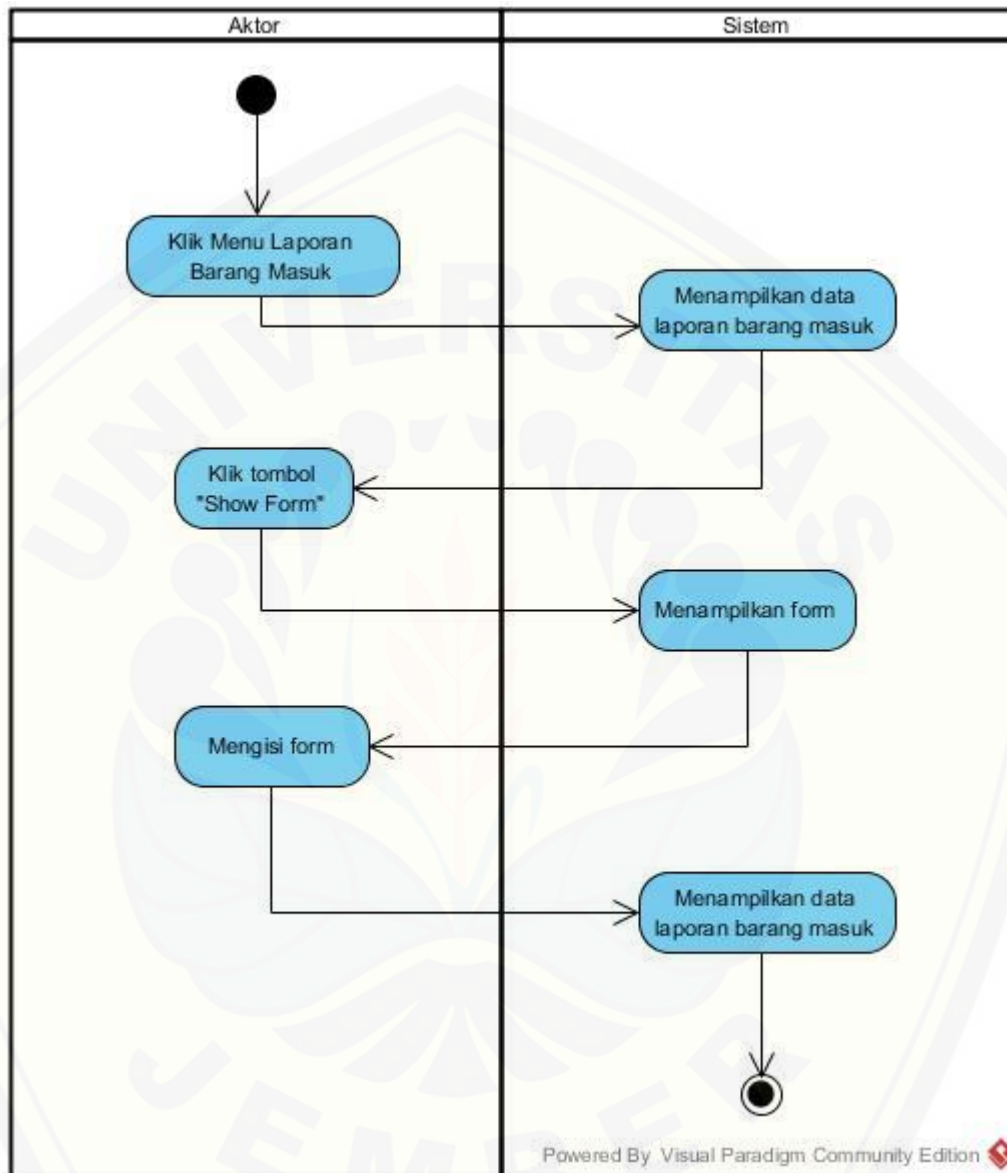
3.4 Activity Diagram Mengelola Data Member

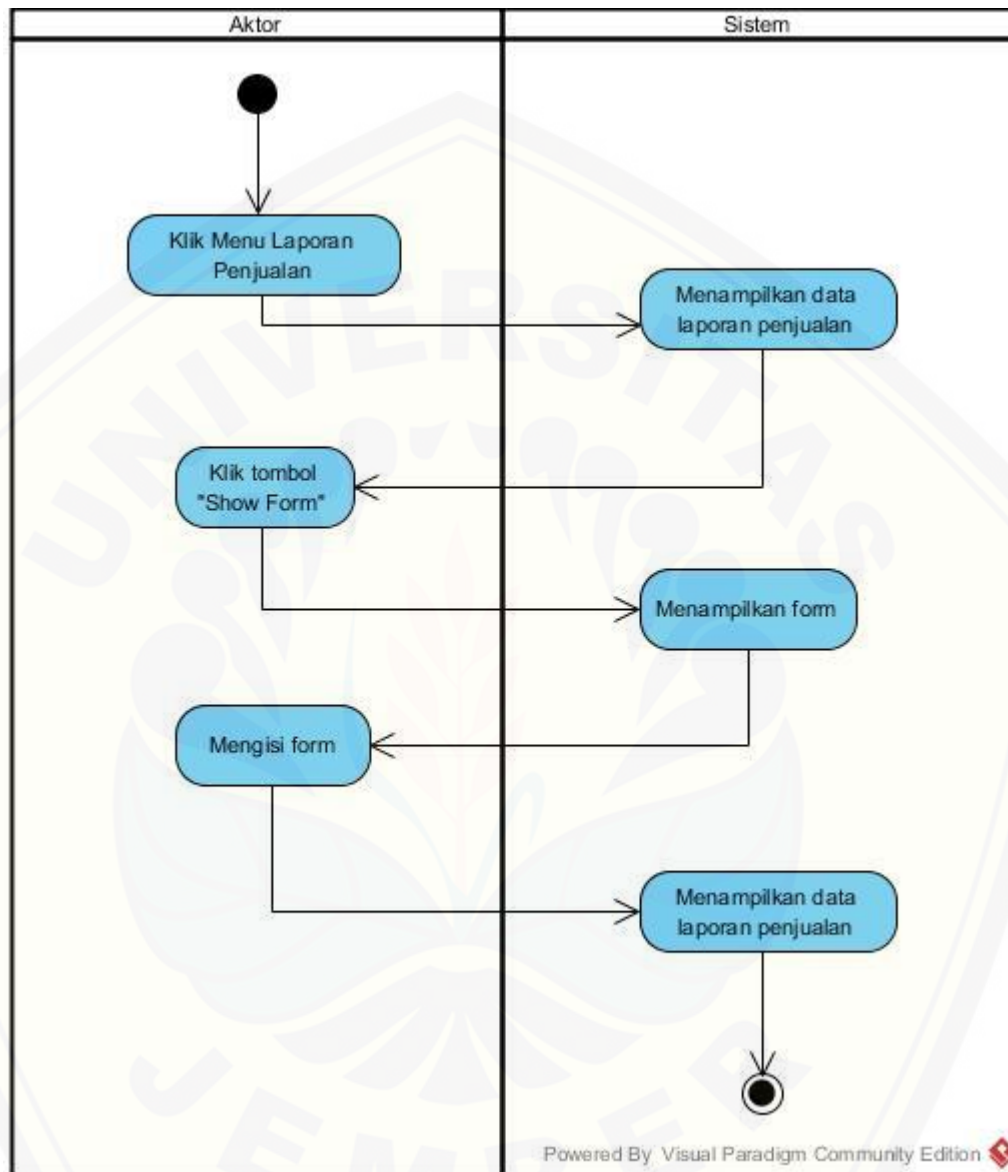


3.5 Activity Diagram Melihat Data Supplier

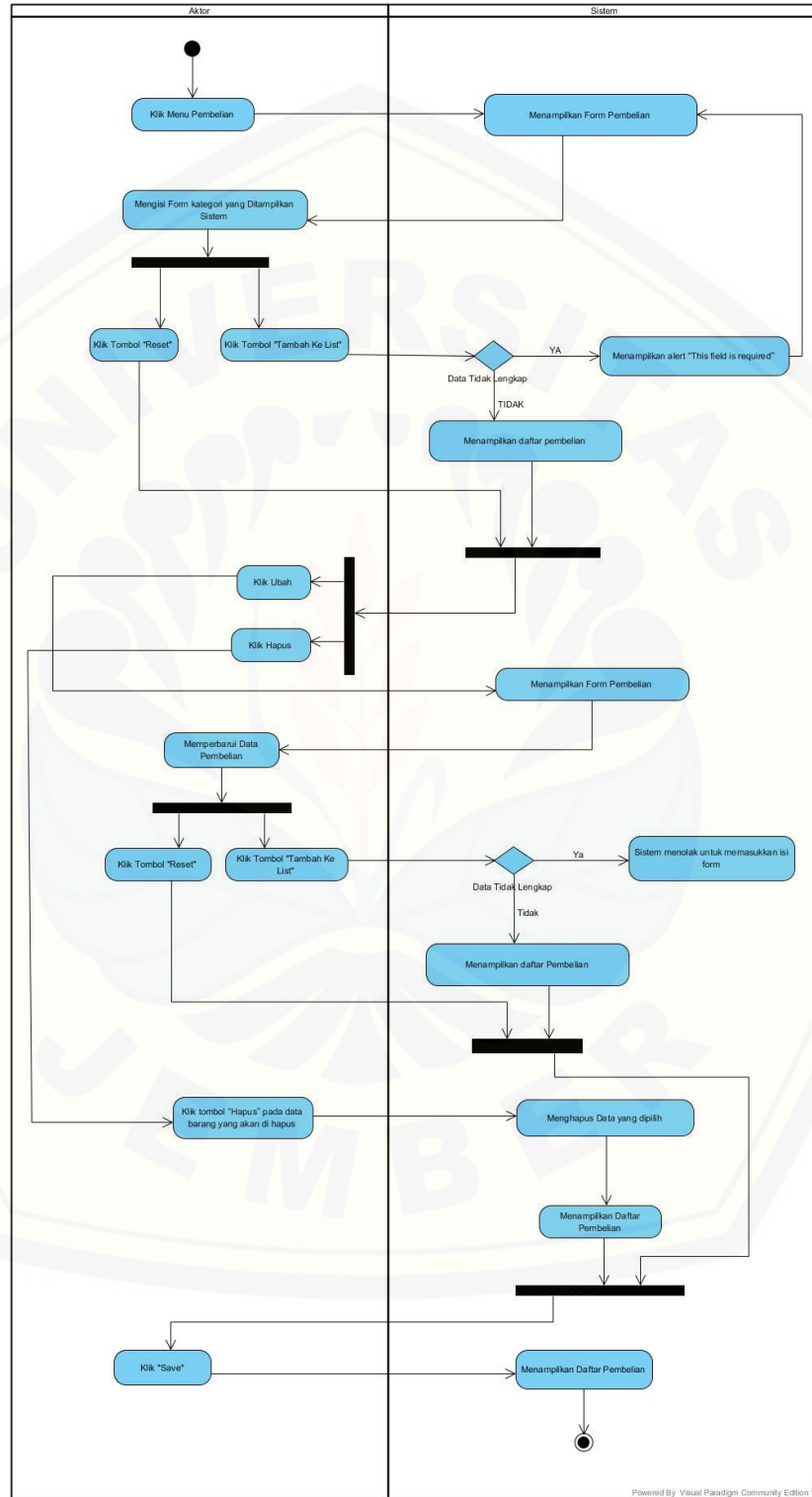


3.6 Activity Diagram Melihat Data Kategori Produk

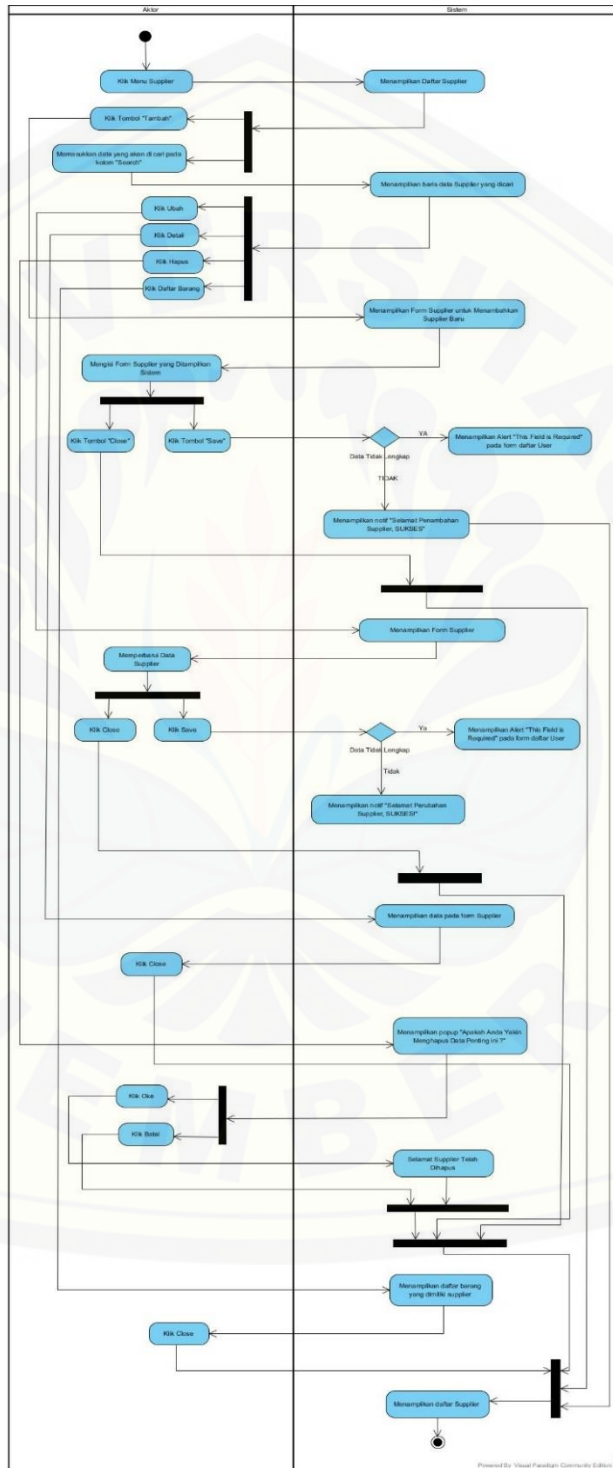
3.7 Activity Diagram Melihat Laporan Barang Masuk

3.8 Activity Diagram Melihat Laporan Penjualan

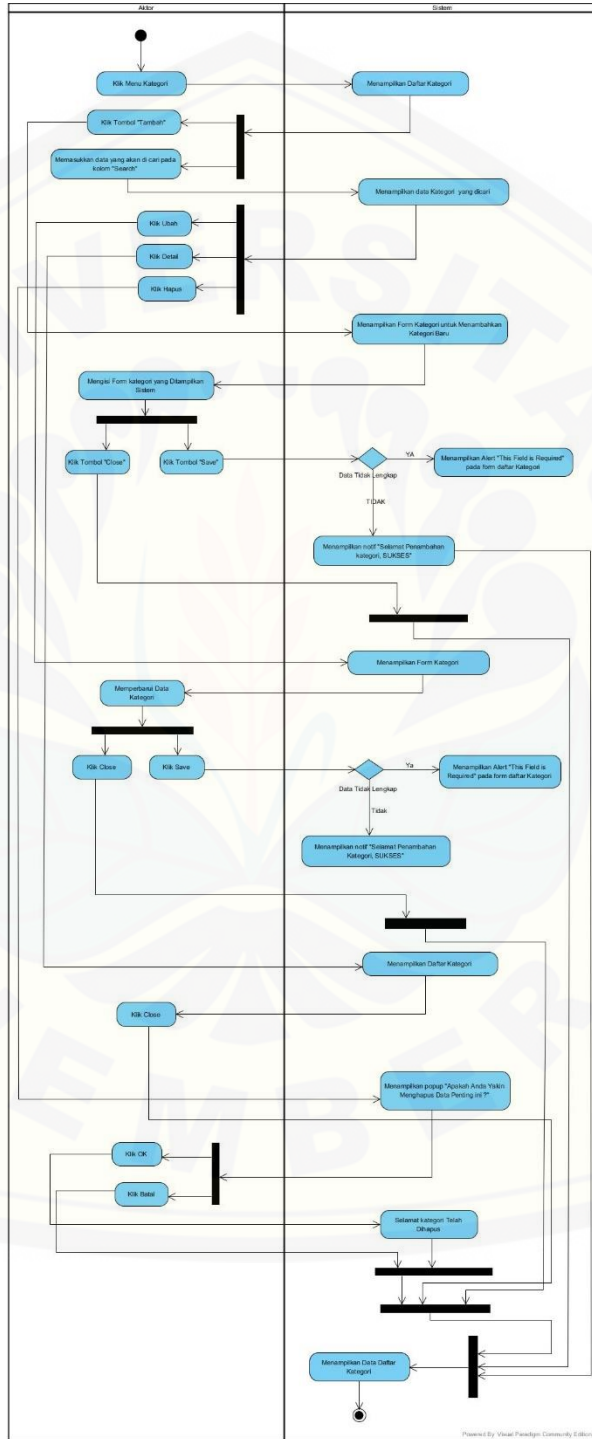
3.9 Activity Diagram Mengelola Pembelian Barang



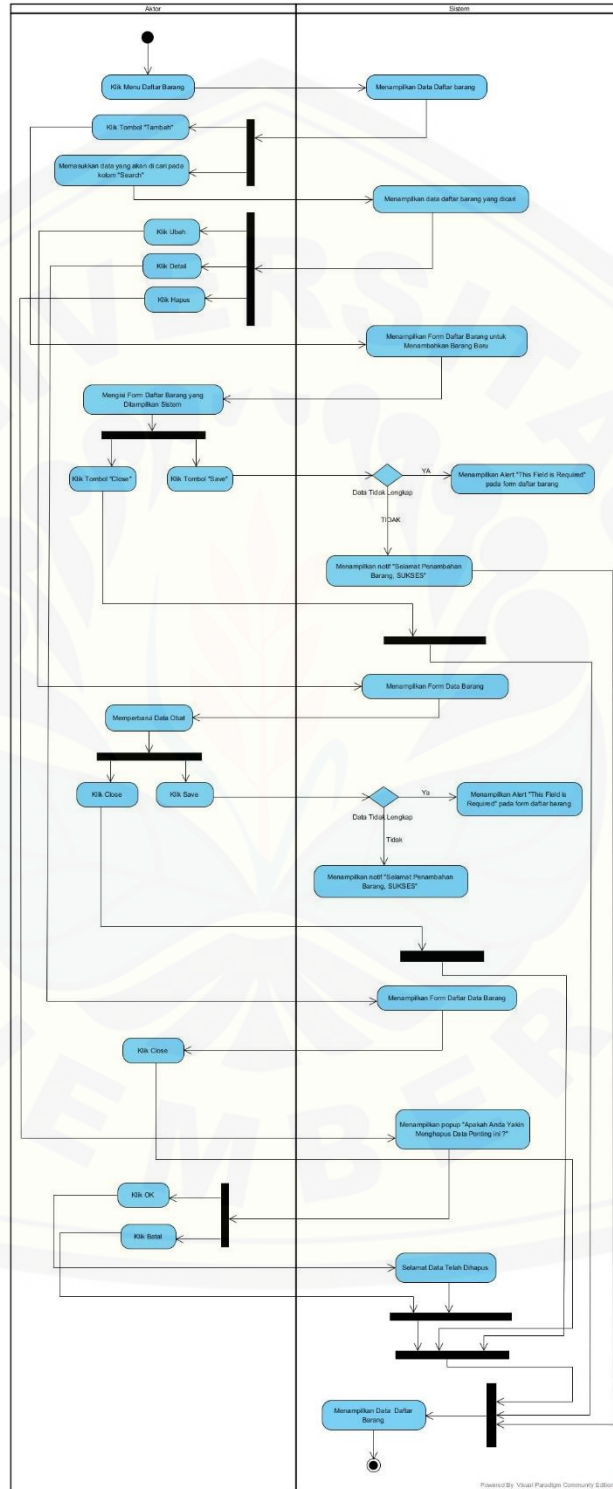
3.10 Activity Diagram Mengelola Data Supplier



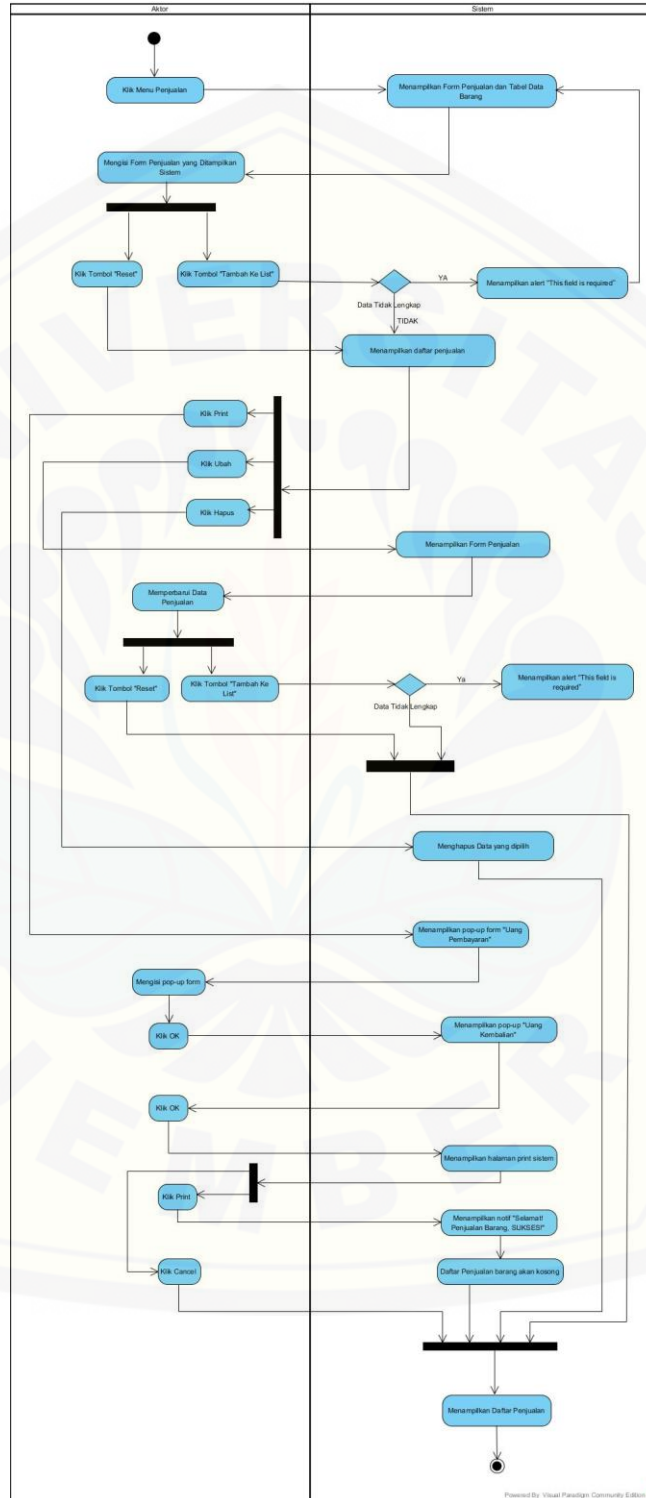
3.11 Activity Diagram Mengelola Data Kategori Produk



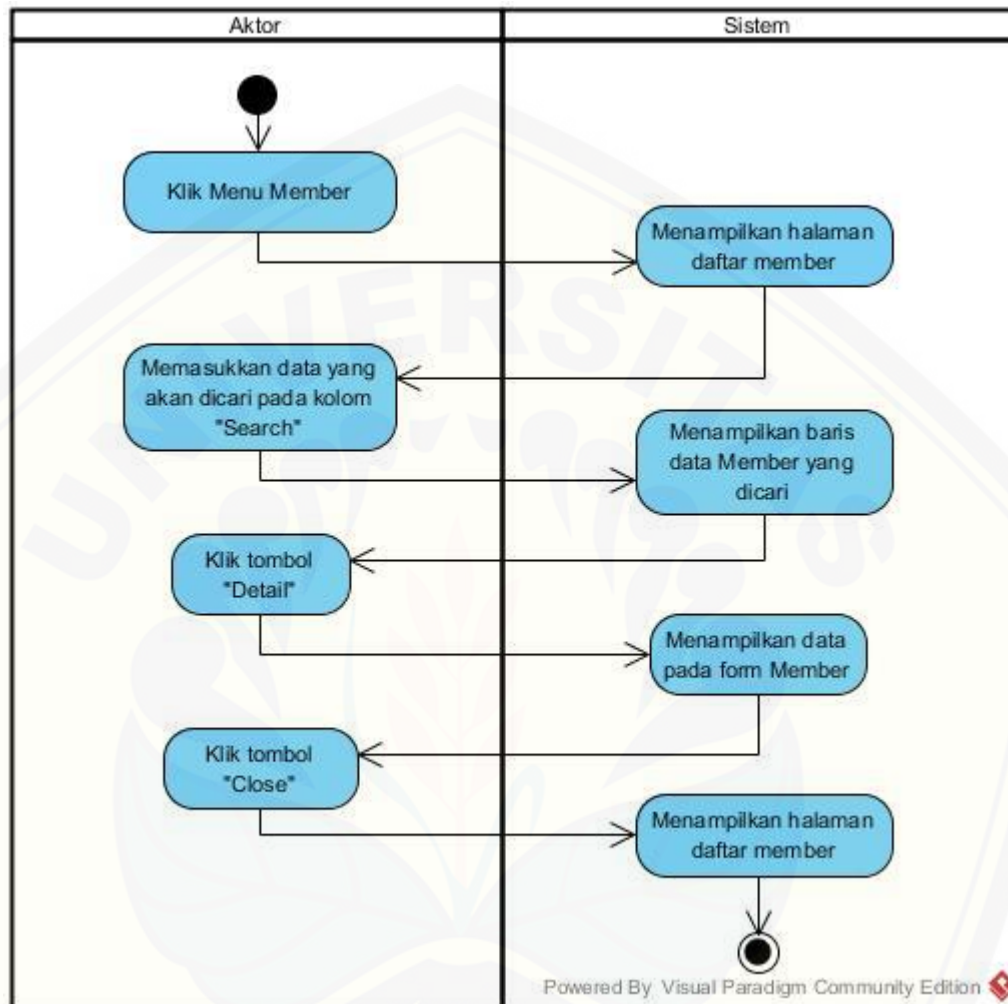
3.13 Activity Diagram Mengelola Data Barang



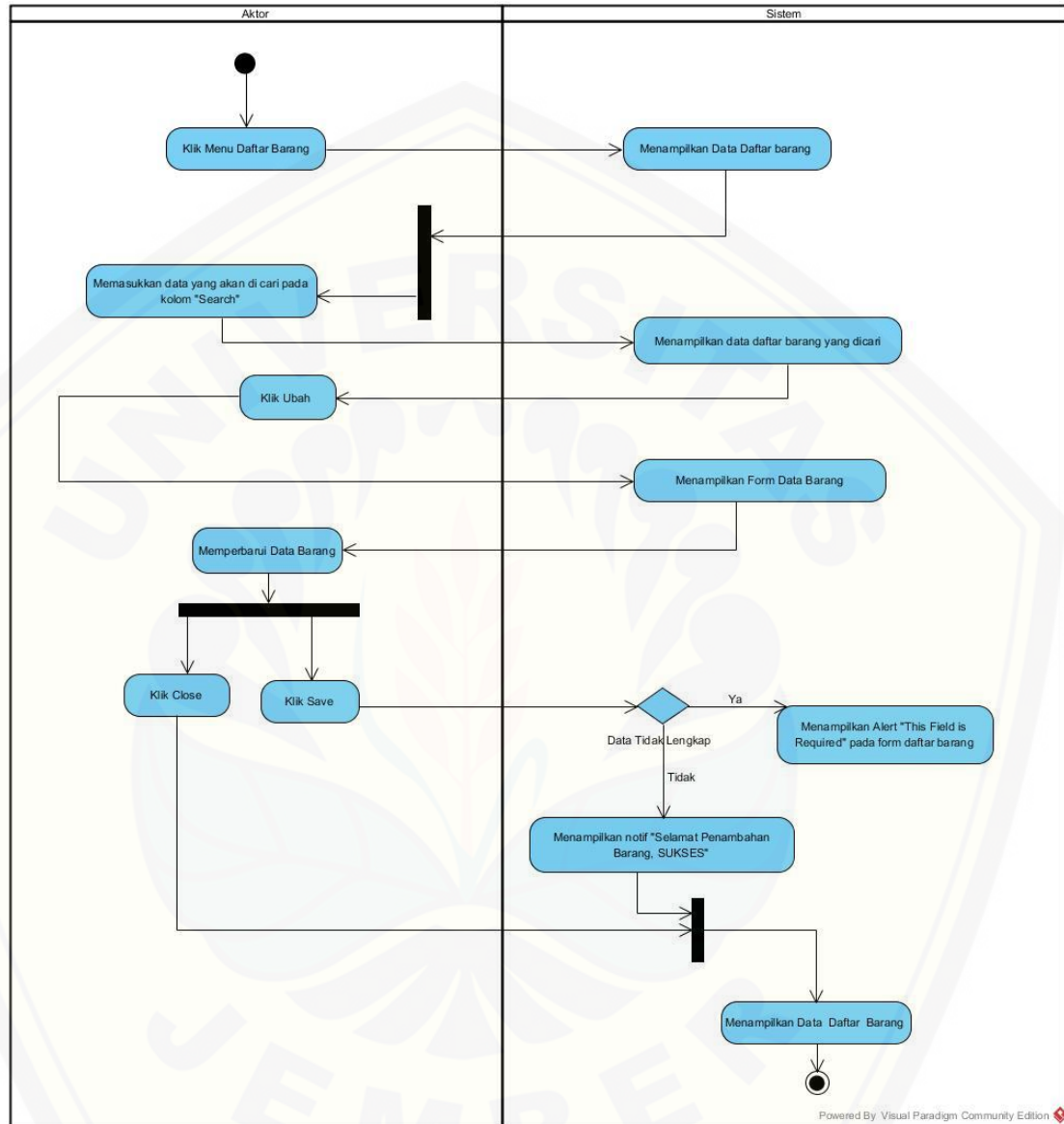
3.14 Activity Diagram Mengelola Penjualan

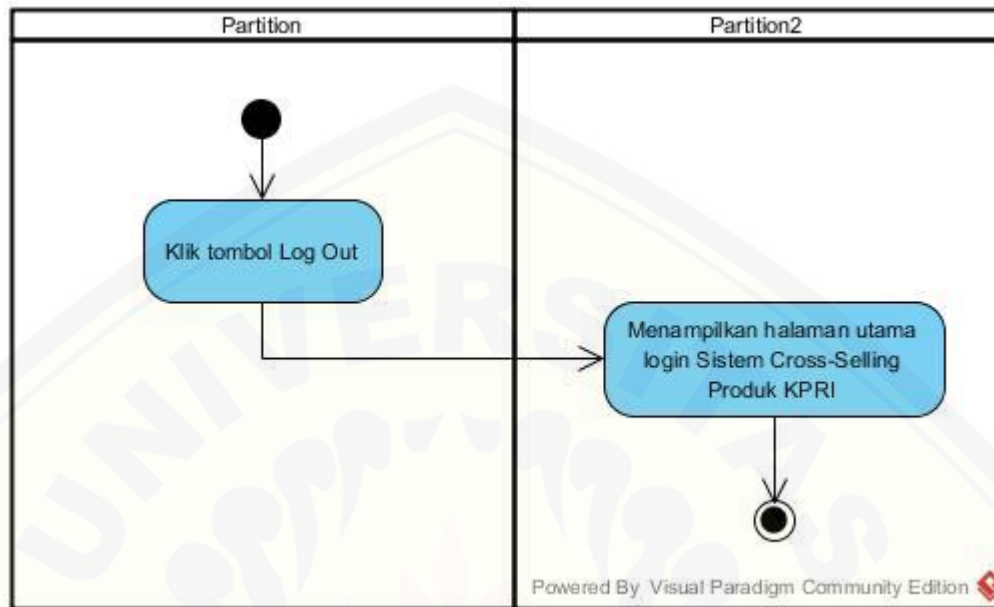


3.15 Activity Diagram Melihat Data Member



3.16 Activity Diagram Mengubah Data Barang



3.17 Activity Diagram Mengelola Logout

LAMPIRAN 4 (Black Box)

No	Menu	Fungsi	Hasil	Keterangan
1	Login	Mengisi <i>username</i> dan <i>password</i> klik tombol <i>login</i>	Login Berhasil dan menampilkan halaman dashboard	[√] Berhasil [] Gagal
		<i>Username</i> dan <i>password</i> salah, klik <i>login</i>	Menampilkan alert <i>username</i> dan <i>password</i> salah	[√] Berhasil [] Gagal
2	Menambah data user	Mengisi form data user lalu klik <i>save</i>	Menyimpan data user dan menampilkan pesan selamat penambahan user berhasil	[√] Berhasil [] Gagal
		<i>Text field</i> belum diisi, klik tombol <i>save</i>	Menampilkan pesan "Please fill out this field." dengan menunjuk pada <i>field</i> yang belum diisi	[√] Berhasil [] Gagal
3	Mengubah data user	Mengubah data user pada form klik <i>save</i>	Menyimpan data user yang dirubah dan menampilkan pesan perubahan user berhasil	[√] Berhasil [] Gagal
		<i>Text field</i> belum diisi, klik <i>save</i>	Menampilkan pesan "Please fill out this field." dengan menunjuk pada <i>field</i> yang belum diisi	[√] Berhasil [] Gagal
4	Menghapus Data User (Nonaktif)	Klik tombol <i>delete</i> pada user yang akan di hapus (dinonaktifkan)	Merubah status user menjadi 0 dan menampilkan halaman data user dengan pesan user telah dihapus	[√] Berhasil [] Gagal
5	Menampilkan detail data user	Klik tombol <i>detail</i> pada data user yang ingin ditampilkan	Menampilkan halaman detail user berupa tabel	[√] Berhasil [] Gagal

6	Menampilkan Laporan Transfer Barang	Klik menu LaporanTransfer Barang	Menampilkan laporan transfer barang pada hari ini	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
7	Menampilkan Laporan Transfer Barang sesuai tanggal yang dipilih	Klik Show Form,lalu pilih form tanggal dan isi	Menampilkan laporan transfer barang pada tanggal yang dipilih	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
8	Tambah Cross-selling	Mengisi <i>form</i> tambah cross-selling, klik save	Menyimpan databaru dan menampilkan halaman cross-selling dengan pesan penambahan cross-selling berhasil	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		<i>Text field</i> belum diisi, klik tombol save	Menampilkan pesan "Please fill out this field." dengan menunjuk pada <i>field</i> yang belum diisi	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
9	Mengubah Cross-selling	Mengubah data Cross-selling yang dipilih klik save	Mengubah data Cross-selling pada database dan menampilkan halaman Cross-selling dengan pesan perubahan Cross-selling berhasil	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		<i>Text field</i> belum diisi, klik tombol submit	Menampilkan pesan "Please fill out this field." dengan menunjuk pada <i>field</i> yang belum diisi	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
10	Menghapus Cross-selling	Klik tombol delete pada Cross-selling yang akan dihapus	Menghapus data didatabase dan menampilkan halaman Cross-selling dengan pesan Cross-selling telah dihapus	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

11	Menampilkan hasil Cross-selling	Klik Detail Hasil pada Cross-selling yang ingin ditampilkan	Menampilkan hasil pengaturan Cross-selling untuk cross-selling yang dipilih	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
12	Menampilkan detail cross-selling	Klik tombol <i>Detail</i> pada Cross-selling yang ingin dilihat	Menampilkan halaman detail Cross-selling yang dipilih	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
13	Menampilkan daftar member pada halaman member	Klik menu member	Menampilkan halaman member yang berisi daftar member yang suda tersimpan didatabase	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
14	Tambah Member	Mengisi <i>form</i> tambah member, klik save	Menyimpan databaru dan menampilkan halaman member dengan pesan penambahan member berhasil	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		<i>Text field</i> belum diisi, klik tombol save	Menampilkan pesan “Please fill out this field.” dengan menunjuk pada <i>field</i> yang belum diisi	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
15	Mengubah Member	Mengubah data Member yang dipilih klik save	Mengubah data Member pada database dan menampilkan halaman Member dengan pesan perubahan Member berhasil	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		<i>Text field</i> belum diisi, klik tombol save	Menampilkan pesan “Please fill out this field.” dengan menunjuk pada <i>field</i> yang belum diisi	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
16	Menghapus Member	Klik tombol delete pada Member yang akan dihapus	Menghapus data didatabase dan menampilkan halaman Member dengan pesan Member telah dihapus	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

17	Menampilkan detail Member	Klik tombol <i>Detail</i> pada Member yang ingin dilihat	Menampilkan halaman detail Member yang dipilih	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
18	Menampilkan Supplier	Klik menu supplier	Menampilkan daftar supplier yang sudah tersimpan di database berupa tabel	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
19	Mencari data Supplier	Isi form pada <i>field</i> search	Menampilkan daftar supplier yang dicari	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
20	Menampilkan Daftar Barang	Klik tombol Daftar Barang pada supplier yang dipilih	Menampilkan daftar barang yang dimiliki supplier dan sudah tersimpan di database berupa tabel	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
21	Menampilkan Kategori Produk	Klik menu Kategori	Menampilkan daftar kategori yang sudah tersimpan di database berupa tabel	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
22	Mencari data Kategori Produk	Isi form pada <i>field</i> search	Menampilkan daftar kategori yang dicari	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
23	Menampilkan Laporan Barang Masuk	Klik menu Laporan Barang Masuk	Menampilkan halaman laporan barang masuk pada hari ini	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
24	Menampilkan data laporan barang masuk pada tanggal tertentu	Klik tombol show form pada halaman menu laporan barang masuk dan pilih tanggal	Menampilkan halaman laporan barang masuk dengan data barang masuk sesuai tanggal yang dipilih	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

25	Menampilkan Laporan Penjualan	Klik menu Laporan Penjualan	Menampilkan halaman laporan penjualan pada hari ini	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
26	Menampilkan data laporan penjualan pada tanggal tertentu	Klik tombol show form pada halaman menu laporan penjualan dan pilih tanggal	Menampilkan halaman laporan penjualan dengan data penjualan sesuai tanggal yang dipilih	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
27	Tambah Pembelian	Mengisi <i>form</i> tambah pembelian, klik Tambah Ke List	Menyimpan databaru dan menampilkan data barang yang dibeli pada tabel daftar pembelian	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		<i>Text field</i> belum diisi, klik tombol Tambah Ke List	Menampilkan pesan "Please fill out this <i>field</i> ." dengan menunjuk pada field yang belum diisi	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
28	Mengubah Pembelian barang	Mengubah data barang yang dipilih klik Tambah Ke List	Mengubah data pembelian barang pada database dan menampilkan data barang yang dibeli pada tabel daftar pembelian	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
29	Menghapus barang yang dibeli	Klik tombol delete pada barang yang dibeli dan akan dihapus	Menghapus data didatabase dan menampilkan halaman pembelian barang	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
30	Menyimpan data pembelian barang	Klik tombol save	Menyimpan data pembelian di database dan Menampilkan halaman pembelian barang	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
31	Tambah Supplier	Mengisi <i>form</i> tambah supplier, klik save	Menyimpan databaru dan menampilkan data supplier	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

		<i>Text field</i> belum diisi, klik tombol save	Menampilkan pesan “Please fill out this field.” dengan menunjuk pada <i>field</i> yang belum diisi	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
32	Mengubah supplier	Mengubah data supplier yang dipilih klik save	Mengubah data supplier pada database dan menampilkan data supplier	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
33	Menghapus supplier	Klik tombol delete pada supplier yang akan dihapus	Menghapus data didatabase dan menampilkan halaman supplier	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
34	Menampilkan detail supplier	Klik tombol <i>Detail</i> pada supplier yang ingin dilihat	Menampilkan halaman detail supplier yang dipilih	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
35	Tambah Kategori Produk	Mengisi <i>form</i> tambah kategori, klik save	Menyimpan databaru dan menampilkan data kategori	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		<i>Text field</i> belum diisi, klik tombol save	Menampilkan pesan “Please fill out this field.” dengan menunjuk pada <i>field</i> yang belum diisi	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
36	Mengubah kategori	Mengubah data kategori yang dipilih klik save	Mengubah data kategori pada database dan menampilkan data kategori	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
37	Menghapus kategori	Klik tombol delete pada kategori yang akan dihapus	Menghapus data didatabase dan menampilkan halaman kategori	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
38	Menampilkan detail kategori	Klik tombol <i>Detail</i> pada kategori yang ingin dilihat	Menampilkan halaman detail kategori yang dipilih	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

39	Tambah Transfer Barang	Mengisi <i>form</i> tambah transfer barang, klik Tambah Ke List	Menyimpan databaru dan menampilkan data barang yang ditransfer pada tabel daftar transfer barang	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		<i>Text field</i> belum diisi, klik tombol Tambah Ke List	Menampilkan pesan “Please fill out this field.” dengan menunjuk pada <i>field</i> yang belum diisi	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
40	Mengubah Transfer Barang	Mengubah data barang yang dipilih klik Tambah Ke List	Mengubah data transfer barang pada database dan menampilkan data barang yang ditransfer pada tabel daftar transfer barang	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
41	Menghapus barang yang akan ditransfer	Klik tombol delete pada barang yang dibeli dan akan dihapus	Menghapus data didatabase dan menampilkan halaman transfer barang	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
42	Menyimpan data transfer barang	Klik tombol save	Menyimpan data transfer di database dan Menampilkan halaman transfer barang	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
43	Tambah Data Barang	Mengisi <i>form</i> tambah barang, klik save	Menyimpan databaru dan menampilkan data barang	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		<i>Text field</i> belum diisi, klik tombol save	Menampilkan pesan “Please fill out this field.” dengan menunjuk pada <i>field</i> yang belum diisi	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
44	Mengubah Data Barang	Mengubah data barang yang dipilih klik save	Mengubah data barang pada database dan menampilkan data barang	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
45	Menghapus Data Barang	Klik tombol delete pada barang yang akan dihapus	Menghapus data didatabase dan menampilkan halaman barang	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

46	Menampilkan detail barang	Klik tombol <i>Detail</i> pada barang yang ingin dilihat	Menampilkan halaman detail barang yang dipilih	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
47	Menampilkan cross-selling di penjualan	Klik tombol Tambah Ke List	Menampilkan alertt cross-selling produk sesuai nama produk	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
48	Tambah Penjualan Barang	Mengisi <i>form</i> tambah penjualan barang, klik Tambah Ke List	Menyimpan databaru dan menampilkan data barang yang dibeli pada tabel daftar transfer barang	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		<i>Text field</i> belum diisi, klik tombol Tambah Ke List	Menampilkan pesan “Please fill out this field.” dengan menunjuk pada <i>field</i> yang belum diisi	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
49	Mengubah Penjualan Barang	Mengubah data penjualan barang yang dipilih klik Tambah Ke List	Mengubah data penjualan barang pada database dan menampilkan data barang yang dibeli pada tabel daftar penjualan barang	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
50	Menghapus barang yang akan dijual	Klik tombol delete pada barang yang dijual dan akan dihapus	Menghapus data didatabase dan menampilkan halaman penjualan barang	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
51	Menyimpan dan mencetak data penjualan barang	Klik tombol print	Menyimpan data penjualan di database dan Menampilkan halaman print penjualan barang	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
52	Menampilkan Data Member	Klik menu Member	Menampilkan daftar Member yang sudah tersimpan di database berupa tabel	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

53	Mencari Data Member	Isi form pada <i>field</i> search	Menampilkan daftar Data Member yang dicari	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
54	Logout	Klik tombol sign out	Menampilkan halaman login	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

