



**SISTEM INFORMASI PENENTUAN PAKET PEMBELIAN  
PRODUK MENGGUNAKAN ALGORITMA FREQUENT  
PATTERN-GROWTH PADA KPRI JEMBER.**

**SKRIPSI**

Oleh:

**Mohammad Ivan Noorkholid  
152410101043**

**PROGRAM STUDI SISTEM INFORMASI  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS JEMBER  
2019**



**SISTEM INFORMASI PENENTUAN PAKET PEMBELIAN  
PRODUK MENGGUNAKAN ALGORITMA FREQUENT  
PATTERN-GROWTH PADA KPRI JEMBER.**

**SKRIPSI**

Diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat  
untuk menyelesaikan Program Studi Sistem Informasi (S1)  
dan mencapai gelar Sarjana

Oleh:

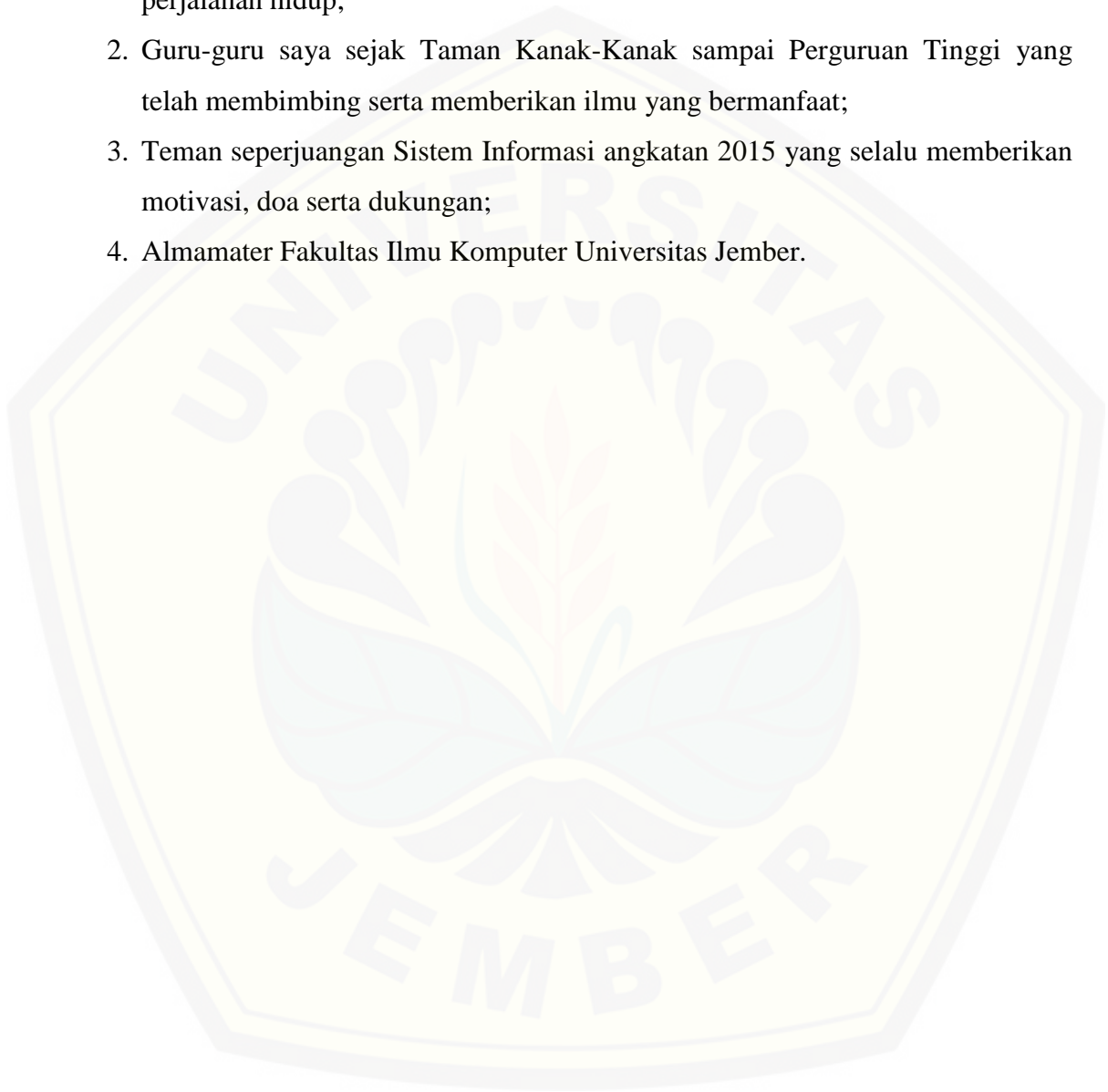
**Mohammad Ivan Noorkholid  
152410101043**

**PROGRAM STUDI SISTEM INFORMASI  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS JEMBER  
2019**

**PERSEMBAHAN**

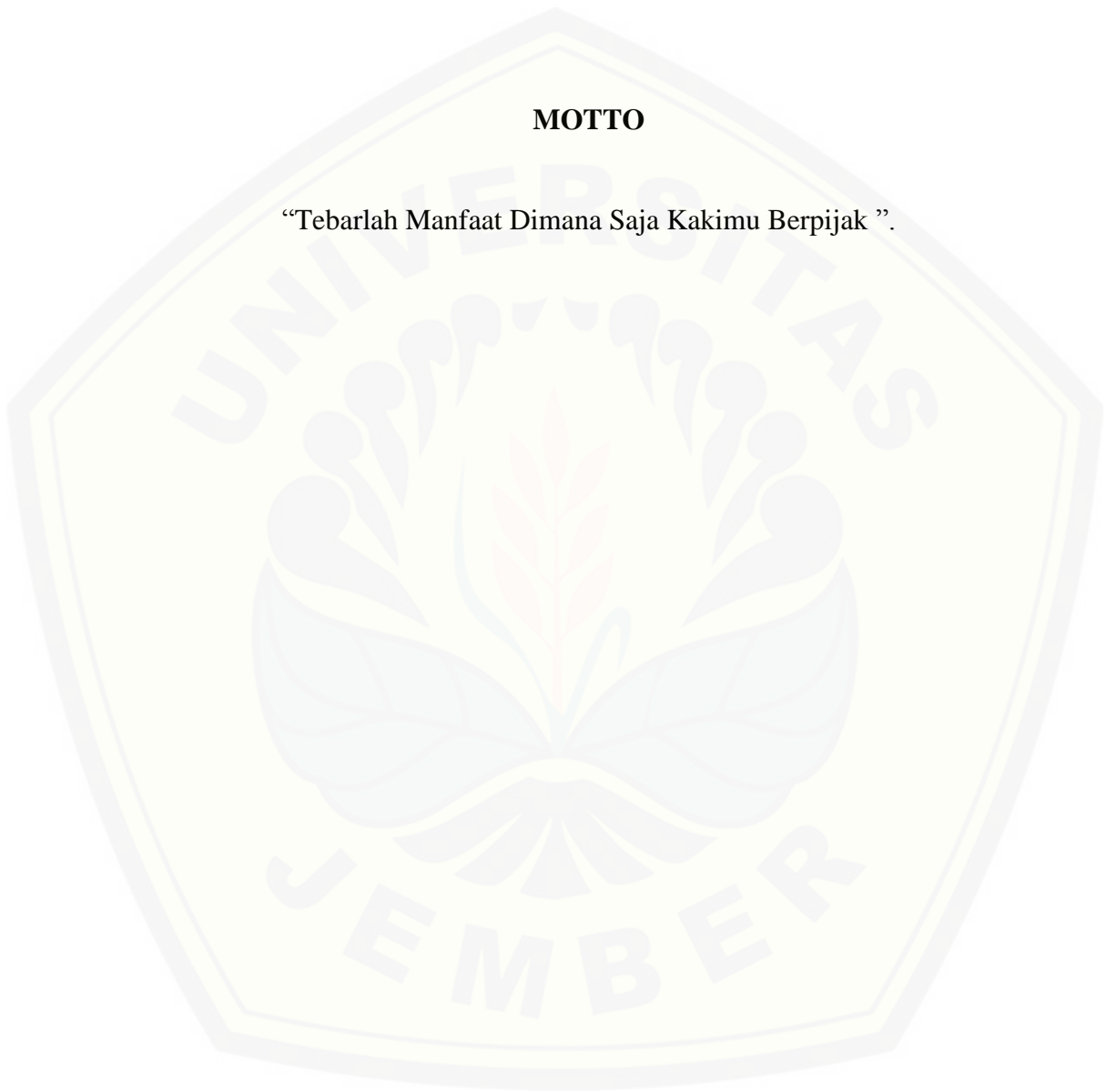
Skripsi ini saya persembahkan untuk:

1. Kedua orang tua saya Burhan Noordin dan Ibu Saidah serta adik adik saya yang senantiasa memberikan doa, motivasi dan dukungan dalam setiap langkah perjalanan hidup;
2. Guru-guru saya sejak Taman Kanak-Kanak sampai Perguruan Tinggi yang telah membimbing serta memberikan ilmu yang bermanfaat;
3. Teman seperjuangan Sistem Informasi angkatan 2015 yang selalu memberikan motivasi, doa serta dukungan;
4. Almamater Fakultas Ilmu Komputer Universitas Jember.



**MOTTO**

“Tebarlah Manfaat Dimana Saja Kakimu Berpijak ”.



**PERNYATAAN**

Saya yang bertanda tangan di bawah ini:

Nama : Mohammad Ivan Noorkholid

NIM : 152410101043

Menyatakan dengan sesungguhnya bahwa skripsi yang berjudul “Sistem Informasi Penentuan Paket Pembelian Produk Menggunakan Algoritma Frequent Pattern-Growth Pada Kpri Jember.” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi manapun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan 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 ternyata di kemudian hari pernyataan ini tidak benar.

Jember,

Yang menyatakan,

Mohammad Ivan Noorkholid

NIM. 152410101043

**PENGESAHAN PEMBIMBING**

Skripsi berjudul **“Sistem Informasi Penentuan Paket Pembelian Produk Menggunakan Algoritma Frequent Pattern-Growth Pada Kpri Jember”**, telah diuji dan disahkan pada:

Hari, tanggal : ...Juli 2019

Tempat : Fakultas Ilmu Komputer Universitas Jember

Disetujui oleh:

Pembimbing I,

Pembimbing II,

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

Gama Wisnu Fajarianto, S.kom., M.Kom.  
NIP. 1760015717

**SKRIPSI**

**SISTEM INFORMASI PENENTUAN PAKET PEMBELIAN  
PRODUK MENGGUNAKAN ALGORITMA FREQUENT  
PATTERN-GROWTH PADA KPRI JEMBER**

Oleh

**Mohammad Ivan Noorkholid**

**152410101043**

Pembimbing

Dosen Pembimbing Utama : Muhammad Arief Hidayat, S.Kom., M.Kom.

Dosen Pembimbing Anggota : Gama Wisnu Fajarianto, S.kom., M.Kom.

**PENGESAHAN**

Skripsi berjudul “**Sistem Informasi Penentuan Paket Pembelian Produk Menggunakan Algoritma Frequent Pattern-Growth Pada Kpri Jember**”, telah diuji dan disahkan pada:

Hari, tanggal : Senin, 29 Juli 2019

Tempat : Fakultas Ilmu Komputer Universitas Jember

Tim penguji

Penguji I,

Penguji II,

Drs.Antonius Cahya P,M. App,.Sc., Ph.D  
NIP. 196909281993021001

Priza Pandunata S.kom., M.Sc.  
NIP. 198301312015041001

Mengesahkan  
a.n Dekan  
Wakil Dekan I Fakultas Ilmu Komputer

Drs.Antonius Cahya P,M. App,.Sc., Ph.D  
NIP. 196909281993021001



## RINGKASAN

**Sistem Informasi Penentuan Paket Pembelian Produk Menggunakan Algoritma Frequent Pattern-Growth Pada Kpri Jember;** Mohammad Ivan Noorkholid, 152410101043; 2019: 75 halaman; Program Studi Sistem Informasi Fakultas Ilmu Komputer Universitas Jember.

Perekonomian saat ini tengah berkembang sangat pesat. Persaingan dalam dunia bisnis mulai mengarah pada persaingan bebas yang semakin kompleks. Untuk mencapai hasil yang maksimal, konsumen harus ditempatkan pada posisi teratas, strategi pemasaran perlu diatur untuk mendapatkan hasil yang lebih efektif dan efisien. Penelitian ini bertujuan untuk merancang dan membangun sistem informasi penentuan paket pembelian produk di sebuah kopraasi. Sistem ini menggunakan algoritma *Fp-Growth (Frequent Pattern growth)* untuk menghasilkan informasi tentang paket pembelian produk dengan menangkap fenomena yang terjadi dalam transaksi penjualan. Implementasi algoritma *Frequent Pattern Growth* menggunakan PHP. Data hasil perhitungan algoritma *Frequent Pattern Growth* divisualisasikan dalam halaman website. Penerapan Algoritma *Frequent Pattern Growth* didukung dengan metode *associtation rules* untuk menghasilkan data yang lebih lengkap dan akurat. Prhitungan membutuhkan data masukan berupa *minimum support* dan *minimum confidence* untuk memproses data transaksi penjualan menjadi paket pembelian produk. Dengan memasukkan *minimum support* 1 dan *minimum confidence* sebesar 50. Hasil perhitungan didapatkan bahwa paket pembelian produk yang muncul sebanyak 10 paket dengan *minimum support* yang bermacam-macam berurutan dari produk pertama sampai produk terakhir. semakin besar *minimum support* dan *minimum confidence* yang dimasukkan maka semakin sedikit paket pembelian produk yang dihasilkan.

## PRAKATA

Puji syukur kehadiran Allah SWT, atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Sistem Informasi Penentuan Paket Pembelian Produk Menggunakan Algoritma Frequent Pattern-Growth Pada Kpri Jember”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata satu (S1) pada Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Jember.

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

1. Prof. Saiful Bukhori, ST., M.Kom. selaku Dekan Fakultas Sistem Informasi Universitas Jember yang telah memberikan fasilitas sehingga skripsi ini dapat selesai;
2. Fahrobby Adnan, S.Kom., M.S.S.I selaku dosen pembimbing akademik yang telah memberikan fasilitas dan motivasi sehingga penulis dapat menyelesaikan skripsi ini;
3. Muhammad Arief Hidayat, S.Kom., M.Kom., selaku Dosen Pembimbing Utama yang telah memberikan bimbingan dan motivasi dalam segala aspek serta meluangkan waktunya kepada penulis sehingga dapat menyelesaikan skripsi ini;
4. Gama Wisnu Fajarianto, S.kom., M.Kom., selaku Dosen Pembimbing Anggota yang telah memberikan bimbingan dan motivasi dalam segala aspek serta meluangkan waktunya kepada penulis sehingga dapat menyelesaikan skripsi ini
5. Drs.Antonius Cahya Prihandoko, M.App.Sc., Ph.D., selaku dosen penguji utama yang telah meluangkan waktu untuk memberikan saran dan kritik yang membangun demi kesempurnaan skripsi ini;
6. Priza Pandunata S.kom., M.Sc. selaku dosen penguji anggota yang telah meluangkan waktu untuk memberikan saran dan kritik yang membangun demi kesempurnaan skripsi ini;

7. Keluarga Sistem Informasi angkatan 15 yang selalu memberikan motivasi dan semangat;
8. Semua pihak yang tidak dapat disebutkan satu-persatu yang telah memberikan kontribusi dan bantuannya demi kelancaran pengerjaan skripsi ini.

Penulis juga menerima segala kritik dan saran dari semua pihak demi kesempurnaan skripsi ini. Akhir kata, penulis berharap semoga skripsi ini dapat bermanfaat. Aamiin.

Jember, Juli 2019

Penulis



## DAFTAR ISI

### Daftar Isi

DAFTAR ISI.....	xi
DAFTAR TABEL .....	xiii
DAFTAR GAMBAR.....	xiv
BAB 1. PENDAHULUAN .....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	4
1.3 Tujuan Penelitian .....	4
1.4 Manfaat Penelitian.....	4
1.5 Batasan Masalah .....	5
BAB 2. TINJAUAN PUSTAKA .....	6
2.1 Data Mining.....	6
2.2 Market Basket Analysis.....	6
2.3 Association Rule.....	7
2.4 FP-Growth .....	7
2.4.1 Analisa Pola Frekuensi Tinggi Dengan Algoritma FP-Growth .....	7
2.4.2 Pembuatan FP-Tree.....	8
2.4.3 Penerapan Algoritma FP-Growth.....	9
2.4.4 Pembentukan Aturan Asosiasi .....	11
BAB 3. METODE PENELITIAN.....	12
3.1 Jenis Penelitian.....	12
3.3 Pengumpulan Data.....	12
3.4 Perancangan Sistem.....	13
3.4.1 Analisa Kebutuhan .....	14
3.4.2 Desain Sistem .....	14
3.4.3 Implementasi Metode.....	15
3.4.4 Pengujian Sistem.....	15
3.4.5 Maintenance .....	16
BAB 4. PERANCANGAN SISTEM.....	17
4.1 Deskripsi Sistem .....	17
4.2 Analisa Kebutuhan Sistem .....	17
4.2.1 Kebutuhan Fungsional .....	17

4.2.2	Perangkat Keras .....	18
4.2.3	Perangkat Lunak.....	18
4.3	Desain Sistem.....	19
4.3.1	<i>Business process</i> .....	19
4.3.2	<i>Usecase Diagram</i> .....	19
4.3.3	Skenario .....	22
4.3.4	<i>Sequence Diagram</i> .....	28
4.3.5	<i>Activity Diagram</i> .....	35
4.3.6	<i>Class Diagram</i> .....	39
4.3.7	<i>Entity Relationship Diagram</i> .....	39
4.4	Implementasi .....	39
4.5	Pengujian Program.....	39
4.5.1	Pengujian <i>White Box</i> .....	39
4.5.2	Pengujian <i>Black Box</i> .....	44
BAB 5 HASIL DAN PEMBAHASAN .....		45
5.1	Sistem Informasi Penentuan Paket Pembelian Produk .....	45
5.2	Hasil Implementasi Algoritma Fp-Growth Pada Sistem .....	52
5.3	Pembahasan.....	53
BAB 6 PENUTUP .....		59
6.1	Kesimpulan.....	59
6.2	Saran .....	60
DAFTAR PUSTAKA .....		61
LAMPIRAN .....		63
LAMPIRAN 1 (Skenario Sistem) .....		63
LAMPIRAN 2 (Sequence Diagram) .....		71
LAMPIRAN 3 (Activity Diagram) .....		80
LAMPIRAN 4 (Black Box) .....		87
LAMPIRAN 5 (Perhitungan Manual).....		89

**DAFTAR TABEL**

Tabel 4.1 Kebutuhan Fungsional .....	18
Tabel 4.2 Definisi Aktor .....	22
Tabel 4.3 Definisi Usecase.....	22
Tabel 4.4 Skenario Melihat Data Pegawai .....	23
Tabel 4.5 Skenario Menambah Data Pegawai .....	24
Tabel 4.6 Mengubah Data Pegawai .....	25
Tabel 4.7 Test Case __construct .....	44
Tabel 4.8 Test Case index .....	44
Tabel 4.9 Test Case add .....	44
Tabel 4.10 Test Case buy .....	44
Tabel 4.11 Test Case Function checkOut .....	45
Tabel 5.1 Hasil Perhitungan Manual.....	55



## DAFTAR GAMBAR

Gambar 2.1 Konstruksi Fp-Tree .....	10
Gambar 2.2 Frequent Itemset .....	10
Gambar 3.1 Alur Penelitian.....	12
Gambar 3.2 SDLC Waterfall .....	13
Gambar 4.1 Bussines Process .....	19
Gambar 4.2 Usecase Diagram.....	21
Gambar 4.3 Sequence Diagram Melihat Data Pegawai .....	29
Gambar 4.4 Sequence Diagram Menambah Data Pegawai .....	30
Gambar 4.5 Sequence Diagram Mengubah Data Pegawai .....	31
Gambar 4.6 Activity Diagram Melihat Data Pegawai .....	34
Gambar 4.7 Activity Diagram Mengelola Data Pegawai.....	35
Gambar 4.8 Class Diagram .....	38
Gambar 4.9 Entity Relationship Diagram.....	39
Gambar 4.10 Listing Program Function Test Case __construct .....	40
Gambar 4.11 Listing Program Function Test Case index .....	40
Gambar 4.12 Listing Program Function Test Case add .....	40
Gambar 4.13 Listing Program Function Test Case buy.....	40
Gambar 4.14 Listing Program Function checkOut .....	41
Gambar 4.15 Diagram Alir Function Test Case __construct.....	41
Gambar 4.16 Diagram Alir Function Test Case index.....	42
Gambar 4.17 Diagram Alir Function Test Case add.....	42
Gambar 4.18 Diagram Alir Function Test Case buy.....	42
Gambar 4.19 Diagram Alir Function checkOut.....	43
Gambar 5.1 Tampilan Halaman Utama Sistem (pegawai).....	46

Gambar 5.2 Tampilan Halaman Utama Sistem (manajer).....	47
Gambar 5.3 Halaman Mengelola Data Pegawai .....	47
Gambar 5.4 Form Update Data Pegawai.....	48
Gambar 5.5 Form Tambah Data Pegawai .....	48
Gambar 5.6 Halaman Data Transaksi .....	49
Gambar 5.7 Halaman Mengelola Data Produk .....	49
Gambar 5.8 Form Update Data Produk.....	50
Gambar 5.9 Form Tambah Data Produk .....	50
Gambar 5.10 Halaman Mengelola Data Supplier .....	50
Gambar 5.11 Form Update Data Supplier.....	51
Gambar 5.12 Form Tambah Data Supplier .....	51
Gambar 5.13 Halaman Mengelola Data Kategori.....	52
Gambar 5.14 Form Update Data Kategori .....	52
Gambar 5.15 Form Tambah Data Kategori.....	52
Gambar 5.16 Tampilan Fitur Analisa Penjualan.....	53
Gambar 5.17 Tampilan Menambahkan Paket Pembelian Produk .....	54
Gambar 5.18 Tampilan Form Penjualan Prduk.....	54
Gambar 5.19 Paket Pembelian Produk.....	55
Gambar 5.20 Perhitungan Oleh Sistem.....	56
Gambar 5.21 Hasil Percobaan 1 .....	57
Gambar 5.22 Hasil Percobaan 2 .....	57
Gambar 5.23 Hasil Percobaan 3 .....	57
Gambar 5.24 Hasil Percobaan 4 .....	57
Gambar 5.25 Hasil Percobaan 5 .....	57



## BAB 1. PENDAHULUAN

### 1.1 Latar Belakang

Koperasi merupakan suatu bentuk perekonomian sebagai usaha bersama berdasarkan atas kekeluargaan. Karena Koprasi yang menyatakan kerjasama antara mereka yang berusaha sebagai suatu keluarga (Harsoyo, 2006). Menurut UU No 25 Tahun 1992, koperasi adalah badan usaha yang beranggotakan orang-orang atau badan hukum koperasi, dengan melandaskan kegiataannya berdasarkan prinsip koperasi sekaligus sebagai gerakan ekonomi rakyat yang berdasar atas asas kekeluargaan. Dari Penjelasan diatas dapat disimpulkan, bahwa koperasi adalah suatu perkumpulan badan hukum atau orang-orang yang memiliki tujuan untuk kesejahteraan bersama dengan menjunjung tinggi asas kekeluargaan yang saling tolong menolong dan gotong royong antar anggota koperasi. KPRI (Koperasi Pegawai Republik Indonesia) merupakan salah satu bentuk dari koperasi yang ada di Indonesia. Jember Memiliki beberapa KPRI, salah satunya adalah KPRI Universitas Jember.

KPRI Universitas Jember memiliki beberapa usaha, salah satunya ialah unit pertokoan. Unit pertokoan ini memiliki beragam jenis produk dan juga memiliki harga yang bersaing. KPRI Universitas Jember ini memiliki banyak konsumen yang beragam. Mulai dari Masyarakat Sekitar Unej sampai mahasiswa dari Unej itu Sendiri. Beberapa mahasiswa lebih memilih berbelanja di KPRI Universitas Jember dengan alasan KPRI Universitas Jember Memiliki produk yang lengkap dan juga harganya tidak kalah saing dengan minimarket pada umumnya (Sugiyanto, 2015).

Saat ini perekonomian tengah berkembang dengan pesatnya. Persaingan dalam dunia bisnis mulai mengarah pada persaingan bebas yang semakin kompleks. Swalayan dan mini market semakin tidak terbendung kemunculannya. Untuk mencapai hasil yang maksimal, konsumen harus ditempatkan pada posisi teratas, strategi pemasaran perlu diatur untuk mendapatkan hasil yang lebih efektif dan efisien.

Dalam menjalankan proses bisnisnya, KPRI Universitas Jember Menggunakan sebuah sistem penjualan. Sistem penjualan ini mengelola semua transaksi penjualan yang terjadi. Data transaksi yang tersimpan dalam *database* dapat digunakan untuk mengetahui laba dan rugi penjualan. *Database* konsumen yang tepat sasaran dan lengkap dapat digunakan sebagai senjata yang sangat ampuh bagi pelaku usaha. Karena *database* pelanggan dapat digunakan sebagai penunjuk arah penjualan, sebagai gambaran potensi pasar, untuk menganalisa penjualan, dan juga sebagai sarana komunikasi dengan konsumen (Warni, 2016). Salah satu strategi pemasaran yang dapat diterapkan dengan menggunakan *database* tersebut adalah strategi *cross-selling* produk untuk menentukan paket pembelian produk.

*Cross-selling* merupakan teknik pemasaran yang dapat memikat konsumen untuk melengkapi produk awal yang telah dibeli dengan produk yang dapat melengkapinya. *Cross-selling* akan berhasil disaat konsumen mendapatkan nilai tambah. Barang yang ditawarkan tersebut harusnya memiliki nilai lebih daripada barang asli yang dibeli secara terpisah. Menurut Ramaswarni *cross-selling* adalah salah satu seni dalam berjualan yang sangat menarik. Karena dengan teknik ini, dapat memungkinkan terjadinya penjualan kembali setelah terjadi penjualan pertama, karena adanya keterkaitan kebutuhan (Kamakura, Ramaswami, & R. K., 1991.).

Untuk mengembangkan strategi pemasaran pada KPRI Universitas Jember, maka dirancang sebuah sistem untuk menentukan paket pembelian produk yang merupakan salah satu strategi pemasaran produk. *Database* yang ada dalam sistem penjualan dapat digunakan sebagai data untuk menentukan paket pembelian produk untuk lebih memikat konsumen membeli barang lebih dari biasanya. Dengan diterapkannya strategi pemasaran ini diharapkan omset penjualan KPRI Universitas Jember juga semakin meningkat.

Dalam Penelitian ini penulis akan mencoba untuk melakukan suatu percobaan terhadap data penjualan produk KPRI Universitas Jember dengan menggunakan metode *market basket analysis* untuk melihat korelasi antara produk satu dengan yang lain. Algoritma yang akan digunakan adalah *Frequent Pattern Growth (FP-Growth)*. *FP-Growth* adalah salah satu dari beberapa algoritma yang dapat digunakan untuk menentukan data yang paling sering muncul (*frequent itemset*)

dalam kumpulan data. *Frequent itemset* tersebut nantinya akan dijadikan sebagai acuan untuk merumuskan aturan-aturan asosiasi yang dihasilkan oleh model *market basket analysis*. Aturan-aturan asosiasi yang dihasilkan akan dianalisa dan di evaluasi untuk menemukan algoritma yang paling baik dan sesuai (*best-fit algorithm*) untuk permasalahan dalam penelitian ini. Menurut Bayu Adhi dalam penelitiannya yang membahas tentang penerapan strategi penjualan menggunakan *association rules* mengemukakan suatu hasil bahwa metode *association rules* berdampak langsung pada peningkatan *revenue* perusahaan. Data mining menjadi konsep yang berperan sangat penting untuk mengekstrak data transaksi menjadi informasi yang berguna yang didapatkan dari mengimplementasikan metode *association rules* (Tama, 2010).

Dinna Yunika (2018) melakukan sebuah penelitian yang membahas tentang pemakaian algoritma *Frequent Pattern Growth (FP-Growth)* dalam memberikan rekomendasi buku. Penelitian ini bertujuan untuk mengembangkan sistem informasi perpustakaan yang dilengkapi dengan fitur rekomendasi buku untuk memberikan referensi buku yang lain bagi anggota yang akan meminjam buku. Buku yang direkomendasikan merupakan buku yang memiliki kesamaan topic yang dibahas dengan buku yang dicari oleh anggota. Pengimplementasian algoritma *FP-Growth* ini pada sistem informasi perpustakaan terbukti berhasil dengan tampilnya rekomendasi buku sebanyak lima buah pada halaman pencarian. Rekomendasi ini berdasarkan hasil pencarian item set menggunakan *FP-Tree* dan hasil pencarian pola (*rule*) dengan mengetahui nilai *support* dan *confident* dari sebuah rule. Sedangkan pada penelitian Naufal Farras (2016) yang membahas tentang pengimplementasian algoritma *FP-Growth* pada analisa tingkat kelulusan mahasiswa Politeknik Negeri Malang membuktikan bahwa algoritma *FP-Growth* dapat digunakan untuk memberikan informasi tingkat kelulusan berdasarkan beberapa variable, serta dapat mengetahui tingkat kecocokan pekerjaan berdasarkan beberapa variable tertentu.

Berdasarkan Uraian diatas, dengan penelitian ini, diharapkan dapat memberi hasil berupa informasi yang sangat bermanfaat bagi *stakeholder*, terutama yang berkaitan dengan strategi pemasaran dan penjualan, khususnya pada KPRI Universitas Jember.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, maka dapat dibuat rumusan masalah sebagai berikut:

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

## 1.3 Tujuan Penelitian

Adapun yang menjadi tujuan penulisan penelitian ini adalah sebagai berikut:

1. Dapat menerapkan metode *FP-Growth* untuk mengetahui kombinasi produk yang sering dibeli oleh konsumen dan membentuk rekomendasi paket pembelian produk dari data penjualan yang diperoleh.
2. Dapat merancang dan membangun sistem informasi penentuan paket pembelian produk pada KPRI Universitas Jember menggunakan metode *FP-Growth*.

## 1.4 Manfaat Penelitian

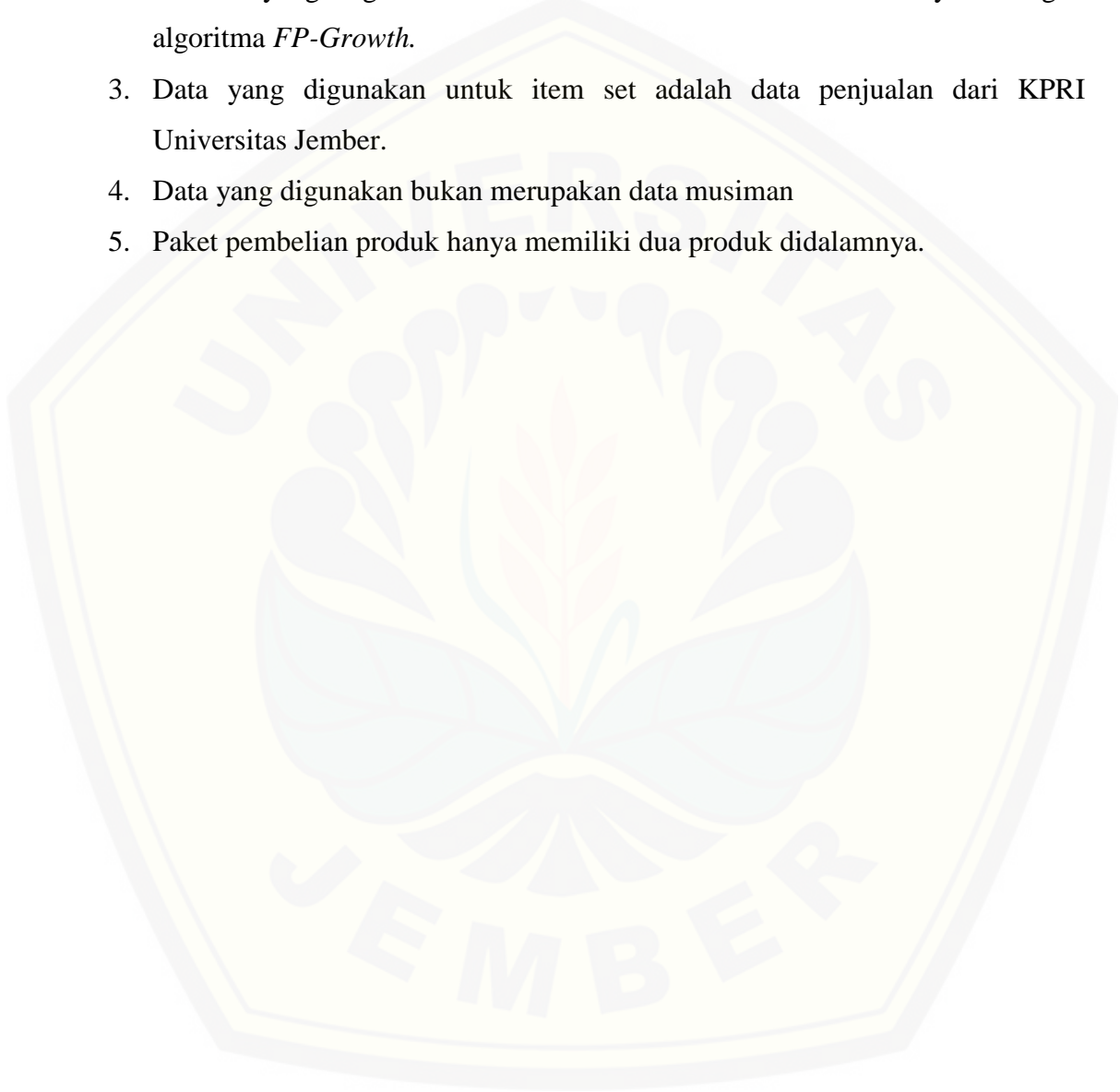
Manfaat dari penelitian yang dapat diperoleh antara lain :

1. Diharapkan dapat dijadikan sebagai referensi untuk merancang strategi pemasaran yang lebih tepat sasaran
2. Meningkatkan keilmuan tentang *Datamining* sebagai media bagi penyelesaian Tugas Akhir untuk jenjang S1 pada Program Studi Sistem Informasi Universitas Jember

### 1.5 Batasan Masalah

Agar dalam penelitian ini dapat mencapai sasaran dan tujuan yang diharapkan, maka permasalahan yang ada hanya dibatasi pada:

1. Perancangan dan pembangunan sistem ini digunakan untuk membuat sebuah paket pembelian produk di KPRI Universitas Jember.
2. Metode yang digunakan adalah metode market basket analysis dengan algoritma *FP-Growth*.
3. Data yang digunakan untuk item set adalah data penjualan dari KPRI Universitas Jember.
4. Data yang digunakan bukan merupakan data musiman
5. Paket pembelian produk hanya memiliki dua produk didalamnya.





## BAB 2. TINJAUAN PUSTAKA

### 2.1 Data Mining

Data mining adalah proses pengolahan data menggunakan teknik statistik, matematika, kecerdasan buatan, dan machine learning untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai *database* besar (Turban, 2005).

Data mining merupakan suatu proses pendukung pengambil keputusan dimana kita mencari pola informasi dalam data. Pencarian ini dapat dilakukan oleh pengguna, misalnya dengan menggunakan *query* atau dapat dibantu dengan suatu sistem yang secara otomatis mencari pola informasi pada basis data. Pencarian ini disebut *discovery*. Data mining dapat dibagi menjadi beberapa kelompok berdasarkan tugas yang dapat dilakukan, antara lain : deskripsi, estimasi, klasifikasi, prediksi, pengklusteran, dan asosiasi (Syaifullah, 2010).

### 2.2 Market Basket Analysis

*Market basket analysis* adalah metodologi yang digunakan untuk melakukan analisa *buying habit* atau kebiasaan membeli konsumen dengan menemukan asosiasi antara beberapa item yang berbeda, yang dibeli secara bersamaan atau dalam keranjang belanja yang sama (Gunaidi & Sensue, 2012).

*Market basket analysis* menggunakan teknik matematis yang dipakai secara umum untuk mengungkap persamaan atau hubungan keterkaitan antara produk satu dengan yang lain. *Market basket analysis* menentukan produk apa yang dibeli bersamaan. Informasi yang didapat setelah mengetahui ketika satu produk akan dibeli bersamaan dengan produk lainnya dapat digunakan untuk memasarkan dan menjadikan pembeli produk tertentu sebagai target prospek untuk produk lainnya. Hal ini akan meningkatkan tingkat efektivitas taktik penjualan dan pemasaran dengan menggunakan data konsumen yang sudah tersedia (Marakas , 20003) .

Umumnya *market basket analysis* dimanfaatkan sebagai titik awal untuk mencari strategi pemasaran yang efektif dengan cara mengetahui pola spesifik yang terlihat dari data transaksi yang sudah ada. *Market basket analysis*

menghasilkan suatu aturan asosiasi (*association rules*). *Association rules* adalah pola-pola korelasi data dalam basis data.

### 2.3 Association Rule

*Association rule* merupakan teknik data mining yang menjadi dasar dari berbagai metode data mining lainnya. *Association rule* digunakan untuk menentukan korelasi antar item dalam suatu kumpulan data (*dataset*) yang telah ditentukan. Pentingnya suatu aturan asosiatif dapat diketahui dengan dua parameter yaitu *support* (nilai penunjang) dan *confidence* (nilai kepastian). *Support* adalah prosentase kombinasi item tersebut dalam *database* sedangkan *confidence* adalah kuatnya hubungan antar item dalam aturan asosiatif. Analisis asosiasi didefinisikan sebagai suatu proses untuk menemukan semua suatu aturan asosiatif yang memenuhi syarat minimum untuk *support* (*minimum support*) dan juga syarat minimum untuk *confidence* (*minimum confidence*) (Pramudiono, 2003).

### 2.4 FP-Growth

*FP-Growth* merupakan salah satu algoritma data mining yang dapat digunakan untuk menentukan himpunan data mana saja yang paling sering muncul. Karakteristik algoritma *FP-Growth* adalah penjabaran struktur data menggunakan *tree* yang disebut *FP-Tree*. Dengan menggunakan *FP-Tree*, data dapat langsung diekstrak oleh *FP-Growth* untuk mendapatkan *frequent itemset* dari *FP-Tree*. Algoritma Frequent Pattern-Growth atau yang biasa disebut dengan *FP-Growth* merupakan pengembangan dari algoritma Apriori, sehingga dalam algoritma *FP-Growth* segala kekurangan dalam algoritma Apriori telah diperbaiki (Erwin, 2009).

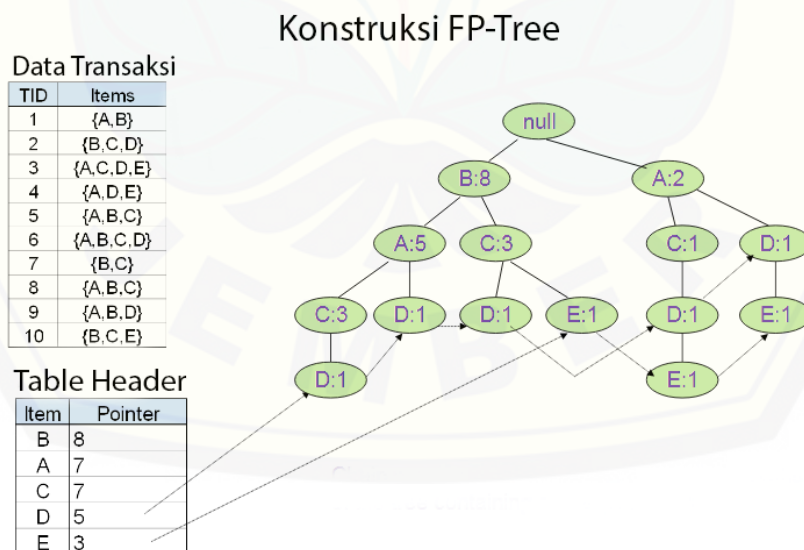
#### 2.4.1 Analisa Pola Frekuensi Tinggi Dengan Algoritma FP-Growth

*Frequent itemset* dapat ditentukan dengan melakukan 2 proses yang harus dilakukan yaitu pembuatan *FP-Tree* dan penerapan *FP-Growth* untuk menemukan *frequent itemset*. Algoritma *FP-Growth* dapat mencari *frequent itemset* dengan memperluas penggunaan dari *FP-Tree*.

### 2.4.2 Pembuatan FP-Tree

*FP-Tree* merupakan struktur penyajian data yang dimampatkan. *FP-Tree* dibangun dengan memetakan setiap data transaksi menjadi lintasan data tertentu dalam *FP-Tree*. Dalam setiap transaksi, jika ada transaksi yang memiliki item yang sama, maka lintasan data dapat saling menimpa. Semakin banyak item yang sama dalam suatu transaksi, maka lintasan *FP-Tree* yang terbentuk semakin efektif (Gunaidi & Sensue, 2012). Adapun definisi dari pohon *FP-Tree* sebagai berikut :

- FP-Tree* dibentuk oleh akar yang berlabel *null*, sekumpulan *sub-tree* yang berisi beberapa item tertentu yang sudah diurutkan berdasarkan *frekuensi* munculnya item tersebut, dan sebuah tabel *frequent header*.
- Dalam setiap simpul *FP-Tree* berisi tiga informasi penting, yaitu lebel item yang berisi informasi jenis item yang ada dalam simpul tersebut, *support count* yang berisi jumlah lintasan transaksi yang melalui simpul tersebut, dan *pointer* yang berfungsi sebagai penghubung antara simpul-simpul dengan label item sama antar lintasan, biasanya ditandai dengan garis panah putus-putus.



Gambar 2.1 Konstruksi Fp - Tree



Berdasarkan gambar 1, menjelaskan bahwa terdapat dua tabel dimana tabel data transaksi berisi tentang nama nama item yang terdapat dalam konstruksi *FP-Tree* sedangkan tabel *header* menjelaskan tentang berapa kali item tersebut muncul dalam tabel data transaksi, sedangkan dalam konstruksi *FP-Tree* itu sendiri terdapat nama item dan juga frekuensi titik tersebut dilalui.

### 2.4.3 Penerapan Algoritma FP-Growth

Setelah *FP-Tree* telah dibentuk, maka algoritma *FP-Growth* akan diterapkan untuk mencari *frequent itemset* yang signifikan. Algoritma *FP-Growth* memiliki tiga langkah utama, yaitu :

a. Pembangkitan *Conditional Pattern Base*

*Conditional Pattern Base* merupakan *sub-database* yang berisikan lintasan. *Conditional Pattern Base* yang dibangkitkan didapatkan melalui *FP-Tree* yang dibangun sebelumnya. Pada gambar 2.2 terdapat kolom item yang berisikan data semua produk yang muncul pada tabel data transaksi dengan diurutkan berdasarkan frekuensi munculnya barang. Kolom *patternbase* merupakan kolom yang berisi data produk yang muncul bersamaan dengan produk yang berada pada tabel item. Pada saat pembeli melakukan pembelian item E, B dan C secara bersamaan, pengdeskripsian pada tabel *patternbase* dapat digambarkan seperti pada gambar 2.2 dimana item E sebagai item yang memiliki *frekuensi* pembelian terbanyak diletakkan pada kolom item dengan diikuti oleh penulisan item pendamping pada kolom *patternbase*.

item	Patternbase				
E	(B,C : 1)	(A,C,D : 1)	(A,D : 1)		
D	(B,A,C : 1)	(B,A : 1)	(B,C : 1)	(A,C : 1)	(A : 1)
C	(B,A : 2)	(B : 1)	(A : 1)		
A	(B : 1)				
B	-				

Gambar 2.2 *Pattern base*

b. Pembangkitan *Conditional FP-Tree*

Tahap ini menjumlahkan *support count* dari setiap *item* pada setiap *conditional pattern base*. setelah penjumlahan selesai dilakukan, setiap *item*

yang memiliki *support count* lebih besar dari atau sama dengan *minimum support count* akan dibangkitkan dengan *conditional FP-Tree*. Data patternbase yang terdapat pada gambar 2.2 dipecah menjadi satu produk sendiri dengan penambahan frekuensi munculnya barang tersebut. Item B dan C pada gambar 2.2 yang terdapat pada satu sell dipecah menjadi dua data yang berbeda dengan frekuensi sendiri-sendiri sesuai dengan berapa banyak data tersebut muncul. Penggambaran *conditional fp-tree* dapat dilihat pada gambar 2.3

item	Conditional Fp Tree			
E	B : 1	A : 2	C : 2	D : 2
D	B : 3	A : 4	C : 4	
C	B : 3	A : 3		
A	B : 1			
B				

Gambar 2.3 *Conditional FP-Tree*

c. Pencarian *frequent itemset*

Jika *Conditional FP-Tree* merupakan lintasan tunggal (*single path*), maka *frequent item set* akan didapatkan dengan mengkombinasikan *item* untuk setiap *conditional FP-Tree*. Jika tidak merupakan lintasan tunggal, maka harus membangkitkan jalur *FP-Tree* yang lain. Data *conditional fp-tree* yang telah terbentuk kemudian di gabungkan dengan produk yang ada pada kolom item untuk menghasilkan *frequent itemset*.

item	Frequent Itemset			
E	(B,E : 1)	(A,E : 2)	(C,E : 2)	(D ,E: 2)
D	(B,E : 3)	(A,E : 4)	(C,E : 4)	
C	(B,E : 3)	(A,E : 3)		
A	(B,E : 1)			
B				

Gambar 2.4 *Frequent Itemset*

#### 2.4.4 Pembentukan Aturan Asosiasi

Setelah menemukan pola frekuensi tinggi, maka langkah selanjutnya adalah mencari aturan asosiasi yang memenuhi syarat yang telah ditentukan. Untuk mendapatkan nilai *support* item A dapat diperoleh dari rumus sebagai berikut :

$$S = \frac{\Sigma(\text{Transaksi item A})}{\Sigma(\text{Transaksi})} \quad (1)$$

Sementara itu untuk mencari nilai *support* dari 2 item dapat diperoleh dari rumus sebagai berikut :

$$S(A, B) = \frac{\Sigma(\text{Transaksi item A dan B})}{\Sigma(\text{Transaksi})} \quad (2)$$

Sedangkan untuk nilai *confidence* nya dapat diperoleh dari rumus sebagai berikut :

$$C(A, B) = \frac{\Sigma(\text{Transaksi item A dan B})}{\Sigma(\text{Transaksi A})} \quad (3)$$

## BAB 3. METODE PENELITIAN

### 3.1 Jenis Penelitian

Jenis penelitian yang dilakukan merupakan jenis penelitian kuantitatif karena jenis data dalam penelitian ini dalam bentuk angka yang bertujuan untuk medapat gambaran yang cukup jelas mengenai suatu keadaan data yang didapatkan dengan cara mengumpulkan, menyajikan dan menganalisa data tersebut sehingga data tersebut menjadi informasi yang lebih bermakna yang dapat digunakan untuk menganalisa masalah yang sedang diteliti.

### 3.2 Tahapan Penelitian

Pada tahapan ini digunakan agar penelitian tidak melenceng dari rumusan yang telah dibuat oleh penulis, maka dibuatlah *Flow Chart Diagram* seperti pada gambar 2 berikut ini :



Gambar 3.1 Alur Penelitian

### 3.3 Pengumpulan Data

Pengumpulan data dilakukan untuk memperoleh data dan informasi yang dibutuhkan dan juga untuk memenuhi kebutuhan penelitian. Data yang digunakan penelitian ini berdasarkan sumber datanya terbagi menjadi dua yaitu data primer yang didapatkan dari hasil wawancara langsung ke pihak kpri unej dan data sekunder berupa informasi lain yang membantu dalam penelitian :

a. Data Primer

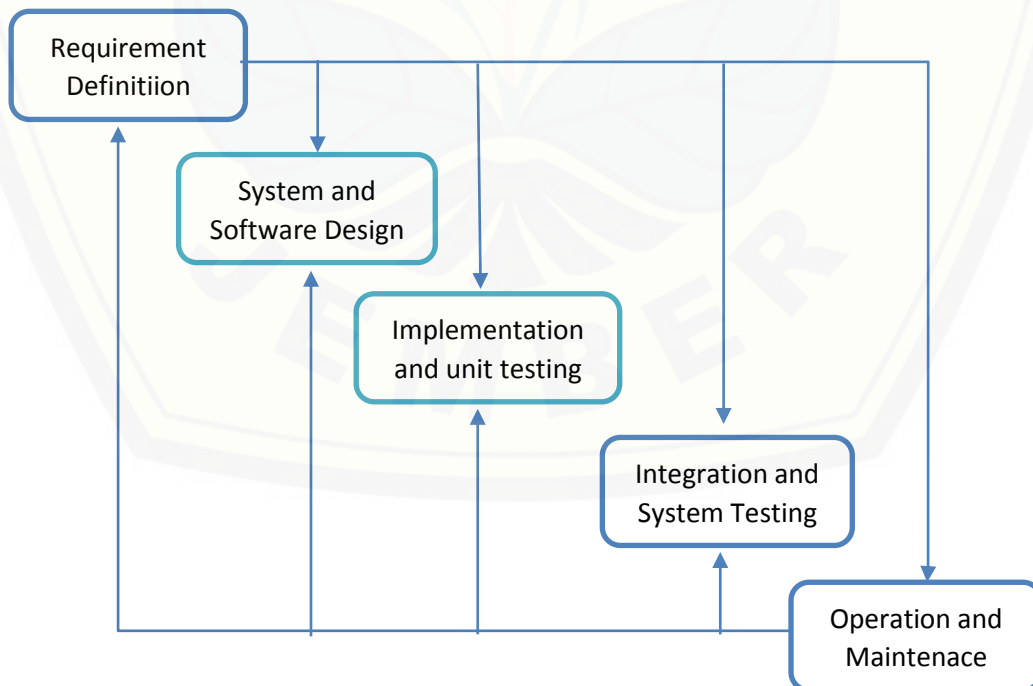
Data primer langsung diperoleh dari pihak KPRI Unej yang menjadi tempat penelitian. Adapun data yang didapatkan dari sumber data meliputi data penjualan KPRI Universitas Jember, dan juga data transaksi

b. Data Sekunder

Penulis memperoleh sumber data sekunder untuk mendukung penelitian seperti literatur yang didapatkan dari buku teks, jurnal yang diperoleh dari internet, dan penelitian sebelumnya.

### 3.4 Perancangan Sistem

Tahap ini adalah tahap perancangan sistem, dalam penelitian ini, perancangan sistem menggunakan SDLC *Waterfall*. SDLC *Waterfall* adalah proses pengembangan perangkat lunak yang paling sederhana, umum digunakan dalam proyek-proyek perangkat lunak yang memiliki *requirement* yang sudah jelas. Model ini adalah model sekuensial, sehingga ketika satu set kegiatan selesai maka aktivitas berikutnya dapat dimulai. Hal ini disebut *Waterfall* karena prosesnya mengalir "secara sistematis dari tahap satu ke tahap lainnya dalam mode ke bawah" (Waterfall Model, 2017).



Gambar 3.2 SDLC Waterfall

### 3.4.1 Analisa Kebutuhan

Pada tahap ini yang dilakukan adalah menganalisa persyaratan, batasan, dan tujuan dari sistem. Analisa kebutuhan didapatkan dari konsultasi dengan user, dapat berupa data, fitur dan gambaran yang diperlukan untuk menunjang pembuatan sistem. Data yang didapat akan dianalisa menggunakan algoritma *FP-Growth* sebagai pengambilan keputusan

### 3.4.2 Desain Sistem

Tahap ini adalah tahap untuk mentranslasikan kebutuhan perangkat lunak dari tahap analisis ke tahap desain sistem agar dapat di implementasikan menjadi program pada tahap selanjutnya. Pembuatan desain menggunakan pemodelan Unified Modelling Language (UML). Penggunaan UML dikarenakan pemodelan ini sudah mendukung konsep programing berbasis Object Orienter Programing(OOP). Pemodelan UML yang digunakan antara lain :

a. Bussines Process

Business process merupakan diagram yang menggambarkan proses dari sebuah sistem yang meliputi input, output dan goal yang merupakan tujuan dari sebuah sistem yang dibangun.

b. Use Case Diagram

Use Case Diagram menggambarkan fungsionalitas dari sebuah sistem dan interaksi antara pengguna dengan sistem untuk melakukan suatu pekerjaan tertentu. Use Case Diagram juga menggambarkan hak akses dari pengguna.

c. Scenario

Scenario digunakan untuk menjelaskan fitur sistem yang terdapat pada Use Case Diagram.

d. Sequence Diagram

Sequence Diagram digunakan untuk menunjukkan rangkaian pesan yang dikirim antar object juga interaksi antar object.

e. Activity Diagram

Activity Diagram digunakan untuk menjelaskan fitur sistem yang terdapat pada Use Case Diagram dalam bentuk gambar.



f. Class Diagram

Class Diagram digunakan untuk menggambarkan struktur statis class dalam sistem.

g. Entity Relationship Diagram (ERD)

Entity Relationship Diagram digunakan untuk menunjukkan relasi antar object.

### 3.4.3 Implementasi Metode

Pada Tahap ini dilakukan pegimplemantasian desain ke dalam kode program menggunakan bahasa pemrograman Page Hyper Text Pre-Prosesor (PHP), Cascading Style Sheet (CSS) dan javascript. *Database* Manajemen System (DBMS) yang digunakan adalah MySQL dengan tool yang digunakan adalah XAMPP. Karena XAMPP menyediakan tool yang memudahkan koneksi *database* dengan sistem berbasis web.

### 3.4.4 Pengujian Sistem

Pengujian fokus pada perangkat lunak secara dari segi logika dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (error) dan memastikan keluaran yang dihasilkan sesuai dengan yang diharapkan. Pengujian akan dilakukan dengan dua metode, yakni metode white box testing dan metode black box testing.

a. White Box Testing

White Box Testing White Box Testing merupakan pengujian pada modul pengkodean program yang dilakukan untuk mengetahui apakah sistem yang dibangun sudah sesuai antara desain program dengan kebutuhan sistem.

b. Black Box Testing

Black Box Testing merupakan pengujian yang lebih mengarah pada uji fungsionalitas dari program yang dibuat. Pengujian ini digunakan untuk menemukan ketidak sesuaian antara program dengan kebutuhan fungsional maupun non-fungsional. Hal yang dilakukan pada pengujian ini adalah dengan melakukan pengujian terhadap interface atau tampilan program untuk

memastikan suatu masukan diproses oleh sistem dengan benar dan menghasilkan keluaran yang sesuai dengan perancangan.

#### **3.4.5 Maintenance**

Tahap ini dilakukan untuk mengkoreksi beberapa kesalahan yang tidak ditemukan pada tahap sebelumnya. Perangkat lunak yang sudah disampaikan pada perusahaan sangat mungkin akan mengalami perubahan. Perubahan tersebut dapat terjadi seiring pengerjaan program dengan penyesuaian kerja sistem. Penyempurnaan bug (error) akan membantu penggunaan perangkat lunak.





## BAB 4. PERANCANGAN SISTEM

Bab ini akan menguraikan tentang perancangan desain sistem untuk mengimplementasikan metode *FP-Growth* pada sistem informasi penentuan paket pembelian produk KPRI Unej. Pada perancangan sistem, yang pertama dilakukan adalah menganalisa kebutuhan fungsional dan non fungsional sistem, kemudian dilanjutkan dengan mendisain *business process*, *use case diagram*, *scenario*, *activity diagram*, *sequence diagram*, *class diagram*, dan *entity relationship diagram (ERD)*.

### 4.1 Deskripsi Sistem

Sistem informasi penentu paket pembelian produk Kpri Unej adalah sebuah sistem informasi yang menangani transaksi penjualan yang terjadi pada Kpri unej. Sistem ini memiliki beberapa fitur yaitu mengelola data pegawai, mengelola data barang, transaksi penjualan, dan juga fitur analisa data penjualan menggunakan algoritma *Fp-Growth* guna menemukan *cross-selling* produk tertentu.

### 4.2 Analisa Kebutuhan Sistem

Pada tahap analisa kebutuhan sistem, yang pertama dilakukan adalah pendefinisian kebutuhan untuk merancang dan membangun sistem informasi penentuan paket pembelian produk KPRI Unej. Sekumpulan literatur yang telah didapat dianalisis untuk mendefinisikan kebutuhan sistem ke dalam kebutuhan fungsional dan kebutuhan non fungsional

#### 4.2.1 Kebutuhan Fungsional

Kebutuhan fungsional adalah suatu proses yang dilakukan oleh perangkat lunak untuk mendapatkan keluaran sesuai kebutuhan dengan memasukkan beberapa inputan data. Kebutuhan fungsional pada Sistem informasi penentuan paket pembelian produk Kpri Unej digambarkan pada tabel 4.1

Tabel 4. 1 Kebutuhan Fungsional

No	Identifikasi
1	Sistem dapat mengidentifikasi user yang masuk sistem dan juga hak akses yang diberikan pada user
2	Sistem dapat mengelola data barang meliputi melihat, mengubah dan menambah
3	Sistem dapat mengelola data kategori meliputi melihat, mengubah dan menambah
4	Sistem dapat mengelola data supplier meliputi melihat, mengubah dan menambah
5	Sistem dapat mengelola transaksi penjualan
6	Sistem dapat menampilkan informasi kombinasi barang yang sering dibeli secara bersamaan dalam satu keranjang menggunakan algoritma <i>Fp-Growth</i>
7	Sistem dapat menampilkan <i>cross seling</i> produk

#### 4.2.2 Perangkat Keras

Perangkat keras yang dibutuhkan untuk menggunakan sistem informasi penentuan paket pembelian produk KPRI Unej ini adalah PC atau Laptop yang harus memiliki spesifikasi minimal sebagai berikut :

1. RAM (Random Access Memory) minimal 1 GB
2. Monitor
3. Keyboard
4. Mouse

#### 4.2.3 Perangkat Lunak

Perangkat lunak yang diperlukan untuk mengoperasikan sistem informasi penentuan paket pembelian produk KPRI Unej ini antara lain :

1. Sistem Operasi Windows 7/8/10
2. Browser untuk mengakses sistem
3. XAMPP untuk manajemen database

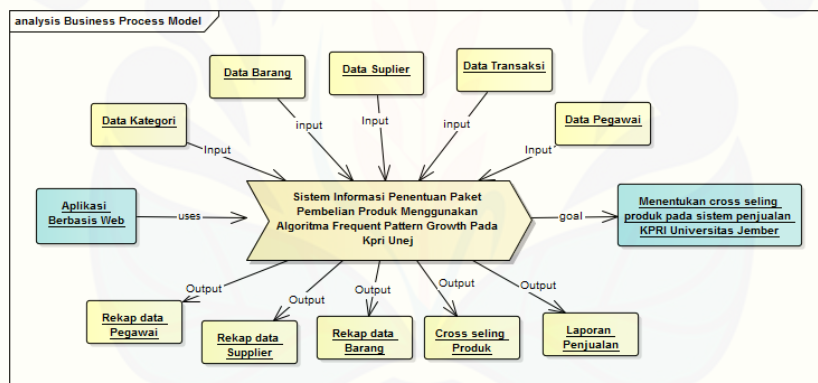
### 4.3 Desain Sistem

Tahapan ini menggambarkan desain sistem informasi dengan beberapa tahap yaitu *business process*, *use case diagram*, *scenario*, *activity diagram*, *sequence diagram*, *class diagram*, dan *entity relationship diagram (ERD)*.

#### 4.3.1 Business process

*Business process* merupakan diagram yang menggambarkan tentang kebutuhan data yang diperlukan oleh sistem. Dalam *business process* terdapat beberapa poin diantaranya :

1. *Input* : Data masukan yang dimasukkan ke dalam sistem.
2. *Output* : Data yang dihasilkan oleh sistem.
3. *Used* : Platform yang digunakan oleh sistem
4. *Process* : Kegiatan yang dilakukan oleh sistem
5. *Goal* : Tujuan yang ingin dicapai

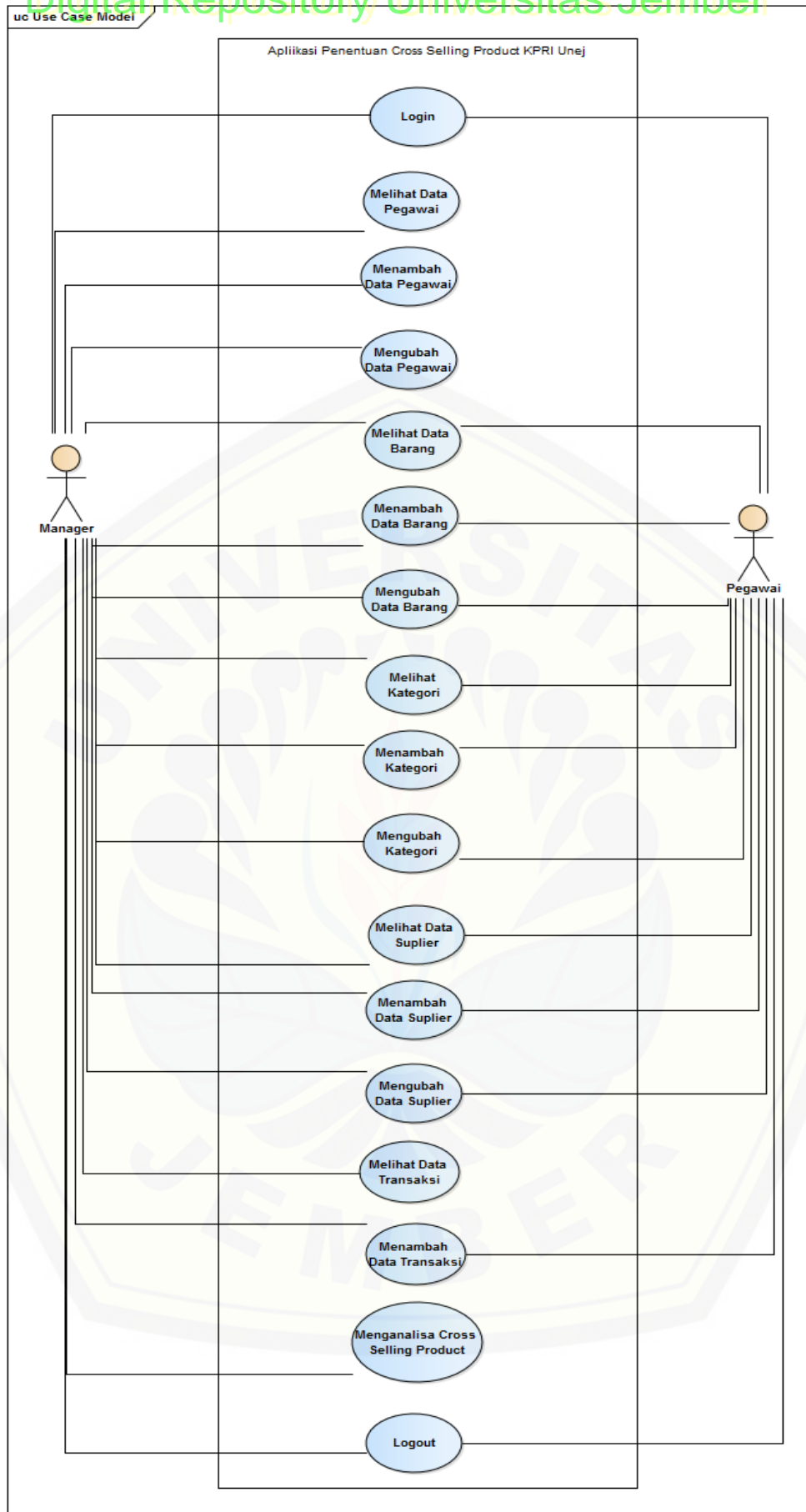


Gambar 4.1 Business Process

Gambar 4.1 merupakan *business process* dari sistem informasi penentuan paket pembelian produk KPRI Unej. Business process menjelaskan *input*, *output* dan *goals* yang ingin dicapai serta *uses* yang diaplikasikan pada sistem.

#### 4.3.2 Usecase Diagram

*Usecase diagram* merupakan permodelan yang dibuat untuk menggambarkan interaksi antara aktor dengan sistem yang akan dibangun. Dengan menggunakan *Usecase diagram*, peneliti dapat mengetahui interaksi yang bisa dilakukan antara aktor terhadap sistem sesuai dengan hak akses yang telah ditentukan. *Usecase diagram* ditunjukkan pada gambar 4.2



Gambar 4.2 Usecase Diagram

Penjelasan mengenai aktor-aktor sebagai pengguna sistem pada gambar 4.2 dapat dilihat pada tabel 4.1. penjelasan lain mengenai fitur-fitur yang tersedia dalam sistem informasi paket pembelian produk KPRI Unej dapat dilihat pada tabel 4.2.

Tabel 4. 2 Definisi Aktor

No	Aktor	Deskripsi
1.	Manager	Aktor manager memiliki hak akses untuk melakukan login, melihat data pegawai, menambah data pegawai, mengubah data pegawai, melihat data barang, melihat data jategori, melihat data supplier, melihat data transaksi, menganalisa <i>cross selling product</i> dan log out
2.	Pegawai	Aktor pegawai memiliki hak akses untuk melakukan login, melihat data barang, menambah data barang, mengubah data barang, melihat kategori, menambah kategori, mengubah kategori, melihat data supplier, menambah data supplier, mengubah data supplier, menambah data transaksi dan log out

Tabel 4. 3 Definisi Usecase

No.	Use Case	Deskripsi
1	Login	Menggambarkan proses <i>autentifikasi</i> untuk masuk sistem
2	Melihat Data Pegawai	Menggambarkan proses melihat data pegawai yang sudah ditambahkan
3	Menambah Data Pegawai	Menggambarkan proses menambahkan data pegawai pada sistem
4	Mengubah Data Pegawai	Menggambarkan proses mengubah data pegawai yang sudah ada pada sistem
5	Melihat Data Barang	Menggambarkan proses melihat data barang yang sudah ditambahkan
6	Menambah Data Barang	Menggambarkan proses menambahkan data barang pada sistem
7	Mengubah Data Barang	Menggambarkan proses mengubah data barang yang sudah ada pada sistem
8	Melihat Kategori	Menggambarkan proses melihat data kategori yang sudah ditambahkan
9	Menambah Kategori	Menggambarkan proses menambahkan data kategori pada sistem
10	Mengubah Kategori	Menggambarkan proses mengubah data kategori yang sudah ada pada sistem
11	Melihat Data	Menggambarkan proses melihat data supplier yang sudah



	Supplier	ditambahkan
12	Menambah Data Supplier	Menggambarkan proses menambahkan data supplier pada sistem
13	Mengubah Data Supplier	Menggambarkan proses mengubah data supplier yang sudah ada pada sistem
14	Melihat Data Transaksi	Menggambarkan proses melihat data transaksi yang sudah ditambahkan
15	Menambah Data Transaksi	Menggambarkan proses menambahkan data transaksi pada sistem
16	Menganalisa <i>Cross Selling Product</i>	Menggambarkan proses menganalisa cross seling produk dengan memasukkan beberapa inputan yang dibutuhkan
17	<i>Logout</i>	Menggambarkan proses untuk keluar dari sistem

### 4.3.3 Skenario

Skenario merupakan gambaran alur kerja sistem yang terdapat dalam *usecase* diagram baik scenario utama maupun alternatif. Skenario sistem yang sesuai dengan *usecase* diagram dapat dilihat dibawah ini.

#### 1. Skenario *Login*

Skenario *Login* merupakan alur yang menggambarkan aksi dari aktor dan reaksi dari sistem pada saat aktor akan masuk sistem. Terdapat dua actor yang dapat masuk sistem yaitu pegawai dan manager. Scenario ini dapat dilihat pada Lampiran 1 tabel 1.1

#### 2. Skenario Melihat Data Pegawai

Skenario melihat data pegawai merupakan alur dimana user dapat melihat data pegawai yang ada. Data pegawai yang dapat dilihat meliputi nama pegawai, jabatan, nomor ktp, nomor telepon, username dan password. Skenario ini dapat dilihat pada Gambar 4.4.

Tabel 4. 4 Skenario Melihat Data Pegawai

<b>No Usecase</b>	UCS 002
<b>Nama Usecase</b>	Melihat Data Pegawai
<b>Aktor</b>	Manager
<b>Deskripsi</b>	Aktor dapat mlihat data pegawai
<b>Prekondisi</b>	Aktor telah masuk sistem
<b>Prakondisi</b>	Aktor berhasil melihat data pegawai

Flow Of Events	
Skenario Normal : Lihat Data Pegawai	
Aktor	Sitem
1. Klik menu pegawai	
	2. Menampilkan daftar pegawai

### 3. Skenario Menambah Data Pegawai

Skenario menambah data pegawai merupakan alur dimana user dapat menambahkan data pegawai pada sistem. Apabila ada penambahan pegawai yang akan mengakses sistem ini, maka manajer dapat menggunakan fitur menambah data pegawai untuk menambahkan data pegawai tersebut kedalam database. Skenario menambahkan data pegawai tersebut dapat dilihat pada Tabel 4.5.

Tabel 4. 5 Skenario Menambah Data Pegawai

No Usecase	UCS 003
Nama Usecase	Menambah Data Pegawai
Aktor	Manager
Deskripsi	Aktor dapat menambah data pegawai
Prekondisi	Aktor telah masuk system
Prakondisi	Aktor berhasil menambah data pegawai
Skenario Normal : Menambah Data Pegawai	
Aktor	Sitem
1. Klik menu pegawai	
	2. Menampilkan daftar pegawai
3. Klik tombol add	
	4. Menampilkan form pengisian data pegawai
5. Mengisi form data pegawai	
6. Klik tombol Save	
	7. Menampilkan notifikasi “ pengisian data berhasil.”
	8. Menampilkan daftar pegawai
Skenario Alternatif : Data Tidak lengkap	
Aktor	Sitem

5.a Mengisi form data pegawai	
6.a Klik tombol Save	
	7.a Menampilkan notifikasi “Please fill out this field” pada form pegawai
<b>Skenario Alternatif : No Ktp/No telp tidak sesuai</b>	
<b>Aktor</b>	<b>Sitem</b>
5.b Mengisi form data pegawai	
6.b Klik tombol Save	
	7.b Menampilkan notifikasi “Data harus diisi dengan data yang benar” pada form pegawai
<b>Skenario Alternatif : Username sudah ada</b>	
<b>Aktor</b>	<b>Sitem</b>
5.c Mengisi form data pegawai	
6.c Klik tombol Save	
	7.c Menampilkan notifikasi “Username sudah digunakan, coba lagi” pada form pegawai
<b>Skenario Alternatif : Username/Password tidak sesuai</b>	
<b>Aktor</b>	<b>Sitem</b>
5.d Mengisi form data pegawai	
6.d Klik tombol Save	
	7.d Menampilkan notifikasi “Password/Username harus lebih dari 5 karakter” pada form pegawai

Skenario mengubah data pegawai merupakan alur dimana user dapat mengubah data pegawai jika ada data yang ingin dirubah. Semua data dalam daabase pegawai dapat dirubah tanpa terkecuali. Perubahan data harus memenuhi syarat syarat yang telah ditentukan oleh sistem. Skenario fitur untuk mengubah data pegawai dapat dilihat pada Tabel 4.6.

Tabel 4. 6 Skenario Mengubah Data Pegawai

<b>No Usecase</b>	UCS 004
<b>Nama Usecase</b>	Mengubah Data Pegawai
<b>Aktor</b>	Manager
<b>Deskripsi</b>	Aktor dapat mengubah data pegawai
<b>Prekondisi</b>	Aktor telah masuk sistem



<b>Prakondisi</b>	Aktor berhasil mengubah data pegawai
<b>Flow Of Events</b>	
<b>Skenario Normal : Ubah Data Pegawai</b>	
<b>Aktor</b>	<b>Sitem</b>
1. Klik menu pegawai	
	2. Menampilkan daftar pegawai
3. Klik tombol Ubah	
	4. Menampilkan form pengisian data pegawai
5. Merubah form data pegawai	
6. Klik tombol Save	
	7. Menampilkan notifikasi “ pengisian data berhasil.”
	8. Menampilkan daftar user
<b>Skenario Alternatif : Data Tidak lengkap</b>	
<b>Aktor</b>	<b>Sitem</b>
5.a Merubah form data pegawai	
6.a Klik tombol Save	
	7.a Menampilkan notifikasi “Please fill out this field” pada form pegawai
<b>Skenario Alternatif : No Ktp/No telp tidak sesuai</b>	
<b>Aktor</b>	<b>Sitem</b>
5.b Merubah form data pegawai	
6.b Klik tombol Save	
	7.b Menampilkan notifikasi “Data harus diisi dengan data yang benar” pada form pegawai
<b>Skenario Alternatif : Username sudah ada</b>	
<b>Aktor</b>	<b>Sitem</b>
5.c Merubah form data pegawai	
6.c Klik tombol Save	
	7.c Menampilkan notifikasi “Username sudah digunakan, coba lagi” pada form pegawai
<b>Skenario Alternatif : Username/Password tidak sesuai</b>	
<b>Aktor</b>	<b>Sitem</b>

5.d Merubah form data pegawai	
6.d Klik tombol Save	
	7.d Menampilkan notifikasi "Password/Username harus lebih dari 5 karakter" pada form pegawai

#### 4. Skenario Melihat Data Barang

Skenario melihat data barang merupakan alur dimana user dapat melihat data barang yang ada. Setiap barang yang terdapat dalam toko akan dicatat ke dalam database melalui fitur ini, sehingga user dapat melihat semua data barang yang terdapat dalam toko tersebut. Skenario melihat data barang dijelaskan pada lampiran 1 tabel 1.1 .

#### 5. Skenario Menambah Data Barang

Skenario menambah data barang merupakan alur dimana user dapat menambahkan data barang pada sistem. Ketika ada barang masuk, user dapat menambahkan data barang ke dalam sistem melalui fitur ini. Skenario menambah data barang dapat dilihat pada lampiran 1 tabel 1.3.

#### 6. Skenario Mengubah Data Barang

Skenario mengubah data barang merupakan alur dimana user dapat mengubah data barang yang sudah ada. Data barang yang telah tersimpan dapat diubah dengan menggunakan fitur ini. Fitur ini digunakan jika ada perubahan data pada produk atau barang yang ada dalam swalayn. Skenario untuk mengubah data barang dapat dilihat pada lampiran 1 tabel 1.4.

#### 7. Skenario Melihat Data Kategori

Skenario melihat data kategori merupakan alur dimana user dapat melihat data kategori yang ada. Data kategori ini berisi id kategori dan nama dari kategori itu sendiri. Data kategori digunakan untuk mengelompokkan barang dalam kelompok kecil dalam setiap kategori. Skenario melihat data ketegori dijelaskan pada lampiran 1 tabel 1.5.

#### 8. Skenario Menambah Data Kategori

Skenario menambah data kategori merupakan alur dimana user dapat menambahkan data kategori pada sistem. Penambahan data kategori memiliki proes yang sama dengan menambah data produk, perbedaannya hanya terdapat

dalam isi datanya saja. Skenario menambah data kategori dijelaskan pada lampiran 1 tabel 1.6.

#### 9. Skenario Mengubah Data Kategori

Skenario mengubah data kategori merupakan alur dimana user dapat mengubah data kategori yang sudah ada. Skenario ini dapat dilihat pada lampiran 1 tabel 1.7. skenario yang disajikan memiliki fungsi yang sama dengan fitur mengubah data barang. Fitur ini hanya memiliki satu alternatif skenario yaitu skenario alternatif ketika data tidak lengkap.

#### 10. Skenario Melihat Data Supplier

Skenario melihat data supplier merupakan alur dimana user dapat melihat data supplier yang ada. Data yang dapat dilihat pada fitur ini adalah data supplier yang meliputi nama dan id supplier. Skenario melihat data supplier hanya memiliki skenario normal saja. Skenario melihat data supplier ini dijelaskan pada lampiran 1 tabel 1.8.

#### 11. Skenario Menambah Data Supplier

Skenario menambah data supplier merupakan alur dimana user dapat menambahkan data supplier pada sistem. Skenario tersebut dapat dijelaskan pada lampiran 1 tabel 1.9. User dapat menggunakan fitur ini ketika user ingin memasukkan supplier baru yang belum tersimpan dalam database. Skenario ini memiliki dua jalur yaitu ketika skenario normal dijalankan dan pada saat alternatif skenario berjalan ketika data yang dimasukkan tidak lengkap.

#### 12. Skenario Mengubah Data Supplier

Skenario mengubah data supplier merupakan alur dimana user dapat mengubah data supplier yang sudah ada. Ketika ada perubahan data dalam sistem, maka fitur ini dapat digunakan untuk merubah data supplier tersebut, hal ini dapat dilihat pada lampiran 1 tabel 1.10.

#### 13. Skenario Melihat Data Transaksi

Skenario melihat data transaksi merupakan alur dimana user dapat melihat data transaksi yang berhasil. Transaksi yang terjadi dalam sistem ini disimpan melalui fitur menambahkan data transaksi dan dapat dilihat menggunakan fitur ini. Skenario tersebut terdapat pada lampiran 1 tabe; 1.11

#### 14. Skenario Menambah Data Transaksi

Skenario menambah data transaksi merupakan alur dimana user dapat menambahkan data transaksi penjualan pada sistem. Sesuai dengan skenario pada lampiran 1 tabel 1.12, menjelaskan bahwa terdapat satu skenario normal dan dua skenario alternatif yaitu skenario data tidak lengkap dan skenario stok barang tidak mencukupi.

#### 15. Skenario Menganalisa *Cross Selling Product*

Skenario menganalisa *cross selling product* merupakan alur dimana user dapat menganalisa data produk untuk mendapatkan *cross selling product* berdasarkan metode yang diterapkan pada sistem. Semua data penjualan yang telah tersimpan dalam database dapat diproses pada fitur ini untuk menghasilkan paket pembelian produk. Skenario ini dapat dilihat pada lampiran 1 tabel 1.13.

#### 16. Skenario *Logout*

Skenario *Logout* merupakan alur yang menggambarkan aksi dari aktor dan reaksi dari sistem pada saat aktor akan keluar dari sistem. Skenario ini dapat dilihat pada Lampiran 1 tabel 1.14.

### 4.3.4 *Sequence Diagram*

*Sequence Diagram* merupakan diagram yang menggambarkan interaksi antar satu kelas dengan kelas yang lain sesuai dengan alurnya masing-masing. *Sequence Diagram* memvisualkan aliran logika dalam sistem. *Sequence Diagram* dalam sistem ini antara lain sebagai berikut :

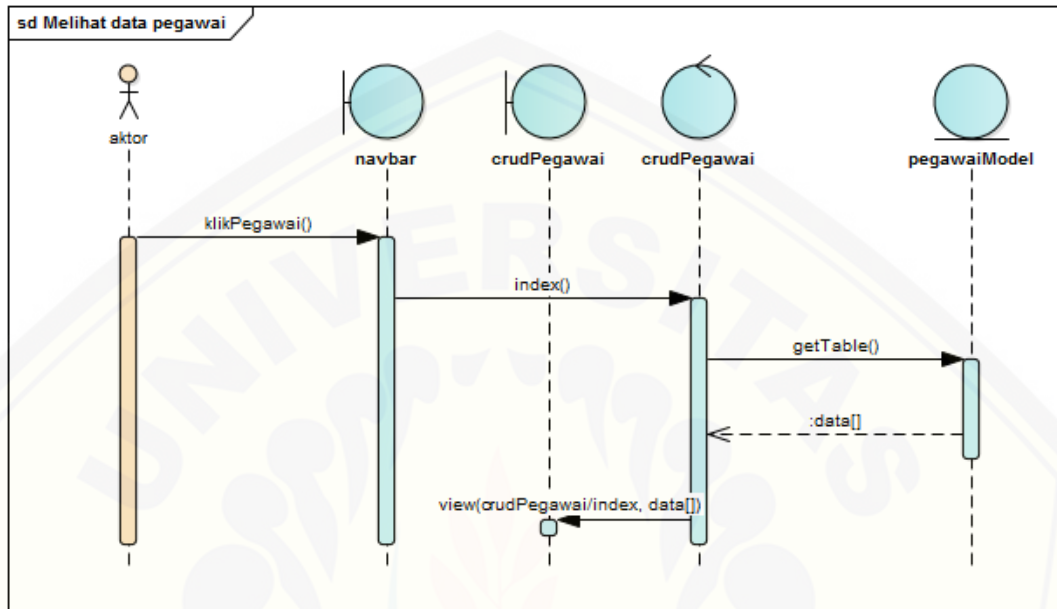
#### 1. *Sequence Diagram Login*

Penggambaran *sequence diagram login* digunakan untuk menjelaskan method atau fungsi yang digunakan untuk masuk sistem. Dalam *sequence diagram login* terdapat tiga alternatif yaitu pada saat data kosong, password salah dan username salah. Penggambaran fungsi atau method yang akan digunakan seperti yang ditunjukkan pada Lampiran 2 gambar 2.1.

#### 2. *Sequence Diagram Melihat Data Pegawai*

Penggambaran melihat data pegawai pada *sequence diagram* dapat dilihat pada gambar 4.3. penggambaran tersebut digunakan untuk menjelaskan method atau fungsi yang digunakan untuk menjalankan fitur melihat data pegawai. Data

pegawai yang telah disimpan dalam *database* dapat dilihat melalui fitur ini. Aktor dapat menekan menu pegawai pada navbar untuk menjalankan fungsi ini. Setelah aktor memberika perintah ke pada sistem untuk menampilkan data pegawai maka sistem akan menanggapi dengan melanjutkan perintah tersebut ke *class – class* lainnya untuk menampilkan data tersebut.

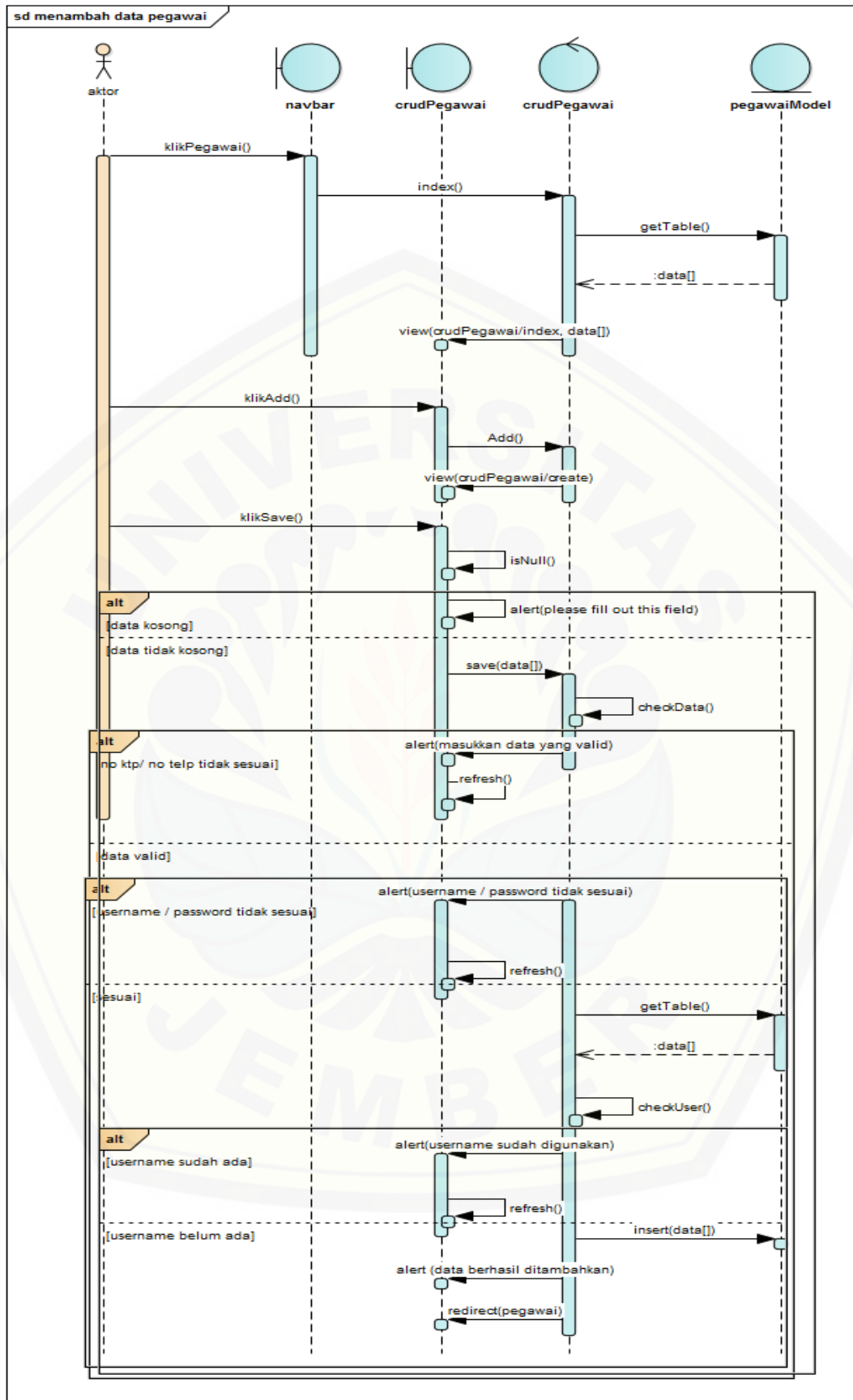


Gambar 4.3 *Sequence Diagram* Melihat Data Pegawai

### 3. *Sequence Diagram* Menambah Data Pegawai

Penggambaran *sequence diagram* menambah data pegawai digunakan untuk menjelaskan fungsi yang dijalankan untuk menambah data pegawai. Untuk menjalankan fungsi menambahkan data pegawai, aktor harus sudah berada dalam halaman melihat data pegawai. Ketika aktor menekan tombol *Add* maka sistem akan menanggapi dengan memanggil *controller crudProduk* untuk menjalankan fungsi menambahkan data pegawai. Penggambaran fungsi atau method yang akan digunakan ditunjukan pada Gambar 4.4.



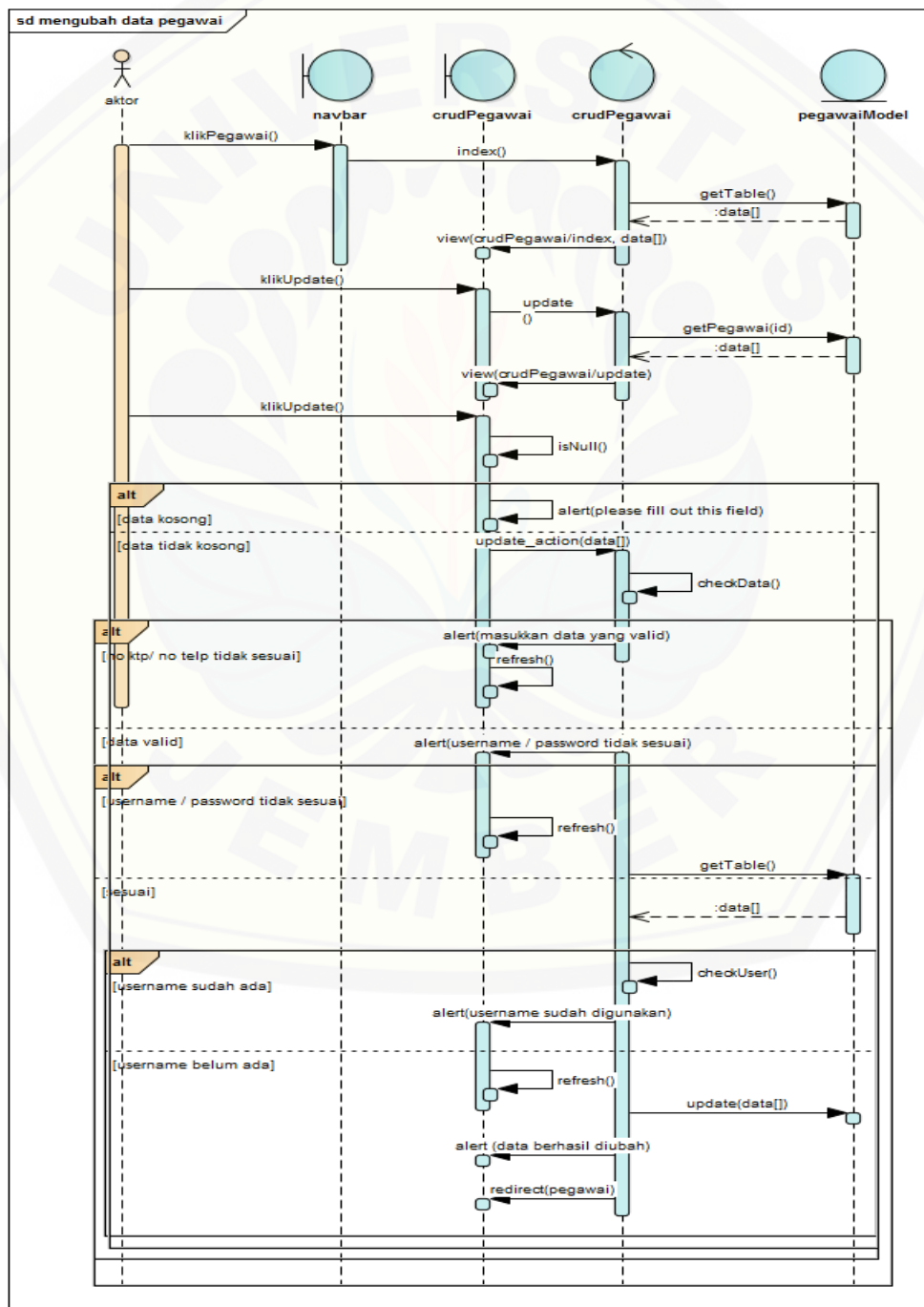


Gambar 4.4 Sequence Diagram Menambah Data Pegawai



4. Sequence Diagram Mengubah Data Pegawai

Penggambaran *sequence diagram* mengubah data pegawai digunakan untuk menjelaskan method atau fungsi yang digunakan untuk mengubah data pegawai. Aktor dapat menjalankan fungsi ini dengan menekan tombol update. *Sequence diagram* mengubah data pegawai memiliki empat alternatif yaitu ketika data kosong, nomor telepon tidak sesuai, username atau password salah dan ketika username sudah digunakan. Penggambaran fungsi atau method yang akan digunakan seperti yang ditunjukkan pada Gambar 4.5.



Gambar 4.5 Sequence Diagram Mengubah Data Pegawai

#### 5. *Sequence Diagram* Melihat Data Barang

Penggambaran *sequence diagram* melihat data barang digunakan untuk menjelaskan method atau fungsi yang digunakan untuk melihat data barang. Ketika aktor menekan tombol produk maka navbar akan menanggapi dengan meneruskan ke *class controller* melalui fungsi *index*. Setelah *controller* mendapatkan perintah dari aktor maka *controller* akan meminta data dari model dan kemudian dikirimkan kembali untuk menampilkan data kepada aktor. Penggambaran fungsi atau method yang akan digunakan seperti yang ditunjukkan pada lampiran 2 gambar 2.2.

#### 6. *Sequence Diagram* Menambah Data Barang

Penggambaran *sequence diagram* menambah data barang digunakan untuk menjelaskan method atau fungsi yang digunakan untuk menambah data barang. Fungsi ini akan berjalan jika aktor menekan tombol *add* pada halaman melihat data barang. Sistem akan memanggil fungsi *add* pada *class controller* untuk menjalankan fitur ini. Setelah aktor selesai mengisi form data barang, aktor dapat menekan tombol *save* untuk menyimpan data tersebut ke dalam *database*. Penggambaran *sequence diagram* melihat data barang dapat dilihat pada lampiran 2 gambar 2.3.

#### 7. *Sequence Diagram* Mengubah Data Barang

Penggambaran *sequence diagram* mengubah data barang digunakan untuk menjelaskan method atau fungsi yang digunakan untuk mengubah data barang. Berbeda dengan menambahkan data barang, untuk mengubah data barang sistem harus mengambil dahulu data yang akan diperbarui dari *database*. Sistem mengambil data dari *database* melalui fungsi *getTable*. Pada lampiran 2 gambar 2.4 terdapat rincian gambar *sequence diagram* mengubah data barang.

#### 8. *Sequence Diagram* Melihat Data Kategori

Penggambaran *sequence diagram* melihat data kategori digunakan untuk menjelaskan method atau difungsikan untuk melihat data kategori. Fungsi ini akan berjalan ketika aktor menekan menu kategori pada navbar. Sistem akan menampilkan data kategori yang telah diambil dari *class kategoriModel* pada halaman melihat data kategori. Penggambaran fungsi atau method yang akan digunakan tersebut ditunjukkan pada Lampiran 2 gambar 2.5.

#### 9. *Sequence Diagram* Menambah Data Kategori

Penggambaran menambahkan data kategori yang dapat menjelaskan suatu metode atau fungsi dari fitur tersebut dapat dijelaskan oleh *sequence diagram* menambah data kategori. Fitur ini akan berjalan jika aktor menekan tombol *add*. Ketika tombol *add* ditekan, maka sistem akan menampilkan form pengisian data melalui fungsi *view*. Aktor dapat menyimpan data yang sudah diisi dengan menekan tombol *save*. Saat perintah *save* dijalankan, sistem akan menanggapi dengan menambahkan data pada database melalui fungsi *insert*. Lampiran 2 gambar 2.6 merupakan penggambaran fungsi dari *sequence diagram* menambah data kategori

#### 10. *Sequence Diagram* Mengubah Data Kategori

*Sequence diagram* mengubah data kategori dapat digambarkan sebagai fungsi atau method yang digunakan untuk mengubah data kategori. Fungsi yang terdapat dalam fitur mengubah data kategori hampir sama dengan fungsi untuk menambah data kategori, perbedaannya terdapat pada fungsi *update* yang terdapat dalam fitur mengubah data kategori. Penggambaran method dapat dilihat pada lampiran 2 gambar 2.7.

#### 11. *Sequence Diagram* Melihat Data Supplier

Lampiran 2 gambar 2.8 merupakan penggambaran *sequence diagram* melihat data supplier yang digunakan untuk menjelaskan method atau fungsi dalam melihat data supplier. Untuk menjalankan fitur ini aktor dapat menekan menu supplier pada navbar.

#### 12. *Sequence Diagram* Menambah Data Supplier

Penggambaran *sequence diagram* menambah data supplier yang ditunjukkan pada lampiran 2 gambar 2.9 ini menjelaskan method atau fungsi yang digunakan untuk menambah data supplier. Aktor dapat menjalankan fitur ini dengan menekan tombol *add*. Sistem akan bereaksi ketika perintah menambahkan data dijalankan. Sistem akan menampilkan halaman pengisian data melalui fungsi *view* kemudian ketika data sudah dilengkapi, aktor dapat menekan tombol *save* untuk menyimpan data kedalam *database*

### 13. *Sequence Diagram* Mengubah Data Supplier

*sequence diagram* mengubah data supplier digunakan untuk menggambarkan method atau fungsi yang digunakan untuk mengubah data supplier. Ketika tombol *update* ditekan, sistem akan memanggil fungsi *getSupplier* untuk menampilkan data yang akan dirubah. Aktor dapat menyimpan perubahan data dengan menekan tombol *save*. Penjelasan fungsi atau method yang akan digunakan seperti yang ditunjukkan pada Lampiran 2 gambar 2.10.

### 14. *Sequence Diagram* Melihat Data Transaksi

Lampiran 2 gambar 2.11 menjelaskan tentang *sequence diagram* melihat data transaksi yang digunakan untuk menggambarkan method atau fungsi yang digunakan untuk melihat data transaksi yang berhasil. Untuk menggunakan fitur ini aktor dapat menekan menu transaksi pada navbar. Sistem akan menanggapi dengan memanggil *controller crudTransaksi* ketika menu transaksi pada navbar ditekan.

### 15. *Sequence Diagram* Menambah Data Transaksi

Penggambaran *sequence diagram* menambah data transaksi digunakan untuk mendeskripsikan method atau fungsi yang digunakan untuk menambah data transaksi penjualan. Fitur menambah data transaksi memiliki dua alternatif yaitu ketika data kosong dan data pembelian melebihi stok barang. Penggambaran method atau fungsi tersebut terdapat pada Lampiran 2 gambar 2.12.

### 16. *Sequence Diagram* Menganalisa *Cross Selling Product*

*Sequence diagram* menganalisa *cross selling product* digunakan untuk mendeskripsikan method atau fungsi yang digunakan untuk menganalisa *cross selling product*. Fitur ini akan berjalan ketika aktor menekan tombol analisa. Sistem akan menanggapi dengan melakukan perhitungan melalui fungsi analisa pada *controller fpGrowth*. Data hasil perhitungan akan ditampilkan melalui fungsi *view*. fungsi atau method digambarkan sesuai dengan yang tertera pada Lampiran 2 gambar 2.13.

### 17. *Sequence Diagram* Logout

*sequence diagram logout* yang tertera pada lampiran 2 gambar 2.14 menjelaskan tentang penggambaran method atau fungsi yang digunakan untuk keluar dari sistem. Aktor dapat keluar dengan menekan tombol *log out*. Saat aktor

menekan tombol *log out*, sistem akan memanggil fungsi index yang terdapat pada *controller login* dan mengeluarkan user dari sistem

#### 4.3.5 Activity Diagram

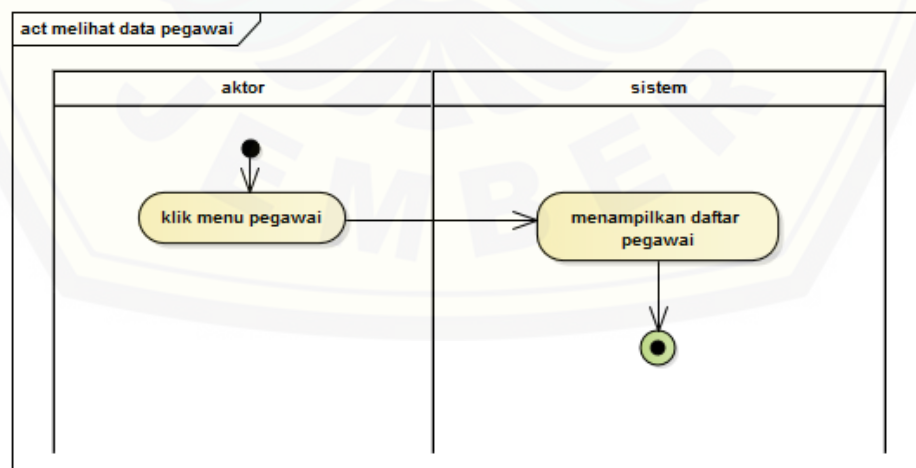
*Activity diagram* merupakan diagram yang menggambarkan berbagai alir aktivitas yang ada dalam sistem yang sedang dirancang, bagaimana awal dari masing-masing alir, *decision* yang mungkin terjadi, dan bagaimana masing-masing alir berakhir.

##### 1. Activity Diagram Login

*Activity diagram login* menggambarkan tentang proses yang terjadi ketika aktor akan memasuki sistem. Activity diawali dengan sistem menampilkan halaman login kemudian aktor dapat mengisi form dan klik login untuk masuk kedalam sistem. Aktor dapat masuk kedalam sistem jika data yang diisikan kedalam form adalah data yang valid. Penjelasan tentang *Activity Diagram login* dapat dilihat pada lampiran 3 gambar 3.1.

##### 2. Activity Diagram Melihat Data Pegawai

*Activity diagram* melihat data pegawai menggambarkan tentang proses yang terjadi ketika aktor akan melihat data pegawai. Ketika aktor menekan menu pegawai, maka sistem akan menanggapi dengan menampilkan halaman daftar pegawai. Penggambaran *Activity diagram* melihat data pegawai terdapat pada Gambar 4.9.

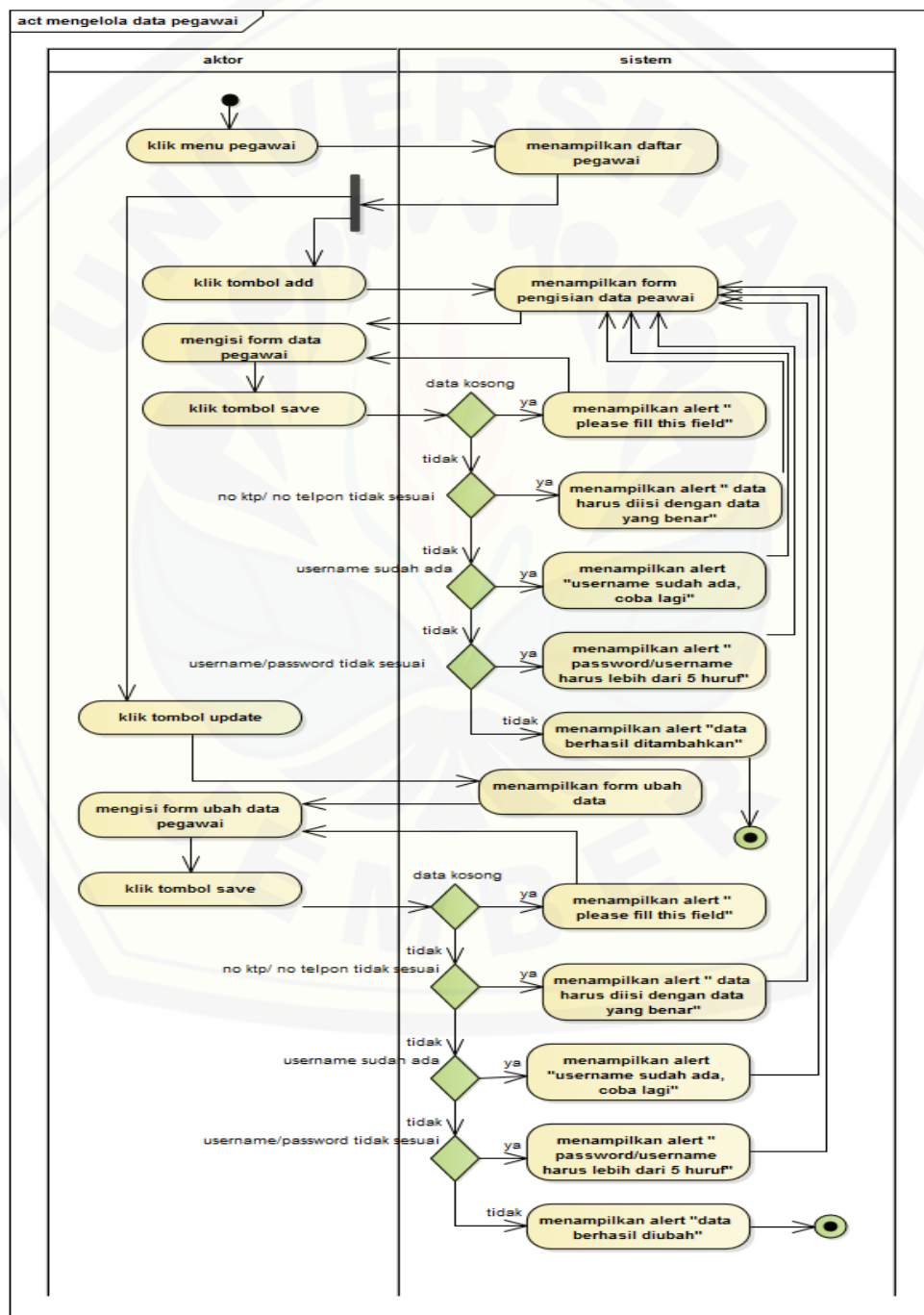


Gambar 4.6 Activity Diagram Melihat Data Pegawai



### 3. Activity Diagram Mengelola Data Pegawai

Activity diagram mengelola data pegawai menggambarkan tentang proses yang terjadi ketika aktor akan mengubah maupun menambah data pegawai. Aktor dapat memilih fitur yang akan digunakan. Aktor dapat menekan tombol *add* untuk menambahkan data namun jika aktor ingin mengubah data, aktor dapat menekan tombol *update*. Penjelasan tentang *activity diagram* mengelola data pegawai dapat dilihat pada Gambar 4.7.



Gambar 4.7 Activity Mengelola Data Pegawai



#### 4. *Activity Diagram* Melihat Data Barang

*Activity diagram* melihat data barang menggambarkan tentang proses yang terjadi ketika aktor akan melihat data barang. *Activity diagram* ini berjalan ketika aktor menekan menu produk. Sistem akan menanggapi dengan menampilkan halaman daftar produk ketika aktor menekan menu produk. Penggambaran *activity diagram* melihat data barang dapat dilihat pada Lampiran 3 gambar 3.2.

#### 5. *Activity Diagram* Mengelola Data Barang

*Activity diagram* mengelola data barang menjelaskan tentang proses yang terjadi ketika aktor akan mengubah atau menambah data barang. Menambah atau mengubah data barang memiliki beberapa kondisi dimana sistem akan menolak data yang dimasukkan yaitu ketika data kosong, data tidak sesuai dan stok kurang dari satu. *Activity diagram* mengelola data barang digambarkan pada lampiran 3 gambar 3.3.

#### 6. *Activity Diagram* Melihat Data Kategori

*Activity diagram* melihat data kategori mendeskripsikan tentang proses yang terjadi ketika aktor akan melihat data kategori. Aktor dapat melihat data kategori dengan menekan menu kategori. Setelah aktor menekan menu kategori, sistem akan menanggapi dengan menampilkan data kategori. Penggambaran *Activity diagram* melihat data kategori terdapat pada lampiran 3 gambar 3.4.

#### 7. *Activity Diagram* Mengelola Data Kategori

Lampiran 3 gambar 3.4 menggambarkan tentang *Activity diagram* mengelola data kategori. *Activity diagram* ini berisi tentang proses yang terjadi ketika aktor mengubah atau menambah data kategori. Sistem hanya memiliki satu kondisi dimana sistem tidak akan menerima data yang dimasukkan yaitu ketika ada data yang kosong.

#### 8. *Activity Diagram* Melihat Data Supplier

Penggambaran tentang proses yang terjadi ketika aktor akan melihat data supplier terdapat pada lampiran 3 gambar 3.5. Sistem akan menampilkan daftar supplier ketika aktor menekan menu supplier.

#### 9. *Activity Diagram* Mengelola Data Supplier

*Activity diagram* mengelola data supplier merupakan penggambaran tentang proses yang terjadi ketika aktor akan mengubah maupun menambah data supplier. Sistem hanya akan menolak data yang dimasukkan ketika ada salah satu data yang belum diisi. *Penggambaran* lengkap *activity diagram* mengelola data supplier terdapat pada lampiran 3 gambar 3.7.

#### 10. *Activity Diagram* Melihat Data Transaksi

*Activity diagram* melihat data transaksi menjelaskan tentang proses ketika aktor akan melihat data transaksi. Aktor dapat menekan menu transaksi untuk melihat data transaksi. Pendeskripsian yang lebih jelas tentang *activity diagram* melihat data transaksi dapat dilihat pada Lampiran 3 gambar 3.8.

#### 11. *Activity Diagram* Menambah Data Transaksi

Lampiran 3 gambar 3.9 menjelaskan tentang *Activity diagram* menambah data transaksi. *Activity diagram* tersebut menggambarkan proses yang terjadi ketika aktor akan menambah data transaksi. Sistem akan menolak data yang dimasukkan ketika ada data yang belum terisi atau stok dalam gudang tidak cukup.

#### 12. *Activity Diagram* Menganalisa *Cross Selling Product*

*Activity diagram* menganalisa *cross selling product* mendeskripsikan tentang proses yang terjadi ketika aktor akan melakukan analisa *cross selling product*. Aktor dapat menggunakan fitur ini dengan menekan menu analisa penjualan. Setelah mengisi form analisa aktor dapat menekan tombol analisa untuk melihat hasil analisa produk penjualan. Penjelasan tentang *Activity diagram* menganalisa *cross selling product* dapat dilihat pada Lampiran 3 gambar 3.10.

#### 13. *Activity Diagram Logout*

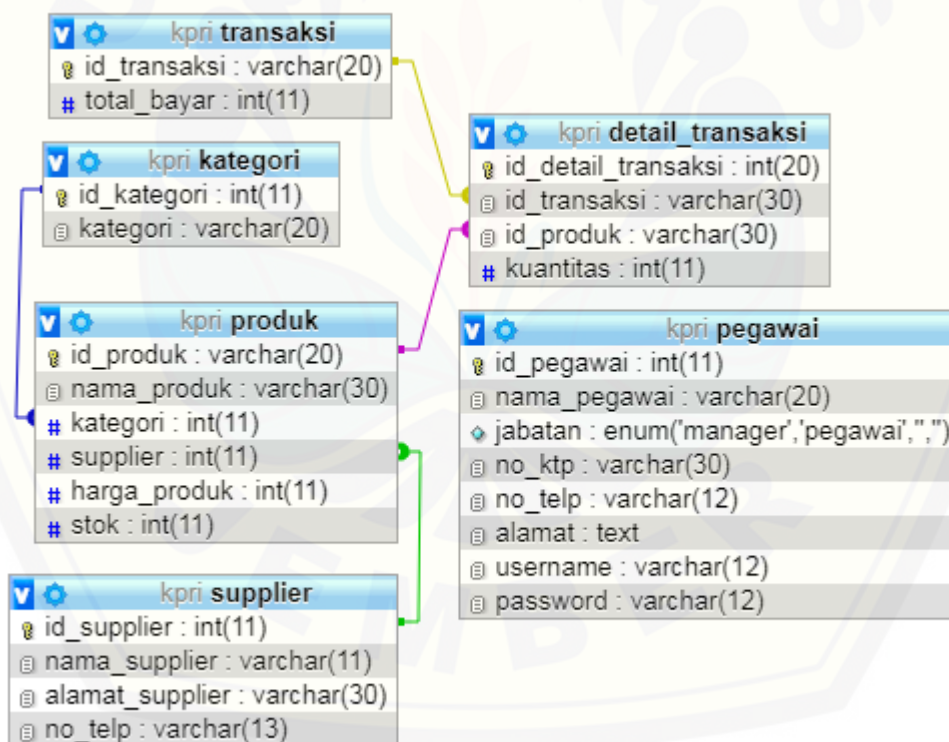
*Activity diagram logout* menggambarkan tentang proses yang terjadi ketika aktor akan keluar dari sistem. Aktor cukup menekan menu *logout* untuk keluar dari sistem. Sistem akan mengeluarkan aktor dan menampilkan halaman login ketika aktor menekan menu *logout*. *Activity diagram logout* dapat dilihat pada Lampiran 3 gambar 3.11.

#### 4.3.6 Class Diagram

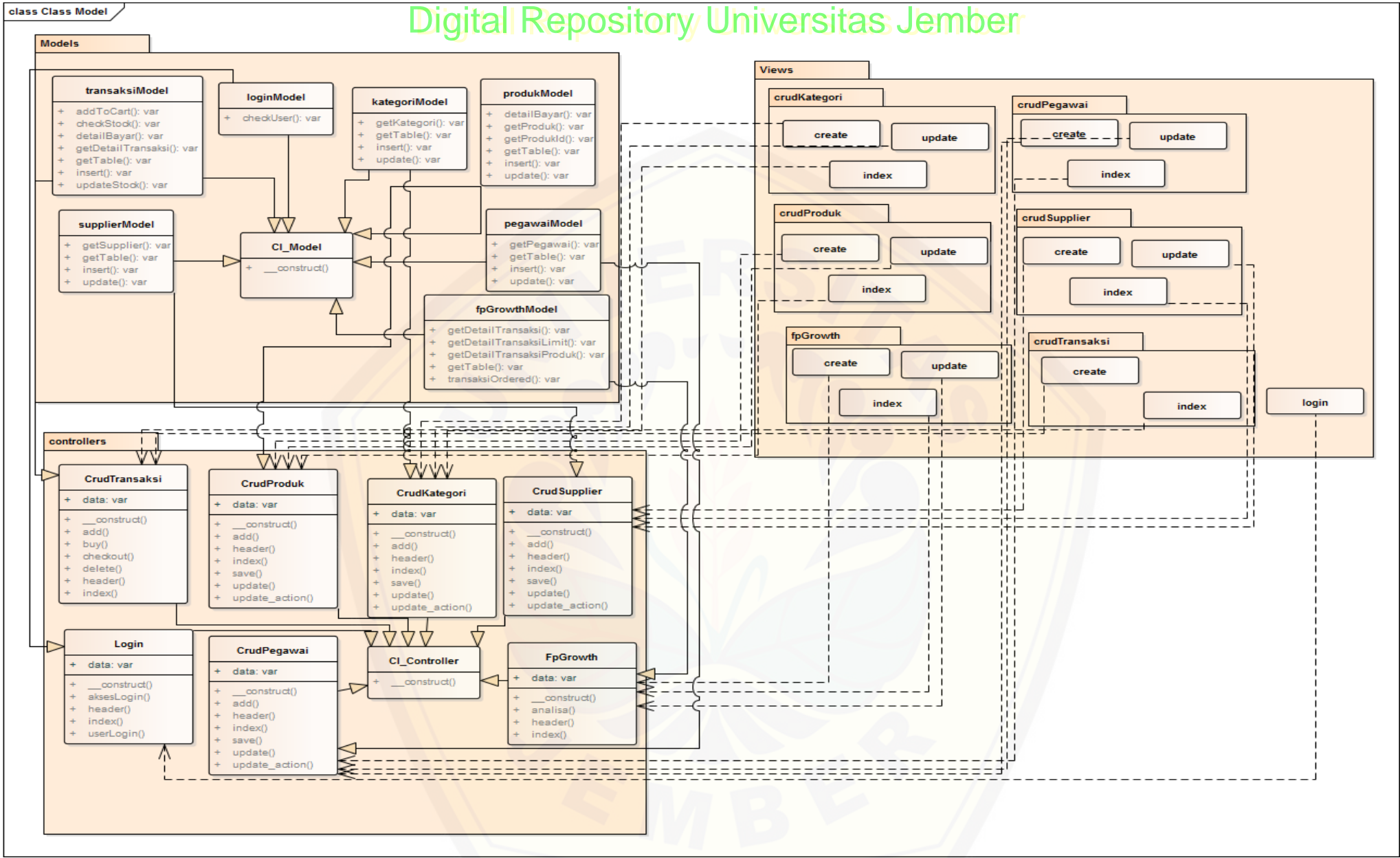
Class diagram menggambarkan kelas-kelas dan juga hubungan antar kelas pada sistem yang nanti akan dibangun. Dalam kelas ada tiga komponen yaitu nama, operasi dan atribut. Nama berfungsi untuk memberi identitas pada suatu kelas, atribut berfungsi untuk memberi karakteristik pada data yang dipunya, sedangkan operasi bertujuan untuk memberikan sebuah fungsi ke sebuah objek. Class diagram sistem informasi penentua paket pembelian produk KPRI Unej dapat dilihat pada Gambar 4.9.

#### 4.3.7 Entity Relationship Diagram

*Entity Relationship Diagram* (ERD) merupakan dokumentasi yang menggambarkan struktur dan komponen *database* yang digunakan dalam membuat suatu sistem. ERD sistem informasi penentuan paket pembelian produk KPRI Unej dapat dilihat pada Gambar 4.8



Gambar 4.8 *Entity Relationship Diagram*



Gambar 4. 9 Class Diagram



#### 4.4 Implementasi

Tahap implementasi berisi penulisan kode program hasil dari penerapan desain sistem yang telah dibuat. Bahasa pemrograman yang digunakan adalah bahasa pemrograman PHP dan menggunakan *DBMS MySQL*. Pada tahap implementasi perancangan ini menjelaskan fitur-fitur yang ada dalam sistem.

#### 4.5 Pengujian Program

##### 4.5.1 Pengujian *White Box*

Pengujian *white box* merupakan pengujian program meliputi grafik alir, listing program, kompleksitas *siklomatis* dan *test case*. Pengujian *white box testing* bertujuan untuk mengevaluasi apakah sistem dapat bekerja sesuai dengan yang diharapkan atau tidak. Tahapan *white box testing* meliputi :

1. *Listing program*

```

6     public function __construct()
7     {
8         parent::__construct();
9         //Load Dependencies
10        $this->Header();
11        $this->load->helper('form');
12        $this->load->helper('string');
13        $this->load->model('transaksiModel');
14    }

```

Gambar 4.10 Listing Program *Function \_\_construct*

```

16    public function index( )
17    {
18
19        $data['transaksi']      = $this->transaksiModel->getTable('transaksi');
20        $data['detailTransaksi'] = $this->transaksiModel->getDetailTransaksi();
21
22        $this->load->view('crudTransaksi/index', $data);
23    }

```

Gambar 4.11 Listing Program *Function index*

```

24    public function add()
25    {
26
27        $data['produk'] = $this->transaksiModel->getTable('produk');
28        $this->load->view('crudTransaksi/create', $data);
29    }

```

Gambar 4.12 Listing Program *Function add*

```

30    public function buy()
31    {
32
33        redirect('crudTransaksi/add');
34    }
35
36    // product -> produk,
37    // name -> (string) produk -> nama_produk
38    );
39    $this->cart->product_name_rules = '[:print:]';
40    $this->cart->insert($data);

```

Gambar 4. 13 Listing Program *Function buy*

```

55 public function checkOut()
56 {
57     $id = $this->transaksiModel->detailBayar();
58     $id2 = substr($id->id_transaksi, 0,9);
59     $id3 = (int)substr($id->id_transaksi, -5)+1;
60     $id4 = $id2 . $id3;
61     //cek apakah stock masih ada
62     foreach ($this->cart->contents() as $key) {
63         $check = $this->transaksiModel->checkStock($key['id']);
64         if ((int)$check->stok <= $key['qty']) {
65             $message = "kuantitas $check->nama_produk melebihi stok";
66             $this->session->set_flashdata('flsh_msg', $message);
67             $this->session->set_flashdata('alert', 'alert alert-danger');
68             redirect('crudTransaksi/add');
69         }
70     }
71     $data = array(
72         'id_transaksi' => $id4,
73         'total_bayar' => $this->cart->total()
74     );
75     $this->transaksiModel->insert('transaksi',$data);
76     foreach ($this->cart->contents() as $key) {
77         $data = array(
78             'id_transaksi' => $id4,
79             'id_produk' => $key['id'],
80             'kuantitas' => $key['qty']
81         );
82         $this->transaksiModel->insert('detail_transaksi',$data);
83     }
84     //update stok produk sesuai barang yang keluar
85     foreach ($this->cart->contents() as $key) {
86         $id_produk = $key['id'];
87         $kuantitas = $key['qty'];
88         $this->transaksiModel->updateStock($id_produk,$kuantitas);
89     }
90     $this->cart->destroy();
91     $this->session->set_flashdata('flsh_msg', 'Data transaksi berhasil dimasukkan');
92     $this->session->set_flashdata('alert', 'alert alert-success');
93     redirect('crudTransaksi/add');
94 }
95 }
96

```

Gambar 4.14 Listing Program *Function checkOut*

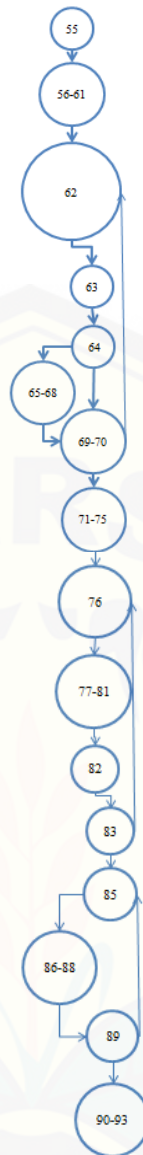
## 2. Diagram Alir

Diagram alir merupakan notasi-notasi sederhana yang menunjukkan alur kontrol. Alur kontrol yang digambarkan merupakan hasil dari penomoran listing program. Diagram alir digambarkan dengan simpul yang dihubungkan dengan garis.

Gambar 4. 11 Diagram Alir *Function \_\_construct*



Gambar 4.12 Diagram *Function index*Gambar 4.16 Diagram *Function add*Gambar 4.17 Diagram *Function buy*



Gambar 4.18 Diagram *Function checkout*

### 3. Cyclomatic Complexity

*Cyclomatic complexity* atau *kompleksitas siklomatis* merupakan matriks untuk mengetahui kompleksitas logika suatu program. Perhitungan kompleksitas siklomatis dari listing program adalah sebagai berikut :

#### 3.1 *Function \_\_construct()*

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

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

#### 3.2 *Function index*

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

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

### 3.3 Function add

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

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

### 3.4 Function buy

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

$$V(G) = 7 - 8 + 2 = 1$$

### 3.5 Function checkout

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

$$V(G) = 18 - 16 + 2 = 4$$

## 4. Test Case

Tabel 4. 7 Test Case \_\_construct

Test Case	Jika <i>function __construct</i> dijalankan
Target yang Diharapkan	Berhasil menampilkan halaman utama
Hasil Pengujian	Benar, halaman utama berhasil ditampilkan
Jalur	6, 7-13, 14

Tabel 4. 8 Test Case index

Test Case	Jika <i>function index</i> dijalankan
Target yang Diharapkan	Berhasil mengambil data dengan memanggil <i>function getTable</i> dan <i>getTableTransaksi</i> pada class <i>transaksiModel</i> dalam bentuk array <i>\$data</i> dan menampilkannya dalam class <i>views</i>
Hasil Pengujian	Benar, benar data transaksi yang tersimpan dalam database berhasil ditampilkan di halaman view
Jalur	16,17-22,23

Tabel 4. 9 Test Case add

Test Case	Jika <i>function add</i> dijalankan
Target yang Diharapkan	Berhasil mengambil data dengan memanggil <i>function getTable</i> pada class <i>transaksiModel</i> dalam bentuk array <i>\$data</i> dan menampilkannya dalam class <i>views</i>
Hasil Pengujian	Benar, data produk berhasil ditampilkan dalam field produk
Jalur	24,25-28,29

Tabel 4. 10 Test Case buy

Test Case	Jika <i>function buy</i> dijalankan
Inputan	Nama produk
Target yang Diharapkan	Berhasil mengambil data dengan memanggil <i>function addToCart</i> pada class <i>transaksiModel</i> dalam bentuk array <i>\$product</i> , memasukkan data ke array <i>\$data</i> dan kemudian memasukkan isi dari array <i>\$data</i> ke dalam session cart untuk ditampilkan dalam class <i>views</i>
Hasil Pengujian	Benar, data produk berhasil dimasukkan ke keranjang
Jalur	30,31,32,33-38,39,40,42,43

Tabel 4. 11 Test Case Function checkOut

Test Case	Jika <i>function checkout</i> dijalankan
Inputan	Data produk yang sudah ada dalam keranjang
Target yang Diharapkan	Berhasil mengambil data stock barang dengan memanggil <i>function checkstock</i> dalam <i>class transaksiModel</i> untuk memeriksa ketersediaan stock, kemudian menyimpan data penjualan yang tersimpan dalam array \$data kedalam database dengan memanggil <i>function insert</i> dalam <i>class transaksiModel</i>
Hasil Pengujian	Benar, data yang ada dalam keranjang berhasil dimasukkan ke dalam database
Jalur	55,56-61,62-64,69-70,71-89,90-93

#### 4.5.2 Pengujian *Black Box*

Pengujian black box digunakan untuk menguji sistem dari segi fungsional sistem untuk mengetahui apaakah inputan, fungsi-fungsi, dan keluaran sistem sesuai dengan yang dibutuhkan pengguna. Hasil pengujian sistem dengan metode *black box* dapat dilihat pada lampiran 4.

## BAB 6 PENUTUP

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

### 6.1 Kesimpulan

Kesimpulan dari penelitian yang telah dilakukan adalah:

1. Algoritma *Fp-Growth* dapat diimplementasikan dengan memasukkan beberapa variable seperti *minimum confidence*, *minimum support* dan jumlah transaksi yang akan dianalisa. Data paket pembelian produk yang dihasilkan dari algoritma *Fp-Growth* dapat dijadikan acuan sebagai strategi penjualan dengan memberikan pilihan paket pembelian pada konsumen. Merujuk pada pembahasan tentang korelasi antara data yang dimasukkan ke dalam sistem dengan data yang dihasilkan pada bab pembahasan membuktikan bahwa banyaknya paket pembelian produk yang dihasilkan dipengaruhi oleh besarnya nilai *minimum support* dan *minimum confidence* yang dimasukkan. Semakin besar nilai *minimum support* dan *minimum confidence* yang dimasukkan maka paket pembelian produk yang terbentuk akan semakin sedikit.
2. Pengembangan sistem informasi penentuan paket pembelian produk dibangun dengan basis *website* dengan dua hak akses yaitu pegawai dan manajer. Manajer memiliki akses terhadap fitur utama yaitu menganalisa data penjualan untuk menentukan paket pembelian produk. Sistem dikembangkan dengan mengadopsi *SDLC Waterfall*. Model waterfall sangat cocok untuk mengembangkan sistem ini karena memiliki proses yang urut dan juga data yang dibutuhkan sudah memiliki rincian yang tidak berubah dari awal pembuatan sampai penyelesaian program. Data yang dibutuhkan oleh sistem adalah data transaksi, data *minimum support* dan data *minimum confidence* sebagai data masukan untuk menentukan paket pembelian produk.

## 6.2 Saran

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

1. Pemanfaatan perhitungan algoritma *Fp-growth* pada sistem ini hanya sampai pada proses penentuan paket pembelian produk, namun akan lebih baik jika terdapat fitur yang bisa melacak keefektifan paket pembelian produk yang dihasilkan sehingga paket pembelian produk yang dihasilkan dapat dipantau keberhasilannya.
2. Proses penentuan paket pembelian produk dapat dilakukan dengan menggunakan metode atau algoritma lain dengan objek penelitian yang sama. Penentuan paket pembelian produk dengan menggunakan algoritma *Fp-Growth* juga dapat lebih dikembangkan dengan menggabungkan dengan metode lain yang saling mendukung.



**DAFTAR PUSTAKA**

- (2017, 08 14). Retrieved from Waterfall Model: <http://www.waterfall-model.com/>
- Erwin. (2009). Analisis Market Basket Dengan Algoritma Apriori dan FP-Growth. *Jurnal Generic*.
- Gunaidi, G., & Sensue, D. I. (2012). Penerapan Metode Data Mining Market Basket Analysis Terhadap Data Penjualan Produk Buku dengan Menggunakan Algoritma Apriori dan Frequent Pattern Growth (FP-GROWTH) Studi Kasus Percetakan PT. Gramedia.
- Harsoyo, Y. (2006). *Ideologi koperasi menatap masa depan*. tanggerang: Pustaka Widyatama.
- Kamakura, W., Ramaswami, S., & R. K., S. (1991.). Applying Latern Analysis in Evaluation of Prospects for Cross-selling of Financial Services.
- Marakas, M. (2003). Modern Data Warehousing, Mining, And Visualization.
- Pramudiono, I. (2003). Menambang Permata Pengetahuan di Gunung Data.
- Stephanie, K. (n.d.). *Cara Menggunakan Cross-Selling Dan Upselling Untuk Memaksimalkan Penjualan Online*. Retrieved from koinworks.com: <https://koinworks.com/blog/cross-selling-dan-upselling-memaksimalkan-penjualan-online/>
- Sugiyanto, R. S. (2015). Analisis Pengaruh Kualitas Layanan Terhadap Loyalitas Konsumen Mahasiswa Pada Koperasi Pegawai Republik Indonesia (KPRI) Universitas Jember.
- Syaifullah, M. A. (2010). *Implementasi Data Mining Algoritma Apriori Pada Sistem Penjualan*. Yogyakarta: STMIK AMIKOM.
- Tama, B. A. (2010). Penetapan Strategi Penjualan Menggunakan Association rules dalam konteks CRM.

Turban, E. A. (2005). *Sistem Pendukung Keputusan dan Sistem Cerdas*. Yogyakarta: Andi.

Warni, S. (2016, 01 13). *Manfaat Database Pelanggan*. Retrieved from zahiraccounting.com: <https://zahiraccounting.com/id/blog/manfaat-database-pelanggan/>



## LAMPIRAN

## LAMPIRAN 1 (Skenario Sistem)

## 1.1 Skenario Login

<b>No Usecase</b>	UCS 001
<b>Nama Usecase</b>	Login
<b>Aktor</b>	All Aktor
<b>Deskripsi</b>	Aktor login untuk mengakses system
<b>Prekondisi</b>	Aktor belum melakukan login
<b>Prakondisi</b>	Aktor berhasil login
<b>Flow Of Events</b>	
<b>Skenario Normal : Login</b>	
<b>Aktor</b>	<b>Sitem</b>
1. Membuka Sistem Pejualan Kpri	2. Menampilkan halaman login Sistem Penjualan Kpri
3. Masukan Username dan Password	
4. Klik tombol Sign in	
	5. Menampilkan halaman utama sistem
	6. Menampilkan Alert "Login Berhasil! Selamat Datang"
<b>Skenario Alternatif : Username atau Password Salah</b>	
<b>Aktor</b>	<b>Sitem</b>
4. Klik tombol Sign in	
	5. Menampilkan Alert "Username atau Password yang anda masukan salah.
	6. Menampilkan halaman login Sistem

## 1.2 Skenario Melihat Data Barang

<b>No Usecase</b>	UCS 005
<b>Nama Usecase</b>	Melihat data barang
<b>Aktor</b>	All Aktor
<b>Deskripsi</b>	Aktor dapat melihat data barang
<b>Prekondisi</b>	Aktor telah masuk system
<b>Prakondisi</b>	Aktor berhasil melihat data barang
<b>Flow Of Events</b>	
<b>Skenario Normal : Melihat data barang</b>	
<b>Aktor</b>	<b>Sitem</b>
1. Klik menu Produk	
	2. Menampilkan daftar produk

## 1.3 Skenario Menambah Data Barang

<b>No Usecase</b>	UCS 006
<b>Nama Usecase</b>	Menambah data barang
<b>Aktor</b>	Produk
<b>Deskripsi</b>	Aktor dapat menambah data barang
<b>Prekondisi</b>	Aktor telah masuk system
<b>Prakondisi</b>	Aktor berhasil menambah data barang
<b>Skenario Normal : Menambah Data Barang</b>	
<b>Aktor</b>	<b>Sitem</b>
1. Klik menu produk	
	2. Menampilkan daftar produk
3. Klik tombol add	
	4. Menampilkan form pengisian data produk
5. Mengisi form data produk	
6. Klik tombol Save	
	7. Menampilkan notifikasi “ pengisian data berhasil.”
	8. Menampilkan daftar produk
<b>Skenario Alternatif : Data Tidak lengkap</b>	
<b>Aktor</b>	<b>Sitem</b>
5.a Mengisi form data produk	
6.a Klik tombol Save	
	7.a Menampilkan notifikasi “Please fill out this field” pada form produk
<b>Skenario Alternatif : Harga/stok kurang dari 0</b>	
<b>Aktor</b>	<b>Sitem</b>
5.b Mengisi form data produk	
6.b Klik tombol Save	
	7.b Menampilkan notifikasi “Data masukan harus lebih dari 0”
	8.b Menampilkan form pengisian data produk

## 1.4 Skenario Mengubah Data Barang

<b>No Usecase</b>	UCS 007
<b>Nama Usecase</b>	Mengubah data barang
<b>Aktor</b>	Produk
<b>Deskripsi</b>	Aktor dapat mengubah data barang
<b>Prekondisi</b>	Aktor telah masuk system
<b>Prakondisi</b>	Aktor berhasil mengubah data barang
<b>Skenario Normal : Mengubah Data Barang</b>	
<b>Aktor</b>	<b>Sitem</b>
1. Klik menu produk	
	2. Menampilkan daftar produk
3. Klik tombol Update	
	4. Menampilkan form pengisian data produk
5. Mengisi form data produk	
6. Klik tombol Save	
	7. Menampilkan notifikasi “ pengisian data berhasil.”
	8. Menampilkan daftar produk
<b>Skenario Alternatif : Harga/stok kurang dari 0</b>	
<b>Aktor</b>	<b>Sitem</b>
5.a Mengisi form data produk	
6.a Klik tombol Save	
	7.a Menampilkan notifikasi “Data masukan harus lebih dari 0”
	8.a Menampilkan form pengisian data produk
<b>Skenario Alternatif : Data Tidak lengkap</b>	
<b>Aktor</b>	<b>Sitem</b>
5.b Mengisi form data produk	
6.b Klik tombol Save	
	7.b Menampilkan notifikasi “Please fill out this field” pada form produk

## 1.5 Skenario Melihat Data Kategori

<b>No Usecase</b>	UCS 008
<b>Nama Usecase</b>	Melihat data kategori
<b>Aktor</b>	All Aktor
<b>Deskripsi</b>	Aktor dapat melihat data kategori
<b>Prekondisi</b>	Aktor telah masuk system
<b>Prakondisi</b>	Aktor berhasil melihat data kategori
<b>Flow Of Events</b>	
<b>Skenario Normal : Melihat data kategori</b>	
<b>Aktor</b>	<b>Sitem</b>
1. Klik menu kategori	
	2. Menampilkan daftar kategori

## 1.6 Skenario Menambah Data Kategori

<b>No Usecase</b>	UCS 009
<b>Nama Usecase</b>	Menambah data kategori
<b>Aktor</b>	Kategori
<b>Deskripsi</b>	Aktor dapat menambah data kategori
<b>Prekondisi</b>	Aktor telah masuk system
<b>Prakondisi</b>	Aktor berhasil menambah data kategori
<b>Skenario Normal : Menambah Data Kategori</b>	
<b>Aktor</b>	<b>Sitem</b>
1. Klik menu kategori	
	2. Menampilkan daftar kategori
3. Klik tombol add	
	4. Menampilkan form pengisian data kategori
5. Mengisi form data kategori	
6. Klik tombol Save	
	7. Menampilkan notifikasi “ pengisian data berhasil.”
	8. Menampilkan daftar kategori
<b>Skenario Alternatif : Data Tidak lengkap</b>	
<b>Aktor</b>	<b>Sitem</b>
5.a Mengisi form data kategori	
6.a Klik tombol Save	
	7.a Menampilkan notifikasi “Please fill out this field” pada form kategori



## 1.7 Skenario Mengubah Data Kategori

<b>No Usecase</b>	UCS 010
<b>Nama Usecase</b>	Mengubah data kategori
<b>Aktor</b>	Kategori
<b>Deskripsi</b>	Aktor dapat mengubah data kategori
<b>Prekondisi</b>	Aktor telah masuk system
<b>Prakondisi</b>	Aktor berhasil mengubah data kategori
<b>Skenario Normal : Mengubah Data Kategori</b>	
<b>Aktor</b>	<b>Sitem</b>
1. Klik menu kategori	
	2. Menampilkan daftar kategori
3. Klik tombol Update	
	4. Menampilkan form pengisian data kategori
5. Mengisi form data kategori	
6. Klik tombol Save	
	7. Menampilkan notifikasi “ pengisian data berhasil.”
	8. Menampilkan daftar kategori
<b>Skenario Alternatif : Data Tidak lengkap</b>	
<b>Aktor</b>	<b>Sitem</b>
5.a Mengisi form data kategori	
6.a Klik tombol Save	
	8.a Menampilkan notifikasi “Please fill out this field” pada form kategori

## 1.8 Skenario Melihat Data Supplier

<b>No Usecase</b>	UCS 011
<b>Nama Usecase</b>	Melihat data supplier
<b>Aktor</b>	All Aktor
<b>Deskripsi</b>	Aktor dapat melihat data supplier
<b>Prekondisi</b>	Aktor telah masuk system
<b>Prakondisi</b>	Aktor berhasil melihat data supplier
<b>Flow Of Events</b>	
<b>Skenario Normal : Melihat data supplier</b>	
<b>Aktor</b>	<b>Sitem</b>
1. Klik menu supplier	
	2. Menampilkan daftar supplier

## 1.9 Skenario Menambah Data Supplier

<b>No Usecase</b>	UCS 012
<b>Nama Usecase</b>	Menambah data supplier
<b>Aktor</b>	Pegawai
<b>Deskripsi</b>	Aktor dapat menambah data supplier
<b>Prekondisi</b>	Aktor telah masuk system
<b>Prakondisi</b>	Aktor berhasil menambah data supplier
<b>Skenario Normal : Menambah Data Supplier</b>	
<b>Aktor</b>	<b>Sitem</b>
1. Klik menu supplier	

	2. Menampilkan daftar supplier
3. Klik tombol add	
	4. Menampilkan form pengisian data supplier
5. Mengisi form data supplier	
6. Klik tombol Save	
	7. Menampilkan notifikasi “ pengisian data berhasil.”
	8. Menampilkan daftar supplier
<b>Skenario Alternatif : Data Tidak lengkap</b>	
<b>Aktor</b>	<b>Sitem</b>
5.a Mengisi form data supplier	
6.a Klik tombol Save	
	7.a Menampilkan notifikasi “Please fill out this field” pada form supplier

### 1.10 Skenario Mengubah Data Supplier

<b>No Usecase</b>	UCS 013
<b>Nama Usecase</b>	Mengubah data supplier
<b>Aktor</b>	Pegawai
<b>Deskripsi</b>	Aktor dapat mengubah data supplier
<b>Prekondisi</b>	Aktor telah masuk system
<b>Prakondisi</b>	Aktor berhasil mengubah data supplier
<b>Skenario Normal : Mengubah Data Supplier</b>	
<b>Aktor</b>	<b>Sitem</b>
1. Klik menu supplier	
	2. Menampilkan daftar supplier
3. Klik tombol Update	
	4. Menampilkan form pengisian data supplier
5. Mengisi form data supplier	
6. Klik tombol Save	
	7. Menampilkan notifikasi “ pengisian data berhasil.”
	8. Menampilkan daftar supplier
<b>Skenario Alternatif : Data Tidak lengkap</b>	
<b>Aktor</b>	<b>Sitem</b>
5.a Mengisi form data supplier	
6.a Klik tombol Save	
	7.a Menampilkan notifikasi “Please fill out this field” pada form supplier

### 1.11 Skenario Melihat Data Transaksi

<b>No Usecase</b>	UCS 014
<b>Nama Usecase</b>	Melihat data transaksi
<b>Aktor</b>	Manager
<b>Deskripsi</b>	Aktor dapat melihat data transaksi
<b>Prekondisi</b>	Aktor telah masuk system
<b>Prakondisi</b>	Aktor berhasil melihat data transaksi

Flow Of Events	
Skenario Normal : Melihat data tranksaksi	
Aktor	Sitem
1. Klik menu tranksaksi	
	2. Menampilkan daftar tranksaksi

### 1.12 Skenario Menambah Data Transaksi

<b>No Usecase</b>	UCS 015
<b>Nama Usecase</b>	Menambah data tranksaksi
<b>Aktor</b>	Pegawai
<b>Deskripsi</b>	Aktor dapat menambah data tranksaksi
<b>Prekondisi</b>	Aktor telah masuk system
<b>Prakondisi</b>	Aktor berhasil menambah data tranksaksi
Skenario Normal : Menambah Data Tranksaksi	
Aktor	Sitem
1. Klik menu tranksaksi	
	2. Menampilkan daftar transaksi
3. Klik tombol add	
	4. Menampilkan form pengisian data transaksi
5. Mengisi form data supplier	
6. Klik tombol beli	
	7. Menampilkan daftar produk yang dibeli
8. Klik bayar	
	9. Menampilkan notifikasi “ transaksi berhasil dimasukkan”
	10. Menampilkan form pengisian data transaksi
Skenario Alternatif : Data Tidak lengkap	
Aktor	Sitem
5.a Mengisi form data transaksi	
6.a Klik tombol Save	
	7.a Menampilkan notifikasi “Please fill out this field” pada form tranksaksi
Skenario Alternatif : Stok Tidak Mencukupi	
Aktor	Sitem
5.a Mengisi form data transaksi	
6.a Klik tombol Save	
	7.a Menampilkan notifikasi “Kuantitas produk melebihi stok gudang” pada form tranksaksi
	8.a Menampilkan form pengisian data transaksi

### 1.13 Skenario Menganalisa Cross Selling Product

<b>No Usecase</b>	UCS 016
<b>Nama Usecase</b>	Menganalisa cross selling product
<b>Aktor</b>	Manager
<b>Deskripsi</b>	Aktor dapat melihat analisa cross seling

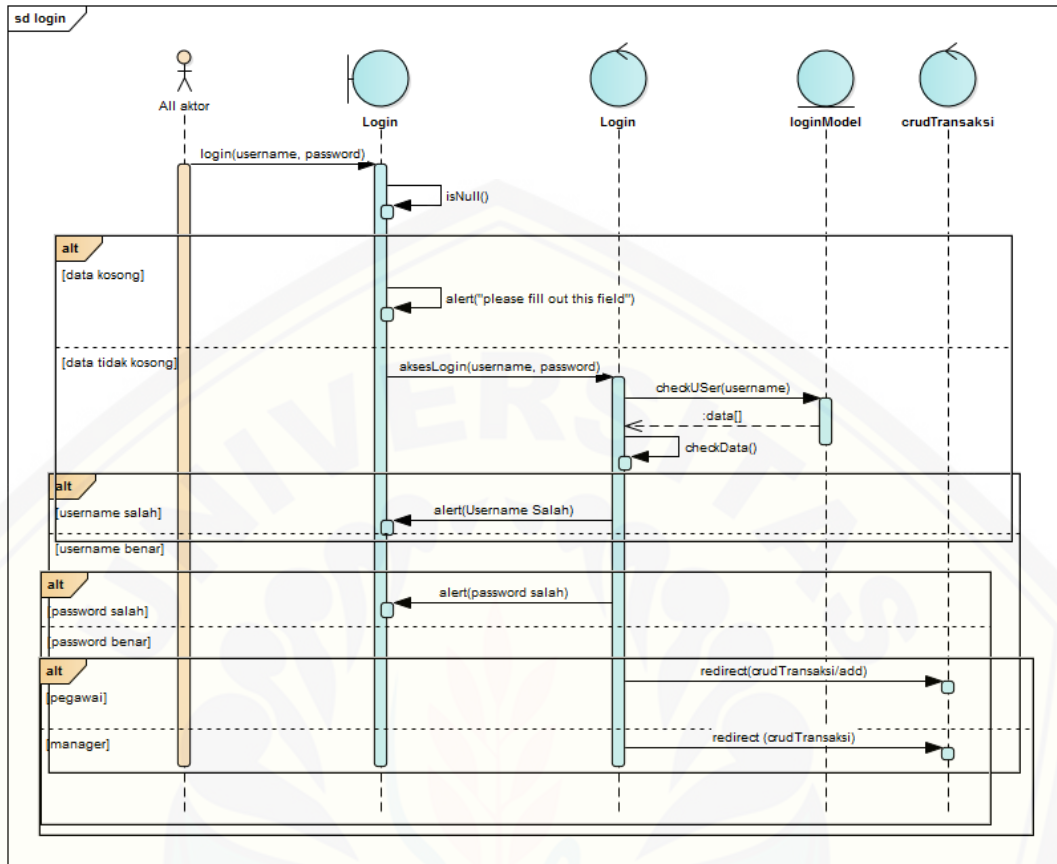
	produk
<b>Prekondisi</b>	Aktor telah masuk system
<b>Prakondisi</b>	Aktor berhasil melihat analisa cross seling produk
<b>Skenario Normal : Menganalisa Cross Selling Produk</b>	
<b>Aktor</b>	<b>Sitem</b>
1. Klik menu analisa penjualan	
	2. Menampilkan form pengisian data analisa cross selling produk
3. Mengisi form data analisa penjualan	
4. Klik tombol Analisa	
	5. Menampilkan daftar Cross Selling Produk
<b>Skenario Alternatif : Data Tidak lengkap</b>	
<b>Aktor</b>	<b>Sitem</b>
3.a Mengisi form data transaksi	
4.a Klik tombol Save	
	5.a Menampilkan notifikasi “Please fill out this field” pada form tranksaksi

#### 1.14 Skenario Log Out

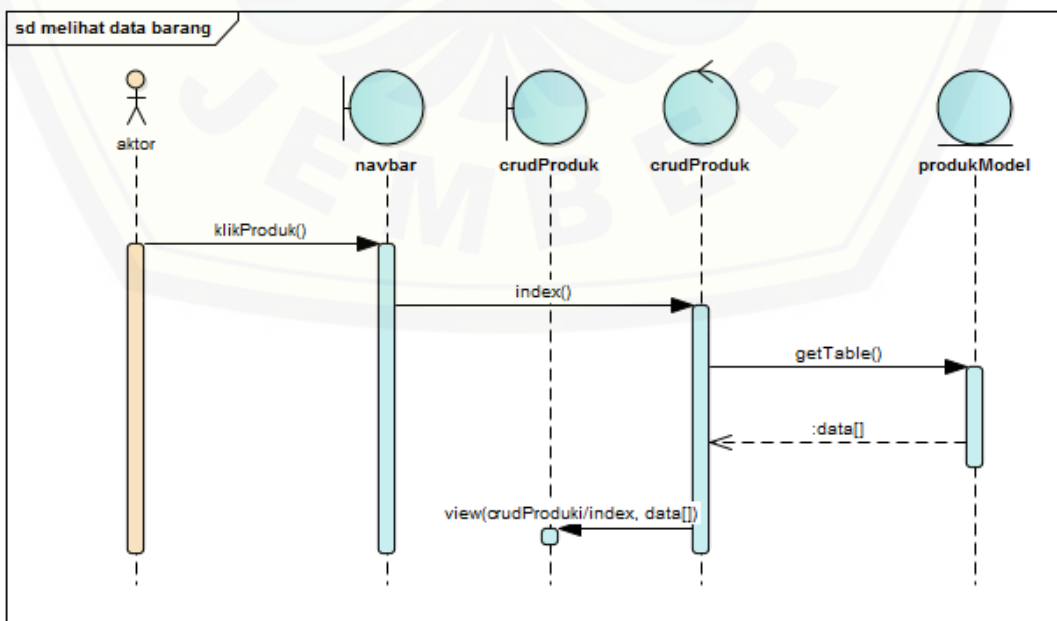
<b>No Usecase</b>	UCS 016
<b>Nama Usecase</b>	Log Out
<b>Aktor</b>	All actor
<b>Deskripsi</b>	Aktor dapat keluar dari system
<b>Prekondisi</b>	Aktor telah masuk system
<b>Prakondisi</b>	Aktor berhasil keluar sistem
<b>Skenario Normal : Menganalisa Cross Selling Supplier</b>	
<b>Aktor</b>	<b>Sitem</b>
1. Klik menu Log Out	
	2. Menampilkan halaman login

LAMPIRAN 2 (Sequence Diagram)

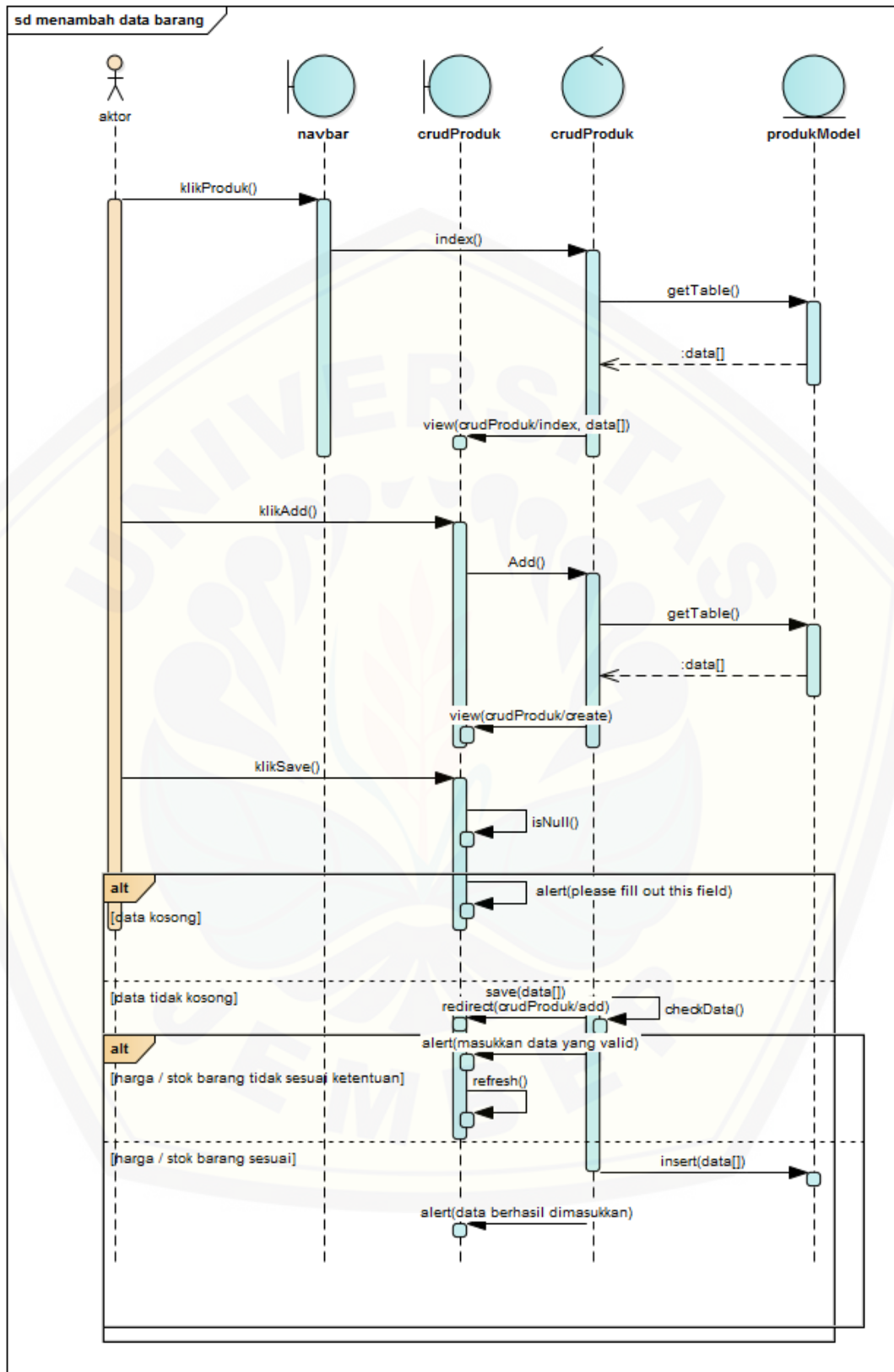
2.1 Sequence Diagram Login



2.2 Sequence Diagram Melihat Data Barang



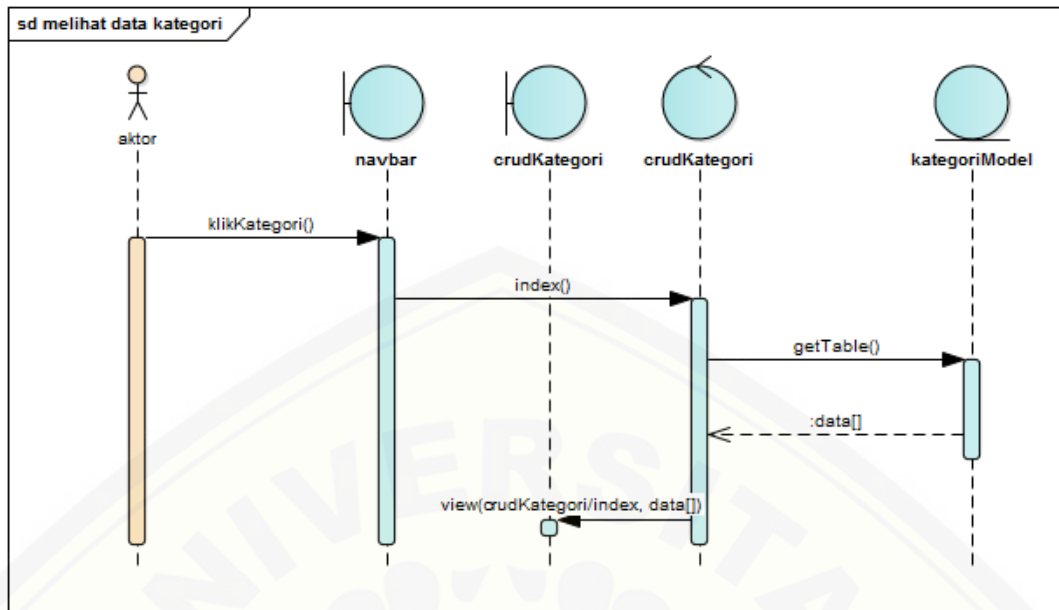
2.3 Sequence Diagram Menambah Data Barang



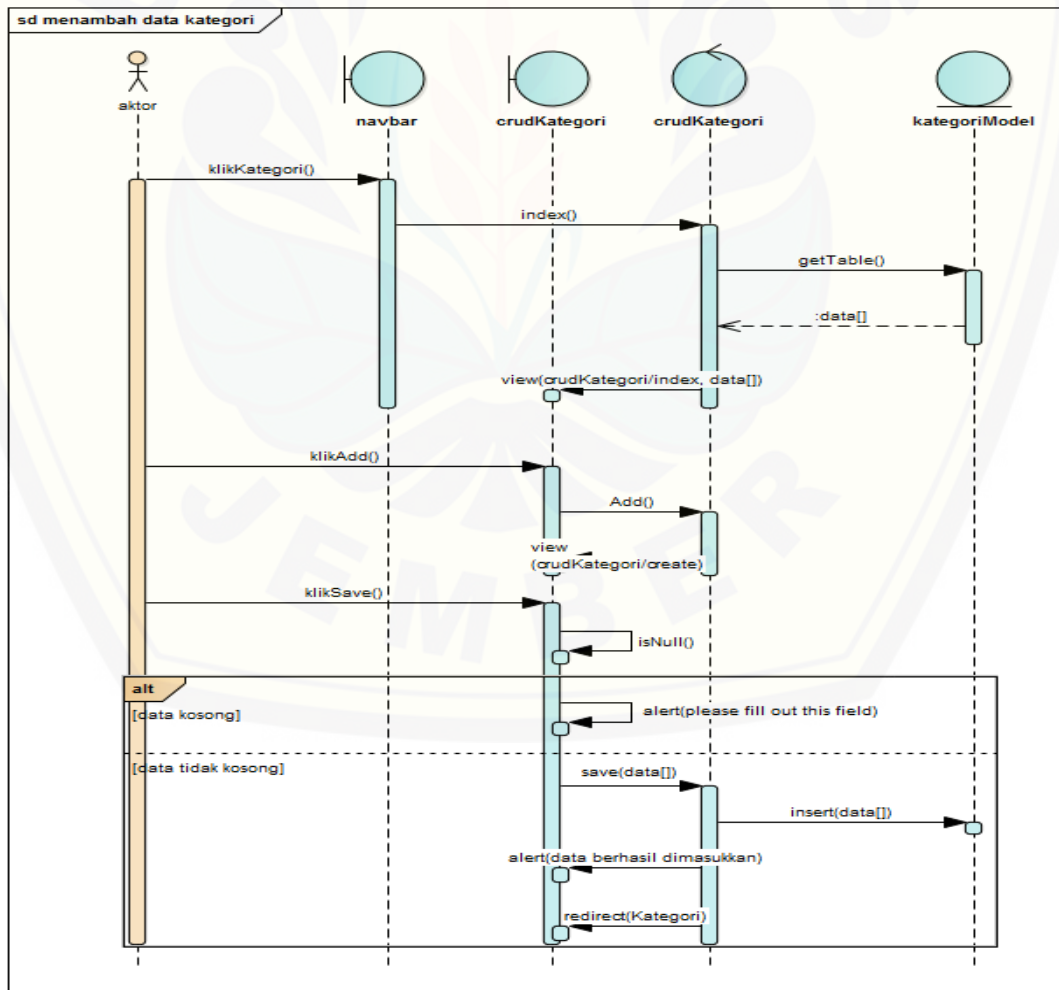




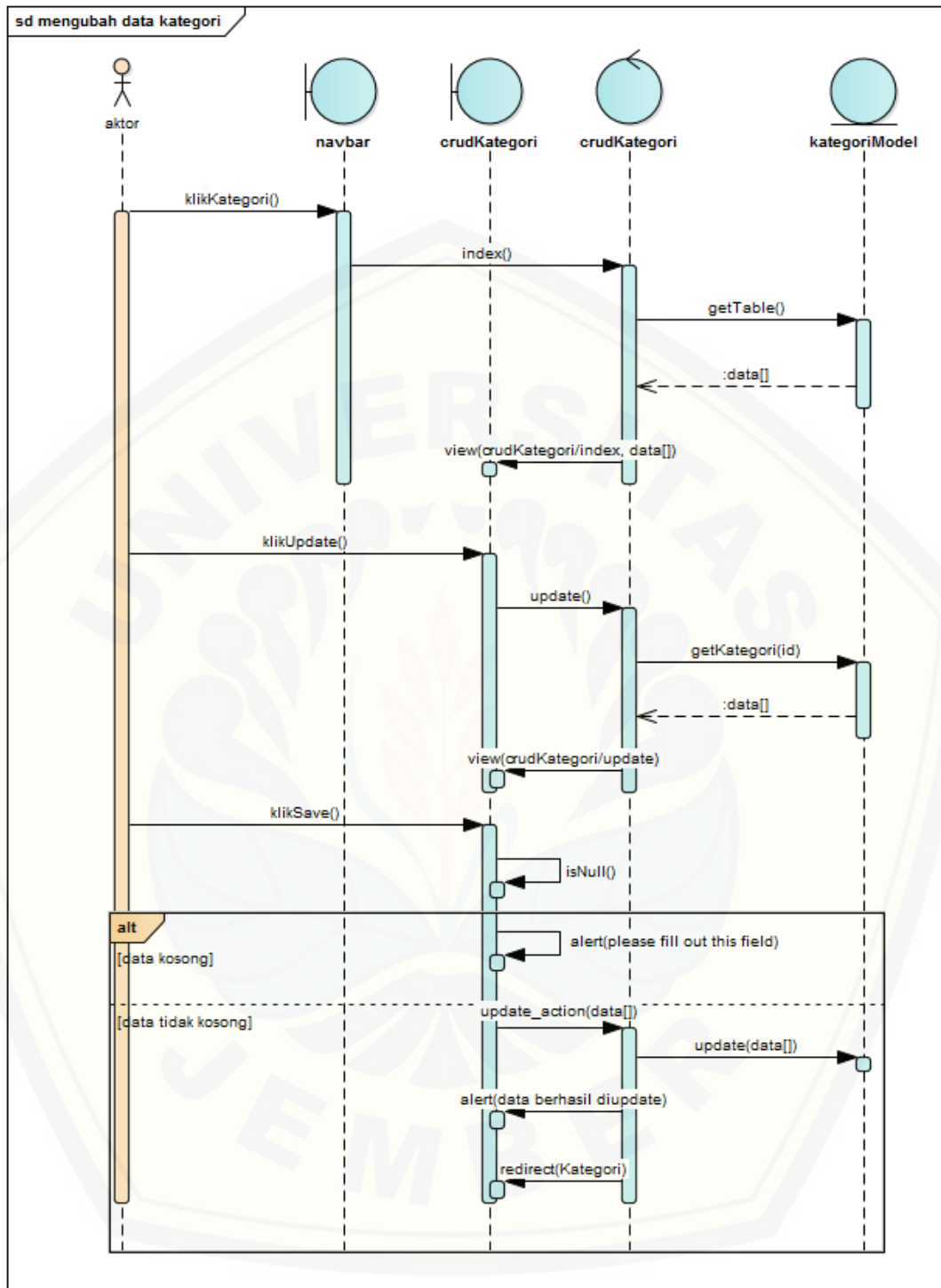
2.5 Sequence Diagram Melihat Data Kategori



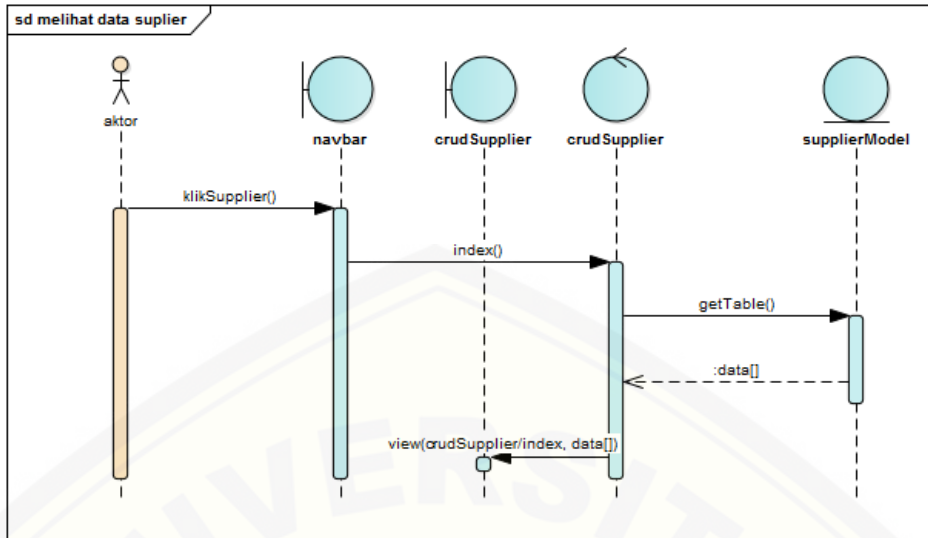
2.6 Sequence Diagram Menambah Data Kategori



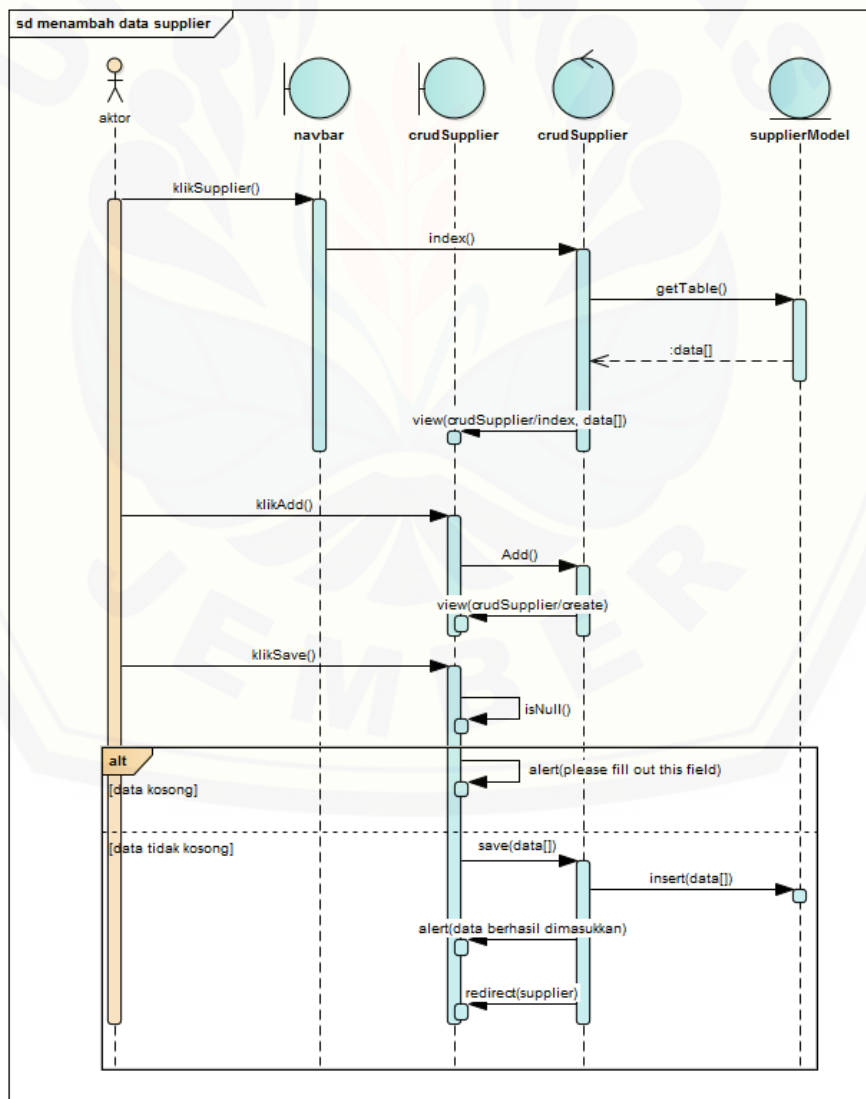
2.7 Sequence Diagram Mengubah Data Kategori



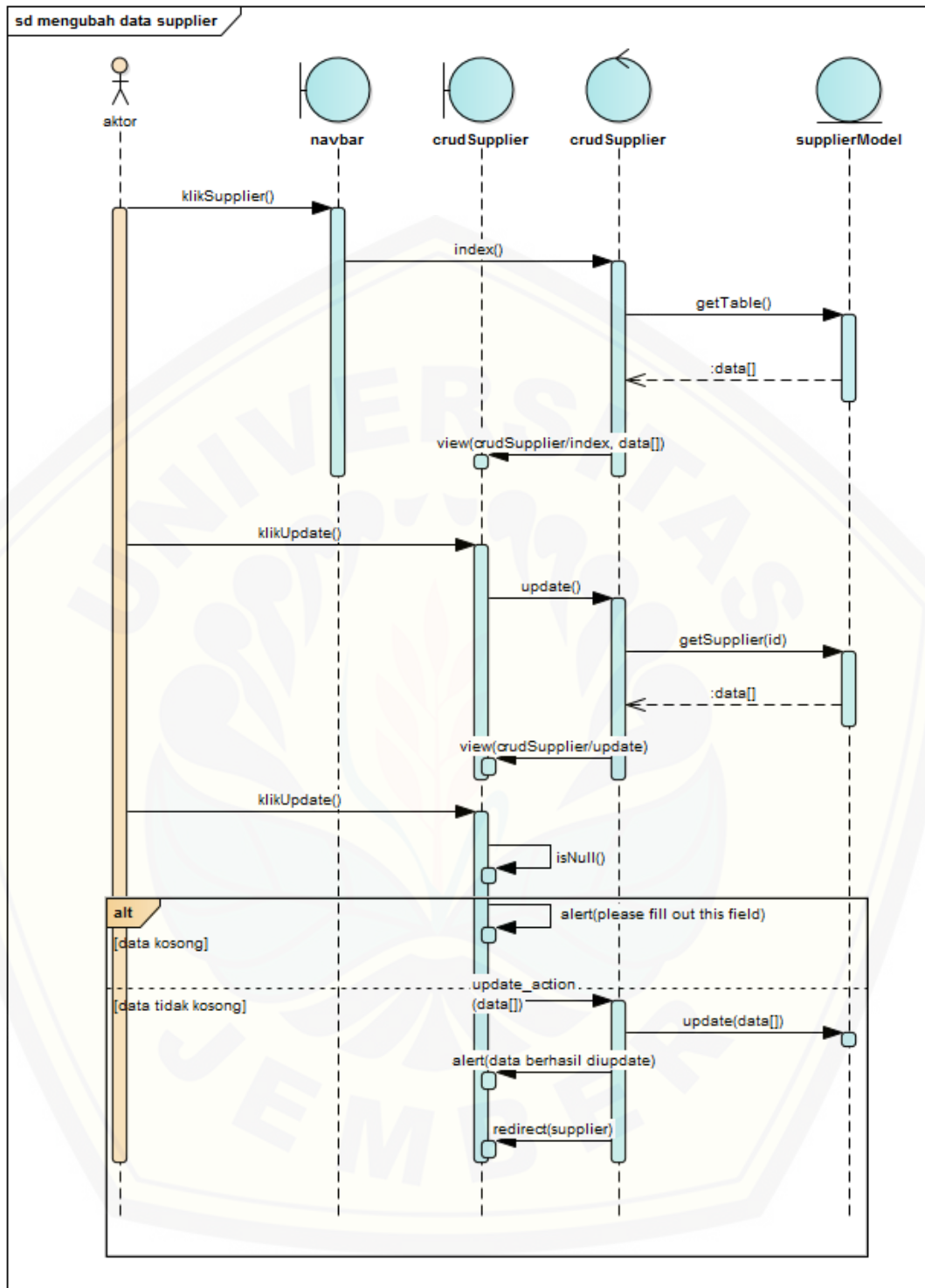
2.8 Sequence Diagram Melihat Data Supplier



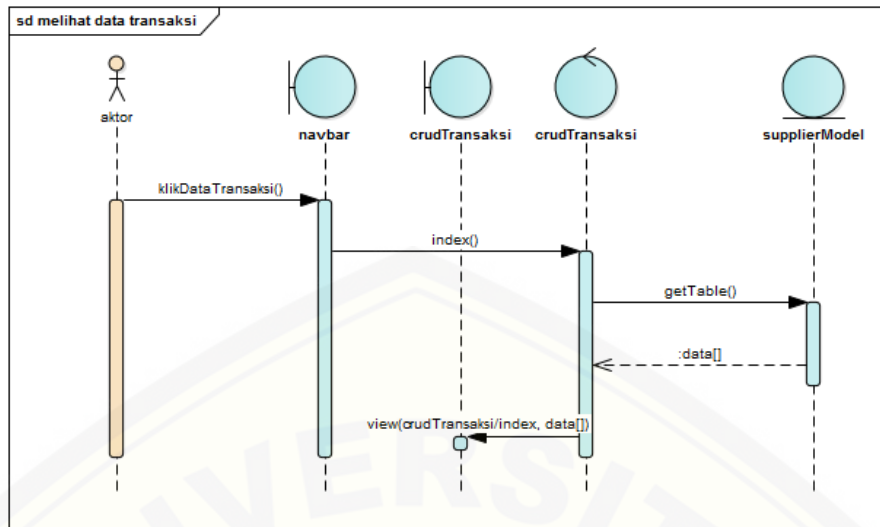
2.9 Sequence Diagram Menambah Data Supplier



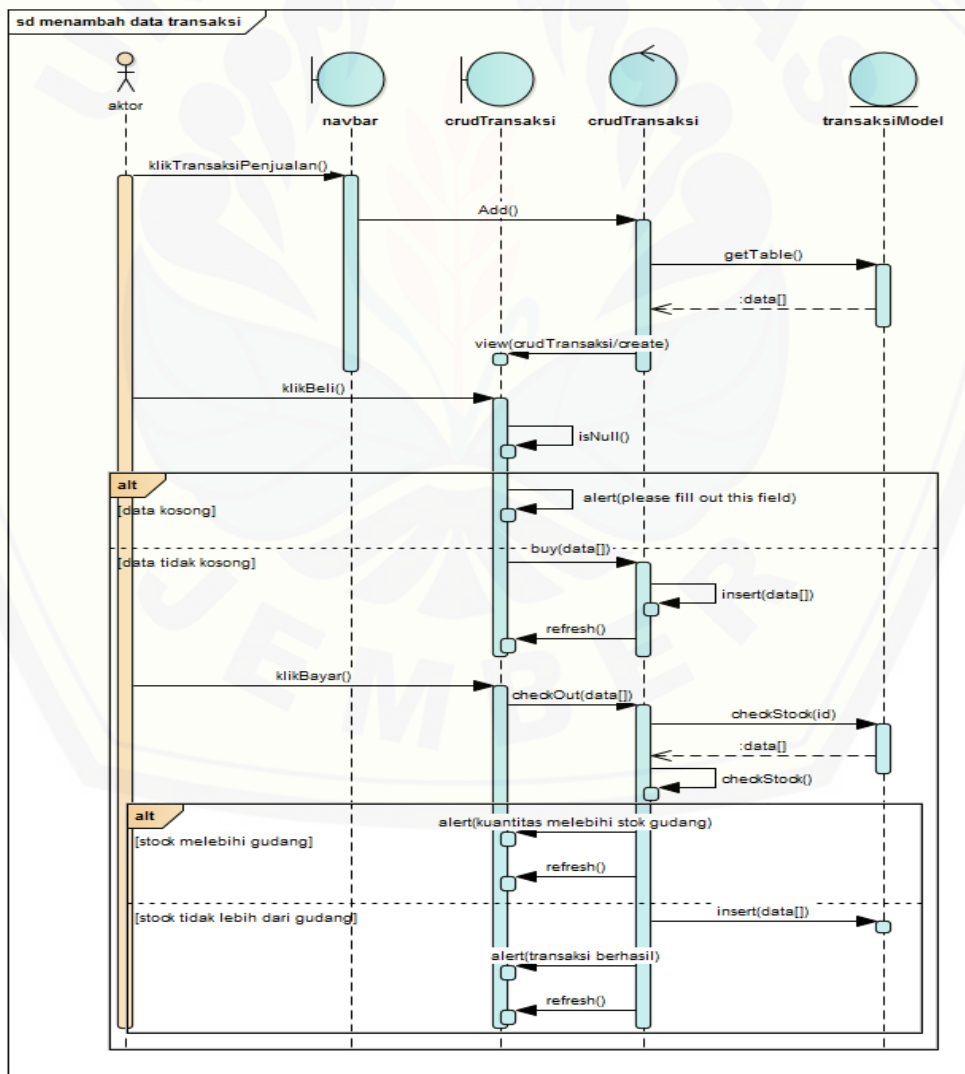
2.10 Sequence Diagram Mengubah Data Supplier



2.11 Sequence Diagram Melihat Data Transaksi

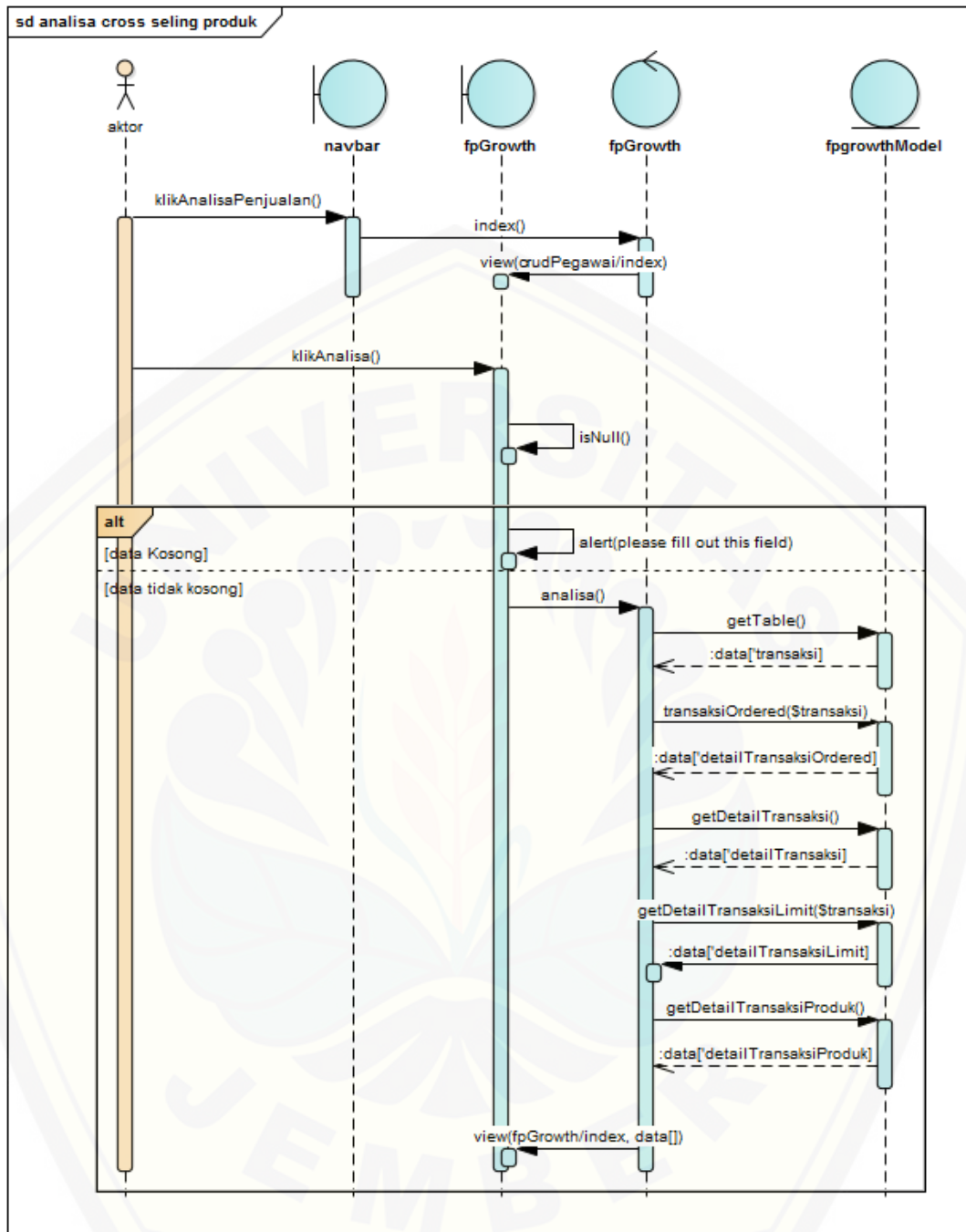


2.12 Sequence Diagram Menambah Data Transaksi

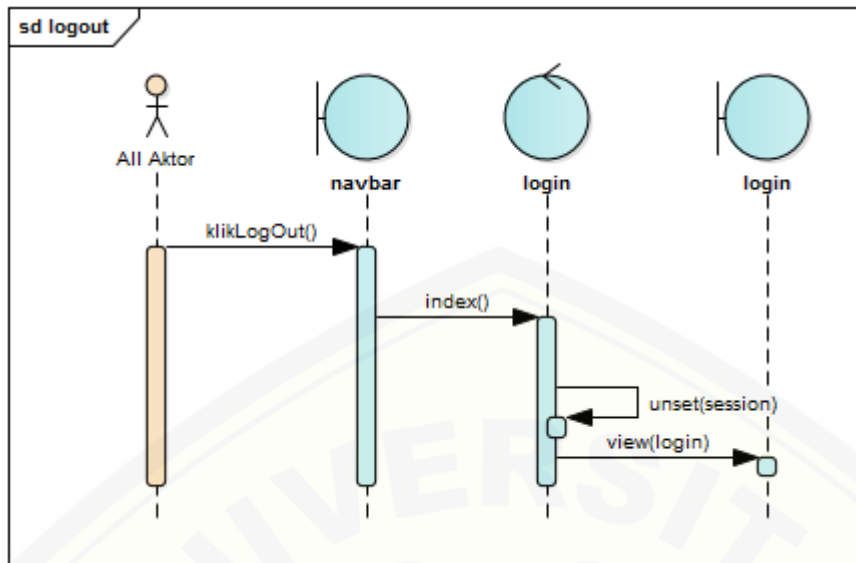




2.13 Sequence Diagram Menganalisa Cross Selling

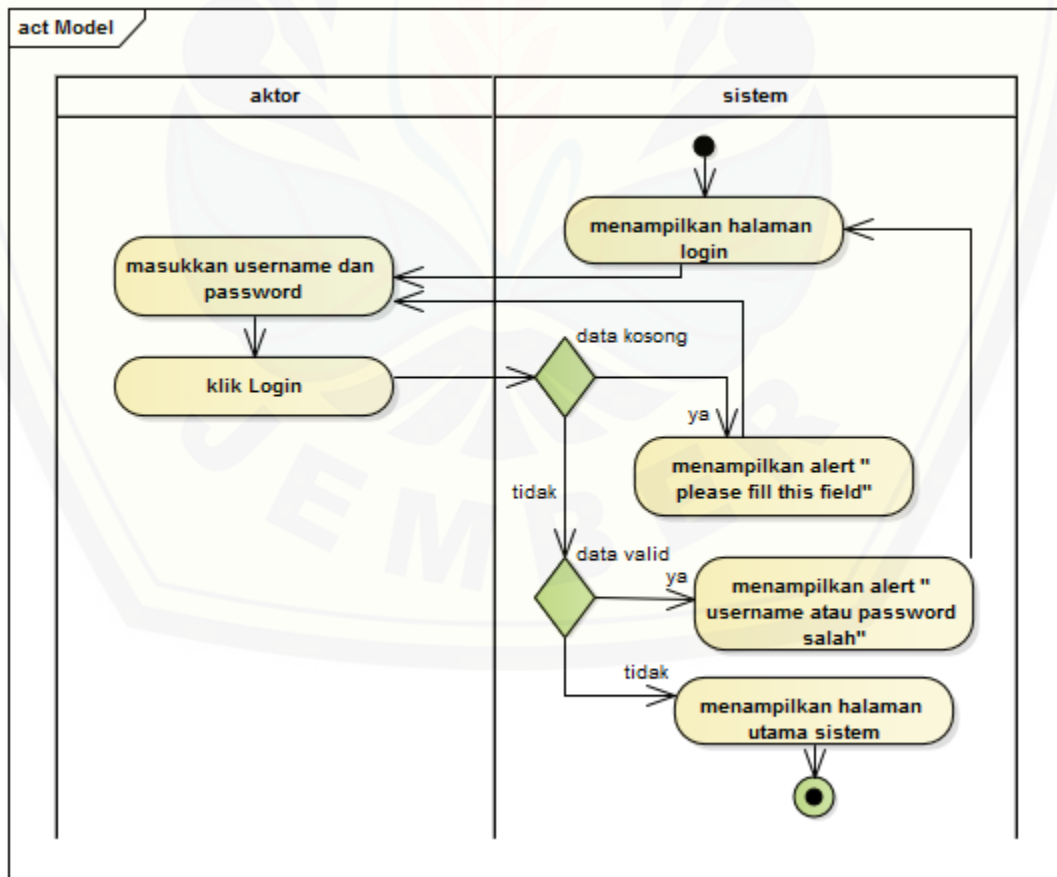


2.14 Sequence Diagram Logout

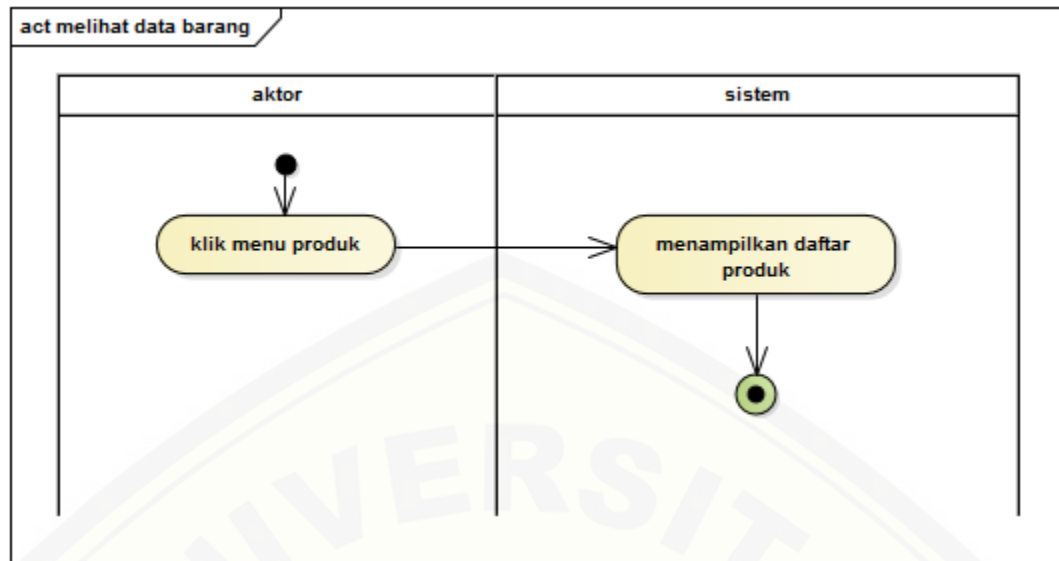


LAMPIRAN 3 (Activity Diagram)

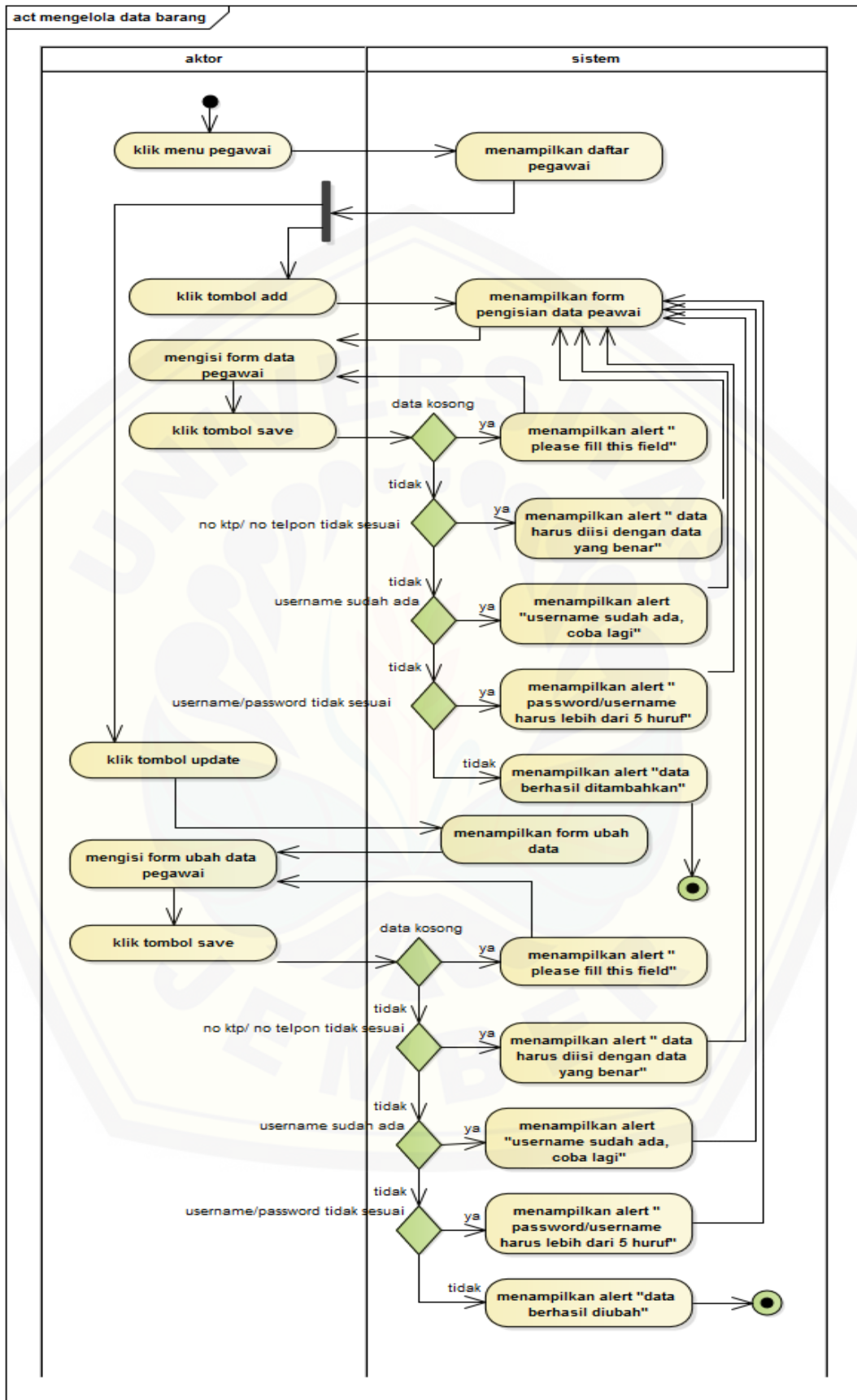
3.1 Activity Diagram Login



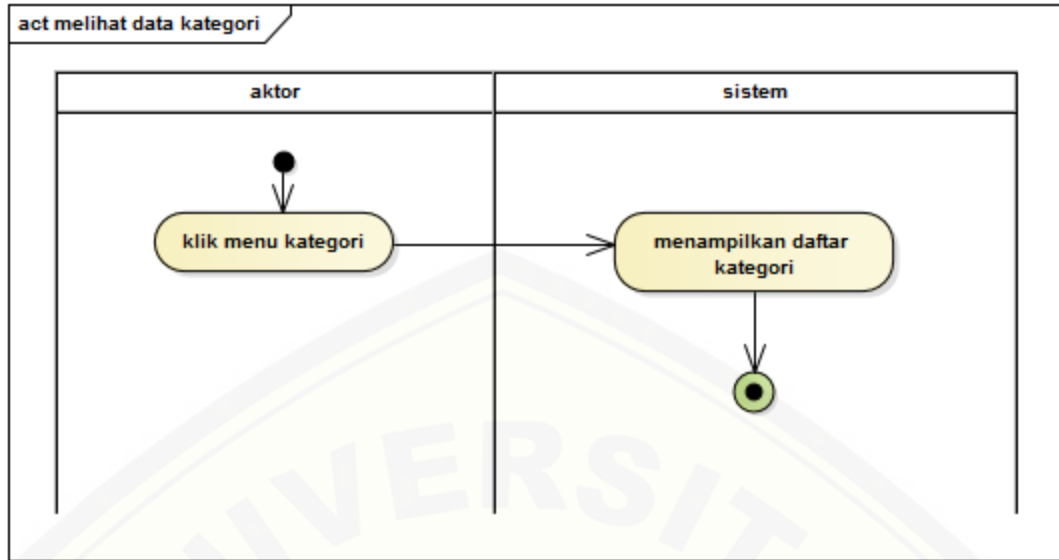
### 3.2 Activity Diagram Melihat Data Barang



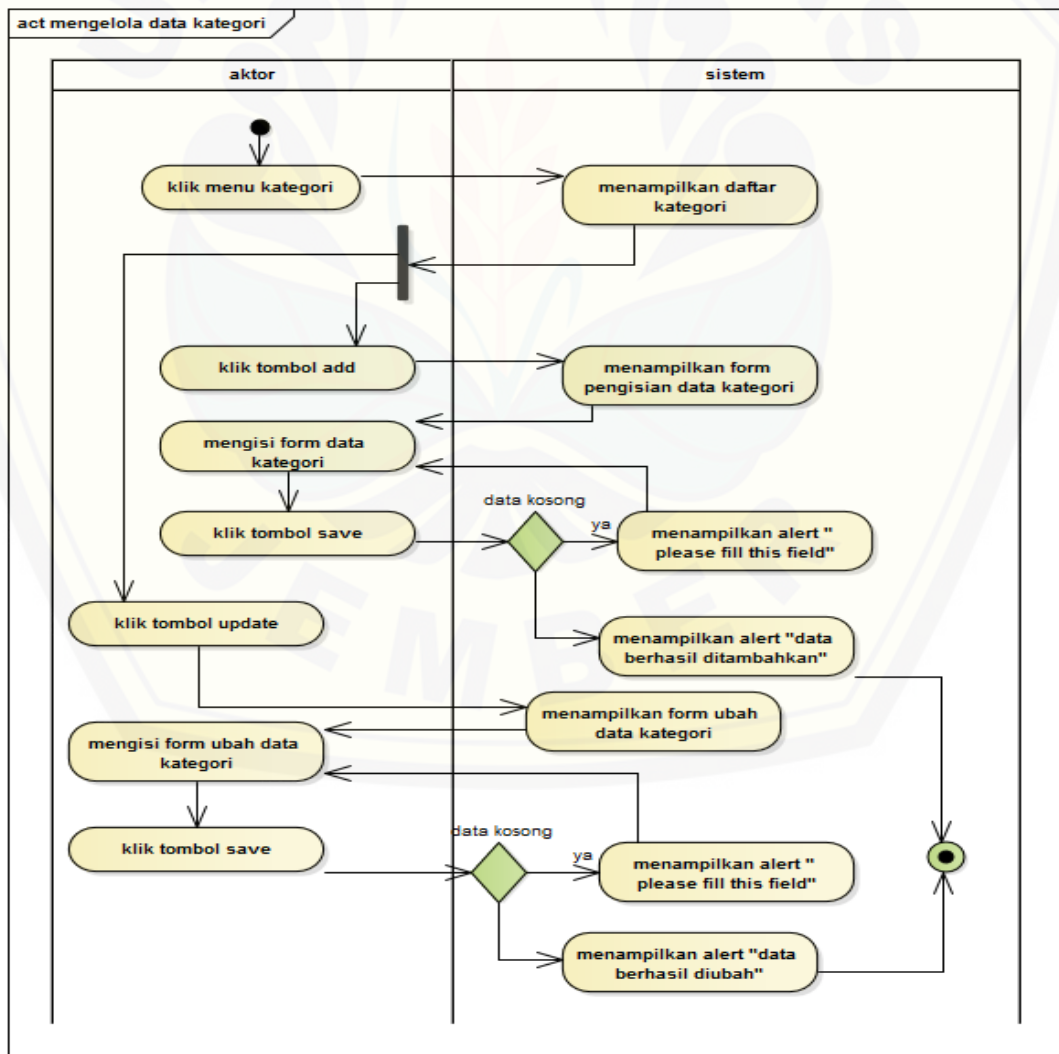
3.3 Activity Diagram Mengelola Data Barang



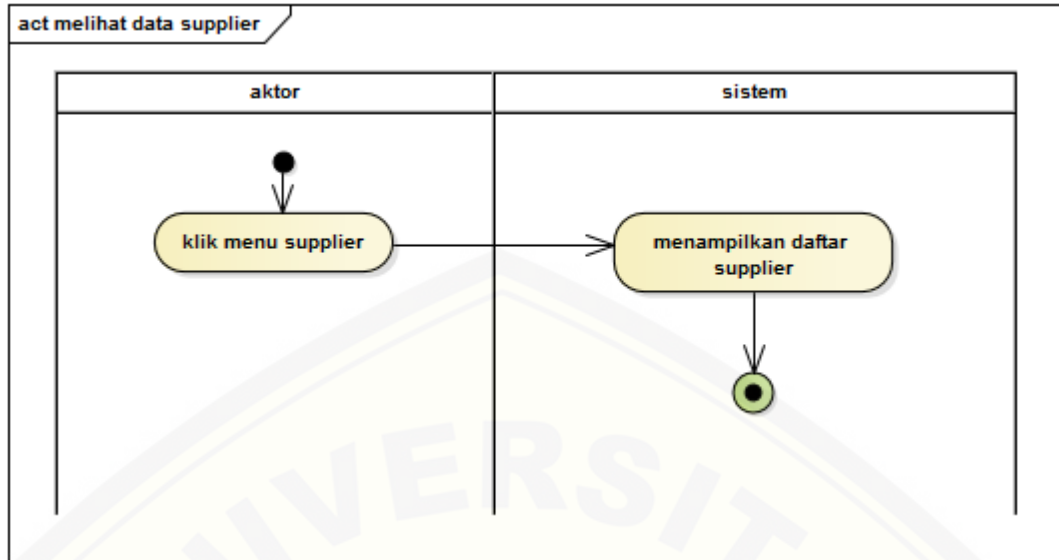
3.4 Activity Diagram Melihat Data Kategori



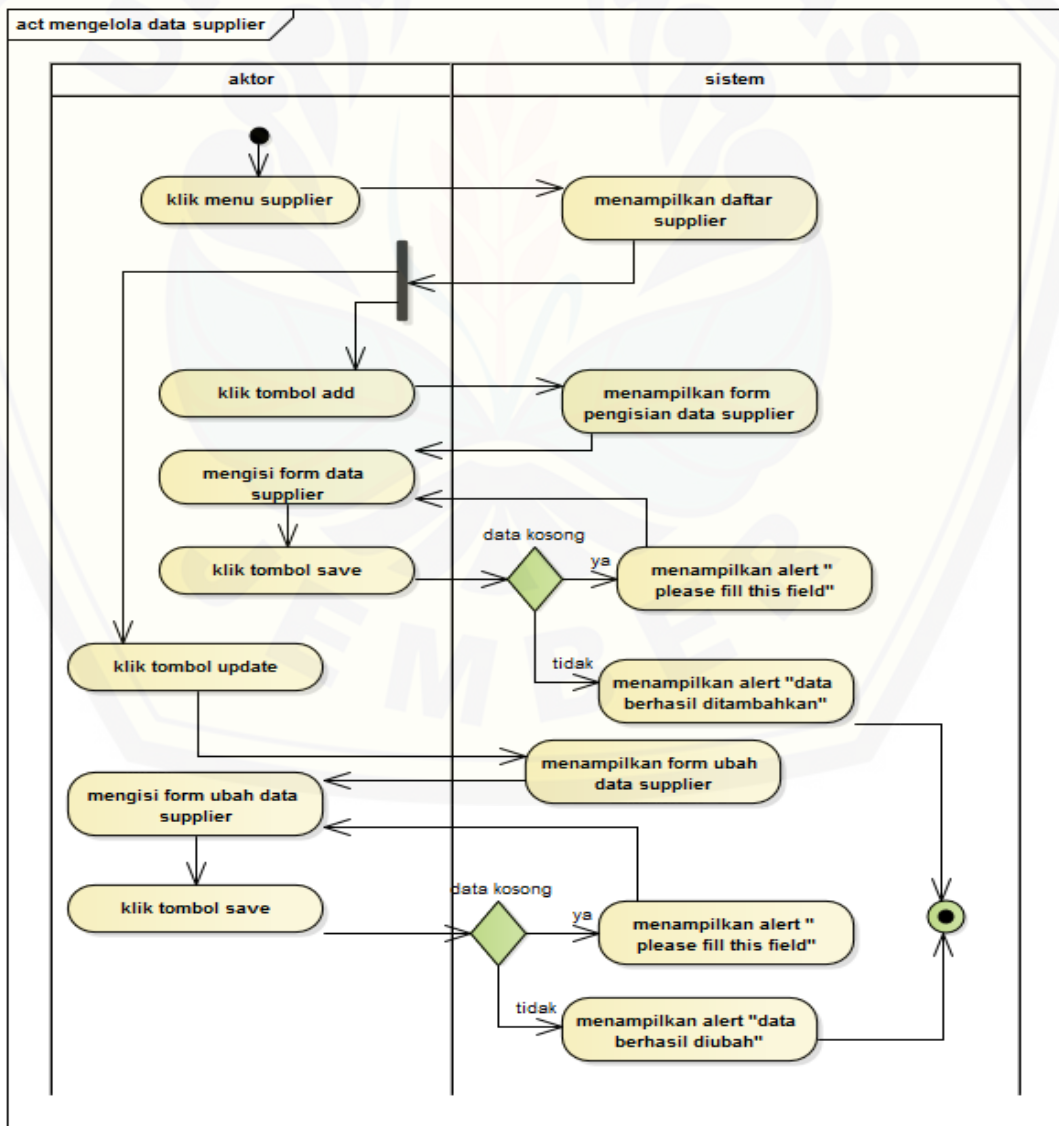
3.5 Activity Diagram Mengelola Data Kategori



3.6 Activity Diagram Melihat Data Supplier

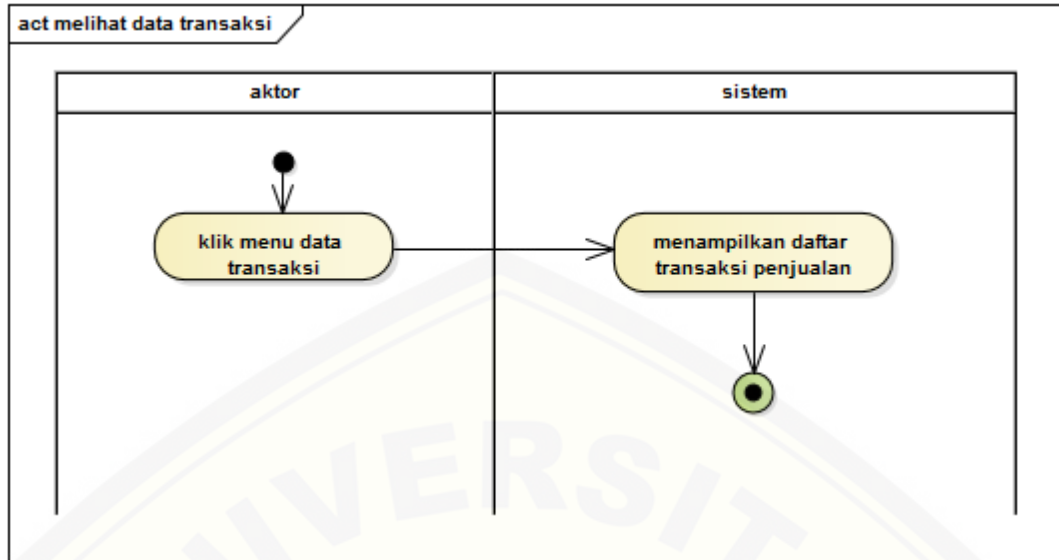


3.7 Activity Diagram Mengelola Data Supplier

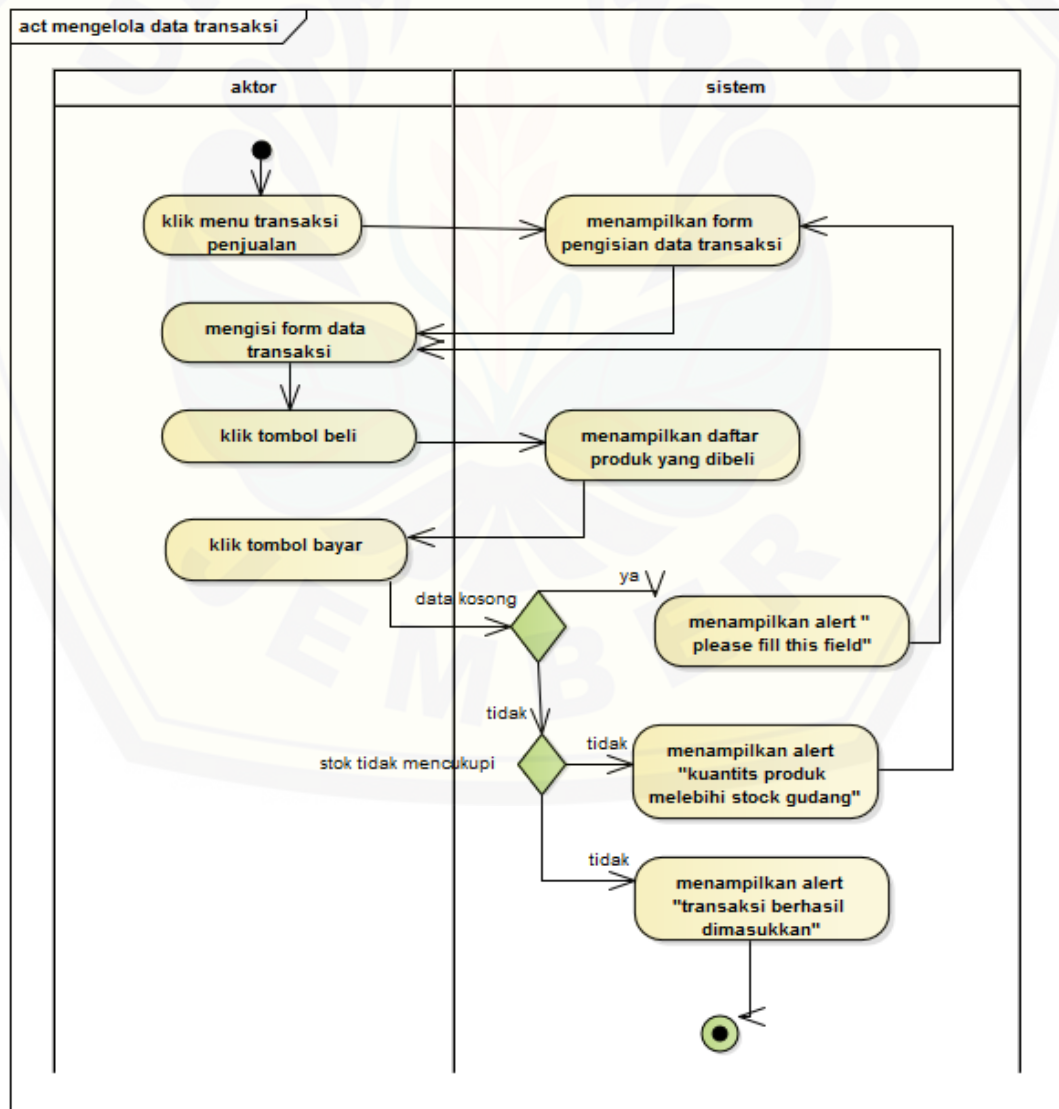




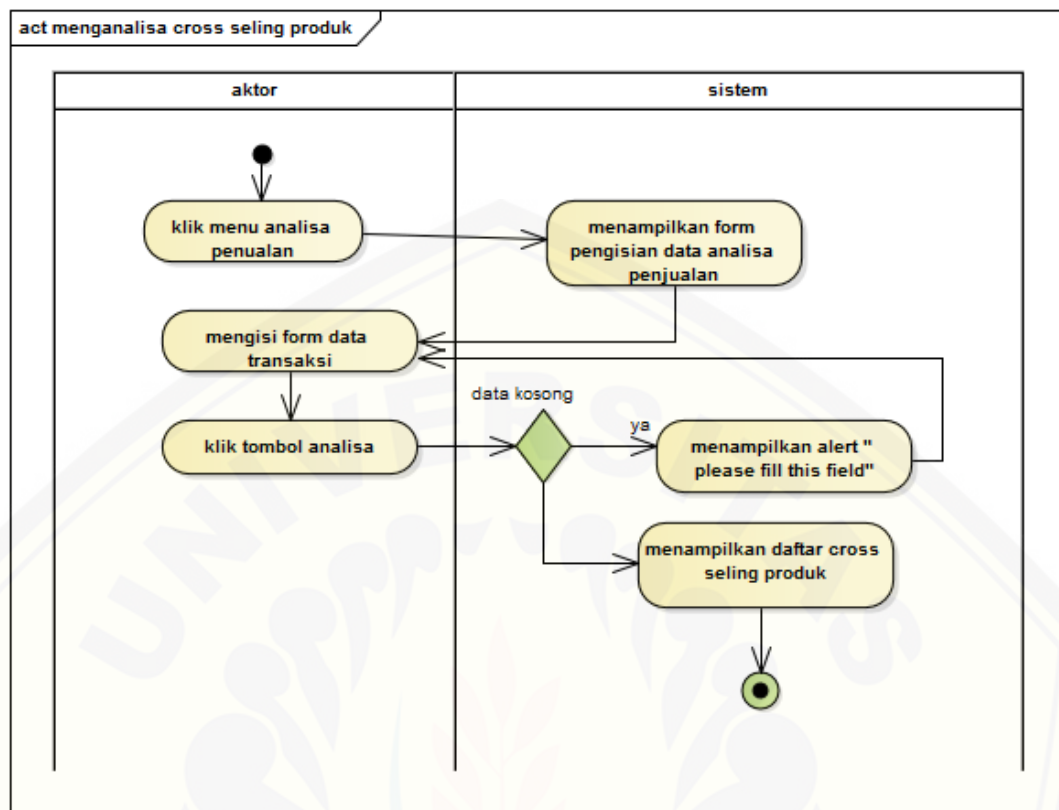
3.8 Activity Diagram Melihat Data Transaksi



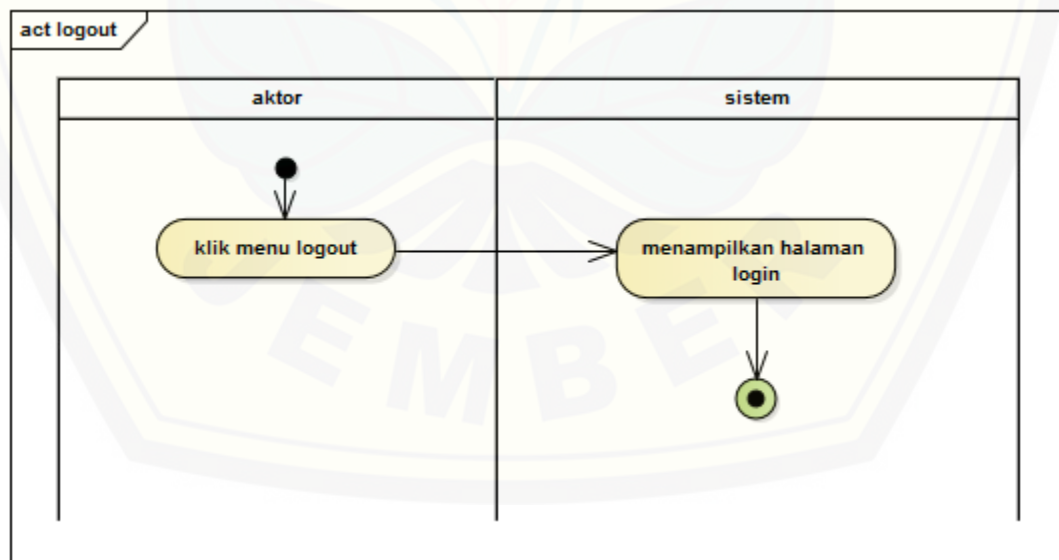
3.9 Activity Diagram Menambah Data Transaksi



3.10 Activity Diagram Menganalisa Cross Selling product



3.11 Activity Diagram Logout



## LAMPIRAN 4 (Black Box)

No.	Fungsi	Kasus	Hasil	Keterangan
1.	Melakukan login sistem	Login dengan <i>username</i> dan <i>password</i> yang benar	Login Berhasil dan menampilkan halaman utama	[√] Berhasil [ ] Gagal
		Login dengan <i>username</i> dan <i>password</i> yang salah	Menampilkan notifikasi <i>password</i> dan <i>username salah</i>	[√] Berhasil [ ] Gagal
		Login dengan form <i>login</i> tidak terisi	Menampilkan notifikasi "Please fill out this field"	[√] Berhasil [ ] Gagal
2.	Melihat data pegawai	Menekan menu pegawai	Menampilkan data semua pegawai	[√] Berhasil [ ] Gagal
3.	Menambah dan mengubah data pegawai	Input data dengan form terisi dan valid semua	Menampilkan notifikasi "Data pegawai berhasil ditambahkan/diupdate"	[√] Berhasil [ ] Gagal
		Input form dengan salah satu field kosong	Menampilkan notifikasi "Please fill out this field"	[√] Berhasil [ ] Gagal
		Input form dengan data ktp atau nomor telepon tidak valid	Menampilkan notifikasi "data harus diisi dengan data yang benar"	[√] Berhasil [ ] Gagal
		Input form dengan data <i>username</i> atau <i>password</i> tidak valid	Menampilkan notifikasi "Password/Username harus lebih dari 5 karakter"	[√] Berhasil [ ] Gagal
		Input form dengan <i>username</i> yang sudah ada	Menampilkan notifikasi "Username sudah digunakan, ganti dengan username lain."	[√] Berhasil [ ] Gagal
4.	Melihat data barang	Menekan menu produk	Menampilkan data semua barang	[√] Berhasil [ ] Gagal
5.	Menambah dan mengubah data barang	Input data dengan form terisi dan valid semua	Menampilkan notifikasi "Data barang berhasil ditambahkan/diupdate"	[√] Berhasil [ ] Gagal
		Input form dengan salah satu field kosong	Menampilkan notifikasi "Please fill out this field"	[√] Berhasil [ ] Gagal
		Input form dengan data harga atau stok barang tidak valid	Menampilkan notifikasi "Data masukan harus lebih dari 0"	[√] Berhasil [ ] Gagal

6.	Melihat data kategori	Menekan menu kategori	Menampilkan data semua kategori	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
7.	Menambah dan mengubah data kategori	Input data dengan form terisi dan valid semua	Menampilkan notifikasi “Data kategori berhasil ditambahkan/diupdate”	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Input form dengan salah satu field kosong	Menampilkan notifikasi “Please fill out this field”	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
8.	Melihat data supplier	Menekan menu supplier	Menampilkan data semua supplier	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
9.	Menambah dan mengubah data supplier	Input data dengan form terisi dan valid semua	Menampilkan notifikasi “Data supplier berhasil ditambahkan/diupdate”	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Input form dengan salah satu field kosong	Menampilkan notifikasi “Please fill out this field”	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
10.	Melihat data transaksi	Menekan menu produk	Menampilkan data semua transaksi	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
11.	Menambah data transaksi	Input data dengan form terisi dan valid semua	Menampilkan notifikasi “Data transaksi berhasil ditambahkan”	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Input form dengan salah satu field kosong	Menampilkan notifikasi “Please fill out this field”	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Input form dengan stok pembelian melebihi stok gudang	Menampilkan notifikasi “kuantitas barang melebihi stok gudang”	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Menganalisa <i>cross selling product</i>	Input data dengan form terisi, kemudian klik analisa	Menampilkan data <i>cross selling product</i>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Input form dengan salah satu field kosong	Menampilkan notifikasi “Please fill out this field”	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
				<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
				<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
				<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

**LAMPIRAN 5 (Perhitungan Manual)**

Perhitungan manual dilakukan dengan menggunakan nilai minimum support 2 dan nilai minimum confidence 30 pada 5 transaksi penjualan terakhir. Produk yang memiliki nilai support yang tidak memenuhi ketentuan yang telah ditentukan akan dihilangkan. Produk yang telah memenuhi ketentuan yang telah ditentukan akan lanjut ke tahap selanjutnya. Langkah langkah untuk menentukan paket pembelian produk antara lain

1. Tentukan *minimum support*, *minimum confidence* dan juga banyak transaksi yang akan digunakan

Support : 1  
 Confidence : 30  
 Transaksi : 5

2. Scan data transaksi dan hitung frekuensi kemunculan dari setiap produk

Id	Produk
1	pensil
2	pensil , Buku Tulis ,
3	Buku Tulis , Bolpoin ,
4	pensil warna , Buku Gambar ,
5	pensil , pensil warna , Buku Tulis , Buku Gambar , Bolpoin ,

3. Tentukan itemset yang memenuhi nilai support yang telah ditentukan

Produk	Support
Buku Tulis	3
Pensil	3
Bolpoin	2
Buku Gambar	2
pensil warna	2

4. Tulis ulang data transaksi yang tidak tereliminasi dan diurutkan berdasarkan support

Id	Produk
1	Buku Tulis , pensil , Bolpoin , Buku Gambar , pensil warna ,
2	Buku Gambar , pensil warna ,
3	Buku Tulis , Bolpoin ,
4	Buku Tulis , pensil ,
5	pensil ,

5. Bangkitkan *conditional pattern base* dari data transaksi tersebut

Item	Pattern
pensil warna	{ Buku Tulis, pensil, Bolpoin, Buku Gambar : 1 } { Buku Gambar, : 1 }
Buku Gambar	{ Buku Tulis, pensil, Bolpoin : 1 }
Bolpoin	{ Buku Tulis, pensil : 1 } { Buku Tulis : 1 }
Pensil	{ Buku Tulis : 2 }

6. Jumlahkan support count dari setiap item yang sama

Item	Conditional Pattern Base
pensil warna	{ Buku Tulis : 1, pensil : 1, Bolpoin : 1, Buku Gambar : 1, } { Buku Gambar : 1, }
Buku Gambar	{ Buku Tulis : 1, pensil : 1, Bolpoin : 1, }
Bolpoin	{ Buku Tulis : 1, pensil : 1, } { Buku Tulis : 1, }
Pensil	{ Buku Tulis : 2, }

Item	Conditional Fp – Tree
pensil warna	{Buku Tulis : 1 } {pensil : 1 } {Bolpoin : 1 } {Buku Gambar : 2 }
Buku Gambar	{Buku Tulis : 1 } {pensil : 1 } {Bolpoin : 1 }
Bolpoin	{Buku Tulis : 2 } {pensil : 1 }
Pensil	{Buku Tulis : 2 }

7. Kombinasikan item untuk setiap *conditional fp tree*

Item	Frequent Item Set
pensil warna	{pensil warna, Buku Tulis : 1 } {pensil warna, pensil : 1 } {pensil warna, Bolpoin : 1 } {pensil warna, Buku Gambar : 2 }
Buku Gambar	{Buku Gambar, Buku Tulis : 1 } {Buku Gambar, pensil : 1 } {Buku Gambar, Bolpoin : 1 }
Bolpoin	{Bolpoin, Buku Tulis : 2 } {Bolpoin, pensil : 1 }
Pensil	{pensil, Buku Tulis : 2 }

8. Menghitung nilai *support* dan *confidence* untuk membentuk aturan asosiasi

Item	Item Set	Support	Confidence
1	pensil warna, Buku Tulis	0.25	0.50
2	pensil warna, pensil	0.25	0.50
3	pensil warna, Bolpoin	0.33	0.50
4	pensil warna, Buku Gambar	1.00	1.00