

**APLIKASI PENJUALAN PULSA *MULTI OPERATOR*
BERBASIS *JAVA* DAN *SMS GATEWAY*.**

TUGAS AKHIR

**Diajukan Untuk Memenuhi Salah Satu Syarat Mencapai Gelar Ahli Madya
Program Diploma III Ilmu Komputer**



Diajukan oleh :

AMINUDIN NOOR

NIM M3107068

**PROGRAM DIPLOMA III ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS SEBELAS MARET
SURAKARTA**

2010

HALAMAN PERSETUJUAN

**APLIKASI PENJUALAN PULSA *MULTI OPERATOR*
BERBASIS *JAVA* DAN *SMS GATEWAY*.**

Disusun Oleh :

AMINUDIN NOOR

M 3107068

Tugas Akhir ini telah disetujui dan dipertahankan di depan dewan penguji pada
tanggal 19 Juli 2010

Pembimbing Utama

Viska Ina Variani, M.Si.

NIP. 19720617 199702 2 001

HALAMAN PENGESAHAN**APLIKASI PENJUALAN PULSA *MULTI OPERATOR*
BERBASIS *JAVA* DAN *SMS GATEWAY*.**

Disusun Oleh
AMINUDIN NOOR
M3107068

Dibimbing Oleh
Pembimbing Utama

Viska Inda Variani, M.Si.
NIP. 19720617 199702 2 001

Tugas Akhir ini telah diterima dan disahkan oleh dewan penguji Tugas Akhir
Program Diploma III Ilmu Komputer
Pada hari Senin tanggal 19 Juli 2010

Dewan Penguji

1. Penguji 1. Viska Inda Variani, M.Si. ()
NIP. 19720617 199702 2 001
2. Penguji 2. Muhammad Asri Safi'ie, S.Si. ()
NIDN. 0603118103
3. Penguji 3. Rudi Hartono, S.Si. ()
NIDN. 0626128402

Disahkan Oleh

a.n. Dekan FMIPA UNS
Pembantu Dekan I

Ketua Program Studi
DIII Ilmu Komputer UNS

Ir. Ari Handono Ramelan, M.Sc. Ph.D
NIP. 19610223 198601 1 001

Drs. Y.S. Palgunadi, M.Sc
NIP. 19560407 198303 1 004

ABSTRACT

Aminudin Noor. 2010. SELLING APPLICATION OF MULTI OPERATOR CREDIT CELLULAR BASED ON JAVA PROGRAM AND SMS GATEWAY. Diploma III Computer Science, Science and Mathematics Faculty, Sebelas Maret University of Surakarta in Surakarta, June 2010.

Selling application of multi operator credit cellular is an application service that handles the selling data of the multi operator credit cellular. Purpose of making this application is to record the transaction processing and presentation of the transactions quickly, accurately, and efficiently.

This selling application was built using the Java programming with NetBeans software and using Gammu as an SMS server. While for the database it was used MySQL. This application was running by sending a message / SMS in the form of selling data that has been entered to the server of multi-operator credit cellular using Gammu as a server. Sales data were also entered in the database storage. This application was expected to provide facilities that are relatively more efficient in the records of transactions and delivery of credit cellular.

It can be concluded that selling application of multi-operator credit cellular has already been designed and implemented.

Keywords: Applications, Java, NetBeans, MySQL, and SMS Gateway.

ABSTRAK

Aminudin Noor. 2010. APLIKASI PENJUALAN PULSA *MULTI OPERATOR* BERBASIS *JAVA* DAN *SMS GATEWAY*. Diploma III Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Sebelas Maret Surakarta, Juni 2010.

Aplikasi penjualan pulsa *multi operator* adalah aplikasi yang menangani pendataan penjualan pulsa *multi operator*. Tujuan pembuatan aplikasi ini adalah untuk mendata proses transaksi dan penyajian transaksi secara cepat, tepat, dan efisien.

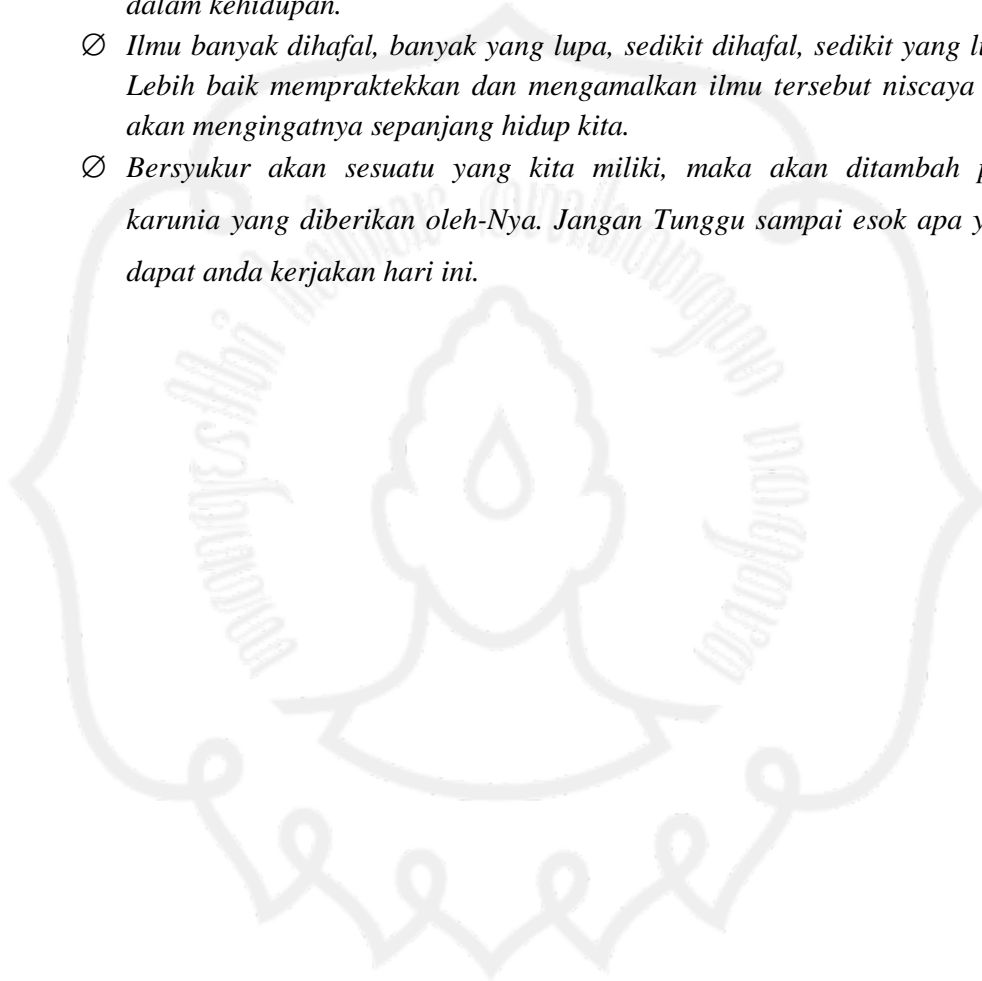
Aplikasi penjualan ini dibangun menggunakan pemrograman *Java* dengan software *NetBeans* serta menggunakan *Gammu* sebagai *SMS server*nya. Sedangkan untuk *databasenya* menggunakan *MySQL*. Aplikasi ini berjalan dengan mengirimkan pesan/SMS berupa data penjualan yang telah diinputkan kepada *server* pulsa *multi-operator* dengan menggunakan *Gammu* sebagai *server*nya. Data penjualan tersebut juga masuk pada *database* penyimpanan. Aplikasi ini diharapkan dapat memberikan fasilitas yang relatif lebih efisien dalam mendata transaksi dan pengiriman pulsa.

Dapat disimpulkan bahwa Aplikasi penjualan pulsa *multi-operator* ini telah dirancang dan diimplementasikan.

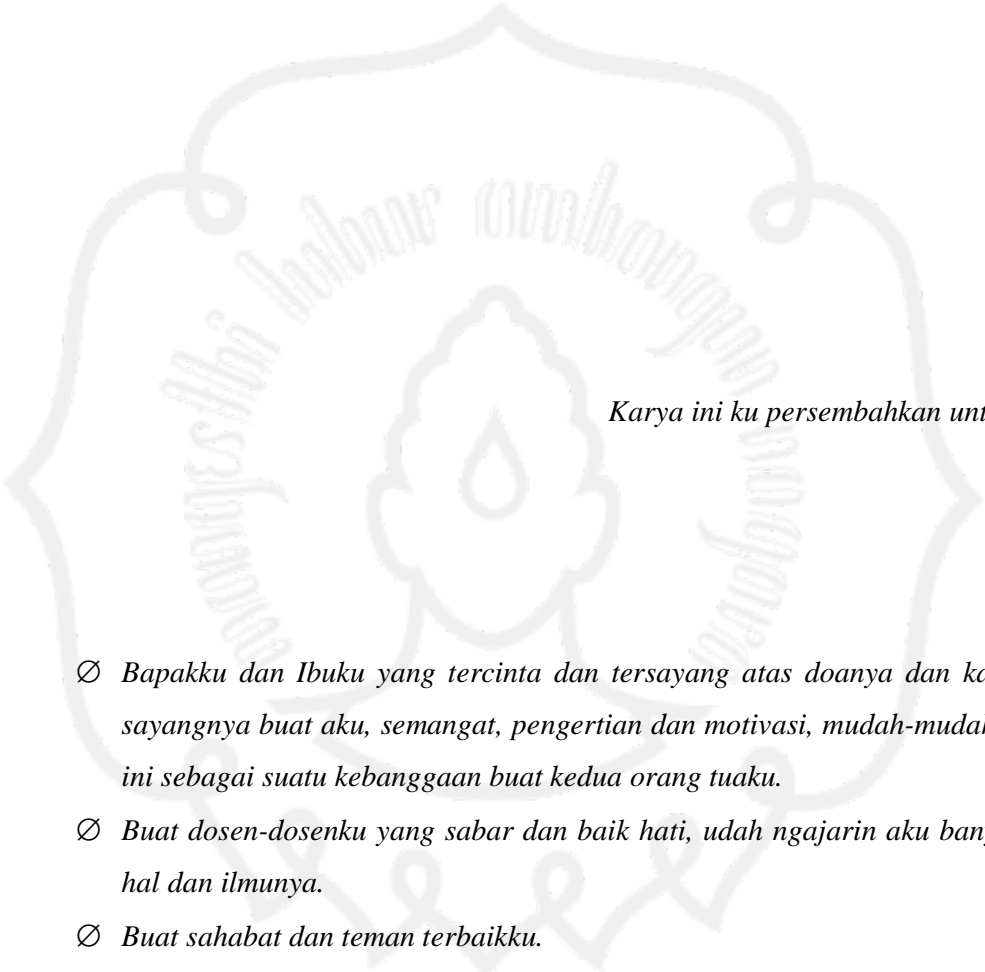
Kata Kunci : Aplikasi, *Java*, *NetBeans*, *MySQL*, dan *SMS Gateway*.

MOTTO

- Ø *Sayangi diri sendiri sebelum kamu menyayangi orang lain karena dengan menyayangi diri sendiri maka secara tidak langsung kita akan menyayangi dan menghargai orang lain.*
- Ø *Belajar, berdoa selalu, dan memanfaatkan waktu adalah kunci kesuksesan dalam kehidupan.*
- Ø *Ilmu banyak dihafal, banyak yang lupa, sedikit dihafal, sedikit yang lupa. Lebih baik mempraktekkan dan mengamalkan ilmu tersebut niscaya kita akan mengingatnya sepanjang hidup kita.*
- Ø *Bersyukur akan sesuatu yang kita miliki, maka akan ditambah pula karunia yang diberikan oleh-Nya. Jangan Tunggu sampai esok apa yang dapat anda kerjakan hari ini.*



PERSEMBAHAN



Karya ini ku persembahkan untuk:

- Ø *Bapakku dan Ibuku yang tercinta dan tersayang atas doanya dan kasih sayangnya buat aku, semangat, pengertian dan motivasi, mudah-mudahan ini sebagai suatu kebanggaan buat kedua orang tuaku.*
- Ø *Buat dosen-dosenku yang sabar dan baik hati, udah ngajarin aku banyak hal dan ilmunya.*
- Ø *Buat sahabat dan teman terbaikku.*
- Ø *Buat temen-temenku D3 Ilmu Komputer 2007.*
- Ø *Buat semua pihak yang telah membantu aku, terima kasih buat semuanya*

KATA PENGANTAR

Bismillahirrohmanirrohim

Segala puji bagi Allah Tuhan semesta alam, yang dengan kuasa-Nya kita dapat menikmati hidup dan segala karunia-Nya, Sholawat serta salam selalu tercurahkan kepada junjungan kita Nabi Muhammad SAW, serta sahabat, keluarganya dan pengikutnya yang setia hingga akhir jaman.

Penulis tidak henti-hentinya mengucap syukur atas karunia yang telah diberikan oleh Tuhan Yang Maha Esa karena penulis dapat menyelesaikan tugas akhir dengan judul “Aplikasi Penjualan Pulsa *Multi Operator* Berbasis *Java* Dan *SMS Gateway*” sesuai waktu yang ditentukan.

Penyusunan tugas akhir ini dimaksudkan untuk memenuhi persyaratan kelulusan Program Diploma III Teknik Informatika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Sebelas Maret Surakarta.

Penulis menyadari bahwa dalam penyusunan tugas akhir ini tidak lepas dari berbagai pihak yang telah banyak membantu. Dalam kesempatan ini penulis mengucapkan banyak terimakasih kepada :

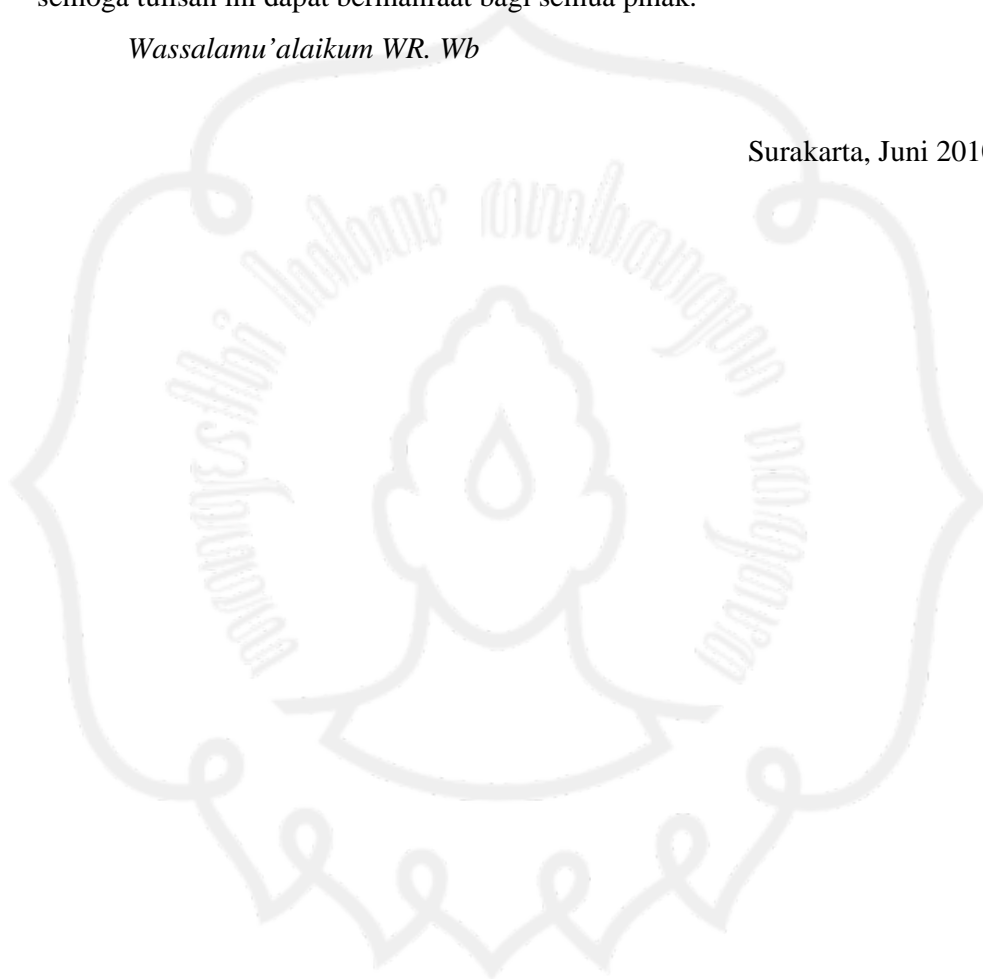
1. Prof. Drs. Sutarno, M.Sc, Ph.D selaku dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Sebelas Maret Surakarta
2. Drs. YS. Palgunadi, M.Sc selaku kepala Program Diploma III Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Sebelas Maret Surakarta
3. Viska Inda Variani, M.Si selaku dosen pembimbing tugas akhir, yang telah memberikan petunjuk dan masukan selama pembuatan tugas akhir.
4. Dra. Diari Indriati, M.Si selaku dosen pembimbing akademik yang telah banyak memberi masukan dan bimbingan akademik selama menjadi mahasiswa DIII Teknik Informatika
5. Semua mahasiswa Diploma III teknik Informatika 2007 yang telah banyak membantu secara moril ataupun materiil selama pelaksanaan pembuatan tugas akhir.

6. Semua pihak yang secara langsung ataupun tidak langsung telah membantu penulis.

Semoga Allah SWT membalas semua kebaikan. Laporan tugas akhir ini masih belum sempurna, oleh karena itu mohon saran dan kritiknya. Akhir kata semoga tulisan ini dapat bermanfaat bagi semua pihak.

Wassalamu'alaikum WR. Wb

Surakarta, Juni 2010



DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PERSETUJUAN	ii
HALAMAN PENGESAHAN	iii
ABSTRACT	iv
ABSTRAK	v
MOTTO	vi
PERSEMBAHAN	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR TABEL	xiii
DAFTAR GAMBAR	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	1
1.3 Batasan Masalah	1
1.4 Tujuan Dan Manfaat	2
1.5 Metodologi Penelitian	2
1.6 Sistematika Penulisan	3
BAB II LANDASAN TEORI	5
2.1 Bahasa Pemrograman <i>Java</i>	5
2.2 Pengetian <i>DBMS MySQL</i>	6
2.3 Koneksi Via <i>Java Database Connectivity (JDBC)</i>	6
2.3.1 Arsitektur <i>JDBC</i>	7
2.3.2 Koneksi <i>MySQL</i>	7
2.3 Pengertian <i>Gammu SMS Server</i>	9
2.5 Pengetian <i>Unified Modelling Language (UML)</i>	9
2.6 Pengetian <i>Use Case Diagram</i>	10
2.7 Pengetian <i>Class Diagram</i>	11

2.8	Pengetian <i>Sequence Diagram</i>	11
2.9	Pengetian <i>Activity Diagram</i>	12
2.10	Pengetian <i>Apache</i>	12
BAB III	HASIL DAN PEMBAHASAN	14
3.1	Perancangan Sistem.....	14
3.2	Model Analisis	14
3.2.1	<i>Use Case Diagram</i>	14
3.2.2	Skenario <i>Use Case</i>	18
3.2.3	<i>Activity Diagram</i>	24
3.3	Model Perancangan	32
3.3.1	<i>Sequence Diagram</i>	32
3.3.2	<i>Class Diagram</i>	36
3.4	Kebutuhan <i>Hardware</i> dan <i>Software</i>	40
3.4.1	Kebutuhan <i>Hardware</i>	40
3.4.2	Kebutuhan <i>Software</i>	40
3.4	Perancangan Antarmuka	41
3.4.1	Rancangan <i>Form</i> Utama	41
3.4.2	Rancangan <i>Form Login</i>	31
3.4.3	Rancangan <i>Form Operator</i>	41
3.4.5	Rancangan <i>Form Deposit</i>	42
3.4.6	Rancangan <i>Form PIN</i>	43
3.4.7	Rancangan <i>Form Pulsa</i>	43
3.4.8	Rancangan <i>Form Gateway</i>	44
3.4.9	Rancangan <i>Form Transaksi</i>	44
3.4.10	Rancangan <i>Form Laporan Transaksi</i>	45
BAB IV	IMPLEMENTASI	46
4.1	Implementasi	46
3.4.10	Implementasi Antarmuka	56

BAB IV	PENUTUP	54
	4.1 Kesimpulan	54
	4.2 Saran	54
BAB IV	DAFTAR PUSTAKA	55



DAFTAR TABEL

Tabel 2.1	Tabel Simbol <i>Use Case Diagram</i>	10
Tabel 2.2	Tabel Simbol <i>Class Diagram</i>	11
Tabel 2.3	Tabel Simbol <i>Sequence Diagram</i>	12
Tabel 2.4	Tabel Simbol <i>Activity Diagram</i>	12
Tabel 3.1	Tabel Definisi Aktor	14
Tabel 3.2	Tabel Definisi <i>Use case</i>	15
Tabel 3.3	Tabel Skenario <i>Login</i>	18
Tabel 3.4	Tabel Skenario Mendata <i>Operator</i>	19
Tabel 3.5	Tabel Skenario Menambah Deposit	19
Tabel 3.6	Tabel Skenario Mengganti PIN.....	20
Tabel 3.7	Tabel Skenario Mendata Pulsa.....	21
Tabel 3.8	Tabel Skenario Mendata <i>Gateway</i>	22
Tabel 3.9	Tabel Skenario Mendata Transaksi	22
Tabel 3.10	Tabel Skenario Membuat Laporan	23

DAFTAR GAMBAR

Gambar 2.1	Arsitektur JDBC..... xiii	7
Gambar 3.1	<i>Use Case Diagram</i> Aplikasi Penjualan Pulsa	17
Gambar 3.2	<i>Activity Diagram</i> Login	24
Gambar 3.3	<i>Activity Diagram</i> Mendata Deposit	25
Gambar 3.4	<i>Activity Diagram</i> Penggantian PIN	26
Gambar 3.5	<i>Activity Diagram</i> Mendata Pulsa.....	27
Gambar 3.6	<i>Activity Diagram</i> Mendata Gateway	28
Gambar 3.7	<i>Activity Diagram</i> Mendata Operator.....	29
Gambar 3.8	<i>Activity Diagram</i> Mendata Transaksi	30
Gambar 3.9	<i>Activity Diagram</i> Membuat Laporan.....	31
Gambar 3.10	<i>Sequence Diagram</i> Login.....	32
Gambar 3.11	<i>Sequence Diagram</i> Menambah Deposit	33
Gambar 3.12	<i>Sequence Diagram</i> Mengganti PIN.....	33
Gambar 3.13	<i>Sequence Diagram</i> Mendata Pulsa.....	34
Gambar 3.14	<i>Sequence Diagram</i> Mendata Operator	34
Gambar 3.15	<i>Sequence Diagram</i> Mendata Gateway	35
Gambar 3.16	<i>Sequence Diagram</i> Mendata Transaksi	35
Gambar 3.17	<i>Sequence Diagram</i> Membuat Laporan	36
Gambar 3.18	<i>Class Diagram</i> Login	36
Gambar 3.19	<i>Class Diagram</i> Mendata Operator	37
Gambar 3.20	<i>Class Diagram</i> Menambah Deposit	37
Gambar 3.21	<i>Class Diagram</i> Mengganti PIN.....	38
Gambar 3.22	<i>Class Diagram</i> Mendata Pulsa.....	38
Gambar 3.23	<i>Class Diagram</i> Mendata Gateway	39
Gambar 3.24	<i>Class Diagram</i> Mendata Transaksi	39
Gambar 3.25	<i>Class Diagram</i> Membuat Laporan	40
Gambar 3.26	Rancangan <i>Form</i> Utama	41
Gambar 3.27	Rancangan <i>Form</i> Login	41
Gambar 3.28	Rancangan <i>Form</i> Operator.....	42

Gambar 3.29	Rancangan <i>Form</i> Deposit	42
Gambar 3.30	Rancangan <i>Form</i> PIN	43
Gambar 3.31	Rancangan <i>Form</i> Pulsa	43
Gambar 3.32	Rancangan <i>Form Gateway</i>	44
Gambar 3.33	Rancangan <i>Form</i> Transaksi	45
Gambar 3.34	Rancangan <i>Form</i> Laporan.....	45
Gambar 4.1	<i>Form</i> Jendela Utama.....	46
Gambar 4.2	<i>Form Login</i>	47
Gambar 4.3	<i>Form</i> Deposit	47
Gambar 4.4	<i>Form</i> PIN	48
Gambar 4.5	<i>Form</i> Operator.....	49
Gambar 4.6	<i>Form</i> Pulsa	50
Gambar 4.7	<i>Form Gateway</i>	51
Gambar 4.8	<i>Form</i> Transaksi	52
Gambar 4.9	<i>Form</i> Laporan Penjualan	53
Gambar 4.10	Tampilan Laporan Panjualan	53

BAB I

PENDAHULUAN

1.1 Latar belakang Masalah

Perkembangan telekomunikasi seluler di Indonesia berjalan dengan pesat. Perkembangan telekomunikasi ini mempengaruhi segala bidang kehidupan manusia. Hampir di setiap saat kita tidak bisa lepas dengan penggunaan alat telekomunikasi seluler.

Perkembangan telekomunikasi seluler ini membuka peluang usaha bagi kalangan manapun untuk berjualan pulsa. Bisnis pulsa yang sekarang sedang marak adalah penjualan pulsa elektrik *multi operator*. Pulsa *multi operator* ini merupakan sistem penjualan pulsa yang menggunakan sebuah *server* yang menampung semua pulsa dari semua operator. Banyak agen pulsa atau *counter* pulsa yang sekarang beralih menggunakan sistem penjualan pulsa ini.

Tetapi dalam bisnis penjualan pulsa ini belum ada sistem yang mendata penjualan secara terkomputerisasi. Sistem penjualan pulsa sekarang ini masih banyak yang dilakukan manual. Sistem penjualan secara manual ini dapat menimbulkan kesulitan dalam perekapan data dan pengecekan data kembali.

Sehubungan dengan hal tersebut di atas, penulis tertarik membuat Aplikasi Penjualan Pulsa *Multi Operator* yang dapat mendata semua transaksi secara terkomputerisasi. Diharapkan aplikasi ini dapat membantu agen penjual pulsa *multi operator* dalam melakukan proses transaksi dan penyajian data secara cepat, tepat, dan efisien.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah di atas, maka rumusan masalahnya adalah bagaimana cara membuat aplikasi penjualan pulsa *multi operator* berbasis *java* dan *sms gateway*?

1.3 Batasan Masalah

Untuk menghindari kekompleksitasan dalam pembahasan masalah penulisan Tugas Akhir ini, maka penulis membatasi masalah hanya pada proses

transaksi penjualan dan proses ganti nomer PIN yang menggunakan *SMS Gateway*. Proses pendataan nominal pulsa, proses pendataan spesifikasi pulsa, proses pendataan *gateway server* pulsa, dan proses pendataan operator di inputkan oleh operator pada *formnya* masing-masing. Program ini juga dibatasi hanya untuk satu *server* pulsa saja.

1.4 Tujuan dan Manfaat

1.4.1 Tujuan

Tujuan dari penelitian Tugas Akhir ini adalah membuat Aplikasi Penjualan Pulsa *Multi Operator* yang dapat memudahkan agen penjual pulsa *multi operator* dalam perekapan data transaksi dan penyajian data kembali data transaksi tersebut.

1.4.2 Manfaat

Pelaksanaan Tugas Akhir ini diharapkan dapat memberikan manfaat sebagai berikut;

1. Dapat mempraktekkan ilmu yang telah diperoleh selama masa perkuliahan.
2. Hasil penelitian akan menambah pengetahuan dan wawasan penulis dalam pembuatan suatu aplikasi.
3. Dapat mengetahui cara pengolahan data yang lebih cepat, akurat dan efisien dari penggunaan sistem komputer.
4. Dapat dijadikan sebagai bahan referensi dan tambahan informasi untuk pengkajian topik, berkaitan dengan masalah yang sama dengan penelitian ini dan dapat digunakan sebagaimana mestinya.

1.6 Metodologi Penelitian

Metode pengumpulan data yang digunakan data yang digunakan pada penyusunan Tugas Akhir ini adalah sebagai berikut;

1. Studi Lapangan

Dengan cara meneliti obyek penelitian secara langsung untuk mendapatkan data-data dan keterangan-keterangan yang berhubungan dengan masalah yang sedang diteliti. Dapat dilakukan dengan wawancara

langsung kepada pelaku bisnis pulsa *multi operator* untuk mendapatkan informasi dan melakukan observasi tentang operasional penjualan yang dilakukan selama ini. Metode pengumpulan data dengan cara melakukan pengamatan secara langsung pada obyek yang akan diteliti.

2. Studi Pustaka

Mempelajari kepustakaan yang berhubungan dengan pemrograman *java*, *MySQL*, basis data, dan referensi-referensi lainnya yang dapat mendukung pembuatan aplikasi ini. Setelah melakukan penelitian dan mendapatkan data-data yang dibutuhkan dalam pembuatan sistem, maka langkah yang dilakukan selanjutnya adalah analisa data dan kebutuhan sistem, perancangan sistem, pembuatan sistem, percobaan, implementasi dan evaluasi.

3. Metode Wawancara

Metode wawancara artinya penulis mengadakan wawancara langsung dengan pelaku bisnis pulsa *multi operator*.

1.7 Sistematika Penulisan

Sistematika penulisan laporan Tugas Akhir ini terdiri dari beberapa bab dan sub bab sebagai berikut;

1. BAB I : PENDAHULUAN

Pada pendahuluan diberikan gambaran umum tentang laporan yang berisi latar belakang masalah, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

2. BAB II : LANDASAN TEORI

Bab ini menjelaskan landasan teoritis yang digunakan oleh penulis dalam pembuatan tugas akhir. Landasan teori ini diperoleh pada saat penelitian studi pustaka. Dasar-dasar teori tersebut berdasarkan penjelasan mengenai sistem yang akan dibuat dan metode pemecahan masalah pada sistem yang diterapkan pada tugas akhir ini. Bab ini juga berisikan tentang kajian pustaka.

3. BAB III : DESAIN DAN PERANCANGAN SISTEM

Bab ini menerangkan mengenai perancangan dan desain sistem dari tugas akhir yang dibangun oleh penulis. Perancangan dan desain sistem tersebut meliputi *Use case Diagram*, *Activity Diagram*, *Sequence Diagram*, dan *Class Diagram*..

4. BAB IV : IMPLEMENTASI DAN EVALUASI

Pada implementasi berisikan mengenai penjelasan dari desain antarmuka aplikasi yang dibuat penulis. Pada evaluasi memuat kekurangan dan kelebihan sistem.

5. BAB V : PENUTUP

Pada penutup terdapat kesimpulan hasil penelitian dan pembahasan beserta saran yang ditujukan untuk penyempurnaan dan pengembangan sistem di masa mendatang.

BAB II LANDASAN TEORI

2.1 Bahasa Pemrograman JAVA

JAVA adalah sebuah bahasa pemrograman komputer berbasis pada *Object Oriented Programming*. Java diciptakan setelah C++ dan didesain sedemikian sehingga ukurannya kecil, sederhana, dan *portable* (dapat dipindah-pindahkan di antara bermacam *platform* dan sistem operasi).

Salah satu keunggulan Java adalah sifatnya yang *platform independence*, artinya baik *source* program maupun hasil kompilasinya sama sekali tidak bergantung kepada sistem operasi dan *platform* yang digunakan. *Source code* sebuah aplikasi dengan bahasa Java yang ditulis di atas sistem *Windows NT* misalnya, dapat dipindahkan ke sistem operasi UNIX tanpa harus mengubah satu baris kode pun. Ini tentunya merupakan satu nilai tambah tersendiri.

Generasi yang saat ini sedang berkembang dari *platform Java* ialah *Java 2*. Agar sebuah program Java dapat dijalankan, maka file dengan ekstensi Java harus dikompilasi menjadi file *bytecode*. Untuk menjalankan *bytecode* ini, dibutuhkan JRE (*Java Runtime Environment*) yang memungkinkan pemakai untuk menjalankan program Java, hanya menjalankan, dan tidak untuk membuat kode baru lagi. JRE berisi JVM dan *library Java* yang digunakan.

Java Platform dibagi menjadi 3 kategori, yaitu :

a. *Java Standard Edition (JSE)*

Kategori ini digunakan untuk mengembangkan dan menjalankan aplikasi Java berbasis PC.

b. *Java Enterprise Edition (JEE)*

Kategori ini digunakan untuk mengembangkan dan menjalankan aplikasi Java pada lingkungan *enterprise*, dengan fungsi-fungsi seperti *Enterprise Java Bean (EJB)*, *Servlet*, dan *Java Server Page (JSP)*.

c. *Java Micro Edition (JME)*

Kategori ini digunakan untuk mengembangkan dan menjalankan aplikasi *java* berbasis *handl 5 vice*, seperti *Personal Digital Assistant (PDA)*, *handphone*, dan *pocketPC*.

(Andy Prasetyo Utomo, 2009)

2.2 Pengertian DBMS *MySQL*

DBMS adalah suatu perangkat lunak yang digunakan untuk mengelola, membuat, menghapus, memasukan data, dan melakukan manajemen *database* lainnya. Salah satu DBMS yang handal adalah *MySQL*. *MySQL* merupakan salah satu DBMS (*Dababase Management System*) yang menggunakan stuktur *database* relasional (RDBMS).

Java telah menyediakan berbagai fitur untuk pengembangan aplikasi *database*. Sehingga hampir semua DBMS (*Dababase Management System*) populer yang ada saat ini sudah *disupport* penuh oleh *Java*, tidak terkecuali *MySQL*, *Microsoft SQL Server*, dan *Oracle*. Banyak sekali cara yang digunakan untuk melakukan koneksi dengan *database*, dan di antara cara-cara tersebut yang paling populer adalah koneksi melalui JDBC, *JDBC-ODBC Bridge* (disebut juga via ODBC), an XML, tentunya dengan kelebihan dan kekurangan masing-masing terhadap yang ditangani atau permasalahan yang akan diselesaikan.

Secara umum langkah-langkah yang dilakukan dalam bahasa pemrograman *database* adalah:

- a. Melakukan *loading database driver*.
- b. Membangun koneksi.
- c. Melakukan *statement*.
- d. Mengambil data dengan *ResultSet*.

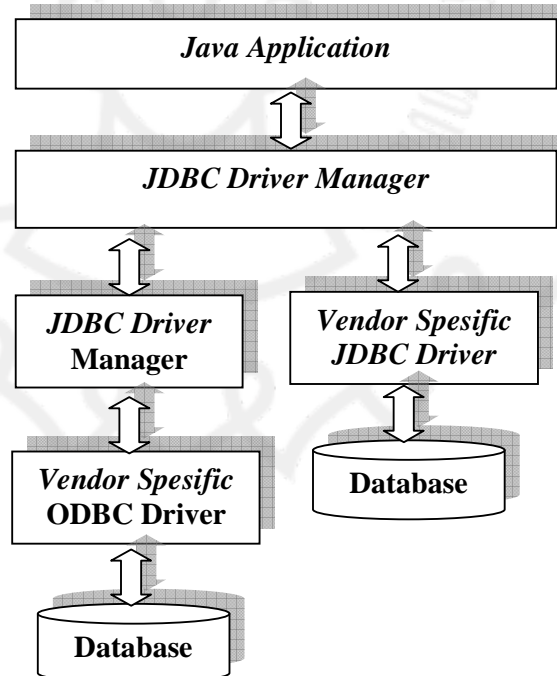
(Wahana Komputer, 2010)

2.3 Koneksi Via *Java Database Connectivity (JDBC)*

Java Database Connectivity (JDBC) berfungsi untuk melakukan koneksi *database* dengan aplikasi yang ditulis dengan bahasa pemrograman *Java*. Bentuk fisik dari JDBC adalah file *driver* yang berisi *class* atau kumpulan *class-class* (termasuk *interface*, *exception* dan juga spesifikasi baik bagi *vendor driver* JDBC maupun *Driver Manager* bagi pengembangan aplikasi yang memanfaatkan JDBC) yang digunakan untuk menghubungkan *Java* dengan *database* tertentu, sehingga apabila *database* yang akan dikoneksikan berbeda, maka file *driver* yang digunakan juga berbeda, meskipun jenis koneksinya sama, yaitu JDBC.

2.3.1 Arsitektur JDBC

Sebuah spesifikasi yang terbuka, seperti JDBC yang dispesifikasikan oleh *Sun Microsystem*, harus dapat bersifat terbuka bagi para *vendor* lain, terutama dalam pendefinisian *driver* JDBC untuk masing-masing RDBMS. Untuk itu JDBC disusun dalam sebuah arsitektur sebagai berikut:



Gambar 2.1 Arsitektur JDBC

Lapisan *Vendor Specific JDBC Driver* merupakan *driver JDBC* yang dikeluarkan oleh para *vendor* pengembang RDBMS. Sedangkan *JDBC-ODBC Bridge* berfungsi sebagai perantara untuk mengakses *database* melalui *ODBC driver*. Baik *JDBC driver* maupun *JDBC-ODBC Bridge* diatur dan diakses melalui *JDBC Driver Manager*.

Aplikasi yang dikembangkan untuk mengakses *database* dengan memanfaatkan JDBC akan berinteraksi dengan *JDBC Driver Manager*. Arsitektur ini memudahkan pemrogram untuk mengakses *database*.

JDBC API tersedia dalam paket *java.sql* dan *javax.sql*. Berikut ini adalah *class-class*, *interface*, dan *exception JDBC* penting yang sering digunakan selama pemrograman:

- a. *Driver Manager* : memanggil *driver JDBC* ke memori, dan juga digunakan untuk membuka koneksi ke sumber data.
- b. *Connection* : mempresentasikan suatu koneksi dengan suatu *data source*, juga digunakan untuk membuat objek *Statement*, *PreparedStatement*, dan *CallableStatement*.
- c. *Statement* : mempresentasikan suatu perintah *SQL*, dan dapat digunakan untuk menerima objek *ResultSet*.
- d. *PreparedStatement* : merupakan *alternative* untuk objek *Statement SQL* yang telah dikompilasi awal.

(Wahana Komputer, 2010)

2.3.2 Koneksi dengan *MySQL*

Seperti pada umumnya koneksi ke *database* dengan JDBC, maka perlu meng-*include*-kan (dengan perintah *import*) paket berupa *java.sql*, kemudian diikuti dengan penggunaan *class* yang ada pada *driver JDBC for MySQL*.

Kode program yang digunakan untuk *loading driver JDBC for MySQL* adalah sebagai berikut:

```
Class.forName("com.mysql.jdbc.Driver");
```

9

Sedangkan kode program untuk melakukan koneksi ke *database MySQL* adalah sebagai berikut:

```
Connectin conn = DriverManager.getConnection  
("jdbc:mysql://localhost:3306/pulsa", "root", "root");
```

(Wahana Komputer, 2010)

2.4 Pengertian Gammu SMS Server

Gammu merupakan aplikasi yang berbasis *open source* yang digunakan sebagai konektor atau penghubung antara komputer dengan *Handphone*. *Gammu* berjalan menggunakan *command line*. Aplikasi *Gammu* ini dapat digunakan untuk mengirim dan menerima SMS dari komputer via *handphone* atau modem GSM.

Gammu pada awalnya merupakan aplikasi yang dikembangkan dari *Gnokii*, dan sampai sekarang perkembangan *Gammu* berkembang secara bebas. *Gammu* mendukung *handphone* yang bekerja dengan *AT command*. *Gammu* pada dasarnya mendukung semua tipe *handphone* dari produk Sony Ericsson, karena hampir semua tipe Sony Ericsson menggunakan *protokol AT command* untuk komunikasi dengan komputer. Namun *Gammu* juga tidak menutup kemungkinan untuk *handphone* dari produk lain meski tidak menggunakan *AT Command*.

(www.novit.groja.info)

2.5 Pengertian Unified Modelling Language (UML)

Unified Modelling Language (UML) adalah bahasa standar untuk melakukan spesifikasi, visualisasi, konstruksi dan dokumentasi dari komponen-komponen perangkat lunak dan digunakan untuk pemodelan bisnis. UML menggunakan notasi grafis untuk melakukan suatu desain. Pemodelan UML berarti menggambarkan yang ada dalam dunia nyata ke dalam bentuk yang dapat dipahami dengan menggunakan notasi standar

UML. Pemodelan UML terdiri dari 8 tipe diagram yang berbeda untuk memodelkan sistem perangkat lunak. Masing-masing diagram (10) didesain untuk menunjukkan satu sisi dari bermacam-macam sudut pandang (*perspektif*) dan terdiri dari tingkat abstraksi yang berbeda.

Kedelapan tipe diagram itu adalah:


- a. *Use case Diagram*
- b. *Class Diagram*
- c. *Object Diagram*
- d. *State Diagram*
- e. *Activity Diagram*
- f. *Sequence Diagram*
- g. *Collaboration Diagram*
- h. *Component Diagram*
- i. *Deployment Diagram*



(Tessy Badriyah, 2007)

2.6 Pengertian *Use Case Diagram*

Use Case digunakan pada saat pelaksanaan tahap *requirement* dalam pengembangan suatu sistem informasi. *Use Case* menggambarkan hubungan antara *entitas* yang biasa disebut *actor* dengan suatu proses yang dapat dilakukannya. Simbol-simbol yang digunakan dalam *Use Case* beserta deskripsinya dapat dilihat pada Tabel 2.2.

Tabel 2.1 Simbol *Use Case Diagram*

No.	Simbol	Nama	Deskripsi
1		<i>Case</i>	Menggambarkan proses/ kegiatan yang dapat dilakukan oleh actor

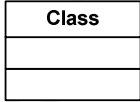
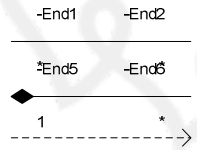
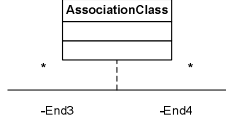
2		<i>Actor</i>	Menggambarkan <i>entitas/sub</i> 11 yang dapat melakukan suatu proses
3		<i>Relation</i>	Relasi antara <i>case</i> dengan <i>actor</i> ataupun <i>case</i> dengan <i>case</i> lain.

(Doug Rosenberg, Scot Kendall. 2001)

2.7 Pengertian *Class Diagram*

Class/Static Diagram digunakan untuk menggambarkan struktur kelas dan obyek yang akan digunakan dalam sistem yang akan dibangun. *Static Diagram* digunakan pada tahap analisa dan desain aplikasi. Simbol-simbol yang digunakan dalam *Static Diagram* dapat dilihat pada Tabel 2.3.

Tabel 2.2 Simbol *Class Diagram*

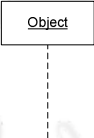
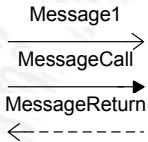
No.	Simbol	Nama	Deskripsi
1		<i>Class</i>	Menggambarkan sebuah kelas yang terdiri dari atribut dan <i>method</i>
2		<i>Relation</i>	Menggambarkan hubungan komponen-komponen didalam <i>Static Diagram</i> .
3		<i>Association Class</i>	<i>Class</i> yang terbentuk dari hubungan antara dua buah <i>Class</i>

(Doug Rosenberg, Scot Kendall. 2001)

2.8 Pengertian *Sequence Diagram*

Sequence Diagram digunakan untuk menjelaskan aliran pesan 12 suatu *Class* ke *Class* lain secara *sequensial* (berurutan). *Sequence Diagram* digunakan pada tahap desain aplikasi. Simbol yang digunakan dalam *Sequence Diagram* dapat dilihat pada Tabel 2.5.

Tabel 2.3 Simbol *Sequence Diagram*

No.	Simbol	Nama	Deskripsi
1		<i>Object</i>	Menggambarkan pos-pos obyek yang pengirim dan penerima message
2		<i>Message</i>	Menggambarkan aliran pesan yang dikirim oleh pos-pos obyek

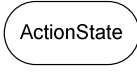

(Doug Rosenberg, Scot Kendall. 2001)

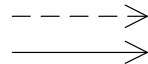




2.9 Pengertian *Activity Diagram*

Activity Diagram menggambarkan berbagai alur aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alur berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir.

Activity Diagram digunakan untuk menjelaskan tanggung jawab elemen. *Activity Diagram* biasa dikolaburasikan dengan *Sequence Diagram* dalam pendiskripsian visual dari tahap desain aplikasi. Simbol-simbol yang digunakan dalam *Activity Diagram* dapat dilihat pada Tabel 2.6

Tabel 2.4 Simbol *Activity Diagram*

No.	Simbol	Nama	Deskripsi
1		<i>Action State</i>	Menggambarkan keadaan dari suatu elemen dalam suatu aliran aktifitas
2		<i>State</i>	Menggambarkan kondisi suatu elemen.

3		<i>Flow Control</i>	Menggambarkan aliran aktifit 13 dari suatu elemen ke elemen lain
4		<i>Initial State</i>	Menggambarkan titik awal siklus hidup suatu elemen
5		<i>Final State</i>	Menggambarkan titik akhir yang menjadi kondisi akhir suatu elemen
6		<i>Transition Fork</i>	Menggambarkan aliran siklus state (kondisi) suatu elemen.
7		<i>Transition Join</i>	Menggambarkan aliran siklus state (kondisi) suatu elemen

(Doug Rosenberg, Scot Kendall. 2001)

2.10 Pengertian Apache

Apache adalah salah satu aplikasi *web server*, *apache* bertugas menerjemahkan *Uniform Resource Locator (URL)* menuju *file*, kemudian mengirimkan *file* tersebut melalui internet atau ke program yang kemudian dijalankan oleh program tersebut dan mengirimkan hasilnya (Laurie, B and Laurie, P, 2002). Kelebihan-kelebihan *Apache* menurut Kabir (2002) adalah:

- Apache is highly configurable web server with a modular design* (didesain dengan sistem modular dan dapat dikonfigurasi)
- Apache is free, open source technology* (gratis dan *source code* diberikan secara bebas)
- Apache works great with Perl, PHP and other scripting language* (dapat bekerja dengan *Perl*, *PHP* dan bahasa *script* lainnya)
- Apache runs on linux and other unix system* (dapat berjalan pada *linux* dan *system unix* lainnya)
- Apache also runs on windows* (dapat juga berjalan pada *system operasi windows*).

BAB III

DESAIN DAN PERANCANGAN

3.1 Perancangan Sistem

Pembuatan aplikasi penjualan pulsa *multi operator* ini dibuat dengan menggunakan *java jdk1.6.0* dengan *editor Java J2SE* dan *database MySQL*. Dengan menggunakan *software* tersebut aplikasi ini diharapkan dapat membantu perekapan data penjualan pada setiap *counter* penjualan pulsa *multi-operator*.

3.2 Model Analisis

Pengembangan aplikasi berorientasi objek memerlukan beberapa tahapan analisa yang harus dilalui. Pada tahap ini dilakukan pemodelan menggunakan UML. Untuk menggunakan UML ada beberapa hal yang harus dilakukan antara lain:

1. Membuat *use case diagram* untuk memetakan memetakan kemungkinan skenario yang terjadi.
2. Skenario untuk mendefinisikan secara tepat *fungsiionalitas* yang harus dimiliki sistem.
3. Dari skenario yang didapat maka dirancanglah *sequence diagram* dari setiap *use case* yang terjadi.
4. Dari *sequence diagram* tersebut, dibuat lah *class diagram* yang langsung pada program aplikasi ini.

3.2.1 Use case Diagram

a. Definisi Aktor

Dalam aplikasi ini terdapat dua aktor yang berperan dalam menjalankan dan menggunakan aplikasi ini, *admin* dan *operator*.

Tabel 3.1 Definisi Aktor

No	Aktor	Definisi
1.	<i>Admin</i>	<i>Admin</i> merupakan pengguna yang memiliki hak untuk menggunakan semua menu yang ada di aplikasi ini. Mengetahui semua <i>database</i> dari

		aplikasi penjualan ini tersebut dan merekap semua data apabila diperlukan.
2.	<i>Operator</i>	<i>Operator</i> merupakan pengguna aplikasi ini tetapi tidak dapat menggunakan semua menu yang tersedia pada aplikasi ini.

b. Definisi *Use case*

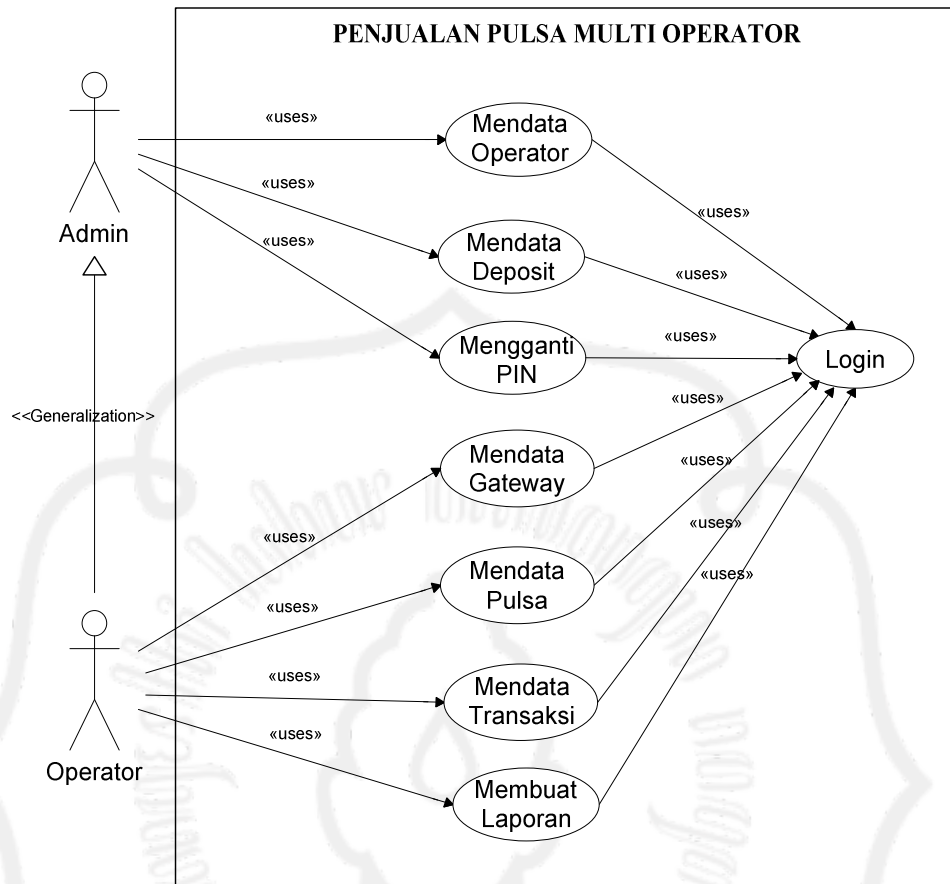
Tabel 3.2 Definisi *Use case*.

No.	<i>Use case</i>	Definisi
1.	Mendata Operator	Pendataan ini digunakan digunakan oleh <i>admin</i> untuk mendata <i>operator</i> (pengguna) yang mempunyai hak untuk menggunakan aplikasi ini. Setiap <i>operator</i> yang telah terdaftar akan memiliki sebuah <i>account</i> yang terdiri dari <i>username</i> dan <i>password</i> untuk dapat mengakses aplikasi ini.
2.	<i>Login</i>	<i>Admin</i> atau <i>operator</i> dapat menggunakan aplikasi ini setelah melakukan proses <i>login</i> dengan memasukkan <i>username</i> dan <i>password</i> yang mereka miliki,.
3.	Menambah Deposit	<i>Use case</i> pendataan deposit merupakan <i>session</i> dimana <i>admin</i> menambah deposit yang dimiliki. <i>Admin</i> dan <i>operator</i> dapat melakukan penambahan dan pengecekan deposit.
4.	Mengganti PIN	<i>Use case</i> penggantian PIN ini merupakan <i>session</i> dimana <i>admin</i> mengganti pin yang digunakan sebagai <i>format</i> pengiriman.
5	Mendata pulsa	Pada <i>use case</i> ini <i>admin</i> dan <i>operator</i> mendata pulsa yang dijual oleh <i>server</i> pulsa tempat mereka membeli deposit. Pendataan pulsa ini

		meliputi penambahan data pulsa dan pengeditan data pulsa.
6	Mendata gateway	Pada <i>use case</i> ini <i>admin</i> dan <i>operator</i> mendata gateway yang digunakan sebagai nomor <i>server</i> pengiriman pulsa. Pendataan gateway ini meliputi penambahan data gateway dan pengeditan data gateway.
7	Mendata Transaksi	Pada <i>use case</i> ini <i>admin</i> dan <i>operator</i> mendata penjualan pulsa. Dalam penjualan ini <i>admin</i> atau <i>operator</i> dapat menambah data penjualan dengan menginputkan data kemudian memilih gateway sebagai tujuan pengiriman. <i>Admin</i> atau <i>operator</i> juga dapat melakukan pengeditan data penjualan.
8	Membuat Laporan	Pada <i>use case</i> ini <i>admin</i> atau <i>operator</i> melakukan pencetakan laporan penjualan atau data pulsa.

c. Diagram Use case

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem, yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah *use case* dapat meng-include fungsionalitas *use case* lain sebagian bagian proses dari dirinya, secara umum diasumsikan bahwa *use case* yang di-include akan dipanggil setiap kali *use case* yang meng-include dieksekusi secara normal.



Gambar 3.1 Use Case Diagram Aplikasi Penjualan Pulsa.

3.2.2 Skenario *Use case*

Dari *use case* yang terbentuk, maka didapatkan skenario dari *use case* tersebut. Skenario dari *use case* ini merupakan penjabaran pengertian dari *use case* itu sendiri. Di skenario *use case* ini dapat diketahui secara detail setiap langkah yang terjadi pada *use case* tersebut.

Tabel 3.3 Skenario *Login*

Use case : Login	
Kondisi : Normal	
Aktor	Sistem
1. Melakukan <i>login</i> (Memasukkan <i>username</i> dan <i>password</i>)	2. Koneksi <i>database</i> berhasil. 3. Mengecek kelengkapan pengisian <i>username</i> dan <i>password</i> . 4. Mengecek tersedianya <i>username</i> dan <i>password</i> pada tabel <i>operator</i> .
5. <i>Login</i> berhasil	
Kondisi : Koneksi Gagal	
	3. Koneksi <i>database</i> gagal. 4. Penyampaian pesan <i>error</i> .
5. Mengulang langkah 1	
Kondisi : Username dan Password Belum Terisi.	
	4. Pengisian kurang lengkap 5. Penyampaian pesan <i>error</i> .
6. Mengulang langkah 1	
Kondisi : Username dan Password Tidak Tersedia.	
	5. <i>Username</i> dan <i>password</i> tidak tersedia 6. Penyampaian pesan <i>error</i> .
7. Mengulang langkah 1	

Tabel 3.4 Skenario Mendata *Operator*.

Use case : Mendata Operator	
Kondisi : Normal	
Aktor	Sistem
1. Admin menambah atau mengedit data <i>operator</i> .	2. Mengecek pengisian data sudah lengkap. 3. Pengecekan <i>username</i> sudah ada atau belum. 4. Menyimpan ke tabel <i>operator</i> . 5. Data <i>operator</i> baru tampil pada tabel.
Kondisi : Tidak Terisi Semua	
	3. Pengisian kurang lengkap 4. Penyampaian pesan <i>error</i> .
5. Mengulang langkah 1	
Kondisi : Username Sudah Ada	
	4. <i>Username</i> sudah ada 5. Penyampaian pesan <i>error</i> .
6. Mengulang langkah 1	
Kondisi : Gagal Memasukkan Data ke Tabel	
	5. Gagal memasukkan ke tabel. 6. Penyampaian pesan <i>error</i> .
7. Mengulang langkah 1	

Tabel 3.5 Skenario Menambah Deposit

Use case : Menambah Deposit	
Kondisi : Normal	
Aktor	Sistem
1. Admin menambah deposit.	2. Mengecek pengisian data sudah lengkap.

	<ol style="list-style-type: none"> 3. Menambahkan deposit lama dengan yang baru. 4. Menyimpan ke tabel deposit. 5. Data deposit tampil pada tabel.
Kondisi : Tidak Terisi Semua	
	<ol style="list-style-type: none"> 3. Pengisian kurang lengkap 4. Penyampaian pesan <i>error</i>.
5. Mengulang langkah 1	
Kondisi : Gagal Memasukkan Data ke Tabel	
	<ol style="list-style-type: none"> 5. Gagal memasukkan ke tabel 6. Penyampaian pesan <i>error</i>.
7. Mengulang langkah 1	

Tabel 3.6 Skenario Mengganti PIN

Use case : Mengganti PIN	
Kondisi : Normal	
Aktor	Sistem
1. Admin mengganti PIN.	<ol style="list-style-type: none"> 2. Mengecek pengisian data sudah lengkap. 3. Mengecek kesamaan pin lama. 4. Menyimpan ke tabel pin. 5. Data deposit tampil pada tabel.
Kondisi : Tidak Terisi Semua	
	<ol style="list-style-type: none"> 3. Pengisian kurang lengkap 4. Penyampaian pesan <i>error</i>.
5. Mengulang langkah 1	
Kondisi : Pin Tidak Sama	
	<ol style="list-style-type: none"> 4. Pin tidak sesuai. 5. Penyampaian pesan <i>error</i>.

6.	
Kondisi : Gagal Memasukkan Data ke Tabel	
	8. Gagal memasukkan ke tabel. 9. Penyampaian pesan <i>error</i> .
10. Mengulang langkah 1	

Tabel 3.7 Skenario Mendata Pulsa

Use case : Mendata Pulsa	
Kondisi : Normal	
Aktor	Sistem
1. Admin atau operator menambah atau mengedit data pulsa.	2. Mengecek pengisian data sudah lengkap atau belum. 3. Pengecekan id pulsa sudah ada atau belum. 4. Menyimpan ke tabel pulsa. 5. Data <i>operator</i> baru tampil pada tabel.
Kondisi : Tidak Terisi Semua	
	3. Pengisian kurang lengkap 4. Penyampaian pesan <i>error</i> .
5. Mengulang langkah 1	
Kondisi : Id Pulsa Sudah Ada	
	4. Id Pulsa sudah ada 5. Penyampaian pesan <i>error</i> .
6. Mengulang langkah 1	
Kondisi : Gagal Memasukkan Data ke Tabel	
	5. Gagal memasukkan data ke tabel. 6. Penyampaian pesan <i>error</i> .
7. Mengulang langkah 1	

Tabel 3.8 Skenario Mendata *Gateway*

Use case : Mendata <i>Gateway</i>	
Kondisi : Normal	
Aktor	Sistem
1. Admin atau operator menambah atau mengedit data <i>gateway</i> .	2. Mengecek pengisian data sudah lengkap atau belum. 3. Menyimpan ke tabel <i>gateway</i> . 4. Data <i>operator</i> baru tampil pada tabel.
Kondisi : Tidak Terisi Semua	
	3. Pengisian kurang lengkap 4. Penyampaian pesan <i>error</i> .
5. Mengulang langkah 1	
Kondisi : Gagal Memasukkan Data ke Tabel	
	4. Gagal memasukkan data ke tabel. 5. Penyampaian pesan <i>error</i> .
6. Mengulang langkah 1	

Tabel 3.9 Skenario Mendata Transaksi

Use case : Mendata Transaksi	
Kondisi : Normal	
Aktor	Sistem
1. Admin atau operator menambah atau mengedit data penjualan.	2. Mengecek pengisian data sudah lengkap atau belum. 3. Mengecek stok deposit. 4. Menyimpan ke tabel penjualan. 5. Data <i>operator</i> baru tampil pada tabel.
Kondisi : Tidak Terisi Semua	
	3. Pengisian kurang lengkap

	4. Penyampaian pesan <i>error</i> .
2. Mengulang langkah 1	
Kondisi : Stok Tidak tersedia	
	4. Pengisian kurang lengkap 5. Penyampaian pesan <i>error</i> .
3. Mengulang langkah 1	
Kondisi : Gagal Memasukkan Data ke Tabel	
	8. Gagal memasukkan data ke tabel. 9. Penyampaian pesan <i>error</i> .
4. Mengulang langkah 1	

Tabel 3.10 Skenario Membuat Laporan

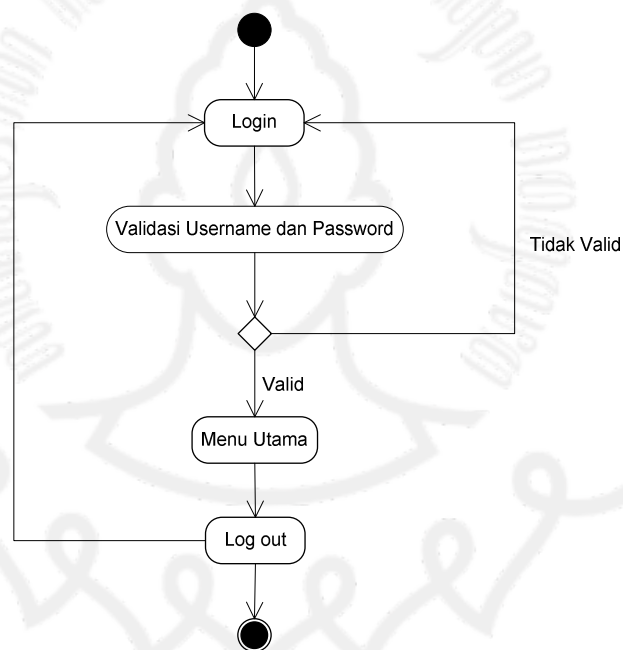
Use case : Membuat Laporan	
Kondisi : Normal	
Aktor	Sistem
1. Admin atau operator membuat laporan (memilih data laporan).	2. Mengecek data yang terpilih tersedia 3. Menampilkan data yang terpilih pada layar. 4. Mencetak laporan
Kondisi : Data Tidak Tersedia	
	3. Data tidak tersedia. 4. Tampilan kosong.
5. Mengulang langkah 1	
Kondisi : Data Tidak Tampil	
	4. Gagal mengambil data dari database. 5. Penyampaian pesan <i>error</i> .

6. Mengulang langkah 1

3.2.3 Activity Diagram

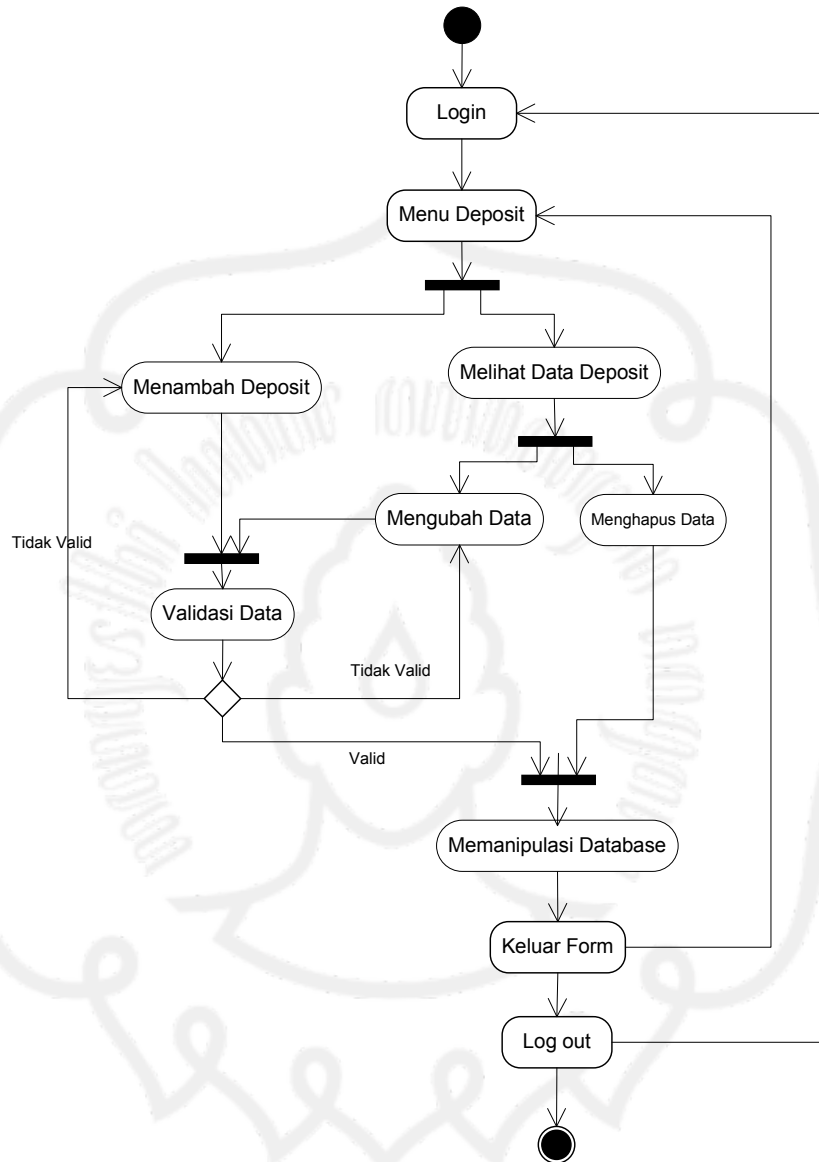
Activity Diagram menggambarkan berbagai alur aktifitas dalam *system* yang sedang dirancang, bagaimana masing-masing alur berawal, *decision* yang terjadi, dan bagaimana sistem akan berakhir. *Activity Diagram* juga menggambarkan proses *parallel* yang mungkin terjadi pada beberapa eksekusi.

Berikut *Activity Diagram* dari aplikasi penjualan pulsa ini:

1. *Activity Diagram Login*

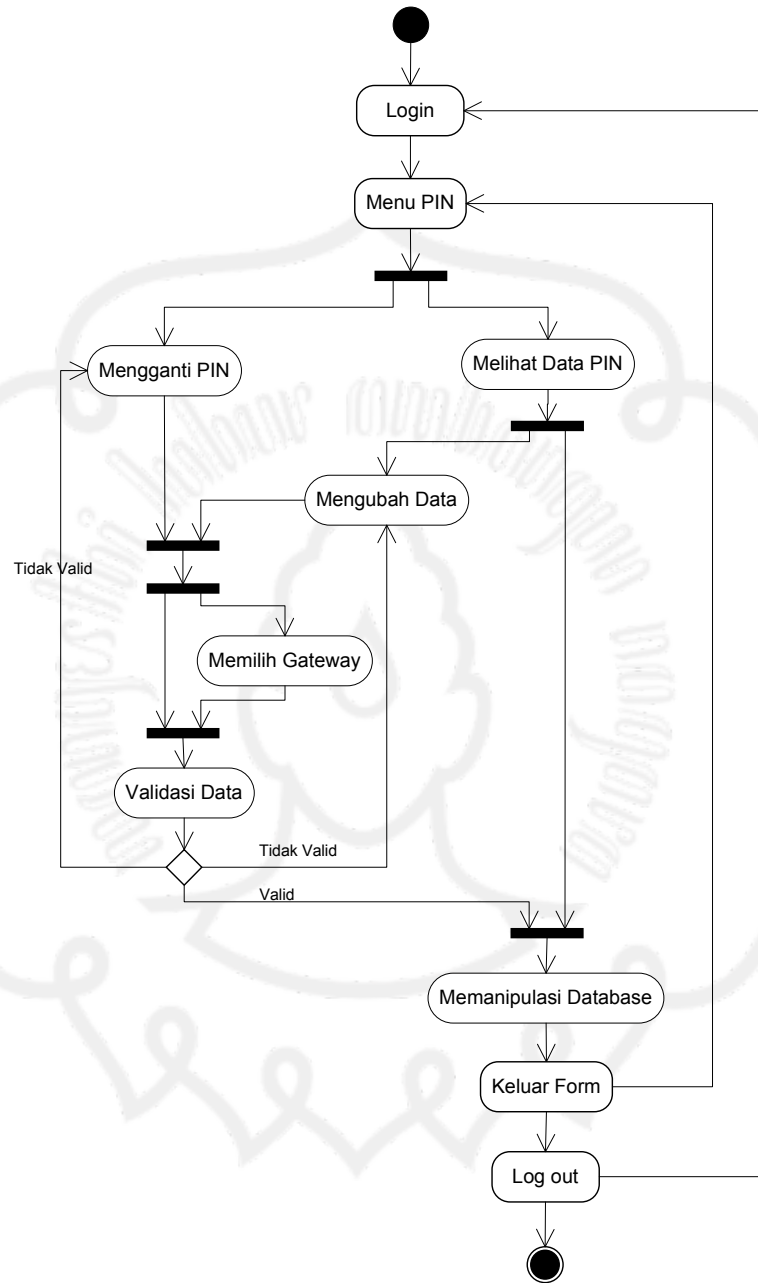
Gambar 3.2 *Activity Diagram Login*

2. Activity Diagram Menambah Deposit



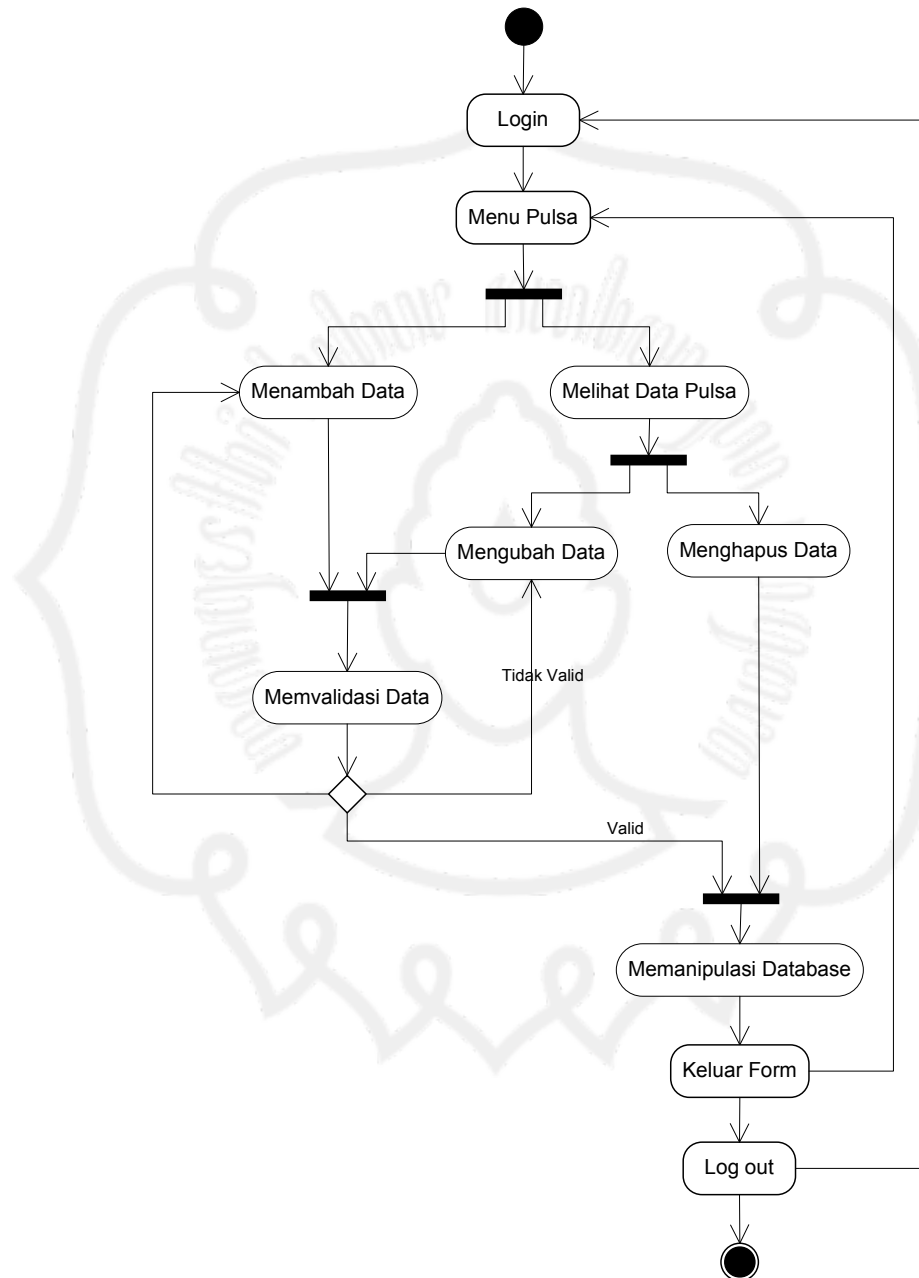
Gambar 3.3 Activity Diagram Menambah Deposit

3. Activity Diagram Mengganti PIN



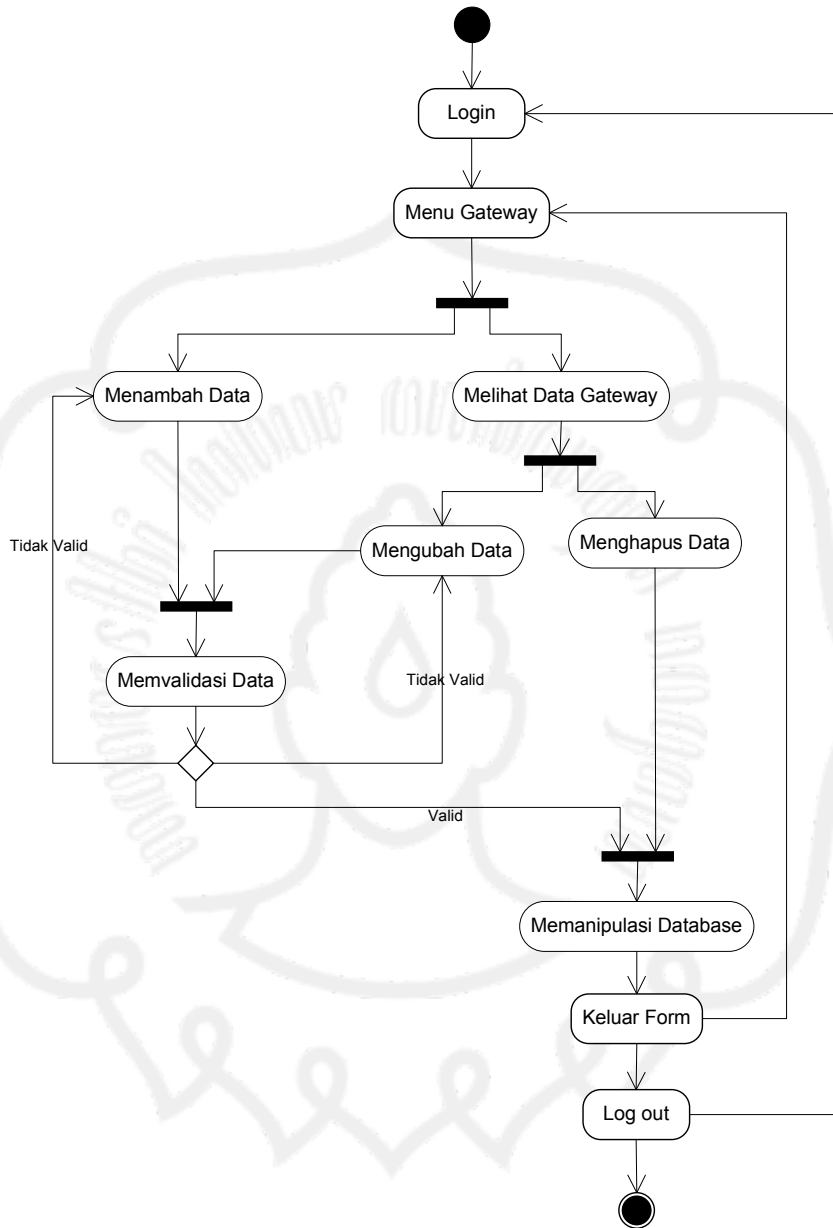
Gambar 3.4 Activity Diagram Mengganti PIN

4. Activity Diagram Mendata Pulsa



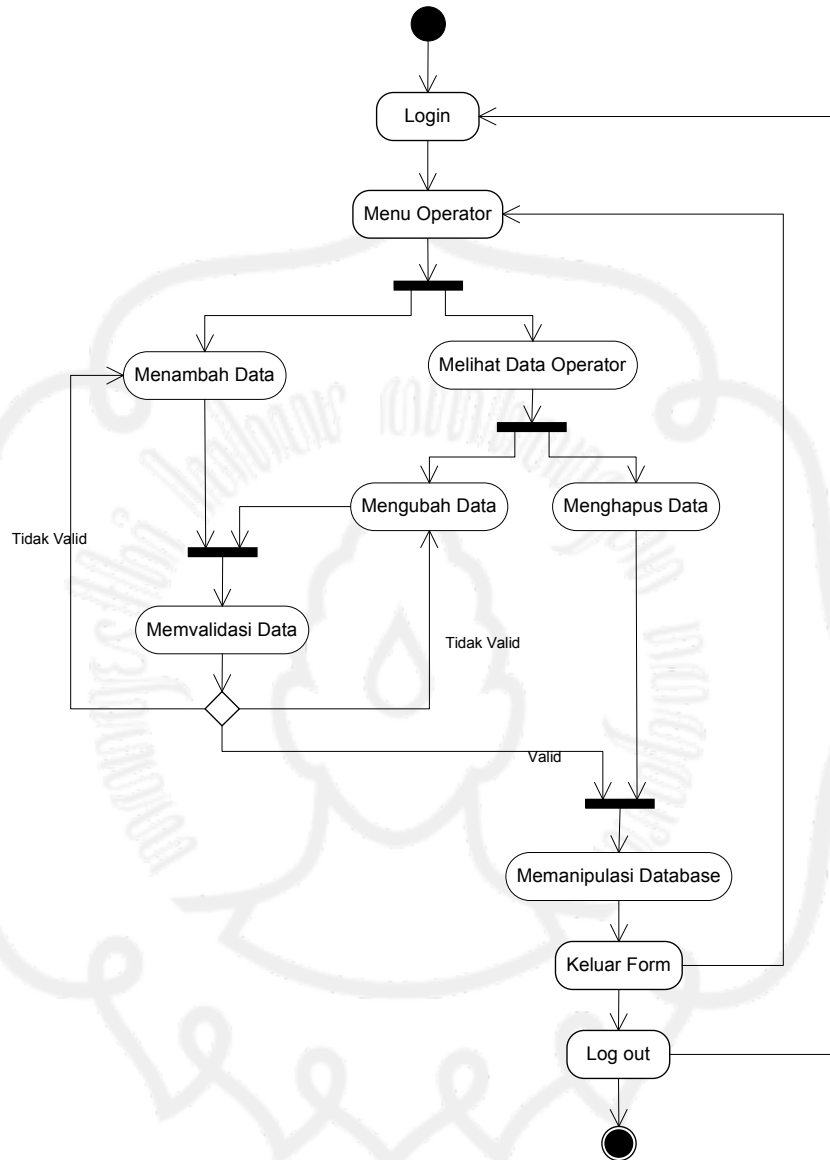
Gambar 3.5 Activity Diagram Mendata Pulsa

5. Activity Diagram Mendata Gateway



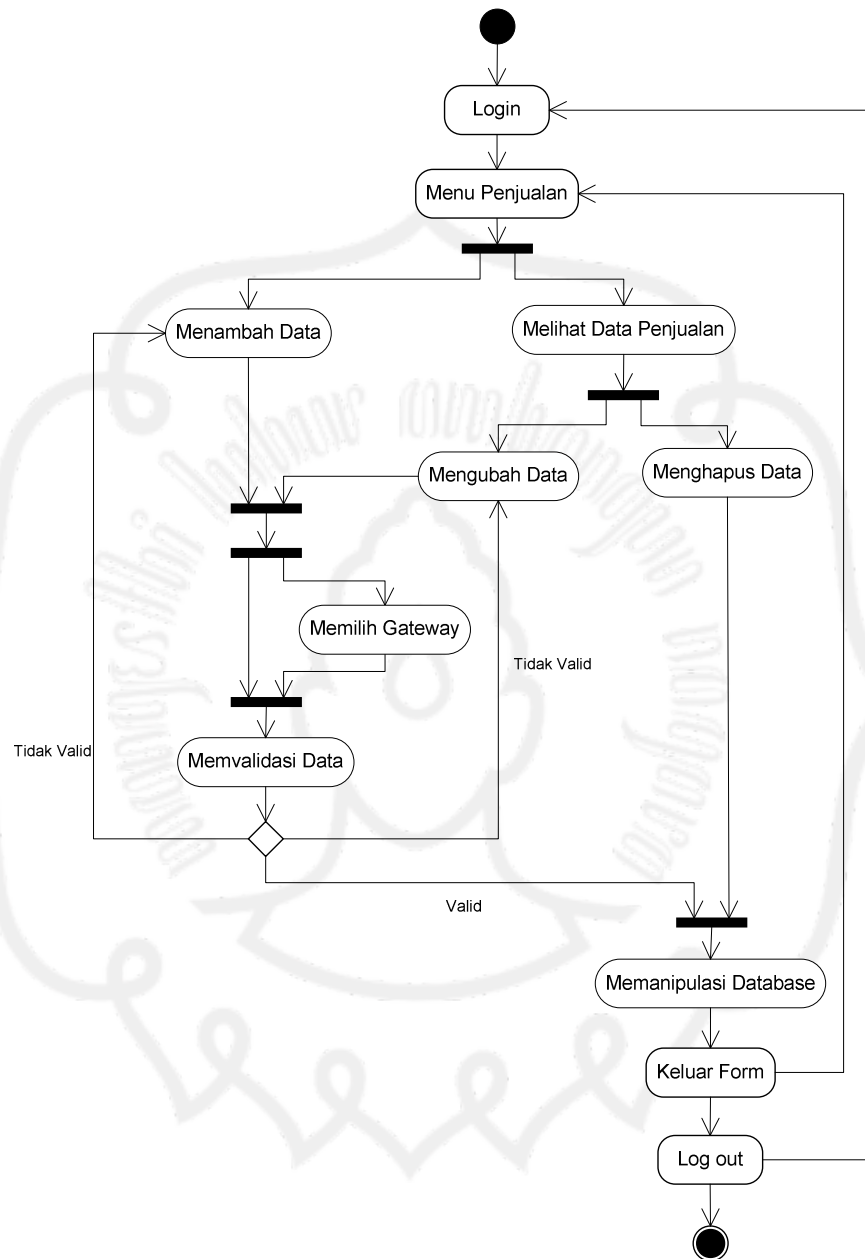
Gambar 3.6 Activity Diagram Mendata Gateway

6. Activity Diagram Mendata Operator

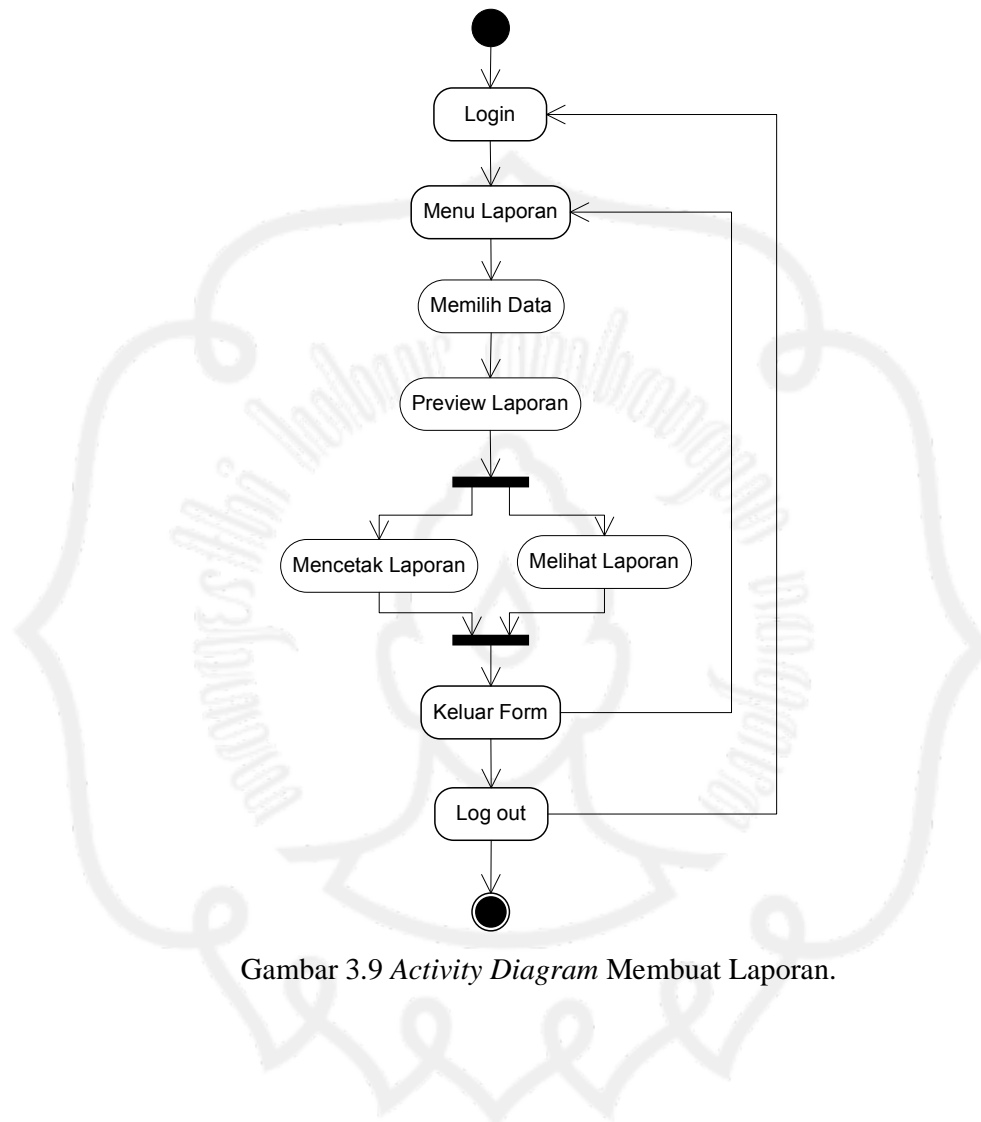


Gambar 3.7 Activity Diagram Mendata Operator

7. Activity Diagram Mendata Transaksi



Gambar 3.8 Activity Diagram Mendata Transaksi

8. *Activity Diagram* Membuat Laporan.Gambar 3.9 *Activity Diagram* Membuat Laporan.

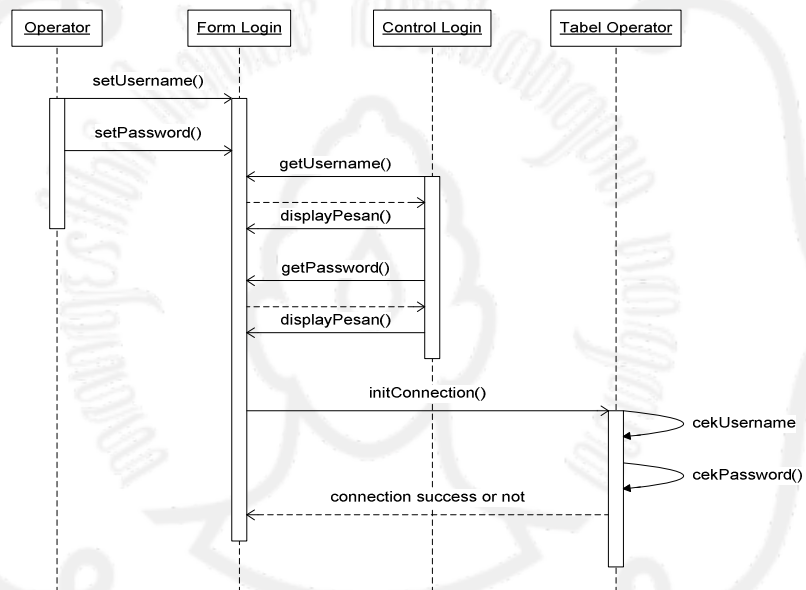
3.3 Model Perancangan

Setelah menyelesaikan tahap analisis, maka langkah selanjutnya adalah mengidentifikasi *sequence diagram*.

3.3.1 Sequence Diagram

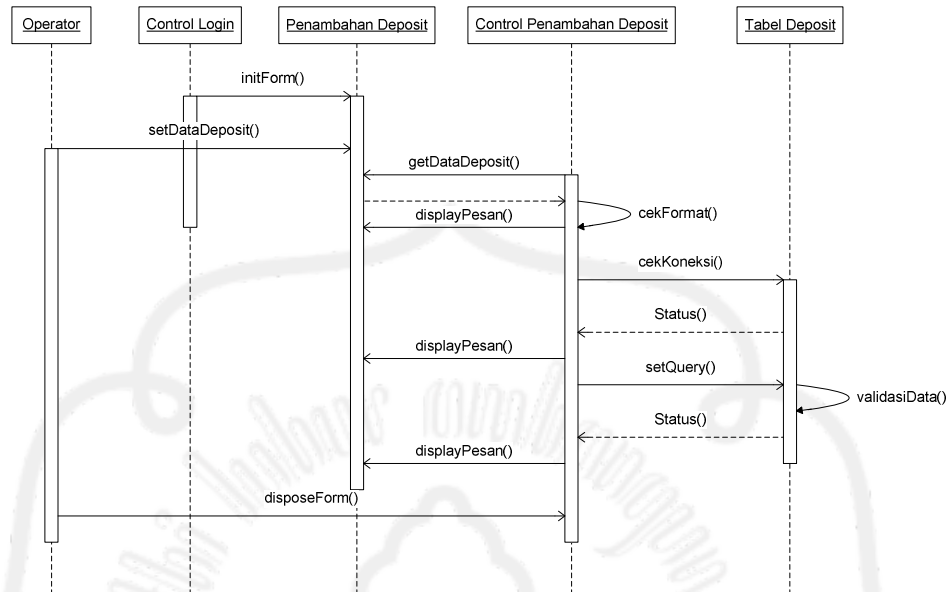
Setelah menganalisa *Activity Diagram* maka *sequence diagram* dari aplikasi ini dapat digambar, berikut *sequence diagram* yang dapat diambil dari penggambaran *Activity Diagram*:

1. Sequence Diagram Login



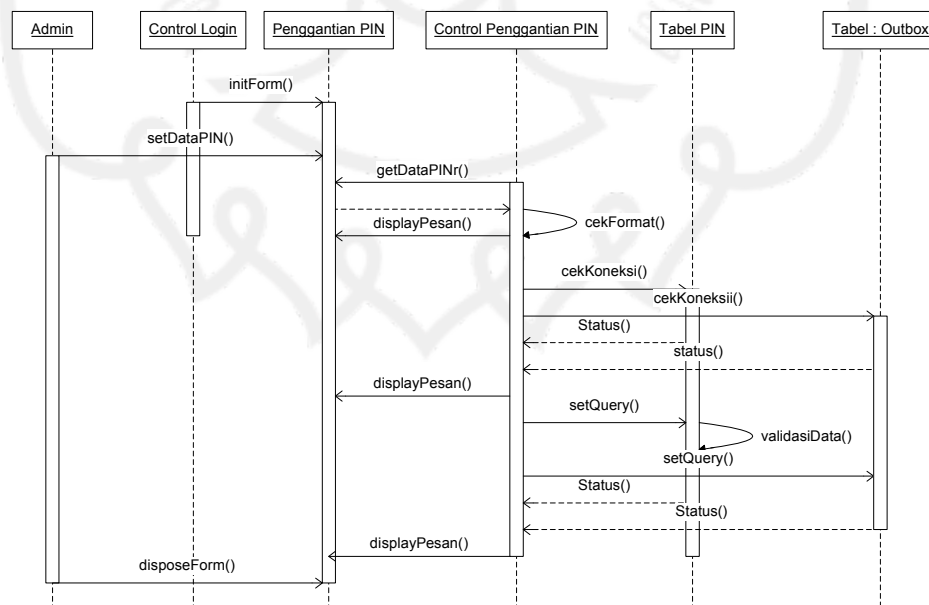
Gambar 3.10 Sequence Diagram Login

2. Sequence Diagram Menambah Deposit



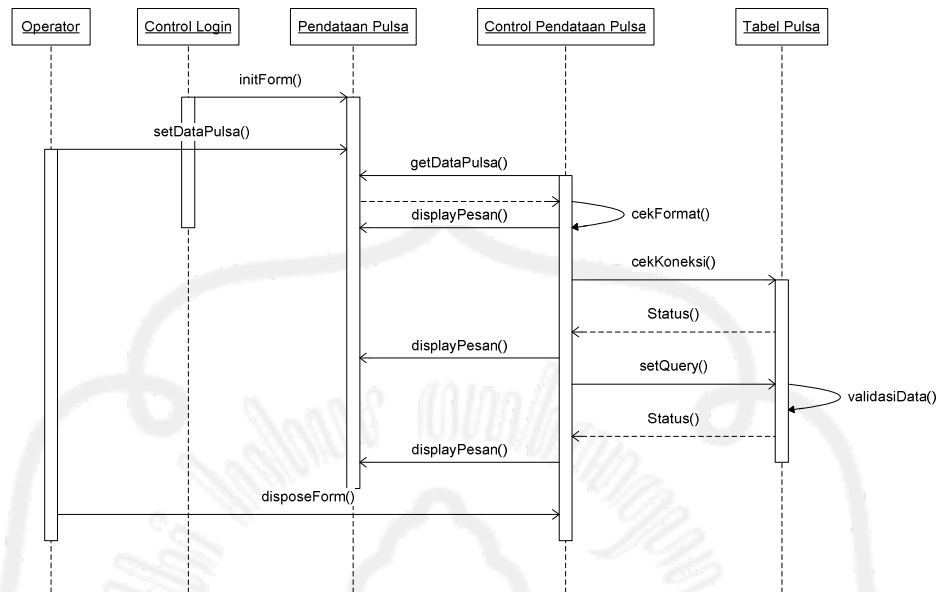
Gambar 3.11 Sequence Menambah Deposit

3. Sequence Diagram Mengganti PIN



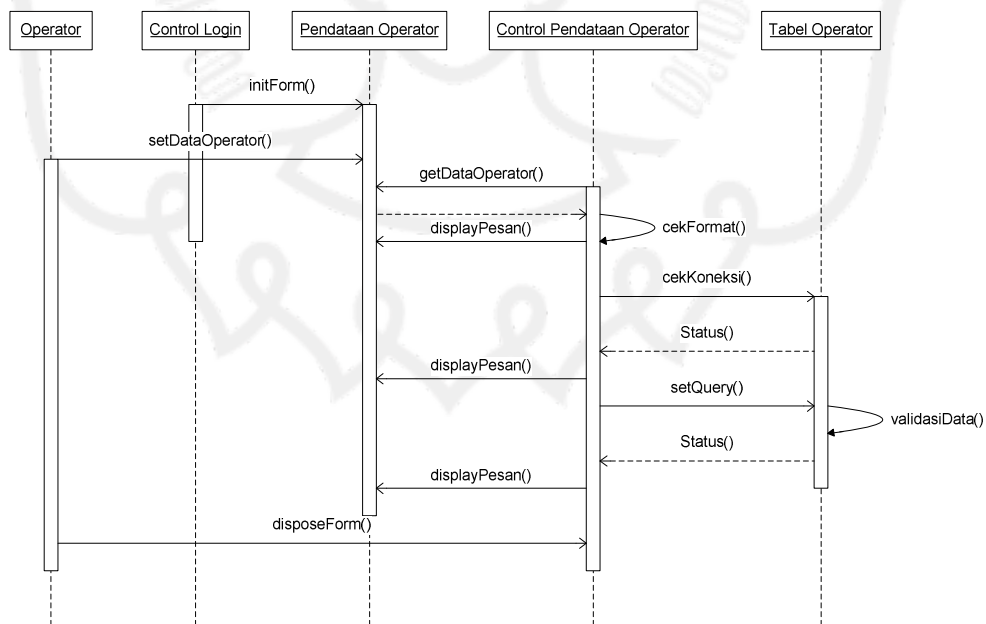
Gambar 3.12 Sequence Diagram Mengganti PIN

4. Sequence Diagram Mendata Pulsa



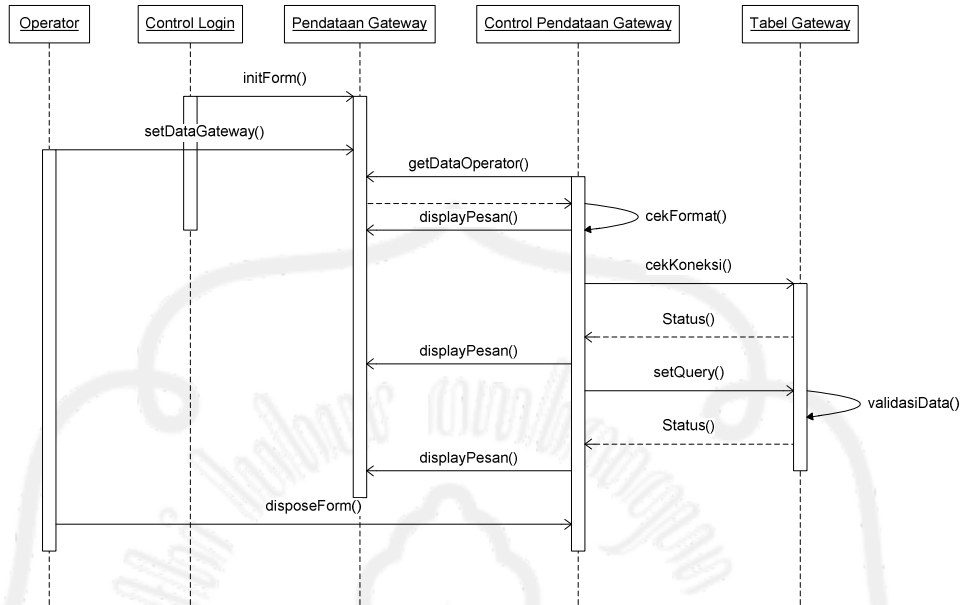
Gambar 3.13 Sequence Diagram Mendata Pulsa

5. Sequence Diagram Mendata Operator



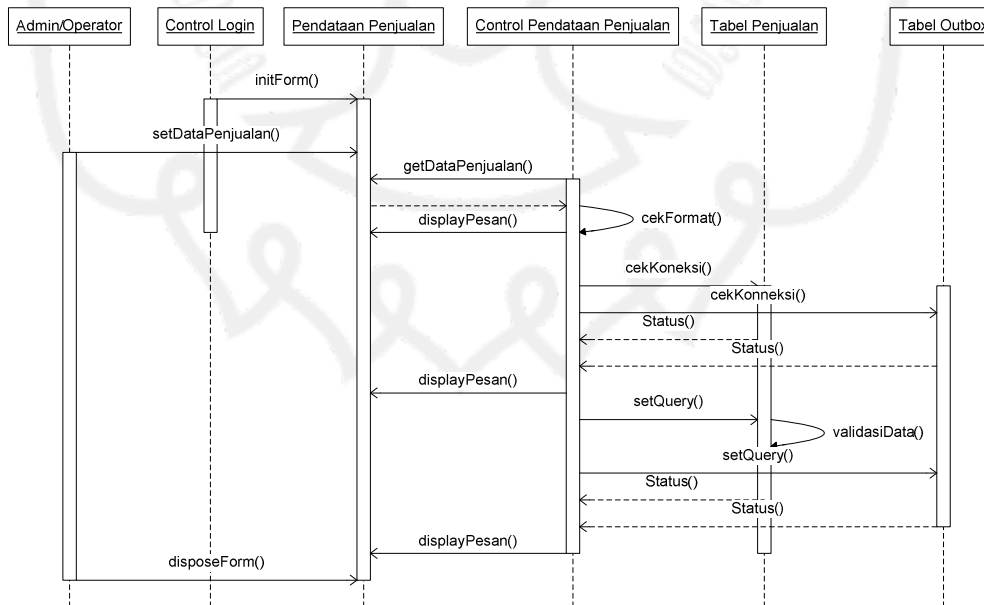
Gambar 3.14 Sequence Diagram Mendata Operator

6. Sequence Diagram Pendataan Gateway



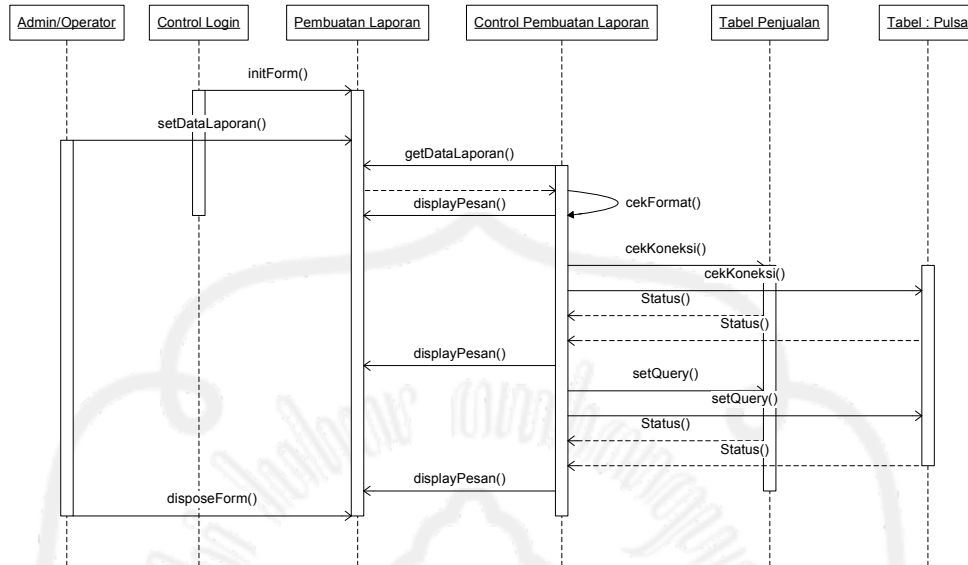
Gambar 3.15 Sequence Diagram Mendata Gateway

7. Sequence Diagram Mendata Transaksi



Gambar 3.16 Sequence Diagram Mendata Transaksi

8. Sequence Diagram Membuat Laporan

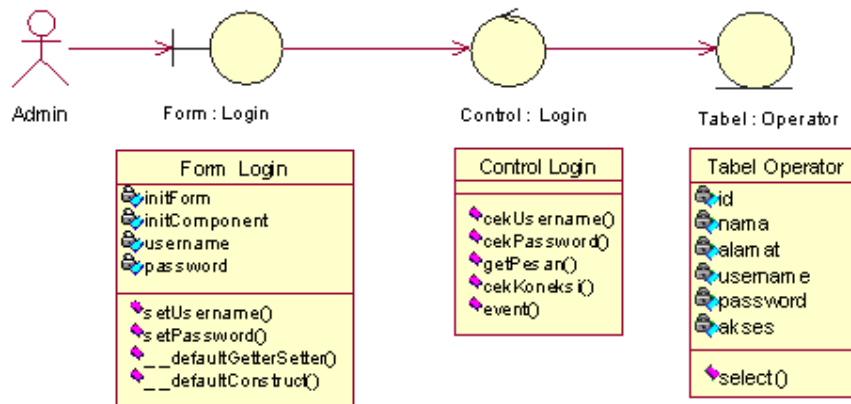


Gambar 3.17 Sequence Diagram Membuat Laporan

3.3.2 Class Diagram

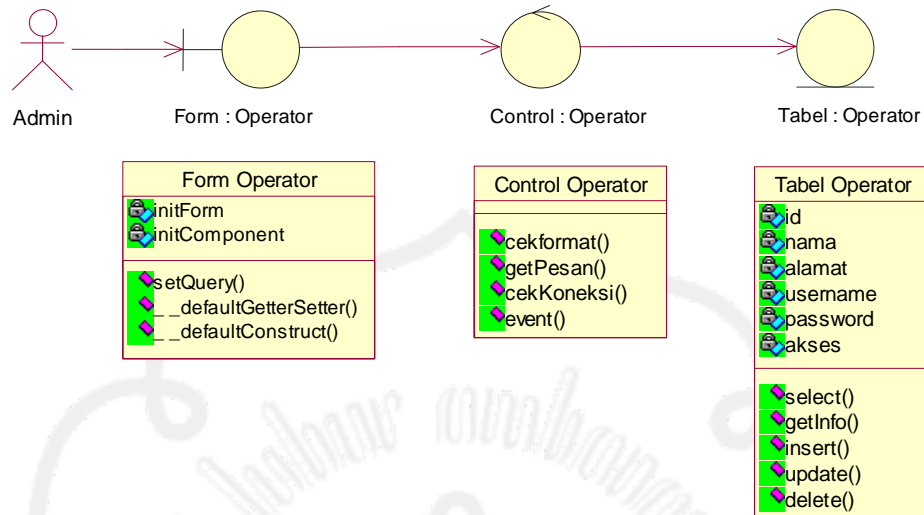
Class Diagram dibuat setelah objek, property, dan method yang ada pada sequence diagram teranalisa, berikut ini class diagram dari aplikasi penjualan ini :

1. Class Diagram Login



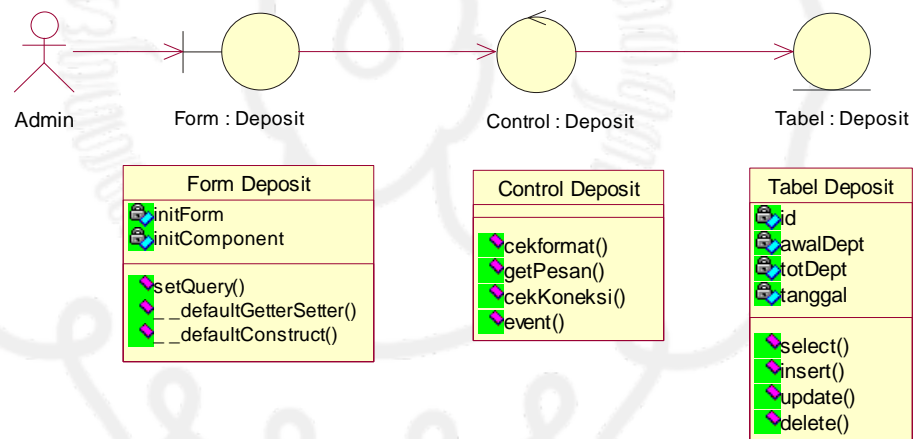
Gambar 3.18 Class Diagram Login

2. Class Diagram Menambah Operator



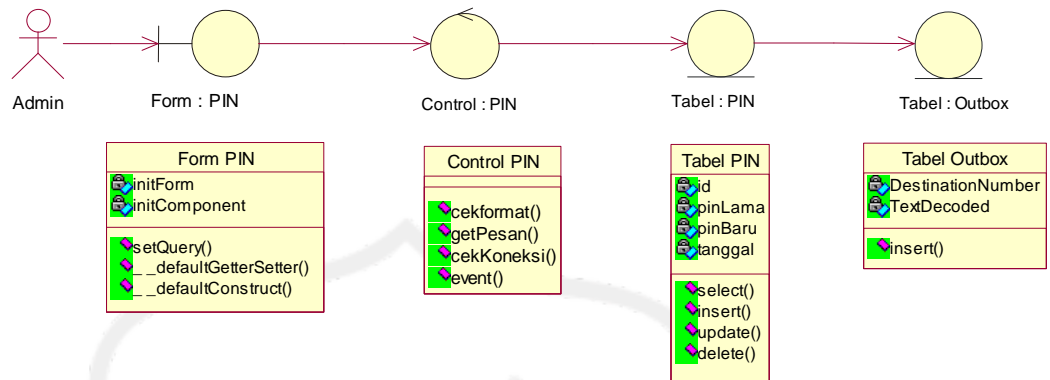
Gambar 3.18 Class Diagram Menambah Operator

3. Class Diagram Mendata Deposit



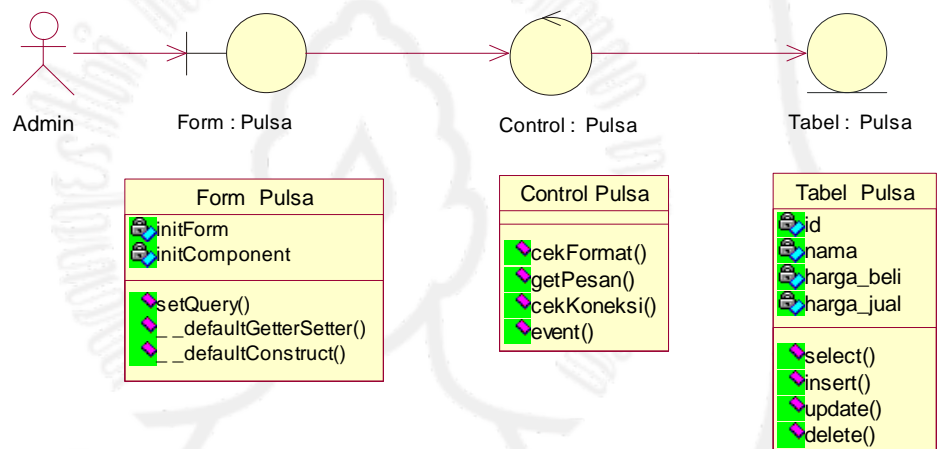
Gambar 3.19 Class Diagram Mendata Deposit

4. Class Diagram Mengganti PIN



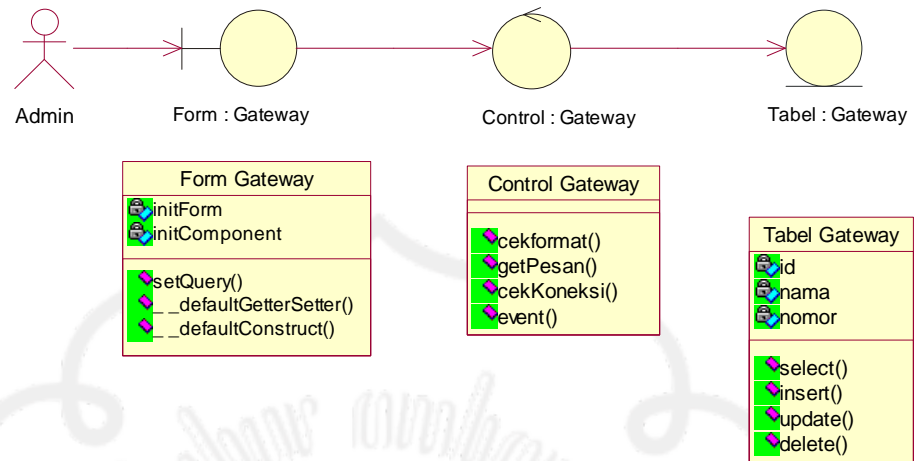
Gambar 3.20 Class Diagram Mengganti PIN

5. Class Diagram Mendata Pulsa



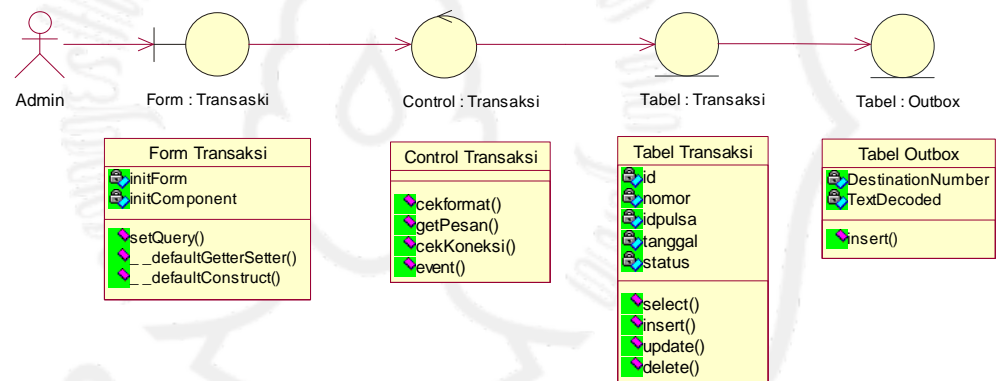
Gambar 3.21 Class Diagram Mendata Pulsa

6. Class Diagram Mendata Gateway



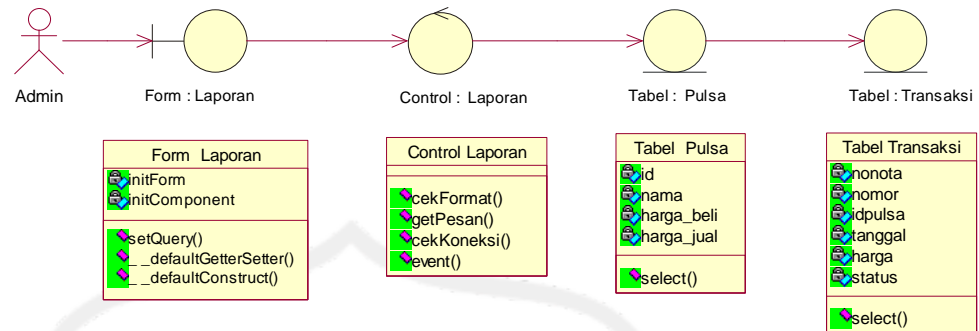
Gambar 3.22 Class Diagram Mendata Gateway

7. Class Diagram Mendata Transaksi



Gambar 3.23 Class Diagram Mendata Transaksi

8. Class Diagram Membuat Laporan



Gambar 3.24 Class Diagram Membuat Laporan

3.4 Kebutuhan *Hardware* dan *Software*

3.4.1 Kebutuhan *Software*

Spesifikasi *software* yang dibutuhkan untuk lingkungan implementasi aplikasi ini adalah:

1. *Netbean* 6.7
2. *MySQL* dan *AppServ Open Project* 2.5.7
3. *Java SQL* 2
4. *Java jdk*1.6.0
5. *Gammu SMS Gateway*.

3.4.2 Kebutuhan *Hardware*

Spesifikasi *Hardware* yang dibutuhkan untuk lingkungan implementasi aplikasi ini adalah:

1. *Prosesor* Pentium IV
2. Memori 128 MB
3. *Harddisk* 40 GB
4. *Port USB*
5. *Modem*
6. *Handphone*

3.5 Perancangan Antar Muka

3.5.1 Rancangan Form Utama

Form Utama merupakan halaman utama dari program ini yang memiliki beberapa menu yang dapat diakses oleh admin maupun operator.

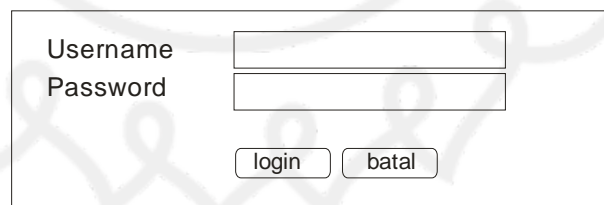


The image shows a wireframe for the main form. It consists of a rectangular box with a thin border. At the top left corner, the text "Menu Bar" is written. The rest of the box is empty, representing the main content area of the application.

Gambar 3.25 Rancangan *Form* Utama.

3.5.2 Rancangan Form Login

Form login ini merupakan *form* yang pertama kali muncul saat operator mengklik file pulsaMO.jar. Adapun gambar rancangan *form login* adalah sebagai berikut



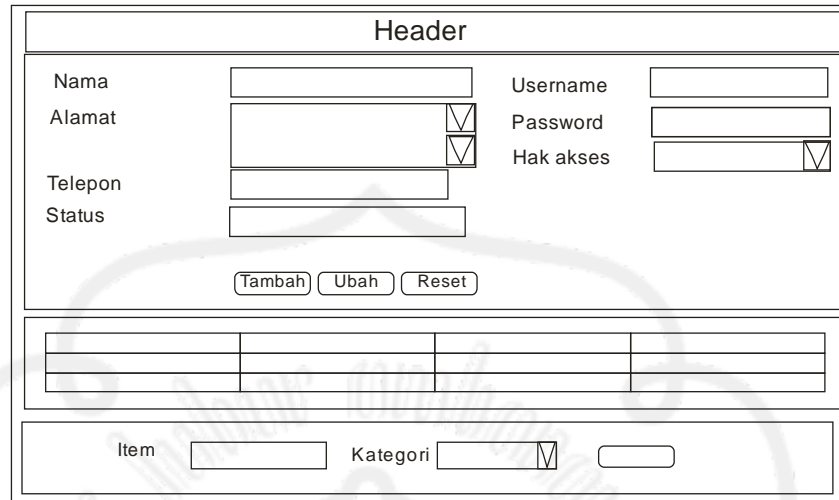
The image shows a wireframe for the login form. It is a rectangular box containing two input fields. The first field is labeled "Username" and the second is labeled "Password". Below the input fields are two buttons: "login" and "batal".

Gambar 3.26 Rancangan *Form Login*.

3.5.3 Rancangan Form Operator

Form pendataan ini digunakan untuk menginputkan data operator. *Form* ini memiliki beberapa *textfield* dan *combobox* yang digunakan untuk mengisi data. Serta beberapa tombol yang akan menjalankan

perintah sesuai *script* aksinya. Adapun gambar rancangan *form* Operator adalah sebagai berikut.

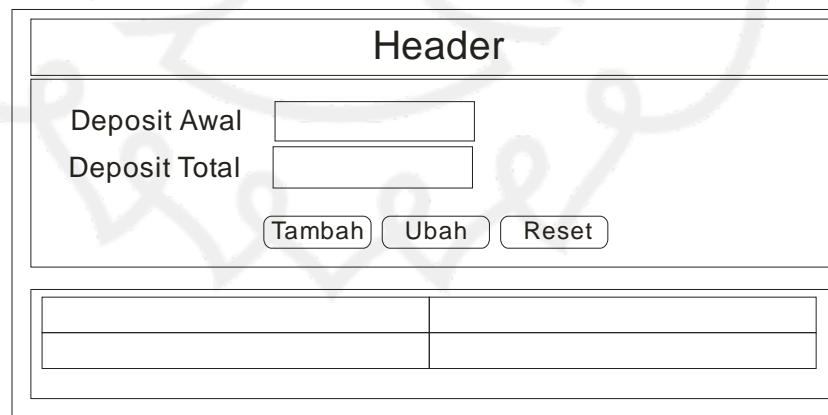


The image shows a web form titled "Header" for an operator. It contains several input fields: "Nama", "Alamat", "Telepon", "Status", "Username", "Password", and "Hak akses". The "Alamat" and "Hak akses" fields have dropdown arrows. Below the input fields are three buttons: "Tambah", "Ubah", and "Reset". Below the buttons is a table with four columns and two rows. At the bottom, there are two more input fields labeled "Item" and "Kategori", with a dropdown arrow next to "Kategori".

Gambar 3.27 Rancangan *Form* Operator.

3.5.4 Rancangan Form Deposit

Form pendataan ini digunakan untuk menambah nilai deposit. *Form* ini memiliki beberapa *textfield* yang digunakan untuk mengisi data. Serta beberapa tombol yang akan menjalankan perintah sesuai *script* aksinya. Adapun gambar rancangan *form* deposit adalah sebagai berikut

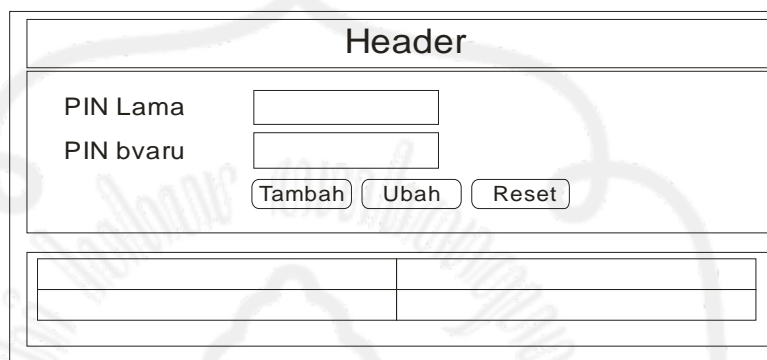


The image shows a web form titled "Header" for a deposit. It contains two input fields: "Deposit Awal" and "Deposit Total". Below the input fields are three buttons: "Tambah", "Ubah", and "Reset". Below the buttons is a table with two columns and two rows.

Gambar 3.28 Rancangan *Form* Deposit

3.5.5 Rancangan Form PIN

Form ini digunakan untuk mangganti nomor PIN yang dimiliki dengan nomor pin yang baru. *Form* ini memiliki beberapa *textfield* yang digunakan untuk mengisi data. Serta beberapa tombol yang akan menjalankan perintah sesuai *script* aksinya. Adapun gambar rancangan *form* Nilai PIN adalah sebagai berikut

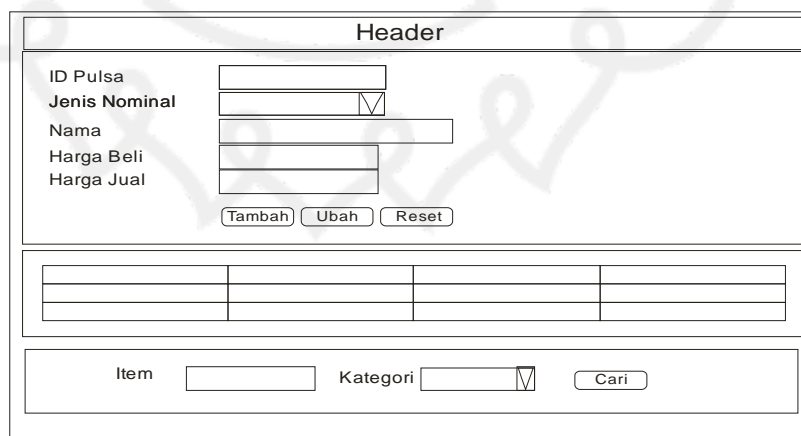


The image shows a form titled "Header" for PIN management. It contains two text input fields labeled "PIN Lama" and "PIN bvaru". Below these fields are three buttons: "Tambah", "Ubah", and "Reset". At the bottom of the form is a table with two columns and two rows.

Gambar 3.29 Rancangan *Form* PIN.

3.5.6 Rancangan Form Pulsa

Form pendataan ini digunakan untuk menginputkan data pulsa. *Form* ini memiliki beberapa *textfield* yang digunakan untuk mengisi data. Serta beberapa tombol yang akan menjalankan perintah sesuai *script* aksinya. Adapun gambar rancangan *form* pulsa adalah sebagai berikut



The image shows a form titled "Header" for pulse management. It contains five text input fields labeled "ID Pulsa", "Jenis Nominal", "Nama", "Harga Beli", and "Harga Jual". Below these fields are three buttons: "Tambah", "Ubah", and "Reset". At the bottom of the form is a table with four columns and two rows. Below the table are two more text input fields labeled "Item" and "Kategori", followed by a "Cari" button.

Gambar 3.30 Rancangan *Form* Pulsa.

3.5.7 Rancangan *Form Gateway*

Form Pendataan ini digunakan untuk menginputkan data *gateway* dari *server* pulsa yang merupakan tujuan pengiriman. *Form* ini memiliki beberapa *textfield* yang digunakan untuk mengisi data. Serta beberapa tombol yang akan menjalankan perintah sesuai *script* aksinya. Adapun gambar rancangan *form gateway* adalah sebagai berikut

The image shows a web form titled "Header". It contains the following elements:

- Two text input fields: "Nama Gateway" and "No. Gateway".
- Three buttons: "Tambah", "Ubah", and "Reset".
- A table with 4 columns and 2 rows.
- Two dropdown menus: "Item" and "Kategori".
- A search button labeled "Cari".

Gambar 3.31 Rancangan *Form Gateway*.

3.5.8 Rancangan *Form Transaksi*

Form pendataan ini digunakan untuk menginputkan data transaksi penjualan pulsa. *Form* ini memiliki beberapa *textfield* yang digunakan untuk mengisi data. Serta beberapa tombol yang akan menjalankan perintah sesuai *script* aksinya Tombol keluar untuk keluar *form*. Adapun gambar rancangan *form* transaksi adalah sebagai berikut

Header															
No. Handphone	<input type="text"/>	Tanggal	<input type="text"/>												
Pulsa	<input type="text"/> ▾														
Harga	<input type="text"/>	Deposit	<input type="text"/>												
Gateway	<input type="text"/> ▾														
<input type="checkbox"/> manual	<input type="button" value="Tambah"/>	<input type="button" value="Ubah"/>	<input type="button" value="Reset"/>												
<table border="1"> <tr><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td></tr> </table>															
Item	<input type="text"/>	Kategori	<input type="text"/> ▾ <input type="button" value="Cari"/>												

Gambar 3.32 Rancangan *Form* Transaksi.

3.5.9 Rancangan Form Laporan Transaksi

Rancangan *form* report digunakan untuk mencetak data transaksi. Data transaksi dapat dicetak berdasarkan harian bulanan, atau tahunan. Adapun gambar rancangan *form* report adalah sebagai berikut

Header			
Rekap Data :	<input type="text"/> ▾	<input type="text"/> ▾	<input type="text"/> ▾
<input type="button" value="Print"/>			

Gambar 3.33 Rancangan *Form* Laporan.

BAB IV

IMPLEMENTASI

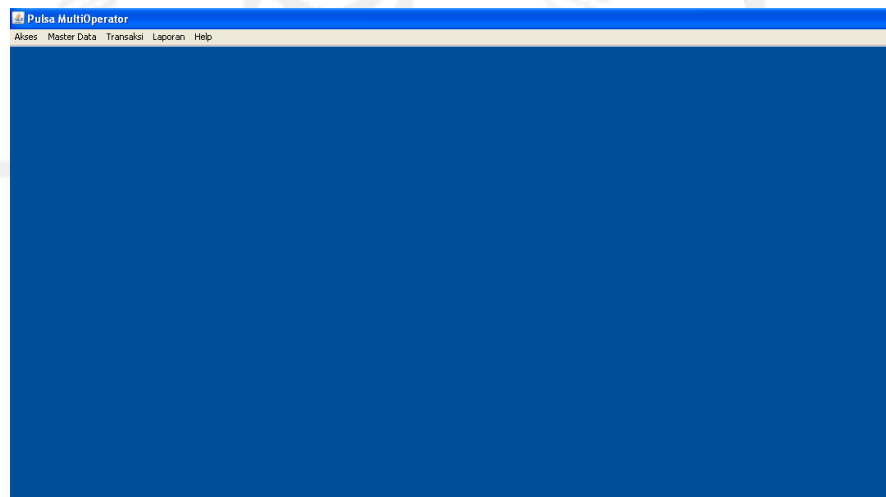
5.1 Implementasi

Tampilan pada Aplikasi Penjualan Pulsa *Multi Operator* terdapat 1 *Form* utama yang dapat memiliki menu item yang dapat menampilkan *form* yang lainnya ketika *form* tersebut dipanggil sesuai dengan menu itemnya. Adapun *form* yang akan muncul ketika dipilih menu itemnya adalah *Form Login*, *Form Deposit*, *Form PIN*, *Form Pulsa*, *Form Operator*, *Form Gateway*, *Form Transaksi*, dan *Form Laporan Penjualan*.

5.1.1 Implementasi Antarmuka

1. *Form* Jendela Utama

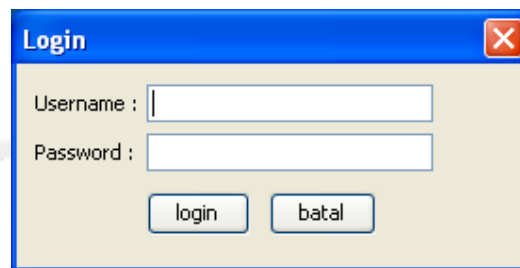
Form Jendela Utama merupakan *form* yang ditampilkan ketika aplikasi ini dijalankan. *Form* Jendela Utama merupakan *form* utama dari aplikasi ini. *Form* ini memiliki sebuah *screen* yang disebut *desktop pane* yang akan menampilkan *form* yang dipilih oleh operator.



Gambar 4.1 *Form* Jendela Utama.

2. Form Login

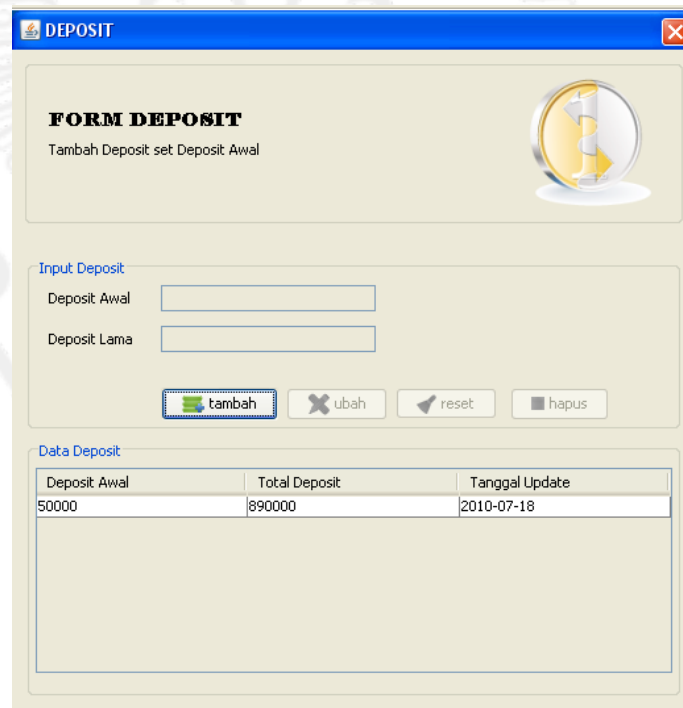
Form login merupakan *form* yang digunakan oleh operator atau admin untuk dapat mengakses menu yang tersedia dengan memasukkan *username* dan *password*..



Gambar 4.2 *Form Login*.

3. Form Deposit

Form Deposit ini ditampilkan apabila kita memilih menu item deposit pada menu berkas di menu bar. *Form* ini hanya digunakan untuk menambah atau mengupdate stok deposit yang dimiliki.



Deposit Awal	Total Deposit	Tanggal Update
50000	890000	2010-07-18

Gambar 4.3 *Form Deposit*.

4. Form PIN

Form PIN ini ditampilkan apabila kita memilih menu item PIN pada menu berkas di menu bar. *Form* ini digunakan untuk mengganti PIN yang dimiliki dengan PIN yang baru.



FORM PIN
Input Penggantian PIN set PIN baru, Pilih Gateway

Input PIN

PIN Baru :

PIN Lama :

Gateway : Pilih

Data PIN

PIN Lama	PIN Baru	Tanggal
123456	0000	2010-07-10

Gambar 4.4 *Form* PIN.

5. Form Operator

Form Operator ini ditampilkan apabila kita memilih menu item operator pada menu berkas di menu bar. *Form* ini digunakan untuk menambah operator yang dapat melakukan *login* pada aplikasi ini.

FORM OPERATOR
Input Data Operator set Nama, Alamat, Telepon, Username, Password, dan Pilih Hak Akses

Input Operator

Nama : Username :
Alamat : Password :
Telepon : Hak Akses : Operator

tambah ubah reset hapus

Daftar Operator

Nama	Alamat	Telepon	Username	Password
Muhammad	Wonogiri	085621464587	ahmad	ahmad
aminudin noor	Gedfingan, Jebres, ...	085625022577	amin	amin
admin	Sumberagung RT2/R...	08562502257	admin	admin

Pencarian

Gambar 4.5 *Form Operator*

6. Form Pulsa

Form pulsa ini ditampilkan apabila kita memilih menu item pulsa pada menu berkas di menu bar. *Form* pulsa ini digunakan untuk mendata pulsa.

FORM PULSA

Input Data Pulsa set Id. Pulsa, Nama, Nominal, Harga Beli, Harga Jual

Input Pulsa

Id. Pulsa

Nama

Harga Beli

Harga Jual

tambah ubah reset hapus

Daftar Pulsa

ID.Pulsa	Nama	Harga Beli	Harga Jual
A10	AS 10000	9950	11000
A100	AS 100000	97750	100000
A15	AS 15000	15700	17000
A20	AS 20000	19950	21000
A25	AS 25000	25200	26000
A5	AS 5000	5100	6000
A50	AS 50000	49300	50000
AX1	AXIS 1000	1175	1500
AX10	AXIS 10000	9950	11000
AX100	AXIS 100000	97200	98000
AX2	AXIS 2000	2150	2500
AX25	AXIS 25000	24700	25500

Pencarian

Item : Kategori : Id Pulsa

Gambar 4.6 *Form* Pulsa.

7. Form Gateway

Form gateway ini ditampilkan apabila kita memilih menu item *gateway* pada menu berkas di menu bar. *Form gateway* ini digunakan untuk mendaftarkan nomer *gateway* yang merupakan nomer dari *server* pulsa yang menjadi tujuan pengiriman.



ID Gateway	Nama	No. Gateway
G-006	Telkomsel Satu	085229337999
G-005	Telkomsel Dua	086229770229
G-004	XL Satu	081904791999
G-003	XL Dua	087736057999
G-002	Indosat Dua	085867244409
G-001	Indosat Satu	085728121999
assss	cukasss	3333

Gambar 4.7 *Form Gateway*.

8. Form Transaksi

Form transaksi ini ditampilkan apabila kita memilih menu item penjualan pada menu transaksi di menu bar. *Form* ini digunakan untuk mendata transaksi penjualan yang dilakukan.

FORM TRANSAKSI
Input Data Transaksi No. HP, Pilih Pulsa, dan Pilih Gateway

Input Transaksi

No. HP: Tanggal: July, 19-2010
Pulsa: Deposit: Rp. 890000,00
Harga Jual:
Gateway: Pilih

tambah ubah reset hapus

Daftar Transaksi

Nomor HP	Id Pulsa	Tanggal	Harga Jual	Status
122***	I55	2010-07-18	Rp. 6000	Belum Terkirim
12***	IG5	2010-07-18	Rp. 6000	Belum Terkirim
1233***	A10	2010-07-18	Rp. 11000	Belum Terkirim
12333333***	I55	2010-07-18	Rp. 6000	Belum Terkirim
23333333***	IG5	2010-07-18	Rp. 6000	Belum Terkirim
12333333***	A10	2010-07-18	Rp. 11000	Belum Terkirim
12222222***	IG5	2010-07-18	Rp. 6000	Belum Terkirim
1233***	A5	2010-07-18	Rp. 6000	Belum Terkirim
***	A10	2010-07-18	Rp. 11000	Belum Terkirim
1***	S5	2010-07-18	Rp. 6000	Belum Terkirim
1***	S5	2010-07-18	Rp. 6000	Belum Terkirim
12***	IG5	2010-07-18	Rp. 6000	Belum Terkirim

Refresh

Pencarian

Item: Kategori: No. HP

Gambar 4.8 *Form* Transaksi.

9. Form Laporan

Form laporan ini ditampilkan apabila kita memilih menu item laporan pada menu Laporan di menu bar. *Form* ini digunakan untuk mencetak data transaksi penjualan. Data transaksi dapat dicetak berdasarkan harian bulanan, atau tahunan.



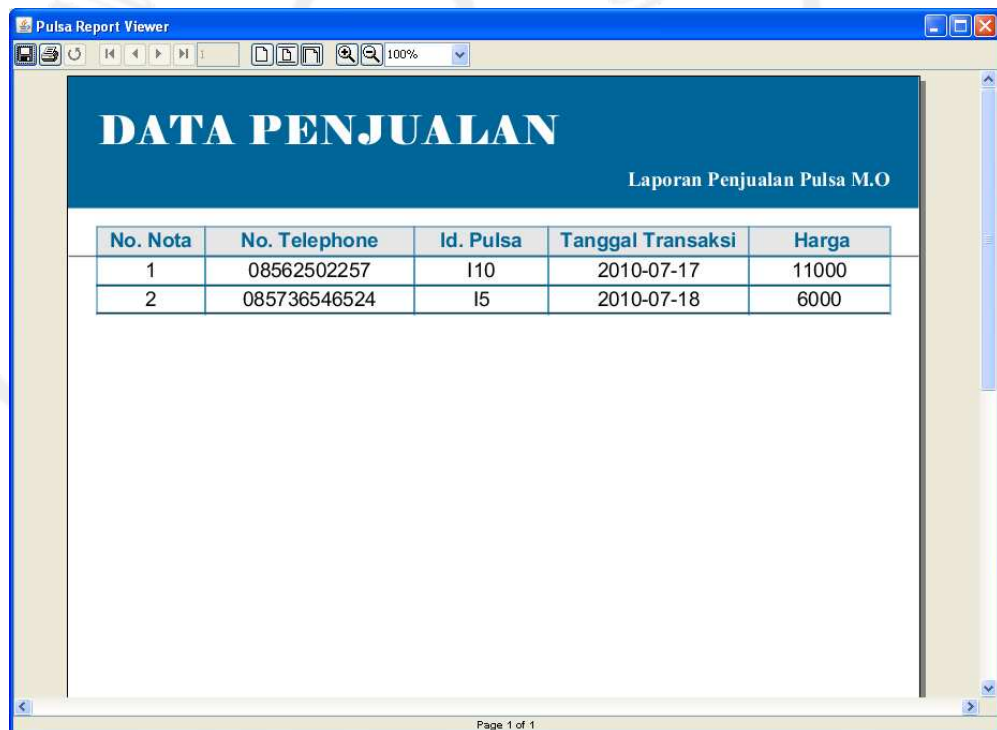
FORM LAPORAN
Cetak Laporan Pilih Data Laporan

Item: [Dropdown Menu]

Tanggal Awal: 2010-07-19

Tanggal Akhir: 2010-07-19

Preview

Gambar 4.9 *Form* Laporan Penjualan.

DATA PENJUALAN
Laporan Penjualan Pulsa M.O

No. Nota	No. Telephone	Id. Pulsa	Tanggal Transaksi	Harga
1	08562502257	I10	2010-07-17	11000
2	085736546524	I5	2010-07-18	6000

Page 1 of 1

Gambar 4.10 Tampilan Laporan Panjualan.

BAB V

PENUTUP

5.1. Kesimpulan

Dari hasil penelitian terhadap masalah dan aplikasi yang dikembangkan maka dapat ditarik beberapa kesimpulan, antara lain:

1. Aplikasi Penjualan Pulsa *Multi Operator* didesain berbasis *Java J2SE* yang bersifat offline yang menangani penjualan pulsa *multi operator* secara terkomputerisasi.
2. Aplikasi Penjualan Pulsa *Multi Operator* mampu mempercepat pendataan dan memudahkan pengolahan data. Sehingga efektifitas kerja dapat tercapai dan arus informasi dapat berjalan dengan cepat dan efisien.

5.2. Saran

Aplikasi ini masih jauh dari sempurna, sehingga untuk memaksimalkan kinerjanya perlu dilakukan pengembangan lebih lanjut, diantaranya adalah :

1. Peng-*update*-an nilai deposit dapat dilakukan secara otomatis setiap kali terjadi transaksi.
2. Perbaiki sistem sehingga aplikasi ini dapat digunakan untuk semua *server* pulsa *Multi Operator*.

DAFTAR PUSTAKA

- Badriyah, Tessy. 2007. *UML Class Diagram*. <http://lecturer.eepis-its.edu/~tessy/tutorial/java/5UML%20Class%20Diagram.pdf>. diakses pada tanggal 16 Juni 2010.
- Doug Rosenberg, Scot Kendall. 2001. *Applying Use Case Driven Object with UML : an Annotated e-Commerce Example* . Upper Sadle River : Adison-Wesley.
- Kabir, M. J. 2002. *Apache Server 2 Bible*. USA : Hungry Minds.
- Laurie, B., dan Laurie, P. 2002. *Apache The Definitive Guide*. California : O'reilly & Associates.
- Utomo, Andy Prasetyo. *Pemrograman Berorientasi Objek (Java)*. <http://apu0226.files.wordpress.com/2009/05/pbo-perkenalan-java.pdf>. diakses tanggal 16 Juni 2010.
- Wahana Komputer. 2010. *Pengembangan Aplikasi Database Berbasis JavaDB dan Netbean*. Jakarta : Penerbit Andi.