



**PENGEMBANGAN MODUL *BUNDLING* MENGGUNAKAN *ASSOCIATION
RULE* ALGORITMA APRIORI PADA SISTEM INFORMASI PENJUALAN
ADIEN MART SEBAGAI STRATEGI PROMOSI PENJUALAN BARANG
KURANG LARIS**

SKRIPSI

Oleh

Eky Angliara

NIM 122410101041

PROGRAM STUDI SISTEM INFORMASI

UNIVERSITAS JEMBER

2017



**PENGEMBANGAN MODUL *BUNDLING* MENGGUNAKAN *ASSOCIATION
RULE* ALGORITMA APRIORI PADA SISTEM INFORMASI PENJUALAN
ADIEN MART SEBAGAI STRATEGI PROMOSI PENJUALAN BARANG
KURANG LARIS**

SKRIPSI

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat
untuk menyelesaikan Program Studi Sistem Informasi (SI)
dan mencapai gelar Sarjana Komputer

Oleh

Eky Angliara

NIM 122410101041

**PROGRAM STUDI SISTEM INFORMASI
UNIVERSITAS JEMBER**

2017

PERSEMBAHAN

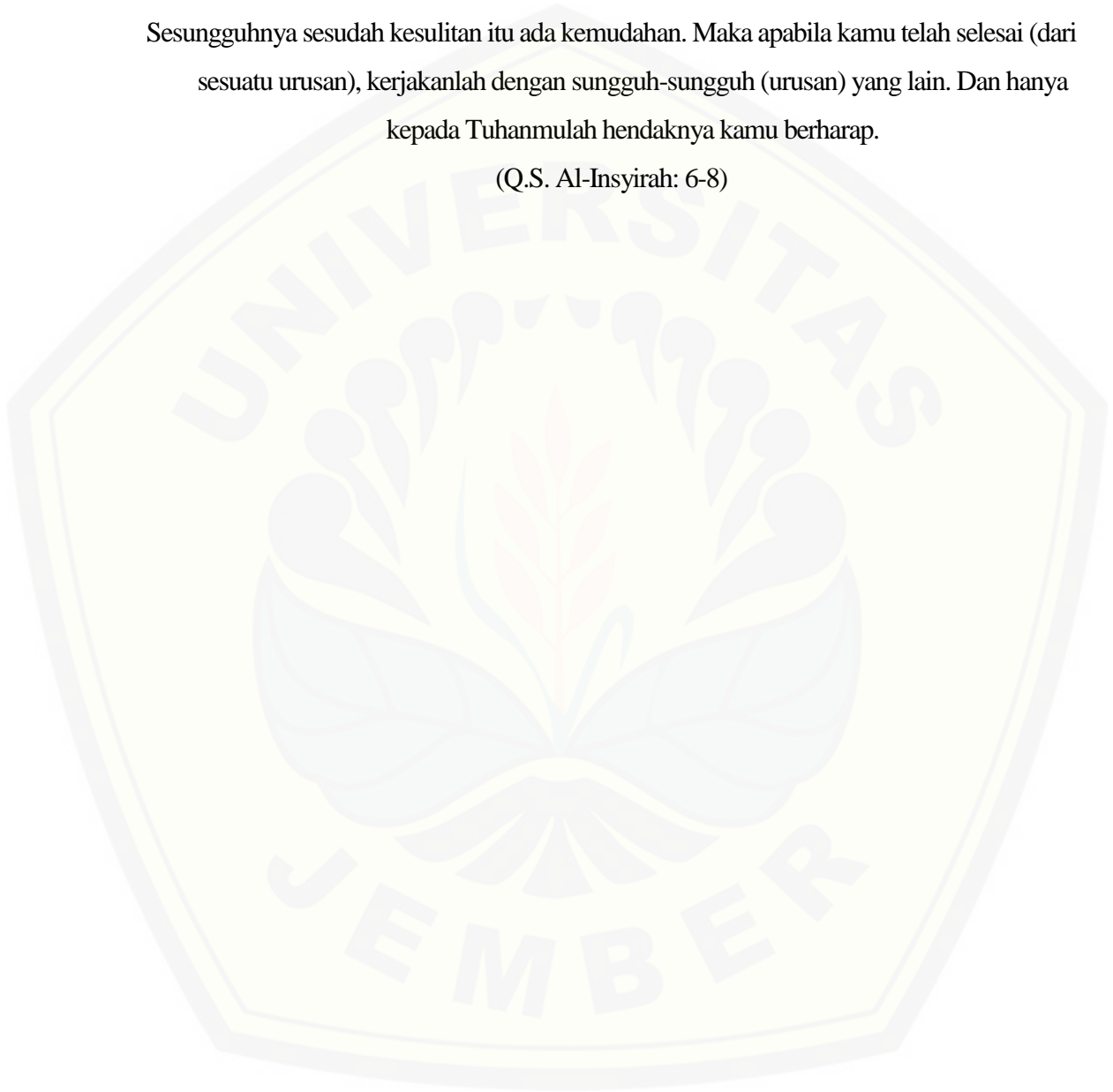
Skripsi ini saya persembahkan untuk :

1. Allah SWT yang senantiasa memberikan rahmat dan hidayah-Nya untuk mempermudah dan melancarkan dalam mengerjakan skripsi;
2. Ibunda Lilis Tri Astuti dan Ayahanda Heru Gatot Subegti tersayang yang selalu memberikan do'a, kasih sayang dan motivasi yang luar biasa;
3. Kakak Ganda Satya Bramara dan Adik Dinda Putri Astuti tercinta yang selalu memberi dukungan;
4. Seluruh keluarga besar yang selalu mendukung dan menghibur;
5. Usman, Yoko, Ainul, Mada, Marceli, Fariz serta sahabat kontrakan Puri Bunga Nirwana yaitu Lintang, Puput, Sefty, Ghausilia, Dwi, Jeje, Esa, Delia serta sahabat seperjuangan Haris yang selalu meluangkan waktu untuk berdiskusi;
6. Guru-guruku sejak taman kanak-kanak hingga perguruan tinggi, baik formal maupun informal;
7. Almamater Program Studi Sistem Informasi Universitas Jember.

MOTTO

Sesungguhnya sesudah kesulitan itu ada kemudahan. Maka apabila kamu telah selesai (dari sesuatu urusan), kerjakanlah dengan sungguh-sungguh (urusan) yang lain. Dan hanya kepada Tuhanmulah hendaknya kamu berharap.

(Q.S. Al-Insyirah: 6-8)



PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Eky Angliara

NIM : 122410101041

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Pengembangan Modul *Bundling* Menggunakan *Association Rule* Algoritma Apriori Pada Sistem Informasi Penjualan Adien Mart Sebagai Strategi Promosi Penjualan Barang Kurang Laris”, 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 ada tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika di kemudian hari pernyataan ini tidak benar.

Jember, 24 Mei 2017

Yang menyatakan,

Eky Angliara

NIM 122410101041

SKRIPSI

**PENGEMBANGAN MODUL *BUNDLING* MENGGUNAKAN ASSOCIATION
RULE ALGORITMA APRIORI PADA SISTEM INFORMASI PENJUALAN
ADIEN MART SEBAGAI STRATEGI PROMOSI PENJUALAN BARANG
KURANG LARIS**

Oleh

Eky Angliara

NIM 122410101041

Pembimbing :

Dosen Pembimbing Utama : Anang Andrianto., S.T., M.T.

Dosen Pembimbing Pendamping : Muhamad Arief Hidayat.,S.Kom.,M.Kom

PENGESAHAN PEMBIMBING

Skripsi berjudul “Pengembangan Modul *Bundling* Menggunakan *Association Rule* Algoritma Apriori Pada Sistem Informasi Penjualan Adien Mart Sebagai Strategi Promosi Penjualan Barang Kurang Laris” telah diuji dan disahkan pada:

hari, tanggal : Kamis, 15 juni 2017

tempat : Program Studi Sistem Informasi Universitas Jember.

Disetujui oleh :

Pembimbing I,

Pembimbing II,

Anang Andrianto.,S.T.,M.T

NIP 196909281993021001

Muhamad Arief H.,S.Kom.,M.Kom

NIP 198101232010121003

PENGESAHAN PENGUJI

Skripsi berjudul “Pengembangan Modul *Bundling* Menggunakan *Association Rule* Algoritma Apriori Pada Sistem Informasi Penjualan Adien Mart Sebagai Strategi Promosi Penjualan Barang Kurang Laris” telah diuji dan disahkan pada:

hari, tanggal : Kamis, 15 juni 2017

tempat : Program Studi Sistem Informasi Universitas Jember.

Tim penguji:

Penguji I,

Penguji II,

Drs. Antonius Cahya P,M.App., Sc.,Ph.D

NIP.196909281993021001

Diah Ayu Retnani W, ST., M.Eng

NIP.198603052014042001

Mengesahkan

Ketua Program Studi,

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

NIP 196704201992011001

RINGKASAN

Pengembangan Modul *Bundling* Menggunakan *Association Rule* Algoritma Apriori Pada Sistem Informasi Penjualan Adien Mart Sebagai Strategi Promosi Penjualan Barang Kurang Laris; Eky Angliara, 122410101041; 2017: 250 halaman; Program Studi Sistem Informasi Universitas Jember.

Adien Mart terletak di daerah Kabupaten Jember yang beralamatkan di Jl. Cokroaminoto, Kalisat, Kabupaten Jember. Adien Mart merupakan toko perlengkapan kebutuhan sehari-hari yang meliputi, kebutuhan sekolah, rumah tangga dan kantor. Adien mart menggunakan aplikasi penjualan berbasis desktop untuk bertransaksi. Sehingga aplikasi tersebut dapat membantu pihak Adien Mart dalam melakukan rekapitulasi dalam penjualan. Tetapi, belum ada startegi khusus yang dilakukan untuk meningkatkan penjualan barang misalkan untuk meningkatkan penjualan barang kurang laris. Strategi penjualan bisa diterapkan dengan menambah fitur *bundling* yang mampu merekomendasikan barang apa saja yang tepat untuk di paketkan (*bundling*). Fitur *bundling* dapat di kembangkan dengan menggunakan data penjualan sebelumnya untuk mengetahui sebuah informasi baru yang dapat dimanfaatkan untuk pencarian pola pembelian konsumen.

Pada penelitian ini dibangun sebuah Sistem Informasi Penjualan yang di dalamnya terdapat pengembangan fitur *budling* yang diharapkan mampu sebagai strategi promosi barang kurang laris. Metode yang digunakan dalam sistem ini adalah *Associatioon Rule* Algoritma Apriori. Sistem ini dibangun menggunakan bahas pemrograman *Page Hypertext Pre-Processor* (PHP) dan *framework Code Ignither* (CI). Berdasarkan hasil pengujian terhadap sistem, banyaknya rule dalam pencarian pola pembelian konsumen dipengaruhi oleh nilai minimum *support* dan nilai minimum *confidence* yang menjadi batasan.

PRAKATA

Puji syukur kehadirat Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul Pengembangan Modul *Bundling* Menggunakan *Assosiation Rule* Algoritma Apriori Pada Sistem Informasi Penjualan Adien Mart Sebagai Strategi Promosi Penjualan Barang Kurang Laris. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sistem Informasi Universitas Jember.

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

1. Prof. Drs. Slamini, M.Comp.Sc., Ph.D., selaku Ketua Program Studi Sistem Informasi Universitas Jember;
2. Anang Andrianto, S.T., M.T., selaku Dosen Pembimbing Utama sekaligus Dosen Pembimbing Akademik dan Muhammad Arief Hidayat S.Kom., M.Kom., selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi;
3. Seluruh Bapak dan Ibu dosen beserta staf karyawan di Program Studi Sistem Informasi Universitas Jember;
4. Ibunda Lilis Tri Astuti dan Ayahanda Heru Gtaot Subegti tercinta yang selalu mendukung dan mendoakan;
5. Kakak Ganda Satya Bramara dan Dinda Putri Astuti yang selalu mendukung dan menghibur;
6. Adien Mart sebagai lokasi penelitian yang telah meluangkan waktu yang membantu dalam melancarkan skripsi ini;
7. Keluarga besar FORMATION angkatan 2012 yang telah menjadi keluarga selama menempuh pendidikan S1;

8. Sahabat kontrakan Puri Bunga Nirwana yaitu Puput, Seftya, Ghausilia, Dwi, Jeje, Esa, Delia dan sahabat seperjuangan yaitu Haris, bintang, bagus setya yang selalu membantu, menghibur dan memberi dukungan;
9. Direktorat Pembelajaran dan Kemahasiswaan Ditjen Dikti yang telah memberikan beasiswa bidikmisi kepada penulis sehingga dapat menyelesaikan pendidikan S1;
10. Teman-Teman Program Studi Sistem Informasi di semua angkatan.
11. Kelompok KKN 139 desa Gunungsari kecamatan Umbulsari ;
12. Keluarga besar laboratorium basis data Prodi Sistem Informasi;
13. Semua pihak yang tidak dapat disebutkan satu persatu.

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

Jember, 25 April 2017

Penulis,

Eky Angliara

NIM.122410101041

DAFTAR ISI

	Halaman
PERSEMBAHAN.....	ii
MOTTO	iii
PERNYATAAN.....	iv
SKRIPSI.....	v
PENGESAHAN PEMBIMBING.....	vi
PENGESAHAN PENGUJI.....	vii
RINGKASAN	viii
PRAKATA.....	ix
DAFTAR ISI.....	xi
DAFTAR TABEL.....	xv
DAFTAR GAMBAR	xvii
DAFTAR LAMPIRAN.....	xxii
BAB I. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan dan Manfaat	3
1.3.2 Tujuan.....	3
1.3.2 Manfaat.....	4
1.4 Batasan Masalah	4
1.5 Sistematika Penulisan.....	5
BAB 2. TINJAUAN PUSTAKA	7
2.1 Penelitian Terdahulu	7
2.2 Kondisi Lingkungan Minimarket Adien mart.....	8
2.2.1 Struktur Organisasi Adiem Mart.....	8

2.2.2 Strategi <i>Bundling</i>	8
Minimarket Adien mart memiliki banyak transaksi penjualan setiap harinya, dengan adanya data transaksi penjualan Adien mart dapat memanfaatkan data tersebut untuk diolah kembali dan menghasilkan sebuah informasi yang dapat digunakan sebagai strategi promosi. strategi promosi yang dapat digunakan yaitu <i>Product bundling</i> . <i>Product bundling</i> ini digunakan untuk mempromosikan barang-barang yang selama ini kurang laris/kurang diminati oleh konsumen.	9
2.3 Promosi.....	9
2.4 <i>Data Mining</i>	10
2.5 <i>Association Rule</i>	11
2.6 Metode <i>Apriori</i>	12
2.6.1 Algoritma <i>Apriori</i>	12
2.6.2 Cara Kerja Algoritma <i>Apriori</i>	13
2.7 Pengujian Tingkat Akurasi Hasil <i>Association Rule</i>	15
BAB 3. METODOLOGI PENELITIAN.....	17
3.1 Jenis Penelitian.....	17
3.2 Tempat dan Waktu Penelitian	17
3.3 Tahapan Penelitian	17
3.3.1 <i>Requirement</i> (Analisis Kebutuhan)	18
3.3.2 System and Software Desain.....	19
3.3.3 Implementation and UnitTesting.....	20
3.3.4 Integration and System Testing.....	20
3.3.5 <i>Operasional Maintenance</i>	21
Pemeliharaan merupakan proses perawatan sistem digunakan oleh pengguna. Pemeliharaan dilakukan dengan mengecek kinerja sistem secara berkala.	

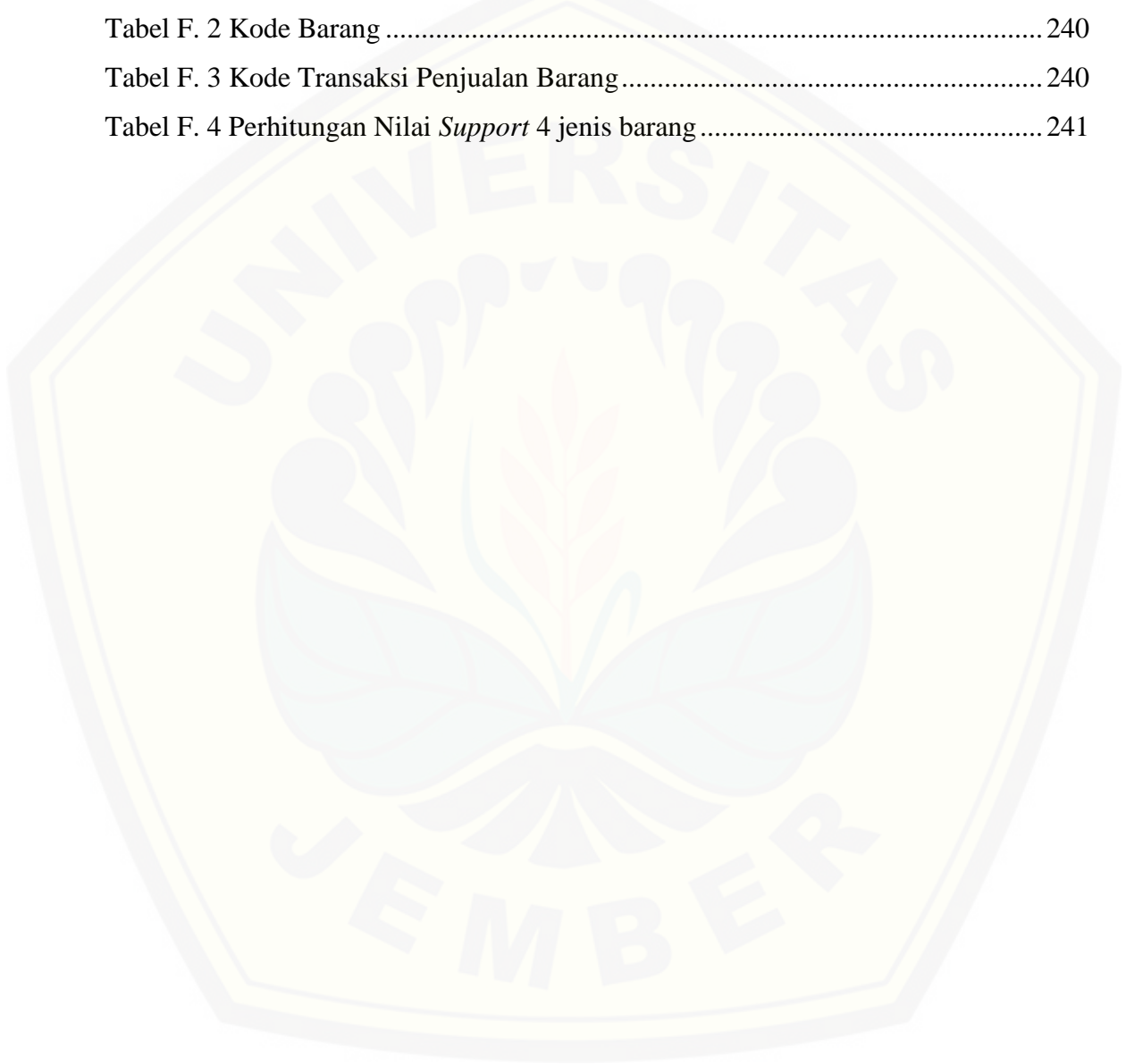
Pengecekan dilakukan apakah kinerja sistem masih berjalan dengan baik dan memperbaiki jika terdapat kerusakan. Operasional Maintenance	21
BAB 4. PENGEMBANGAN SISTEM	22
4.1 Analisis Kebutuhan	22
4.1.1 SOP (<i>Statement of purpose</i>).....	23
4.1.2 Pengumpulan Data	23
4.2 <i>Desain System</i>	25
4.2.1. <i>Bussiness Process</i>	25
4.2.2. <i>Use Case Diagram</i>	26
4.2.3. <i>Use Case Skenario</i>	29
4.2.4. <i>Activity Diagram</i>	41
4.2.5. <i>Sequence Diagram</i>	55
4.2.6. <i>Class Diagram</i>	68
4.2.7. <i>Entity Relationship Diagram (ERD)</i>	69
4.3 Penulisan Kode Program	72
4.4 Pengujian	72
4.4.1 White Box	72
4.4.2 Metode Black Box.....	79
4.4.3 Uji Akurasi <i>Frequent Itemsets</i>	79
BAB 5. HASIL DAN PEMBAHASAN	80
5.1 Hasil Penelitian	80
5.1.1. Perhitungan Algoritma Apriori untuk Penentuan Barang yang Sering Dibeli Bersamaan (<i>Frequent Itemsets</i>).	80
5.1.2. Implementasi Algoritma <i>Apriori</i> untuk Pengembangan Modul <i>Bundling</i>	80

5.2 Hasil Pembuatan Pengembangan Modul <i>Bundling</i> Menggunakan Association Rule Algoritma Apriori pada Sistem Informasi Penjualan Adien Mart sebagai Strategi Promosi Penjualan Barang Kurang laris.....	87
5.2.1. Fitur <i>Login</i>	87
5.2.2. Halaman Beranda Admin.....	87
5.2.3. Fitur Manajemen Barang.....	88
5.2.4. Fitur Data Transaksi.....	94
5.2.5. Fitur Data Pola.....	95
5.2.6. Fitur Supplier.....	96
5.2.7. Fitur User.....	98
5.2.8. Fitur Data Barang Pegawai.....	99
5.2.9. Fitur Transaksi Penjualan (Pegawai).....	100
5.2.10. Kelebihan Sistem.....	101
5.2.11. Keterbatasan Sistem.....	102
5.3 Pembahasan Implementasi Algoritma <i>Apriori</i> dalam Pencarian Barang yang Sering Dibeli Bersamaan (<i>Frequent Itemsets</i>).....	102
BAB 6. PENUTUP.....	111
6.1 Kesimpulan.....	111
6.2 Saran.....	112
DAFTAR PUSTAKA.....	113
LAMPIRAN.....	115

DAFTAR TABEL

	Halaman
Tabel 2.1 Hasil aturan asosiasi.....	7
Tabel 2.2 Hasil Perhitungan Pola penjualan	8
Tabel 4.1 Deskripsi Aktor	28
Tabel 4.2 Deskripsi <i>Use case</i>	28
Tabel 4.3 <i>Usecase</i> Skenario Manajemen Data Barang	29
Tabel 4.4 <i>Usecase</i> Skenario Manajemen Data Kategori.....	34
Tabel 4. 5 <i>Test case function</i> pencarianpola().....	78
Tabel 4. 6 <i>Test case function</i> datapilih(\$kode)	78
Tabel 5.1 Daftar Dataset Transaksi	103
Tabel 5.2 Hasil <i>Frequent Itemsets</i> dengan Pengaturan Nilai Terkecil.....	105
Tabel 5.3 Hasil <i>Frequent Itemsets</i> dengan <i>Minimal Support 5</i> dan <i>minimal confidence</i>	108
Tabel 5.4 Hasil <i>Frequent Itemsets Minimal Confidence 50%</i>	109
Tabel 5.5 Hasil AkurasiHasil <i>Frequent Itemsets</i>	110
Tabel A.1 <i>Use Case</i> Skenario Manajemen Data Subkategori.....	115
Tabel A.2 <i>Use Case</i> Skenario Manajemen Data Supplier	120
Tabel A.3 <i>Use Case</i> Skenario Manajemen Data User	124
Tabel A.4 <i>Use Case</i> Skenario Manajemen Stok Barang.....	128
Tabel A.5 <i>Use Case</i> Skenario Melihat Transaksi Penjualan.....	131
Tabel A.6 <i>Use Case</i> Skenario Input Parameter Pencarian Pola.....	132
Tabel A.7 <i>Use Case</i> Skenario Rekomendasi Promosi	134
Tabel A.8 <i>Use Case</i> Skenario Transaksi Penjualan	135
Tabel A.9 <i>Use Case</i> Skenario Melihat Data Data Barang	138
Tabel A.10 <i>Use Case</i> Skenario Login.....	139

Tabel E.1 Pengujian <i>Black Box</i>	232
Tabel F. 1 Sample Transaksi Penjualan Produk.....	239
Tabel F. 2 Kode Barang	240
Tabel F. 3 Kode Transaksi Penjualan Barang.....	240
Tabel F. 4 Perhitungan Nilai <i>Support</i> 4 jenis barang.....	241



DAFTAR GAMBAR

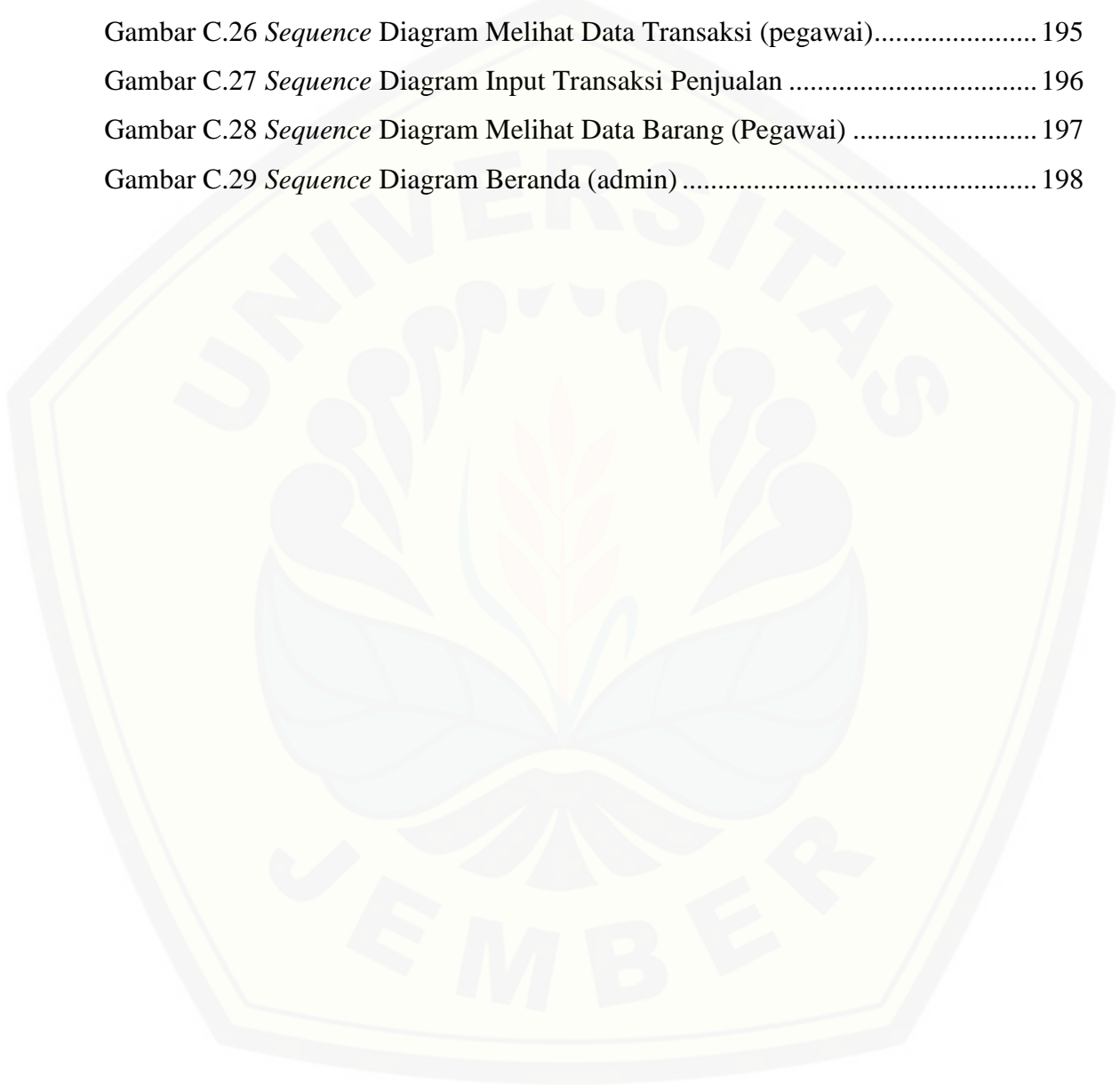
	Halaman
Gambar 2.1 <i>Flowchart</i> Algoritma Apriori	15
Gambar 3.1 Tahapan Model <i>Waterfall</i>	18
Gambar 4.1 <i>Bussiness Procces</i> Sistem Informasi Penjualan Adien Mart.....	26
Gambar 4.2 <i>Use case</i> Diagram Sistem Informasi Penjualan Adien Mart.....	27
Gambar 4.3 <i>Activity</i> Diagram Manajemen Data Barang (<i>view</i>).....	42
Gambar 4.4 <i>Activity</i> Diagram Manajemen Data Barang (<i>input</i>).....	43
Gambar 4.5 <i>Activity</i> Diagram Manajemen Data Barang (<i>edit</i>)	45
Gambar 4.6 <i>Activity</i> Diagram Manajemen Data Barang (<i>edit status aktif</i>).....	46
Gambar 4.7 <i>Activity</i> Diagram Manajemen Data Barang (<i>edit status non aktif</i>).....	47
Gambar 4.8 <i>Sequence</i> Diagram Manajemen Data Barang (<i>view</i>).....	56
Gambar 4.9 <i>Sequence</i> Diagram Manajemen Data Barang (<i>input</i>).....	59
Gambar 4.10 <i>Sequence</i> Diagram Manajemen Data Barang (<i>edit</i>)	60
Gambar 4. 11 <i>Sequence</i> Diagram Manajemen Data Barang (<i>edit status aktif</i>).....	61
Gambar 4. 12 <i>Sequence</i> Diagram Manajemen Data Barang (<i>edit status non aktif</i>)...	62
Gambar 4. 13 Class Digaram	70
Gambar 4. 14 Entity Relational Diagram.....	71
Gambar 4.15 <i>Listing</i> Program <i>Function__construct()</i>	74
Gambar 4. 16 <i>Listing</i> Program <i>Function</i> pencarianpola()	74
Gambar 4. 17 <i>Listing</i> Program <i>Function</i> datapilih(\$kode)	75
Gambar 4. 18 <i>Listing</i> Program <i>Function</i> apriori()	75
Gambar 4. 19 <i>Diagram Alir</i> Function <i>pencarianpola()</i>	76
Gambar 4.20 <i>Diagram Alir</i> Function <i>datapilih()</i>	76
Gambar 4. 21 <i>Diagram Alir</i> Function <i>apriori()</i>	77

Gambar 5.1 Halaman <i>Form Parameter</i>	81
Gambar 5.2 Halaman Rekomendasi Hasil Analisa Pencarian <i>Frequent Itemsets</i>	82
Gambar 5.3 Kode Program pada <i>function</i> <i>datapilih()</i>	83
Gambar 5.4 Kode Program pada <i>Function</i> <i>apriori()</i>	84
Gambar 5.5 Kode Program Pencarian <i>Frequent Itemset</i>	85
Gambar 5.6 Potongan Kode Program <i>Class</i> <i>datapilih</i>	85
Gambar 5.7 Kode Program Mencari Rekomendasi Barang Kurang Laris.....	86
Gambar 5.8 Halaman <i>Login</i>	87
Gambar 5.9 Halaman Beranda Admin	88
Gambar 5.10 Tampilan Submenu Manajemen Barang.....	88
Gambar 5.11 Halaman Data Barang	89
Gambar 5.12 Halaman Form Data Barang.....	89
Gambar 5.13 Halaman <i>Form</i> Edit Barang	90
Gambar 5.14 Halaman Stok Barang.....	90
Gambar 5.15 Halaman Form Data Barang.....	91
Gambar 5.16 Halaman Kategori	91
Gambar 5.17 Halaman <i>Form</i> Kategori.....	92
Gambar 5.18 Halaman <i>Form</i> Edit Kategori.....	92
Gambar 5.19 Halaman Subkategori	93
Gambar 5.20 Halaman Form Subkategori	93
Gambar 5.21 Halaman <i>Form</i> Edit Subkategori.....	94
Gambar 5.22 Halaman Data Transaksi	94
Gambar 5.23 Halaman Detail Transaksi	95
Gambar 5.24 Halaman Pencarian Pola.....	95
Gambar 5.25 Halaman Rekomendasi.....	96
Gambar 5.26 Halaman Supplier.....	96
Gambar 5.27 Halaman Form Supplier	97
Gambar 5.28 Halaman <i>Form</i> Edit Supplier	97
Gambar 5.29 Halaman <i>User</i>	98

Gambar 5.30 <i>Halaman Form User</i>	98
Gambar 5.31 <i>Halaman Form Edit User</i>	99
Gambar 5.32 <i>Halaman Data Barang</i>	100
Gambar 5.33 <i>Halaman Transaksi Penjualan</i>	100
Gambar 5.34 <i>Halaman Detail Transaksi</i>	101
Gambar 5.35 <i>Halaman Form Tambah Transaksi</i>	101
Gambar B.1 <i>Activity Diagram Manajemen Data kategori (view)</i>	141
Gambar B.2 <i>Activity Diagram Manajemen Data Kategori (input)</i>	142
Gambar B.3 <i>Activity Diagram Merubah Status Aktif Data Kategori</i>	143
Gambar B.4 <i>Activity Diagram Merubah Status Non Aktif Data Kategori</i>	144
Gambar B.5 <i>Activity Diagram Manajemen Data Kategori (edit)</i>	145
Gambar B.6 <i>Activity Diagram Manajemen Data Subkategori (view)</i>	146
Gambar B.7 <i>Activity Diagram Manajemen Data Subkategori (input)</i>	147
Gambar B.8 <i>Activity Diagram Merubah Status Aktif Data Subkategori</i>	148
Gambar B.9 <i>Activity Diagram Merubah Status Non Aktif Data Subkategori</i>	149
Gambar B.10 <i>Activity Diagram Manajemen Data Subkategori (edit)</i>	150
Gambar B.11 <i>Activity Diagram Manajemen Data Supplier (view)</i>	151
Gambar B.12 <i>Activity Diagram Manajemen Data Supplier (input)</i>	152
Gambar B.13 <i>Activity Diagram Merubah Status Aktif Data Supplier</i>	153
Gambar B.14 <i>Activity Diagram Merubah Status Non Aktif Data Supplier</i>	154
Gambar B.15 <i>Activity Diagram Manajemen Data Supplier (edit)</i>	155
Gambar B.16 <i>Activity Diagram Manajemen Data User (view)</i>	156
Gambar B.17 <i>Activity Diagram Manajemen Data User (input)</i>	157
Gambar B.18 <i>Activity Diagram Merubah Status aktif Data User</i>	158
Gambar B.19 <i>Activity Diagram Merubah Status Non Aktif Data User</i>	159
Gambar B.20 <i>Activity Diagram Manajemen Data User (edit)</i>	160
Gambar B.21 <i>Activity Diagram Manajemen Stok Barang (view)</i>	161
Gambar B.22 <i>Activity Diagram Manajemen Stok Barang (input)</i>	162
Gambar B.23 <i>Activity Diagram Melihat Data Transaksi (view)</i>	163

Gambar B.24 <i>Activity Diagram</i> Input Parameter Pencarian Pola	164
Gambar B. 25 <i>Activity Diagram</i> Rekomendasi Promosi	165
Gambar B.26 <i>Activity Diagram</i> Input Transaksi Penjualan	166
Gambar B.27 <i>Activity Diagram</i> Melihat Data Transaksi (pegawai)	167
Gambar B.28 <i>Activity Diagram</i> Melihat Data Barang (pegawai)	168
Gambar B.29 <i>Activity Diagram</i> Melihat Beranda (Admin)	168
Gambar B.30 <i>Activity Diagram</i> Login	169
Gambar C.1 <i>Sequence Diagram</i> Manajemen Data Kategori (view)	170
Gambar C.2 <i>Sequence Diagram</i> Manajemen Data Kategori (input).....	171
Gambar C.3 <i>Sequence Diagram</i> Merubah Status Aktif Data Kategori.....	172
Gambar C.4 <i>Sequence Diagram</i> Merubah Status Non Aktif Data Kategori.....	173
Gambar C.5 <i>Sequence Diagram</i> Manajemen Data Kategori (edit).....	174
Gambar C.6 <i>Sequence Diagram</i> Manajemen Data Subkategori (view).....	175
Gambar C.7 <i>Sequence Diagram</i> Manajemen Data Subkategori (input)	176
Gambar C.8 <i>Sequence Diagram</i> Merubah Status Aktif Data Subkategori.....	177
Gambar C.9 <i>Sequence Diagram</i> Merubah Status Non Aktif Data Subkategori.....	178
Gambar C.10 <i>Sequence Diagram</i> Manajemen Data Subkategori (edit).....	179
Gambar C.11 <i>Sequence Diagram</i> Manajemen Data Supplier (view).....	180
Gambar C.12 <i>Sequence Diagram</i> Manajemen Data Supplier (input)	181
Gambar C.13 <i>Sequence Diagram</i> Merubah Status Aktif Data Supplier	182
Gambar C.14 <i>Sequence Diagram</i> Merubah Status Non Aktif Data Supplier	183
Gambar C.15 <i>Sequence Diagram</i> Manajemen Data Supplier (edit)	184
Gambar C.16 <i>Sequence Diagram</i> Manajemen Data User (view).....	185
Gambar C.17 <i>Sequence Diagram</i> Manajemen Data User (input)	186
Gambar C. 18 <i>Sequence Diagram</i> Merubah Status Aktif Data User	187
Gambar C.19 <i>Sequence Diagram</i> Merubah Status Non Aktif Data User.....	188
Gambar C.20 <i>Sequence Diagram</i> Manajemen Data user (edit)	189
Gambar C.21 <i>Sequence Diagram</i> Manajemen Stok Barang (view).....	190
Gambar C.22 <i>Sequence Diagram</i> Manajemen Stok Barang (input)	191

Gambar C.23 <i>Sequence</i> Diagram Melihat Data Transaksi (admin).....	192
Gambar C.24 <i>Sequence</i> Diagram Input Parameter Pola	193
Gambar C.25 <i>Sequence</i> Diagram Rekomendasi Promosi.....	194
Gambar C.26 <i>Sequence</i> Diagram Melihat Data Transaksi (pegawai).....	195
Gambar C.27 <i>Sequence</i> Diagram Input Transaksi Penjualan	196
Gambar C.28 <i>Sequence</i> Diagram Melihat Data Barang (Pegawai)	197
Gambar C.29 <i>Sequence</i> Diagram Beranda (admin)	198



DAFTAR LAMPIRAN

	Halaman
Lampiran A. <i>Use Case Skenario</i>	115
Lampiran B. <i>Activity Diagram</i>	141
Lampiran C. <i>Sequence Diagram</i>	170
Lampiran D. <i>Kode Program</i>	199
Lampiran E. <i>Pengujian</i>	232
Lampiran F. <i>Perhitungan Algoritma Apriori</i>	239
Lampiran G. <i>Hasil Wawancara</i>	251

BAB I. PENDAHULUAN

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

1.1 Latar Belakang

Adien mart merupakan minimarket serba ada yang menyediakan berbagai macam kebutuhan pokok dan kebutuhan sehari-hari. minimarket yang beralamat di jalan Cokroaminoto, Kalisat, Kabupaten Jember ini menyediakan lebih dari 5000 jenis produk makanan dan non makanan. Setiap hari Adien mart melakukan banyak transaksi penjualan. Agar memudahkan proses pencatatan dan manajemen penjualan, Adien mart menggunakan sebuah sistem informasi penjualan sederhana. Sistem informasi yang digunakan hanya sebatas mencatat transaksi penjualan saja.

Perkembangan sistem informasi saat ini sudah semakin maju terutama dalam dunia bisnis. Salah satunya adalah bisnis minimarket yang menjamur dengan menggunakan sistem yang lebih modern. Sistem yang modern dan kompleks diperlukan untuk membantu kegiatan bisnis agar lebih efektif dan efisien. Adien mart membutuhkan sistem yang lebih modern dan kompleks untuk meningkatkan pelayanan dan layanan sehingga dapat mengontrol secara *detail* manajemen barang dan manajemen transaksi dengan tepat guna.

Setiap minimarket memiliki strategi bisnis masing-masing untuk mencapai keuntungan sebesar-besarnya. Banyak strategi yang diterapkan, salah satunya adalah strategi penggabungan barang menjadi satu paket (*product bundling*). Strategi *product bundling* diterapkan pada banyak perusahaan untuk mendongkrak penjualan, misal *bundling* kartu perdana dengan *handphone*, *bundling* komputer dengan printer dan *bundling* tas dengan sepatu. *Bundling* barang berpengaruh positif dan signifikan

terhadap variabel perilaku pembelian konsumen baik secara parsial maupun simultan (Fauziah, 2012)

Strategi *Product bundling* juga dapat diterapkan pada Adien mart. Strategi ini dapat dimanfaatkan untuk meningkatkan penjualan barang kurang laris, yaitu dengan cara mencari barang yang sering dibeli bersamaan dan mengganti salah satu barang dengan barang yang tingkat penjualannya rendah. Informasi barang yang sering dibeli bersamaan dapat diketahui dari histori transaksi penjualan yang akan menghasilkan pola penjualan. Histori tersebut merupakan data transaksi yang dapat diterapkan proses *datamining* dengan metode *Assosiation rule* sehingga menghasilkan informasi baru yang bermanfaat. histori yang digunakan sesuai dengan periode yang diinginkan.

Assosiation rule adalah teknik *datamining* untuk menemukan aturan assosiatif antara suatu kombinasi item. Pencarian pola penjualan barang dapat dilakukan dengan menggunakan *assosiation rule* dan perhitungan algoritma apriori. Misalkan aturan assosiatif dari analisa pembelian di suatu pasar swalayan yang dapat diketahui berapa besar kemungkinan seorang pelanggan membeli roti bersamaan dengan susu. Dengan pengetahuan tersebut pemilik pasar swalayan dapat mengatur penempatan barang dan startegi promosi dengan memakai kupon diskon untuk kombinasi barang barang tertentu.

Penelitian mengenai pencarian pola penjualan telah banyak dilakukan, salah satunya dilakukan oleh (Amalia, 2015) dalam jurnalnya yang berjudul pencarian pola transaksi penjualan sebagai strategi penjualan barang di Senyum Media Jember menggunakan *association rule* algoritma apriori. Penelitian tersebut melakukan pencarian pola penjualan dengan memanfaatkan histori data transaksi yang dapat menghasilkan data pola penjualan yang nantinya dapat dijadikan sebagai rekomendasi barang promosi. Barang promosi dapat dilihat dari pola penjualan yang terbentuk dan dengan melihat frekuensi penjualan terendah dari suatu barang.

Berdasarkan latar belakang diatas, penggunaan algoritma apriori dapat membantu dalam pencarian pola penjualan yang dapat menghasilkan informasi baru berupa barang yang paling sering dibeli bersamaan. Data barang yang sering dibeli

bersamaan tersebut dapat di jadikan acuan dalam menentukan strategi promosi penjualan berupa *bundling* rekomendasi barang kurang laris dalam satu subkategori yang sama.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan di atas, maka rumusan masalah dalam penitian ini adalah sebagai berikut:

1. Bagaimana mengimplementasikan *association rule* algoritma apriori dalam pengembangan modul *bundling* sebagai strategi promosi penjualan barang kurang laris?
2. Bagaimana merancang dan membangun sebuah sistem informasi penjualan dengan pengembangan modul *bundling* menggunakan *association rule* algoritma *apriori*?

1.3 Tujuan dan Manfaat

Berikut merupakan tujuan yang ingin dicapai dan manfaat yang ingin diperoleh dalam penelitian ini.

1.3.2 Tujuan

Adapun tujuan yang ingin dicapai dalam penelitian ini adalah:

- a. Mengimplementasikan *assosiation rule* algoritma apriori pada pengembangan modul *bundling* pada sistem informasi penjualan Adien mart sebagai strategi promosi penjualan barang kurang laris.
- b. Merancang dan membangun sebuah sistem informasi penjualan dengan pengembangan modul *bundling* menggunakan *assosiation rule* algoritma apriori.

1.3.2 Manfaat

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Manfaat akademis

Hasil dari penelitian ini diharapkan dapat membantu siapa saja yang membutuhkan informasi yang berhubungan dengan judul penelitian ini. Selain itu, untuk menambah varian judul penelitian yang ada Pada Program Studi Sistem Informasi Universitas Jember.

2. Manfaat bagi peneliti

Mengetahui bagaimana proses penerapan *association rule* algoritma *apriori* pada pengembangan modul *bundling* sistem informasi penjualan Adien mart sebagai strategi promosi penjualan barang kurang laris.

3. Manfaat bagi objek penelitian

Membantu Adien mart dalam memberi rekomendasi *bundling* sebagai strategi promosi penjualan barang kurang laris.

1.4 Batasan Masalah

Agar tidak terjadi penyimpangan dalam proses penelitian dan pembangunan Pengembangan Modul *Bundling* menggunakan *Association Rule* Algoritma *Apriori* pada Sistem Informasi Penjualan Adien mart sebagai Strategi Promosi Penjualan Barang Kurang Laris, maka ditetapkan beberapa batasan permasalahan sebagai berikut:

- a. Pengembangan Modul *Bundling* menggunakan *Association Rule* Algoritma *Apriori* pada Sistem Informasi Penjualan Adien mart terfokus pada pengembangan modul *bundling* yang menghasilkan *frequent itemset* dan digunakan untuk rekomendasi *budling* barang kurang laris.
- b. Penelitian menggunakan data transaksi penjualan Adien mart.
- c. Sistem menggunakan *assosiation rule* dengan algoritma *apriori*.
- d. Sistem dapat menampilkan pola penjualan.

- e. Sistem dapat menampilkan rekomendasi barang promosi.
- f. Sistem dibangun berbasis web menggunakan bahasa pemrograman php.

1.5 Sistematika Penulisan

Adapun sistematika penulisan skripsi ini adalah sebagai berikut:

1. Pendahuluan

Bab ini memuat uraian tentang latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika penulisan skripsi yang masing-masing diuraikan dalam subbab tersendiri.

2. Tinjauan Pustaka

Bab ini memaparkan tinjauan terhadap hasil-hasil penelitian terdahulu berkaitan dengan masalah yang dibahas, landasan materi dan konsep mengenai *datamining* dan *association rule* algoritma apriori.

3. Metodologi Penelitian

Bab ini menguraikan tentang tempat dan waktu penelitian, metode penelitian, metode pengumpulan data, metode analisis data, dan teknik pengembangan sistem yang digunakan dalam penelitian.

4. Perancangan Sistem

Bab ini menguraikan tentang tentang perancangan desain sistem. 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).

5. Hasil dan Pembahasan

Bab ini memaparkan hasil dan pembahasan dari sistem yang sudah dikembangkan.

6. Penutup

Bab ini berisi kesimpulan penelitian yang telah dilakukan dan saran untuk penelitian selanjutnya.



BAB 2. TINJAUAN PUSTAKA

Pada bagian ini dipaparkan tinjauan yang berkaitan dengan masalah yang dibahas, kajian teori yang berkaitan dengan masalah, dan juga penelitian-penelitian terdahulu.

2.1 Penelitian Terdahulu

Penelitian sebelumnya yang berjudul “Implementasi Data Mining dengan Metode Algoritma Apriori dalam Menentukan Pola Pembelian Obat” yang dilakukan oleh (Robi Yanto, 2015) Jurusan Sistem Informasi, STIMIK Bina Nusantara Jaya Lubuklingau, meneliti pola transaksi penjualan obat dengan menggunakan batasan berupa minimum *support* sebesar 40% dan minimal *confidence* sebesar 70%, sehingga menghasilkan aturan asosiasi yang terdapat pada Tabel 2.1

Tabel 2.1 Hasil aturan asosiasi

Aturan	Confidence	
Jika Membeli amoxicilin maka membeli asamefenamat	9/12	75%
Jika Membeli cefadroxil maka membeli sanmol	7/9	77.77%

Sumber: (Robi Yanto, 2015)

Penelitian lainnya dengan judul “Pencarian Pola Transaksi Penjualan Sebagai Strategi Penjualan Barang Menggunakan *Association Rule* Algoritma Apriori (Studi Kasus : Senyum Media Jember)” dilakukan oleh Eka Amalia K.P dari Program Studi Sistem Informasi, Universitas Jember. Dalam penelitian ini dilakukan pencarian pola penjualan barang yang menghasilkan *lift ratio* sebanyak 1,5955 yang menunjukkan bahwa pola yang (Amalia, 2015) terbentuk memiliki manfaat dan hasil aturan asosiasi tersebut semakin kuat (Robi Yanto, 2015). Hasil perhitungan pola penjualan dapat dilihat pada Tabel 2.2

Tabel 2.2 Hasil Perhitungan Pola penjualan

Rule	Frekuensi	<i>Confidence</i>	<i>Benchmark Ratio</i>	<i>Lift Ratio</i>
BRG-004, BRG-015	43	87,7551	0,55	1,5955
BRG-015, BRG-004	43	86	0,55	1,5955
Persentase Split	183.34	0.41	1.7	

Sumber: (Amalia, 2015)

2.2 Kondisi Lingkungan Minimarket Adien mart

2.2.1 Struktur Organisasi Adiem Mart

Adien Mart adalah Minimarket serba ada yang menjual berbagai kebutuhan sehari-hari. Adien mart merupakan salah satu minimarket grosir terbesar di daerah Kalisat. Adien Mart memiliki 10 pegawai yang terdiri dari 1 admin, 4 kasir dan 5 pramuniaga. Admin adalah pegawai yang bertugas untuk pengadaan barang, kasir bertugas untuk input data transaksi dan pramuniaga bertugas untuk mengatur letak barang dan melayani pertanyaan konsumen.

Adien mart menjual banyak barang dengan berbagai jenis barang dan merk. Adanya banyak barang yang mempunyai kegunaan yang sama mengakibatkan tingkat penjualan yang tidak seimbang sehingga beberapa barang tergolong dalam barang yang kurang laris dengan tingkat penjualannya yang rendah. Mengatasi permasalahan tersebut Adien mart dapat melakukan Strategi *Bundling*, dimana strategi *bundling* ini bisa menjadi solusi menumpuknya barang yang kurang laris dan strategi untuk meningkatkan penjualan.

2.2.2 Strategi *Bundling*

Strategi *Bundling* adalah salah satu strategi dalam pemasaran yang merupakan penggabungan beberapa produk menjadi satu paket penjualan. *Bundling* dibagi menjadi 2 dimensi penting yaitu *product bundling* dan *price bundling*. *Price bundling* adalah penjualan dua atau lebih barang yang berbeda dalam satu paket harga yang di

diskon dengan aspek harga yang lebih ditonjolkan. Sedangkan, *product bundling* adalah menyatukan barang menjadi satu paket dengan harapan konsumen bisa lebih dimudahkan dalam membeli barang yang diinginkan (Insiroh, 2013). Pada penelitian (Setiani, 2011) di jelaskan bahwa 87% dari 100 orang membeli handphone nexian karena program *product bundling* yang di terapkan oleh PT. Nexian. *Product budling* dapat menarik minat konsumen.

Minimarket Adien mart memiliki banyak transaksi penjualan setiap harinya, dengan adanya data transaski penjualan Adien mart dapat memanfaatkan data tersebut untuk diolah kembali dan menghasilkan sebuah informasi yang dapat digunakan sebagai strategi promosi. strategi promosi yang dapat digunakan yaitu *Product bundling*. *Product bundling* ini digunakan untuk mempromosikan barang-barang yang selama ini kurang laris/kurang diminati oleh konsumen.

2.3 Promosi

Promosi dalam bisnis penjualan sangat penting karena dapat meningkatkan keuntungan perusahaan. Promosi merupakan jalan untuk mencapai penghasilan yang optimal bagi perusahaan. Promosi penjualan yang tepat harus dilakukan, karena pada dasarnya promosi penjualan mempunyai tujuan penting dalam mendukung tercapainya keuntungan omset yang optimal (Nurul, 2013)

Promosi memiliki berbagai macam jenis strategi, hal yang terpenting adalah bagaimana memberikan suatu nilai baru dalam promosi. Manfaat promosi adalah mendorong atau merangsang pelanggan untuk membeli. Promosi dapat berupa jaminan kualitas, asuransi pengembalian, hadiah, potongan harga dan pemaketan produk. Berbagai jenis promosi tersebut dapat membuat pelanggan terdorong untuk membeli. Iklan adalah alat jangka panjang untuk membentuk perilaku pasar terhadap suatu merk, sedangkan promosi penjualan adalah sebagai alat jangka pendek untuk memicu terjadinya tindakan pembelian (Nurul, 2013).

Berdasarkan definisi di atas dapat disimpulkan promosi adalah suatu kegiatan bidang marketing yang merupakan komunikasi perusahaan kepada pembeli atau konsumen untuk menarik perhatian, mempengaruhi, mengajak dan menarik minat konsumen. Mini market Adien mart dapat melakukan promosi dengan memanfaatkan pola penjualan transaksi.

2.4 Data Mining

Data Mining merupakan serangkaian proses untuk menggali nilai tambah berupa informasi yang selama ini tidak diketahui secara manual dari suatu basis data atau bisa disebut dengan KDD (*Knowledge Discovery in Database*). Informasi diperoleh dengan cara mengekstraksi dan mengenali pola yang penting atau menarik dari data yang terdapat dalam basis data (Kusrini & Luthfi, 2009) KDD (*Knowledge Discovery in Database*) adalah keseluruhan proses non-trivial untuk mencari dan mengidentifikasi pola (*pattern*) dalam data, dimana pola yang ditemukan bersifat sah, baru, dapat bermanfaat dan dapat dimengerti (Fayyad dikutip dalam Kusrini & Luthfi, 2009).

Proses data mining dilakukan dalam beberapa proses dengan tahapan sebagai berikut (Mujib & dkk, 2013):

a. Pembersihan data (*Data Cleaning*)

Pembersihan data merupakan proses menghilangkan *noise* data dan data yang tidak konsisten atau tidak relevan.

b. Integrasi data (*Data Integration*)

Integrasi data merupakan penggabungan data dari berbagai *database* ke dalam satu *database* baru, proses ini dilakukan jika terdapat data yang berada dalam *database* yang berbeda.

c. Seleksi data (*Data Selection*)

Data yang ada pada *database* sering kali tidak semuanya dipakai karena kurang konsisten atau tidak relevan, oleh karena itu hanya data yang sesuai untuk dianalisis yang diambil dari *database*.

d. Transformasi data (*Data Transformation*)

Data diubah atau disatukan menjadi format yang sesuai untuk diproses dalam *data mining*.

e. Proses Mining

Merupakan proses utama saat metode diterapkan untuk menemukan pengetahuan berharga dan tersembunyi dari data. Terdapat beberapa metode yang dapat digunakan salah satunya klasifikasi.

f. Evaluasi pola (*Pattern Evaluation*)

Untuk mengidentifikasi pola-pola tertentu ke dalam *knowledge based* yang ditemukan.

g. Presentasi pengetahuan (*Knowledge Presentation*)

Merupakan visualisasi dan penyajian pengetahuan hasil proses mining sesuai dengan metode yang digunakan untuk memperoleh pengetahuan yang disajikan kepada pengguna.

2.5 Association Rule

Association dalam *data mining* adalah pekerjaan untuk menentukan mana atribut yang akan didapatkan bersamaan. Tugas dari *association rule* adalah mencari aturan antara dua barang atau lebih. *Association rule* adalah bentukjika “kejadian sebelumnya” kemudian “konsekuensinya”. (*IF antecedent, THEN consequent*). Bersamaan dengan perhitungan aturan *support* dan *confidence*. Pola asosiasi menjadi salah satu fungsionalitas yang paling menarik dalam penggalian data.

Association rule adalah teknik *data mining* untuk menemukan aturan asosiatif antara suatu kombinasi item. Contoh *association rule* yaitu analisa pembelian disuatu pasar swalayan dapat diketahui berapa besar kemungkina seorang pembeli membeli roti bersamaan dengan susu. Dengan pengetahuan tersebut pemilik pasar swalayan dapat mengatur penempatan barang atau merancang kampanye pemasaran dengan memakai kupon diskon untuk kombinasi barang tertentu (Suwarningsih, 2008)

2.6 Metode *Apriori*

2.6.1 Algoritma *Apriori*

Algoritma *Apriori* merupakan ruang pencarian kombinasi *item*, sehingga analisis dapat dilakukan dengan lebih cepat. Aturan-aturan yang dihasilkan dari algoritma *apriori* dapat diidentifikasi lagi untuk menentukan aturan mana yang dapat menghasilkan informasi lebih banyak dengan menggunakan ukuran *support* dan *confidence*. kemudian, aturan-aturan asosiasi yang telah dihasilkan dapat digunakan sebagai bahan pertimbangan untuk pengambilan keputusan dalam strategi bisnis, adapun dua proses utama yang dilakukan dalam algoritma *Apriori* (Han, 2006) yaitu :

1. *Join* (penggabungan)

Pada proses ini setiap *item* dikombinasikan dengan *item* yang lainnya sampai tidak terbentuk kombinasi lagi.

2. *Prune* (pemangkasan)

Pada proses ini hasil dari *item* yang telah dikombinasikan lalu dipangkas dengan menggunakan minimum *support* yang telah ditentukan oleh *user*.

Algoritma *apriori* menggunakan histori data yang telah diketahui sebelumnya untuk memproses informasi selanjutnya. Pada algoritma *apriori* menentukan kandidat yang muncul dengan cara memperhatikan minimum *support* dan minimum *confidence*. *Support* adalah nilai persentase kombinasi sebuah *item* dalam *database*.

Rumus *support* sebagai berikut:

Persamaan 2.1 rumus *support*

$$\text{Support}(A, B) = \frac{\sum G}{\sum T} \times 100\% \quad \dots \text{persamaan 1}$$

Keterangan:

A = Barang A

B = Barang B

$\sum G$ = Jumlah transaksi mengandung barang A dan barang B

$\sum T$ = Jumlah total transaksi

Confidence adalah nilai kepercayaan yaitu kuatnya hubungan antara item dalam sebuah apriori. *Confidence* dapat di cari setelah pola frekuensi munculnya sebuah item ditemukan. Rumus *confidence* sebagai berikut:

Persamaan 2.2 rumus *confidence*

$$Confidence(A, B) = \frac{\sum G}{\sum Z} \times 100\% \quad \dots \text{persamaan 2}$$

Keterangan:

A = Barang A

B = Barang B

$\sum G$ = Jumlah transaksi mengandung barang A dan barang B

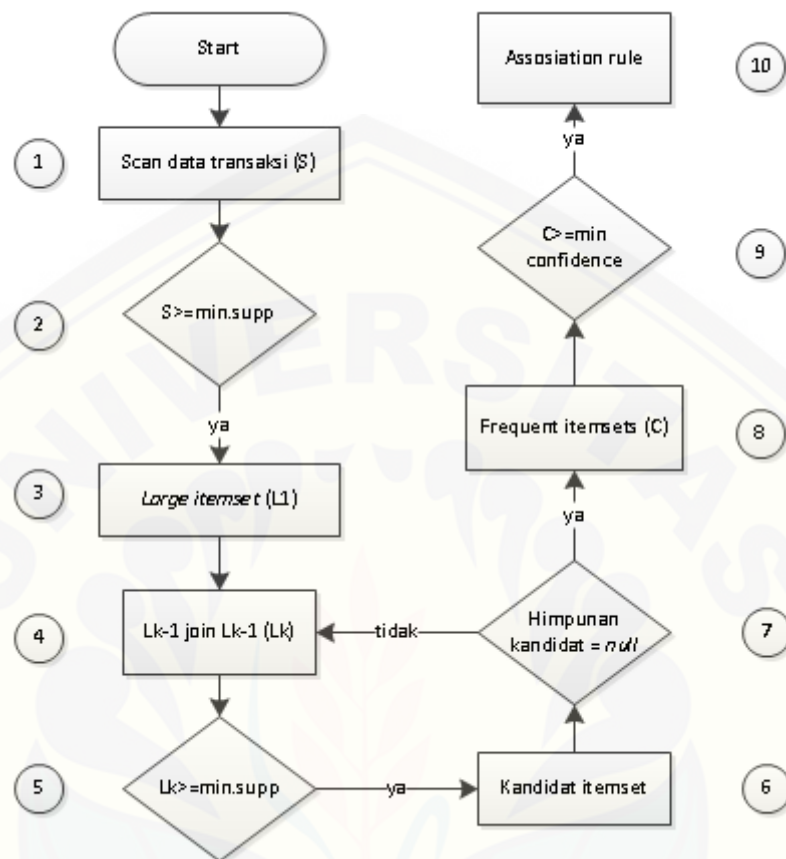
$\sum Z$ = Jumlah transaksi mengandung barang A

2.6.2 Cara Kerja Algoritma Apriori

Alur dari Algoritma Apriori adalah sebagai berikut (Tyas, 2008)

1. Sistem scan *database* untuk mendapatkan kandidat *1-itemset* (himpunan *item* yang terdiri dari 1 *item*) dan menghitung nilai *support*.
2. Kemudian nilai *support* tersebut dibandingkan dengan *minimum support* yang telah ditentukan.
3. Nilai perbandingan *support* lebih besar atau sama dengan *minimum support* maka *itemset* tersebut termasuk dalam *large itemset*.
4. Pada iterasi kedua sistem akan menggunakan hasil *large itemset* pada iterasi pertama (L_1) untuk membentuk kandidat *itemset* kedua (L_2). Pada iterasi selanjutnya sistem akan menggunakan hasil *large itemset* pada iterasi sebelumnya (L_{k-1}) untuk membentuk kandidat *itemset* berikutnya (L_k). Sistem akan menggabungkan (*join*) L_{k-1} dengan L_{k-1} untuk mendapatkan L_k , seperti pada iterasi sebelumnya sistem akan menghapus (*prune*) kombinasi *itemset* yang tidak termasuk dalam *large itemset*.

5. Setelah dilakukan operasi *join*, maka pasangan *itemset* baru hasil proses *join* tersebut juga di hitung *support*-nya.
6. Pasangan *itemset* yang nilai *support*-nya sama atau sama dengan minimum *support* masuk dalam kandidat *itemset*.
7. Proses pembentukan kandidat yang terdiri dari proses *join* dan *prune* akan terus dilakukan sehingga himpunan kandidat *itemset*-nya *null* atau sudah tidak adalagi kandidat yang akan dibentuk.
8. Himpunan kandidat yang terbentuk akan menghasilkan *frequent itemset*.
9. Nilai *confidence frequent itemset* yang terbentuk dibandingkan dengan *minimum confidence*.
10. *frequent itemset* yang memenuhi syarat *minimum confidence* akan membentuk *association rule*.



Gambar 2.1 Flowchart Algoritma Apriori

2.7 Pengujian Tingkat Akurasi Hasil *Assosiation Rule*

Pengujian akurasi digunakan untuk mengetahui tingkat akurasi hasil *association rule* yang dilakukan dengan menggunakan data test (*test set*), terdapat beberapa metode dalam menguji akurasi hasil *association rule* salah satunya yaitu *Lift Ratio / Improvement Ratio*. *Lift Ratio* adalah parameter penting selain *support* dan *confidence* dalam *association rule*. *Lift Ratio* mengukur seberapa penting *rule* yang telah terbentuk berdasarkan nilai *support* dan nilai *confidence*. *Lift Ratio* merupakan nilai yang menunjukkan kevalidan proses transaksi dan memberikan informasi apakah barang A dibeli bersamaan dengan barang B. *Lift Ratio* adalah perbandingan antara

confidence sebuah aturan dengan nilai *benchmark confidence*. *Benchmark confidence* adalah perbandingan antara jumlah semua item *consequent* terhadap total transaksi. Rumus adalah sebagai berikut :

$$Lift\ Ratio = \frac{confidence\ AB}{benchmark\ confidence} \times 100\% \quad \dots persamaan\ 3$$

Sebuah transaksi dikatakan valid jika mempunyai nilai *Lift Ratio / Improvement Ratio* lebih dari 1, yang berarti bahwa dalam transaksi tersebut, barang A dan B benar-benar dibeli secara bersamaan.

BAB 3. METODOLOGI PENELITIAN

Bab ini menjelaskan tentang metode yang digunakan dalam merancang dan membangun sistem yaitu metode pengembangan.

3.1 Jenis Penelitian

Jenis penelitian yang digunakan pada penelitian ini merupakan jenis penelitian pengembangan, karena tujuan penelitian adalah membangun sebuah sistem informasi. Penelitian pengembangan yang bertujuan untuk membuat dan mengembangkan suatu produk yang efektif untuk digunakan. Penelitian ini bukan jenis penelitian yang di tunjukkan untuk menemukan teori atau menguji kebenaran dari suatu teori dalam bentuk eksperimen.

3.2 Tempat dan Waktu Penelitian

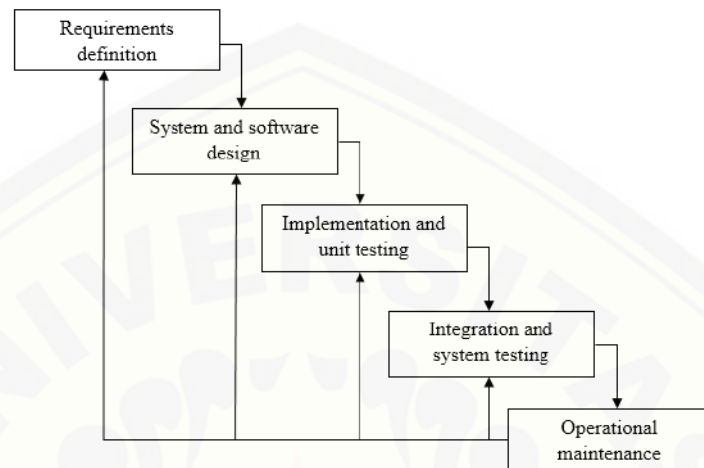
Tempat yang dilaksanakan untuk penelitian adalah minimarket Adien mart Jember. Waktu penelitian dilakukan selama 11 (sebelas) bulan, dimulai pada bulan Juni 2016 sampai dengan bulan Mei 2017.

3.3 Tahapan Penelitian

Penelitian ini akan dilakukan dengan *Software Development Life Cycle (SDLC) waterfall* yang dibagi menjadi beberapa tahapan, yaitu analisis kebutuhan, desain sistem, implementasi, pengujian dan pemeliharaan.

Waterfall merupakan salah satu model proses perangkat lunak yang mengambil kegiatan proses dasar seperti spesifikasi, pengembangan, validasi dan evaluasi dengan mempresentasikan sebagai fase-fase proses yang berbeda seperti analisis dan definisi persyaratan, perancangan perangkat lunak, implementasi dan unit testing, integrasi

dan pengujian serta operasi dan pemeliharaan (Sommerville, 2011). Tahapan penelitian seperti pada Gambar 3.1 berikut:



Gambar 3.1 Tahapan Model *Waterfall*

(Sommerville, 2011)

3.3.1 *Requirement* (Analisis Kebutuhan)

Tahapan pertama pada proses perancangan sistem adalah tahap analisis kebutuhan. Pada tahap ini dilakukan penelitian mencari permasalahan yang ada untuk dapat dianalisis kebutuhan yang diperlukan sebagai solusi dari permasalahan yang muncul. Data-data yang telah didapat kemudian dikelompokkan menjadi kebutuhan fungsional dan kebutuhan non-fungsional. teknik Analisis kebutuhan merupakan tahap untuk mengumpulkan data, informasi, serta mencari kebutuhan fungsional dan non fungsional sistem. Pada tahap ini, peneliti mencari permasalahan yang ada untuk dapat dianalisis kebutuhan yang diperlukan, sebagai solusi dari permasalahan yang muncul. Data-data yang telah didapat kemudian dikelompokkan menjadi kebutuhan fungsional dan non-fungsional. Teknik pengumpulan data yang dilakukan antara lain sebagai berikut:

1. Studi literature, jurnal, serta mencari informasi menggunakan media internet tentang *bundling* barang sebagai strategi promosi penjualan untuk meningkatkan penjualan barang kurang laris
2. Melakukan studi penggunaan library berbasis bahasa pemrograman *Page Hyper Text Pre-Processor* (PHP) dengan metode algoritma *Apriori*.
3. Melakukan Wawancara kepada pemilik minimarket dan pegawai tentang bagaimana selama ini sistem penjualan dan strategi apa saja yang sudah digunakan untuk meningkatkan penjualan.

3.3.2 System and Software Desain

Tahap desain merupakan proses yang berfokus pada struktur arsitektur perangkat lunak, representasi *interface* dan detail algoritma. Tahapan ini akan menghasilkan dokumen yang disebut *software requirement* langkah ini dilakukan oleh pengembang sistem. Perancangan sistem dengan konsep perancangan berorientasi objek, dengan menggunakan tools *UML (Unified Modeling Language) Visual Paradigm*. Diagram-diagram yang digunakan yang digunakan sbb:

1. *Business process* merupakan diagram yang menggambarkan proses dari sebuah sistem yang meliputi *input*, *output*, dan *goal* yang merupakan tujuan dari sebuah sistem yang dibangun.
2. *Use Case Diagram* menggambarkan fungsionalitas sistem dan hubungan antara sistem dengan aktor berkaitan dengan fungsi atau tugas yang dilakukan oleh aktor. *Use Case Diagram* juga menggambarkan hak akses dari pengguna.
3. *Scenario* menjelaskan alur sistem dan keadaan yang akan terjadi ketika terjadi suatu event tertentu.
4. *Activity Diagram* mendeskripsikan aktifitas yang dibentuk dalam suatu operasi
5. *Sequence Diagram* digunakan untuk menunjukkan rangkaian pesan yang dikirim dan diterima antar *object*.

6. *Class Diagram* digunakan untuk menggambarkan relasi antar objek dan struktur sematik yang umum.
7. *Entity Relationship Diagram* digunakan untuk menggambarkan relasi data yang terdapat dalam *database*.

3.3.3 Implementation and UnitTesting

Pada tahap ini mengimplementasikan desain diagram yang telah dibuat pada tahap desain dalam bentuk baris kode program. Perangkat lunak yang akan dibangun ini menggunakan bahasa pemrograman *PHP* dengan *framework codeigniter* dengan editor *Sublime* yang terintegrasi dengan *DBMS MySQL*.

3.3.4 Integration and System Testing

Pengujian digunakan untuk mengetahui sejauh mana sistem ini dapat berjalan. Testing berfungsi untuk mengetahui apakah sistem ini dapat berfungsi dengan baik sesuai dengan yang diharapkan. Serta untuk mengetahui letak kekurangan yang ada pada sistem ini. Terdapat dua metode yang digunakan untuk pengujian ini yakni :

a. *White box testing*

White box testing merupakan cara pengujian dengan melihat modul yang telah dibuat dan program-program yang ada. Pengujian ini dilakukan oleh (*develeoper*) pembuat program. Jika ada modul yang menghasilkan output yang tidak sesuai, maka baris-baris program, variabel dan parameter yang terlibat pada unit tersebut satu persatu akan dicek dan diperbaiki, kemudian di compile ulang. (Pressman, 2001)

b. *Black box testing*

Black box testing merupakan metode pengujian perangkat lunak yang memeriksa fungsionalitas dari aplikasi yang berkaitan dengan struktur internal atau kerja. Metode ini memfokuskan pada keperluan fungsionalitas dari software. (Pressman, 2001)

c. Uji Tingkat Akurasi Hasil *Association Rule*

Pada tahap ini *frequent itemsets* yang di hasilkan akan di uji tingkat akurasinya. Apakah *frequent itemsets* yang digunakan memiliki manfaat atau tidak. Pengujian tingkat akurasi menggunakan lift ratio. *Frequent itemsets* dapat dikatakan valid jika mempunyai *lift ratio/improvement ratio* lebih dari 1.

3.3.5 *Operasional Maintenance*

Pemeliharaan merupakan proses perawatan sistem digunakan oleh pengguna. Pemeliharaan dilakukan dengan mengecek kinerja sistem secara berkala. Pengecekan dilakukan apakah kinerja sistem masih berjalan dengan baik dan memperbaiki jika terdapat kerusakan. *Operasional Maintenance*

BAB 4. PENGEMBANGAN SISTEM

Bab ini menguraikan mengenai analisis kebutuhan, desain, implementasi, dan pengujian sistem yang digunakan dalam proses pengembangan modul *bundling* menggunakan association rule algoritma apriori pada sistem informasi penjualan adien mart sebagai strategi promosi penjualan barang kurang laris. Tahapan analisis hingga pengujian dilakukan sesuai dengan metode pengembangan *waterfall*.

4.1 Analisis Kebutuhan

Berdasarkan metode pengembangan sistem model *waterfall*, tahapan awal yang dilakukan adalah tahap analisis. Pada tahap analisis ini menggunakan teknik pengumpulan data yang di jelaskan pada Sub subbab 3.3.1. Analisa kebutuhan data pada penelitian ini menghasilkan data barang, data kategori, data subkategori, data supplier, data user, data transaksi penjualan, teori bundling, dan metode algoritma *Apriori*. Berdasarkan hasil wawancara yang dilakukan, minimarket Adien mart mempunyai kendala untuk penjualan beberapa barang yang tingkat penjualannya rendah untuk itu diperlukan solusi mengatasi kendala tersebut.

Pada penelitian ini dilakukan pengembangan modul *bundling* untuk mengatasi kendala yang dialami oleh Adien mart. *Bundling* barang dapat dimanfaatkan dengan menggunakan data transaksi penjualan. Data transaksi penjualan dapat diolah untuk mendapatkan data pola penjualan yaitu berupa barang yang sering di beli bersamaan (*frequent itemsets*). *Frequent itemsets* di dapat dengan menerapkan metode algoritma *apriori*. Setelah *frequent itemsets* ditemukan, misalkan jika konsumen membeli telur maka membeli indomie kuah soto, jadi dari *frequent itemsets* tersebut dapat dilakukan *bundling* yaitu dengan mengganti salah satu barang pada *frequent itemsets* yang memiliki subkategori sama dengan tingkat penjualan rendah atau kurang laris. Misalkan barang kurang laris pada subkategori mie kuah adalah indomie kuah rasa

bawang, sehingga akan muncul rekomendasi barang yang akan *dibundling* yaitu telur dan indomie kuah rasa bawang.

4.1.1 SOP (*Statement of purpose*)

Pengembangan modul *bundling* menggunakan association rule algoritma apriori pada sistem informasi penjualan adien mart sebagai strategi promosi penjualan barang kurang laris merupakan sistem yang mengelola data transaksi penjualan yang akan menghasilkan data pola penjualan. Pola data penjualan yang di hasilkan dapat di gunakan sebagai acuan strategi penjualan barang kurang laris. Tujuan pengembangan sistem adalah untuk membantu menentukan barang apa saja yang akan di *bundling* berdasarkan data pola penjualan yang sudah dihasilkan. Terdapat 2 hak akses sistem yaitu admin dan pegawai. Sistem dapat memanajemen data user, memanajemen data barang, memanajemen supplier, memanajemen data kategori, memanajemen data subkategori, memanajemen stok barang, melakukan transaksi penjualan, mengetahui data transaksi penjualan, menambah data pola, mengetahui data barang dan hasil rekomendasi promosi. Sistem dapat mencatat riwayat aktivitas yang digunakan oleh setiap pengguna. Sistem dibangun berbasis *web* untuk memudahkan komunikasi data antar pengguna.

4.1.2 Pengumpulan Data

Pengumpulan data menggunakan data hasil wawancara dengan pemilik minimarket Adien mart yang bertempat di minimarket Adienmart jalan Cokroaminoto, Kalisat, Jember, Jawa Timur. Hasil dari wawancara yang dilakukan dapat dilihat pada Lampiran G. Hasil Wawancara. Data hasil wawancara tersebut dapat dianalisis yang menghasilkan kebutuhan fungsional dan kebutuhan non fungsional yaitu sebagai berikut:

a. Kebutuhan Fungsional

Kebutuhan fungsional sistem berisi fitur-fitur inti yang harus dipenuhi dalam sistem agar sistem berfungsi sesuai dengan tujuan dan kebutuhan pengguna terhadap sistem. Kebutuhan fungsional sistem informasi penjualan Adien mart yaitu:

1. Sistem mampu menghasilkan data pola penjualan dengan menggunakan algoritma apriori.
2. Sistem mampu merekomendasikan barang kurang laris untuk di *bundling*
3. Sistem mampu mengelola data barang (*view, input, edit, status*).
4. Sistem mampu mengelola data user (*view, input, edit, status*).
5. Sistem mampu mengelola data supplier (*view, input, edit, status*).
6. Sistem mampu mengelola data kategori (*view, input, edit, status*).
7. Sistem mampu mengelola data subkategori (*view, input, edit, status*).
8. Sistem mampu melakukan transaksi penjualan.
9. Sistem mampu menambah data pola.
10. Sistem mampu mengelola stok barang.

b. Kebutuhan Non-Fungsional

Kebutuhan non-fungsional merupakan fitur-fitur yang dimiliki untuk mendukung sistem dalam memenuhi fungsionalitasnya. Kebutuhan non-fungsional sistem informasi penjualan adien mart yaitu:

1. Sistem memiliki tampilan dan bahasa komunikasi sistem yang mudah dimengerti oleh pengguna agar memberikan kenyamanan pemakaian dan memudahkan pengoperasian.
2. Sistem dapat digunakan oleh banyak pengguna pada saat yang bersamaan pada komputer yang berbeda.
3. Sistem menggunakan *username* dan *password* untuk autentikasi akses pengguna pada sistem.

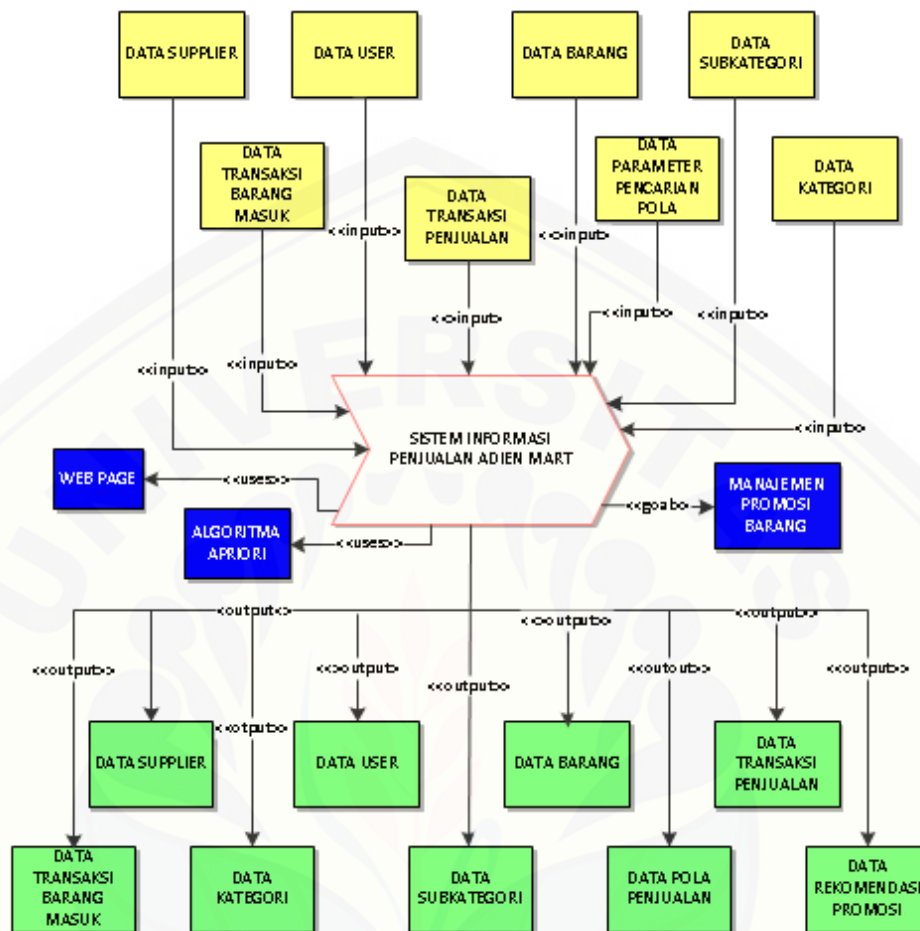
4.2 Desain System

Tahapan yang dilakukan setelah melakukan analisis kebutuhan sistem yaitu tahap perencanaan pembangunan sistem yang digambarkan dengan desain sistem. Desain pengembangan modul *bundling* menggunakan *association rule* algoritma apriori pada system informasi penjualan adien mart sebagai strategi promosi penjualan barang kurang laris ini meliputi *business process*, *use case diagram*, *use case* skenario, *activity diagram*, *sequence diagram*, *class diagram*, dan *entity relationship diagram*.

4.2.1. Bussiness Process

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

6. *Input* : Data yang dimasukkan ke dalam sistem
7. *Output* : Data yang dihasilkan oleh sistem
8. *Goal* : Tujuan dibangun suatu sistem
9. *Uses* : *Platform* yang menjadi basis sistem
10. *Process* : Sistem yang bekerja



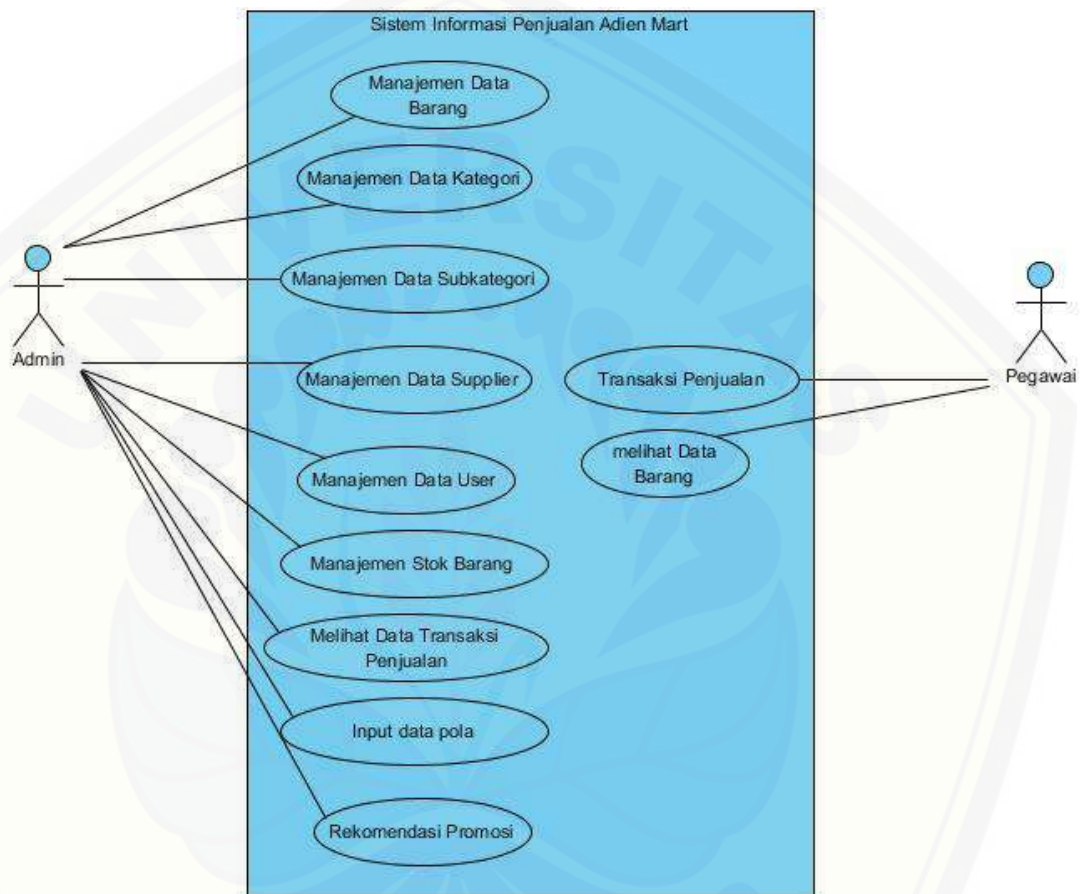
Gambar 4.1 *Business Procces* Sistem Informasi Penjualan Adien Mart

Gambar 4.1 merupakan *business process* dari pengembangan modul *bundling* dengan *association rule* algoritma apriori pada system informasi penjualan Adien mart sebagai strategi promosi penjualan barang kurang laris. *Business process* ini menjelaskan proses *input*, *output*, *goal* dan *uses* yang diaplikasikan pada sistem.

4.2.2. Use Case Diagram

Use case diagram merupakan pemodelan yang digunakan untuk menggambarkan interaksi antara aktor dengan sistem yang dibangun. Melalui *use case* diagram dapat

diketahui interaksi yang dapat dilakukan aktor 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 Sistem Informasi Penjualan Adien Mart

Gambar 4.2 menunjukkan *usecase diagram* pengembangan modul *bundling* menggunakan *association rule* algoritma apriori pada sistem informasi penjualan adien mart sebagai strategi promosi penjualan barang kurang laris yang terdiri dari dua aktor dengan sebelas *use case*.

Berdasarkan *use case diagram* pada Gambar 4.2 terdapat dua aktor atau pengguna, yaitu admin dan Pegawai. Adapun deskripsi dari masing-masing aktor dapat dilihat pada Tabel 4.1.

Tabel 4.1 Deskripsi Aktor

No.	Aktor	Deskripsi
1.	Admin	Aktor admin merupakan aktor yang memiliki hak akses untuk mengatur proses dasar sistem informasi secara penuh. Aktor ini dapat melakukan login, manajemen data barang, manajemen data kategori, manajemen data subkategori, manajemen data supplier, manajemen data user, manajemen transaksi barang, input stok barang, melihat data transaksi, rekomendasi promosi dan dapat melihat data transaksi penjualan.
2.	Pegawai	Aktor pegawai merupakan aktor yang memiliki hak akses untuk login, manajemen transaksi penjualan, melihat data barang dan melihat data transaksi penjualan.

Selain memiliki dua aktor, dalam *use case diagram* juga terdapat dua belas *use case*. Deskripsi dari *use case* terdapat pada pada tabel 4.2

Tabel 4.2 Deskripsi Use case

No.	Usecase	Penjelasan
1.	Manajemen Data Barang	Merupakan usecase yang menggambarkan proses menampilkan, menambah, mengubah status status dan mengubah data barang.
2.	Manajemen Data Kategori	Merupakan usecase yang menggambarkan proses menampilkan, menambah, mengubah status dan mengubah data kategori.
3.	Manajemen Data Subkategori	Merupakan usecase yang menggambarkan proses menampilkan, menambah, mengubah status dan mengubah data subkategori
4.	Manajemen Data Supplier	Merupakan usecase yang menggambarkan proses menampilkan, menambah, mengubah status, dan mengubah data supplier.
5.	Manajemen Data User	Merupakan usecase yang menggambarkan proses menampilkan, menambah, mengubah status, dan mengubah data user.
6.	Manajemen Stok Barang	Merupakan usecase yang menggambarkan proses menampilkan, menambah stok barang,
7.	View Data Transaksi Penjualan	Merupakan usecase yang digunakan untuk melihat data transaksi penjualan.
8.	Input Data Pola	Input data pola merupakan usecase yang menggambarkan proses tambah data pola
9.	Rekomendasi Promosi	Rekomendasi Promosi merupakan usecase yang digunakan untuk manajemen rekomendasi promosi.
10.	Transaksi Penjualan	Manajemen transaksi penjualan merupakan

No.	Usecase	Penjelasan
		usecase yang mengelola dan menginputkan data transaksi.
11.	View Data Barang	Merupakan usecase yang digunakan untuk melihat data barang.

4.2.3. Use Case Skenario

Usecase skenario adalah dokumentasi terhadap kebutuhan fungsional sistem yang menjelaskan alur yang menggambarkan aksi aktor dan reaksi sistem. *Use case* skenario pengembangan modul *bundling* menggunakan *assosiation rule* algoritma apriori pada sistem informasi penjualan adien mart sebagai strategi promosi penjualan barang kurang laris adalah sebagai berikut:

1. Skenario Manajemen Data Barang

Penjelasan urutan aksi aktor untuk manajemen data barang meliputi (*view, input, edit*) dan reaksi sistem pada skenario normal dan skenario alternatif terdapat pada Tabel 4.3.

Tabel 4.3 *Usecase* Skenario Manajemen Data Barang

Nomor <i>Usecase</i>	USC 01
Nama	Manajemen Data Barang
Aktor	Admin
<i>Pre Condition</i>	Admin memilih menu data barang
<i>Post Condition</i>	Admin berhasil melihat, menambah, merubah status dan merubah data barang
SKENARIO UTAMA MELIHAT DATA BARANG	
Aktor	Sistem
1. Memilih menu manajemen barang	
2. Memilih submenu data barang	
	3. Mengambil data barang dari tabel data barang berupa kode barang, nama barang, nama supplier, harga beli, harga jual, stok dan status di database.dengan menggunakan <i>fuction</i> databarang()
	4. Menampilkan data barang berupa

kode barang, nama supplier, harga beli, harga jual, stok, status, tombol tambah data, tombol edit, tombol nonaktif dan tombol aktif pada halaman data barang menggunakan *function* tampilbarang().

SKENARIO UTAMA MENAMBAH DATA BARANG

Aktor	Sistem
5. Menekan tombol “Tambah Data”	6. Menampilkan <i>form</i> isian barang berupa kode barang, kategori, subkategori, supplier, nama barang, harga beli, harga jual dan tombol kembali menggunakan <i>function</i> formdatabarang()
7. Mengisi isian <i>form</i> data barang	
8. Menekan tombol “Simpan”	9. Memeriksa masukan data barang
	10. Menyimpan isian data <i>form</i> barang ke tabel data barang di <i>database</i> , menggunakan <i>function</i> inputbarang() dengan atribut kodebarang, kodekategori, kodesubkategori, kodesupplier, namabarang, hargabeli, hargajual, kodepengguna
	11. Menampilkan pesan “data berhasil ditambah”
	12. Mengambil data barang dari table data barang berupa kode barang, namabarang, nama supplier, harga beli, harga jual, stok dan status di <i>database</i> menggunakan <i>function</i> databarang()
	13. Menampilkan data barang berupa kode barang, nama supplier, harga beli, harga jual, stok, status, tombol tambah data, tombol edit, tombol nonaktif dan tombol aktif pada halaman data barang menggunakan <i>function</i>

tampilbarang()	
SKENARIO UTAMA TAMBAH DATA KEMBALI KE HALAMAN BARANG	
Aktor	Sistem
8. Menekan tombol “Kembali”	9. Mengambil data barang dari table data barang berupa kode barang, nama barang, nama supplier, harga beli, harga jual, stok dan status di database menggunakan <i>function</i> databarang().
	10. Menampilkan data barang berupa kode barang, nama supplier, harga beli, harga jual, stok, status, tombol tambah data, tombol edit, tombol nonaktif dan tombol aktif pada halaman data barang menggunakan <i>function</i> tampilbarang()
SKENARIO UTAMA MERUBAH STATUS AKTIF BARANG	
Aktor	Sistem
5. Menekan tombol “non aktif”	6. Merubah status pada tabel data barang database, menggunakan <i>function</i> formeditstatus() dengan atribut id
	7. Menampilkan pesan “data berhasil diubah”
	8. Mengambil data status dari tabel data barang di database menggunakan <i>function</i> tampileditstatus() dengan atribut id
	9. Menampilkan data barang berupa kode barang, nama supplier, harga beli, harga jual, stok, status, tombol tambah data, tombol edit, tombol nonaktif dan tombol aktif pada halaman data barang menggunakan <i>function</i> tampilbarang().
SKENARIO UTAMA MERUBAH STATUS NONAKTIF BARANG	
Aktor	Sistem
5. Menekan tombol “Aktif”	

	6. Merubah status pada tabel data barang database, menggunakan <i>fuction</i> <i>formeditstatusaktif()</i> dengan atribut <i>id</i>
	7. Menampilkan pesan “data berhasil diubah”
	8. Mengambil data status dari tabel data barang di database menggunakan, <i>fuction</i> <i>tampileditstatusaktif()</i> dengan atribut <i>id</i>
	9. Menampilkan data barang berupa kode barang, nama supplier, harga beli, harga jual, stok, status, tombol tambah data, tombol edit, tombol nonaktif dan tombol aktif pada halaman data barang menggunakan <i>fuction</i> <i>tampilbarang()</i> .

SKENARIO UTAMA MERUBAH DATA BARANG

Aktor	Sistem
5. Menekan tombol “ <i>Edit</i> ” pada baris data yang akan diubah	
	6. Mengambil data barang berupa kode barang, kategori, subkategori, supplier, nama barang, harga beli dan harga jual menggunakan <i>fuction</i> <i>formdatabarangedit()</i>
	7. Menampilkan <i>form</i> isian <i>edit</i> data barang berupa kode barang, kategori, subkategori, supplier, nama barang, harga beli, harga jual dan tombol kembali
8. Mengubah isian <i>form edit</i> barang	
9. Menekan tombol “Simpan”	
	10. Memeriksa masukan data barang
	11. Menyimpan isian <i>edit form</i> barang ke tabel data barang di <i>database</i> , menggunakan <i>fuction</i> <i>editbarang()</i> dengan atribut <i>kodebarang</i> , <i>kodekategori</i> , <i>kodesubkategori</i> , <i>kodesupplier</i> , <i>namabarang</i> , <i>hargabeli</i> , <i>hargajual</i>

	12. Menampilkan pesan “data berhasil diubah”
	13. Mengambil data barang dari table data barang berupa kode barang, namabarang, nama supplier, harga beli, harga jual, stok dan status di database.
	14. Menampilkan data barang berupa kode barang, nama supplier, harga beli, harga jual, stok, status, tombol tambah data, tombol edit, tombol nonaktif dan tombol aktif pada halaman data barang

SKENARIO UTAMA MERUBAH DATA KEMBALI KE HALAMAN DATA BARANG

Aktor	Sistem
9. Menekan tombol “Kembali”	
	10. Mengambil data barang dari table data barang berupa kode barang, nama barang, nama supplier, harga beli, harga jual, stok dan status di database.
	11. Menampilkan data barang berupa kode barang, nama supplier, harga beli, harga jual, stok, status, tombol tambah data, tombol edit, tombol nonaktif dan tombol aktif pada halaman data barang

SKENARIO ALTERNATIF MENAMBAH DATA BARANG

Jika aktor tidak mengisi *form* barang secara lengkap

Aktor	Sistem
7a. Mengisi isian <i>form</i> barang tidak lengkap	
8a. Menekan tombol “Simpan”	
	9a. Memeriksa masukan data barang
	10a. Menampilkan pesan dengan menunjuk lokasi isian yang belum diisi pada <i>form</i> barang. Pesan akan tampil di <i>field</i> yang kosong

SKENARIO ALTERNATIF MERUBAH DATA BARANG

Jika aktor tidak mengisi *form* edit barang secara lengkap

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

9a. Mengisi isian <i>form</i> edit barang tidak lengkap	
10a. Menekan tombol “Simpan”	
	11a. Memeriksa masukan data barang
	12a. Menampilkan pesan dengan menunjuk lokasi isian yang belum diisi pada <i>form</i> edit barang. Pesan akan tampil di <i>field</i> yang kosong

2. Skenario Manajemen Data Kategori

Penjelasan urutan aksi aktor untuk manajemen data kategori meliputi (*view* dan *input*, *edit*) dan reaksi sistem pada skenario normal dan skenario alternatif terdapat pada Tabel 4.4.

Tabel 4.4 *Usecase* Skenario Manajemen Data Kategori

Nomor <i>Usecase</i>	USC 02
Nama	Manajemen Data Kategori
Aktor	Admin
<i>Pre Condition</i>	Admin memilih menu data kategori
<i>Post Condition</i>	Admin berhasil melihat, menambah, merubah status dan merubah data kategori

SKENARIO UTAMA MELIHAT DATA KATEGORI

Aktor	Sistem
1. Memilih menu manajemen barang	
2. Memilih submenu data kategori	
	3. Mengambil data kategori dari tabel datakategori berupa kode kategori, kategori dan status di database.dengan menggunakan <i>fuction</i> datakategori()
	4. Menampilkan data kategori berupa kode kategori, kategori dan status, tombol tambah data, tombol edit, tombol nonaktif dan tombol aktif pada halaman kategori <i>fuction</i> tampilkategori()

SKENARIO UTAMA MENAMBAH DATA KATEGORI

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

5. Menekan tombol “Tambah Data”	6. Menampilkan <i>form</i> isian kategori berupa kode kategori, kategori dan tombol kembali menggunakan <i>function</i> <i>formdatakategori()</i>
7. Mengisi isian <i>form</i> kategori	
8. Menekan tombol “Simpan”	9. Memeriksa masukan data kategori
	10. Menyimpan isian data <i>form</i> kategori ke tabel <i>datakategori</i> di <i>database</i> , menggunakan <i>function</i> <i>inputkategori()</i> dengan atribut <i>kodekategori</i> , <i>kategori</i> , <i>kodepengguna</i>
	11. Menampilkan pesan “data berhasil ditambah”
	12. Mengambil data kategori dari tabel <i>datakategori</i> berupa kode kategori, kategori dan status di <i>database</i> menggunakan <i>function</i> <i>datakategori()</i>
	13. Menampilkan data kategori berupa kode kategori, kategori dan status, tombol tambah data, tombol edit, tombol nonaktif dan tombol aktif pada halaman kategori menggunakan <i>function</i> <i>tampilkategori()</i>

SKENARIO UTAMA TAMBAH DATA KEMBALI KE HALAMAN KATEGORI

Aktor	Sistem
8. Menekan tombol “Kembali”	9. Mengambil data kategori dari tabel <i>datakategori</i> berupa kode kategori, kategori dan status di <i>database</i> menggunakan <i>function</i> <i>datakategori()</i>
	10. Menampilkan data kategori berupa kode kategori, kategori dan status, tombol tambah data, tombol edit, tombol nonaktif dan tombol aktif pada halaman kategori

menggunakan <i>function</i> tampilkategori()	
SKENARIO UTAMA MERUBAH STATUS AKTIF KATEGORI	
Aktor	Sistem
5. Menekan tombol “non aktif”	6. Merubah status pada tabel datakategori di database, menggunakan <i>function</i> kategoristatus() dengan atribut id
	7. Menampilkan pesan “data berhasil diubah”
	8. Mengambil data status dari tabel data kategori di database menggunakan, <i>function</i> kategoristatus() dengan atribut id
	9. Menampilkan data kategori berupa kode kategori, kategori dan status, tombol tambah data, tombol edit, tombol nonaktif dan tombol aktif pada halaman kategori menggunakan <i>function</i> tampilkategori()
SKENARIO UTAMA MERUBAH STATUS NONAKTIF KATEGORI	
Aktor	Sistem
5. Menekan tombol “Aktif”	6. Merubah status pada tabel datakategori di database, menggunakan <i>function</i> kategoristatusaktif() dengan atribut id
	7. Menampilkan pesan “data berhasil diubah”
	8. Mengambil data status dari tabel data kategori di database menggunakan, <i>function</i> kategoristatusaktif() dengan atribut id
	9. Menampilkan data kategori berupa kode kategori, kategori dan status, tombol tambah data, tombol edit, tombol nonaktif dan tombol aktif pada halaman kategori

		menggunakan tampilkategori()	<i>function</i>
SKENARIO UTAMA MERUBAH DATA KATEGORI			
Aktor		Sistem	
5. Menekan tombol “ <i>Edit</i> ” pada baris data yang akan diubah		6. Mengambil data kategori berupa kode kategori dan kategori menggunakan <i>function</i> <code>formdatakategoriedit()</code> dengan atribut <code>kodekategori</code>	
8. Mengubah isian <i>form edit</i> kategori		7. Menampilkan <i>form</i> isian <i>edit</i> data kategori berupa kode kategori, kategori dan tombol kembali dengan <i>function</i> <code>tampilkategoriedit()</code> dengan atribut <code>kodekategori</code> .	
9. Menekan tombol “Simpan”		10. Memeriksa masukan data kategori	
		11. Menyimpan isian <i>edit form</i> kategori ke tabel data kategori di <i>database</i> , menggunakan <i>function</i> <code>editkategori()</code> dengan atribut <code>kodekategori</code> , <code>kategori</code>	
		12. Menampilkan pesan “data berhasil diubah”	
		13. Mengambil data kategori dari tabel kategori berupa <code>kodekategori</code> , <code>kategori</code> dan <code>status</code> di <i>database</i> menggunakan <i>function</i> <code>datakategori()</code>	
		14. Menampilkan data kategori berupa kode kategori, kategori, status, tombol tambah data, tombol edit, tombol nonaktif dan tombol aktif pada halaman kategori menggunakan <i>function</i> <code>tampilkategori()</code>	
SKENARIO UTAMA MERUBAH DATA KEMBALI KE HALAMAN DATA KATEGORI			
Aktor		Sistem	

9. Menekan tombol “Kembali”	10. Mengambil data kategori dari tabel kategori berupa kodekategori, kategori dan status di database menggunakan <i>function</i> datakategori()
	11. Menampilkan data kategori berupa kode kategori, kategori, status, tombol tambah data, tombol edit, tombol nonaktif dan tombol aktif pada halaman kategori menggunakan <i>function</i> tampilkategori()
SKENARIO ALTERNATIF MENAMBAH DATA KATEGORI	
Jika aktor tidak mengisi <i>form</i> kategori secara lengkap	
Aktor	Sistem
7a. Mengisi isian <i>form</i> kategori tidak lengkap	
8a. Menekan tombol “Simpan”	9a. Memeriksa masukan data kategori
	10a. Menampilkan pesan dengan menunjuk lokasi isian yang belum diisi pada <i>form</i> kategori. Pesan akan tampil di <i>field</i> yang kosong
SKENARIO ALTERNATIF MERUBAH DATA KATEGORI	
Jika aktor tidak mengisi <i>form</i> edit kategori secara lengkap	
Aktor	Sistem
8a. Mengisi isian <i>form</i> edit kategori tidak lengkap	
9a. Menekan tombol “Simpan”	10a. Memeriksa masukan data kategori
	11a. Menampilkan pesan dengan menunjuk lokasi isian yang belum diisi pada <i>form</i> edit kategori. Pesan akan tampil di <i>field</i> yang kosong

3. Login

Penjelasan urutan aksi aktor untuk melakukan *login* dan reaksi sistem pada skenario normal dan skenario alternatif terdapat pada Lampiran A (*Usecase* Skenario).

4. Skenario Manajemen Data Subkategori

Penjelasan urutan aksi aktor untuk manajemen data subkategori (*view, input, edit*) dan reaksi sistem pada skenario normal dan skenario alternatif terdapat pada Lampiran A (*Usecase Skenario*).

5. Skenario Manajemen Data Supplier

Penjelasan urutan aksi aktor untuk manajemen data supplier (*view, input, edit*) dan reaksi sistem pada skenario normal dan skenario alternatif terdapat pada Lampiran A (*Usecase Skenario*).

6. Skenario Manajemen Data User

Penjelasan urutan aksi aktor untuk manajemen data user meliputi (*view, input, edit*) dan reaksi sistem pada skenario normal dan skenario alternatif terdapat pada Lampiran A (*Usecase Skenario*).

7. Skenario Menejemen Stok Barang

Penjelasan urutan aksi aktor untuk manajemen stok barang meliputi (*view, input, edit*) dan reaksi sistem pada skenario normal dan skenario alternatif terdapat pada Lampiran A (*Usecase Skenario*).

8. Skenario Melihat Data Transaksi Penjualan

Penjelasan urutan aksi aktor untuk melihat data transaksi penjualan dan reaksi sistem pada skenario normal dan skenario alternatif terdapat pada Lampiran A (*Usecase Skenario*).

9. Skenario Input data Pola

Penjelasan urutan aksi aktor untuk input data pola dan reaksi sistem pada skenario normal dan skenario alternatif terdapat pada Lampiran A (*Usecase Skenario*).

10. Skenario Rekomendasi Promosi

Penjelasan urutan aksi aktor untuk rekomendasi promosi dan reaksi sistem pada skenario normal dan skenario alternatif dapat terdapat Lampiran A (*Usecase Skenario*).

11. Skenario Transaksi Penjualan

Penjelasan urutan aksi aktor untuk transaksi penjualan dan reaksi sistem pada skenario normal dan skenario alternatif terdapat pada Lampiran A (*Usecase Skenario*).

12. Skenario Melihat Data Barang

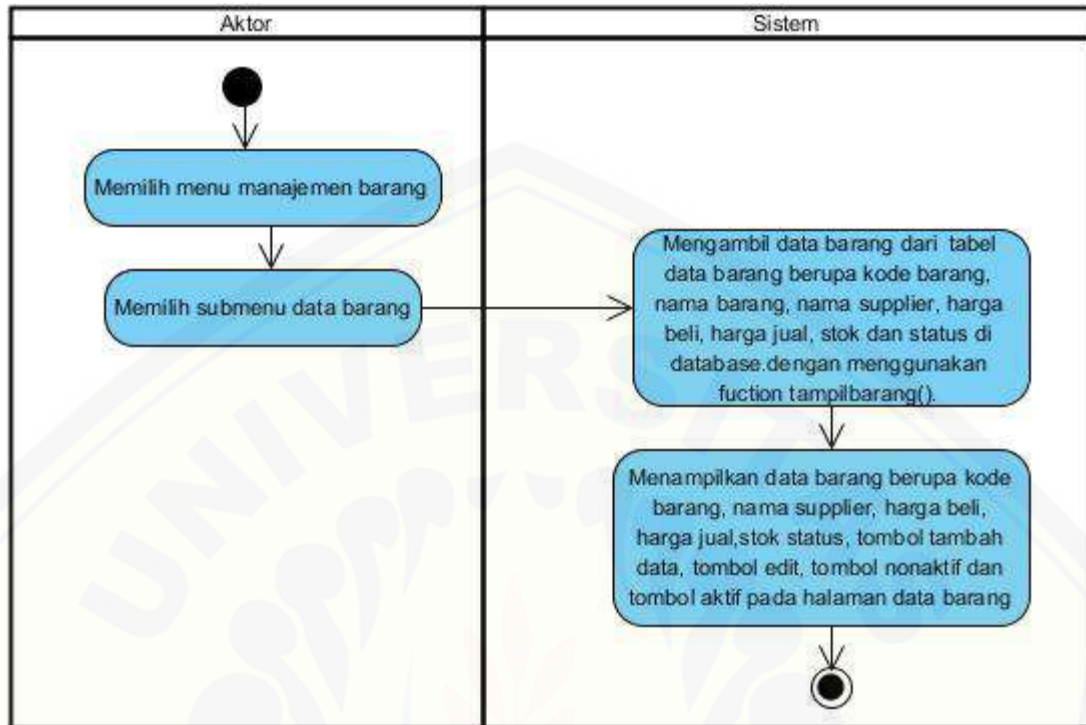
Penjelasan urutan aksi actor untuk melihat data barang dan reaksi sistem pada skenario normal dan skenario alternatif terdapat pada Lampiran A (Usecase Skenario).

4.2.4. Activity Diagram

Activity diagram dokumentasi desain yang menggambarkan aliran aktivitas dalam pengembangan modul *bundling* menggunakan *assosiation rule* algoritma apriori pada sistem informasi penjualan Adien mart sebagai strategi promosi penjualan barang kurang laris yang akan dibangun. Sistem ini terdapat tiga puluh empat *activity* diagram yaitu sebagai berikut:

1. Activity Diagram Manajemen Data Barang (*view*)

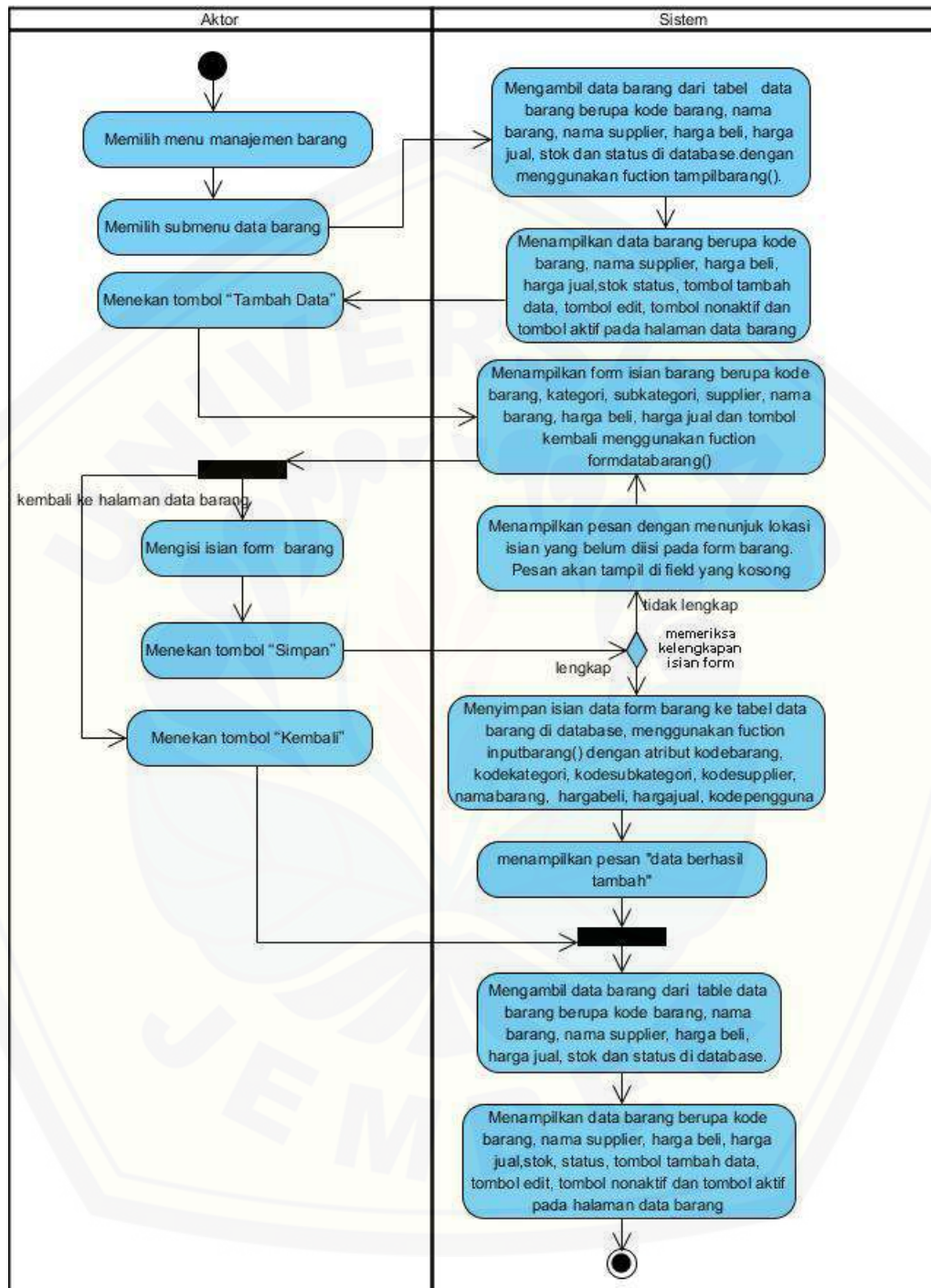
Activity diagram manajemen data barang (*view*) menggambarkan alur aktivitas proses melihat data barang yang dilakukan oleh admin. *Activity* diagram ini menjelaskan tentang alur yang harus dilakukan admin untuk melihat data barang dan reaksi sistem ketika admin memberikan aksi terhadap sistem. *Activity* diagram manajemen data barang (*view*) terdapat pada Gambar 4.3.



Gambar 4.3 *Activity Diagram* Manajemen Data Barang (*view*)

2. *Activity Diagram* Manajemen Data Barang (*input*)

Activity diagram manajemen data barang (*input*) menggambarkan alur aktivitas proses menambah data barang yang dilakukan oleh admin. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan admin untuk menambah data barang dan reaksi sistem ketika admin memberikan aksi terhadap sistem. *Activity diagram* manajemen data barang (*input*) terdapat pada Gambar 4.4.

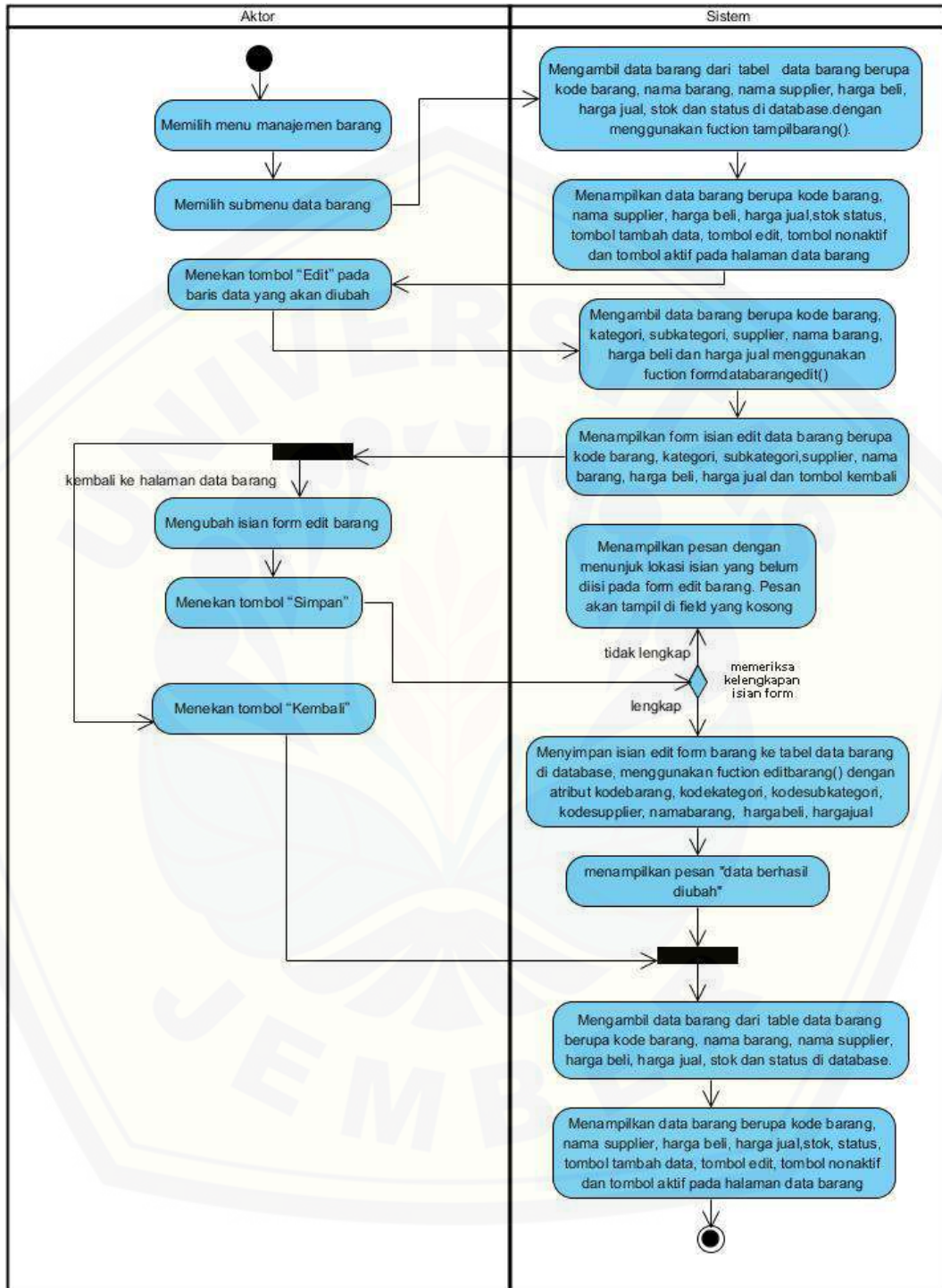


Gambar 4.4 Activity Diagram Manajemen Data Barang (input)

3. Activity Diagram Manajemen Data Barang (*edit*)

Activity diagram manajemen data barang (*edit*) menggambarkan alur aktivitas proses merubah data barang yang dilakukan oleh admin. *Activity* diagram ini menjelaskan tentang alur yang harus dilakukan admin untuk merubah data barang dan reaksi sistem ketika admin memberikan aksi terhadap sistem. *Activity* diagram manajemen data barang (*edit*) terdapat pada Gambar 4.5.

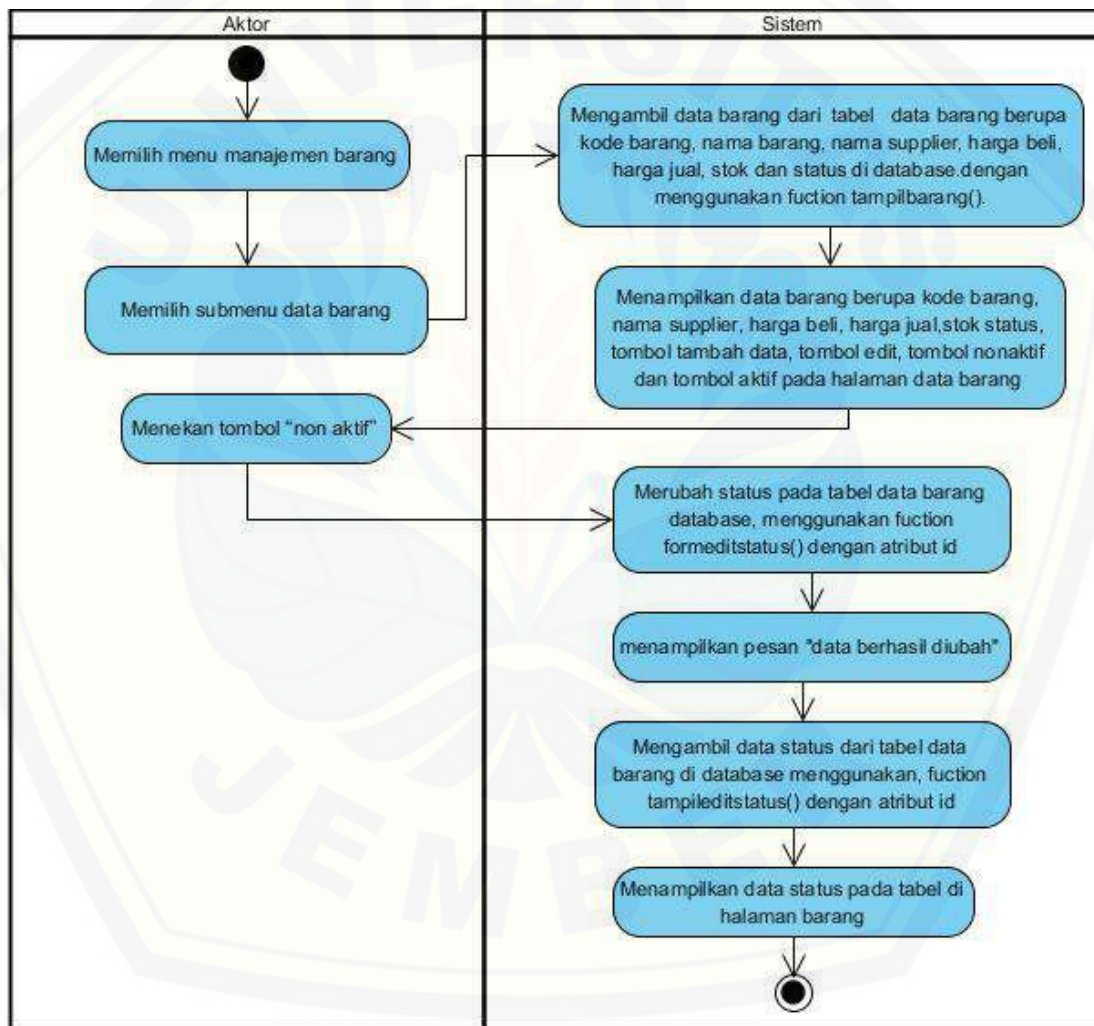




Gambar 4.5 Activity Diagram Manajemen Data Barang (edit)

4. Activity Diagram Manajemen Data Barang (*edit status aktif*)

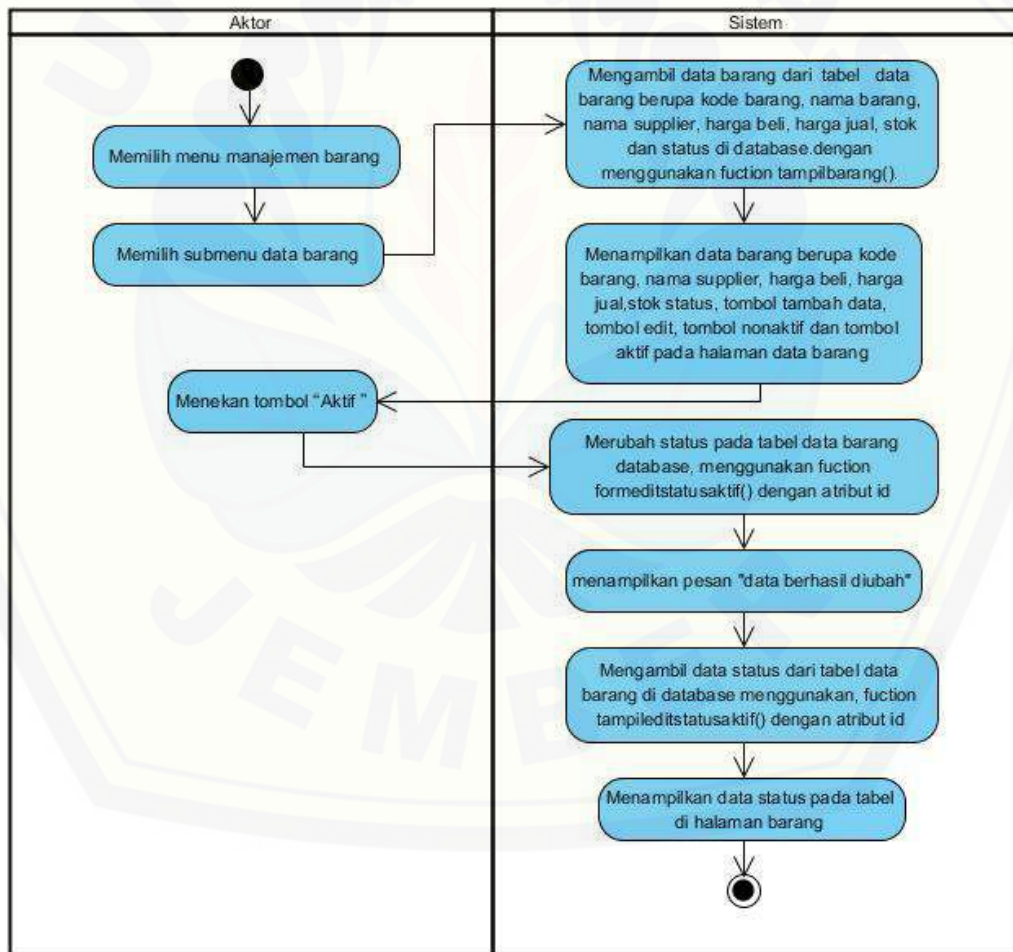
Activity diagram manajemen data barang (*edit status aktif*) menggambarkan alur aktivitas proses merubah status data barang yang dilakukan oleh admin. Activity diagram ini menjelaskan tentang alur yang harus dilakukan admin untuk merubah status data barang dan reaksi sistem ketika admin memberikan aksi terhadap sistem. Activity diagram manajemen data barang (*edit status aktif*) terdapat pada Gambar 4.6.



Gambar 4.6 Activity Diagram Manajemen Data Barang (*edit status aktif*)

5. Activity Diagram Manajemen Data Barang (*edit status non aktif*)

Activity diagram manajemen data barang (*edit status non aktif*) menggambarkan alur aktivitas proses mengubah status data barang yang dilakukan oleh admin. Activity diagram ini menjelaskan tentang alur yang harus dilakukan admin untuk mengubah status data barang dan reaksi sistem ketika admin memberikan aksi terhadap sistem. Activity diagram manajemen data barang (*edit status non aktif*) terdapat pada Gambar 4.7.



Gambar 4.7 Activity Diagram Manajemen Data Barang (*edit status non aktif*)

6. *Activity Diagram Manajemen Data Kategori (view)*

Activity diagram manajemen data kategori (view) menggambarkan alur aktivitas proses melihat data kategori yang dilakukan oleh admin. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan admin untuk melihat data kategori dan reaksi sistem ketika admin memberikan aksi terhadap sistem. *Activity diagram manajemen data kategori (view)* terdapat pada Lampiran B (*Activity Diagram*).

7. *Activity Diagram Manajemen Data Kategori (input)*

Activity diagram manajemen data kategori (input) menggambarkan alur aktivitas proses menambah data kategori yang dilakukan oleh admin. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan admin untuk menambah data kategori dan reaksi sistem ketika admin memberikan aksi terhadap sistem. *Activity diagram manajemen data kategori (input)* terdapat pada Lampiran B (*Activity Diagram*).

8. *Activity Diagram Manajemen Data Kategori (edit)*

Activity diagram manajemen data kategori (edit) menggambarkan alur aktivitas proses mengubah data kategori yang dilakukan oleh admin. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan admin untuk mengubah data kategori dan reaksi sistem ketika admin memberikan aksi terhadap sistem. *Activity diagram manajemen data kategori (edit)* terdapat pada Lampiran B (*Activity Diagram*).

9. *Activity Diagram Manajemen Data Kategori (edit status aktif)*

Activity diagram manajemen data kategori (edit status aktif) menggambarkan alur aktivitas proses merubah status data kategori yang dilakukan oleh admin. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan admin untuk mengubah status kategori dan reaksi sistem ketika admin memberikan aksi terhadap sistem. *Activity diagram manajemen data kategori (edit status aktif)* terdapat pada Lampiran B (*Activity Diagram*).

10. *Activity Diagram Manajemen Data Kategori (edit status non aktif)*

Activity diagram manajemen data kategori (edit status non aktif) menggambarkan alur aktivitas proses merubah status data kategori yang dilakukan oleh admin. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan admin untuk mengubah status kategori dan reaksi sistem ketika admin memberikan aksi terhadap sistem. *Activity diagram manajemen data kategori (edit status non aktif)* terdapat pada Lampiran B (*Activity Diagram*).

11. *Activity Diagram Manajemen Data Subkategori (View)*

Activity diagram manajemen data subkategori (view) menggambarkan alur aktivitas proses melihat data subkategori yang dilakukan oleh admin. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan admin untuk melihat data subkategori dan reaksi sistem ketika admin memberikan aksi terhadap sistem. *Activity diagram manajemen data subkategori (view)* terdapat pada Lampiran B (*Activity Diagram*).

12. *Activity Diagram Manajemen Data Subkategori (input)*

Activity diagram manajemen data subkategori (input) menggambarkan alur aktivitas proses menambah data subkategori yang dilakukan oleh admin. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan admin untuk menambah data subkategori dan reaksi sistem ketika admin memberikan aksi terhadap sistem. *Activity diagram manajemen data subkategori (input)* terdapat pada Lampiran B (*Activity Diagram*).

13. *Activity Diagram Manajemen Data Subkategori (edit)*

Activity diagram manajemen data Subkategori (edit) menggambarkan alur aktivitas proses merubah data subkategori yang dilakukan oleh admin. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan admin untuk merubah

data subkategori dan reaksi sistem ketika admin memberikan aksi terhadap sistem. *Activity* diagram manajemen data subkategori (*edit*) terdapat pada Lampiran B (*Activity Diagram*).

14. *Activity Diagram* Manajemen Data Subkategori (*edit* status aktif)

Activity diagram manajemen data subkategori (*edit* status aktif) menggambarkan alur aktivitas proses merubah status data subkategori yang dilakukan oleh admin. *Activity* diagram ini menjelaskan tentang alur yang harus dilakukan admin untuk mengubah status subkategori dan reaksi sistem ketika admin memberikan aksi terhadap sistem. *Activity* diagram manajemen data subkategori (*edit* status aktif) terdapat pada Lampiran B (*Activity Diagram*).

15. *Activity Diagram* Manajemen Data Subkategori (*edit* status non aktif)

Activity diagram manajemen data subkategori (*edit* status non aktif) menggambarkan alur aktivitas proses merubah status data subkategori yang dilakukan oleh admin. *Activity* diagram ini menjelaskan tentang alur yang harus dilakukan admin untuk mengubah status subkategori dan reaksi sistem ketika admin memberikan aksi terhadap sistem. *Activity* diagram manajemen data subkategori (*edit* status non aktif) terdapat pada Lampiran B (*Activity Diagram*).

16. *Activity Diagram* Manajemen Data Supplier (*view*)

Activity diagram manajemen data supplier (*view*) menggambarkan alur aktivitas proses melihat data supplier yang dilakukan oleh admin. *Activity* diagram ini menjelaskan tentang alur yang harus dilakukan admin untuk melihat data supplier dan reaksi sistem ketika admin memberikan aksi terhadap sistem. *Activity* diagram manajemen data supplier (*view*) terdapat pada Lampiran B (*Activity Diagram*).

17. *Activity Diagram Manajemen Data Supplier (input)*

Activity diagram manajemen data supplier (input) menggambarkan alur aktivitas proses menambah data kategori yang dilakukan oleh admin. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan admin untuk menambah data kategori dan reaksi sistem ketika admin memberikan aksi terhadap sistem. *Activity diagram manajemen data supplier (input)* terdapat pada Lampiran B (*Activity Diagram*).

18. *Activity Diagram Manajemen Data Kategori (edit)*

Activity diagram manajemen data supplier (edit) menggambarkan alur aktivitas proses merubah data supplier yang dilakukan oleh admin. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan admin untuk merubah data supplier dan reaksi sistem ketika admin memberikan aksi terhadap sistem. *Activity diagram manajemen data supplier (edit)* terdapat pada Lampiran B (*Activity Diagram*).

19. *Activity Diagram Manajemen Data Supplier (edit status aktif)*

Activity diagram manajemen data supplier (edit status aktif) menggambarkan alur aktivitas proses merubah status data supplier yang dilakukan oleh admin. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan admin untuk mengubah status supplier dan reaksi sistem ketika admin memberikan aksi terhadap sistem. *Activity diagram manajemen data supplier (edit status aktif)* terdapat pada Lampiran B (*Activity Diagram*).

20. *Activity Diagram Manajemen Data Supplier (edit status non aktif)*

Activity diagram manajemen data supplier (edit status non aktif) menggambarkan alur aktivitas proses merubah status data supplier yang dilakukan oleh admin. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan admin untuk merubah status supplier dan reaksi sistem ketika admin memberikan aksi terhadap sistem. *Activity diagram manajemen data supplier (edit status non aktif)* terdapat pada Lampiran B (*Activity Diagram*).

21. *Activity Diagram Manajemen Data User (view)*

Activity diagram manajemen data user (view) menggambarkan alur aktivitas proses melihat data user yang dilakukan oleh admin. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan admin untuk melihat data user dan reaksi sistem ketika admin memberikan aksi terhadap sistem. *Activity diagram manajemen data user (view)* terdapat pada Lampiran B (*Activity Diagram*).

22. *Activity Diagram Manajemen Data User (input)*

Activity diagram manajemen data user (input) menggambarkan alur aktivitas proses menambah data user yang dilakukan oleh admin. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan admin untuk menambah data user dan reaksi sistem ketika admin memberikan aksi terhadap sistem. *Activity diagram manajemen data user (input)* terdapat pada Lampiran B (*Activity Diagram*).

23. *Activity Diagram Manajemen Data User (edit)*

Activity diagram manajemen data user (edit) menggambarkan alur aktivitas proses merubah data user yang dilakukan oleh admin. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan admin untuk merubah data user dan reaksi sistem ketika admin memberikan aksi terhadap sistem. *Activity diagram manajemen data user (edit)* terdapat pada Lampiran B (*Activity Diagram*).

24. *Activity Diagram Manajemen Data User (edit status aktif)*

Activity diagram manajemen data user (edit status aktif) menggambarkan alur aktivitas proses merubah status data kategori yang dilakukan oleh admin. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan admin untuk merubah status user dan reaksi sistem ketika admin memberikan aksi terhadap sistem. *Activity diagram manajemen data user (edit status aktif)* terdapat pada Lampiran B (*Activity Diagram*).

25. *Activity Diagram Manajemen Data User (edit status non aktif)*

Activity diagram manajemen data user (edit status non aktif) menggambarkan alur aktivitas proses merubah status data user yang dilakukan oleh admin. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan admin untuk merubah status user dan reaksi sistem ketika admin memberikan aksi terhadap sistem. *Activity diagram manajemen data user (edit status non aktif)* terdapat pada Lampiran B (*Activity Diagram*).

26. *Activity Diagram Manajemen Stok Barang (view)*

Activity diagram manajemen stok barang (view) menggambarkan alur aktivitas proses melihat data stok barang yang dilakukan oleh admin. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan admin untuk melihat data stok barang dan reaksi sistem ketika admin memberikan aksi terhadap sistem. *Activity diagram manajemen stok barang (view)* terdapat pada Lampiran B (*Activity Diagram*).

27. *Activity Diagram Manajemen Stok Barang (input)*

Activity diagram manajemen stok barang (input) menggambarkan alur aktivitas proses menambah stok barang yang dilakukan oleh admin. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan admin untuk menambah stok barang dan reaksi sistem ketika admin memberikan aksi terhadap sistem. *Activity diagram manajemen stok barang (input)* terdapat pada Lampiran B (*Activity Diagram*).

28. *Activity Diagram Melihat Data Transaksi Penjualan*

Activity diagram melihat data transaksi penjualan menggambarkan alur aktivitas proses melihat data transaksi penjualan yang dilakukan oleh admin. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan admin untuk melihat data transaksi penjualan dan reaksi sistem ketika admin memberikan aksi terhadap sistem. *Activity diagram melihat data transaksi penjualan* terdapat pada Lampiran B (*Activity Diagram*).

29. *Activity Diagram Input Data Pola*

Activity diagram input data pola menggambarkan alur aktivitas proses menambah data pola yang dilakukan oleh admin. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan admin untuk menambah data pola dan reaksi sistem ketika admin memberikan aksi terhadap sistem. *Activity diagram input data* terdapat pada Lampiran B (*Activity Diagram*).

30. *Activity Diagram Rekomendasi Promosi*

Activity diagram rekomendasi promosi menggambarkan alur aktivitas proses melihat rekomendasi promosi yang dilakukan oleh admin. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan admin untuk melihat rekomendasi promosi dan reaksi sistem ketika admin memberikan aksi terhadap sistem. *Activity diagram rekomendasi promosi* terdapat pada Lampiran B (*Activity Diagram*).

31. *Activity Diagram Transaksi Penjualan*

Activity diagram transaksi penjualan menggambarkan alur aktivitas proses transaksi penjualan yang dilakukan oleh pegawai. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan pegawai untuk melakukan proses transaksi penjualan dan reaksi sistem ketika pegawai memberikan aksi terhadap sistem. *Activity diagram transaksi penjualan* terdapat pada Lampiran B (*Activity Diagram*).

32. *Activity Diagram Melihat Data Barang*

Activity diagram melihat data barang menggambarkan alur aktivitas proses melihat data barang yang dilakukan oleh pegawai. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan pegawai untuk melihat data barang dan reaksi sistem ketika pegawai memberikan aksi terhadap sistem. *Activity diagram melihat data barang* terdapat pada Lampiran B (*Activity Diagram*).

33. *Activity Diagram Beranda Admin*

Activity diagram melihat beranda menggambarkan alur aktivitas proses melihat beranda yang dilakukan oleh admin. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan admin untuk melihat beranda dan reaksi sistem ketika admin memberikan aksi terhadap sistem. *Activity diagram* melihat beranda terdapat pada Lampiran B (*Activity Diagram*).

34. *Activity Diagram login*

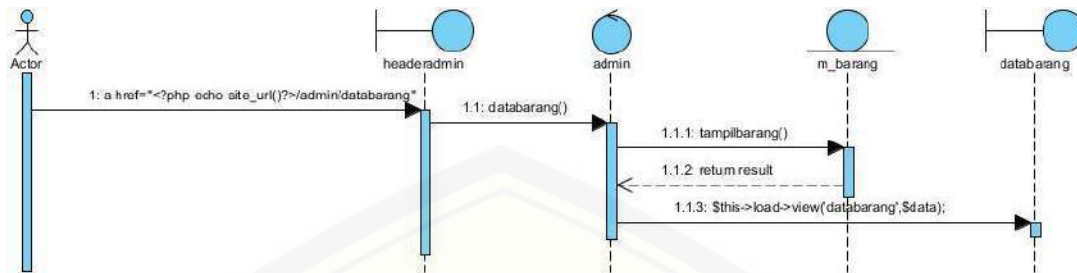
Activity diagram login menggambarkan alur aktivitas login sistem. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan pegawai dan admin untuk login dan reaksi sistem ketika pegawai dan admin memberikan aksi terhadap sistem. *Activity diagram login* terdapat pada Lampiran B (*Activity Diagram*).

4.2.5. *Sequence Diagram*

Sequence diagram merupakan dokumentasi desain berbentuk diagram terurut yang menampilkan interaksi-interaksi antar objek dalam sistem. *Sequence diagram* digunakan untuk menggambarkan skenario dan memodelkan aliran logika dalam sistem secara *visual*. *Sequence diagram* dari pengembangan modul *bundling* menggunakan *association rule* algoritma apriori pada system informasi penjualan adien mart sebagai strategi promosi penjualan barang kurang laris adalah sebagai berikut:

1. *Sequence Diagram Manajemen Data Barang (view)*

Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian atau event untuk melihat data barang yang di lakukan oleh admin. *Sequence diagram* manajemen data training (*view*) dapat dilihat pada Gambar 4.8.



Gambar 4.8 *Sequence* Diagram Manajemen Data Barang (*view*)

2. *Sequence* Diagram Manajemen Data Barang (*input*)

Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian atau event untuk menambah data barang yang dilakukan oleh admin. *Sequence* diagram manajemen data barang (*input*) dapat dilihat pada Gambar 4.9.

3. *Sequence* Diagram Manajemen Data Barang (*edit*)

Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian atau event untuk mengubah data barang. Aktor yang dapat mengubah data barang adalah admin. *Sequence* diagram manajemen data barang (*edit*) dapat dilihat pada Gambar 4.10.

4. *Sequence* Diagram Manajemen Data Barang (*edit status aktif*)

Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian atau event untuk menubah status data barang. Aktor yang dapat mengubah status barang adalah admin. *Sequence* diagram manajemen data barang (*edit status aktif*) dapat dilihat pada Gambar 4.11.

5. *Sequence* Diagram Manajemen Data Barang (*edit status non aktif*)

Sequence diagram ini menggambarkan scenario atau rangkaian langkah-langkah yang dilakukan sebagai suatu kejadian atau event untuk menubah status data barang. Aktor yang dapat mengubah status barang adalah admin. *Sequence* diagram manajemen data barang (*edit status non aktif*) dapat dilihat pada Gambar 4.12.

6. *Sequence* Diagram Manajemen Data Kategori (*view*)

Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melihat data kategori. Aktor yang dapat melihat data kategori adalah admin. *Sequence* diagram manajemen data kategori (*view*) dapat dilihat pada Lampiran C (*Sequence* Diagram).

7. *Sequence* Diagram Manajemen Data Kategori (*input*)

Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk menambah data kategori. Aktor yang dapat menambah data kategori adalah admin. *Sequence* diagram manajemen data kategori (*view*) dapat dilihat pada Lampiran C (*Sequence* Diagram).

8. *Sequence* Diagram Manajemen Data Kategori (*edit*)

Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk mengubah data kategori. Aktor yang dapat mengubah data kategori adalah admin. *Sequence* diagram manajemen data kategori (*edit*) dapat dilihat pada Lampiran C (*Sequence* Diagram).

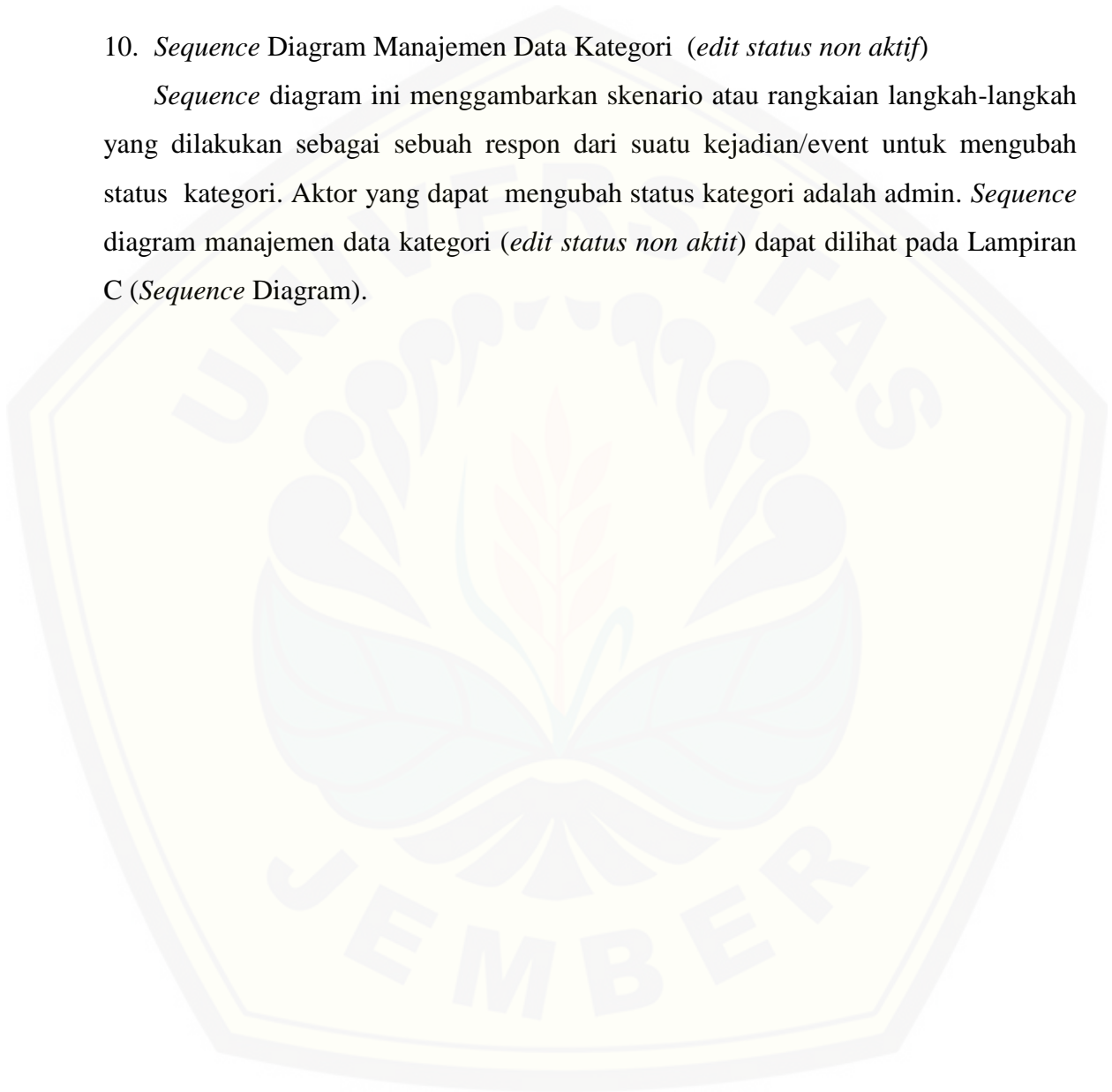
9. *Sequence* Diagram Manajemen Data Kategori (*edit status aktif*)

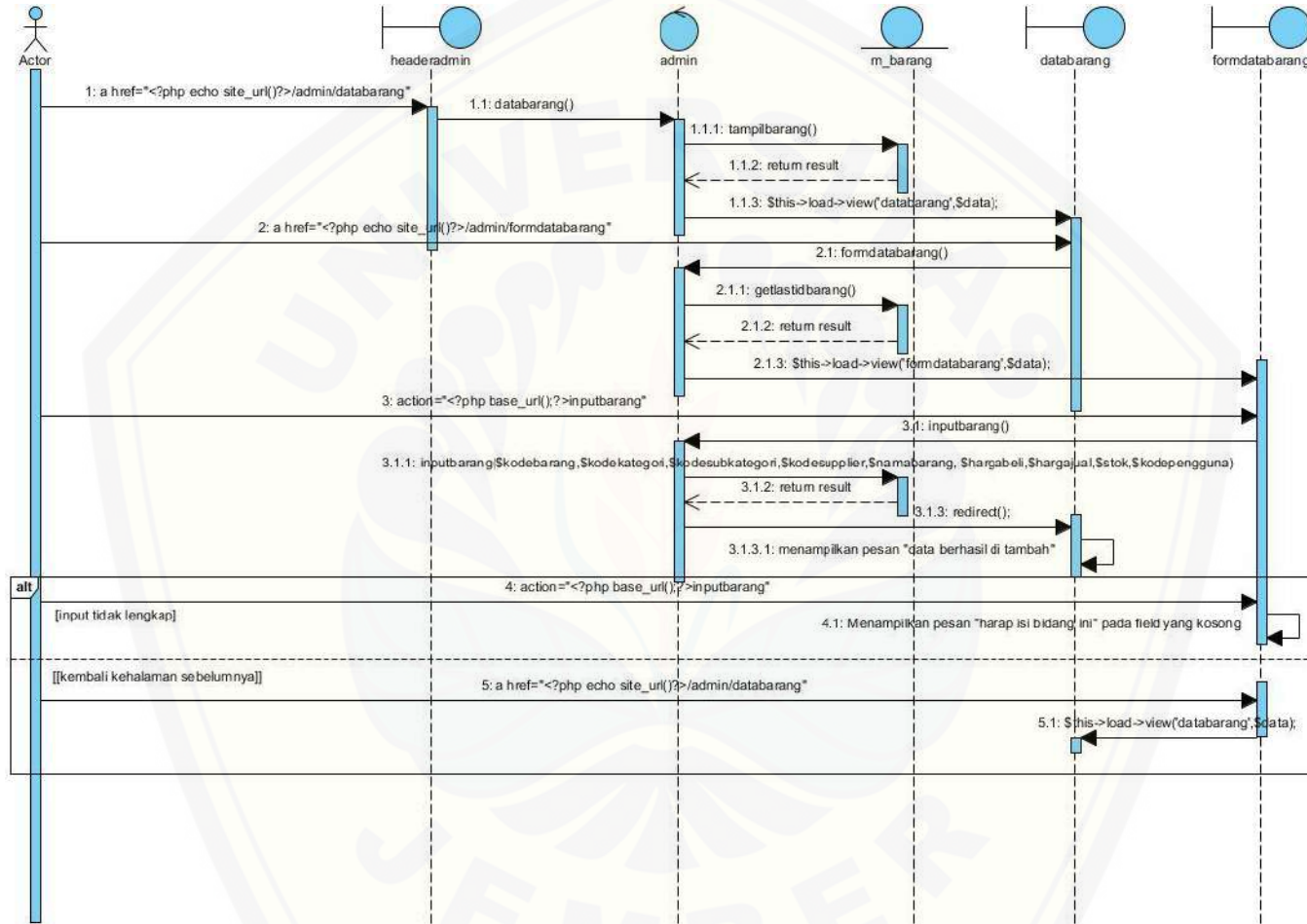
Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk mengubah status kategori. Aktor yang dapat mengubah status kategori adalah admin. *Sequence*

diagram manajemen data kategori (*edit status aktif*) dapat dilihat pada Lampiran C (*Sequence Diagram*).

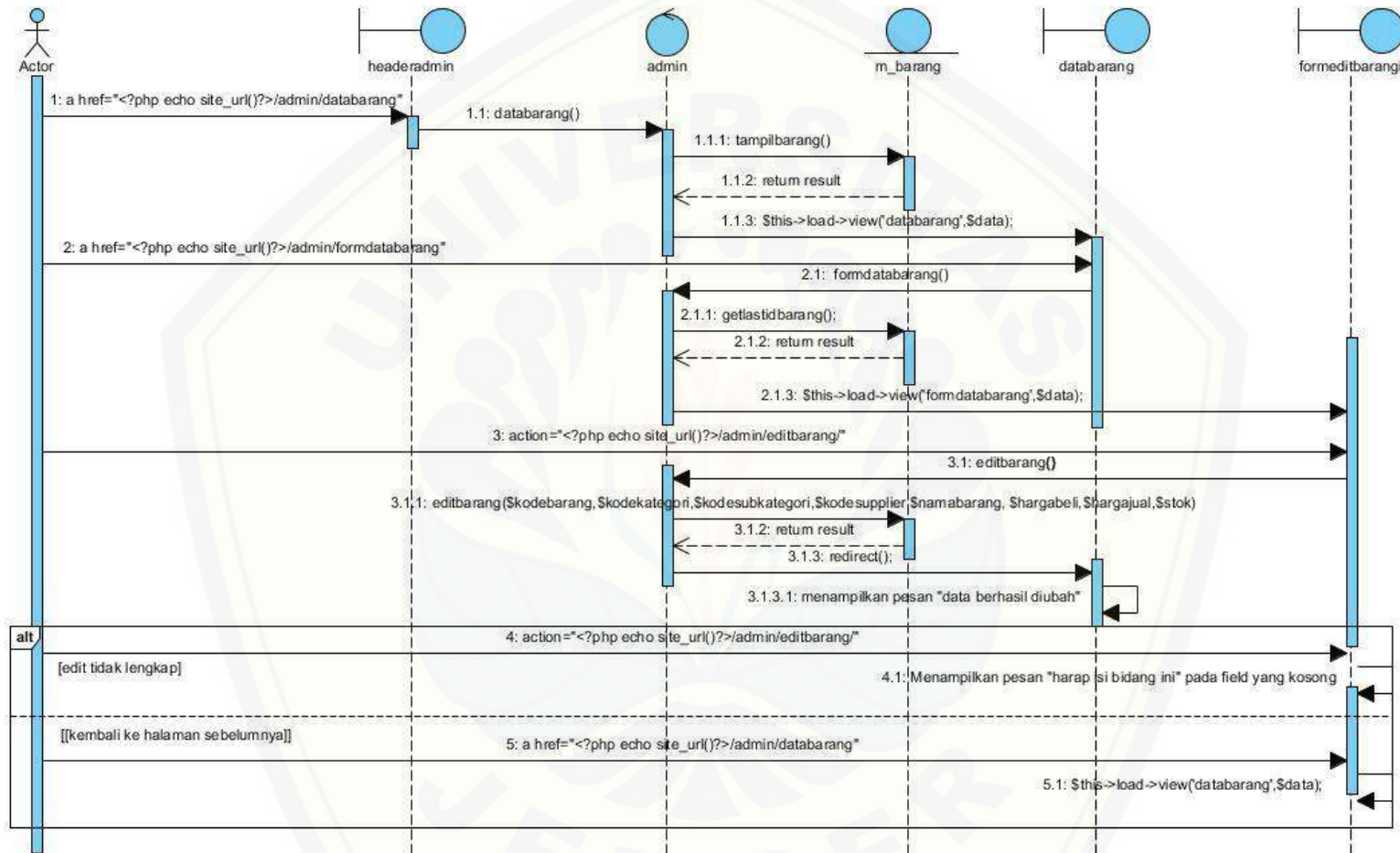
10. *Sequence Diagram* Manajemen Data Kategori (*edit status non aktif*)

Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk mengubah status kategori. Aktor yang dapat mengubah status kategori adalah admin. *Sequence diagram* manajemen data kategori (*edit status non aktif*) dapat dilihat pada Lampiran C (*Sequence Diagram*).

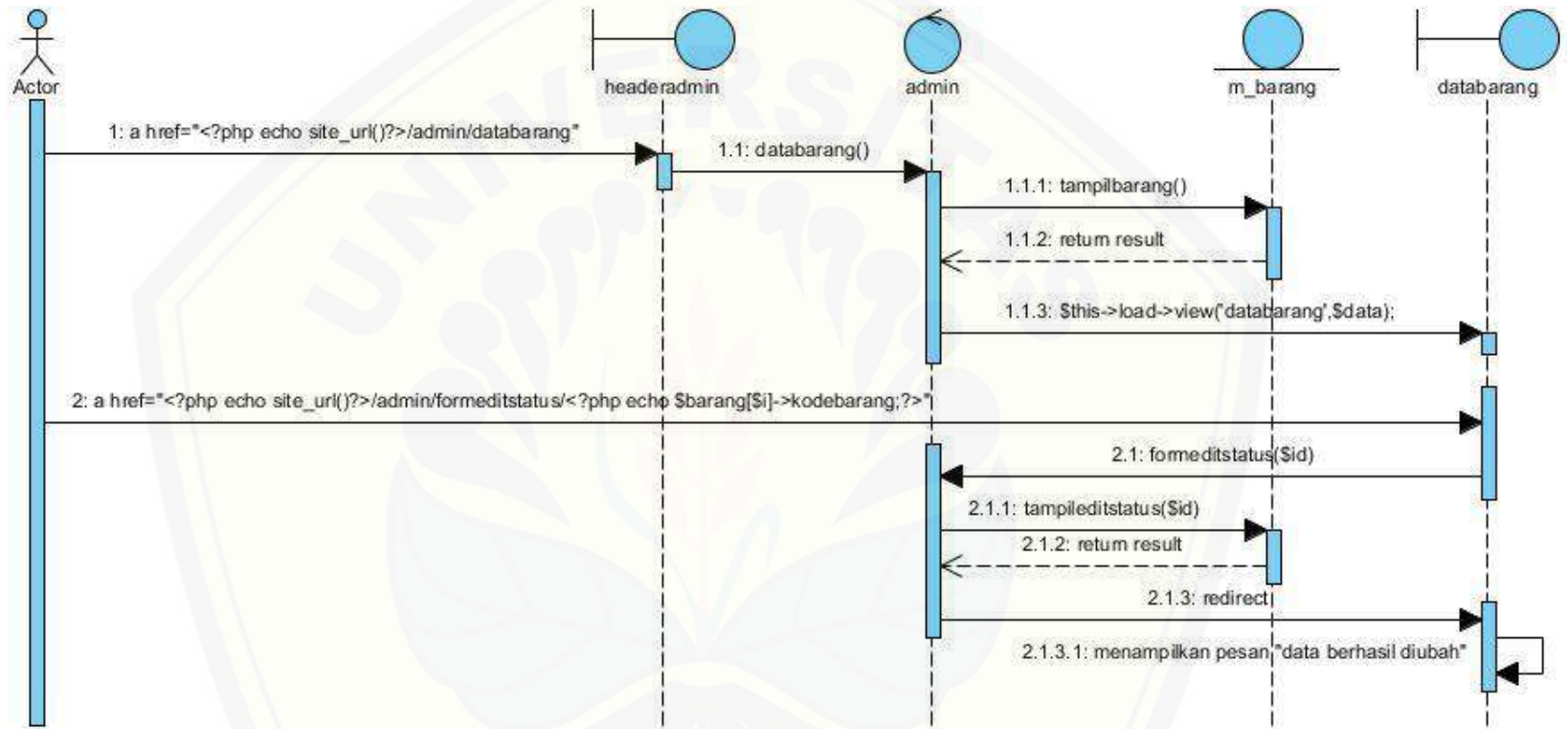




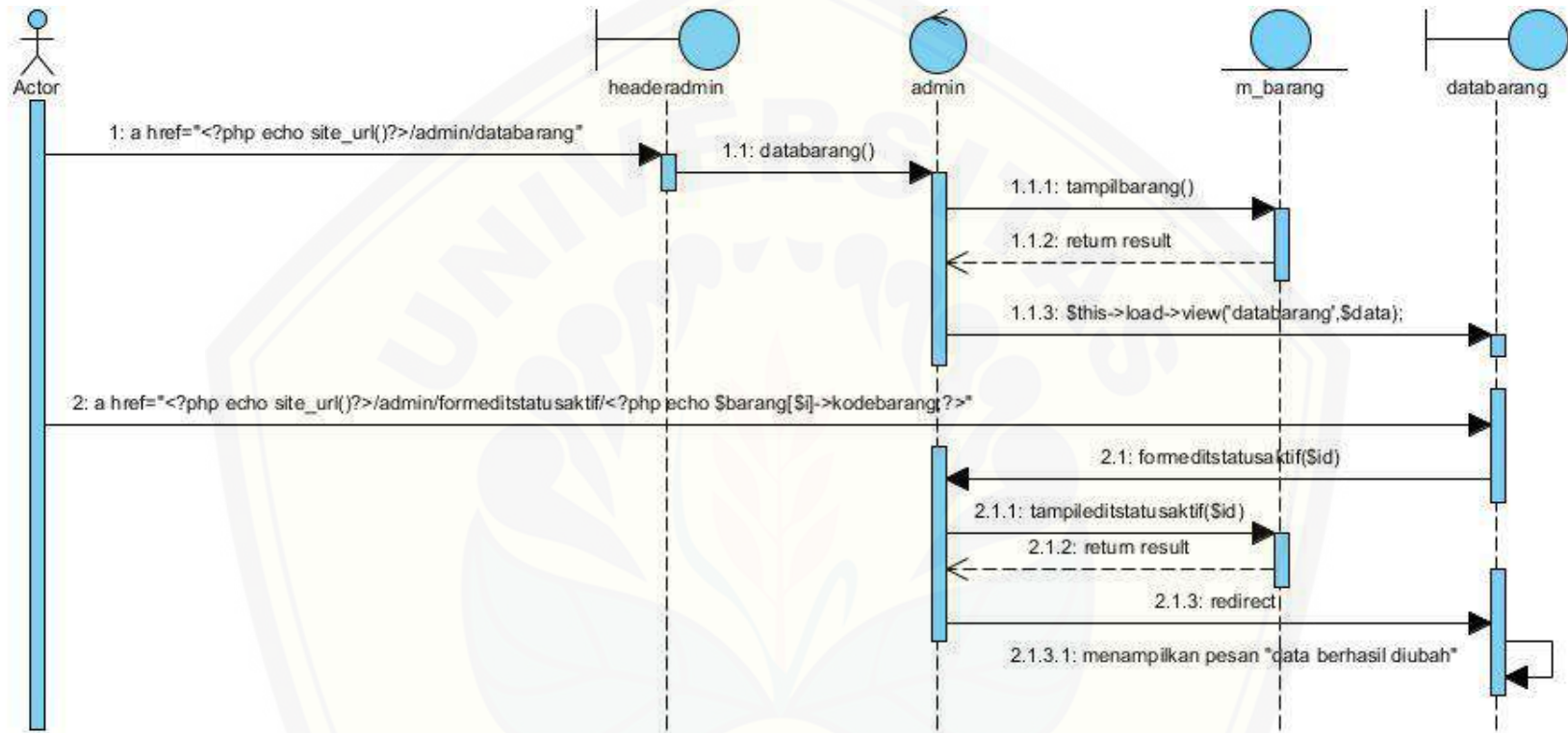
Gambar 4.9 Sequence Diagram Manajemen Data Barang (input)



Gambar 4.10 Sequence Diagram Manajemen Data Barang (edit)



Gambar 4. 11 Sequence Diagram Manajemen Data Barang (*edit status aktif*)



Gambar 4. 12 Sequence Diagram Manajemen Data Barang (*edit status non aktif*)

11. *Sequence* Diagram Manajemen Data Subkategori (*view*)

Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melihat data subkategori. Aktor yang dapat melihat data subkategori adalah admin. *Sequence* diagram manajemen data subkategori (*view*) dapat dilihat pada Lampiran C (*Sequence* Diagram).

12. *Sequence* Diagram Manajemen Data Subkategori (*input*)

Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk menambah data subkategori. Aktor yang dapat menambah data subkategori adalah admin. *Sequence* diagram manajemen data subkategori (*input*) dapat dilihat pada Lampiran C (*Sequence* Diagram).

13. *Sequence* Diagram Manajemen Data Subkategori (*edit*)

Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk mengubah data subkategori. Aktor yang dapat mengubah data subkategori adalah admin. *Sequence* diagram manajemen data subkategori (*edit*) dapat dilihat pada Lampiran C (*Sequence* Diagram).

14. *Sequence* Diagram Manajemen Data Subkategori (*edit status aktif*)

Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk mengubah status subkategori. Aktor yang dapat mengubah status kategori adalah admin. *Sequence* diagram manajemen data subkategori (*edit status aktif*) dapat dilihat pada Lampiran C (*Sequence* Diagram).

15. *Sequence* Diagram Manajemen Data Subkategori (*edit status non aktif*)

Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk mengubah status subkategori. Aktor yang dapat mengubah status subkategori adalah admin. *Sequence* diagram manajemen data subkategori (*edit status non aktif*) dapat dilihat pada Lampiran C (*Sequence* Diagram).

16. *Sequence* Diagram Manajemen Data Supplier (*view*)

Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melihat data supplier. Aktor yang dapat mengubah data supplier adalah admin. *Sequence* diagram manajemen data supplier (*view*) dapat dilihat pada Lampiran C (*Sequence* Diagram).

17. *Sequence* Diagram Manajemen Data Supplier (*input*)

Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk menambah data supplier. Aktor yang dapat menambah data supplier adalah admin. *Sequence* diagram manajemen data supplier (*input*) dapat dilihat pada Lampiran C (*Sequence* Diagram).

18. *Sequence* Diagram Manajemen Data Supplier (*edit*)

Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk mengubah data supplier. Aktor yang dapat mengubah data supplier adalah admin. *Sequence* diagram manajemen data supplier (*edit*) dapat dilihat pada Lampiran C (*Sequence* Diagram).

19. *Sequence* Diagram Manajemen Data Supplier (*edit status aktif*)

Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk mengubah

status supplier. Aktor yang dapat mengubah status supplier adalah admin. *Sequence* diagram manajemen data supplier (*edit status aktif*) dapat dilihat pada Lampiran C (*Sequence Diagram*).

20. *Sequence Diagram* Manajemen Data Supplier (*edit status non aktif*)

Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk mengubah status supplier. Aktor yang dapat mengubah status supplier adalah admin. *Sequence* diagram manajemen data supplier (*edit status non aktif*) dapat dilihat pada Lampiran C (*Sequence Diagram*).

21. *Sequence Diagram* Manajemen Data User (*view*)

Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melihat data user. Aktor yang dapat melihat data user adalah admin. *Sequence* diagram manajemen data user (*view*) dapat dilihat pada Lampiran C (*Sequence Diagram*).

22. *Sequence Diagram* Manajemen Data User (*input*)

Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk menambah data user. Aktor yang dapat menambah data user adalah admin. *Sequence* diagram manajemen data user (*input*) dapat dilihat pada Lampiran C (*Sequence Diagram*).

23. *Sequence Diagram* Manajemen Data User (*edit*)

Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk mengubah data user. Aktor yang dapat mengubah data user adalah admin. *Sequence* diagram manajemen data user (*edit*) dapat dilihat pada Lampiran C (*Sequence Diagram*).

24. *Sequence Diagram Manajemen Data User (edit status aktif)*

Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk mengubah status user. Aktor yang dapat mengubah status user adalah admin. Sequence diagram manajemen data user (edit status aktif) dapat dilihat pada Lampiran C (Sequence Diagram).

25. *Sequence Diagram Manajemen Data User (edit status non aktif)*

Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk mengubah status user. Aktor yang dapat mengubah status user adalah admin. Sequence diagram manajemen data user (edit status non aktif) dapat dilihat pada Lampiran C (Sequence Diagram).

26. *Sequence Diagram Manajemen Stok Barang (view)*

Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melihat stok barang. Aktor yang dapat melihat stok barang adalah admin. Sequence diagram manajemen stok barang (view) dapat dilihat pada Lampiran C (Sequence Diagram).

27. *Sequence Diagram Manajemen Stok Barang (input)*

Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk menambah stok barang. Aktor yang dapat menambah stok barang adalah admin. Sequence diagram manajemen stok barang (view) dapat dilihat pada Lampiran C (Sequence Diagram).

28. *Sequence* Diagram Melihat Data Transaksi Penjualan

Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melihat data transaksi penjualan. Aktor yang dapat melihat data transaksi penjualan adalah admin. *Sequence* diagram melihat data transaksi penjualan dapat dilihat pada Lampiran C (*Sequence* Diagram).

29. *Sequence* Diagram *Input* Data Pola

Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk menambah data pola. Aktor yang dapat menambah data pola adalah admin. *Sequence* diagram *input* data pola dapat dilihat pada Lampiran C (*Sequence* Diagram).

30. *Sequence* Diagram Rekomendasi Promosi

Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melihat rekomendasi promosi. Aktor yang dapat melihat rekomendasi promosi adalah admin. *Sequence* diagram rekomendasi promosi dapat dilihat pada Lampiran V (*Sequence* Diagram).

31. *Sequence* Diagram Transaksi Penjualan

Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan transaksi penjualan. Aktor yang dapat melakukan transaksi penjualan adalah pegawai. *Sequence* diagram melihat data transaksi penjualan dapat dilihat pada Lampiran C (*Sequence* Diagram).

32. *Sequence* Diagram Melihat Data Barang

Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melihat data barang. Aktor yang dapat melihat data barang adalah pegawai. *Sequence* diagram melihat data barang dapat dilihat pada Lampiran C (*Sequence* Diagram).

33. *Sequence* Diagram Melihat Beranda

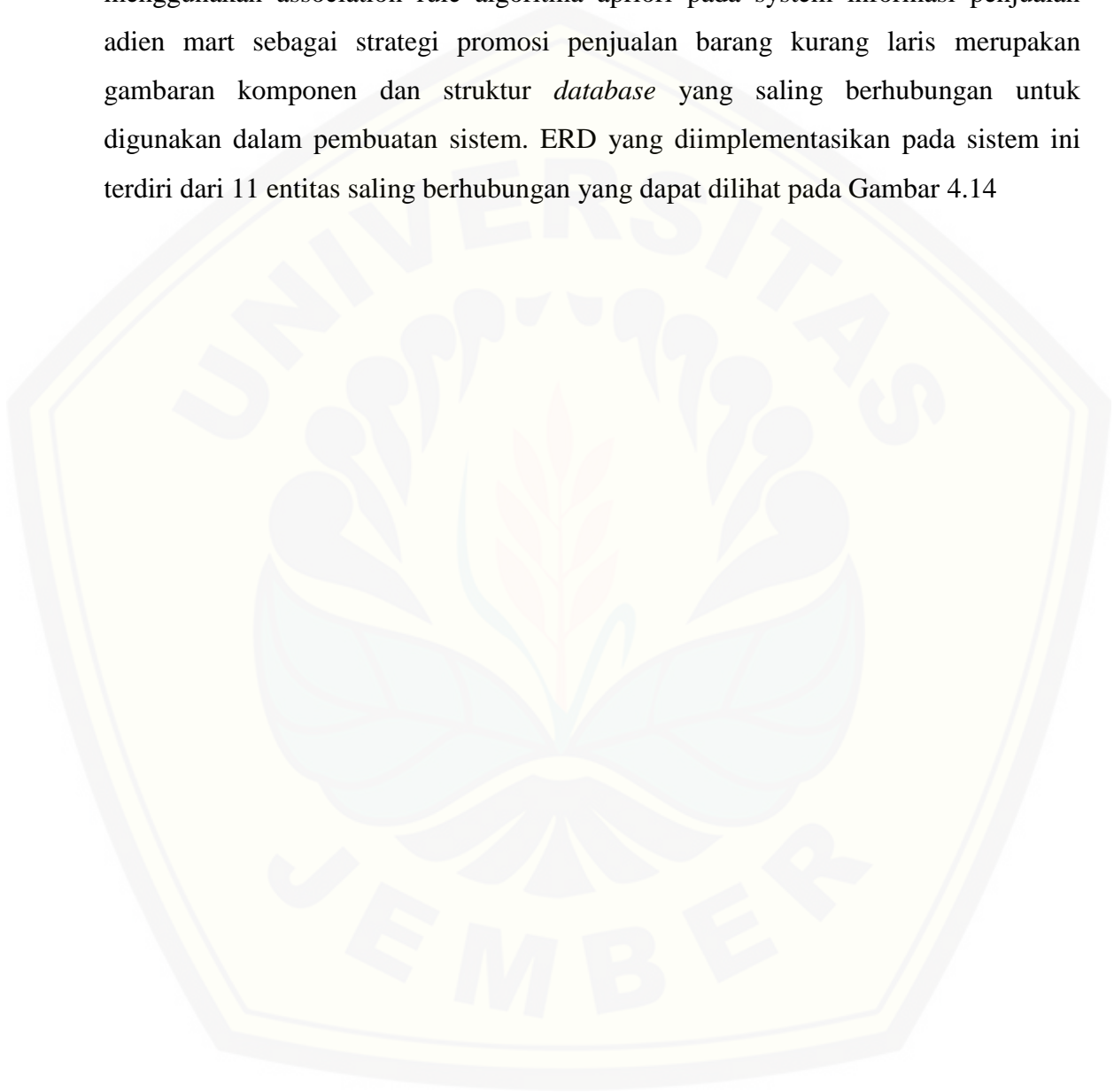
Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melihat beranda. Aktor yang dapat melihat data barang adalah admin. *Sequence* diagram melihat beranda dilihat pada Lampiran C (*Sequence* Diagram).

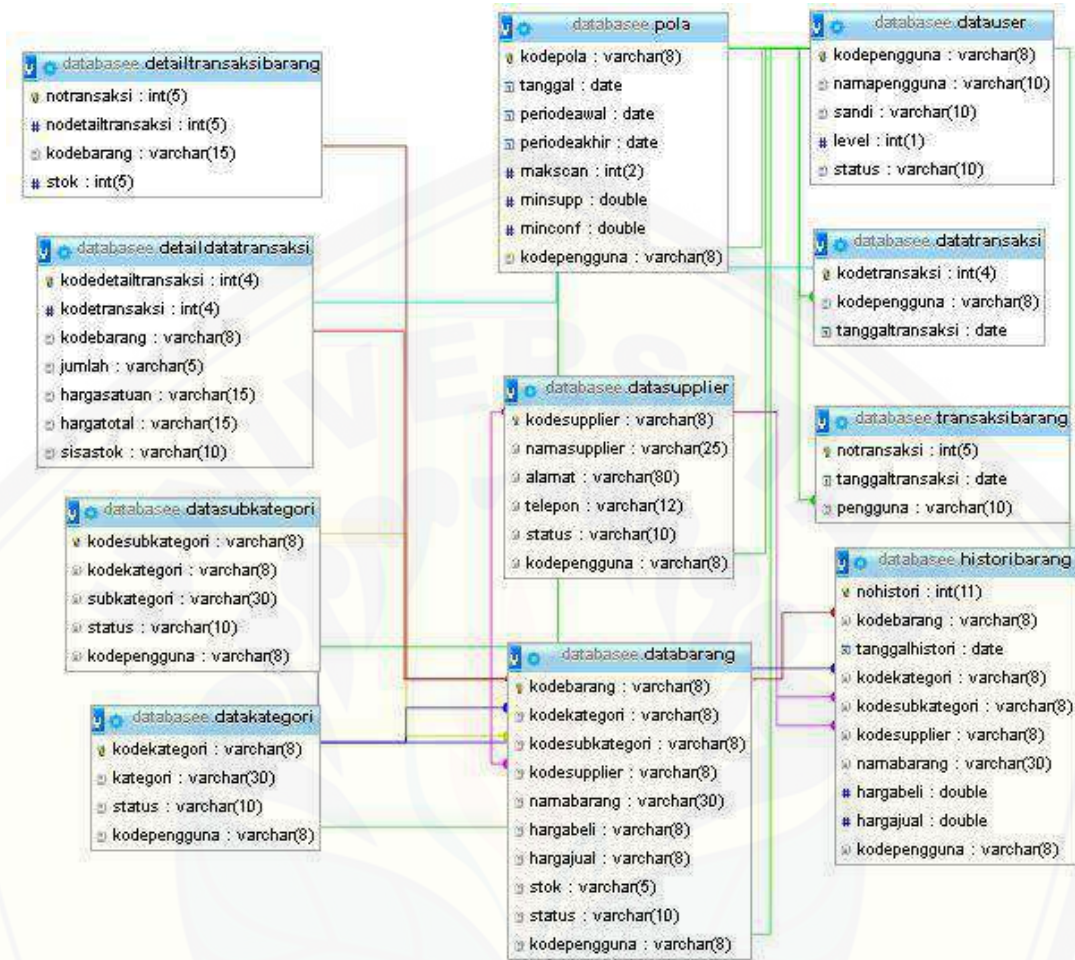
4.2.6. *Class* Diagram

Class diagram menggambarkan hubungan antar kelas yang digunakan untuk membangun suatu sistem, kelas yang digambarkan berisi atribut dan nama method pada setiap kelasnya. Dalam paradigma OOD (*Object Oriented Desain*) terdapat 3 jenis kelas yaitu *model*, *view* dan *controller* dan memiliki berbagai macam relasi yang menggambarkan hubungan antar kelas. Berdasarkan *sequence* diagram yang telah dibangun, *class* diagram pengembangan modul *bundling* menggunakan association rule algoritma apriori pada system informasi penjualan adien mart sebagai strategi promosi penjualan barang kurang laris terdiri dari 3 kelas model yaitu *m_barang*, *m_login*, *model_transaksi*, sementara *view* terdiri dari 43 kelas yang digambarkan dengan loader view dan 4 kelas *controller* yaitu *admin*, *logrizk*, *pegawai* dan *transaksi*. *Class* diagram pengembangan modul *bundling* menggunakan association rule algoritma apriori pada system informasi penjualan adien mart sebagai strategi promosi penjualan barang kurang laris dapat dilihat pada Gambar 4.13.

4.2.7. *Entity Relationship* Diagram (ERD)

Entity Relationship Diagram (ERD) pengembangan modul *bundling* menggunakan association rule algoritma apriori pada system informasi penjualan adien mart sebagai strategi promosi penjualan barang kurang laris merupakan gambaran komponen dan struktur *database* yang saling berhubungan untuk digunakan dalam pembuatan sistem. ERD yang diimplementasikan pada sistem ini terdiri dari 11 entitas saling berhubungan yang dapat dilihat pada Gambar 4.14





Gambar 4. 14 Entity Relational Diagram

4.3 Penulisan Kode Program

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

- A. Penulisan kode program (*coding*) menggunakan bahasa pemrograman *Page Hyper Text Pre-Processor (PHP)*, *HyperText Markup Language (HTML)*, *Cascading Style Sheet (CSS)* dan dengan bantuan *framework Code Igniter (CI)*.
- B. Manajemen basisdata menggunakan *DBMS MySQL*.

Kode program perhitungan metode algoritma *Apriori* terdapat di *class admin* pada *package controllers*, *class m_barang* pada *package models* dan *class aprioriclass* pada *package libraries*. Penulisan kode program terdapat pada Lampiran D (Kode Program)

4.4 Pengujian

4.4.1 White Box

Pengujian sistem dengan metode *white box* dilakukan untuk menguji sistem dari segi desain dan kode program. Pengujian bertujuan untuk mengevaluasi apakah sistem mampu menghasilkan fungsi, inputan, dan keluaran yang sesuai dengan spesifikasi dari kebutuhan sistem. Tahapan pengujian metode *white box* meliputi:

1. *Listing* program

Listing program merupakan kumpulan baris kode yang diuji. Setiap langkah dari kode yang ada diberi nomor ketika menjalankan *statement* biasa atau penggunaan kondisi dalam program.

2. Diagram alir

Diagram alir merupakan notasi 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.

3. Kompleksitas siklomatik (*cyclomatic complexity*)

Kompleksitas siklomatik merupakan metrik perangkat lunak yang menyediakan ukuran kuantitatif dari kompleksitas logis suatu program. Bila digunakan dalam konteks teknik pengujian jalur dasar, nilai yang dihitung untuk kompleksitas siklomatik mendefinisikan jumlah jalur independen dalam basis set suatu program. Perhitungan kompleksitas siklomatik menggunakan

$$V(G) = E - N + 2 \quad \dots \text{persamaan 4}$$

Keterangan:

$V(G)$ = Kompleksitas siklomatik

E = Jumlah *edge* (garis)

N = Jumlah *node* (simpul)

4. Jalur independen (*Independent Path*)

Jalur independen adalah setiap jalur yang melalui program, menunjukkan 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.

5. Pengujian Basis Set (*Test Case*)

Pada bagian ini diberikan contoh data yang menggambarkan pelaksanaan jalur di basis set. Data yang dieksekusi dimasukkan ke dalam grafik alir apakah sudah

melewati basis set yang tersedia. Sistem telah memenuhi syarat kelayakan perangkat lunak jika salah satu jalur yang dieksekusi setidaknya satu kali.

Pengujian pengembangan modul bundling menggunakan *association rule* algoritma *apriori* pada sistem informasi penjualan adien mart sebagai strategi promosi penjualan barang kurang laris akan diterapkan pada fitur yang dinilai dapat mewakili pengembangan modul bundling pada system informasi penjualan adien mart sebagai berikut:

A. Listing Program Admin

```
5 public function __construct()
6 {
7     parent::__construct();
8     $this->load->library('aprioriclass');
9     $this->load->helper('url');
10    $this->load->model('m_barang');
11    $isLoggedIn = $this->session->userdata('level');
12    if($isLoggedIn == null || $isLoggedIn != '1')
13    {
14        redirect("/logrizk");
15    }
16 }
```

Gambar 4.15 Listing Program Function __construct()

```
100 public function pencarianpola()
101 {
102     $data['pola']=$this->m_barang->tampilpola($kode = null)->result();
103     $this->load->view('headeradmin');
104     $this->load->view('pencarianpola',$data);
105     $this->load->view('footer');
106 }
```

Gambar 4. 16 Listing Program Function pencarianpola()

```

430 public function datapilih($kode)
431 {
432     $result = $this->m_barang->tampilpola($kode)->result_array()[0];
433     $data['pilih'] = $this->m_barang->tampilitem($result['periodeawal'], $result['periodeakhir']->result());
434
435     $Apriori = new Aprioriclass();
436
437     $Apriori->setMaxScan($result['makscan']); //Scan 2, 3, ...
438     $Apriori->setMinSup($result['minsup']); //Minimum support 1, 2, 3, ...
439     $Apriori->setMinConf($result['minconf']); //Minimum confidence - Percent 1, 2, ..., 100
440     $Apriori->setDelimiter(','); //Delimiter
441
442     $dataset = $this->m_barang->tampilitem($result['periodeawal'],$result['periodeakhir']->result_array());
443     $Apriori->process($dataset);
444
445     $data['pfi1'] = $Apriori->printFreqItemsets()[0];
446     $data['pfi2'] = $Apriori->printFreqItemsets()[1];
447     $data['pfi3'] = $Apriori->printFreqItemsets()[2]; //nama barang
448     $data['pfi4'] = $Apriori->printFreqItemsets()[3]; //jml
449
450     $data['pai1'] = $Apriori->printAssociationRules()[0];
451     $data['pai2'] = $Apriori->printAssociationRules()[1];
452
453
454     $data['pai3'] = $Apriori->printAssociationRules()[2];
455     $data['pai4'] = $Apriori->printAssociationRules()[3];
456     $data['pai5'] = $Apriori->printAssociationRules()[4];
457     $subkat = array();
458
459     $subkat[0] = $this->uri->segment(3);
460     $subkat[1] = $data['pai5'];
461
462     $data['subkat'] = $this->m_barang->rekom($subkat)->result_array();
463
464     $this->load->view('headeradmin');
465     $this->load->view('datapilih', $data);
466     $this->load->view('footer');
467 }

```

Gambar 4. 17 Listing Program Function datapilih(\$kode)

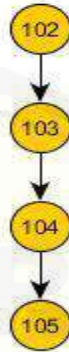
```

469 public function apriori()
470 {
471     $Apriori = new Aprioriclass();
472
473     $Apriori->setMaxScan(20); //Scan 2, 3, ...
474     $Apriori->setMinSup(2); //Minimum support 1, 2, 3, ...
475     $Apriori->setMinConf(75); //Minimum confidence - Percent 1, 2, ..., 100
476     $Apriori->setDelimiter(','); //Delimiter
477
478     $dataset = $this->m_barang->tampilitemsemua()->result_array();
479     $Apriori->process($dataset);
480
481     $data['pfi1'] = $Apriori->printFreqItemsets()[0];
482     $data['pfi2'] = $Apriori->printFreqItemsets()[1];
483
484     $data['pai1'] = $Apriori->printAssociationRules()[0];
485     $data['pai2'] = $Apriori->printAssociationRules()[1];
486
487     $this->load->view('headeradmin');
488     $this->load->view('apriori',$data);
489     $this->load->view('footer');
490 }

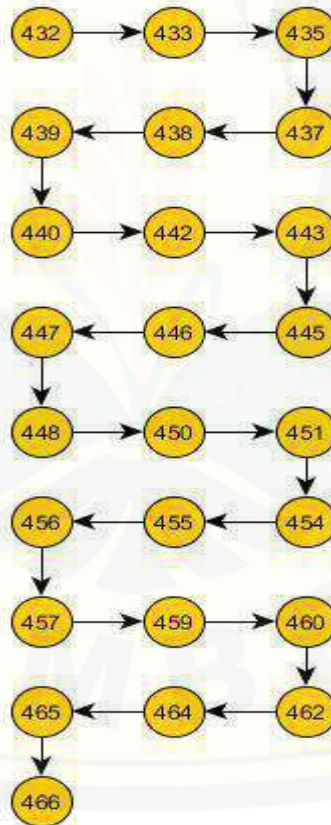
```

Gambar 4. 18 Listing Program Function apriori()

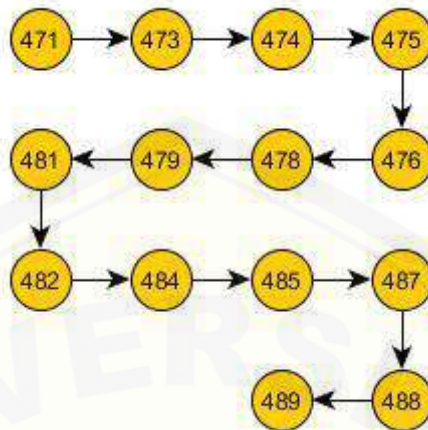
B. Diagram Alir



Gambar 4. 19 Diagram Alir Function pencarianpola()



Gambar 4.20 Diagram Alir Function datapilih()



Gambar 4. 21 Diagram Alir Function apriori()

C. Kompleksitas Siklomatik (*Cyclomatic Complexity*)

1. Function pencariaonpola()

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

$$V(G) = 3 - 4 + 2 = 1$$

2. Function datapilih(\$kode)

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

$$V(G) = 24 - 25 + 2 = 1$$

3. Function apriori()

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

$$V(G) = 13 - 14 + 2 = 1$$

D. Jalur Independen (*Independent Path*)

1. Function pencarianpola()

Jalur = 102-103-104-105

2. Function datapilih(\$kode)

Jalur = 432-433-435-437-438-439-440-442-443-445-446-447-448-450-451-454-255-456-457-459-460-462-464-465-466

3. *Function* apriori()

Jalur = 471-473-474-475-476-478-479-481-482-484-485-487-488-489

E. Pengujian Basis Set (*Test Case*)

Tabel 4. 5 *Test case function* pencarianpola()

<i>Test case function</i> pencarianpola()	
Jalur1	
<i>Test Case</i>	Jika <i>query</i> yang terdapat pada node 100-106 dijalankan
Target yang diharapkan	Berhasil menampilkan view pencarianpola dengan memanggil fuction tampilpola((\$kode = null) pada class m_barang
Hasil pengujian	Benar
Path/ Jalur	102-103-104-105

Tabel 4. 6 *Test case function* datapilih(\$kode)

<i>Test case function</i> datapilih(\$kode)	
Jalur1	
<i>Test Case</i>	Jika <i>query</i> yang terdapat pada node 430-467 dijalankan
Target yang diharapkan	Berhasil mengambil data transaksi sesuai dengan kodepola yang dipilih dan mengambil data dari function tampilpola() dan tampilItem() pada class model m_barang. Proses selanjutnya memanggil class library aprioriclass dengan meng- <i>override</i> dan menyimpan dalam bentuk array \$data[] sehingga data hasil dapat dilihat pada class view datapilih
Hasil pengujian	Benar
Path/ Jalur	432-433-435-437-438-439-440-442-443-445-446-447-448-450-451-454-255-456-457-459-460-462-464-465-466

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 penelitian yang telah dilakukan adalah:

1. Algoritma *apriori* dapat diimplementasikan dengan menginputkan parameter berupa rentang tanggal transaksi penjualan yang akan di cari pola penjualannya, nilai minimal *support* dan nilai minimal *confidence*. Data pola penjualan hasil dari algoritma *apriori* dapat di jadikan acuan dalam penentuan rekomendasi *bundling*, yaitu dengan menggabungkan barang-barang yang berelasi dan mengganti salah satu barang dengan barang tingkat penjualannya rendah atau kurang laris.
2. Pengembangan modul *bundling* pada sistem informasi penjualan Adien mart dibangun berbasis *website* dengan 2 hak akses yaitu admin dan pegawai dengan fitur utama data pola yang digunakan untuk mencari *frequent itemsets* dan rekomendasi *bundling* oleh admin. Sistem dikembangkan dengan mengadopsi model *waterfall*. Model *waterfall* sangat cocok digunakan pada pengembangan modul *bundling* ini karena memiliki proses yang urut, dimulai dari analisis kebutuhan hingga pemeliharaan, setiap proses memiliki spesifikasi sendiri sehingga sistem dapat dikembangkan sesuai keinginan dan setiap proses tidak saling tumpang tindih.

6.2 Saran

Adapun saran yang ditujukan untuk memberikan masukan yang lebih baik yaitu :

1. Pengembangan lebih lanjut diharapkan sistem dapat merekomendasikan lebih dari satu barang yang kurang laris dengan barang yang laris sesuai dengan kebiasaan konsumen.
2. Pengembangan lebih lanjut di harapkan dapat menggunakan metode *association rule* yang lain. Dalam perhitungan algoritma apriori semakin besar jumlah data maka proses perhitungan akan semakin lambat karena banyaknya pengolahan data. Untuk itu perlu dilakukan perbandingan antara algoritma *apriori* dengan metode lain yang dimiliki *association rule*.

DAFTAR PUSTAKA

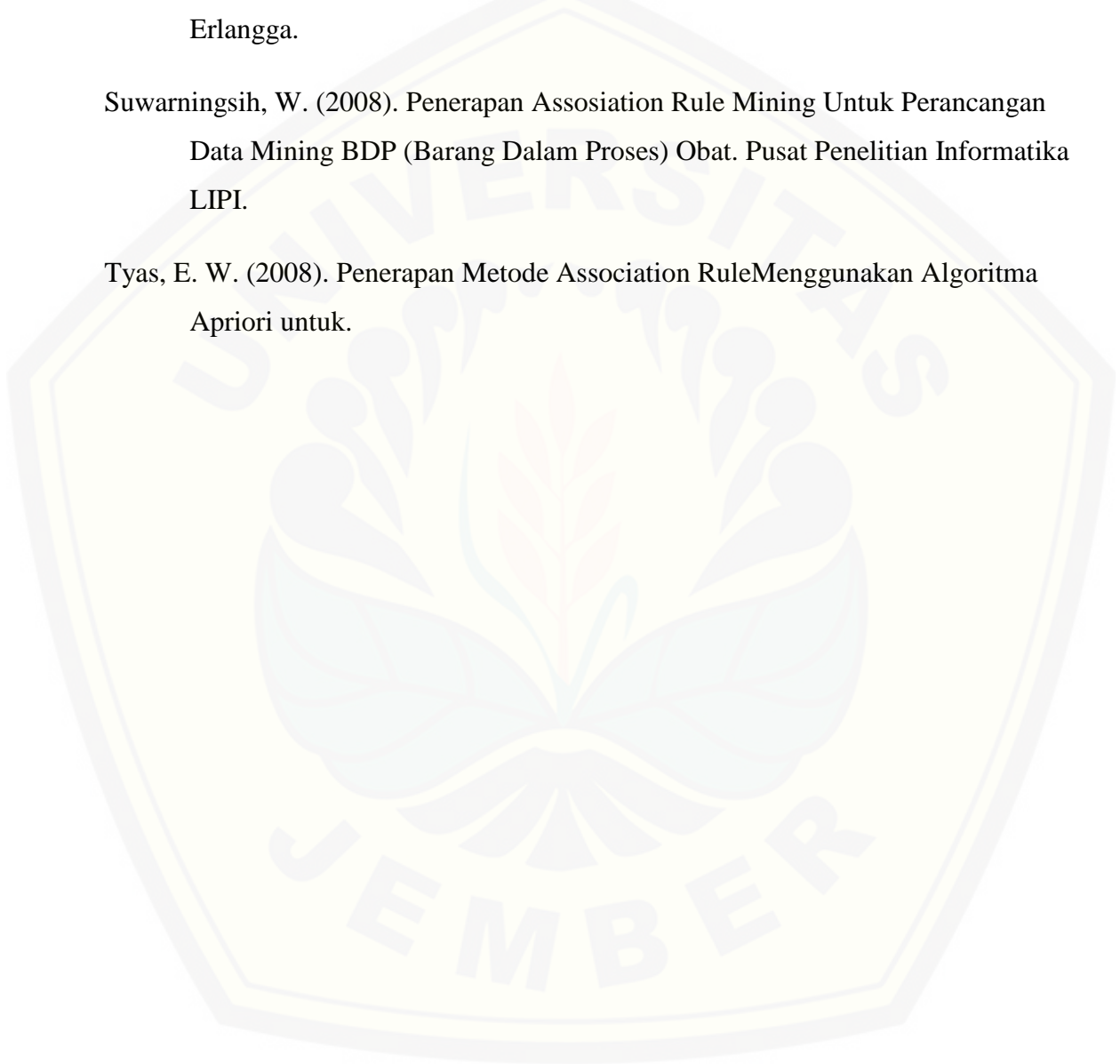
- Amalia, E. (2015). Pola Transaksi Penjualan Sebagai Strategi Penjualan Barang Menggunakan Assosiation Rule Algoritma Apriori (Studi Kasus : Senyum Media Jember).
- Backett, E. M., Davies, A. M., & Barvazian, A. P. (2015, September 15). *Iris(Institutional Repository for Information Sharing) World Health Organization*. Retrieved from http://apps.who.int/iris/bitstream:http://apps.who.int/iris/bitstream/10665/39177/1/WHO_PHP_76.pdf
- Fauziah, R. (2012). *Pengaruh Product Bundling Terhadap Pembelian Handphone Merek Nexian di Kalangan Mahasiswa Universitas Muhammadiyah Purworejo*.
- Han, J. d. (2006). *Data Mining : Concepts and Technique*. San Francisco: Morgan Kauffman.
- Jogiyanto, H. M. (2005). *Analisis dan Desain Sistem Informasi : pendekatan Terstruktur Teori dan Praktik Aplikasi Bisnis*. Yogyakarta: Andi Publisher.
- Kadir, A. (2003). *Pengenalan Sistem Informasi*. Yogyakarta: Andi Publisher.
- Kusrini, & Luthfi, E. T. (2009). *Algoritma Data Mining*. Yogyakarta: Andi Offset.
- Mujib, R., & dkk. (2013, Juni). Penerapan Data Mining untuk Evaluasi Kinerja Akademik Mahasiswa Menggunakan Algoritma Naive Bayes Classifier. *Jurnal EECCIS, Vol 7, No 1*.
- Nurul, H. A. (2013). *Promosi Penjualan*. alvanlovi.
- Pressman, R. S. (2001). *Software Engineering a practitioner's approach* (5 ed.). New York, America: McGraw-Hill.

Robi Yanto, R. K. (2015). *Implementasi Data Mining dengan Metode Algoritma Apriori dalam Menentukan Pola Pembelian Oba*. Lubuklingau: STMIK Bina Nusantara Jaya Lubuklingau.

Sommerville, I. (2011). *Software Engineering (Rekayasa Perangkat Lunak)*. Jakarta: Erlangga.

Suwarningsih, W. (2008). Penerapan Assosiation Rule Mining Untuk Perancangan Data Mining BDP (Barang Dalam Proses) Obat. Pusat Penelitian Informatika LIPI.

Tyas, E. W. (2008). Penerapan Metode Association Rule Menggunakan Algoritma Apriori untuk.



LAMPIRAN

Lampiran A. Use Case Skenario

A1. Use Case Skenario Manajemen Data Subkategori

Tabel A.1 Use Case Skenario Manajemen Data Subkategori

Nomor <i>Usecase</i>	USC 03
Nama	Manajemen Data Subkategori
Aktor	Admin
<i>Pre Condition</i>	Admin memilih menu data subkategori
<i>Post Condition</i>	Admin berhasil melihat, menambah, merubah status dan merubah data subkategori
SKENARIO UTAMA MELIHAT DATA SUBKATEGORI	
Aktor	Sistem
1. Memilih menu manajemen barang	
2. Memilih submenu data subkategori	
	3. Mengambil data subkategori dari tabel <i>datasubkategori</i> berupa kode subkategori, kategori, subkategori dan status di database.dengan menggunakan <i>fuction datasubkategori()</i>
	4. Menampilkan data subkategori kode subkategori, kategori, subkategori dan status, tombol tambah data, tombol edit, tombol nonaktif dan tombol aktif pada halaman subkategori menggunakan <i>function tampilsubkategoritabel()</i>
SKENARIO UTAMA MENAMBAH DATA SUBKATEGORI	
Aktor	Sistem
5. Menekan tombol “Tambah Data”	
	6. Menampilkan <i>form</i> isian subkategori berupa kode subkategori, kategori, status dan tombol kembali menggunakan <i>fuction formdatasubkategori()</i>
7. Mengisi isian <i>form</i> subkategori	
8. Menekan tombol “Simpan”	
	9. Memeriksa masukan data

	subkategori
	10. Menyimpan isian data <i>form</i> subkategori ke tabel <i>datakategori</i> di <i>database</i> , menggunakan <i>function</i> <i>inputsubkategori()</i> dengan atribut <i>kodesubkategori</i> , <i>kodekategori</i> , <i>subkategori</i>
	11. Menampilkan pesan “data berhasil ditambah”
	12. Mengambil data subkategori dari tabel <i>datasubkategori</i> berupa kode subkategori, kategori, subkategori dan status di <i>database</i> menggunakan <i>function</i> <i>datasubkategori()</i>
	13. Menampilkan data subkategori berupa kode subkategori, kategori, subkategori dan status, tombol tambah data, tombol edit, tombol nonaktif dan tombol aktif pada halaman subkategori menggunakan <i>function</i> <i>tampilsubkategoritabel()</i>

SKENARIO UTAMA TAMBAH DATA KEMBALI KE HALAMAN SUBKATEGORI

Aktor	Sistem
8. Menekan tombol “Kembali”	9. Mengambil data subkategori dari tabel <i>datasubkategori</i> berupa kode subkategori, kategori, subkategori dan status di <i>database</i> menggunakan <i>function</i> <i>datasubkategori()</i>
	10. Menampilkan data subkategori berupa kode subkategori, kategori, subkategori dan status, tombol tambah data, tombol edit, tombol nonaktif dan tombol aktif pada halaman subkategori menggunakan <i>function</i> <i>tampilsubkategoritabel()</i>

SKENARIO UTAMA MERUBAH STATUS AKTIF SUBKATEGORI

Aktor	Sistem
5. Menekan tombol “non aktif”	6. Merubah status pada tabel <i>datakategori</i> di <i>database</i> ,

	menggunakan <i>fuction</i> subkategoristatus() dengan atribut id
7.	Menampilkan pesan “data berhasil diubah”
8.	Mengambil data status dari tabel data kategori di database menggunakan, <i>fuction</i> subkategoristatus() dengan atribut id
9.	Menampilkan data subkategori berupa kode subkategori, kategori, subkategori dan status, tombol tambah data, tombol edit, tombol nonaktif dan tombol aktif pada halaman subkategori menggunakan <i>function</i> tampilsubkategoritabel()

SKENARIO UTAMA MERUBAH STATUS NONAKTIF SUBKATEGORI

Aktor	Sistem
5. Menekan tombol “Aktif”	
	6. Merubah status pada tabel datakategori di database, menggunakan <i>fuction</i> subkategoristatusaktif() dengan atribut id
	7. Menampilkan pesan “data berhasil diubah”
	8. Mengambil data status dari tabel data kategori di database menggunakan, <i>fuction</i> subkategoristatusaktif() dengan atribut id
	9. Menampilkan data subkategori berupa kode subkategori, kategori, subkategori dan status, tombol tambah data, tombol edit, tombol nonaktif dan tombol aktif pada halaman subkategori menggunakan <i>function</i> tampilsubkategoritabel()

SKENARIO UTAMA MERUBAH DATA SUBKATEGORI

Aktor	Sistem
5. Menekan tombol “ <i>Edit</i> ” pada baris data yang akan diubah	
	6. Mengambil data subkategori

	berupa kode subkategori, kategori, dan subkategori menggunakan <i>function</i> <code>formdatasubkategoriedit()</code> dengan atribut <code>id</code>
	7. Menampilkan <i>form</i> isian <i>edit</i> data subkategori berupa kode subkategori, kategori, subkategori dan tombol kembali dengan <i>function</i> <code>tampilsubkategoriedit()</code> dengan atribut <code>id</code> .
8. Mengubah isian <i>form edit</i> subkategori	
9. Menekan tombol “Simpan”	
	10. Memeriksa masukan data subkategori
	11. Menyimpan isian <i>edit form</i> subkategori ke tabel data subkategori di <i>database</i> , menggunakan <i>function</i> <code>editsubkategoriedit()</code> dengan atribut <code>kodesubkategori</code> , <code>kodekategori</code> dan <code>subkategori</code>
	12. Menampilkan pesan “data berhasil diubah”
	13. Mengambil data subkategori dari tabel subkategori berupa <code>kodesubkategori</code> , <code>kodekategori</code> , <code>subkategori</code> dan <code>status</code> di <i>database</i> menggunakan <i>function</i> <code>datasubkategoriedit()</code>
	14. Menampilkan data subkategori berupa kode subkategori, <code>kodekategori</code> , <code>subkategori</code> <code>status</code> , tombol tambah data, tombol edit, tombol nonaktif dan tombol aktif pada halaman subkategori dengan menggunakan <i>function</i> <code>tampilsubkategoritabel()</code>
SKENARIO UTAMA MERUBAH DATA KEMBALI KE HALAMAN DATA SUBKATEGORI	
Aktor	Sistem
9. Menekan tombol “Kembali”	10. Mengambil data subkategori dari

	tabel subkategori berupa kodesubkategori, kodekategori, subkategori dan status di database menggunakan <i>function</i> <i>datasubkategori()</i> .
	11. Menampilkan data subkategori berupa kode subkategori, kodekategori, subkategori, status, tombol tambah data, tombol edit, tombol nonaktif dan tombol aktif pada halaman subkategori dengan menggunakan <i>function</i> <i>tampilsubkategoritabel()</i>
SKENARIO ALTERNATIF MENAMBAH DATA SUBKATEGORI	
Jika aktor tidak mengisi <i>form</i> subkategori secara lengkap	
Aktor	Sistem
7a. Mengisi isian <i>form</i> subkategori tidak lengkap	
8a. Menekan tombol “Simpan”	9a. Memeriksa masukan data subkategori
	10a. Menampilkan pesan dengan menunjuk lokasi isian yang belum diisi pada <i>form</i> subkategori. Pesan akan tampil di <i>field</i> yang kosong
SKENARIO ALTERNATIF MERUBAH DATA SUBKATEGORI	
Jika aktor tidak mengisi <i>form</i> edit subkategori secara lengkap	
Aktor	Sistem
8a. Mengisi isian <i>form</i> edit subkategori tidak lengkap	
9a. Menekan tombol “Simpan”	10a. Memeriksa masukan data subkategori
	11a. Menampilkan pesan dengan menunjuk lokasi isian yang belum diisi pada <i>form</i> edit sub kategori. Pesan akan tampil di <i>field</i> yang kosong

A2. Use Case Skenario Manajemen Data Subkategori

Tabel A.2 Use Case Skenario Manajemen Data Supplier

Nomor Usecase	USC 04
Nama	Manajemen Data Supplier
Aktor	Admin
Pre Condition	Admin memilih menu data supplier
Post Condition	Admin berhasil melihat, menambah, merubah status dan merubah data supplier
SKENARIO UTAMA MELIHAT DATA SUPPLIER	
Aktor	Sistem
1. Memilih menu data supplier	2. Mengambil data supplier dari tabel datasupplier berupa kode supplier, nama supplier, alamat, telepon dan status di database.dengan menggunakan <i>function</i> datasupplier()
	3. Menampilkan data supplier berupa kode supplier, nama supplier, alamat, telepon dan status, tombol tambah data, tombol edit, tombol nonaktif dan tombol aktif pada halaman supplier menggunakan <i>fuction</i> tampilsupplier()
SKENARIO UTAMA MENAMBAH DATA SUPPLIER	
Aktor	Sistem
4. Menekan tombol “Tambah Data”	5. Menampilkan <i>form</i> isian supplier berupa kode supplier, nama supplier, alamat dan telepon dan tombol kembali menggunakan <i>fuction</i> formdatasupplier()
6. Mengisi isian <i>form</i> supplier	
7. Menekan tombol “Simpan”	8. Memeriksa masukan data supplier
	9. Menyimpan isian data <i>form</i> supplier ke tabel datasupplier di <i>database</i> menggunakan <i>fuction</i> inputdatasupplier() dengan atribut kodesupplier, namasupplier, alamat dan telepon.

	10. Menampilkan pesan “data berhasil di tambah”
	11. Mengambil data supplier dari tabel datasupplier berupa kode supplier, namasupplier, alamat, telepon dan status di database menggunakan <i>function</i> datasupplier()
	12. Menampilkan data supplier berupa kode supplier, nama supplier, alamat, telepon dan status, tombol tambah data, tombol edit, tombol nonaktif dan tombol aktif pada halaman supplier menggunakan <i>function</i> tampilsupplier()

SKENARIO UTAMA TAMBAH DATA KEMBALI KE HALAMAN SUPPLIER

Aktor	Sistem
7. Menekan tombol “Kembali”	
	8. Mengambil data supplier dari tabel datasupplier berupa kode supplier, namasupplier, alamat, telepon dan status di database menggunakan <i>function</i> datasupplier()
	9. Menampilkan data supplier berupa kode supplier, nama supplier, alamat, telepon dan status, tombol tambah data, tombol edit, tombol nonaktif dan tombol aktif pada halaman supplier menggunakan <i>function</i> tampilsupplier()

SKENARIO UTAMA MERUBAH STATUS AKTIF SUPPLIER

Aktor	Sistem
4. Menekan tombol “non aktif”	
	5. Merubah status pada tabel supplier di database, menggunakan <i>fuction</i> supplierstatus() dengan atribut id
	6. Menampilkan pesan “data berhasil diubah”
	7. Mengambil data status dari tabel data supplier di database menggunakan, <i>fuction</i> supplierstatus() dengan atribut id
	8. Menampilkan data supplier berupa kode supplier, nama supplier,

	alamat, telepon dan status, tombol tambah data, tombol edit, tombol nonaktif dan tombol aktif pada halaman supplier menggunakan <i>function</i> tampilsupplier()
SKENARIO UTAMA MERUBAH STATUS NONAKTIF SUPPLIER	
Aktor	Sistem
4. Menekan tombol “Aktif”	5. Merubah status pada tabel datasupplier di database, menggunakan <i>fuction</i> supplierstatusaktif() dengan atribut id
	6. Menampilkan pesan “data berhasil diubah”
	7. Mengambil data status dari tabel data supplier di database menggunakan, <i>fuction</i> supplierstatusaktif() dengan atribut id
	8. Menampilkan data supplier berupa kode supplier, nama supplier, alamat, telepon dan status, tombol tambah data, tombol edit, tombol nonaktif dan tombol aktif pada halaman supplier menggunakan <i>function</i> tampilsupplier()
SKENARIO UTAMA MERUBAH DATA SUPPLIER	
Aktor	Sistem
4. Menekan tombol “ <i>Edit</i> ” pada baris data yang akan diubah	5. Mengambil data supplier berupa kode supplier, nama supplier, alamat dan telepon menggunakan <i>fuction</i> formdatasupplieredit() dengan atribut id
	6. Menampilkan <i>form</i> isian <i>edit</i> data supplier berupa kode supplier, nama supplier, alamat, telepon dan tombol kembali dengan <i>function</i> tampilsupplieredit() dengan atribut id
7. Mengubah isian <i>form edit</i> supplier	
8. Menekan tombol “Simpan”	

	9. Memeriksa masukan data supplier
	10. Menyimpan isian <i>edit form</i> supplier ke tabel data supplier di <i>database</i> , menggunakan <i>fuction</i> <i>editdatasupplier()</i> dengan atribut <i>kodesupplier</i> , <i>namasupplier</i> , <i>alamat</i> dan <i>telepon</i>
	11. Menampilkan pesan “data berhasil diubah”
	12. Mengambil data supplier dari tabel <i>datasupplier</i> berupa kode supplier, <i>namasupplier</i> , <i>alamat</i> , <i>telepon</i> dan status di <i>database</i> menggunakan <i>function</i> <i>datasupplier()</i>
	13. Menampilkan data supplier berupa kode supplier, nama supplier, <i>alamat</i> , <i>telepon</i> dan status, tombol tambah data, tombol edit, tombol nonaktif dan tombol aktif pada halaman supplier menggunakan <i>function</i> <i>tampilsupplier()</i>

SKENARIO UTAMA MERUBAH DATA KEMBALI KE HALAMAN DATA SUPPLIER

Aktor	Sistem
8. Menekan tombol “Kembali”	
	9. Mengambil data supplier dari tabel <i>datasupplier</i> berupa kode supplier, <i>namasupplier</i> , <i>alamat</i> , <i>telepon</i> dan status di <i>database</i> menggunakan <i>function</i> <i>datasupplier()</i>
	10. Menampilkan data supplier berupa kode supplier, nama supplier, <i>alamat</i> , <i>telepon</i> dan status, tombol tambah data, tombol edit, tombol nonaktif dan tombol aktif pada halaman supplier menggunakan <i>function</i> <i>tampilsupplier()</i>

SKENARIO ALTERNATIF MENAMBAH DATA SUPPLIER

Jika aktor tidak mengisi *form* supplier secara lengkap

Aktor	Sistem
6a. Mengisi isian <i>form</i> supplier tidak lengkap	
7a. Menekan tombol “Simpan”	
	8a. Memeriksa masukan data supplier

	9a. Menampilkan pesan dengan menunjuk lokasi isian yang belum diisi pada <i>form</i> subkategori. Pesan akan tampil di <i>field</i> yang kosong
SKENARIO ALTERNATIF MERUBAH DATA SUPPLIER	
Jika aktor tidak mengisi <i>form</i> edit supplier secara lengkap	
Aktor	Sistem
7a. Mengisi isian <i>form</i> edit supplier tidak lengkap	
8a. Menekan tombol “Simpan”	
	9a. Memeriksa masukan data supplier
	10a. Menampilkan pesan dengan menunjuk lokasi isian yang belum diisi pada <i>form</i> edit supplier. Pesan akan tampil disetiap <i>field</i> yang kosong

A3. Use Case Skenario Manajemen Data User

Tabel A.3 Use Case Skenario Manajemen Data User

Nomor <i>Usecase</i>	USC 05
Nama	Manajemen Data User
Aktor	Admin
<i>Pre Condition</i>	Admin memilih menu data user
<i>Post Condition</i>	Admin berhasil melihat, menambah, merubah status dan merubah data user
SKENARIO UTAMA MELIHAT DATA USER	
Aktor	Sistem
1. Memilih menu data user	
	2. Mengambil data user dari tabel datauser berupa kode pengguna, nama pengguna, sandi, level dan status di database.dengan menggunakan <i>function</i> datuser()
	3. Menampilkan data user berupa kode pengguna, nama pengguna, sandi, level dan status, tombol tambah data, tombol edit, tombol nonaktif dan tombol aktif pada halaman user menggunakan <i>function</i> tampiluser()
SKENARIO UTAMA MENAMBAH DATA USER	

Aktor	Sistem
4. Menekan tombol “Tambah Data”	5. Menampilkan <i>form</i> isian user berupa kode pengguna, nama pengguna, sandi, level dan tombol kembali menggunakan <i>fuction</i> <i>formdatauser()</i>
6. Mengisi isian <i>form</i> user	
7. Menekan tombol “Simpan”	8. Memeriksa masukan data user
	9. Menyimpan isian data <i>form</i> user ke tabel <i>datauser</i> di <i>database</i> , menggunakan <i>fuction</i> <i>inputdatauser()</i> dengan atribut <i>kodepengguna</i> , <i>namapengguna</i> , <i>sandi</i> , <i>level</i> .
	10. Menampilkan pesan “data berhasil di tambah”
	11. Mengambil data user dari tabel <i>datauser</i> berupa kode pengguna, nama pengguna, sandi, level dan status di <i>database</i> . dengan menggunakan <i>function</i> <i>datuser()</i>
	12. Menampilkan data user berupa kode pengguna, nama pengguna, sandi, level dan status, tombol tambah data, tombol edit, tombol nonaktif dan tombol aktif pada halaman user menggunakan <i>fuction</i> <i>tampiluser()</i>

SKENARIO UTAMA TAMBAH DATA KEMBALI KE HALAMAN USER

Aktor	Sistem
7. Menekan tombol “Kembali”	8. Mengambil data user dari tabel <i>datauser</i> berupa kode pengguna, nama pengguna, sandi, level dan status di <i>database</i> . dengan menggunakan <i>function</i> <i>datuser()</i>
	9. Menampilkan data user berupa kode pengguna, nama pengguna, sandi, level dan status, tombol tambah data, tombol edit, tombol nonaktif dan tombol aktif pada halaman user menggunakan <i>fuction</i>

tampiluser()	
SKENARIO UTAMA MERUBAH STATUS AKTIF USER	
Aktor	Sistem
4. Menekan tombol “non aktif”	5. Merubah status pada tabel supplier di database, menggunakan <i>fuction</i> <i>penggunastatus()</i> dengan atribut id
	6. Menampilkan pesan “data berhasil diubah”
	7. Mengambil data status dari tabel data supplier di database menggunakan, <i>fuction</i> <i>penggunastatus()</i> dengan atribut id
	8. Menampilkan data user berupa kode pengguna, nama pengguna, sandi, level dan status, tombol tambah data, tombol edit, tombol nonaktif dan tombol aktif pada halaman user menggunakan <i>fuction</i> <i>tampiluser()</i>
SKENARIO UTAMA MERUBAH STATUS NONAKTIF USER	
Aktor	Sistem
4. Menekan tombol “Aktif”	5. Merubah status pada tabel datasupplier di database, menggunakan <i>fuction</i> <i>penggunastatusaktif()</i> dengan atribut id
	6. Menampilkan pesan “data berhasil diubah”
	7. Mengambil data status dari tabel data supplier di database menggunakan, <i>fuction</i> <i>penggunastatusaktif()</i> dengan atribut id
	8. Menampilkan data user berupa kode pengguna, nama pengguna, sandi, level dan status, tombol tambah data, tombol edit, tombol nonaktif dan tombol aktif pada halaman user menggunakan <i>fuction</i> <i>tampiluser()</i>
SKENARIO UTAMA MERUBAH DATA USER	
Aktor	Sistem

4. Menekan tombol “ <i>Edit</i> ” pada baris data yang akan diubah	5. Mengambil data user berupa kode pengguna, nama pengguna, sandi dan level menggunakan <i>fuction</i> <i>formdatauseredit()</i> dengan atribut <i>id</i>
	6. Menampilkan <i>form</i> isian <i>edit</i> data user berupa kode pengguna, nama pengguna, sandi, level dan tombol kembali dengan <i>function</i> <i>tampiluseredit()</i> dengan atribut <i>id</i>
7. Mengubah isian <i>form edit</i> user	
8. Menekan tombol “Simpan”	9. Memeriksa masukan data user
	10. Menyimpan isian <i>edit form</i> user ke tabel data user di <i>database</i> , menggunakan <i>fuction</i> <i>editdatauser()</i> dengan atribut <i>kodepengguna</i> , <i>namapengguna</i> , <i>sandi</i> dan <i>level</i>
	11. Menampilkan pesan “data berhasil dirubah”
	12. Mengambil data user dari tabel <i>datauser</i> berupa kode pengguna, nama pengguna, sandi, level dan status di <i>database</i> .dengan menggunakan <i>function</i> <i>datuser()</i>
	13. Menampilkan data user berupa kode pengguna, nama pengguna, sandi, level dan status, tombol tambah data, tombol edit, tombol nonaktif dan tombol aktif pada halaman user menggunakan <i>fuction</i> <i>tampiluser()</i>

SKENARIO UTAMA MERUBAH DATA KEMBALI KE HALAMAN DATA USER

Aktor	Sistem
8. Menekan tombol “Kembali”	9. Mengambil data user dari tabel <i>datauser</i> berupa kode pengguna, nama pengguna, sandi, level dan status di <i>database</i> .dengan menggunakan <i>function</i> <i>datuser()</i>

10. Menampilkan data user berupa kode pengguna, nama pengguna, sandi, level dan status, tombol tambah data, tombol edit, tombol nonaktif dan tombol aktif pada halaman user menggunakan *function* tampiluser()

SKENARIO ALTERNATIF MENAMBAH DATA USER

Jika aktor tidak mengisi *form* user secara lengkap

Aktor	Sistem
6a. Mengisi isian <i>form</i> user tidak lengkap	
7a. Menekan tombol “Simpan”	8a. Memeriksa masukan data user
	9a. Menampilkan pesan dengan menunjuk lokasi isian yang belum diisi pada <i>form</i> user. Pesan akan tampil di <i>field</i> yang kosong

SKENARIO ALTERNATIF MERUBAH DATA USER

Jika aktor tidak mengisi *form* edit user secara lengkap

Aktor	Sistem
7a. Mengisi isian <i>form</i> edit user tidak lengkap	
8a. Menekan tombol “Simpan”	9a. Memeriksa masukan data user
	10a. Menampilkan pesan dengan menunjuk lokasi isian yang belum diisi pada <i>form</i> edit user. Pesan akan tampil di <i>field</i> yang kosong

A4. Use Case Skenario Manajemen Stok Barang

Tabel A.4 Use Case Skenario Manajemen Stok Barang

Nomor Usecase	USC 06
Nama	Manajemen Stok Barang
Aktor	Admin
Pre Condition	Admin memilih menu stok barang
Post Condition	Admin berhasil melihat dan menambah stok barang

SKENARIO UTAMA MELIHAT DATA STOK BARANG

Aktor	Sistem
1. Memilih menu manajemen barang	

2. Memilih submenu stok barang	3. Mengambil data stok barang dari tabel data barang dan detailtransaksibarang berupa kode barang, nama barang, tanggal transaksi dan stok di database dengan menggunakan <i>fuction</i> transaksibarangmasuk()
	4. Menampilkan data stok barang berupa kodebarang, nama barang, tanggal transaksi dan stok , <i>drop down</i> bulan dan tahun, tombol tambah stok, tombol submit di halaman stok barang dengan function <i>tampiltambahstoksemua()</i>
5. Memilih bulan dan tahun yang diinginkan	
6. Menekan tombol “submit”	7. Mengambil data stok barang dari tabel databarang dan detailtransaksibarang berupa kode barang, nama barang, tanggal transaksi dan stok di database menggunakan <i>fuction</i> transaksibarangmasuk()
	8. Menampilkan data stok barang berupa kodebarang, nama barang, tanggal transaksi dan jumlah , <i>drop down</i> bulan dan tahun, tombol tambah stok, tombol submit di halaman stok barang menggunakan <i>fuction</i> <i>tampiltambahstok()</i> dengan parameter bulan dan tahun.
SKENARIO UTAMA MENAMBAH DATA STOK BARANG	
Aktor	Sistem
9. Menekan tombol “Tambah Stok”	10. Menampilkan <i>form</i> isian tambah stok berupa nama barang, jumlah, tombol tambah menggunakan <i>fuction</i> <i>tambahstok()</i>
11. Mengisi isian <i>form</i> tambah stok	
12. Menekan tombol “Tambah”	13. Menampilkan data pada tabel di halaman <i>form</i> tambah stok

14. Menekan tombol “Simpan”	15. Memeriksa masukan data stok barang
	16. Menyimpan isian data <i>form</i> stok barang ke tabel databarang dan tabel detailtransaksibarang di <i>database</i> , menggunakan <i>fuction</i> simpantambahstok() dengan atribut nama barang dan jumlah.
	17. Mengambil data stok barang dari table data barang dan detail transaksi barang berupa kode barang, namabarang, tanggal transaksi dan jumlah di database menggunakan <i>fuction</i> transaksibarangmasuk()
	18. Menampilkan data stok barang berupa kode barang, nama barang, tanggal transaksi dan jumlah , <i>drop down</i> bulan dan tahun, tombol tambah stok, tombol submit di halaman stok barang menggunakan <i>fuction</i> tampiltambahstoksemua()

SKENARIO ALTERNATIF MENAMBAH DATA STOK BARANG

Jika aktor tidak mengisi *form* tambah stok secara lengkap

Aktor	Sistem
10a. Mengisi isian <i>form</i> tambah stok barang tidak lengkap	
11a. Menekan tombol “Simpan”	12a. Memeriksa masukan data stok barang
	13a. Menampilkan pesan dengan menunjuk lokasi isian yang belum diisi pada <i>form</i> tambah stok barang.

A5. Use Case Skenario Melihat Transaksi Penjualan

Tabel A.5 Use Case Skenario Melihat Transaksi Penjualan

Nomor Usecase	USC 07
Nama	Melihat Transaksi Penjualan
Aktor	Admin
Pre Condition	Admin memilih menu data transaksi
Post Condition	Admin berhasil melihat data transaksi
SKENARIO UTAMA MELIHAT TRANSAKSI PENJUALAN	
Aktor	Sistem
1. Memilih menu data transaksi	2. Mengambil data transaksi dari tabel <i>datatransaksi</i> dan <i>detaildatatransaksi</i> berupa kode transaksi, tanggal transaksi, harga total di database dengan menggunakan <i>fuction datatransak()</i>
	3. Menampilkan data transaksi berupa kode transaksi, tanggal transaksi, harga total, <i>drop down</i> bulan dan tahun, tombol submit di halaman data transaksi penjualan menggunakan <i>fuction tampildatatransaksemau()</i>
4. Memilih bulan dan tahun yang diinginkan	
5. Menekan tombol “submit”	6. Mengambil data transaksi dari tabel <i>datatransaksi</i> dan <i>detaildatatransaksi</i> berupa kode transaksi, tanggal transaksi, harga total di halaman data transaksi penjualan menggunakan <i>fuction datatransak()</i>
	7. Menampilkan data transaksi berupa kode transaksi, tanggal transaksi, harga total, <i>drop down</i> bulan dan tahun, tombol submit di halaman data transaksi penjualan menggunakan menggunakan <i>fuction tampildatatransak()</i> dengan parameter bulan dan tahun
8. Menekan tombol “detail”	9. Mengambil detail data transaksi

	dari <i>database</i> berupa kode barang dan jumlah dengan menggunakan <i>function</i> <i>infotransaksi()</i> dengan parameter <i>id</i>
10.	Menampilkan detail data transaksi di halaman detail transaksi yang berisi kode barang dan jumlah menggunakan <i>function</i> <i>tampilinfotransaksi()</i> dengan parameter <i>id</i>

SKENARIO UTAMA MELIHAT DETAIL TRANSAKSI KEMBALI KE HALAMAN DATA TRANSAKSI PENJUALAN

Aktor	Sistem
10. Menekan tombol “Kembali”	
	11. Mengambil data transaksi dari tabel <i>datatransaksi</i> dan <i>detaildatatransaksi</i> berupa kode transaksi, tanggal transaksi, harga total di <i>database</i> dengan menggunakan <i>function</i> <i>datatransak()</i>
	12. Menampilkan data transaksi berupa kode transaksi, tanggal transaksi, harga total, <i>drop down</i> bulan dan tahun, tombol submit di halaman data transaksi penjualan menggunakan <i>function</i> <i>tampildatatransaksemau()</i>

A6. Use Case Skenario Input Parameter Pencarian Pola

Tabel A.6 Use Case Skenario Input Parameter Pencarian Pola

Nomor <i>Usecase</i>	USC 08
Nama	Input parameter pencarian pola
Aktor	Admin
<i>Pre Condition</i>	Admin memilih menu data pola
<i>Post Condition</i>	Admin berhasil menambah data parameter pencarian pola

SKENARIO UTAMA MENAMBAH DATA PARAMETER

Aktor	Sistem
1.	Memilih menu data pola

	2. Mengambil data parameter pencarian pola di tabel pola berupa kode pola, tanggal, periode awal, periode akhir, , minimal support, minimal confidence dengan menggunakan <i>function</i> pencarianpola()
	3. Menampilkan semua data parameter pada tabel di halaman pencarian pola berupa kode pola, tanggal, periode awal, periode akhir, , minimal support, minimal confidence, tombol +data parameter dan tombol proses menggunakan <i>function</i> tampilpola()
4. Menekan tombol + Data Parameter	
	5. Menampilkan <i>form</i> isian parameter di halaman form parameter berupa kode pola, tanggal, periode awal, periode akhir, , minimal support, minimal confidence menggunakan <i>function</i> formpencarianpola()
6. Mengisi form parameter	
7. Menekan tombol simpan	
	8. Memeriksa masukan data parameter
	9. Menyimpan data parameter ke <i>database</i> menggunakan <i>function</i> inputpola() dengan atribut kode pola, tanggal, periode awal, periode akhir, , minimal support, minimal confidence
	10. Mengambil data parameter pencarian pola di tabel pola berupa kode pola, tanggal, periode awal, periode akhir, minimal support, minimal confidence dengan menggunakan <i>function</i> pencarianpola()
	11. Menampilkan semua data parameter pada tabel di halaman pencarian pola berupa kode pola, tanggal, periode awal, periode

akhir, , minimal support, minimal confidence, tombol +data parameter dan tombol proses menggunakan *function* tampilpola()

SKENARIO UTAMA MENAMBAH DATA PARAMETER KEMBALI KE HALAMAN PENCARIAN POLA

Aktor	Sistem
6. Menekan tombol “Kembali”	7. Mengambil data parameter pencarian pola di tabel pola berupa kode pola, tanggal, periode awal, periode akhir, , minimal support, minimal confidence dengan menggunakan <i>function</i> pencarianpola()
	8. Menampilkan semua data parameter pada tabel di halaman pencarian pola berupa kode pola, tanggal, periode awal, periode akhir, , minimal support, minimal confidence, tombol +data parameter dan tombol proses menggunakan <i>function</i> tampilpola()

A7. Use Case Skenario Rekomendasi Promosi

Tabel A.7 Use Case Skenario Rekomendasi Promosi

Nomor Usecase	USC 9
Nama	Rekomendasi Promosi
Aktor	Admin
Pre Condition	Admin memilih menu data pola
Post Condition	Admin melihat hasil rekomendasi promosi

SKENARIO UTAMA MELIHAT HASIL REKOMENDASI

1. Memilih menu data pola	2. Mengambil data parameter pencarian pola di tabel pola berupa kode pola, tanggal, periode awal, periode akhir, , minimal support, minimal confidence dengan menggunakan <i>function</i> tampilpola()
---------------------------	--

	3. Menampilkan semua data parameter pada tabel di halaman pencarian pola berupa kode pola, tanggal, periode awal, periode akhir, , minimal support, minimal confidence, tombol +data parameter dan tombol proses
4. Menekan tombol proses	
	5. Mengambil data hasil rekomendasi dari databse menggunakan <i>function</i> <i>datapilih()</i> dengan parameter kode
	6. Menampilkan hasil rekomendasi yaitu berupa nama barang dan <i>confidence</i> pada tabel <i>association rule</i> dan berupa nama barang, pasangan dan rekomendasi pada tabel pilihan pada halaman rekomendasi menggunakan <i>function</i> <i>rekom()</i> dengan parameter subkat

A8. Use Case Skenario Transaksi Penjualan

Tabel A.8 Use Case Skenario Transaksi Penjualan

Nomor <i>Usecase</i>	USC 10
Nama	Transaksi Penjualan
Aktor	Pegawai
<i>Pre Condition</i>	Pegawai memilih menu transaksi penjualan
<i>Post Condition</i>	Pegawai berhasil melihat dan menambah data transaksi penjualan
SKENARIO UTAMA MELIHAT TRANSAKSI PENJUALAN	
Aktor	Sistem
1. Memilih menu transaski penjualan	
	2. Mengambil data transaksi penjualan dari tabel <i>datatransaksi</i> dan <i>detaildatatransaksi</i> <i>database</i> berupa kode tramsaksi, tanggal transaksi, harga total menggunakan <i>function</i> <i>index()</i>
	3. Menampilkan semua data transaksi penjualan pada tabel di halaman transaksi berupa kode tramsaksi,

tanggal transaksi, harga total, tombol tambah transaksi, *drop down* bulan dan tahun, tombol submit dan tombol detail menggunakan *function tampildatatransaksisemua()*

4. Memilih bulan dan tahun yang diinginkan

5. Menekan tombol “submit”

6. Mengambil data transaksi penjualan dari *database* berupa kode transaksi, tanggal transaksi, harga total menggunakan *function index()* dengan parameter bulan dan tahun

7. Menampilkan semua data transaksi penjualan pada tabel di halaman transaksi berdasarkan bulan dan tahun yang berupa kode transaksi, tanggal transaksi, harga total, tombol tambah transaksi, *drop down* bulan dan tahun, tombol submit dan tombol detail menggunakan *function tampildatatransaksi()*

8. Menekan tombol “detail”

9. Mengambil detail data transaksi penjualan dari *database* berupa kode barang dan jumlah menggunakan *function infotransaksi()* dengan parameter id

10. Menampilkan detail transaksi berdasarkan id berupa kode barang, jumlah dan tombol kembali menggunakan *function tampilinfotransaksi()*

**SKENARIO UTAMA MELIHAT TRANSAKSI PENJUALAN
KEMBALI KE HALAMAN TRANSAKSI PENJUALAN**

11. Menekan tombol “kembali”

12. Mengambil data transaksi penjualan dari *database* berupa kode transaksi, tanggal transaksi, harga total

menggunakan *function* index()

13. Menampilkan semua data transaksi penjualan pada tabel di halaman transaksi berupa kode transaksi, tanggal transaksi, harga total, tombol tambah transaksi, *drop down* bulan dan tahun, tombol submit dan tombol detail menggunakan *function* `tampildatatransaksisemua()`

SKENARIO UTAMA MENAMBAH DATA TRANSAKSI PENJUALAN

Aktor	Sistem
4. Menekan tombol “Tambah Transaksi”	5. Menampilkan <i>form</i> isian tambah transaksi berupa no transaksi, nama barang, harga, jumlah menggunakan <i>function</i> <code>detailtransaksi()</code>
6. Mengisi isian <i>form</i> tambah transaksi	8. Menampilkan data pada tabel di halaman <i>form</i> tambah transaksi berupa no, namabarang, harga, jumlah, subtotal dan harga total.
7. Menekan tombol “Tambah”	10. Memeriksa masukan data transaksi 11. Menyimpan masukan data ke tabel <code>detaildatatransaksi</code> di <i>database</i> berupa nama barang, harga, jumlah, subtotal dan harga total dengan <i>function</i> <code>simpantransaksidetail()</code>
9. Menekan tombol “Simpan”	12. Mengambil data transaksi dari tabel <code>datatransaksi</code> dan tabel <code>detaildatatransaksi</code> di <i>database</i> berupa kode transaksi, tanggal transaksi dan harga total menggunakan <i>function</i> <code>index()</code>
	13. Menampilkan data transaksi pada tabel di halaman transaksi penjualan berupa kode transaksi, tanggal transaksi dan harga total, tombol tambah transaksi, <i>drop down</i> bulan dan tahun, tombol

submit dan tombol detail menggunakan <i>function</i> <i>tampildatatransaksisemua()</i>	
SKENARIO ALTERNATIF MENAMBAH DATA TRANSAKSI PENJUALAN	
Jika aktor tidak mengisi <i>form</i> tambah transaksi secara lengkap	
Aktor	Sistem
6a. Mengisi isian <i>form</i> tambah transaksi tidak lengkap	
7a. Menekan tombol “Simpan”	
	8a. Memeriksa masukan data transaksi
	9a. Menampilkan pesan bahwa data tidak lengkap

A9. Use Case Skenario Melihat Data Barang

Tabel A.9 Use Case Skenario Melihat Data Data Barang

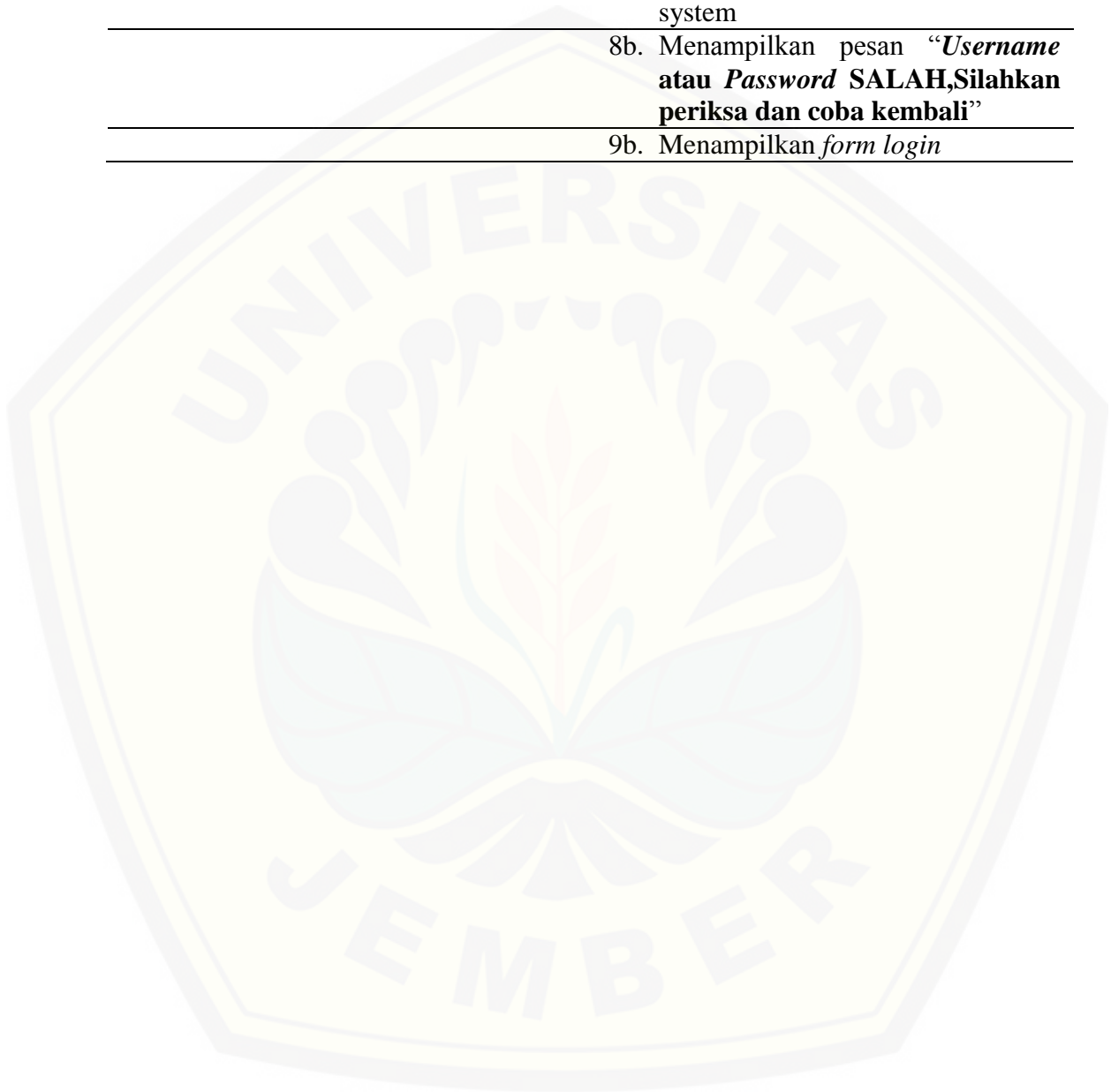
Nomor <i>Usecase</i>	USC 11
Nama	Melihat Data Barang
Aktor	Pegawai
<i>Pre Condition</i>	Pegawai memilih menu data barang
<i>Post Condition</i>	Pegawai berhasil melihat
SKENARIO UTAMA MELIHAT DATA BARANG	
Aktor	Sistem
1. Memilih menu data barang	
	2. Mengambil data barang dari tabel data barang di <i>database</i> berupa kode barang, nama barang, harga dan stok menggunakan <i>function</i> <i>index()</i>
	3. Menampilkan data barang pada tabel di halaman data barang berupa kode barang, nama barang, harga dan stok menggunakan <i>function</i> <i>tampilbarang2()</i>

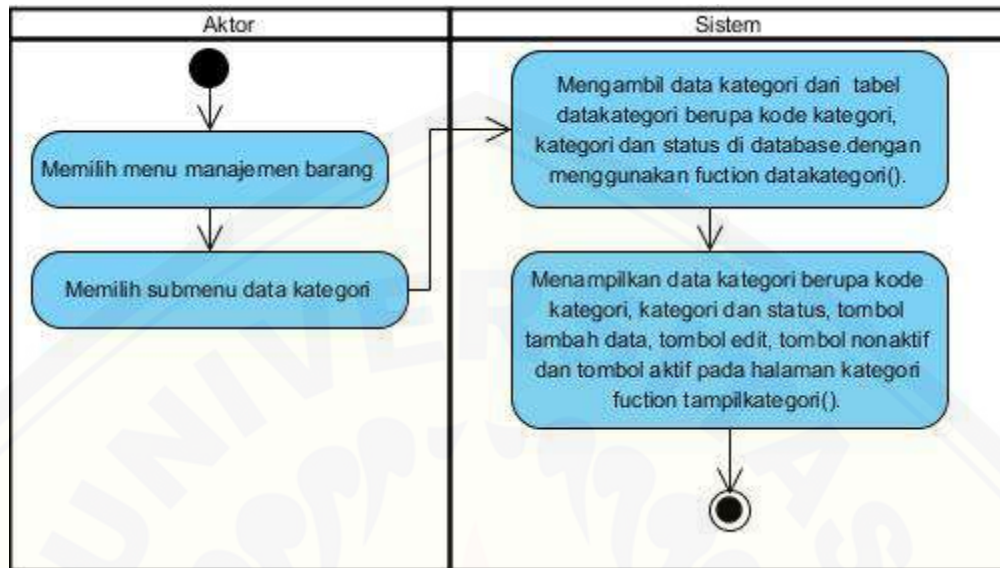
A10. Use Case Skenario Login

Tabel A.10 Use Case Skenario Login

Nomor Usecase	USC 12
Nama	Login
Aktor	Admin dan Pegawai
Pre Condition	Admin dan Pegawai harus mempunyai <i>username</i> dan <i>password</i> dengan level masing-masing untuk masuk kedalam sistem dan mengakses menu sesuai dengan level
Post Condition	Admin dan Pegawai berhasil login dan masuk ke dalam sistem sesuai dengan level
SKENARIO UTAMA LOGIN	
Aktor	Sistem
1. Menjalankan sistem	2. Menampilkan halaman <i>form login</i> berupa <i>username</i> , <i>password</i> dan tombol login menggunakan <i>function index()</i>
3. Mengisi <i>username</i> dan <i>password</i>	
4. Menekan tombol “Login”	5. Memeriksa kelengkapan isian form login
	6. Memeriksa data <i>login</i> ke <i>database</i>
	7. Menampilkan halaman utama sesuai level aktor masing-masing menggunakan <i>function index()</i>
SKENARIO ALTERNATIF	
Jika aktor tidak mengisi <i>form login</i> secara lengkap	
Aktor	Sistem
3a. <i>Username</i> (kosong) atau <i>password</i> (kosong)	
4a. Menekan tombol “Login”	5a. Memeriksa kelengkapan masukan <i>username</i> dan <i>password</i> pada system
	6a. Menampilkan pesan “ <i>field</i> harus diisi” dibawah <i>field</i> yang kosong
	7a. Menampilkan <i>form login</i>
SKENARIO ALTERNATIF	
Jika aktor memasukkan <i>username</i> atau <i>password</i> salah	

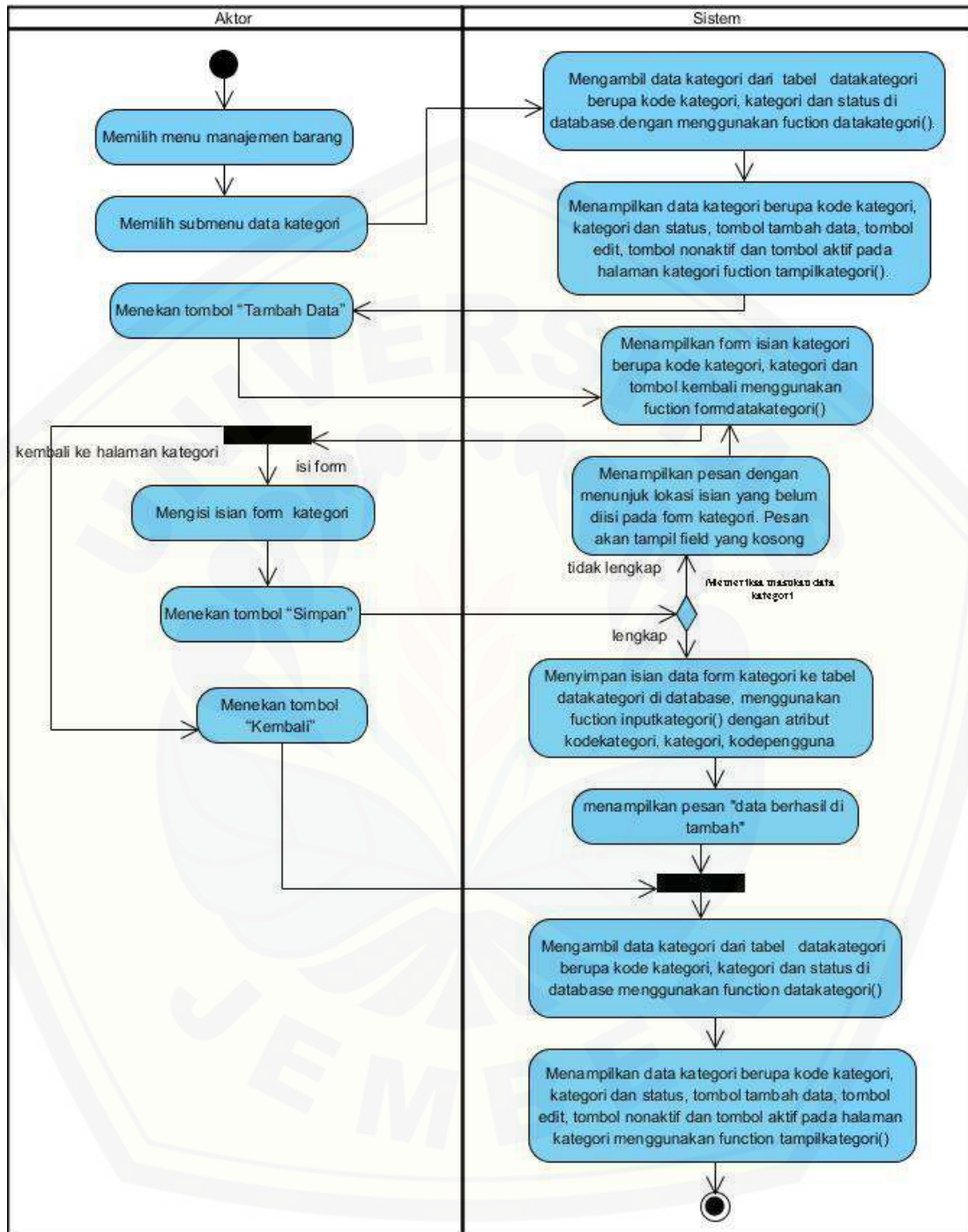
Aktor	Sistem
5b. Mengisi <i>username</i> dan <i>password</i>	
6b. Menekan tombol " <i>Login</i> "	
	7b. Memeriksa ketepatan masukan <i>username</i> dan <i>password</i> pada system
	8b. Menampilkan pesan " <i>Username atau Password SALAH, Silahkan periksa dan coba kembali</i> "
	9b. Menampilkan <i>form login</i>



Lampiran B. Activity Diagram**B1. Activity Diagram Manajemen Data Kategori (view)**

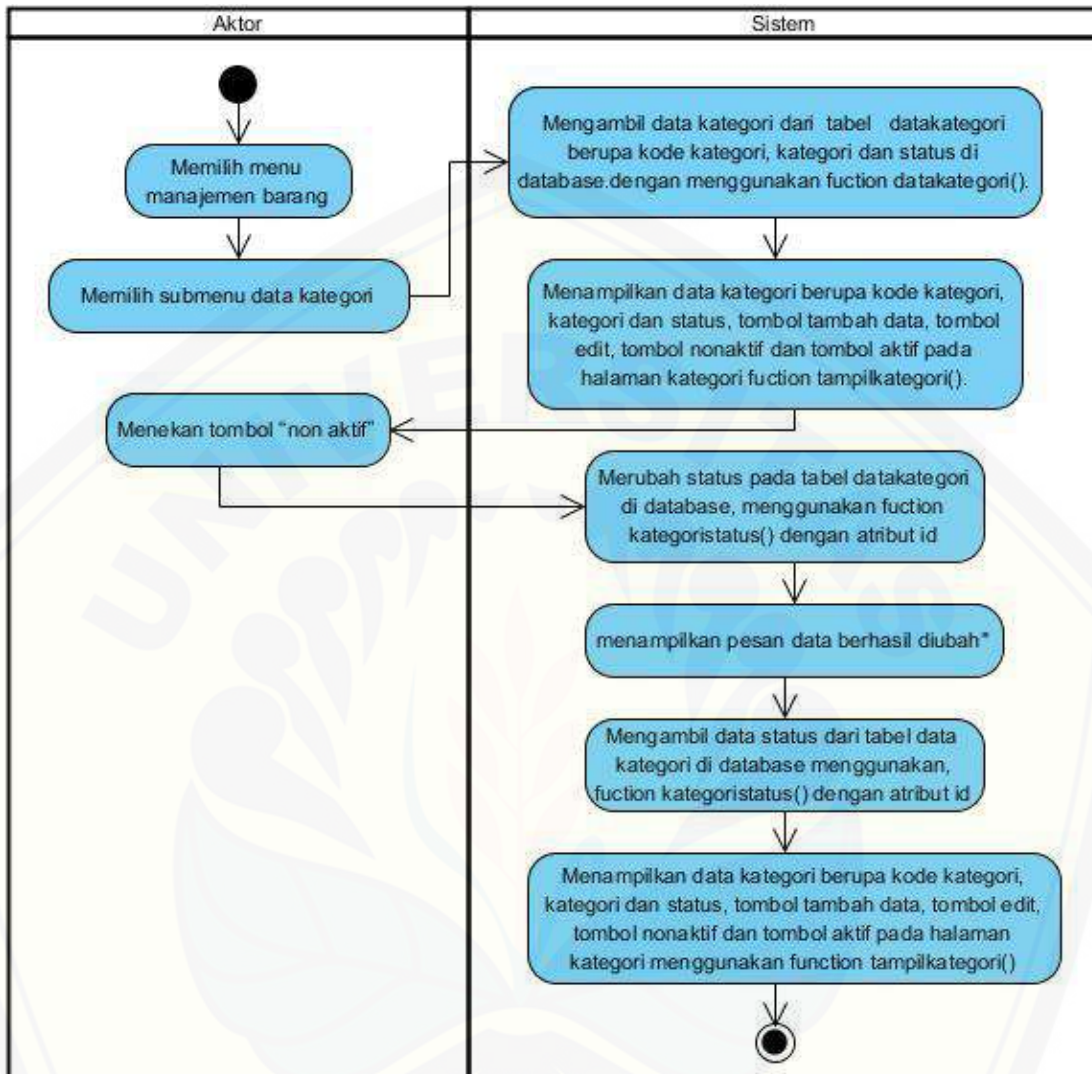
Gambar B.1 Activity Diagram Manajemen Data kategori (view)

B2. Activity Diagram Manajemen Data Kategori (input)



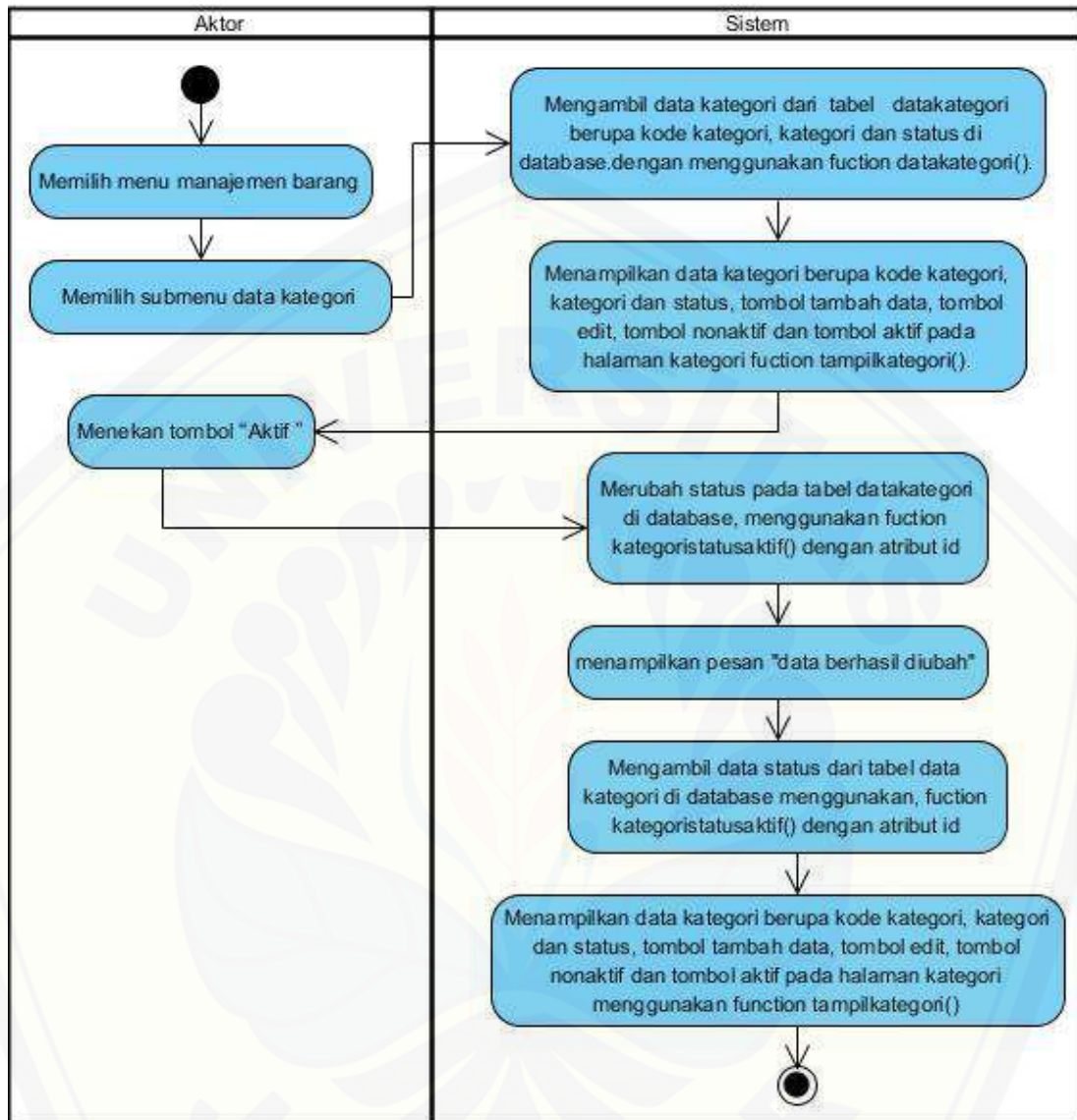
Gambar B.2 Activity Diagram Manajemen Data Kategori (input)

B3. Activity Diagram Merubah Status Aktif Data Kategori



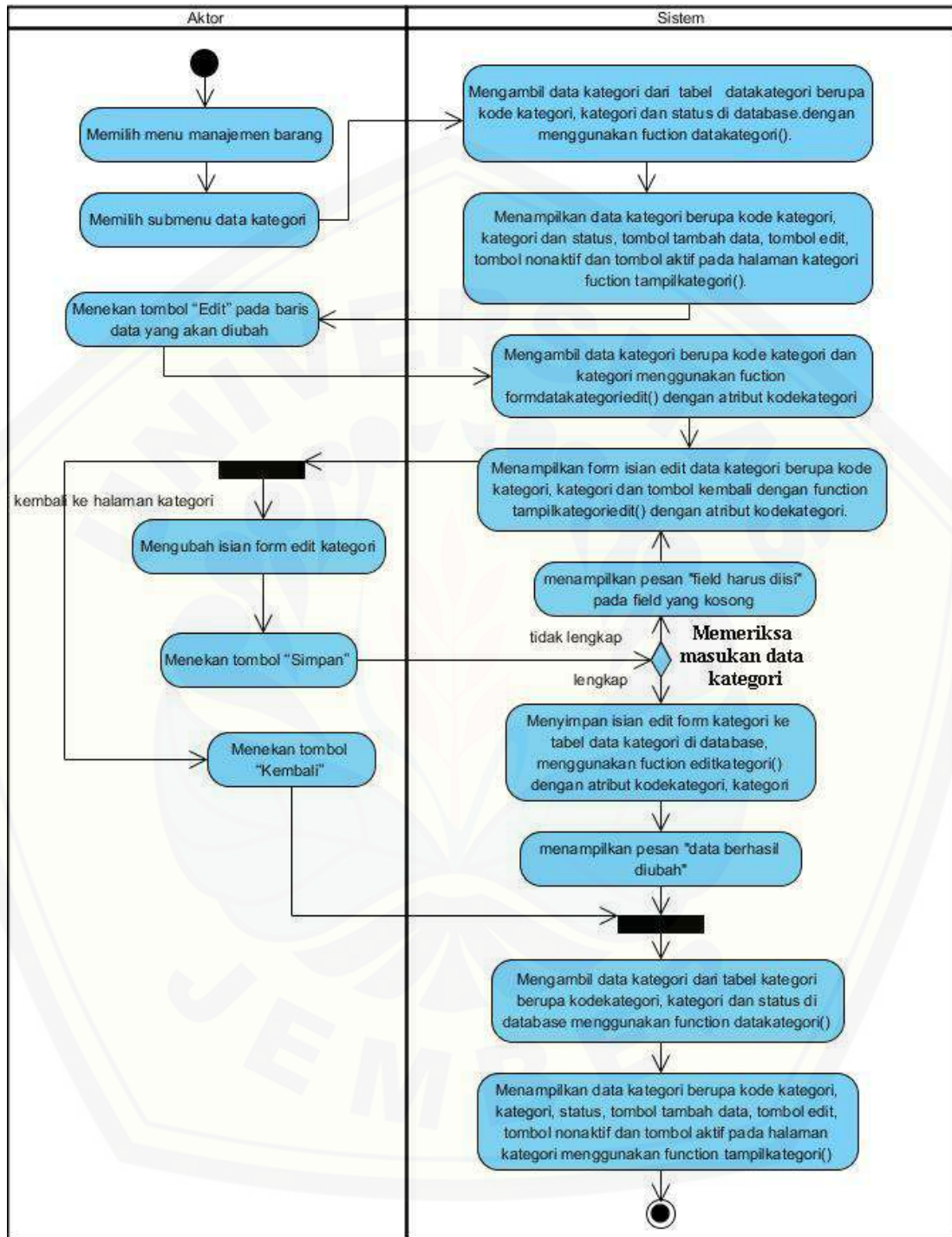
Gambar B.3 Activity Diagram Merubah Status Aktif Data Kategori

B4. Activity Diagram Merubah Status Non Aktif Data Kategori



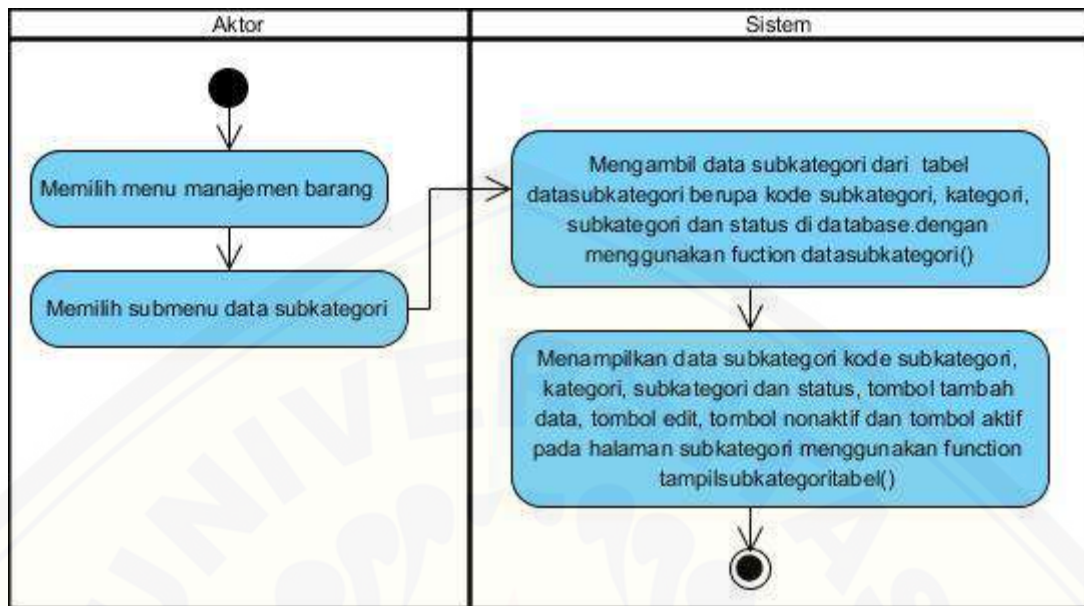
Gambar B.4 Activity Diagram Merubah Status Non Aktif Data Kategori

B5. Activity Diagram Manajemen Data Kategori (edit)



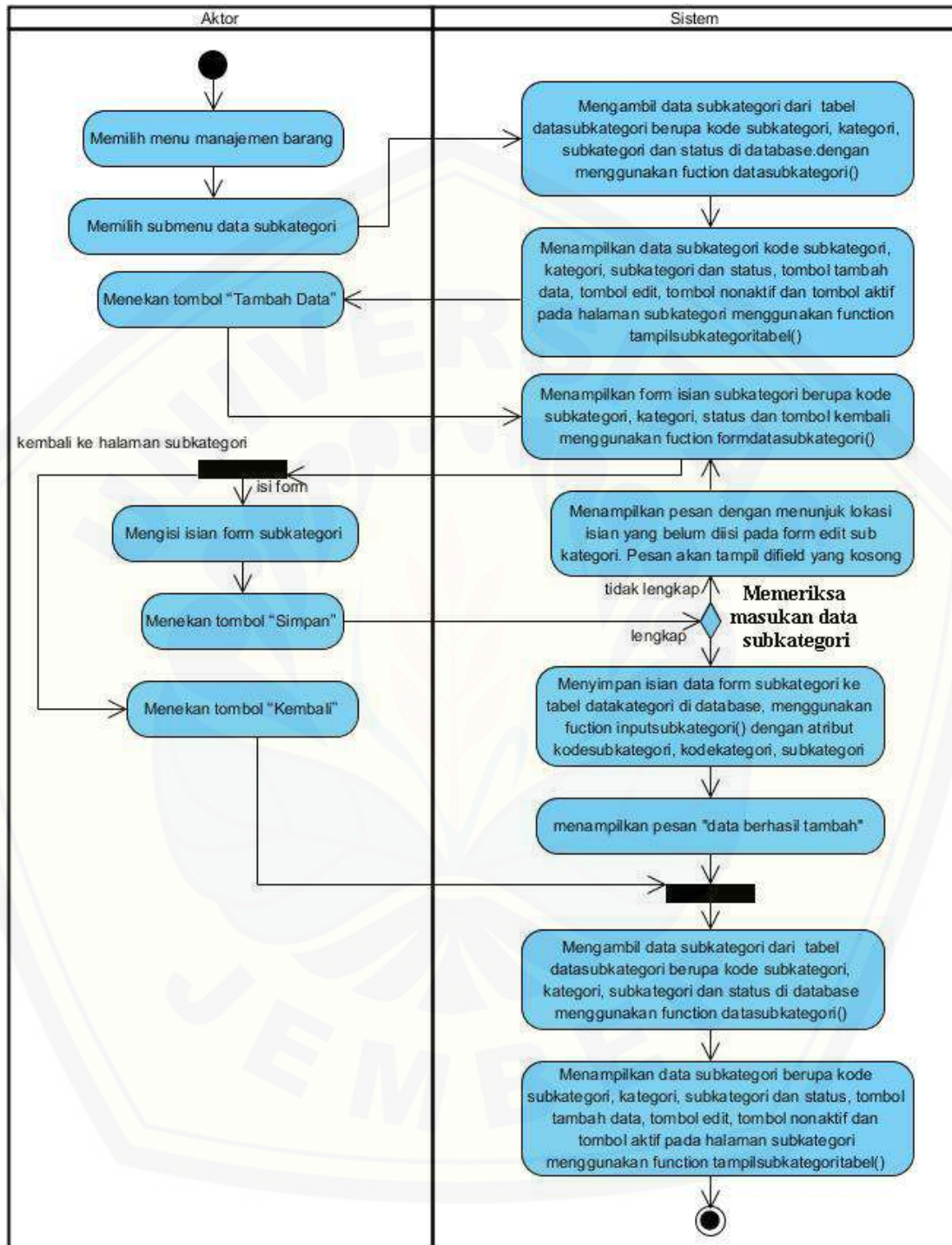
Gambar B.5 Activity Diagram Manajemen Data Kategori (edit)

B6. Activity Diagram Manajemen Data Subkategori (view)



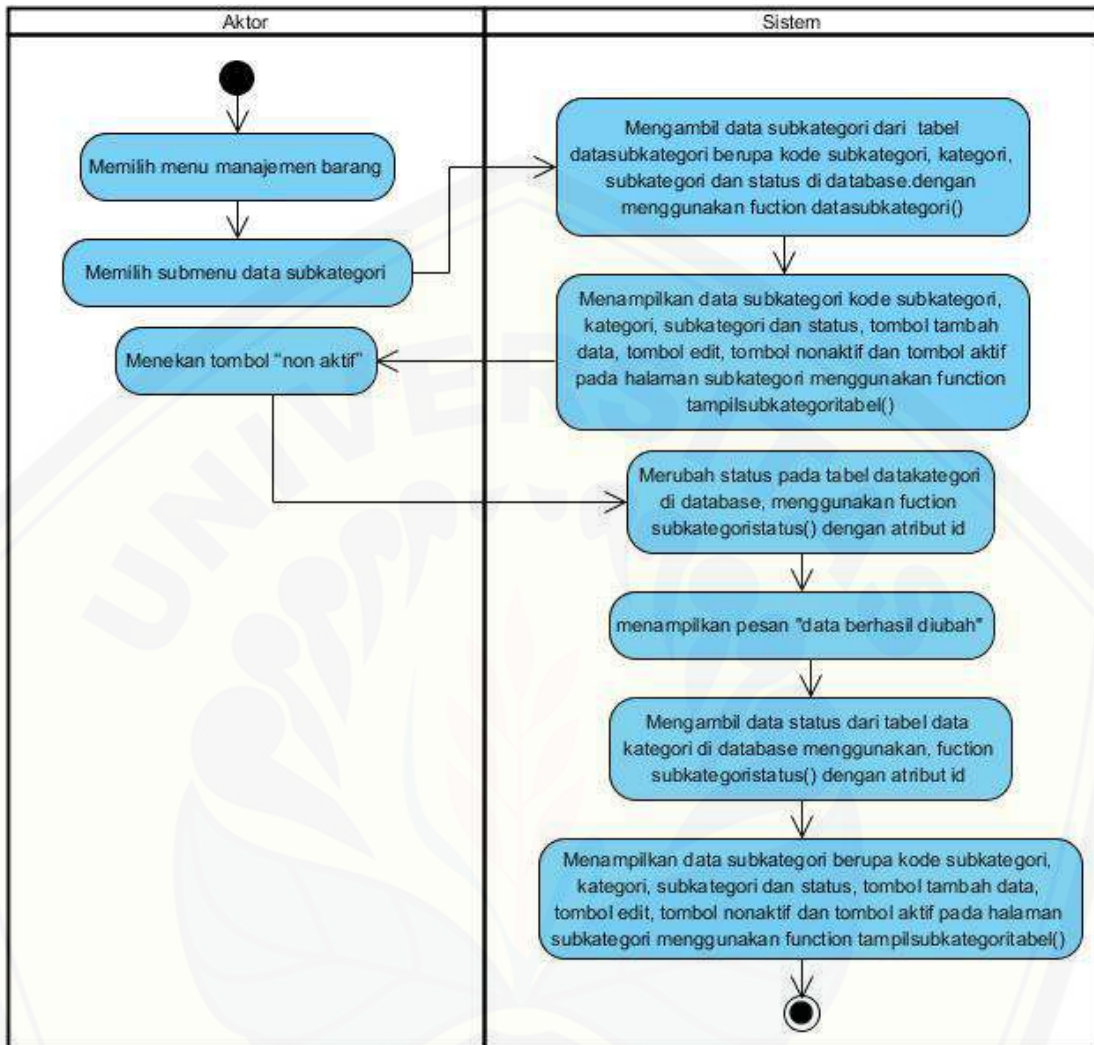
Gambar B.6 Activity Diagram Manajemen Data Subkategori (view)

B7. Activity Diagram Manajemen Data Subkategori (input)



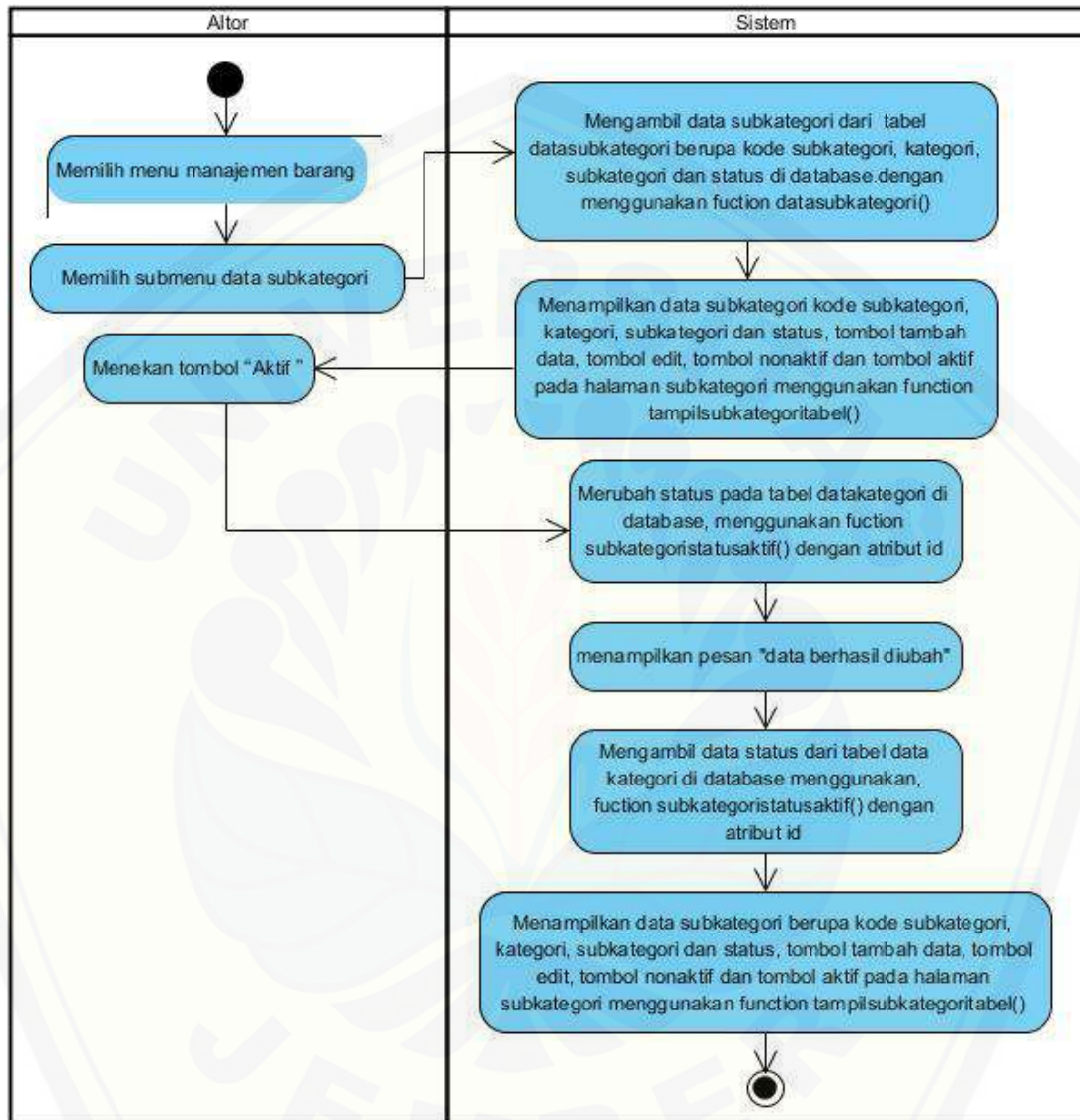
Gambar B.7 Activity Diagram Manajemen Data Subkategori (input)

B8. Activity Diagram Merubah Status Aktif Data Subkategori



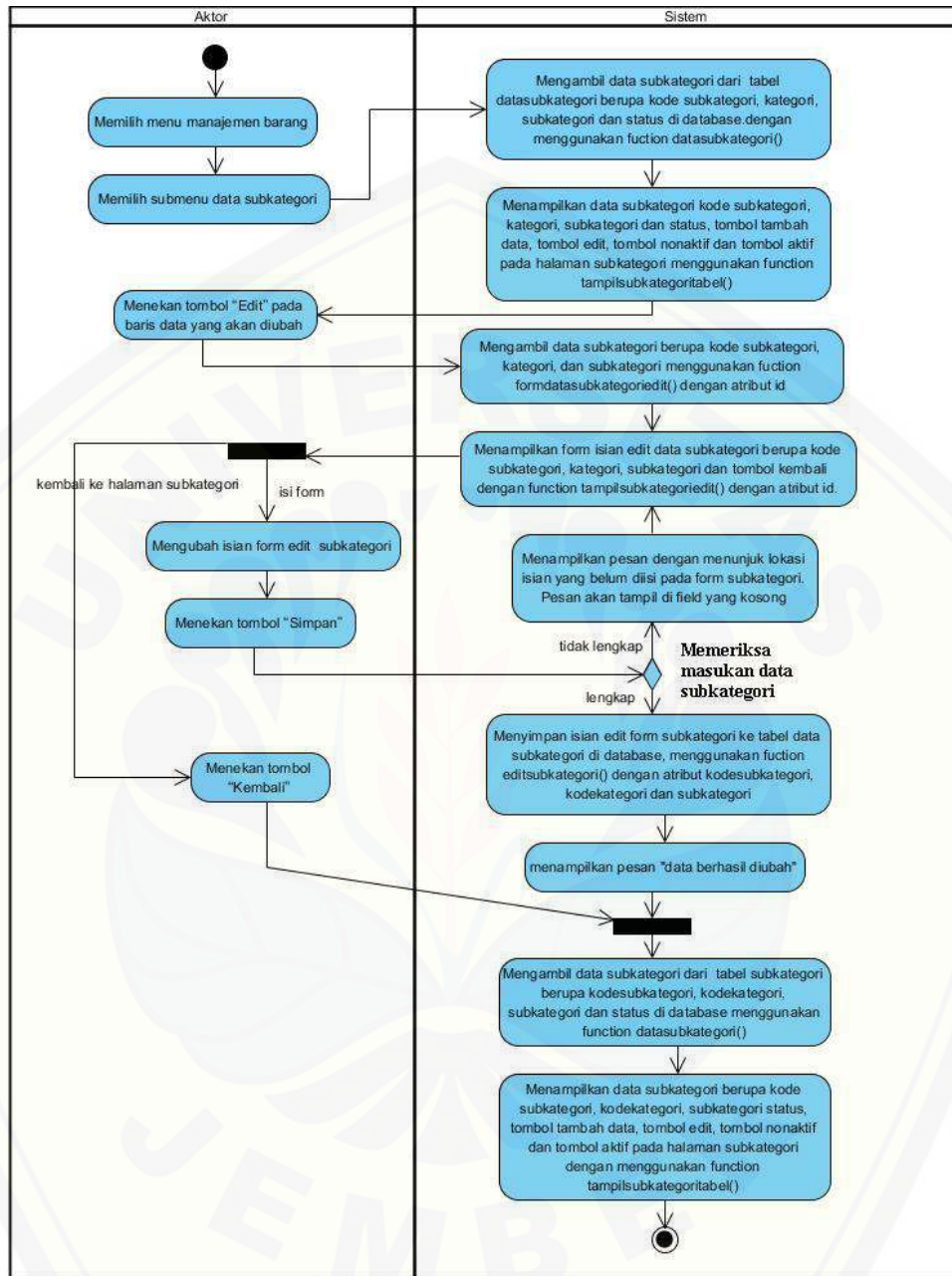
Gambar B.8 Activity Diagram Merubah Status Aktif Data Subkategori

B9. Activity Diagram Merubah Status Non Aktif Data Subkategori



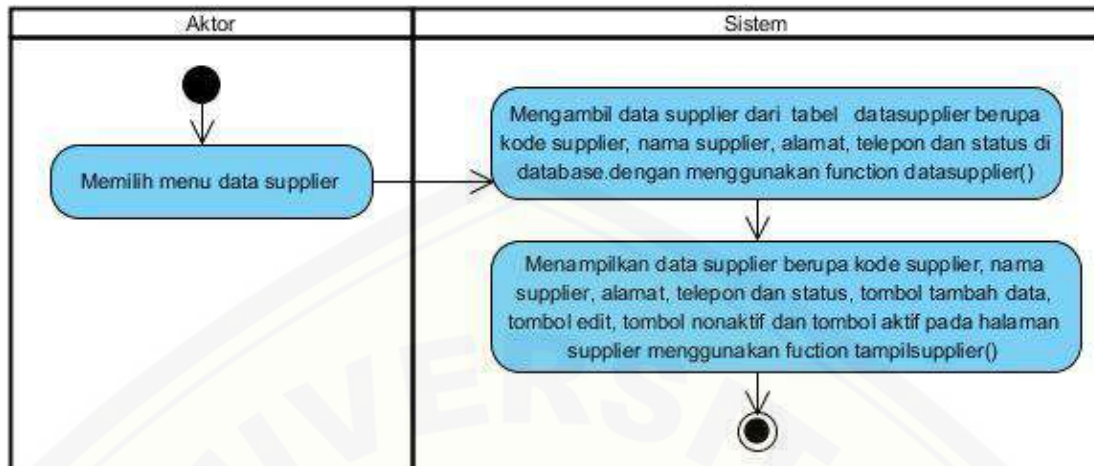
Gambar B.9 Activity Diagram Merubah Status Non Aktif Data Subkategori

B10. Activity Diagram Manajemen Data Subkategori (edit)



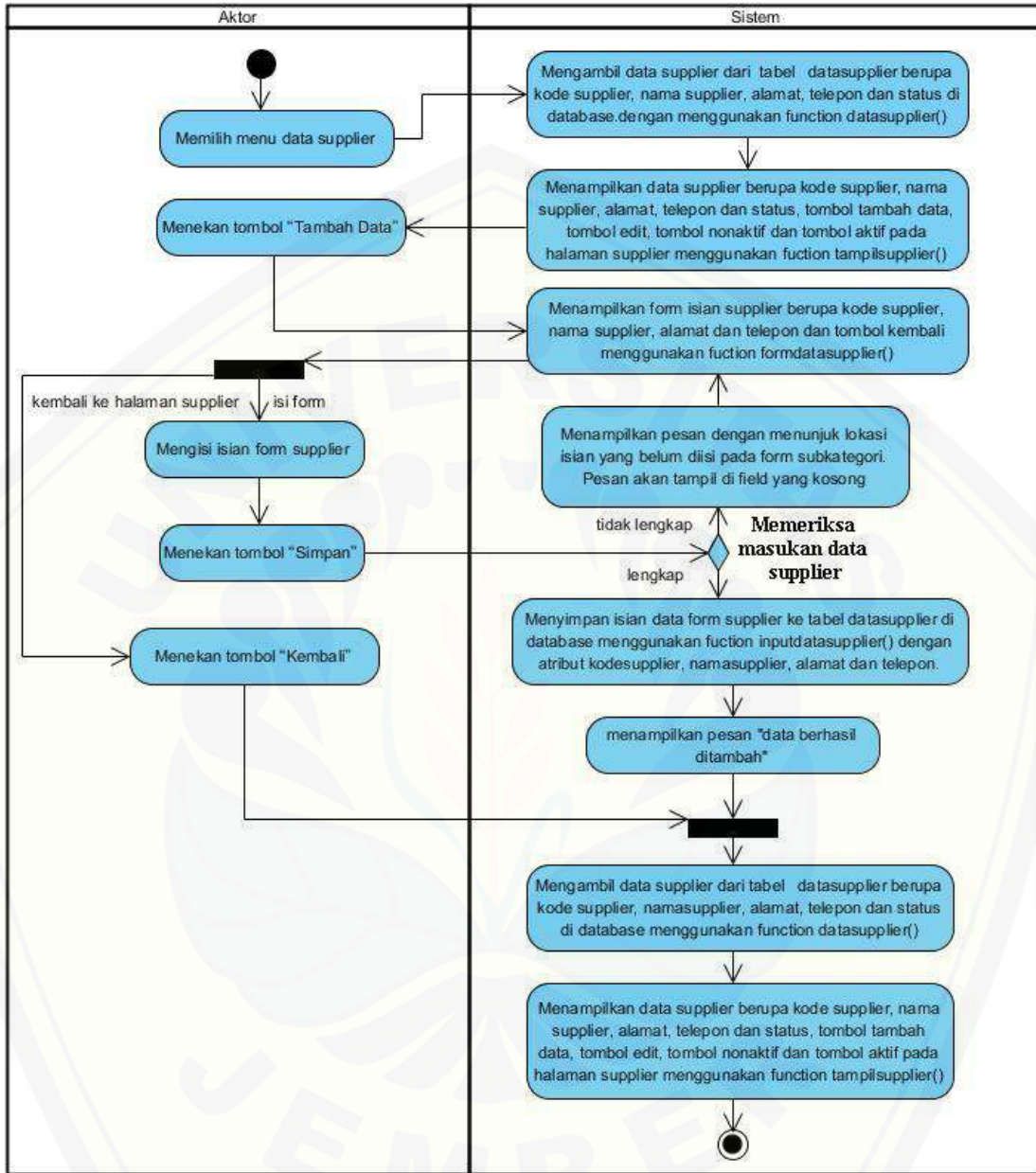
Gambar B.10 Activity Diagram Manajemen Data Subkategori (edit)

B11. Activity Diagram Manajemen Data Supplier (view)



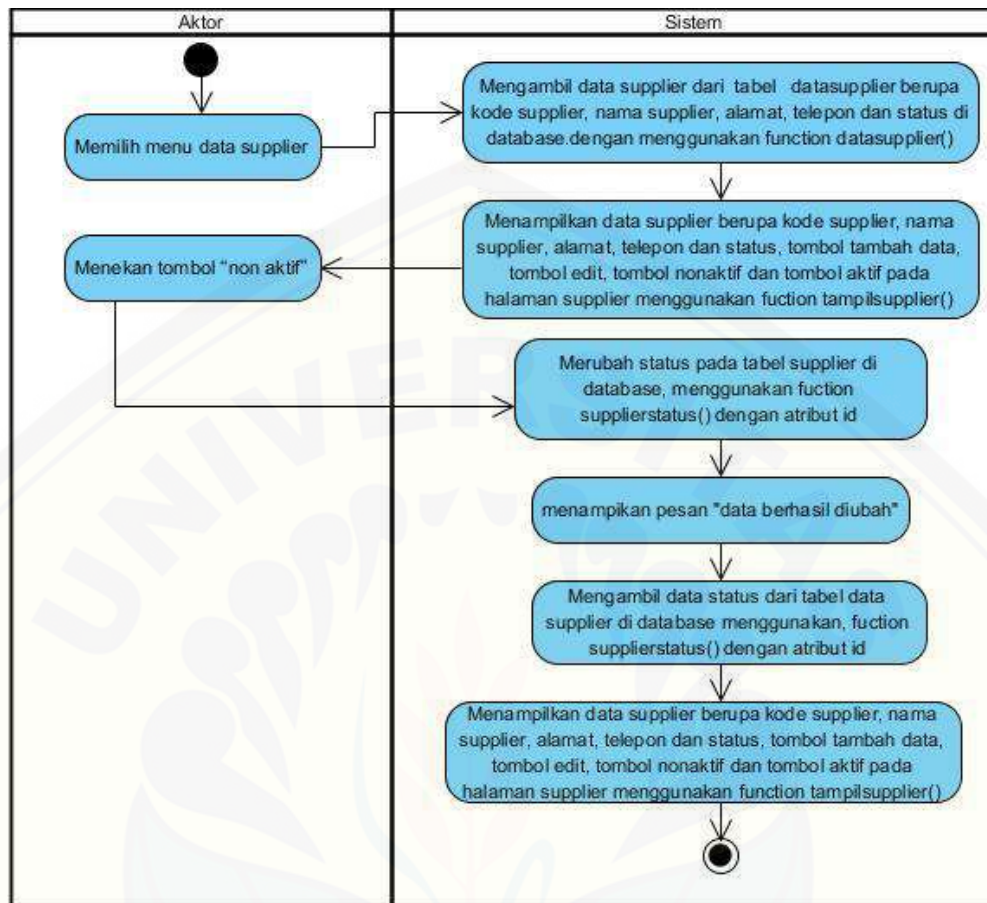
Gambar B.11 Activity Diagram Manajemen Data Supplier (view)

B12. Activity Diagram Manajemen Data Supplier (input)



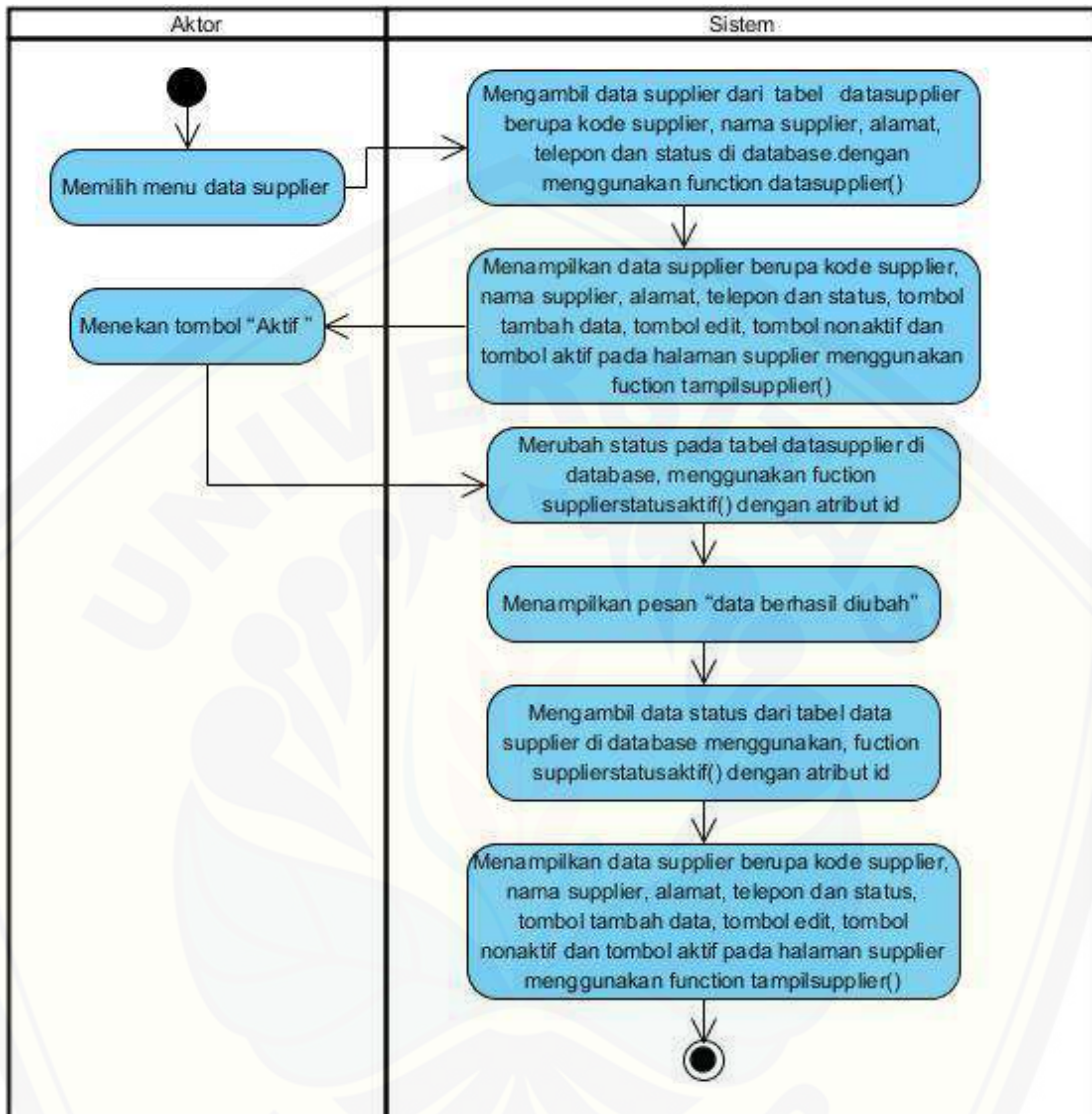
Gambar B.12 Activity Diagram Manajemen Data Supplier (input)

B13. Activity Diagram Merubah Status Aktif Data Supplier



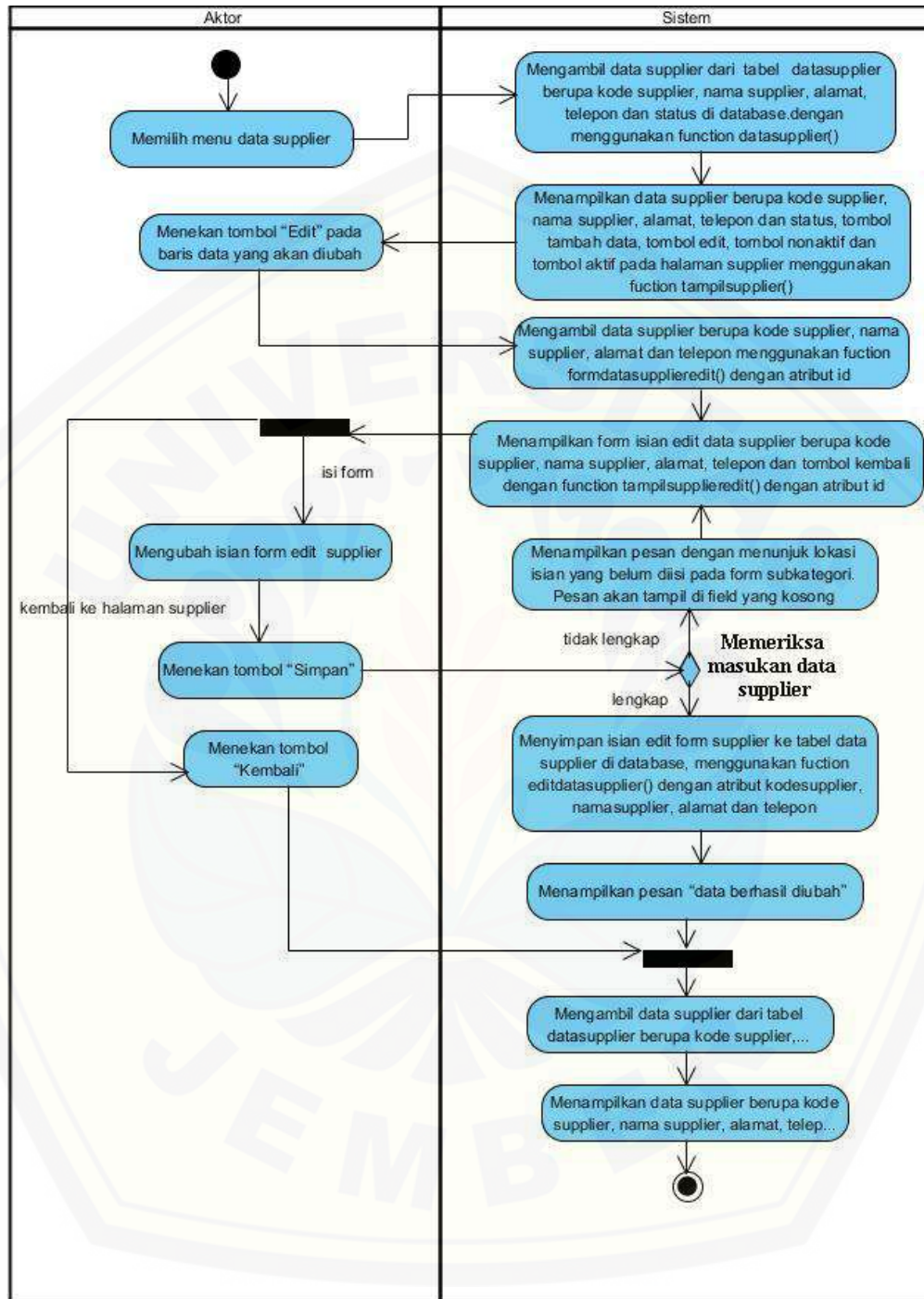
Gambar B.13 Activity Diagram Merubah Status Aktif Data Supplier

B14. Activity Diagram Merubah Status Non Aktif Data Supplier



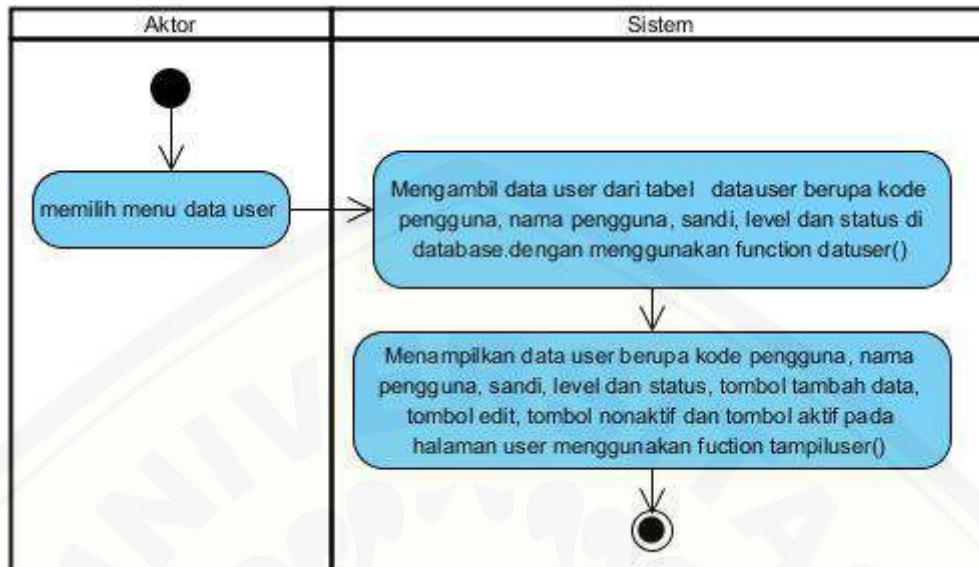
Gambar B.14 Activity Diagram Merubah Status Non Aktif Data Supplier

B15. Activity Diagram Manajemen Data Supplier (*edit*)



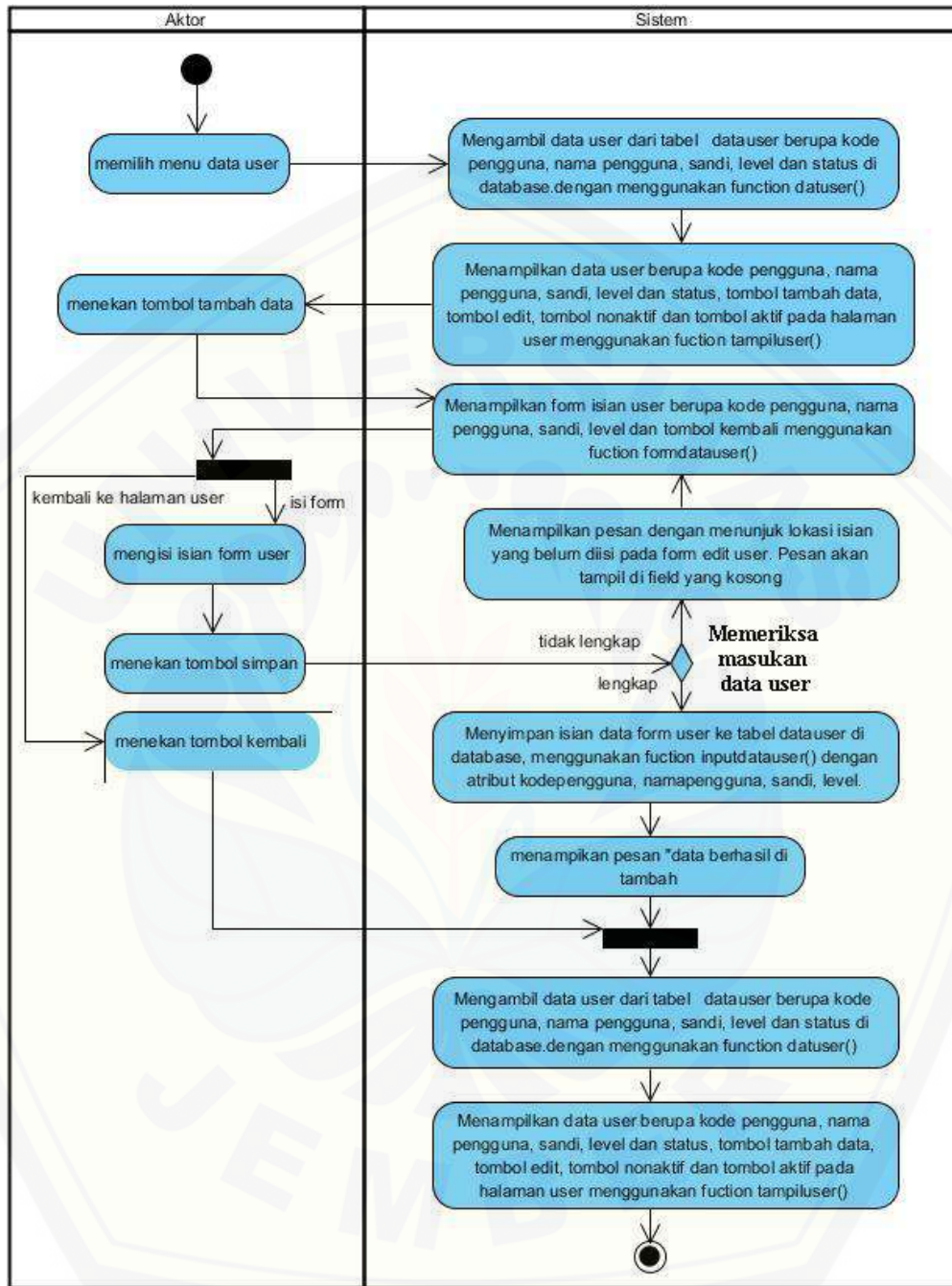
Gambar B.15 Activity Diagram Manajemen Data Supplier (edit)

B16. Activity Diagram Manajemen Data User (view)



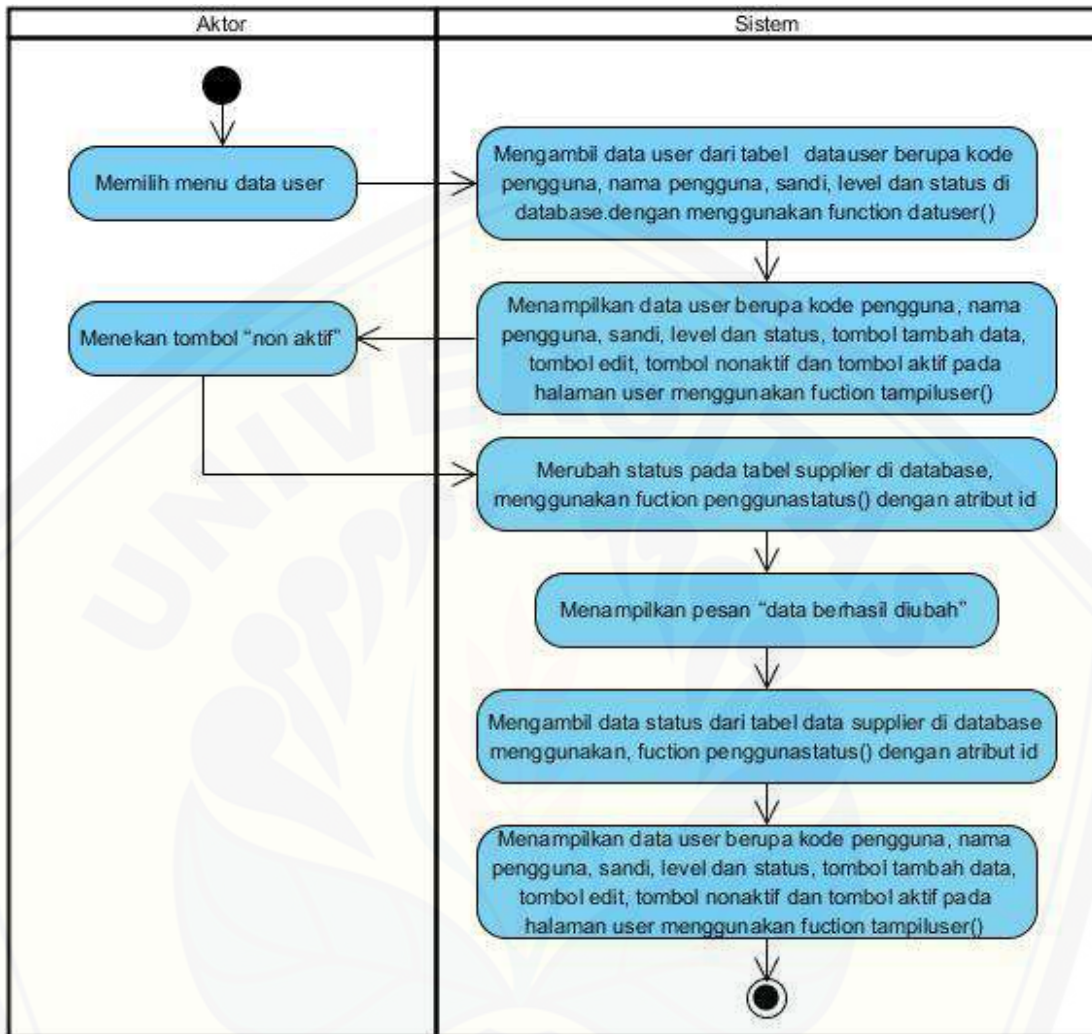
Gambar B.16 Activity Diagram Manajemen Data User (view)

B17. Activity Diagram Manajemen Data User (*input*)



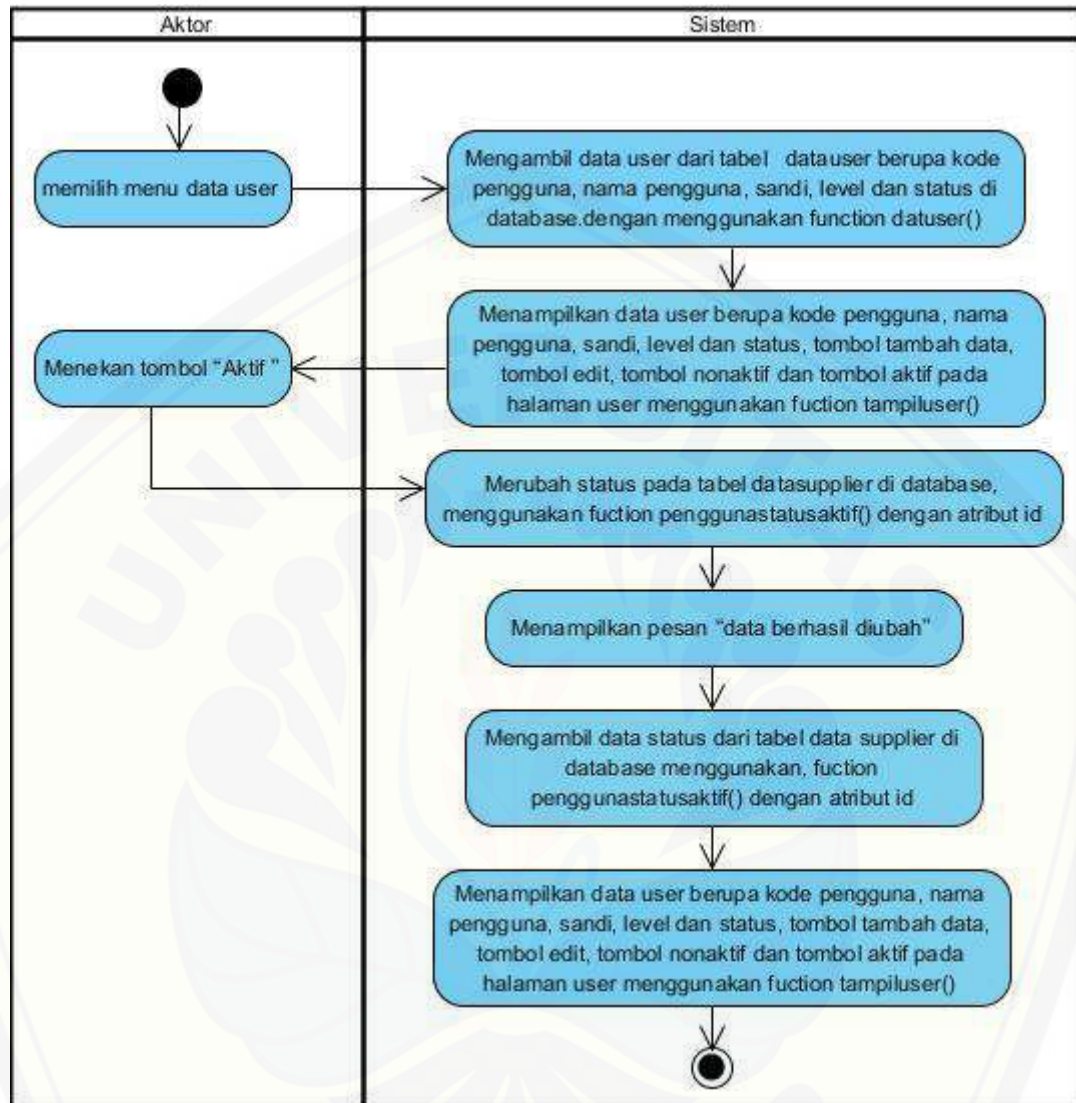
Gambar B.17 Activity Diagram Manajemen Data User (*input*)

B18. Activity Diagram Merubah Status Aktif Data User



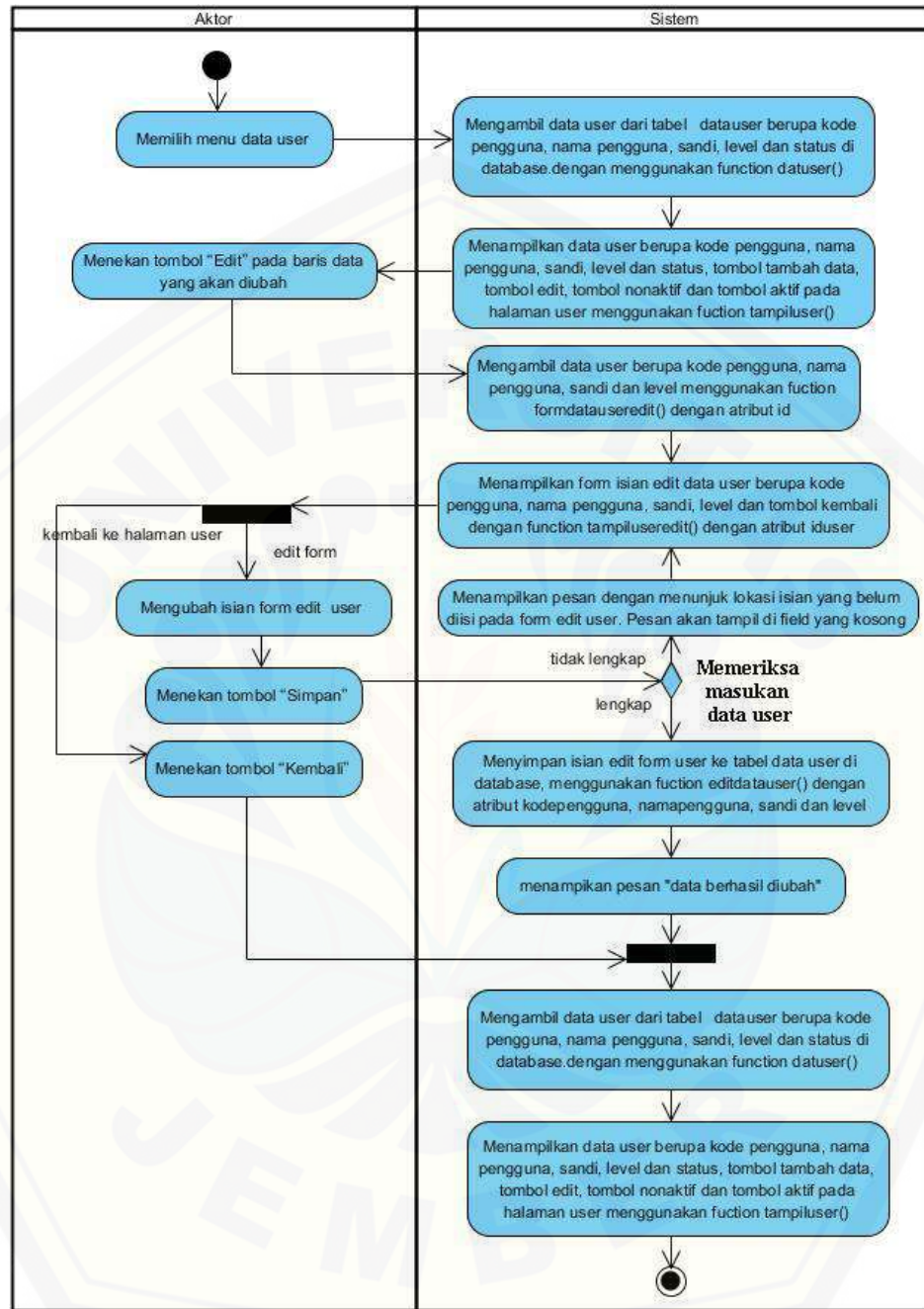
Gambar B.18 Activity Diagram Merubah Status aktif Data User

B19. Activity Diagram Merubah Status Non Aktif Data User



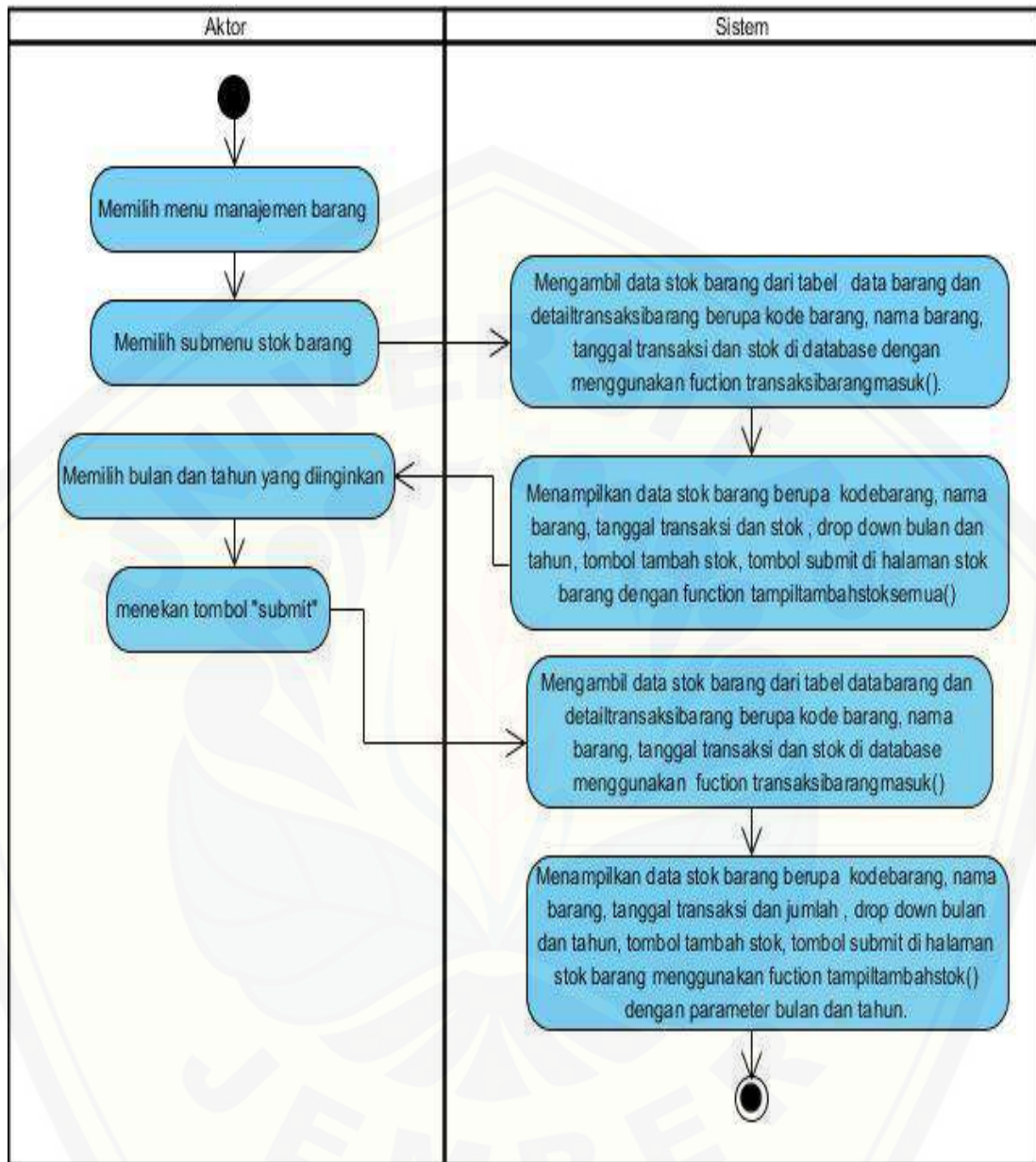
Gambar B.19 Activity Diagram Merubah Status Non Aktif Data User

B20. Activity Diagram Manajemen Data User (*edit*)



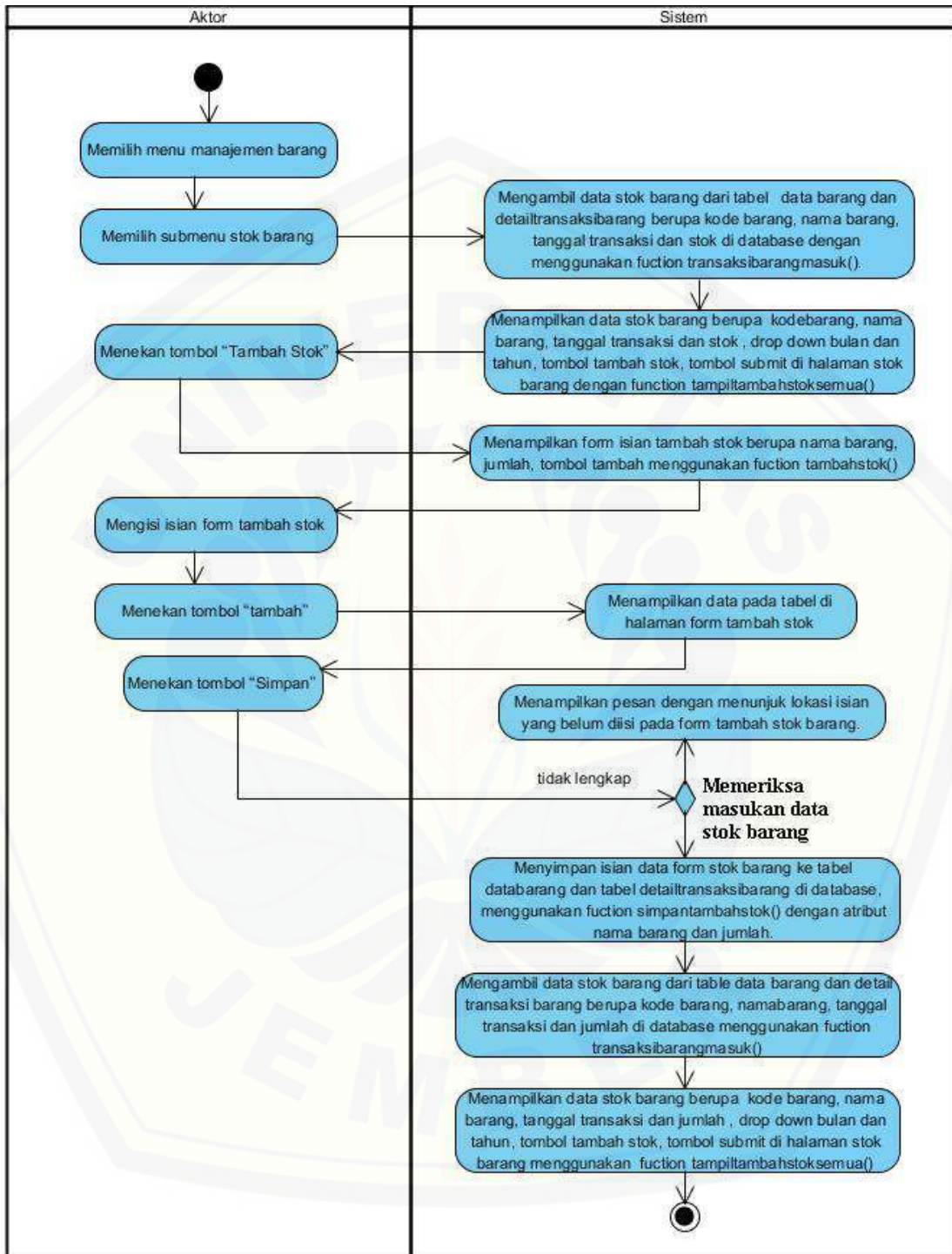
Gambar B.20 Activity Diagram Manajemen Data User (edit)

B21. Activity Diagram Manajemen Stok Barang(view)



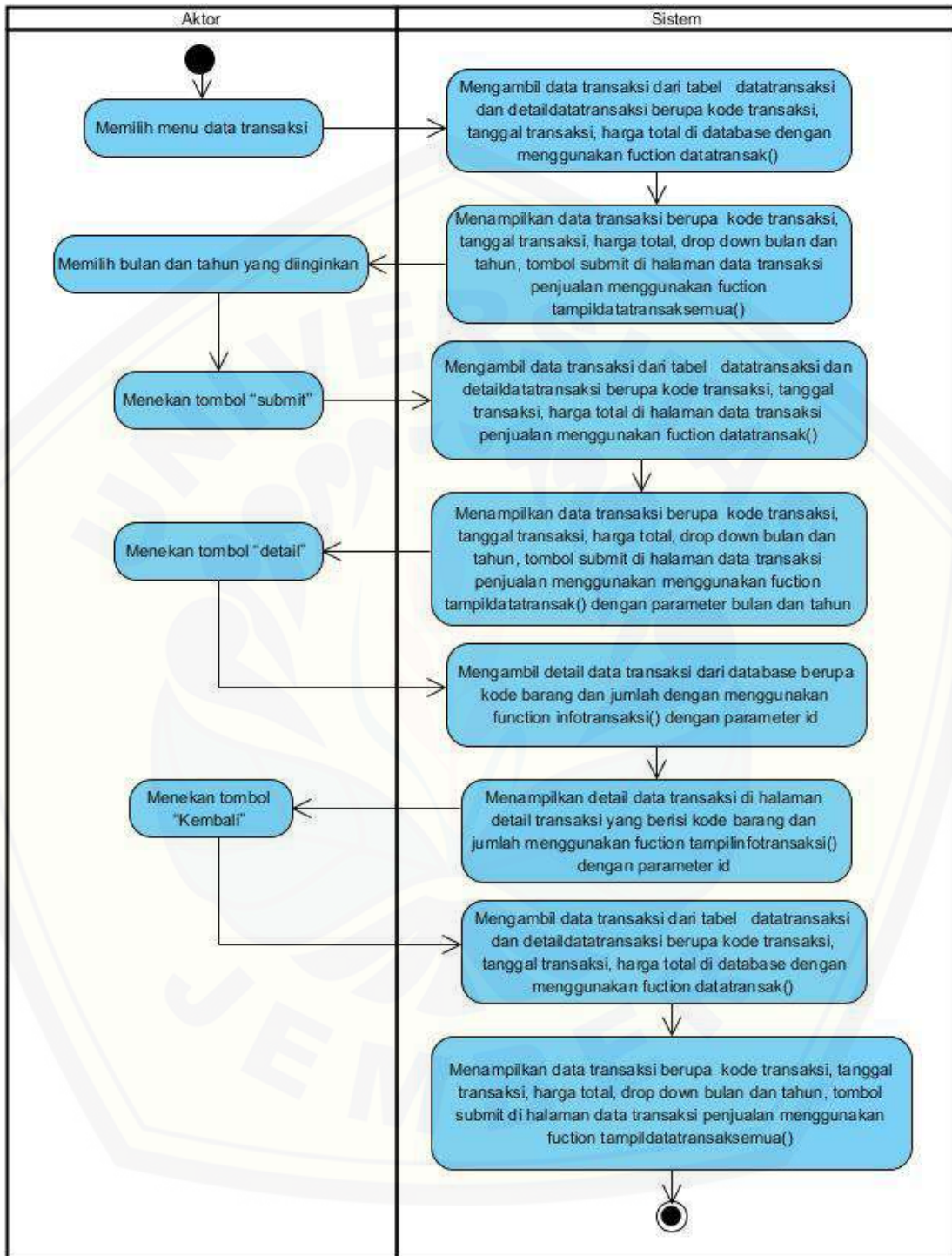
Gambar B.21 Activity Diagram Manajemen Stok Barang (view)

B22. Activity Diagram Manajemen Stok Barang (input)



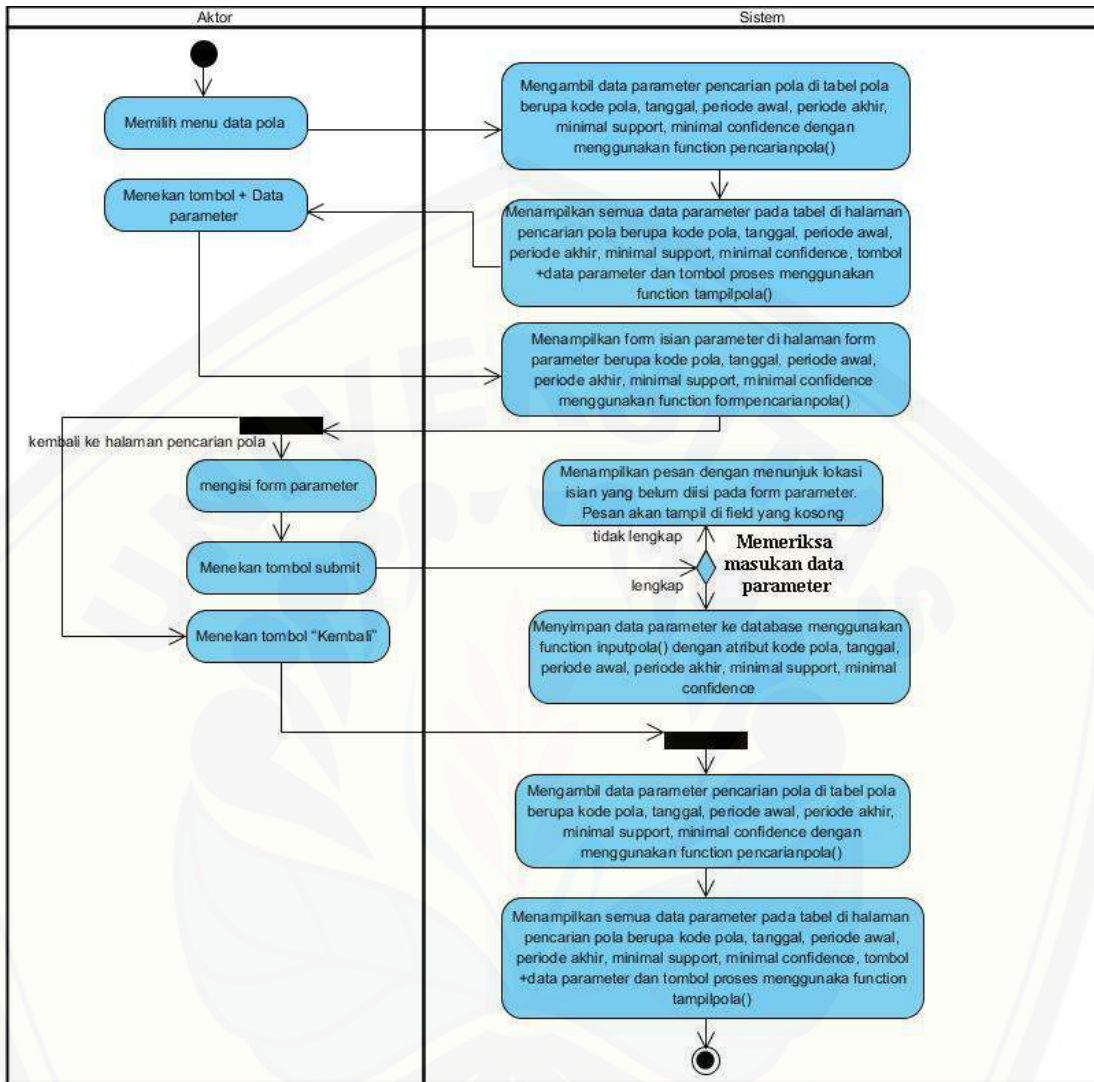
Gambar B.22 Activity Diagram Manajemen Stok Barang (input)

B23. Activity Diagram Melihat Data Transaksi (admin)



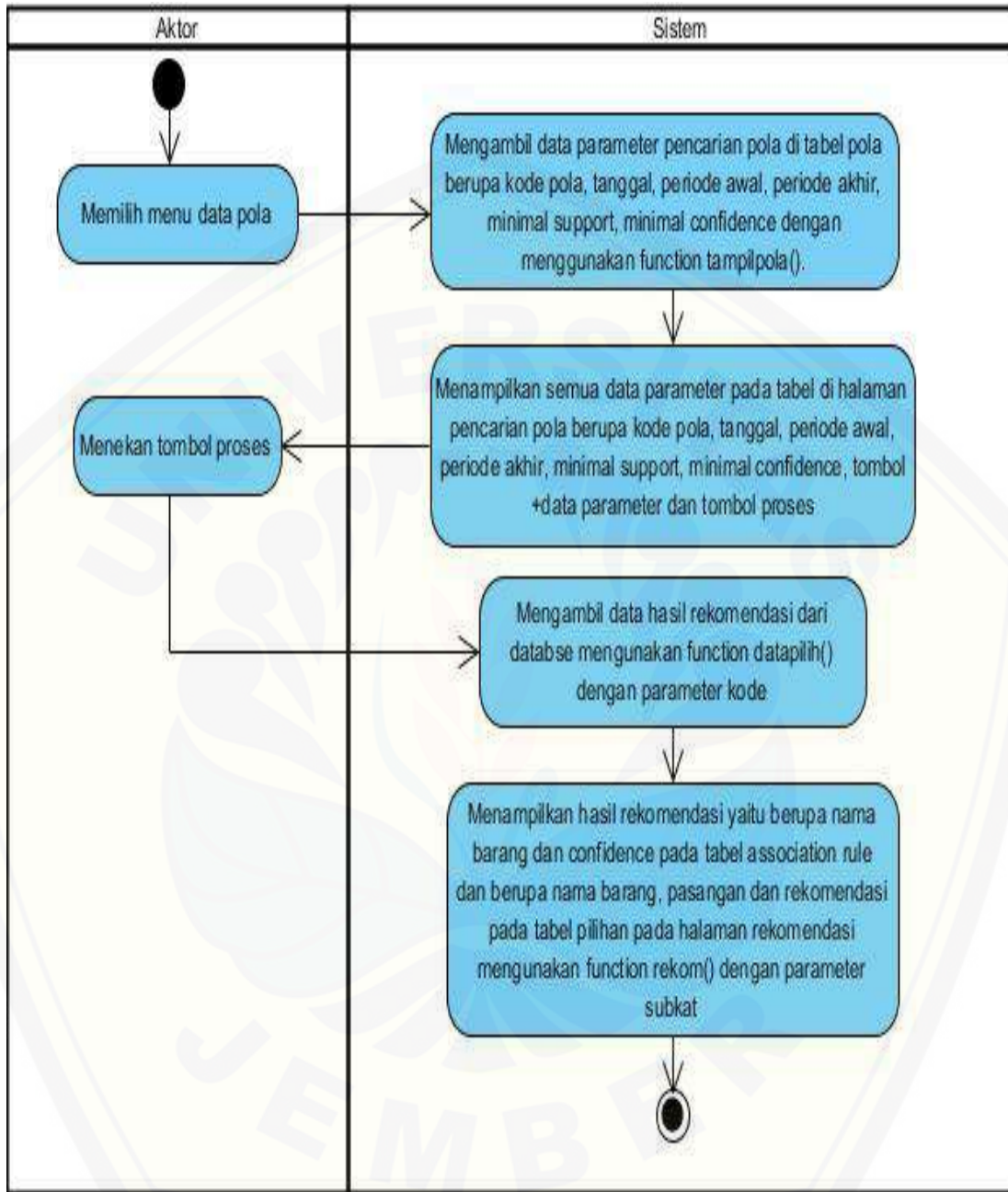
Gambar B.23 Activity Diagram Melihat Data Transaksi (view)

B24. Activity Diagram Input Parameter Pencarian Pola



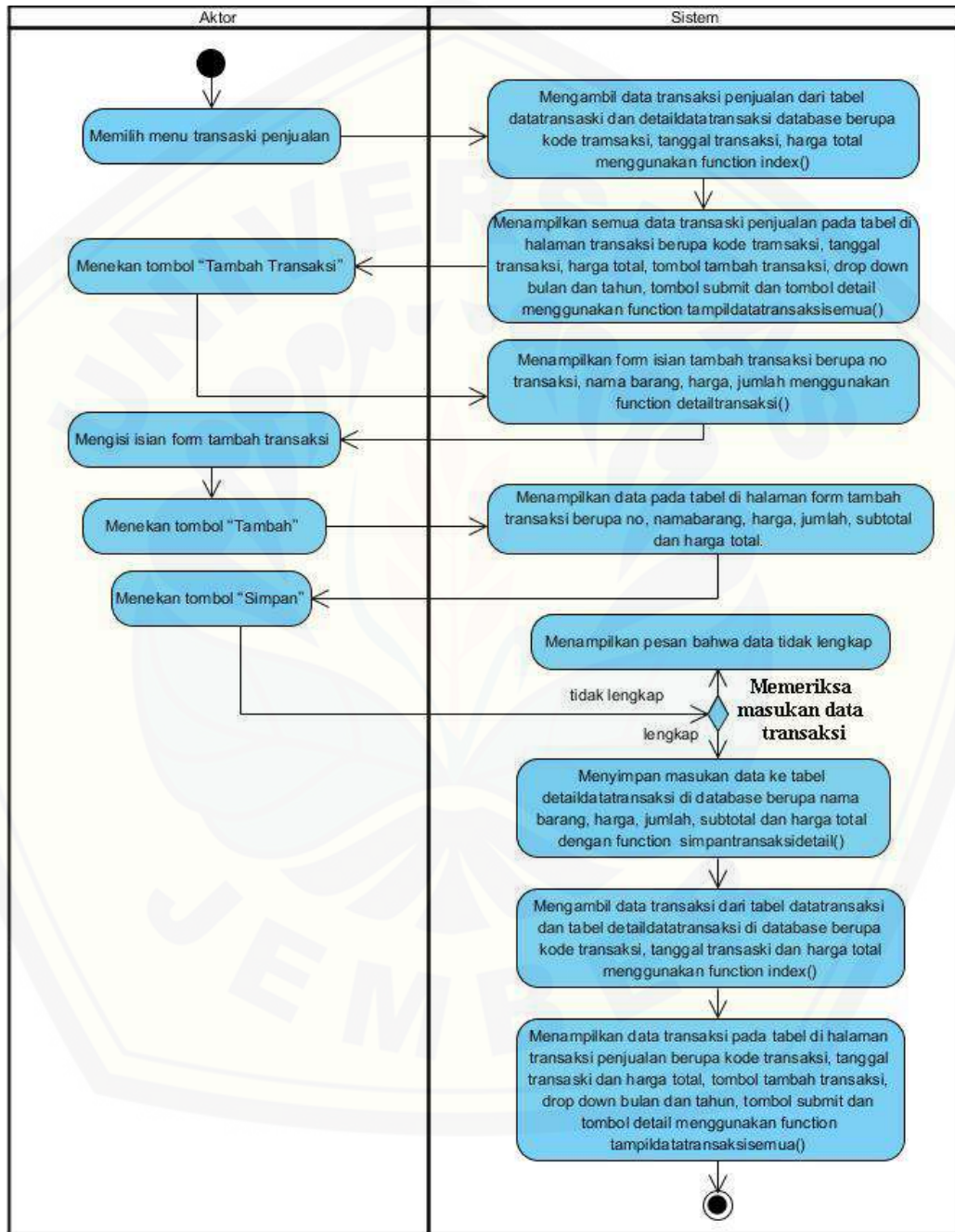
Gambar B.24 Activity Diagram Input Parameter Pencarian Pola

B25. Activity Diagram Rekomendasi Promosi



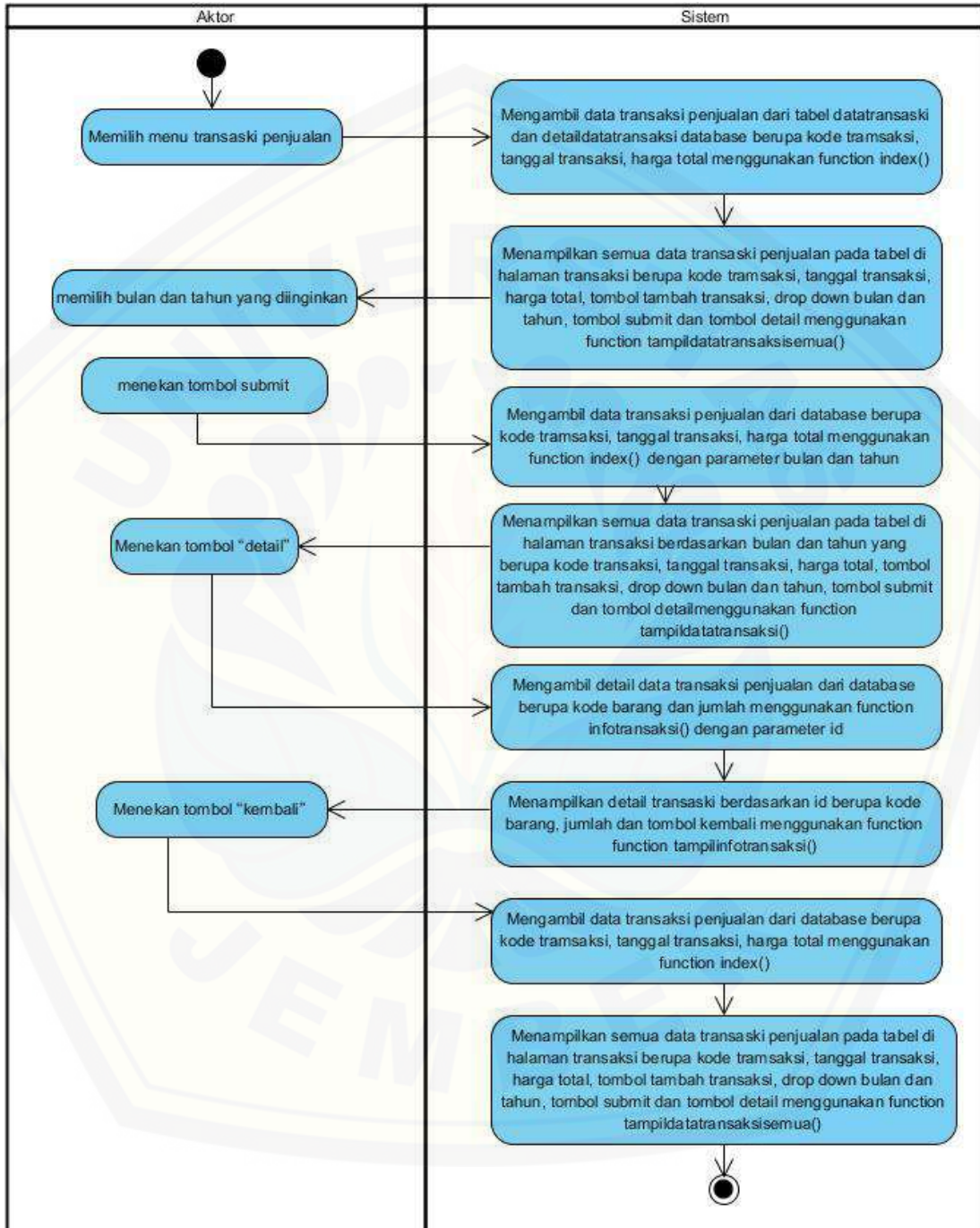
Gambar B. 25 Activity Diagram Rekomendasi Promosi

B26. Activity Diagram Input Transaksi Penjualan



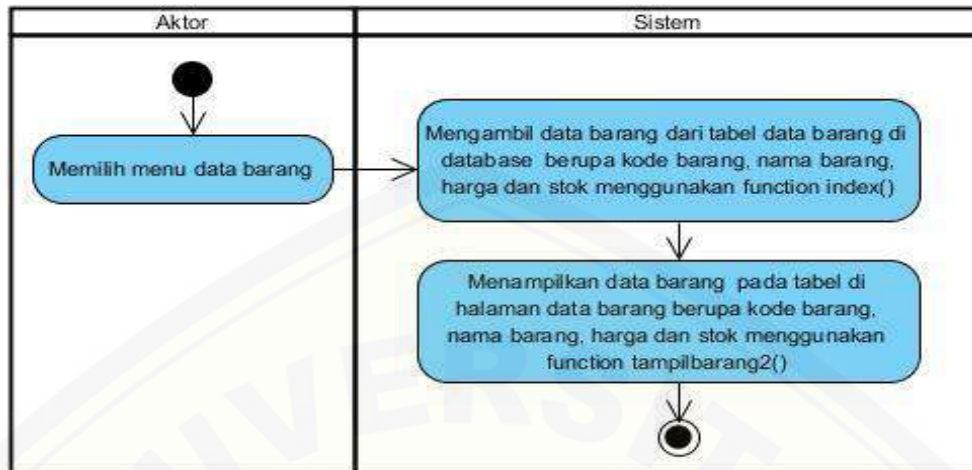
Gambar B.26 Activity Diagram Input Transaksi Penjualan

B27. Activity Diagram Melihat Data Transaksi (pegawai)



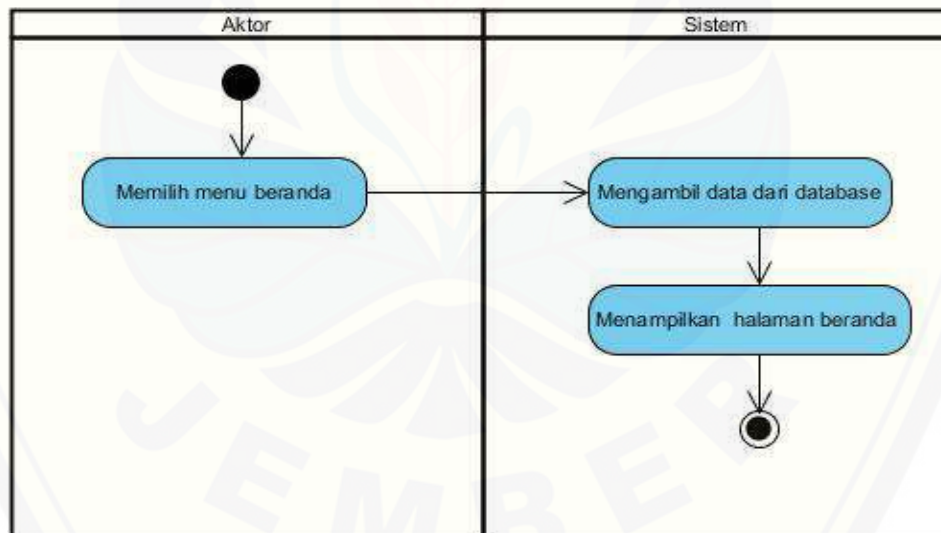
Gambar B.27 Activity Diagram Melihat Data Transaksi (pegawai)

B28. Activity Diagram Melihat Data Barang (pegawai)



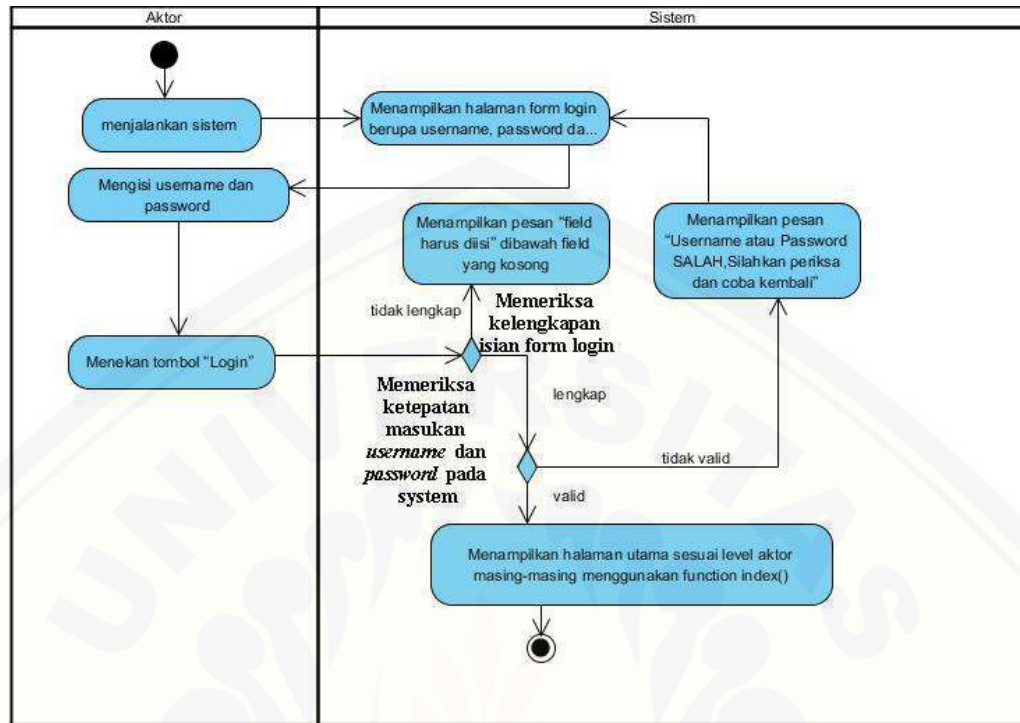
Gambar B.28 Activity Diagram Melihat Data Barang (pegawai)

B29. Activity Diagram Melihat Beranda (Admin)



Gambar B.29 Activity Diagram Melihat Beranda (Admin)

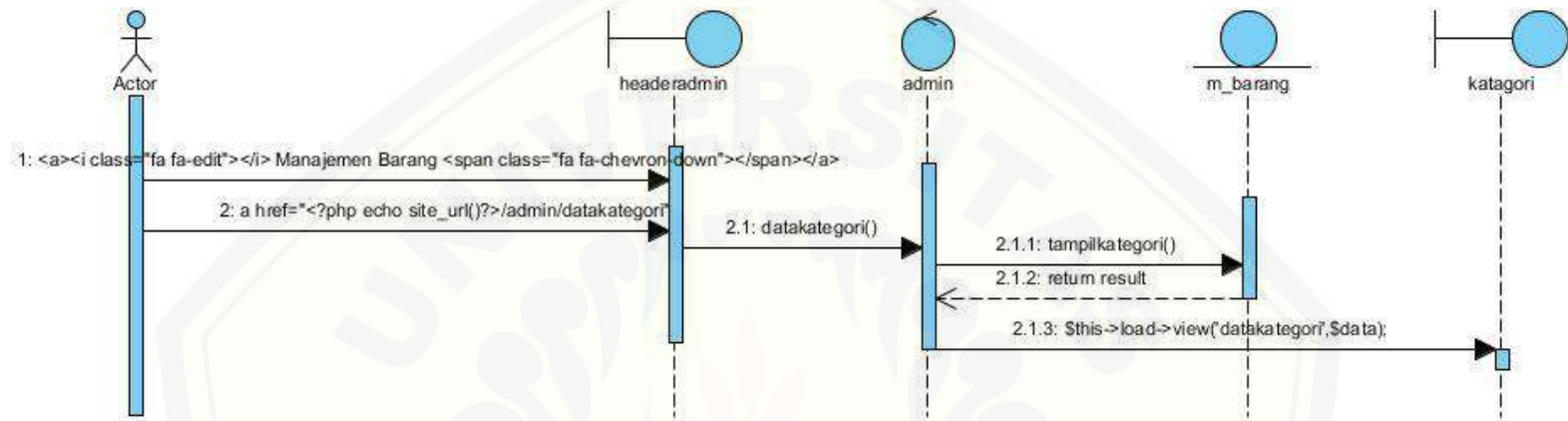
B30. Activity Diagram Login



Gambar B.30 Activity Diagram Login

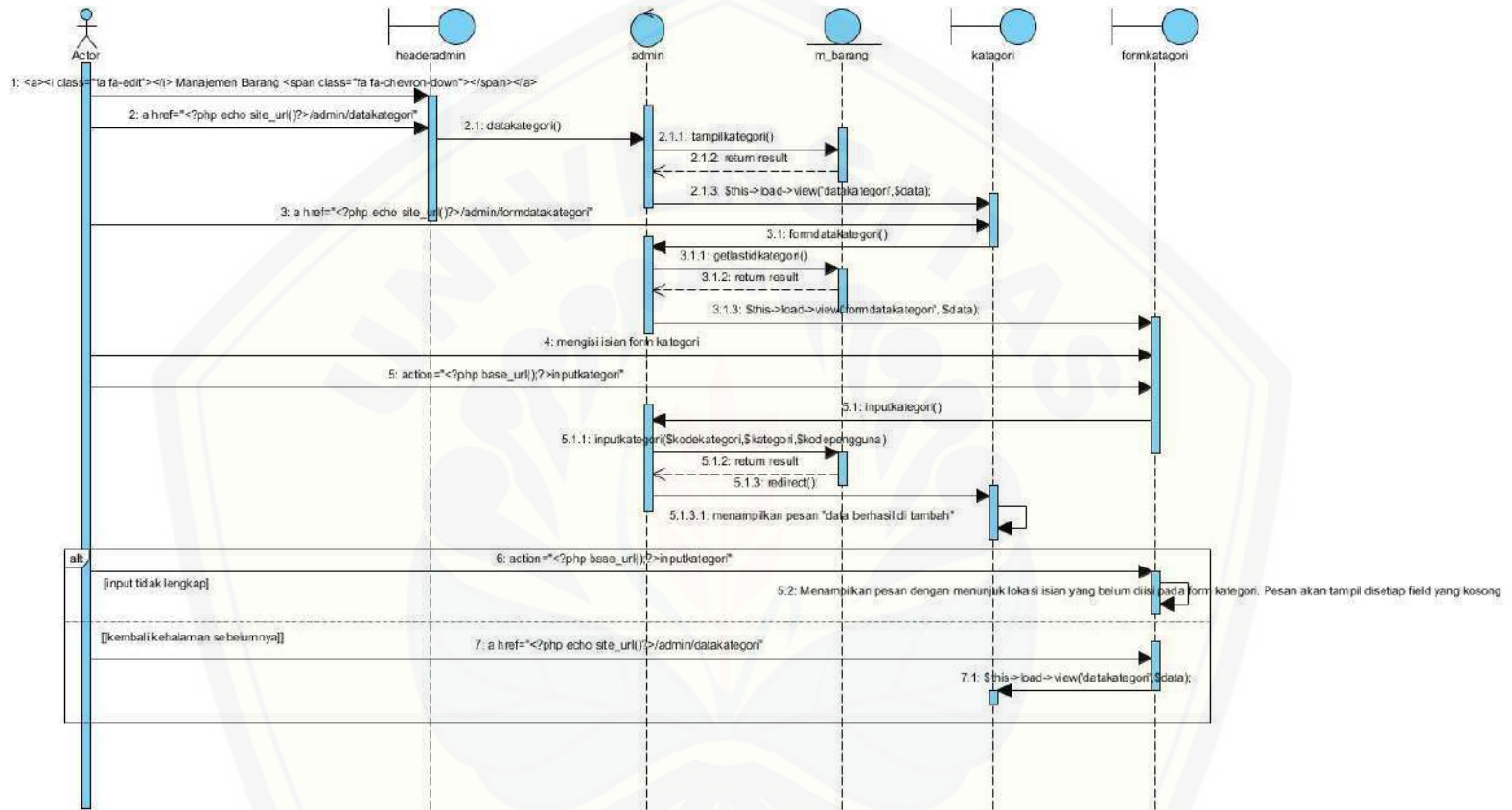
Lampiran C. Sequence Diagram

C1. Sequence Diagram Manajemen Data Kategori (view)



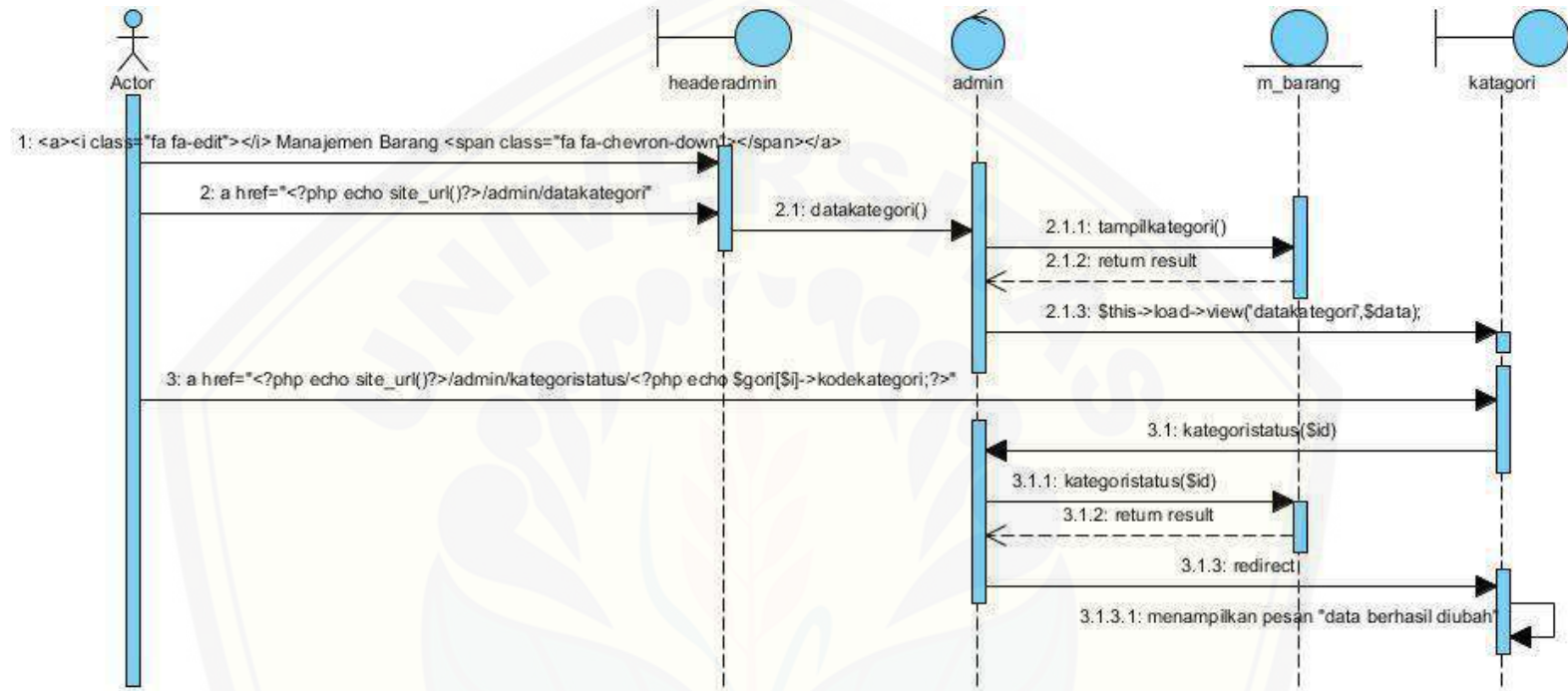
Gambar C.1 Sequence Diagram Manajemen Data Kategori (view)

C2. Sequence Diagram Manajemen Data Kategori (input)



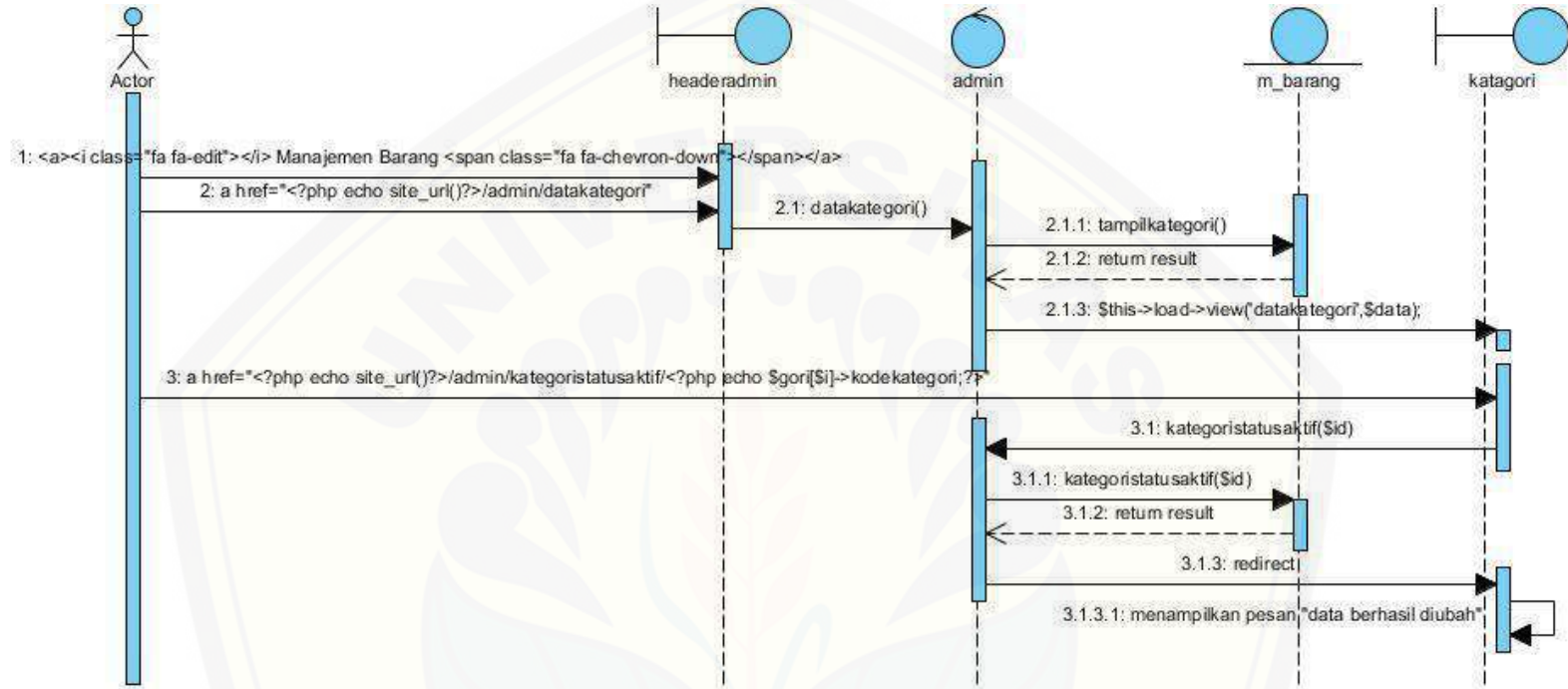
Gambar C.2 Sequence Diagram Manajemen Data Kategori (input)

C3. Sequence Diagram Merubah Status Aktif Data Kategori



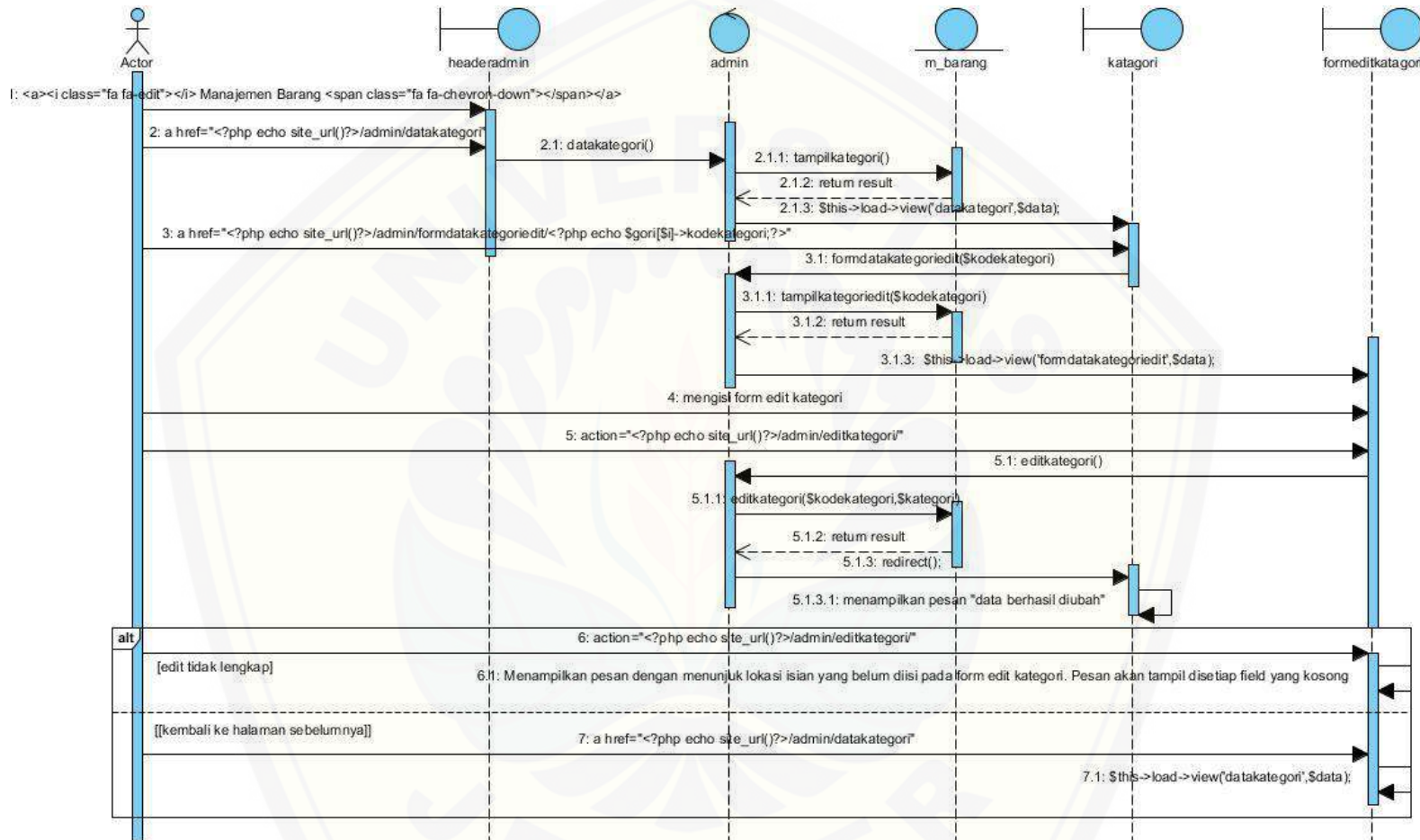
Gambar C.3 Sequence Diagram Merubah Status Aktif Data Kategori

C4. Sequence Diagram Merubah Status Non Aktif Data Kategori



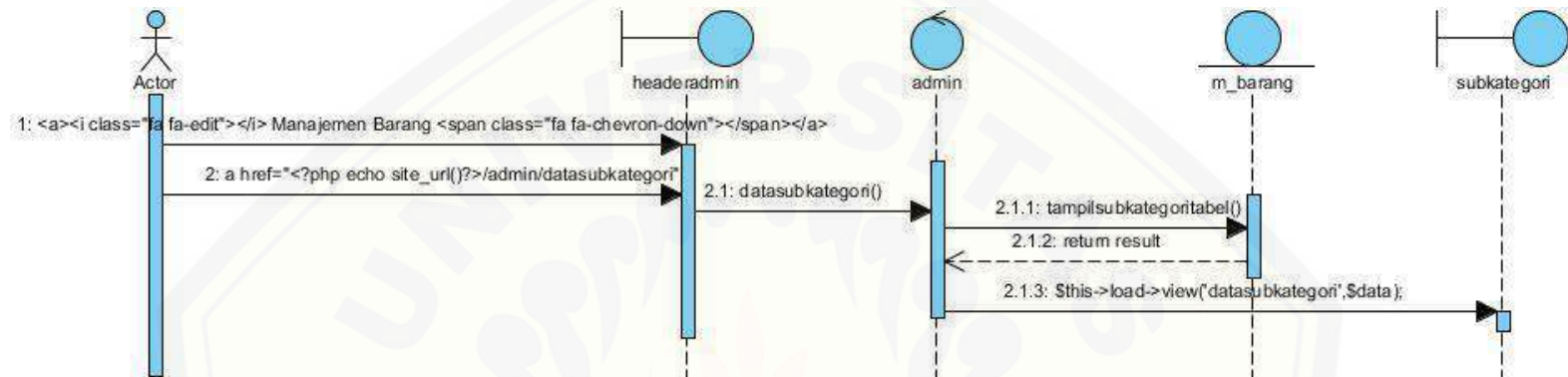
Gambar C.4 Sequence Diagram Merubah Status Non Aktif Data Kategori

C5. Sequence Diagram Manajemen Data Kategori (edit)



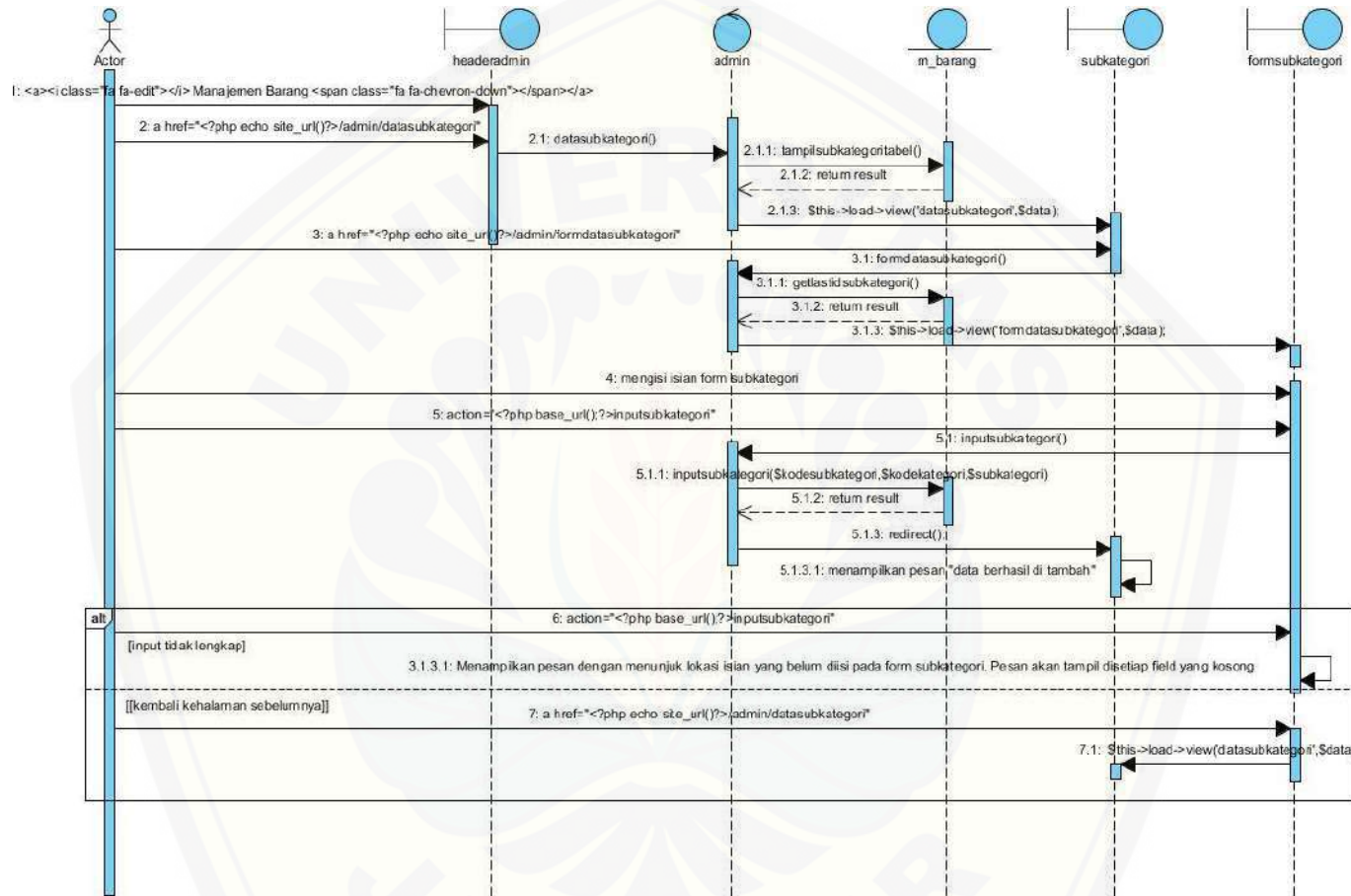
Gambar C.5 Sequence Diagram Manajemen Data Kategori (edit)

C6. Sequence Diagram Manajemen Data Subkategori (view)



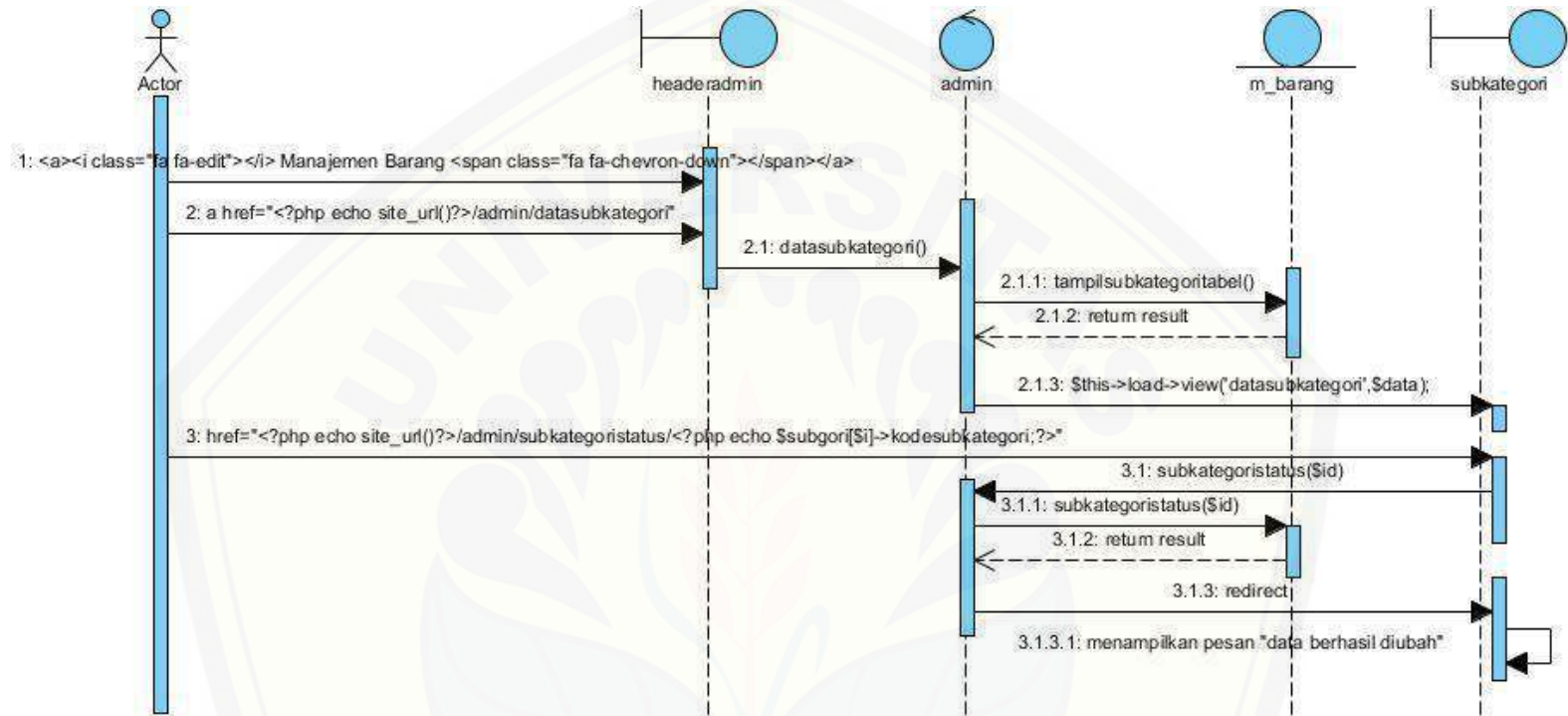
Gambar C.6 Sequence Diagram Manajemen Data Subkategori (view)

C7. Sequence Diagram Manajemen Data Subkategori (input)



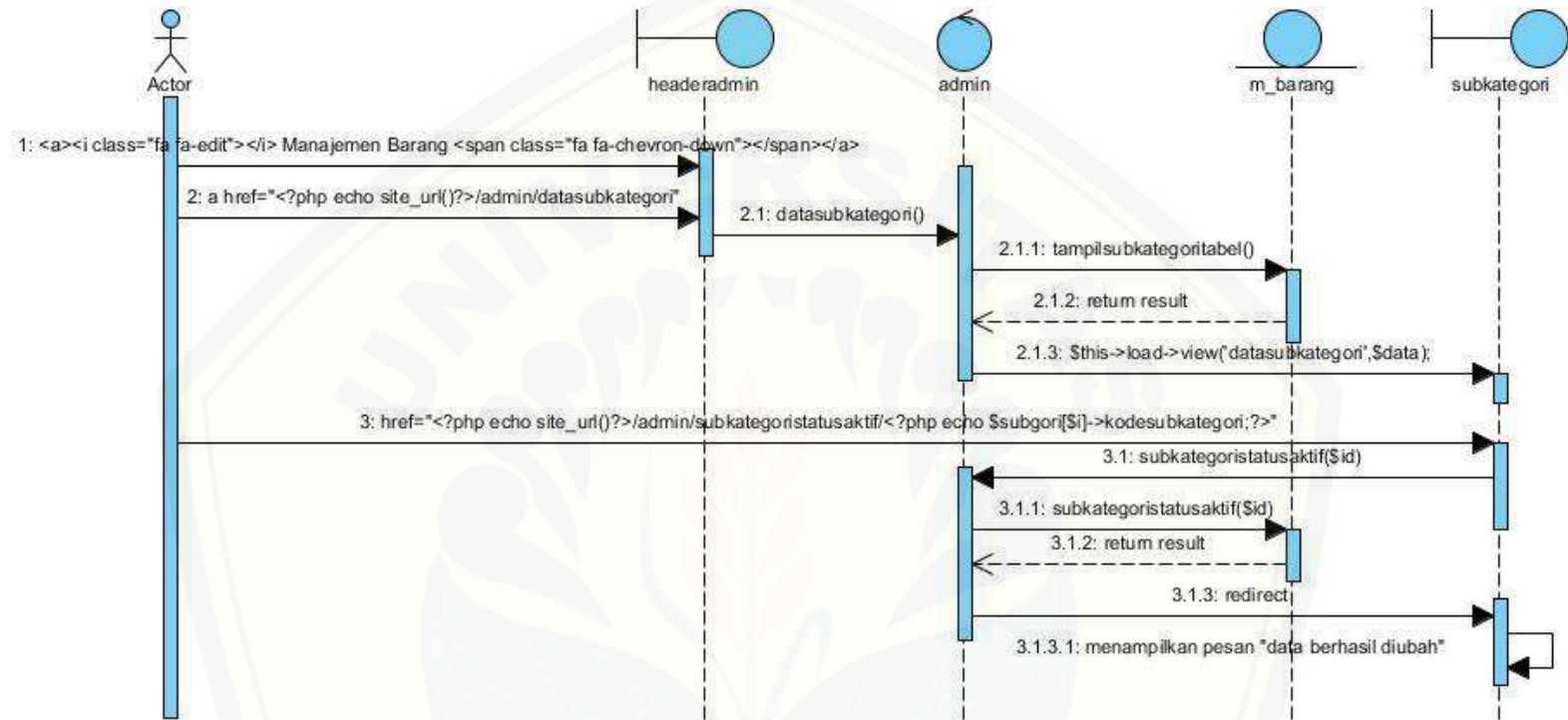
Gambar C.7 Sequence Diagram Manajemen Data Subkategori (input)

C8. Sequence Diagram Merubah Status Aktif Data Subkategori



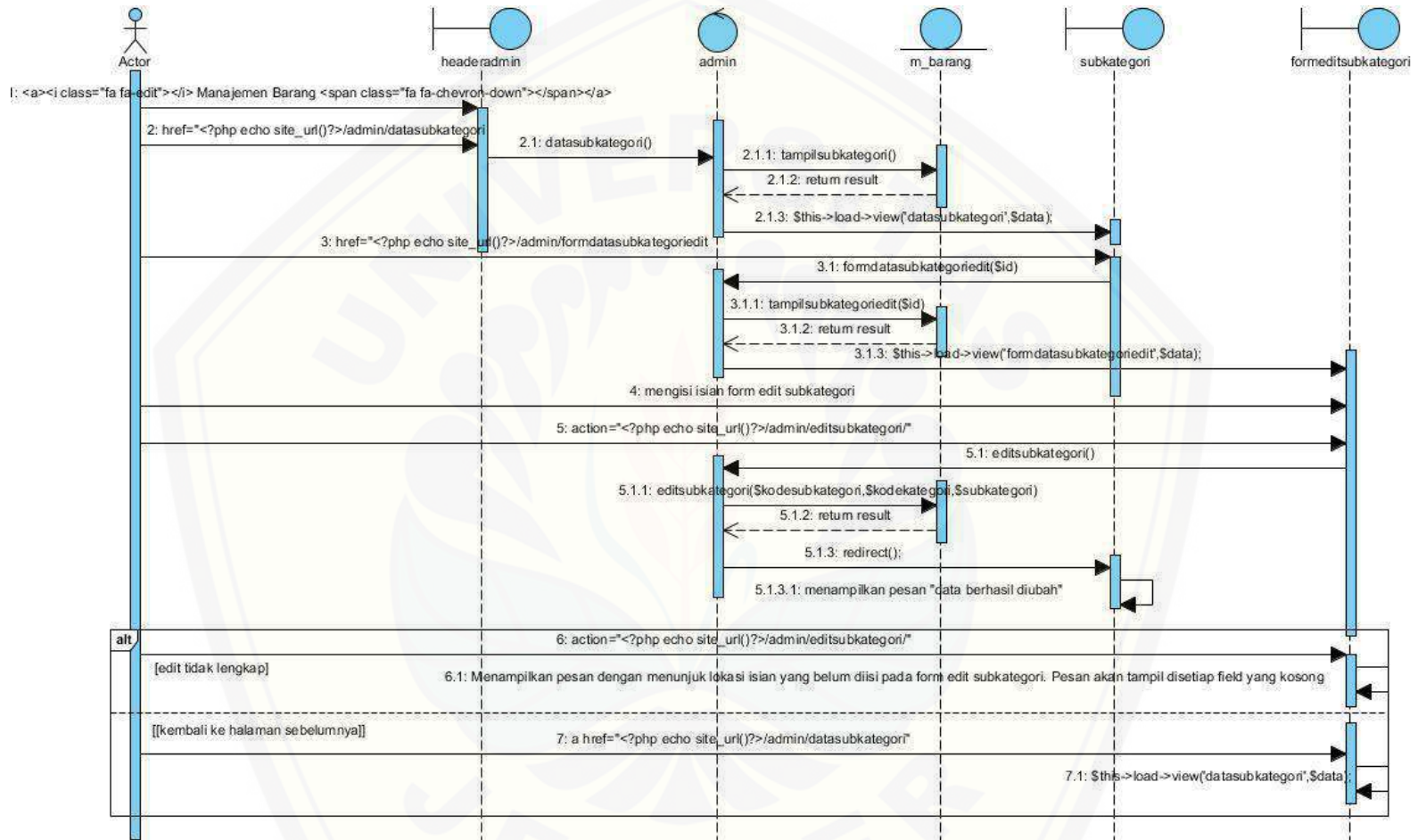
Gambar C.8 Sequence Diagram Merubah Status Aktif Data Subkategori

C9. Sequence Diagram Merubah Status Non Aktif Data Subkategori



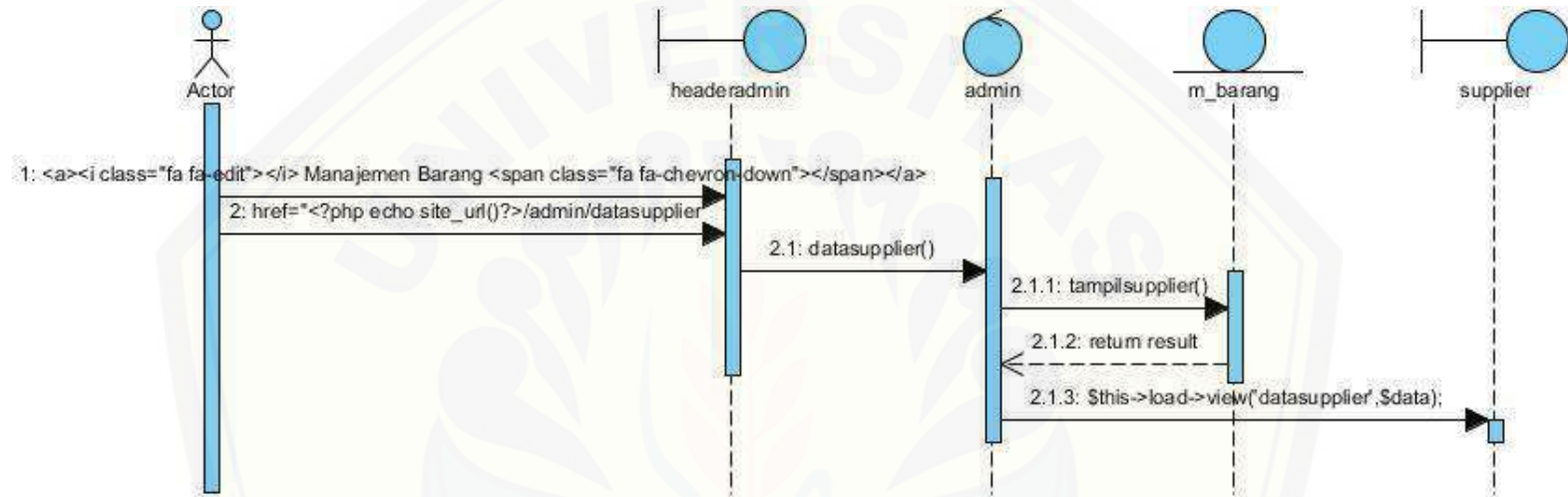
Gambar C.9 Sequence Diagram Merubah Status Non Aktif Data Subkategori

C10. Sequence Diagram Manajemen Data Subkategori(edit)



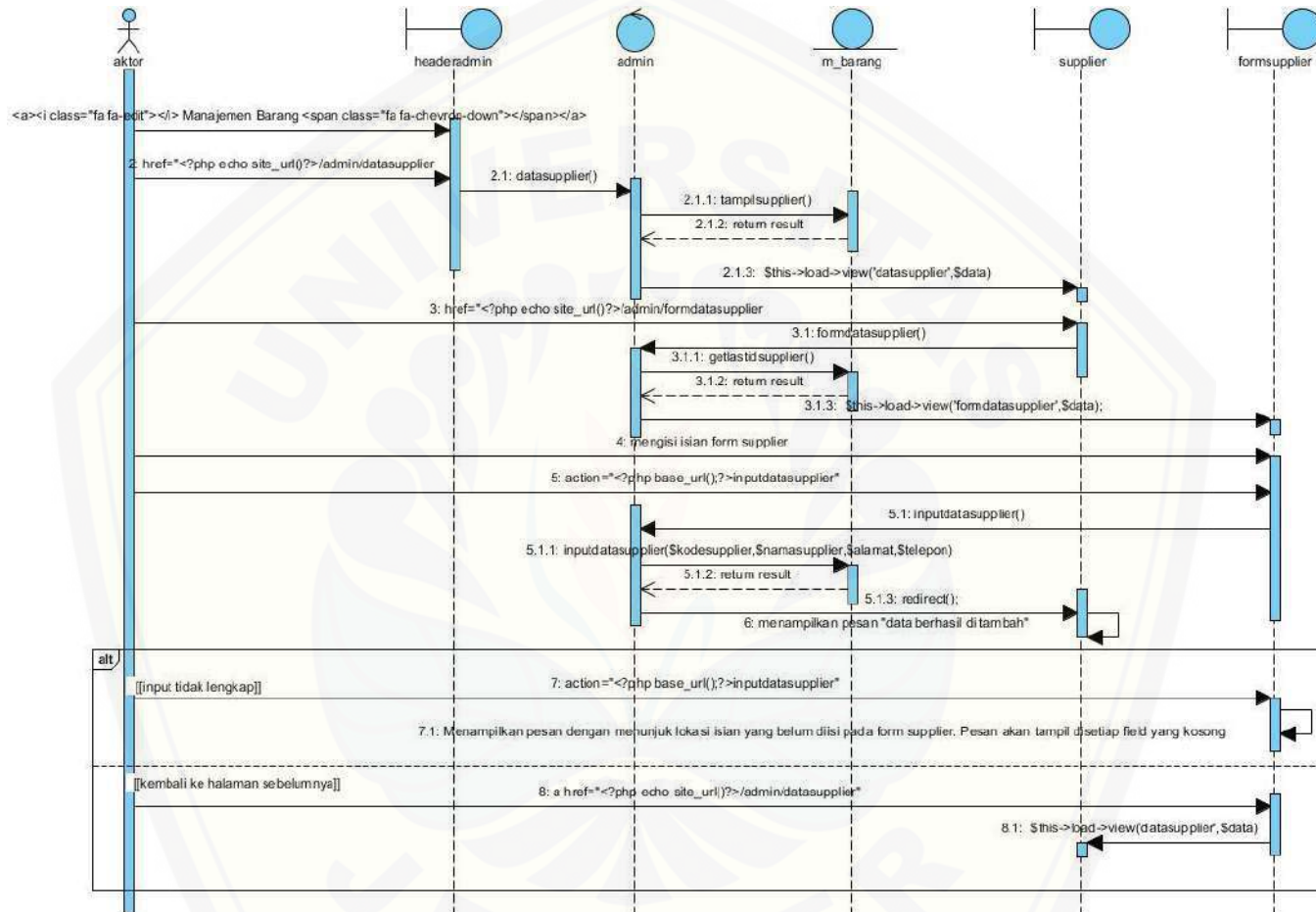
Gambar C.10 Sequence Diagram Manajemen Data Subkategori (edit)

C11. *Sequence Diagram* Manajemen Data Supplier (view)



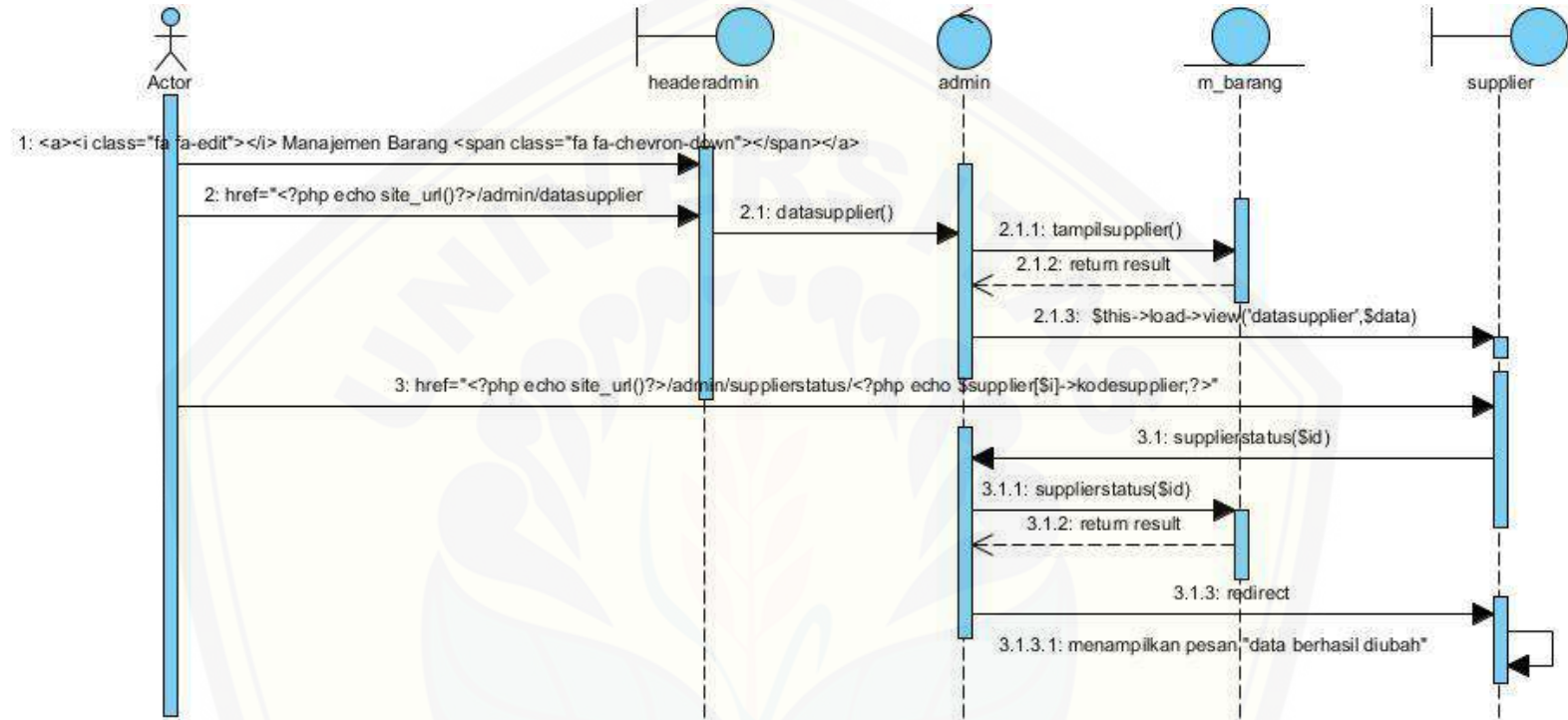
Gambar C.11 *Sequence Diagram* Manajemen Data Supplier (view)

C12. Sequence Diagram Manajemen Data Supplier (input)



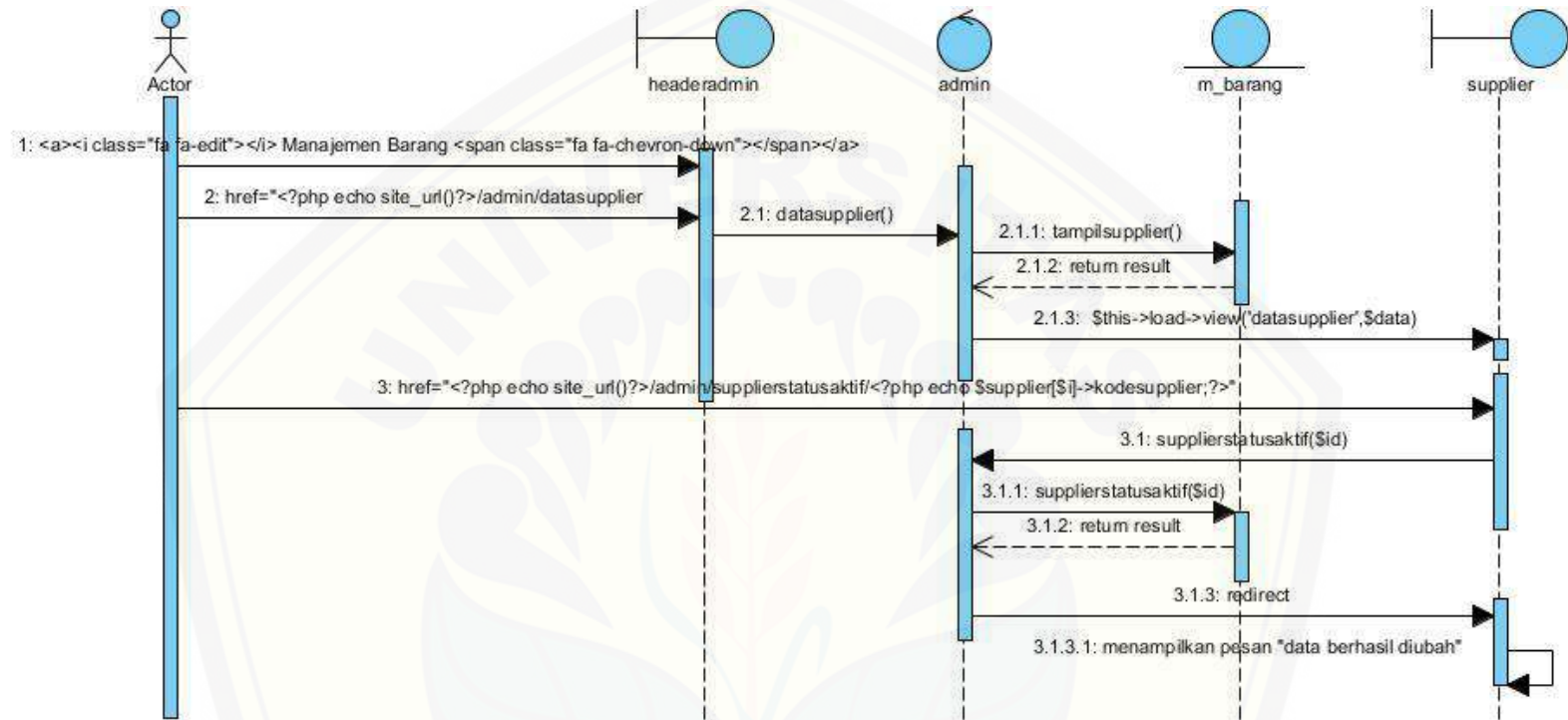
Gambar C.12 Sequence Diagram Manajemen Data Supplier (input)

C13. Sequence Diagram Merubah Status Aktif Data Supplier



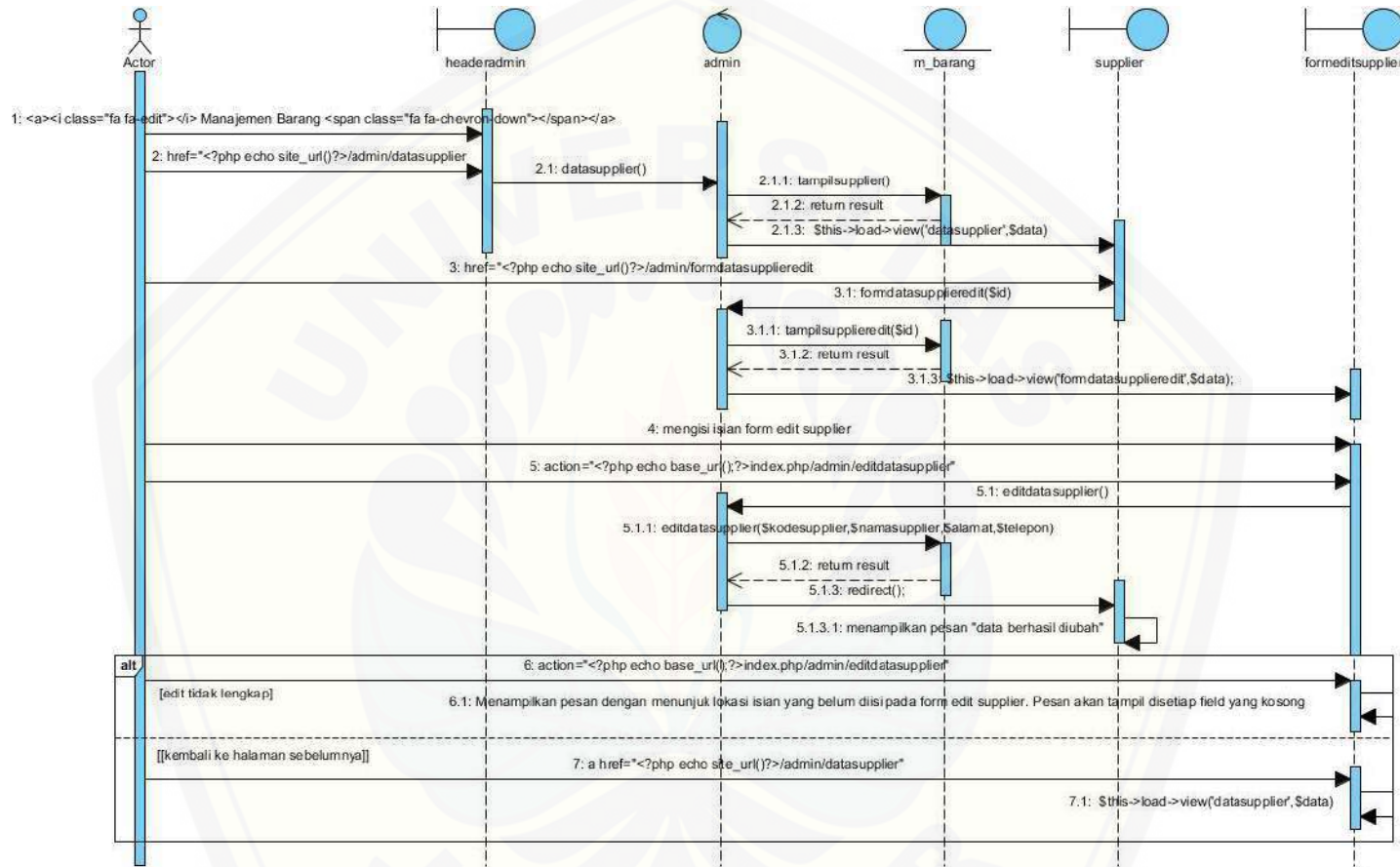
Gambar C.13 Sequence Diagram Merubah Status Aktif Data Supplier

C14. Sequence Diagram Merubah Status Non Aktif Data Supplier



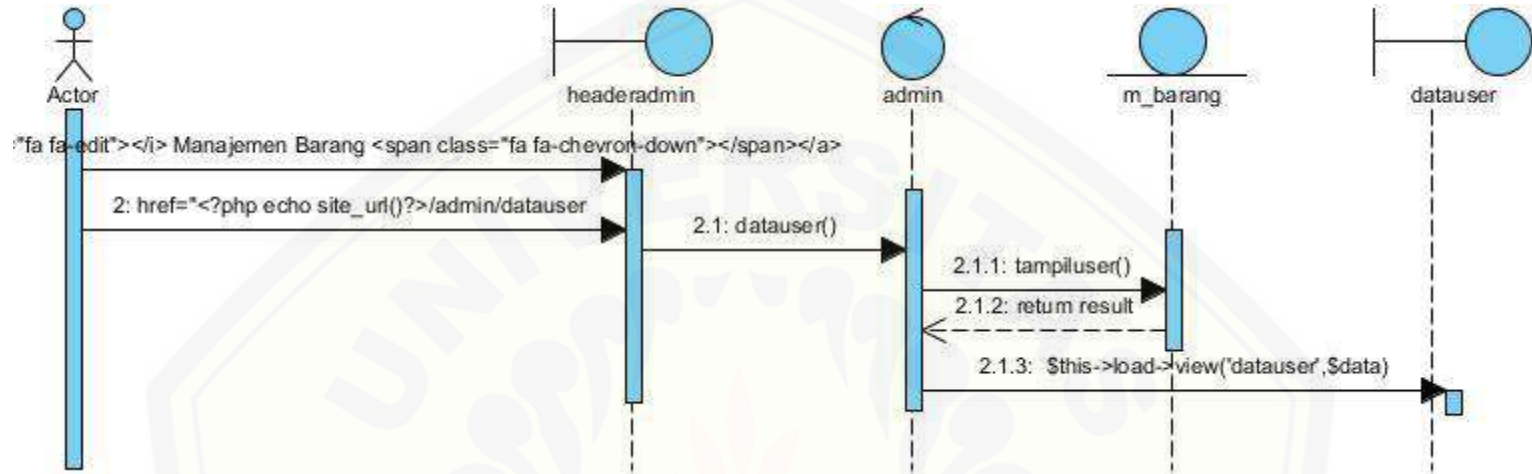
Gambar C.14 Sequence Diagram Merubah Status Non Aktif Data Supplier

C15. Sequence Diagram Manajemen Data Supplier (edit)



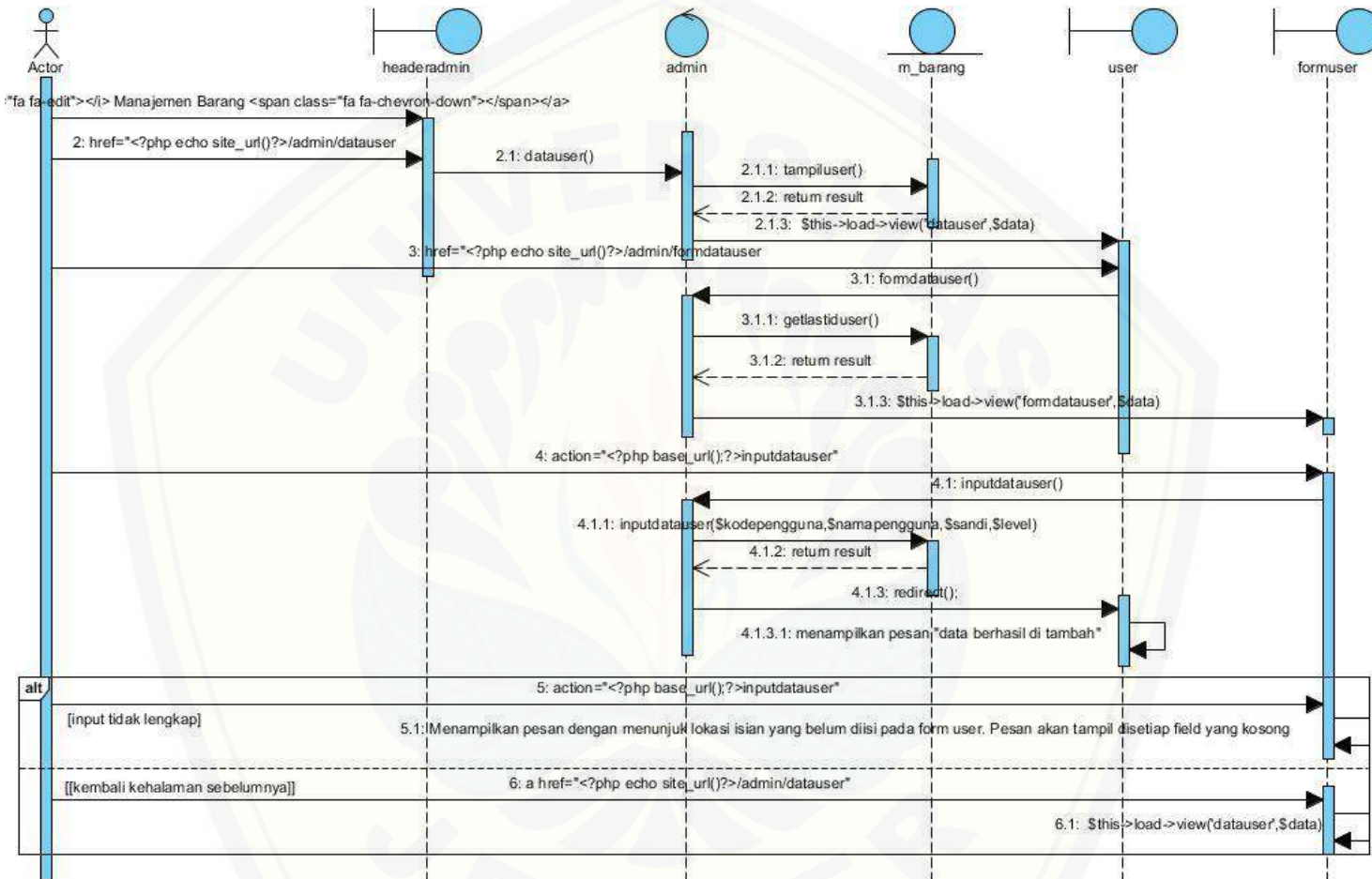
Gambar C.15 Sequence Diagram Manajemen Data Supplier (edit)

C16. Sequence Diagram Manajemen Data User (view)



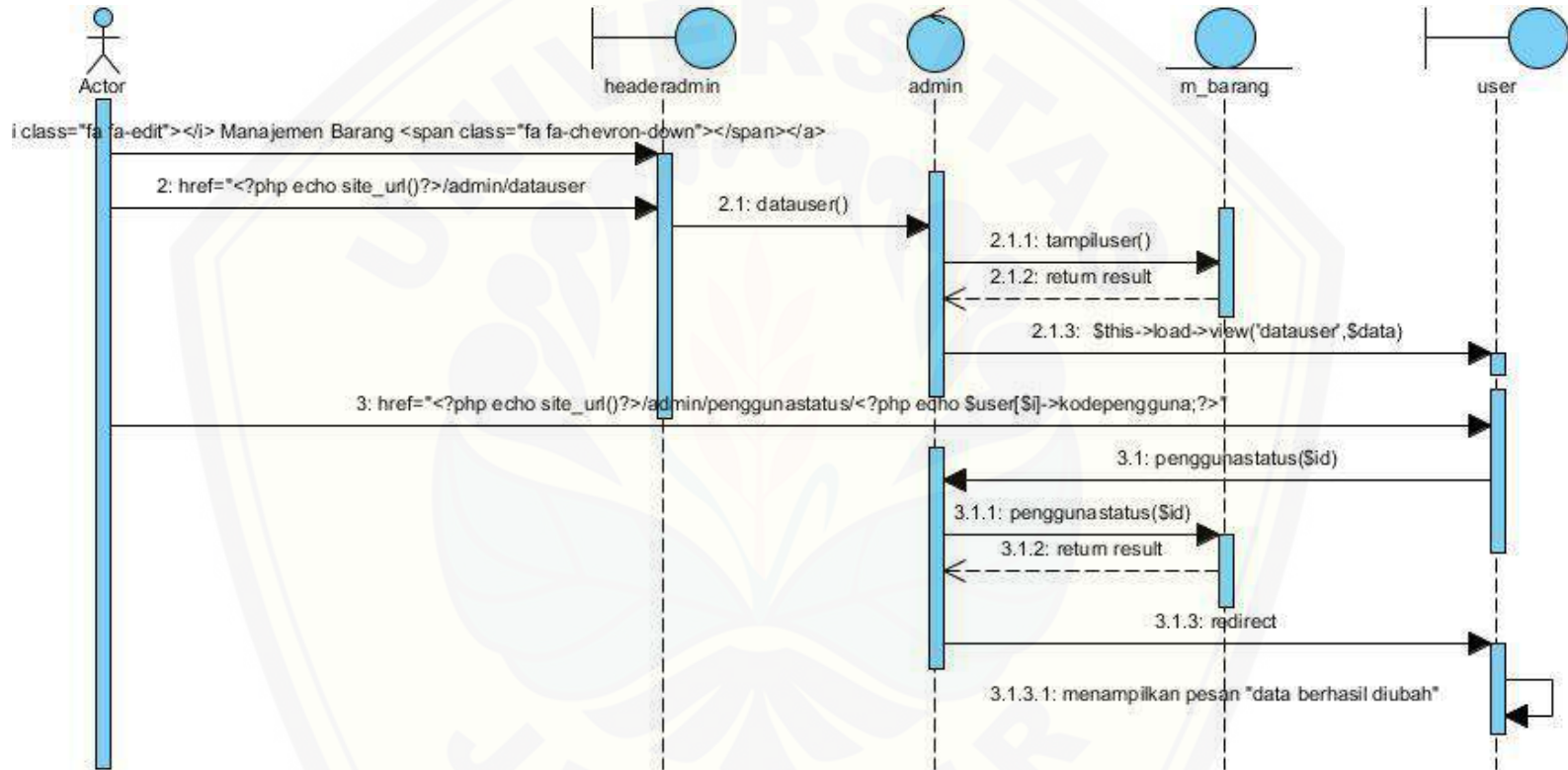
Gambar C.16 Sequence Diagram Manajemen Data User (view)

C17. Sequence Diagram Manajemen Data User (input)



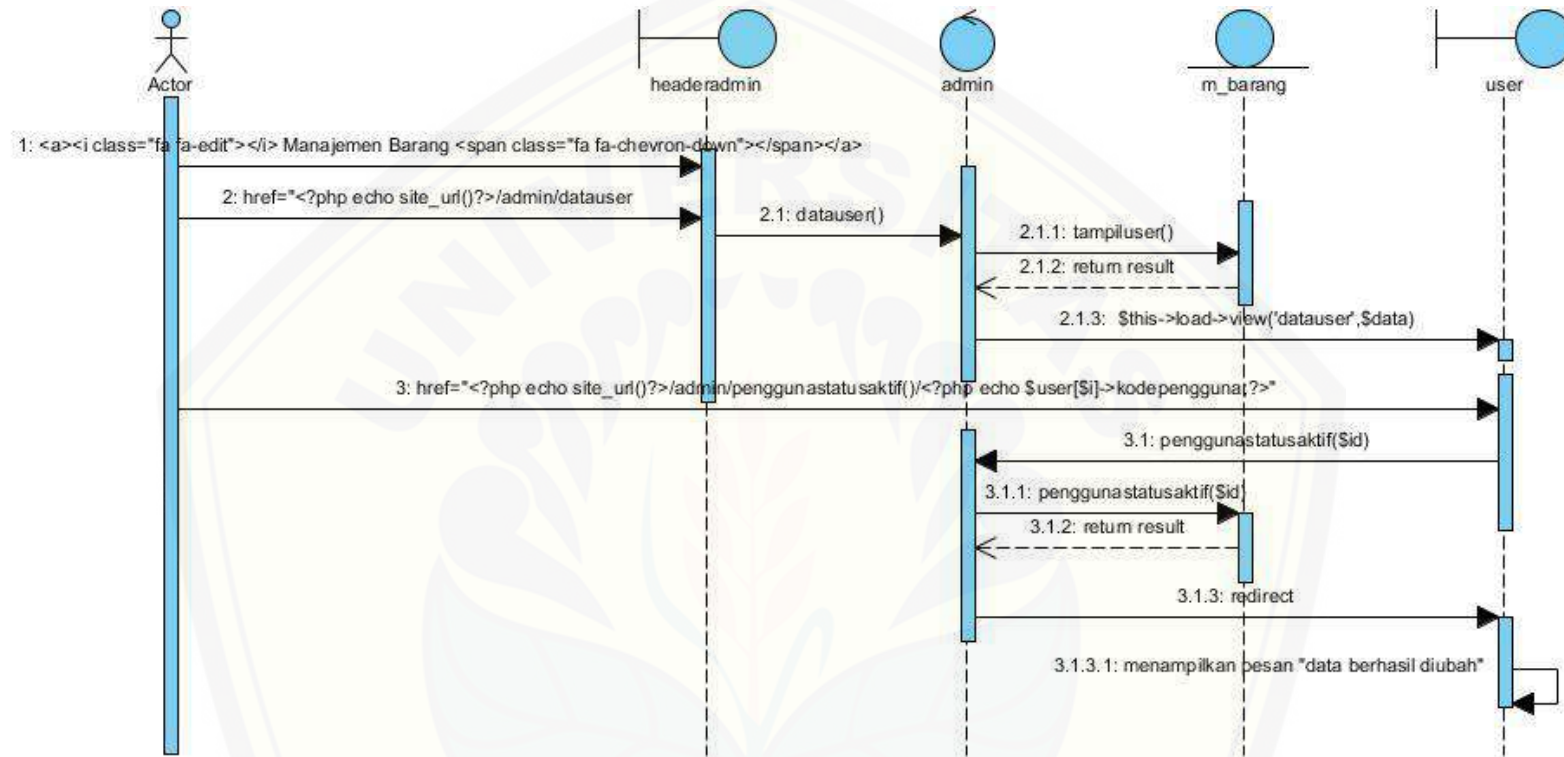
Gambar C.17 Sequence Diagram Manajemen Data User (input)

C18. Sequence Diagram Merubah Status Aktif Data User



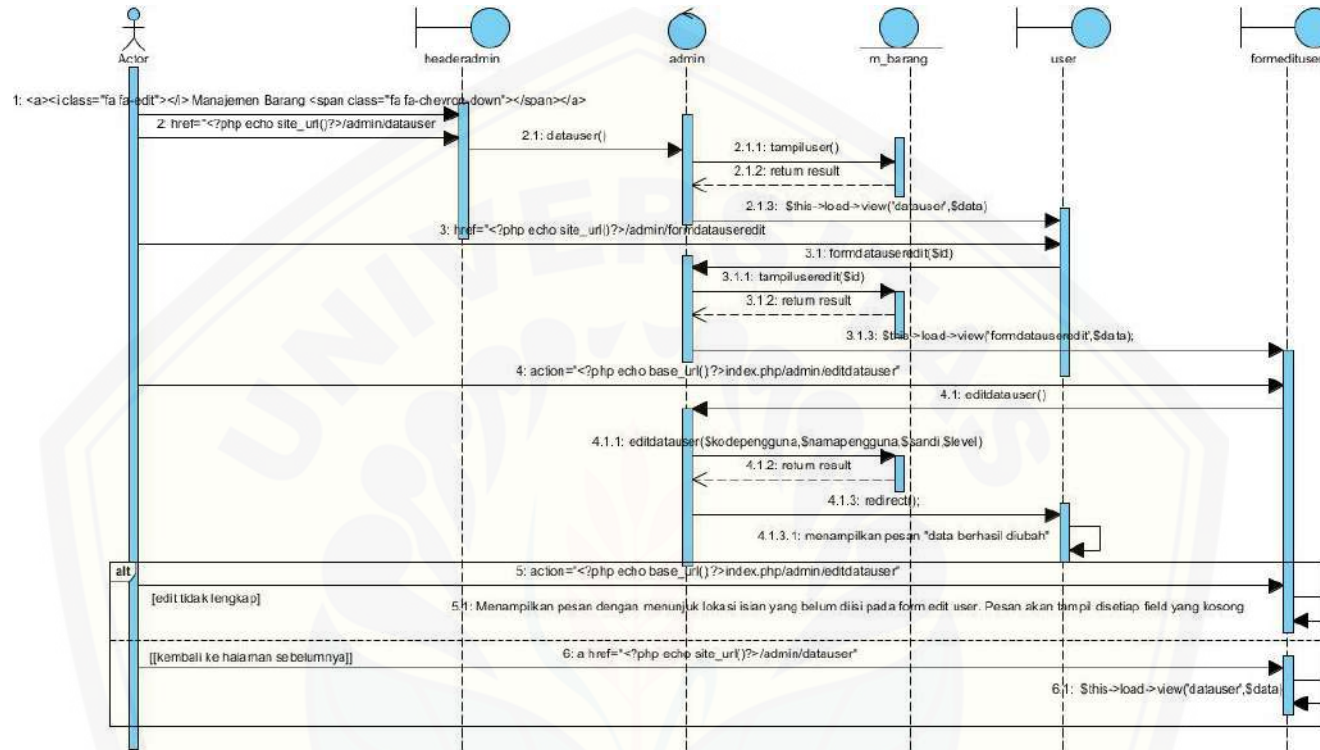
Gambar C. 18 Sequence Diagram Merubah Status Aktif Data User

C19. Sequence Diagram Merubah Status Non Aktif Data User



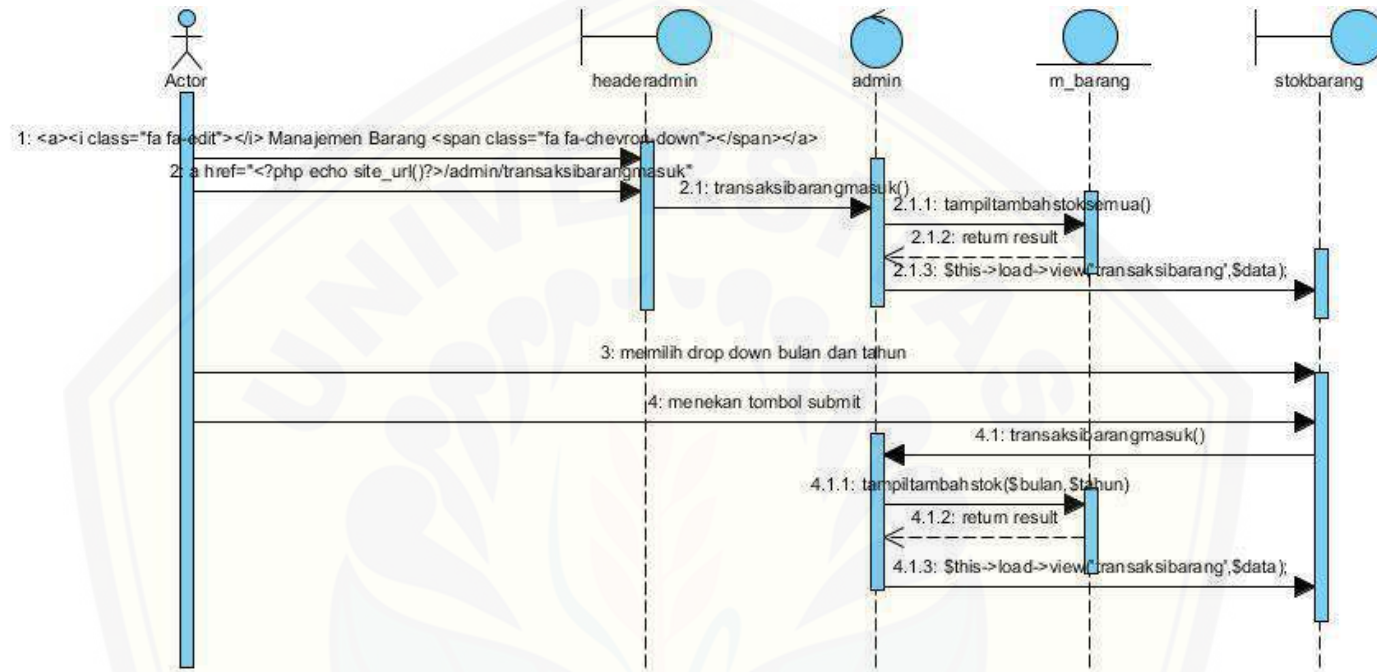
Gambar C.19 Sequence Diagram Merubah Status Non Aktif Data User

C20. Sequence Diagram Manajemen Data User (edit)



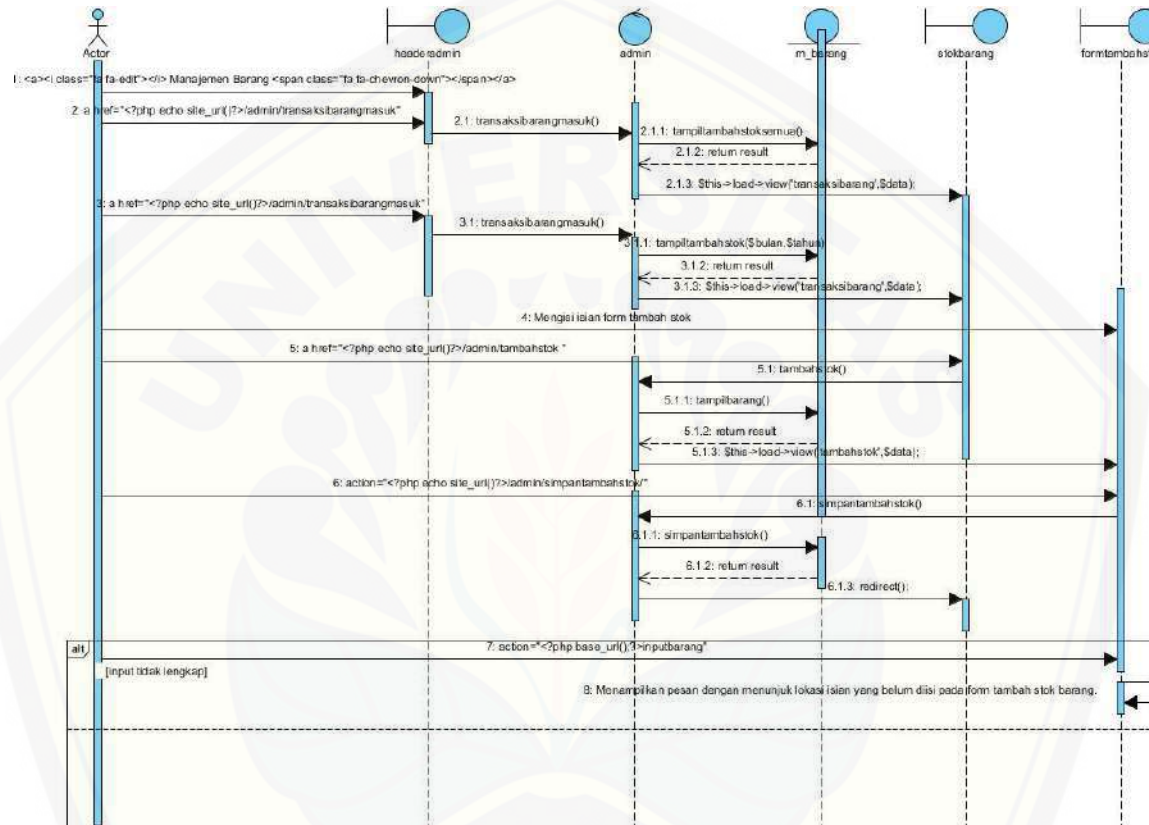
Gambar C.20 Sequence Diagram Manajemen Data user (edit)

C21. Sequence Diagram Manajemen Stok Barang (view)



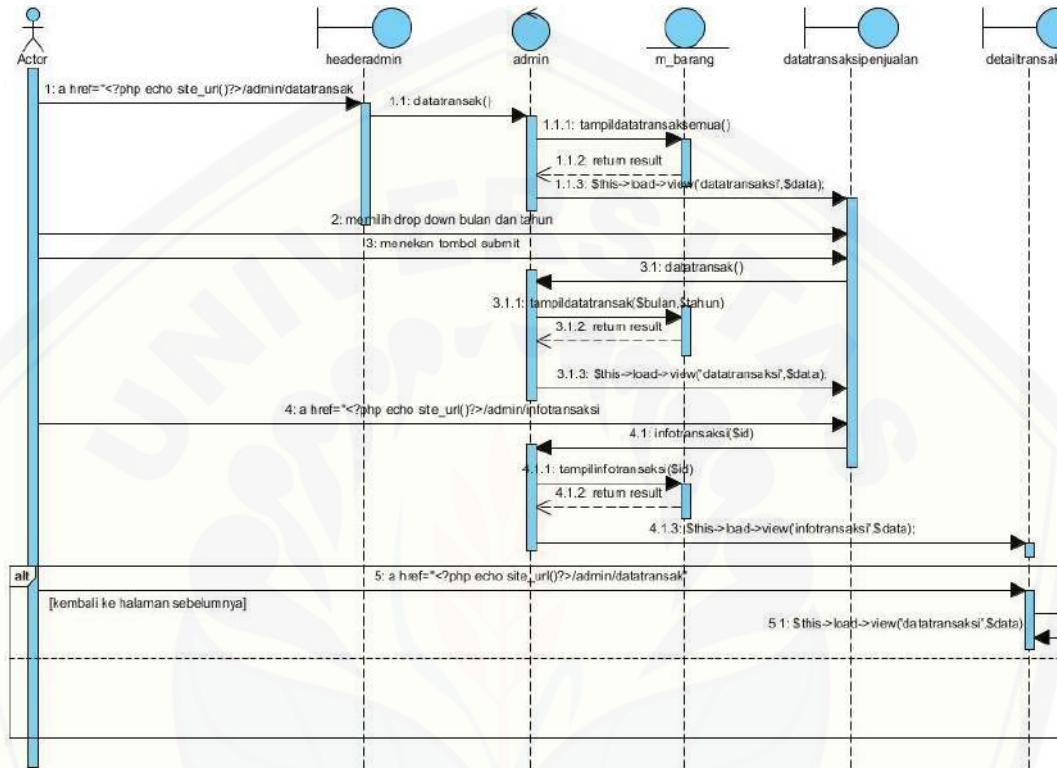
Gambar C.21 Sequence Diagram Manajemen Stok Barang (view)

C22. Sequence Diagram Manajemen Stok Barang (input)



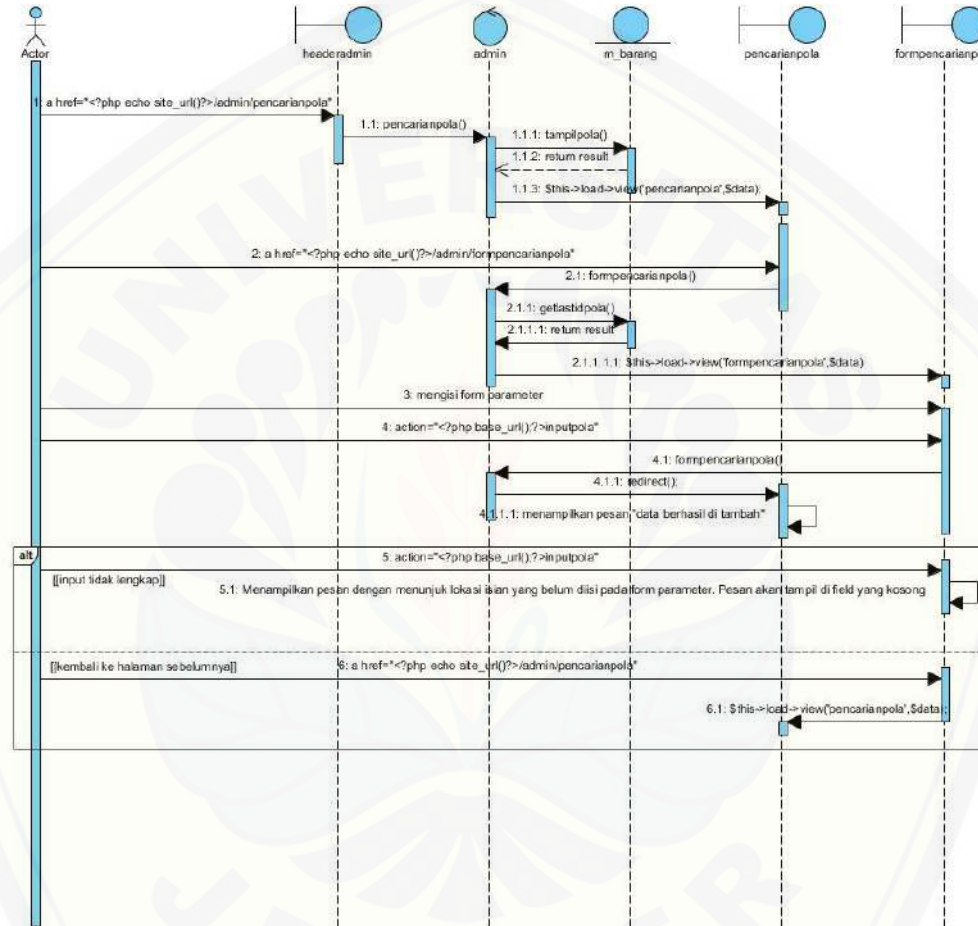
Gambar C.22 Sequence Diagram Manajemen Stok Barang (input)

C23. Sequence Diagram Melihat Data Transaksi (admin)



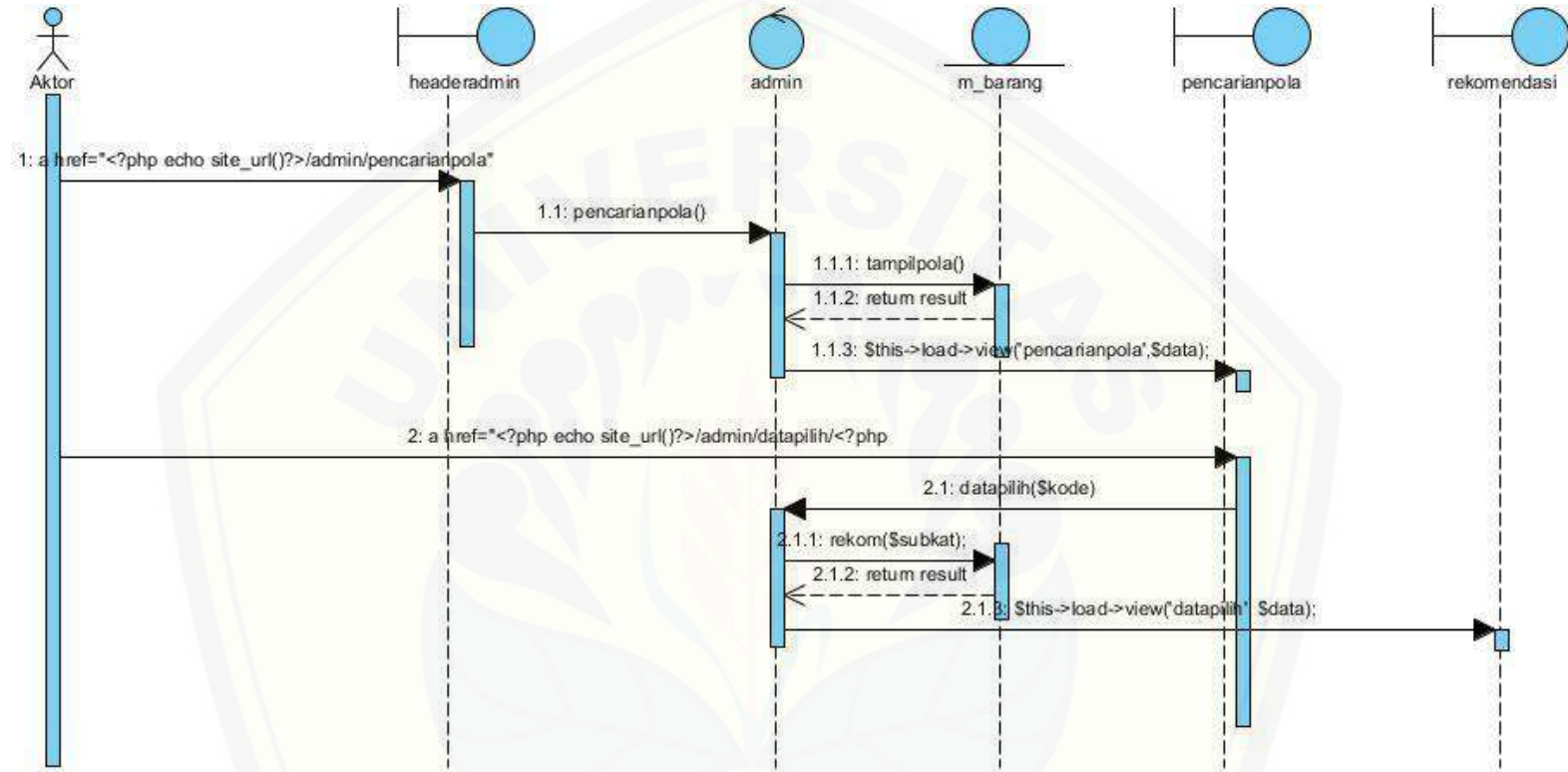
Gambar C.23 Sequence Diagram Melihat Data Transaksi (admin)

C24. Sequence Diagram Input Parameter Pola



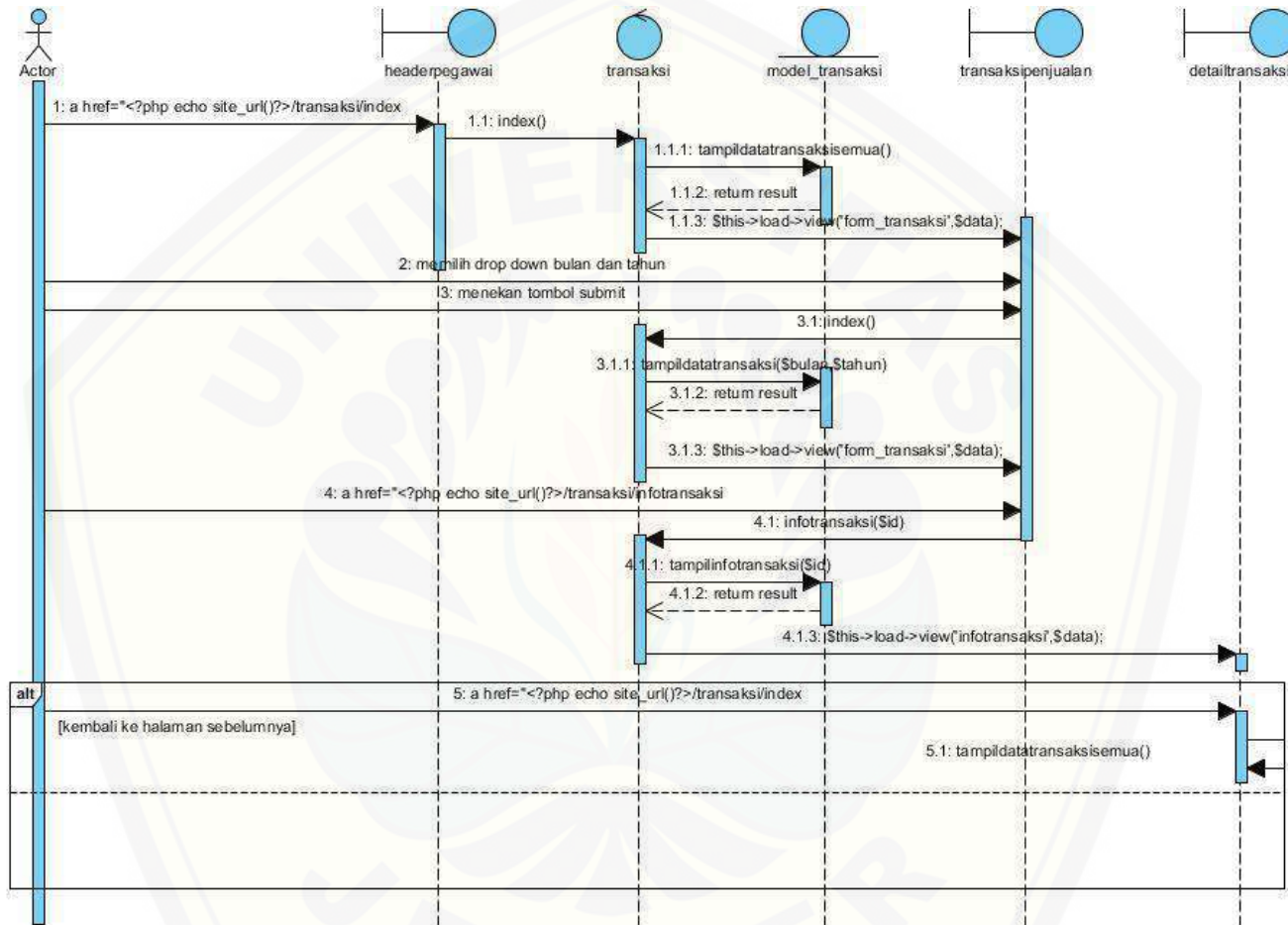
Gambar C.24 Sequence Diagram Input Parameter Pola

C25. Sequence Diagram Rekomendasi Promosi



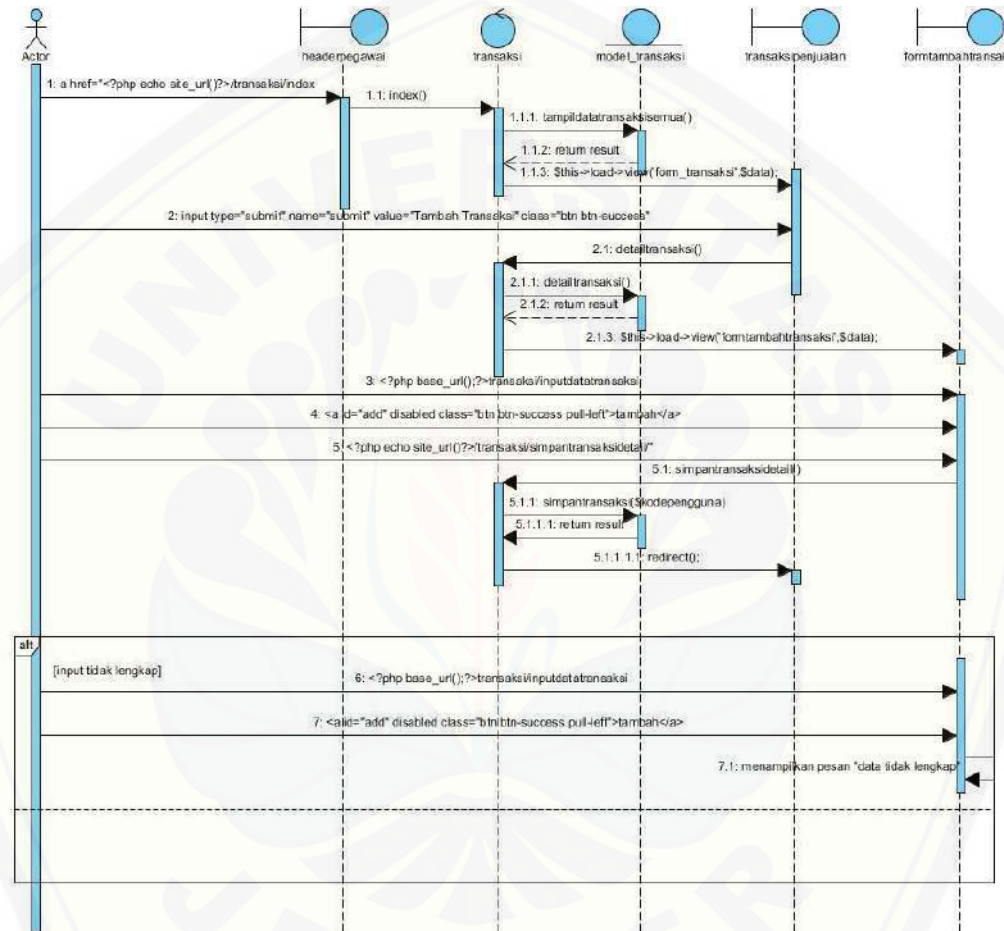
Gambar C.25 Sequence Diagram Rekomendasi Promosi

C26. Sequence Diagram Melihat Data Transaksi (pegawai)



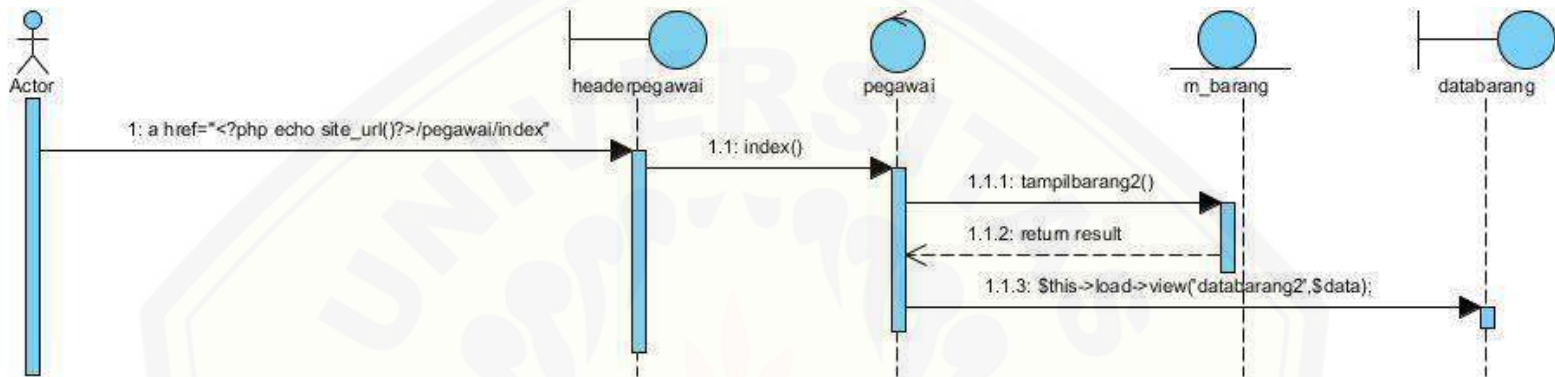
Gambar C.26 Sequence Diagram Melihat Data Transaksi (pegawai)

C27. Sequence Diagram Input Transaksi Penjualan (pegawai)



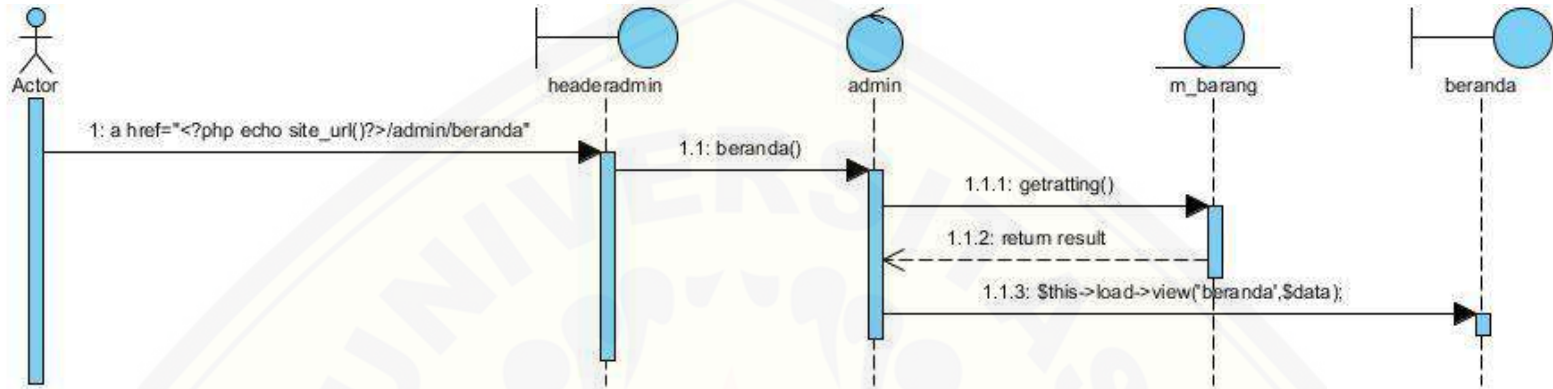
Gambar C.27 Sequence Diagram Input Transaksi Penjualan

C28. Sequence Diagram Melihat Data Barang (pegawai)



Gambar C.28 Sequence Diagram Melihat Data Barang (Pegawai)

C29. Sequence Diagram Beranda (admin)



Gambar C.29 Sequence Diagram Beranda (admin)

Lampiran D. Kode Program

D. 1 Kode program *class* controller/admin

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

class Admin extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        $this->load->library('aprioriclass');
        $this->load->helper('url');
        $this->load->model('m_barang');
        $isLoggedIn = $this->session->userdata('level');
        if($isLoggedIn == null || $isLoggedIn != '1')
        {
            redirect("/logrizk");
        }
    }

    public function beranda()
    {
        $data['barang']=$this->m_barang->gettrating()->result();
        $this->load->view('headeradmin');
        $this->load->view('beranda',$data);
        $this->load->view('footer');
    }

    public function beranda2()
    {
        $this->load->view('headerpegawai');
        $this->load->view('beranda');
        $this->load->view('footer');
    }

    public function databarang()
    {
        $data['barang']=$this->m_barang->tampilbarang()->result();
        $this->load->view('headeradmin');
        $this->load->view('databarang',$data);
        $this->load->view('footer');
    }

    public function databarang2()
    {
        $data['barang']=$this->m_barang->tampilbarang()->result();
        $this->load->view('headerpegawai');
        $this->load->view('databarang',$data);
        $this->load->view('footer');
    }

    public function datakategori()
    {
        $data['gori']=$this->m_barang->tampilkategori()->result();
        $this->load->view('headeradmin');
        $this->load->view('datakategori',$data);
    }
}
```

```
$this->load->view('footer');
}

public function datasubkategori()
{
    $data['subgori']=$this->m_barang->tampilsubkategoritabel()->result();
    $this->load->view('headeradmin');
    $this->load->view('datasubkategori',$data);
    $this->load->view('footer');
}

public function transaksibarangmasuk()
{
    $data['masuk'] = null;
    // $data1['masuk1'] = null;
    if(isset($_POST['submit'])){
        $bulan = $this->input->post('bulan');
        $tahun = $this->input->post('tahun');
        $data['masuk']=$this->m_barang->tampiltambahstok($bulan,$tahun)->result();
        $this->load->view('headeradmin');
        $this->load->view('transaksibarang',$data);
        $this->load->view('footer');
    }

    $data['masuk']=$this->m_barang->tampiltambahstoksemua()->result();
    $this->load->view('headeradmin');
    $this->load->view('transaksibarang',$data);
    $this->load->view('footer');
}

public function transaksihistoriharga()
{
    $data['his']=$this->m_barang->tampilhistoriharga()->result();
    $this->load->view('headeradmin');
    $this->load->view('transaksihistoriharga',$data);
    $this->load->view('footer');
}

public function tambahstok()
{
    $data['barang']=$this->m_barang->tampilbarang()->result();
    $this->load->view('headeradmin');
    $this->load->view('tambahstok',$data);
    $this->load->view('footer');
}

public function pencarianpola()
{
    $data['pola']=$this->m_barang->tampilpola(($kode = null))->result();
    $this->load->view('headeradmin');
    $this->load->view('pencarianpola',$data);
    $this->load->view('footer');
}

public function datatransak()
{
```

```
$data['tam'] = null;
if(isset($_POST['submit'])){
    $bulan = $this->input->post('bulan');
    $tahun = $this->input->post('tahun');
    $data['tam']=$this->m_barang->tampildatatrnsak($bulan,$tahun)->result();
    $this->load->view('headeradmin');
    $this->load->view('datatrnsaksi',$data);
    // $this->load->view('footer');
}
$data['tam']=$this->m_barang->tampildatatrnsaksemua()->result();
$this->load->view('headeradmin');
$this->load->view('datatrnsaksi',$data);
$this->load->view('footer');
}

public function infotrnsaksi($id)
{
    $data['kodetrnsaksi']=$this->m_barang->tampilinfotrnsaksi($id);
    $this->load->view('headeradmin');
    $this->load->view('infotrnsaksi',$data);
    $this->load->view('footer');
}

public function datasupplier()
{
    $data['supplier']=$this->m_barang->tampilsupplier()->result();
    $this->load->view('headeradmin');
    $this->load->view('datasupplier',$data);
    $this->load->view('footer');
}

public function datauser()
{
    $data['user']=$this->m_barang->tampiluser()->result();
    $this->load->view('headeradmin');
    $this->load->view('datauser',$data);
    $this->load->view('footer');
}

public function formdatabarang()
{
    $data['kategori']=$this->m_barang->tampilkategori()->result();
    $data['subkategori']=$this->m_barang->tampilsubkategoritabel()->result();
    $data['supplier']=$this->m_barang->tampilsupplier()->result();
    $data['kodebarang'] = $this->m_barang->getlastidbarang();
    $this->load->view('headeradmin');
    $this->load->view('formdatabarang',$data);
    $this->load->view('footer');
}

public function formdatabarangedit($id)
{
    $data['kategori']=$this->m_barang->tampilkategori()->result();
    $data['subkategori']=$this->m_barang->tampilsubkategoritabel()->result();
    $data['supplier']=$this->m_barang->tampilsupplier()->result();
}
```



```
$data['editbarang']=$this->m_barang->tampilbarangedit($id)->result_array();
$this->load->view('headeradmin');
$this->load->view('formdatabarangedit',$data);
$this->load->view('footer');
}

function get_subkategori($idkategori) {
    $result=$this->m_barang->subkategori($idkategori);

    $HTML="";
    if($result->num_rows() > 0){
        foreach($result->result() as $list){
            $HTML.="<option value='".$list->kodesubkategori."'>".$list->subkategori."</option>";
        }
    }
    echo $HTML;
}

public function formeditstatus($id){
    $sukses = $this->m_barang->tampileditstatus($id);

    if($sukses == false)
    {
        redirect('admin/databarang') . "?pesan=gagal";
    }
    else
    {
        redirect(site_url('admin/databarang') . "?pesan=ubah");
    }
}

public function formeditstatusaktif($id){
    $sukses = $this->m_barang->tampileditstatusaktif($id);

    if($sukses == false)
    {
        redirect('admin/databarang') . "?pesan=gagal";
    }
    else
    {
        redirect(site_url('admin/databarang') . "?pesan=ubah");
    }
}

public function kategoristatus($id){
    $sukses = $this->m_barang->kategoristatus($id);

    if($sukses == false)
    {
        redirect('admin/datakategori') . "?pesan=gagal";
    }
    else
    {
        redirect(site_url('admin/datakategori') . "?pesan=ubah");
    }
}
```

```
public function kategoristatusaktif($id){
    $sukses = $this->m_barang->kategoristatusaktif($id);

    if($sukses == false)
    {
        redirect('admin/datakategori') . "?pesan=gagal";
    }
    else
    {
        redirect(site_url('admin/datakategori') . "?pesan=ubah");
    }
}

public function subkategoristatus($id){
    $sukses = $this->m_barang->subkategoristatus($id);

    if($sukses == false)
    {
        redirect('admin/datasubkategori') . "?pesan=gagal";
    }
    else
    {
        redirect(site_url('admin/datasubkategori') . "?pesan=ubah");
    }
}

public function subkategoristatusaktif($id){
    $sukses = $this->m_barang->subkategoristatusaktif($id);

    if($sukses == false)
    {
        redirect('admin/datasubkategori') . "?pesan=gagal";
    }
    else
    {
        redirect(site_url('admin/datasubkategori') . "?pesan=ubah");
    }
}

public function supplierstatus($id){
    $sukses = $this->m_barang->supplierstatus($id);

    if($sukses == false)
    {
        redirect('admin/datasupplier') . "?pesan=gagal";
    }
    else
    {
        redirect(site_url('admin/datasupplier') . "?pesan=ubah");
    }
}

public function supplierstatusaktif($id){
    $sukses = $this->m_barang->supplierstatusaktif($id);

    if($sukses == false)
    {
```

```
        redirect('admin/datasupplier') . "?pesan=gagal";
    }
    else
    {
        redirect(site_url('admin/datasupplier') . "?pesan=ubah");
    }
}

public function penggunastatus($id){
    $sukses = $this->m_barang->penggunastatus($id);

    if($sukses == false)
    {
        redirect('admin/datauser') . "?pesan=gagal";
    }
    else
    {
        redirect(site_url('admin/datauser') . "?pesan=ubah");
    }
}

public function penggunastatusaktif($id){
    $sukses = $this->m_barang->penggunastatusaktif($id);

    if($sukses == false)
    {
        redirect('admin/datauser') . "?pesan=gagal";
    }
    else
    {
        redirect(site_url('admin/datauser') . "?pesan=ubah");
    }
}

public function formdatakategori()
{
    $data['kodekategori'] = $this->m_barang->getlastidkategori();
    $this->load->view('headeradmin');
    $this->load->view('formdatakategori', $data);
    $this->load->view('footer');
}

public function formdatakategoriedit($kodekategori)
{
    $data['editkategori']=$this->m_barang->tampilkategoriedit($kodekategori);
    $this->load->view('headeradmin');
    $this->load->view('formdatakategoriedit',$data);
    $this->load->view('footer');
}

public function formdatakategoridelete($id)
{
    $sukses = $this->m_barang->tampilkategorihapus($id);

    if($sukses == false)
    {
        redirect('admin/datakategori') . "?pesan=gagal";
    }
}
```

```
    }else
    {
        redirect(site_url('admin/datakategori') . "?pesan=hapus");
    }
}

public function formdatasubkategori()
{
    $data['kategori']=$this->m_barang->tampilkategori()->result();
    $data['kodesubkategori'] = $this->m_barang->getlastidsubkategori();
    $this->load->view('headeradmin');
    $this->load->view('formdatasubkategori',$data);
    $this->load->view('footer');
}

public function formdatasubkategoriedit($id)
{
    $data['kategori']=$this->m_barang->tampilkategori()->result();
    $data['editsubkategori']=$this->m_barang->tampilsubkategoriedit($id);
    $this->load->view('headeradmin');
    $this->load->view('formdatasubkategoriedit',$data);
    $this->load->view('footer');
}

public function formdatasubkategoridelete($id)
{
    $sukses = $this->m_barang->tampilsubkategorihapus($id);

    if($sukses == false)
    {
        redirect('admin/datasubkategori') . "?pesan=gagal";
    }
    else
    {
        redirect(site_url('admin/datasubkategori') . "?pesan=hapus");
    }
}

public function formdatasupplier()
{
    $data['kodesupplier'] = $this->m_barang->getlastidsupplier();
    $this->load->view('headeradmin');
    $this->load->view('formdatasupplier',$data);
    $this->load->view('footer');
}

public function formdatasupplieredit($id)
{
    $data['editsupplier']=$this->m_barang->tampilsupplieredit($id);
    $this->load->view('headeradmin');
    $this->load->view('formdatasupplieredit',$data);
    $this->load->view('footer');
}

public function formdatasupplierdelete($id)
{
    $sukses = $this->m_barang->tampilsupplierhapus($id);
```

```

if($sukses == false)
{
    redirect('admin/datasupplier') . "?pesan=gagal";
}
else
{
    redirect(site_url('admin/datasupplier') . "?pesan=hapus");
}
}

public function formdatauser()
{
    $data['kodeuser'] = $this->m_barang->getlastiduser();
    $this->load->view('headeradmin');
    $this->load->view('formdatauser',$data);
    $this->load->view('footer');
}

public function formdatauseredit($id)
{
    $data['edituser']=$this->m_barang->tampiluseredit($id);
    $this->load->view('headeradmin');
    $this->load->view('formdatauseredit',$data);
    $this->load->view('footer');
}

public function formpencarianpola()
{
    $data['kodepola'] = $this->m_barang->getlastidpola();
    $this->load->view('headeradmin');
    $this->load->view('formpencarianpola',$data);
    $this->load->view('footer');
}

public function datapilih($kode)
{
    $result = $this->m_barang->tampilpola($kode)->result_array()[0];
    $data['pilih'] = $this->m_barang->tampilItem($result['periodeawal'],
$result['periodeakhir'])->result();

    $Apriori = new Aprioriclass();

    $Apriori->setMaxScan($result['makscan']); //Scan 2, 3, ...
    $Apriori->setMinSup($result['minsupp']); //Minimum support 1, 2, 3, ...
    $Apriori->setMinConf($result['minconf']); //Minimum confidence - Percent 1, 2, ...,
100
    $Apriori->setDelimiter(','); //Delimiter

    $dataset = $this->m_barang->tampilItem($result['periodeawal'],$result['periodeakhir'])->result_array();
    $Apriori->process($dataset);

    $data['pfi1'] = $Apriori->printFreqItemsets()[0];
    $data['pfi2'] = $Apriori->printFreqItemsets()[1];
    $data['pfi3'] = $Apriori->printFreqItemsets()[2]; //nama barang
    $data['pfi4'] = $Apriori->printFreqItemsets()[3]; //jml

```

```

$data['pai1'] = $Apriori->printAssociationRules()[0];
$data['pai2'] = $Apriori->printAssociationRules()[1];

$data['pai3'] = $Apriori->printAssociationRules()[2];
$data['pai4'] = $Apriori->printAssociationRules()[3];
$data['pai5'] = $Apriori->printAssociationRules()[4];
$subkat = array();

    $subkat[0] = $this->uri->segment(3);
    $subkat[1] = $data['pai5'];

$data['subkat'] = $this->m_barang->rekom($subkat)->result_array();

$this->load->view('headeradmin');
$this->load->view('datapilih', $data);
$this->load->view('footer');
}

public function apriori()
{
    $Apriori = new Aprioriclass();

    $Apriori->setMaxScan(20);    //Scan 2, 3, ...
    $Apriori->setMinSup(2);      //Minimum support 1, 2, 3, ...
    $Apriori->setMinConf(75);    //Minimum confidence - Percent 1, 2, ..., 100
    $Apriori->setDelimiter(','); //Delimiter

    $dataset = $this->m_barang->tampilitemsemua()->result_array();
    $Apriori->process($dataset);

    $data['pfi1'] = $Apriori->printFreqItemsets()[0];
    $data['pfi2'] = $Apriori->printFreqItemsets()[1];

    $data['pai1'] = $Apriori->printAssociationRules()[0];
    $data['pai2'] = $Apriori->printAssociationRules()[1];

    $this->load->view('headeradmin');
    $this->load->view('apriori',$data);
    $this->load->view('footer');
}

public function inputbarang()
{
    if(isset($_POST['submit'])){
        $kodebarang=$this->input->post('kodebarang');
        $kodekategori=$this->input->post('kodekategori');
        $kodesubkategori=$this->input->post('kodesubkategori');
        $kodesupplier=$this->input->post('kodesupplier');
        $namabarang=$this->input->post('namabarang');
        $hargabeli=$this->input->post('hargabeli');
        $hargajual=$this->input->post('hargajual');
        $kodepengguna=$this->input->post('kodepengguna');

        $input=$this->m_barang->inputbarang($kodebarang,$kodekategori,$kodesubkategori,$kodesupplier,$namabarang,

```

```
$hargabeli,$hargajual,$kodepengguna);

    if($input == false){
        redirect('admin/databarang/formdatabarang') . "?pesan=gagal";
    }else{
        redirect(site_url('admin/databarang') . "?pesan=tambah");
    }
}
}

public function editbarang()
{
    if(isset($_POST['submit'])){
        $kodebarang=$this->input->post('kodebarang');
        $kodekategori=$this->input->post('kodekategori');
        $kodesubkategori=$this->input->post('kodesubkategori');
        $kodesupplier=$this->input->post('kodesupplier');
        $namabarang=$this->input->post('namabarang');
        $hargabeli=$this->input->post('hargabeli');
        $hargajual=$this->input->post('hargajual');

        $input=$this->m_barang->
>editbarang($kodebarang,$kodekategori,$kodesubkategori,$kodesupplier,$namabarang,
$hargabeli,$hargajual);

        if($input == false){
            redirect('admin/databarang/formdatabarangedit') . "?pesan=gagal";
        }else
        {
            redirect(site_url('admin/databarang') . "?pesan=ubah");
        }
        var_dump($input);
    }
}

public function simpantambahstok(){
    if($this->m_barang->simpantambahstok()){
        redirect('admin/transaksibarangmasuk');
    }else{
        redirect('admin/transaksibarangmasuk');
    }
}

public function inputkategori()
{
    if(isset($_POST['submit']))
    {
        $kodekategori=$this->input->post('kodekategori');
        $kategori=$this->input->post('kategori');
        $kodepengguna=$this->input->post('kodepengguna');

        $input=$this->m_barang->inputkategori($kodekategori,$kategori,$kodepengguna);

        if($input == false){
            redirect('admin/datakategori/formdatakategori') . "?pesan=gagal";
        }
        else
    }
}
```

```
        {
            redirect(site_url('admin/datakategori') . "?pesan=tambah");
        }
    }
}

public function editkategori()
{
    if(isset($_POST['submit']))
    {
        $kodekategori=$this->input->post('kodekategori');
        $kategori=$this->input->post('kategori');

        $input=$this->m_barang->editkategori($kodekategori,$kategori);

        if($input == false)
        {
            redirect('admin/datakategori/formdatakategoriedit') . "?pesan=gagal";
        }
        else
        {
            redirect(site_url('admin/datakategori') . "?pesan=ubah");
        }
    }
}

public function inputsubkategori()
{
    if(isset($_POST['submit']))
    {
        $kodesubkategori=$this->input->post('kodesubkategori');
        $kodekategori=$this->input->post('kodekategori');
        $subkategori=$this->input->post('subkategori');

        $input=$this->m_barang->inputsubkategori($kodesubkategori,$kodekategori,$subkategori);

        if($input == false)
        {
            redirect('admin/datasubkategori/formdatasubkategori') . "?pesan=gagal";
        }
        else
        {
            redirect(site_url('admin/datasubkategori') . "?pesan=tambah");
        }
    }
}

public function editsubkategori()
{
    if(isset($_POST['submit']))
    {
        $kodesubkategori=$this->input->post('kodesubkategori');
        $kodekategori=$this->input->post('kodekategori');
        $subkategori=$this->input->post('subkategori');

        $input=$this->m_barang-
```



```
>editsubkategori($kodesubkategori,$kodekategori,$subkategori);

    if($input == false)
    {
        redirect('admin/datasubkategori/formdatasubkategoriedit') . "?pesan=gagal";
    }
    else
    {
        redirect(site_url('admin/datasubkategori') . "?pesan=ubah");
    }
}

public function inputdatasupplier()
{
    if(isset($_POST['submit']))
    {
        $kodesupplier=$this->input->post('kodesupplier');
        $namasupplier=$this->input->post('namasupplier');
        $alamat=$this->input->post('alamat');
        $telepon=$this->input->post('telepon');

        $input=$this->m_barang->
>inputdatasupplier($kodesupplier,$namasupplier,$alamat,$telepon);

        if($input == false)
        {
            redirect('admin/datasupplier/formdatasupplier') . "?pesan=gagal";
        }
        else
        {
            redirect(site_url('admin/datasupplier') . "?pesan=tambah");
        }
    }
}

public function editdatasupplier()
{
    if(isset($_POST['submit']))
    {
        $kodesupplier=$this->input->post('kodesupplier');
        $namasupplier=$this->input->post('namasupplier');
        $alamat=$this->input->post('alamat');
        $telepon=$this->input->post('telepon');

        $input=$this->m_barang->
>editdatasupplier($kodesupplier,$namasupplier,$alamat,$telepon);

        if($input == false)
        {
            redirect('admin/datasupplier/formdatasupplieredit') . "?pesan=gagal";
        }
        else
        {
            redirect(site_url('admin/datasupplier') . "?pesan=ubah");
        }
    }
}
```

```
public function inputdatauser()
{
    if(isset($_POST['submit']))
    {
        $kodepengguna=$this->input->post('kodepengguna');
        $namapengguna=$this->input->post('namapengguna');
        $sandi=$this->input->post('sandi');
        $level=$this->input->post('level');

        $input=$this->m_barang->inputdatauser($kodepengguna,$namapengguna,$sandi,$level);

        if($input == false)
        {
            redirect('admin/datauser/formdatauser') . "?pesan=gagal";
        }
        else
        {
            redirect(site_url('admin/datauser') . "?pesan=tambah");
        }
    }
}

public function editdatauser()
{
    if(isset($_POST['submit']))
    {
        $kodepengguna=$this->input->post('kodepengguna');
        $namapengguna=$this->input->post('namapengguna');
        $sandi=$this->input->post('sandi');
        $level=$this->input->post('level');

        $input=$this->m_barang->editdatauser($kodepengguna,$namapengguna,$sandi,$level);

        if($input == false)
        {
            redirect('admin/datauser/formdatauseredit') . "?pesan=gagal";
        }
        else
        {
            redirect(site_url('admin/datauser') . "?pesan=ubah");
        }
    }
}

public function inputpola()
{
    if(isset($_POST['submit']))
    {
        $kodepola=$this->input->post('kodepola');
        $tanggal=$this->input->post('tanggal');
        $periodeawal=$this->input->post('periodeawal');
        $periodeakhir=$this->input->post('periodeakhir');
        $maksimalscan=$this->input->post('scan');
        $minimalsupport=$this->input->post('minsupp');
        $minimalconfident=$this->input->post('minconf');
```

```

        $input=$this->m_barang-
>inputpola($kodepola,$tanggal,$periodeawal,$periodeakhir,$maksimalscan,$minimalsupport,$
minimalconfident);

        if($input == false)
        {
            redirect('admin/pencarianpola/pencarianpola') . "?pesan=gagal";
        }
        else
        {
            redirect(site_url('admin/pencarianpola') . "?pesan=tambah");
        }
    }
}
function get_nama() {
    $id_produk = $this->input->post("id_produk");
    echo $this->m_barang->nama($id_produk);
}
function get_harga() {
    $id_produk = $this->input->post("id_produk");
    echo $this->m_barang->harga($id_produk);
}
}
?>

```

D.2 Kode program *class* model/m_barang

```

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

class M_barang extends CI_Model {

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

    public function inputbarang($kodebarang,$kodekategori,$kodesubkategori,$kodesupplier,$nama
barang, $hargabeli,$hargajual,$stok, $status,$kodepengguna)
    {
        return $query = $this->db->query("INSERT INTO `databarang`
(`kodebarang`,`kodekategori`,`kodesubkategori`,`kodesupplier`,`namabarang`,`h
argabeli`,`hargajual`,`kodepengguna`)
VALUES

```

```

("$kodebarang.", "$kodekategori.", "$kodesubkategori.", "$kodesupplier.", '$
$.namabarang.', '$hargabeli.', '$hargajual.', '$kodepengguna.'");

}

public function getrating(){
    return $query = $this->db->query("SELECT db.namabarang as
namabarang, SUM( dt.jumlah ) as jumlah
FROM detaildatatransaksi dt join databarang db on
dt.kodebarang=db.kodebarang
GROUP BY dt.kodebarang
ORDER BY SUM( dt.jumlah ) DESC limit 3 ");
}

// public function inputharga
($nohistori,$kodebarang,$tanggalhistori,$hargabaruu)
// {
//     $kodepengguna = $this->session->userdata('kodepengguna');
//     $hargajuallama= $this->db->query("select hargajual from databarang
where kodebarang='".$kodebarang.'"")->result();
//     $harga_jual= $hargajuallama[0]->hargajual;

//     $this->db->query("INSERT INTO `historibarang` (`nohistori`,
`kodebarang`, `tanggalhistori`, `hargajual`, `kodepengguna`) VALUES (NULL,
'".$kodebarang."', now(), '$hargabaruu.', '$kodepengguna.'");
//     $this->db->query("update databarang set hargajual='".$hargabaruu.'"
where kodebarang='".$kodebarang.'");

// }

public function editbarang($kodebarang,$kodekategori,$kodesubkategori,$kodesupplier,$namab
arang, $hargabeli,$hargajual){

    $kodepengguna = $this->session->userdata('kodepengguna');
    $kodekategorilama= $this->db->query("select kodekategori from
databarang where kodebarang='".$kodebarang.'"")->result();
    $kodesubkategorilama= $this->db->query("select kodesubkategori from
databarang where kodebarang='".$kodebarang.'"")->result();kkk.
    $kodesupplierlama= $this->db->query("select kodesupplier from
databarang where kodebarang='".$kodebarang.'"")->result();
    $namabaranglama= $this->db->query("select namabarang from databarang
where kodebarang='".$kodebarang.'"")->result();
    $hargabelilama= $this->db->query("select hargabeli from databarang
where kodebarang='".$kodebarang.'"")->result();

```

```

    $hargajuallama= $this->db->query("select hargajual from databarang
where kodebarang='".$kodebarang.'"")->result();
    // $tanggalmasuklama= $this->db->query("select tanggalmasuk from
databarang where kodebarang='".$kodebarang.'"")->result();
    // $kadaluarsalama= $this->db->query("select kadaluarsa from databarang
where kodebarang='".$kodebarang.'"")->result();
    $kode_kategori= $kodekategorilama[0]->kodekategori;
    $kode_subkategori= $kodesubkategorilama[0]->kodesubkategori;
    $kode_supplier= $kodesupplierlama[0]->kodesupplier;
    $nama_barang= $namabaranglama[0]->namabarang;
    $harga_beli= $hargabelilama[0]->hargabeli;
    $harga_jual= $hargajuallama[0]->hargajual;
    // $tanggal_masuk= $tanggalmasuklama[0]->tanggalmasuk;
    // $kada_luarsa= $kadaluarsalama[0]->kadaluarsa;

    $datalama=$this->db->query("INSERT INTO `historibarang`
(`nohistori`,`kodebarang`,`tanggalhistori`,`kodekategori`,`kodesubkategori`,`ko
desupplier`,`namabarang`,`hargabeli`,`hargajual`,`kodepengguna`)VALUES
(NULL, '".$kodebarang.',now(),'".$kode_kategori.',

'".$kode_subkategori.', '".$kode_supplier.', '".$namabarang.', '".$hargabeli.', '".
$hargajual.', '".$kodepengguna.'");");
    $databaru=$this->db->query("update databarang set kodekategori=
'".$kodekategori.', kodesubkategori= '".$kodesubkategori.',kodesupplier=
'".$kodesupplier.',namabarang=
'".$namabarang.',hargabeli=
'".$hargabeli.',hargajual=
'".$hargajual.'" where kodebarang =
'".$kodebarang.'"");
    return $databaru;
}

// public function tampilhistoriharga($id)
// {
//     return $this->db->query("SELECT hargajual,hargabeli, kodebarang from
databarang where kodebarang='$id'");
// }

public function tampilbarangedit($id)
{
    return $this->db->query("SELECT b.kodebarang,b.kodekategori,
k.kodekategori,k.subkategori,p.namasupplier,
b.namabarang,b.hargabeli,b.hargajual,b.stok from databarang b join
datasubkategori k on (b.kodesubkategori=k.kodesubkategori) join datakategori
g on g.kodekategori=k.kodekategori join
datasupplier p on (b.kodesupplier= p.kodesupplier) where b.kodebarang=
'$id' ");

```

```
}

public function tampileditstatus($id)
{
    return $this->db->query("update databarang set status='nonaktif' WHERE
kodebarang = '$id'");
}

public function tampileditstatusaktif($id)
{
    return $this->db->query("update databarang set status='aktif' WHERE
kodebarang = '$id'");
}

public function kategoristatus($id)
{
    return $this->db->query("update datakategori set status='tidak aktif'
WHERE kodekategori = '$id'");
}

public function kategoristatusaktif($id)
{
    return $this->db->query("update datakategori set status='aktif' WHERE
kodekategori = '$id'");
}

public function subkategoristatus($id)
{
    return $this->db->query("update datasubkategori set status='tidak aktif'
WHERE kodesubkategori = '$id'");
}

public function subkategoristatusaktif($id)
{
    return $this->db->query("update datasubkategori set status='aktif' WHERE
kodesubkategori = '$id'");
}

public function supplierstatus($id)
{
    return $this->db->query("update datasupplier set status='tidak aktif'
WHERE kodesupplier = '$id'");
}

public function supplierstatusaktif($id)
{
```

```
        return $this->db->query("update datasupplier set status='aktif' WHERE
kodesupplier = '$id'");
    }

    public function penggunastatus($id)
    {
        return $this->db->query("update datauser set status='tidak aktif' WHERE
kodepengguna = '$id'");
    }

    public function penggunastatusaktif($id)
    {
        return $this->db->query("update datauser set status='aktif' WHERE
kodepengguna = '$id'");
    }

    public function tampiltambahstok($bulan,$tahun)
    {
        return $this->db->query("SELECT
d.kodebarang,d.namabarang,t.tanggaltransaksi,s.stok from databarang d join
detailtransaksibarang s on s.kodebarang=d.kodebarang join transaksibarang t on
t.notransaksi= s.notransaksi
where month(t.tanggaltransaksi) = '$bulan' and year(t.tanggaltransaksi)
= '$tahun'");
    }

    public function tampiltambahstoksemua()
    {
        return $this->db->query("SELECT
d.kodebarang,d.namabarang,t.tanggaltransaksi,s.stok from databarang d join
detailtransaksibarang s on s.kodebarang=d.kodebarang join transaksibarang t on
t.notransaksi= s.notransaksi ");
    }

    // public function transaksibarangdetail($id)
    // {
    //     return $this->db->query("SELECT o.kodebarang,d.stok,o.namabarang
from detailtransaksibarang d join databarang o on o.kodebarang=d.kodebarang
where o.kodebarang='$id'");
    // }

    public function simpantambahstok()
    {
        // $id_user = $this->session->userdata('session_userid');
        $kodepengguna = $this->session->userdata('kodepengguna');
```

```
$id_barang = $this->input->post("id_produk");
$jumlah = $this->input->post("jumlah");
$status = "aktif";

// $data_harga_barang = $this->input->post("harga_produk");

$query = $this->db->query
(
    "INSERT INTO transaksibarang
    (
        tanggaltransaksi, pengguna
    )
    VALUES(NOW(), '$kodepengguna')"
);

$notransaksi = $this->db->insert_id();

for($i = 0; $i < count($id_barang); $i++){
    $query = $this->db->query
    (
        "INSERT INTO detailtransaksibarang
        (
            nodetailtransaksi
            , kodebarang
            , stok
        )
        VALUES('$notransaksi', '$id_barang[$i]', '$jumlah[$i]')"
    );
}

for($i = 0; $i < count($id_barang); $i++){
    $query = $this->db->query
    (
        "UPDATE databarang
        SET stok = (stok + $jumlah[$i]), status = '$status'
        WHERE kodebarang='$id_barang[$i]' "
    );
}

return true;
}

function subkategori($idkategori) {
    $query = $this->db->query
    (
```



```
"SELECT
    subkategori as subkategori,kodesubkategori
FROM
    datasubkategori
WHERE
    kodekategori = '$idkategori' "
);

return $query;
}

// public function simpantambahstok ()
// {
//     $kodepengguna = $this->session->userdata('kodepengguna');
//     $stok= $this->db->query("select stok from databarang where
kodebarang='$kodebarang'")->result();
//     $stokLama = $stok[0]->stok;

//     $transaksibarang = $this->db->query("INSERT INTO `transaksibarang`
(`notransaksi`, `kodebarang`, `tanggaltransaksi`, `jumlah`,`kodepengguna`)
VALUES (NULL, '$kodebarang', now(), '$jumlah','$kodepengguna');");
//     $databaru = $this->db->query("update databarang set stok=($stokLama
+ $jumlah) where kodebarang='$kodebarang'");
//     return $databaru;

// }

public function tampilbarang()
{
    return $this->db->query("SELECT
d.kodebarang,d.namabarang,s.namasupplier,d.hargabeli,d.hargajual,d.stok,d.stat
us from databarang d join datasupplier s on d.kodesupplier=s.kodesupplier");
}

public function tampilbarang2()
{
    return $this->db->query("SELECT kodebarang, namabarang,hargajual,stok
from databarang");
}

// public function tampiltransaksimasuk()
// {
//     return $this->db->query("SELECT * from transaksibarang");
// }
```

```

public function inputpola
($kodepola,$tanggal,$periodeawal,$periodeakhir,$maksimalscan,$minimalsupport,$minimalconfident)
{
    $kodepengguna = $this->session->userdata('kodepengguna');
    return $query=$this->db->query("INSERT INTO `pola`
(`kodepola`,`tanggal`,`periodeawal`,`periodeakhir`,`makscan`,`minsupp`,`minconf`,`kodepengguna`)
values('".$kodepola."',now(),'".$periodeawal."','".$periodeakhir."','".$maksimalscan."','".$minimalsupport."','".$minimalconfident."','".$kodepengguna."");");
}

public function tampilpola($kode = null)
{
    if(empty($kode))
    {
        return $this->db->query("SELECT * from pola");
    }
    else
    {
        return $this->db->get_where('pola', array('kodepola' => $kode));
    }
}

// public function tampilpolahapus($id)
// {
//     return $this->db->query("DELETE FROM pola WHERE kodepola = '$id'");
// }

// public function

// public function tampildatapilih($pa,$pd)
// {
//     $this->db->query("TRUNCATE TABLE `ofi`");
//     $jumtransak =0;
//     $banyaktransak = $query=$this->db->query("SELECT count(kodetransaksi) as jumlah from datatransaksi where tanggaltransaksi >= '$pa' and tanggaltransaksi <= '$pd'");
//     foreach ($banyaktransak->result_array() as $data) {
//         $jumtransak = $data['jumlah'];
//     }
}

```

```
// $barjum = $this->db->query("SELECT d.namabarang,d.kodebarang,
count(dd.kodetransaksi) as jumlah FROM datatransaksi dt join
detaildatatransaksi dd on (dt.kodetransaksi=dd.kodetransaksi) join databarang d
on(d.kodebarang=dd.kodebarang) where tanggaltransaksi >= '$pa' and
tanggaltransaksi <= '$pd' group by d.kodebarang ");
// foreach ($barjum->result_array() as $data2){
//     $namabarang = $data2['namabarang'];
//     $jumlah = $data2 ['jumlah'];
//     $kode = $data2['kodebarang'];
//     $OFI = ($jumlah/$jumtransak) * 100;
//     $this->db->query("INSERT into ofi values (','$kode',$OFI)");
// }
// }

public function tampilItem($pa,$pd)
{
    return $this->db->query("SELECT
d.kodetransaksi,group_concat(b.namabarang separator ',')as namabarang
FROM `detaildatatransaksi` d join `databarang` b on
(d.kodebarang=b.kodebarang) join datatransaksi dt on (d.kodetransaksi =
dt.kodetransaksi)
where tanggaltransaksi >= '$pa' and tanggaltransaksi <= '$pd'
group by d.kodetransaksi");
}

public function tampilItemsemua()
{
    return $this->db->query("SELECT group_concat(b.namabarang separator
',' )as namabarang
FROM `detaildatatransaksi` d join `databarang` b on
(d.kodebarang=b.kodebarang) join datatransaksi dt on (d.kodetransaksi =
dt.kodetransaksi)
group by d.kodetransaksi");
}

// public function tampilFI($minsupp, $pa,$pd)
// {
//     return $this->db->query("SELECT d.namabarang,
count(dd.kodetransaksi) as jumlahbarang, o.hasil
//     from datatransaksi dt join detaildatatransaksi dd on
(dt.kodetransaksi=dd.kodetransaksi)
//     join databarang d on(d.kodebarang=dd.kodebarang) join ofi o on
(o.kodebarang = d.kodebarang)
```

```
//      where tanggaltransaksi >= '$pa' and tanggaltransaksi <= '$pd' and
o.hasil >= ".$minsupp." group by d.kodebarang order by o.hasil desc");

// }

// public function tampilOFI($minsupp, $pa, $pd)
// {
//      return $this->db->query("SELECT d.kodetransaksi,group_concat(
b.namabarang order by o.hasil desc separator ',' )as namaBarang
//      FROM `detaildatatransaksi` d join `databarang` b on
(d.kodebarang=b.kodebarang)
//      join `ofi` o on (d.kodebarang=o.kodebarang) join datatransaksi dt on
(d.kodetransaksi = dt.kodetransaksi)
//      where tanggaltransaksi >= '$pa' and tanggaltransaksi <= '$pd' and
o.hasil >= ".$minsupp."
//      group by d.kodetransaksi order by d.kodebarang desc,o.hasil desc");
// }

// public function tampilFIS($minsupp, $pa, $pd)
// {
//      return $this->db->query("SELECT d.namabarang as item,
count(dd.kodetransaksi) as CPB, o.hasil as CFT, o.hasil as FIS
//      from datatransaksi dt join detaildatatransaksi dd on
(dt.kodetransaksi=dd.kodetransaksi)
//      join databarang d on(d.kodebarang=dd.kodebarang) join ofi o on
(o.kodebarang = d.kodebarang)
//      where tanggaltransaksi >= '$pa' and tanggaltransaksi <= '$pd' and
o.hasil >= ".$minsupp." group by d.kodebarang order by o.hasil desc");
// }

// public function tampilrekom($cft)
// {
//      $result = array();
//      foreach ($cft as $key => $value) {
//          $result[$key] = array();
//          foreach ($value as $index => $val) {
//              array_push($result[$key], $this->rekom(str_replace('_', " ",
$index), str_replace('_', " ", $key)));
//          }
//      }
//      return $result;
// }
```

```

// public function rekom($nama, $pasangan)
// {
//     return $this->db->query("SELECT namabarang , sum(jumlah) as jum
FROM `databarang` b left join detaildatatransaksi d
//     ON b.kodebarang = d.kodebarang
//     WHERE b.kodesubkategori = (SELECT kodesubkategori FROM
databarang WHERE namabarang like '%".$nama."' limit 1) AND
//     b.namabarang not like '%".$pasangan."'
//     GROUP BY b.kodebarang ORDER BY JUM asc LIMIT 1 ")
->result_array()[0]['namabarang'];
// }

// public function rekom($subkat)
// {
//     return $this->db->query("SELECT b.namabarang as namabarang from
detaildatatransaksi d right join databarang b on d.kodebarang=b.kodebarang
join datasubkategori ds on ds.kodesubkategori=b.kodesubkategori where
b.namabarang <> '$subkat' ");
// }

// public function rekom($subkat)
// {
//     //PROCEDURE (--batas pengulangan query--, --kodepola--, --nama
barang--)
//     //mengulang hasil query dengan nama barang yang berbed a
//     //var_dump($subkat);
//     if(!$subkat[1]) {
//         return $this->db->query("SELECT 0 AS namabarang");
//     } else {
//         for ($i=0; $i < count($subkat[1]); $i++) {
//             //var_dump($subkat[1]);
//             return $this->db->query("SELECT b.namabarang, SUM(s.jumlah)
as jumlah
//             FROM datasubkategori g join databarang b on
b.kodesubkategori=g.kodesubkategori join detaildatatransaksi s on
s.kodebarang=b.kodebarang join datatransaksi t on
s.kodetransaksi=t.kodetransaksi
//             WHERE (t.tanggaltransaksi between (select periodeawal from pola
where kodepola='".$subkat[0]."' ) and
//             (select periodeakhir from pola where
kodepola='".$subkat[0]."' )
// AND subkategori =
//             (SELECT g.subkategori FROM datasubkategori g join databarang b
on b.kodesubkategori=g.kodesubkategori
//             where b.namabarang LIKE '".$subkat[1][$i]."' ) AND
b.namabarang NOT LIKE '".$subkat[1][$i]."'

```

```

// GROUP BY b.namabarang
// ORDER BY SUM(s.jumlah) LIMIT 1");
//     }
//     }

// }

public function rekom($subkat)
{
    if(!$subkat[1]) {
        return $this->db->query("SELECT 0 AS namabarang");
    } else {
        $sql_array = array();
        foreach ($subkat[1] as $key => $value) {
            $sql_array[] = "(SELECT b.namabarang, SUM(s.jumlah) as
jumlah
FROM datasubkategori g join databarang b on
b.kodesubkategori=g.kodesubkategori join detaildatatransaksi s on
s.kodebarang=b.kodebarang join datatransaksi t on
s.kodetransaksi=t.kodetransaksi
WHERE (t.tanggaltransaksi between (select periodeawal from pola
where kodepola='".$subkat[0]."' ) and
(select periodeakhir from pola where
kodepola='".$subkat[0]."' )AND subkategori =
(SELECT g.subkategori FROM datasubkategori g join databarang b
on b.kodesubkategori=g.kodesubkategori
where b.namabarang LIKE '".$value."') AND b.namabarang NOT
LIKE '".$value.'"
GROUP BY b.namabarang
ORDER BY SUM(s.jumlah) LIMIT 1";
}
$sql = implode(" UNION ALL ", $sql_array);
return $this->db->query($sql . " ");
//     $query_bagian = array();
//     foreach ($subkat[1] as $val) {
//         $query_bagian[] = "%".mysql_real_escape_string($val)."%";
//     }
//     $tulisan = implode(' OR b.namabarang LIKE ', $query_bagian);
//     $tulisan2 = implode(' OR b.namabarang NOT LIKE ',
$query_bagian);
//     return $this->db->query("SELECT b.namabarang, SUM(s.jumlah)
as jumlah
//     FROM datasubkategori g join databarang b on
b.kodesubkategori=g.kodesubkategori join detaildatatransaksi s on
s.kodebarang=b.kodebarang join datatransaksi t on

```

```

s.kodetransaksi=t.kodetransaksi
//          WHERE (t.tanggaltransaksi between (select periodeawal from pola
where kodepola=".$subkat[0].") and
//          (select periodeakhir from pola where
kodepola=".$subkat[0]."))
// AND subkategori =
//          (SELECT g.subkategori FROM datasubkategori g join databarang b
on b.kodesubkategori=g.kodesubkategori
//          where b.namabarang LIKE ".$tulisannya.") AND b.namabarang NOT
LIKE ".$tulisannya2."
// GROUP BY b.namabarang
// ORDER BY SUM(s.jumlah)");
// }
}

public function pecahkode($kode){
return $this->db->query("SELECT b.kodesubkategori from databarang b
join datasubkategori ds on ds.kodesubkategori=b.kodesubkategori where
b.namabarang='$kode'");
}

public function tampildatatransak($bulan,$tahun)
{
return $this->db->query("SELECT dt.kodetransaksi as kodetransaksi,
dt.tanggaltransaksi as tanggaltransaksi, SUM(dtd.hargatotal) as hargatotal from
datatransaksi dt join detaildatatransaksi dtd on dt.kodetransaksi =
dtd.kodetransaksi
where month(tanggaltransaksi) = '$bulan' and year(tanggaltransaksi) =
'$tahun'");
}

public function tampildatatransaksemua()
{
return $this->db->query("SELECT dt.kodetransaksi as kodetransaksi,
dt.tanggaltransaksi as tanggaltransaksi, SUM(dtd.hargatotal) as hargatotal from
datatransaksi dt join detaildatatransaksi dtd on dt.kodetransaksi =
dtd.kodetransaksi group by dtd.kodetransaksi ");
}

// public function tampildatatrans($a = '1970-1-1', $b = '2099-12-30')
// {
// return $this->db->query("SELECT * from datatransaksi where

```

```

tanggaltransaksi BETWEEN ".$a." AND ".$b."");
// }

public function inputkategori ($kodekategori,$kategori,$kodepengguna)
{
    $kodepengguna = $this->session->userdata('kodepengguna');
    return $query=$this->db->query("INSERT INTO
`datakategori`(`kodekategori`,`kategori`,`kodepengguna`)
values('".$kodekategori."','".$kategori."','".$kodepengguna."");
}

public function tampilkategori()
{
    return $this->db->query("SELECT * from datakategori");
}

public function editkategori ($kodekategori,$kategori)
{
    return $this->db->query("update datakategori set kategori='".$kategori'
where kodekategori='".$kodekategori'");
}

public function tampilkategoriedit($id)
{
    return $this->db->query("SELECT * from datakategori WHERE
kodekategori = '$id' ");
}

public function tampilkategorihapus($id)
{
    return $this->db->query("DELETE FROM datakategori WHERE
kodekategori = '$id'");
}

public function inputsubkategori($kodesubkategori,$kodekategori,$subkategori)
{
    $kodepengguna = $this->session->userdata('kodepengguna');
    return $query=$this->db->query("INSERT INTO
`datasubkategori`(`kodesubkategori`,`kodekategori`,`subkategori`,`Kodepenggu
na`)
values('".$kodesubkategori."','".$kodekategori."','".$subkategori."','".$kodepengguna."
'");
}

```



```

// public function tampilsubkategori($id= '')
// {
//     return $this->db->query("SELECT * from datasubkategori where
kodekategori='".$id."'");
// }

public function tampilsubkategoritabel()
{
return $this->db->query("SELECT s.kodesubkategori as
kodesubkategori,k.kategori as namakategori, s.subkategori as subkategori,
s.status as status from datasubkategori s join datakategori k on
k.kodekategori=s.kodekategori");
}

public function editsubkategori($kodesubkategori,$kodekategori,$subkategori)
{
return $this->db->query("update datasubkategori set kodekategori=
'".$kodekategori."', subkategori= '".$subkategori.'" WHERE kodesubkategori =
'".$kodesubkategori.'");
}

public function tampilsubkategoriedit($id)
{
return $this->db->query("SELECT * from datasubkategori where
kodesubkategori = '$id'");
}

public function tampilinfotransaksi($id)
{
return $this->db->query("SELECT b.namabarang, d.jumlah from
detaildatatransaksi d join databarang b on b.kodebarang=d.kodebarang where
kodetransaksi='$id'")->result();
}

// public function tampilsubkategorihapus($id)
// {
//     return $this->db->query("DELETE FROM datasubkategori WHERE
kodesubkategori = '$id'");
// }

public function inputdatasupplier($kodesupplier,$namasupplier,$alamat,$telepon)
{
$kodepengguna = $this->session->userdata('kodepengguna');
return $query=$this->db->query("INSERT INTO

```

```

`datasupplier`(`kodesupplier`,`namasupplier`,`alamat`,`telepon`,`kodepengguna`
)
values
(".$kodesupplier.", ".$namasupplier.", ".$alamat.", ".$telepon.", ".$kodepengg
una."));
}

public function tampilsupplier()
{
return $this->db->query("SELECT * from datasupplier");
}

public function editdatasupplier($kodesupplier,$namasupplier,$alamat,$telepon)
{
return $query=$this->db->query(" update datasupplier set namasupplier=
".$namasupplier.", alamat= ".$alamat.", telepon= ".$telepon." where
kodesupplier='$kodesupplier'");
}

public function tampilsupplieredit($id)
{
return $this->db->query("SELECT * from datasupplier where
kodesupplier= '$id'");
}

public function tampilsupplierhapus($id)
{
return $this->db->query("DELETE from datasupplier where kodesupplier=
'$id'");
}

public function inputdatauser($kodepengguna,$namapengguna,$sandi,$level)
{
return $query=$this->db->query("INSERT INTO
`datauser`(`kodepengguna`,`namapengguna`,`sandi`,`level`)
('.$kodepengguna.', ".$namapengguna.", ".$sandi.", ".$level."));
}

public function tampiluser()
{
return $this->db->query("SELECT * from datauser");
}

public function editdatauser($kodepengguna,$namapengguna,$sandi,$level)
{

```

```
        return $query=$this->db->query("update datauser set namapengguna=
        ".$namapengguna."', sandi= ".$sandi."', level= ".$level.'" where
        kodepengguna='$kodepengguna'");
    }

    public function tampiluseredit($id)
    {
        return $this->db->query("SELECT * from datauser where
        kodepengguna='$id'");
    }

    // public function tampiluserhapus($id)
    // {
    //     return $this->db->query("DELETE from datauser where
    kodepengguna='$id'");
    // }

    // public function tampildata()
    // {
    //     $query="select b.kodebarang, b.namabarang, b.hargajual, k.kategori
    from databarang as b, datakategori as k where b.kodekategori=k.kodekategori";
    //     return $this->db->query($query);
    // }

    public function getlastidbarang()
    {
        $kode_before = $this->db->query("SELECT kodebarang from databarang
        order by kodebarang desc limit 1")->result_array();

        if(empty($kode_before))
        {
            $kode_after = 'BRG00001';
        }
        else{
            $kode_before = $kode_before[0]['kodebarang'];
            $digit_before = substr($kode_before,3,5) + 1;
            $digit_after = str_pad($digit_before, 5, 0, STR_PAD_LEFT);
            $kode_after = 'BRG'.$digit_after;
        }
        return $kode_after;
    }

    public function getlastidkategori()
    {
        $kode_before = $this->db->query("SELECT kodekategori from
        datakategori order by kodekategori desc limit 1")->result_array();
```

```
if(empty($kode_before))
{
    $kode_after = 'KTG00001';
}
else{
    $kode_before = $kode_before[0]['kodekategori'];
    $digit_before = substr($kode_before,3,5) + 1;
    $digit_after = str_pad($digit_before, 5, 0, STR_PAD_LEFT);
    $kode_after = 'KTG'.$digit_after;
}
return $kode_after;
}

public function getlastidsubkategori()
{
    $kode_before = $this->db->query("SELECT kodesubkategori from
datasubkategori order by kodesubkategori desc limit 1")->result_array();

    if(empty($kode_before))
    {
        $kode_after = 'SKG00001';
    }
    else{
        $kode_before = $kode_before[0]['kodesubkategori'];
        $digit_before = substr($kode_before,3,5) + 1;
        $digit_after = str_pad($digit_before, 5, 0, STR_PAD_LEFT);
        $kode_after = 'SKG'.$digit_after;
    }
    return $kode_after;
}

public function getlastidsupplier()
{
    $kode_before = $this->db->query("SELECT kodesupplier from
datasupplier order by kodesupplier desc limit 1")->result_array();

    if(empty($kode_before))
    {
        $kode_after = 'SPY00001';
    }
    else{
        $kode_before = $kode_before[0]['kodesupplier'];
        $digit_before = substr($kode_before,3,5) + 1;
        $digit_after = str_pad($digit_before, 5, 0, STR_PAD_LEFT);
        $kode_after = 'SPY'.$digit_after;
    }
}
```

```
}
return $kode_after;
}

public function getlastiduser()
{
    $kode_before = $this->db->query("SELECT kodepengguna from datauser
order by kodepengguna desc limit 1")->result_array();

    if(empty($kode_before))
    {
        $kode_after = 'USR00001';
    }
    else{
        $kode_before = $kode_before[0]['kodepengguna'];
        $digit_before = substr($kode_before,3,5) + 1;
        $digit_after = str_pad($digit_before, 5, 0, STR_PAD_LEFT);
        $kode_after = 'USR'.$digit_after;
    }
    return $kode_after;
}

public function getlastidpola()
{
    $kode_before = $this->db->query("SELECT kodepola from pola order by
kodepola desc limit 1")->result_array();

    if(empty($kode_before))
    {
        $kode_after = 'PLA00001';
    }
    else{
        $kode_before = $kode_before[0]['kodepola'];
        $digit_before = substr($kode_before,4,5) + 1;
        $digit_after = str_pad($digit_before, 5, 0, STR_PAD_LEFT);
        $kode_after = 'PLA'.$digit_after;
    }
    return $kode_after;
}

// public function ambilkodetransaksi()
// {
//     // Ambil data terakhir
//     $query = "SELECT notransaksi FROM transaksibarang ORDER BY
notransaksi desc LIMIT 1";
//     $kode = $this->db->query($query);
```

```
// // Cek ada data apa tidak
// if($kode->num_rows() > 0){
//     // Jika ada datanya....
//     // Ambil BARIS pertama
//     $kode = $kode->result_array();
//     // Ambil data pada KOLOM kode
//     $kode = $kode[0]['notransaksi'];
//     // Ditambah 1
//     $kode++;
// }else{
//     // Jika data kosong, maka kode = 1
//     $kode = 1;
// }
// return $kode;
// }
function nama($id_produk) {
    $query = $this->db->query
    (
        "SELECT
            namabarang as namabarang
        FROM
            databarang
        WHERE
            kodebarang = '$id_produk'"
    );

    return $query->row()->namabarang;
}

function harga($id_produk) {
    $query = $this->db->query
    (
        "SELECT
            hargajual as harga
        FROM
            databarang
        WHERE
            kodebarang = '$id_produk'"
    );

    return $query->row()->harga;
}

}

?>
```

Lampiran E. Pengujian

E.Black Box

Tabel E.1 Pengujian *Black Box*

No	Menu	Fungsi	Aksi	Hasil	Ket
1.	Login	Menu login merupakan menu awal yang digunakan untuk autentikasi pengguna, jika ingin masuk ke dalam sistem	Mengisi <i>username</i> dan <i>password</i> lalu menekan tombol “Login”	Login berhasil dan menampilkan halaman awal masing-masing aktor	√
			<i>Username</i> atau <i>Password</i> salah	Menampilkan pesan “ <i>Username</i> atau <i>Password</i> SALAH, Silahkan periksa dan coba kembali”	√
			<i>Username</i> (kosong) atau <i>password</i> (kosong)	Menampilkan pesan “ <i>field</i> harus diisi” dibawah <i>field</i> yang kosong	√
2.	Manajemen Barang Submenu Data Barang	Sub menu Data Barang digunakan untuk melihat, menambah, merubah data barang dan merubah status data barang	Memilih menu manajemen barang	Menampilkan submenu manajemen barang	√
			Memilih submenu data barang	Menampilkan data barang pada halaman data barang	√
			Menekan tombol “Tambah Data”	Menampilkan form data barang	√
			Menekan tombol “Kembali” pada form data barang	Menampilkan halaman data barang	√
			<i>Field</i> belum diisi	Menampilkan pesan “ <i>field</i> harus diisi” dibawah <i>field</i> yang kosong	√
			Menekan tombol “Simpan” pada form data barang	Menyimpan data form data barang dan menampilkan halaman data barang	√
		Menekan tombol “ <i>Edit</i> ” pada	Menampilkan form <i>edit</i> barang yang dipilih	√	

No	Menu	Fungsi	Aksi	Hasil	Ket
				baris data	
			Menekan tombol “Kembali” pada form <i>edit</i> barang	Menampilkan halaman data barang	√
			Salah satu <i>field</i> kosong	Menampilkan pesan “ <i>field</i> harus diisi” dibawah <i>field</i> yang kosong	√
			Menekan tombol “Simpan” pada form <i>edit</i> barang	Menyimpan data barang dan menampilkan halaman data braang	√
			Menekan tombol aktif	Merubah status data barang dan menampilkan status baru di halaman data barang kolom status	√
			Menekan tombol nonaktif	Merubah status data barang dan menampilkan status baru di halaman data barang kolom status	√
3.	Manajemen Barang Submenu Stok Barang	Submenu stok barang digunakan untuk melihat data stok barang dan input data stok barang	Memilih menu manajemen barang	Menampilkan submenu manajemen barang	√
			Memilih submenu stok barang	Menampilkan data stok barang pada halaman stok barang	√
			Menekan tombol “Tambah Stok”	Menampilkan form data barang	√
4.	Manajemen Barang Submenu Data Kategori	Submenu data kategori digunakan untuk melihat, menambah, merubah data kategori dan merubah status data kategori	Memilih menu manajemen barang	Menampilkan submenu manajemen barang	√
			Memilih submenu data kategori	Menampilkan data kategori pada halaman kategori	√
			Menekan tombol “Tambah Data”	Menampilkan form kategori	√
			Menekan tombol “Kembali” pada form kategori	Menampilkan halaman kategori	√

No	Menu	Fungsi	Aksi	Hasil	Ket
			<i>Field</i> belum diisi	Menampilkan pesan “ <i>field</i> harus diisi” dibawah <i>field</i> yang kosong	√
			Menekan tombol “Simpan” pada form kategori	Menyimpan data form kategori dan menampilkan halaman kategori	√
			Menekan tombol “ <i>Edit</i> ” pada baris data	Menampilkan form <i>edit</i> kategori yang dipilih	√
			Menekan tombol “Kembali” pada form <i>edit</i> kategori	Menampilkan halaman kategori	√
			Salah satu <i>field</i> kosong	Menampilkan pesan “ <i>field</i> harus diisi” dibawah <i>field</i> yang kosong	√
			Menekan tombol “Simpan” pada form <i>edit</i> kategori	Menyimpan data kategori dan menampilkan halaman kategori	√
			Menekan tombol aktif	Merubah status data kategori dan menampilkan status baru di halaman data kategori kolom status	√
			Menekan tombol nonaktif	Merubah status data kategori dan menampilkan status baru di halaman kategori kolom status	√
5.	Manajemen Data Barang Submenu Data subkategori	Submenu subkategori digunakan untuk melihat, menambah, merubah status data subkategori dan merubah status data subkategori	Memilih menu manajemen barang	Menampilkan submenu manajemen barang	√
			Memilih submenu data subkategori	Menampilkan data subkategori pada halaman subkategori	√
			Menekan tombol “Tambah Data”	Menampilkan form subkategori	√
			Menekan tombol “Kembali” pada form subkategori	Menampilkan halaman subkategori	√

No	Menu	Fungsi	Aksi	Hasil	Ket
			<i>Field</i> belum diisi	Menampilkan pesan “ <i>field</i> harus diisi” dibawah <i>field</i> yang kosong	√
			Menekan tombol “Simpan” pada form subkategori	Menyimpan data form subkategori dan menampilkan halaman subkategori	√
			Menekan tombol “ <i>Edit</i> ” pada baris data	Menampilkan form <i>edit</i> subkategori yang dipilih	√
			Menekan tombol “Kembali” pada form <i>edit</i> subkategori	Menampilkan halaman subkategori	√
			Salah satu <i>field</i> kosong	Menampilkan pesan “ <i>field</i> harus diisi” dibawah <i>field</i> yang kosong	√
			Menekan tombol “Simpan” pada form <i>edit</i> subkategori	Menyimpan data subkategori dan menampilkan halaman subkategori	√
			Menekan tombol aktif	Merubah status data subkategori dan menampilkan status baru di halaman data subkategori kolom status	√
			Menekan tombol nonaktif	Merubah status data subkategori dan menampilkan status baru di halaman subkategori kolom status	√
6.	Data Supplier	Menu data supplier digunakan untuk melihat, menambah, merubah data supplier dan merubah status supplier	Memilih menu data supplier	Menampilkan data supplier pada halaman supplier	√
			Menekan tombol “Tambah Data”	Menampilkan form supplier	√
			Menekan tombol “Kembali” pada form supplier	Menampilkan halaman supplier	√
			<i>Field</i> belum diisi	Menampilkan pesan “ <i>field</i> harus diisi” dibawah <i>field</i> yang kosong	√
			Menekan tombol “Simpan” pada	Menyimpan data form supplier dan menampilkan	√

No	Menu	Fungsi	Aksi	Hasil	Ket
			form supplier	halaman supplier	
			Menekan tombol “ <i>Edit</i> ” pada baris data	Menampilkan form <i>edit</i> supplier yang dipilih	√
			Menekan tombol “Kembali” pada form <i>edit</i> supplier	Menampilkan halaman supplier	√
			Salah satu <i>field</i> kosong	Menampilkan pesan “ <i>field</i> harus diisi” dibawah <i>field</i> yang kosong	√
			Menekan tombol “Simpan” pada form <i>edit</i> supplier	Menyimpan data supplier dan menampilkan halaman supplier	√
			Menekan tombol aktif	Merubah status data supplier dan menampilkan status baru di halaman data supplier kolom status	√
			Menekan tombol nonaktif	Merubah status data supplier dan menampilkan status baru di halaman supplier kolom status	√
7.	Data User	Menu data user digunakan untuk melihat, menambah, merubah data user dan merubah status data user	Memilih menu supplier	Menampilkan data user di halaman user	√
			Menekan tombol “Tambah Data”	Menampilkan form user	√
			Menekan tombol “Kembali” pada form user	Menampilkan halaman user	√
			<i>Field</i> belum diisi	Menampilkan pesan “ <i>field</i> harus diisi” dibawah <i>field</i> yang kosong	√
			Menekan tombol “Simpan” pada form user	Menyimpan data form user dan menampilkan halaman user	√
			Menekan tombol “ <i>Edit</i> ” pada	Menampilkan form <i>edit</i> user yang dipilih	√

No	Menu	Fungsi	Aksi	Hasil	Ket
			baris data		
			Menekan tombol “Kembali” pada form <i>edit</i> user	Menampilkan halaman user	√
			Salah satu <i>field</i> kosong	Menampilkan pesan “ <i>field</i> harus diisi” dibawah <i>field</i> yang kosong	√
			Menekan tombol “Simpan” pada form <i>edit</i> user	Menyimpan data user dan menampilkan halaman user	√
			Menekan tombol aktif	Merubah status data user dan menampilkan status baru di halaman data user kolom status	√
			Menekan tombol nonaktif	Merubah status data user dan menampilkan status baru di halaman user kolom status	√
8.	Data Transaksi (admin)	Menu data transaksi digunakan untuk melihat data transaksi penjualan oleh admin	Memilih menu data transaksi	Menampilkan data transaksi pada halaman data transaksi penjualan	√
			Menekan tombol “Detail”	Menampilkan halaman detail transaksi	√
9.	Data Pola	Menu data pola digunakan untuk melihat data parameter pola, menambah parameter pola, melihat hasil rekomendasi promosi	Memilih menu data pola	Menampilkan data parameter pencarian pola pada halaman pencarian pola	√
			Menekan tombol +Data Pola	Menampilkan halaman form parameter	
			Menekan tombol “Kembali” di halaman form parameter	Menampilkan halaman data pola	√
			Menekan tombol ”proses” dihalaman data pola pada baris data	Menampilkan halaman rekomendasi	√
10.	Beranda	Menu beranda di gunakan untuk melihat beranda	Memilih menu beranda	Menampilkan data pada halaman beranda	√

No	Menu	Fungsi	Aksi	Hasil	Ket
11.	Data Barang (pegawai)	Menu data barang digunakan untuk melihat data barang	Memilih menu data barang	Menampilkan databarang pada halaman data barang	√
12.	Transaksi Penjualan	Menu transaksi penjualan digunakan untuk melihat data transaksi dan menambah data transaksi	Memilih transaksi penjualan	Menampilkan data transaksi pada halamann transaksi penjualan	√
			Klik tombol “Detail” pada baris data	Menampilkan detail data transaksi di halaman detail transaksi	√
			Menekan tombol “Kembali” di halaman detail transaksi	Menampilkan halaman transaksi penjualan	√
			Menekan tombol “Tambah Transaksi”	Menampilkan form tambah transaksi	√
			Menekan tombol “tambah” pada form tambah transaksi	Menampilkan pesan “data tidak lengkap”	√
			Menekan tombol “Simpan” pada form tambah transaksi	Menyimpan data transaksi kemudian menampilkan halaman transaksi penjualan	√

Lampiran F. Perhitungan Algoritma Apriori

F. Perhitungan Algoritma Apriori

Perhitungan algoritma *Apriori*, membutuhkan parameter pencarian pola yaitu berupa nilai minimum *support* dan nilai minimal *confidence*. Nilai minimum *support* digunakan untuk menentukan nilai *frequent itemsets* dan juga nilai minimal persentase *confidence*. *Dataset* transaksi yang digunakan berupa data sampel transaksi penjualan.. Sampel yang digunakan dalam perhitungan algoritma *Apriori* kali ini menggunakan 20 transaksi dengan 4 jenis produk. Sampel transaksi penjualan produk terdapat pada Tabel F.1.

Tabel F. 1 Sample Transaksi Penjualan Produk

No.	Transaksi
1	Citra White, Lux Pink
2	Ciptadent Cool, Citra White
3	Ciptadent Cool
4	Citra White, Lux Pink
5	Sunsilk Hitam
6	Ciptadent Cool, Citra White
7	Citra White, Lux Pink, Sunsilk Hitam
8	Ciptadent Cool, Citra White
9	Ciptadent Cool, Sunsilk Hitam
10	Ciptadent Cool, Citra White
11	Citra White, Ciptadent Cool
12	Lux Pink, Citra White, Ciptadent Cool
13	Citra White, Ciptadent Cool
14	Citra White, Lux Pink
15	Ciptadent Cool, Citra White
16	Ciptadent Cool
17	Citra White, Ciptadent Cool
18	Citra White, Lux Pink
19	Sunsilk Hitam, Citra White, Lux Pink
20	Sunsilk Hitam

Tahap awal cara kerja dari perhitungan algoritma *Apriori* adalah dengan melakukan pengkodean pada jenis barang dan transaksi penjualan barang. Pengkodean pada transaksi penjualan barang dapat dilakukan dengan kode 1 jika membeli, dan 0 jika tidak membeli. Kode barang terdapat pada tabel F.2 dan kode transaksi penjualan barang terdapat pada tabel F.3

Tabel F. 2 Kode Barang

Produk	Kode
Lux Pink	a
Citra White	b
Sunsilk Hitam	c
Ciptadent Cool	d

Tabel F. 3 Kode Transaksi Penjualan Barang

Transaksi	a	b	c	d
1	1	1	0	0
2	0	1	0	1
3	0	0	0	1
4	1	1	0	0
5	0	0	1	0
6	0	1	0	1
7	1	1	1	0
8	0	1	0	1
9	0	0	1	1
10	0	1	0	1
11	0	1	0	1
12	1	1	0	1
13	0	1	0	1
14	1	1	0	0
15	0	1	0	1
16	0	0	0	1

17	0	1	0	1
18	1	1	0	0
19	1	1	1	0
20	0	0	1	0

Langkah selanjutnya adalah menentukan nilai minimal *frequent itemsets* berupa nilai *support*. Nilai minimal support pada perhitungan kali ini adalah 3. Perhitungan untuk 4 jenis barang adalah 12 pasang. Penjabaran pasangan yang terbentuk adalah: ab, ac, ad, ba, bc, bd, ca, cb, cd, da, db dan dc

Tabel F. 4 Perhitungan Nilai *Support* 4 jenis barang

ab	a	b	hasil	jumlah	status
	1	1	2	7 benar	
	0	1	1		
	0	0	0		
	1	1	2		
	0	0	0		
	0	1	1		
	1	1	2		
	0	1	1		
	0	0	0		
	0	1	1		
	0	1	1		
	1	1	2		
	0	1	1		
	1	1	2		
	0	1	1		
	0	0	0		
	0	1	1		
	1	1	2		
	1	1	2		
	0	0	0		
ac	a	c	hasil	jumlah	status
	1	0	1	2 salah	
	0	0	0		

0	0	0		
1	0	1		
0	1	1		
0	0	0		
1	1	2		
0	0	0		
0	1	1		
0	0	0		
0	0	0		
1	0	1		
0	0	0		
1	0	1		
0	0	0		
0	0	0		
0	0	0		
1	0	1		
1	1	2		
0	1	1		
ad	a	d	hasil	Jumlah status
	1	0	1	1 salah
	0	1	1	
	0	1	1	
	1	0	1	
	0	0	0	
	0	1	1	
	1	0	1	
	0	1	1	
	0	1	1	
	0	1	1	
	0	1	1	
	1	1	2	
	0	1	1	
	1	0	1	
	0	1	1	
	0	1	1	
	1	0	1	
	1	0	1	
	0	0	0	

ba	b	a	hasil	jumlah	status
		1	1	2	7 benar
		1	0	1	
		0	0	0	
		1	1	2	
		0	0	0	
		1	0	1	
		1	1	2	
		1	0	1	
		0	0	0	
		1	0	1	
		1	0	1	
		1	1	2	
		1	0	1	
		1	1	2	
		1	0	1	
		0	0	0	
		1	0	1	
		1	1	2	
		1	1	2	
		0	0	0	
bc	B	c	hasil	jumlah	status
		1	0	1	2 salah
		1	0	1	
		0	0	0	
		1	0	1	
		0	1	1	
		1	0	1	
		1	1	2	
		1	0	1	
		0	1	1	
		1	0	1	
		1	0	1	
		1	0	1	
		1	1	2	
		1	0	1	
		1	0	1	
		1	0	1	
		0	0	0	
		1	0	1	

	1	0	1		
	1	1	2		
	0	1	1		
bd	B	d	hasil	Jumlah	Status
	1	0	1	9	Benar
	1	1	2		
	0	1	1		
	1	0	1		
	0	0	0		
	1	1	2		
	1	0	1		
	1	1	2		
	0	1	1		
	1	1	2		
	1	1	2		
	1	1	2		
	1	0	1		
	1	1	2		
	0	1	1		
	1	1	2		
	1	0	1		
	1	0	1		
	0	0	0		
ca	c	a	Hasil	jumlah	status
	0	1	1	2	salah
	0	0	0		
	0	0	0		
	0	1	1		
	1	0	1		
	0	0	0		
	1	1	2		
	0	0	0		
	1	0	1		
	0	0	0		
	0	0	0		
	0	1	1		
	0	0	0		
	0	1	1		

	0	0	0		
	0	0	0		
	0	0	0		
	0	1	1		
	1	1	2		
	1	0	1		
cb	c	b	Hasil	jumlah	status
	0	1	1	2	salah
	0	1	1		
	0	0	0		
	0	1	1		
	1	0	1		
	0	1	1		
	1	1	2		
	0	1	1		
	1	0	1		
	0	1	1		
	0	1	1		
	0	1	1		
	0	1	1		
	0	0	0		
	0	1	1		
	0	1	1		
	1	1	2		
	1	0	1		
cd	c	d	Hasil	jumlah	status
	0	0	0	1	salah
	0	1	1		
	0	1	1		
	0	0	0		
	1	0	1		
	0	1	1		
	1	0	1		
	0	1	1		
	1	1	2		
	0	1	1		
	0	1	1		

0	1	1			
0	1	1			
0	0	0			
0	1	1			
0	1	1			
0	1	1			
0	0	0			
1	0	1			
1	0	1			
da	d	a	Hasil	jumlah	status
	0	1	1	1	salah
	1	0	1		
	1	0	1		
	0	1	1		
	0	0	0		
	1	0	1		
	0	1	1		
	1	0	1		
	1	0	1		
	1	0	1		
	1	0	1		
	1	1	2		
	1	0	1		
	0	1	1		
	1	0	1		
	1	0	1		
	1	0	1		
	0	1	1		
	0	1	1		
	0	0	0		
db	d	b	Hasil	jumlah	status
	0	1	1	9	benar
	1	1	2		
	1	0	1		
	0	1	1		
	0	0	0		
	1	1	2		
	0	1	1		
	1	1	2		

1	0	1			
1	1	2			
1	1	2			
1	1	2			
1	1	2			
0	1	1			
1	1	2			
1	0	1			
1	1	2			
0	1	1			
0	1	1			
0	0	0			
dc	d	c	Result	countif	status
	0	0	0	1	salah
	1	0	1		
	1	0	1		
	0	0	0		
	0	1	1		
	1	0	1		
	0	1	1		
	1	0	1		
	1	1	2		
	1	0	1		
	1	0	1		
	1	0	1		
	1	0	1		
	0	0	0		
	1	0	1		
	1	0	1		
	1	0	1		
	0	0	0		
	0	1	1		
	0	1	1		
JUMLAH					
BENAR :				4	

Nilai hasil pada perhitungan nilai *support* dengan mencari barang yang dibeli secara bersamaan diatas merupakan nilai penjumlahan dari kode transaksi barang.

Contoh perhitungan hasil pada Tabel F.4 pada transaksi pertama pasangan ab yaitu bernilai 1 karena $1+0 = 1$. Contoh perhitungan hasil pada Tabel F.4 pada transaksi pertama pasangan ab yaitu bernilai 7, karena terdapat 7 transaksi yang bernilai 2. Perhitungan status merupakan perhitungan nilai *support* apakah nilai jumlah memenuhi syarat minimal *support* yaitu bernilai 3, jika bernilai lebih besar sama dengan 3 maka akan berstatus benar dan jika tidak maka akan bernilai salah. Perhitungan jumlah benar pada setiap tabel nilai *support* merupakan patokan apakah dilanjutkan ke proses berikutnya.

Berdasarkan perhitungan pada Tabel F.4 langkah selanjutnya adalah mencari daftar pasangan yang memenuhi nilai minimal *support*. Hasil perhitungan yang memenuhi syarat minimal *support* terdapat pada Tabel F.5

Tabel F.5 Hasil Perhitungan Minimal *Support*

No.	Rules	Jumlah	Status
1	ab	7	Benar
2	ba	7	Benar
3	bd	9	Benar
4	db	9	Benar

Langkah selanjutnya adalah menghitung nilai *confidence*. Nilai *confidence* merupakan presentase jumlah *support* dibagi jumlah *attendance* (jumlah transaksi barang pertama). Perhitungan persentase *confidence* terdapat pada Tabel F.6

Tabel F. 6 Perhitungan Nilai *Confidence*

No.	Rules	Jml transaksi	Jml support	Jml attendance	Persentase confidence
1	Ab	20	7	7	100%
2	Ba	20	7	15	47%
3	Bd	20	9	15	60%
4	Db	20	9	12	75%

Langkah terakhir adalah mencari nilai minimal *confidence*, dimana nilai minimal *confidence* adalah 50%. Pengkodean pada jenis barang dikembalikan kembali dengan nama produk asal. Hasil perhitungan algoritma *Apriori* terdapat pada Tabel F.7

Tabel F. 7 Hasil Perhitungan Algoritma *Apriori*

No.	Rules	Nilai confidence
1	Lux Pink, Citra White	100%
2	Ciptadent Cool, Citra White	60%
3	Citra White, Ciptadent Cool	75%

Lampiran G. Hasil Wawancara

Transkrip Wawancara Pengumpulan Data Dengan Pemilik Minimarkrt Adien Mart
Kalisat

Nama : H. Nanang
Jabatan : Pemilik Adien Mart
Alamat : Jalan Cokroaminoto, Kalisat, Jember, Jawa Timur

Tabel A.11 Hasil Wawancara

	Materi Wawancara
Peneliti	Selamat siang pak, saya mahasiswi Universitas Jember Program Studi Sistem Informasi, nama saya Eky Angliara. Saya akan melakukan penelitian terkait skripsi saya, yaitu berjudul Pengembangan Modul Bundling Pada Sistem Informasi Penjualan Adien Mart Menggunakan Assosiation Rule Algoritma Apriori Sebagai Strategi Promosi Penjualan Barang Kurang laris.
H. Nanang	Oh iya silakan mbak, ada yang bisa dibantu?
Peneliti	Saya mau bertanya tentang kegiatan pejualan disini. Apakah Adien mart memiliki sistem Informasi penjualan?
H. Nanang	Adien mart sudah menggunakan sistem informasi penjualan sederhana yaitu terdiri dari menu input data barang dan

	menu kasir untuk transaksi penjualan
Peneliti	Apakah setiap transaksi yang ynag dilakukan tersimpan disistem, pak?
H. Nanang	Iya tersimpan mbak, sebenarnya kami membutuhkan sistem yang lebih kompleks untuk mendukung kegiatan transaksi penjualan Adien Mart
Peneliti	Menurut Bapak, bagaimana tentang sistem informasi penjualan yang akan saya buat?
H. Nanang	Oh, itu bagus mbak. Bisa membantu. Karena memang salah satu kendala di Adien Mart adalah penumpukan barang-barang yang tingkat penjualannya rendah. Tapi alangkah baiknya selain promosi barang kurang laris, sistem di lengkapi dengan menu pengadaan barang, menu untuk melihat transaksi penjualan karena sistem yang sekarang belum menampilkan detail transaksi secara lengkap, menu untuk menyimpan data supplier, data user.
Peneliti	Mengenai Skripsi yang akan buat, saya membutuhkan data barang dan data transaksi penjualan Adien Mart untuk pencarian barang kurang laris. Nantinya dari data barang yang bapak berikan akan saya kelompokkan sesuai dengan kategori dan subkategori barang. Jadi nanti di sistem kita bisa melihat barang sesuai

	dengan kategori atau subkategori tertentu dengan dapat memudahkan kita mencari barang yang kurang laris untuk direkomendasikan.
H. Nanang	Untuk data yang dibutuhkan, mbak bisa langsung minta ke Admin saja. Untuk tambahan menu yang dapat mendukung sistem saya serahkan ke mbak saja.
Peneliti	Baik pak. Oiya pak, selama ini apa yang dilakukan Adien Mart untuk menangani permasalahan penumpukan barang?
H. Nanang	Selama ini paling ya degan memberikan diskon, kadang ada barang yang sampai kadaluarsa terpaksa dibuang.
Peneliti	Wah, sayang sekali ya pak. Kalau barangnya dibuang. Berarti sama saja rugi pak.
H. Nanang	Iya Mbak, untuk itu kita butuh strategi lain untuk promosi barang-barang yang kurang laris ini. saya dukung pembuatan sistem yang akan mbak buat, nanti kalau sudah jadi akan saya coba terapkan di Adien Mart.
Peneliti	Baik pak. Oiya Adien Mart mempunyai berapa pegawai?
H. Nanang	Disini saya punya 10 pegawai yang terdiri dari 1 admin, 4 kasir dan 5 pramuniaga.
Peneliti	Lumayan banyak juga ya pak

H. Nanang	Iya mbak, Alhamdulillah dari took di kecil di pasar sampai akhirnya bisa buka minimarket seperti sekarang,
Peneliti	Mengapa bapak memilih usaha minimarket?
H. Nanang	Karena berdagang di sunnahkan oleh Rasulullah, selain itu mengikuti perkembangan jaman dengan bertambahnya kebutuhan masyarakat, selain itu di daerah Kalisat ini masih jarang Minimarket serba ada seperti Adien Mart. Jadi saya ambil saja peluang usaha ini.
Peneliti	Ooh jadi begitu pak, untuk minat dari masyarakat sendiri bagaimana pak antusiasnya? Melihat di kalisat ini ada beberapa minimarket seperti indomaret dan alfamart.
H. Nanang	Seperti yang saya bilang tadi mbak, Adien Mart lebih serba ada dan lebih banyak pilihan barang yang di jualan dari segi harga pun masih di bawah indomaret dan alfamart. Terlebih lagi kalau kita adakan diskon insyaallah daya beli masyarakat meningkat.
Peneliti	Tapi ada saja ya pak stok barang menumpuk karna jarang peminat
H.Nanang	Ya banyak sekali mbak, karena satu jenis

	<p>barang kita bisa nyetok beberapa varian dan beberapa merk. Contoh, misalkan di jenis sabun mandi. Saya jual merek giv, shinsui, harmoni, lux, dove dll belum lagi setiap merk ada beberapa variannya misal sabun merk giv, ada yang giv ungu, giv kuning, giv putih dan giv biru. Belum lagi ada sabun padat dan sabun cair.</p>
Peneliti	<p>Karna banyak pilihan barang dengan fungsi sama dan minat masyarakat yang berbeda-beda sehingga mengakibatkan penumpukan beberapa jenis barang yang kurang laris.</p>
H. Nanang	<p>Betul mbak, saya rasa sistem yang mbk akan buat sangat bagus dan bisa membantu untuk permasalahan ini.</p>
Peneliti	<p>InsyaAllah, kita bisa memanfaatkan pola pembelian konsumen untuk acuan sebagai strategi Promosi <i>bundling</i></p>
H. Nanang	<p>Nanti bagaimana kita bisa implementasinya, mbak?</p>
Peneliti	<p>Jadi begini pak, seperti yang saya bilang diawal. Saya membutuhkan data penjualan Adien Mart. Dari data penjualan ini kita bisa olah data tersebut untuk menghasilkan informasi baru yaitu berupa pola penjualan konsumen. Misal dari data penjualan ternyata kita</p>

	<p>menemukan jika konsumen membeli telur maka akan membeli mi kuah indomie rasa soto. Jadi dari pola tersebut kita bisa membuat <i>bundling</i> barang dan mengganti salah satu barang dari pola yang sudah kita temukan dengan barang yang kurang laris yang memiliki subkategori yang sama. Jadi hasil yang akan di rekomendasikan sistem yaitu dengan mem-<i>bundling</i> telur dan mie kuah rasa ayam bawang.</p>
H.Nanang	Kenapa bisa mie kuah rasa ayam bawang?
Peneliti	<p>Jadi begini pak, seperti yang saya jelaskan diawal tadi, dari data barang yang akan bapak berikan saya akan membuat kategori dan subkategori dari setiap barang yang ada. Fungsinya ya pada saat perekomendasi barang kurang laris ini.. hasil dari pola pembelian konsumen di hasilkan jika membeli telur maka membeli mie kuah rasa soto.. disinilah startegi promosi akan dilakukan yaitu dengan mengganti mie kuah rasa soto dengan mie kuah rasa ayam bawang.. mie kuah rasa ayam bawang direkomendasikan karna pada subkategori mie kuah ternyata rasa ayam bawang memiliki tingkat penjualan yang rendah.jadi akan secara otomatis</p>

	sistem akan mencari mie kuah soto termasuk subkategori apa dan akan mencari barang yang tingkat penjualannya rendah di subkategori yang sudah ditemukan.
H. Nanang	Ooo begitu mbak. Jadi kita memanfaatkan kebiasaan pembelian dari konsumennya?
Peneliti	Iya pak, benar sekali
H. Nanang	Iya sudah mbak segera di selesaikan sistemnya.
Peneliti	Baik pak, akan segera saya selesaikan. Terimakasih pak atas informasi dan sudah melungkan waktu untuk membantu Skripsi saya.
H. Nanang	Iya mbak, saya doakan semoga lancer dan cepat selesai skripsinya. Nanti kalau ada yang kurang dan dibutuhkan saya siap membantu.
Peneliti	Baik pak, sekali lagi terimakasih banyak. Saya pamit dulu. Selamat siang pak
H. Nanang	Selamat siang.