Perancangan Aplikasi Pengisian Pulsa dengan Java Mobile

Ummi Fauziyah, Dr. Poltak Sihombing, M.Kom, Handrizal, S.Si, M.Comp.Sc

Program Studi Ekstensi S1 Ilmu Komputer

Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara

Jalan Universitas No. 9 Kampus USU Medan 20155

ummi_tipatu@yahoo.com

poltakhombing@yahoo.com

handrizal_tanjung@yahoo.com

***Abstrak*— Perkembangan pesat ilmu pengetahuan dan teknologi telah memberikan banyak manfaat pada semua kalangan masyarakat dalam berbagai bidang. Salah satunya yaitu teknologi *handphone* yang saat ini dijadikan sebagai kebutuhan pokok dalam komunikasi serta ikut mendominasi hampir seluruh bidang kehidupan, mulai dari bisnis, sosial dan pendidikan. Perkembangan teknologi seluler merambat pada bisnis pengisian pulsa. Namun dalam transaksi pengisian pulsa, reseller sering melakukan kesalahan dalam penulisan format pesan. Banyaknya menu yang dibutuhkan reseller seperti cek saldo, komplain, cek harga dan lain-lain menjadi kendala bagi reseller, karena sulitnya mengingat setiap format yang dimiliki tiap menu. Masalah lain yang timbul yaitu reseller harus mencatat secara manual semua transaksi pengisian pulsa. Dari permasalahan yang ada maka dirancang aplikasi yang mampu memaksimalkan kinerja teknologi *handphone* sebagai media bantu dalam bisnis penjualan pulsa. Aplikasi pengisian pulsa ini merupakan aplikasi yang digunakan reseller dengan memanfaatkan layanan SMS (Short Message Service). Reseller menggunakan layanan SMS untuk melakukan transaksi ke server. Aplikasi dirancang dengan *java mobile* dan berjalan pada ponsel yang memiliki fitur java MIDP 2.0. Aplikasi yang dirancang mampu menghindari kesalahan penulisan urutan format pesan seperti input kode produk dan input nomor tujuan. Selain melakukan pengisian pulsa aplikasi ini juga dapat melakukan cek saldo, komplain, cek harga dan lain sebagainya yang ditampung dalam menu utama. Tampilan dirancang sehingga mudah dimengerti oleh reseller, pesan keluar pengisian pulsa disimpan dalam aplikasi sehingga dapat dijadikan catatan bagi reseller. Implementasi aplikasi ini mampu mempermudah reseller dalam pengisian pulsa, sehingga menciptakan efisiensi waktu dan menghindari *human error* pada saat transaksi.**

***Kata Kunci*— pulsa, *handphone*, *java mobile*, *reseller*, SMS.**

I. PENDAHULUAN

A. Latar Belakang

Perkembangan ilmu pengetahuan dan teknologi mampu mendorong kemampuan manusia untuk berusaha mengatasi segala permasalahan menjadi lebih sederhana sehingga menjadikan permasalahan dapat dilakukan dengan mudah. Namun saat ini kemudahan dalam memperoleh berbagai informasi dapat dinikmati dalam hitungan detik, ini merupakan bentuk dari perkembangan ilmu pengetahuan dan teknologi. Bentuk lainnya seperti teknologi perangkat *mobile* seperti *handphone*. Kemampuan berkembangnya teknologi seluler ini sangat luar biasa, dalam waktu yang begitu singkat, teknologi seluler ini telah digunakan oleh seluruh lapisan masyarakat dan menjadikannya sebagai suatu kebutuhan pokok dalam berkomunikasi. Semakin hari semakin banyak pengguna *handphone* sehingga memacu para pabrikan *handphone* untuk saling bersaing mengembangkan dan meningkatkan kinerja teknologi selulernya.

Seiring dengan meningkatnya kemampuan perangkat seluler, maka kebutuhan akan pulsa juga berkembang pesat. Namun dalam transaksi penjualan pulsa, terdapat beberapa format yang harus diketik *reseller*. Karena format pesan yang begitu rumit sehingga dapat mengakibatkan ketidak efisienan waktu pada saat transaksi. Bukan hanya mengakibatkan ketidak efisienan waktu tetapi dapat pula mengakibatkan terjadinya kesalahan ketika pengetikan format. Perkembangan teknologi *handphone* diharapkan berdampak pada kemajuan teknologi didalam penjualan pulsa, saat ini *reseller* hanya memanfaatkan teknologi SMS secara konvensional untuk melakukan transaksi, padahal mayoritas *reseller* menggunakan *handphone* yang memiliki fasilitas java, sehingga apabila dibangun aplikasi pulsa dengan J2ME dapat memaksimalkan kemampuan yang ada di *handphone reseller* pulsa yaitu mampu membantu dalam pencatatan setiap transaksi pulsa, memudahkan *reseller* untuk melakukan transaksi tanpa harus mengetikkan format yang ada sehingga mampu meminimalisir *human error* dan menciptakan efisiensi waktu saat transaksi berlangsung.

B. Rumusan Masalah

Rumusan masalah yang dikaji dalam penelitian ini:

1. Bagaimana mengurangi kesalahan reseller yang sering terjadi saat melakukan transaksi pengisian pulsa dikarenakan penulisan format pesan yang salah.
2. Bagaimana mengatasi kesulitan reseller dalam menggunakan menu selain pengisian pulsa seperti cek saldo, cek bonus, komplain dan lain sebagainya sehingga memungkinkan reseller untuk tidak mengingat format-format penulisan tiap menu.
3. Bagaimana mengatasi agar reseller tidak lagi mencatat setiap transaksi pengisian pulsa secara manual.

C. Batasan Masalah

Dalam suatu penelitian diperlukan batasan masalah agar tidak menyimpang dari yang diinginkan, maka penulis membatasi pembahasan yang ada, yaitu:

1. Merancang aplikasi pengisian pulsa bagi *reseller*.
2. Aplikasi ini menggunakan sms sebagai koneksinya.
3. Menggunakan bahasa pemograman java, didalam hal ini menggunakan J2ME.
4. Penggunaan aplikasi ditujukan pada *handphone* yang mendukung fasilitas java.
5. Aplikasi ini hanya dapat digunakan bagi reseller yang terdaftar pada server pabrik pulsa, karena masing-masing server memiliki format pengetikan yang berbeda-beda.

II. LANDASAN TEORI

A. Perangkat Mobile

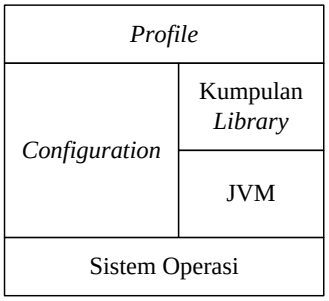
Perangkat *mobile* mempunyai pengertian yaitu perangkat yang bergerak, tidak diam di satu tempat, dan bisa dibawa kemana-mana. Contoh dari perangkat *mobile* ini adalah telepon selular, pagers, PDA (*Personal Digital Assistant*), dan lain-lain [7].

B. Java2 Micro Edition (J2ME)

Java2 Micro Edition atau yang biasa disebut J2ME adalah lingkungan pengembangan yang didesain untuk meletakkan perangkat lunak Java pada barang elektronik beserta perangkat pendukungnya. Pada J2ME, jika perangkat lunak berfungsi baik pada sebuah perangkat maka belum tentu juga berfungsi baik pada perangkat yang lainnya. J2ME membawa ke dunia informasi, komunikasi, dan perangkat komputasi selain perangkat komputer *desktop* yang biasanya lebih kecil dibandingkan perangkat komputer *desktop*. J2ME biasa digunakan pada telepon *selular, pager, personal digital assistants* (PDA's) dan sejenisnya.

J2ME adalah bagian dari J2SE, karena itu tidak semua *library* yang ada pada J2SE dapat digunakan pada J2ME. Tetapi J2ME mempunyai beberapa *library* khusus yang tidak dimiliki J2SE.

Teknologi J2ME juga memiliki beberapa keterbatasan, terutama jika diaplikasikan pada ponsel. J2ME sangat tergantung pada perangkat (*device*) yang digunakan, bisa dari segi merk ponsel, maupun kemampuan ponsel, dan dukungannya terhadap teknologi J2ME.



Gambar 1 Arsitektur J2ME [7]

Seperti yang terlihat pada gambar 1 bahwa konfigurasi merupakan bagian yang berisi JVM (*Java Virtual Machine*) dan beberapa *library* kelas lainnya. Terdapat dua buah konfigurasi yang disediakan yaitu CLDC (*Connected Limited Device Configuration*) yang menangani perangkat yang kecil dan CDC (*Connected Device Configuration*) untuk perangkat yang lebih besar. Berikut adalah perbandingan antara CLDC dan CDC :

TABEL I
PERBANDINGAN CLDC DAN CDC [7]

CLDC	CDC
Mengimplementasikan sebagian dari J2SE	Mengimplementasikan seluruh fitur J2SE
JVM yang digunakan adalah KVM	JVM yang digunakan adalah CVM
Digunakan pada perangkat genggam (*handphone*, PDA, *twoway pager*) dengan memori terbatas (160-512 KB)	Digunakan pada perangkat genggam (internet TV, Nokia *Communicator*, car TV) dengan memori minimal 2MB
Prosessor : 16/32bit	Prosessor : 32bit

1) Connected Limited Device Configuration (CLDC) adalah perangkat dasar J2ME, spesifikasi dasar yang berupa *library* dan API yang diimplementasikan pada J2ME, seperti yang digunakan pada telepon selular, *pager*, dan PDA. Perangkat tersebut dibatasi dengan keterbatasan memori, sumber daya, dan kemampuan memproses. Spesifikasi CLDC pada J2ME adalah spesifikasi minimal dari *package*, kelas, dan sebagian fungsi *Java Virtual Machine* yang dikurangi agar dapat diimplementasikan dengan keterbatasan sumberdaya pada alat-alat tersebut, JVM yang digunakan disebut KVM (*Kilobyte Virtual Machine*). Posisi CLDC pada arsitektur J2ME dapat dilihat pada gambar.

2) Connected Device Configuration (CDC) adalah spesifikasi dari konfigurasi J2ME. CDC merupakan komunitas proses pada Java yang memiliki standardisasi. CDC terdiri dari *virtual machine* dan kumpulan *library* dasar untuk dipergunakan pada *profile* industri. Implementasi CDC pada J2ME adalah *source code* yang menyediakan sambungan dengan macam-macam *platform.*

Profil merupakan bagian perluasan dari konfigurasi. Artinya, selain sekumpulan kelas yang terdapat pada konfigurasi, terdapat juga kelas-kelas spesifik yang didefinisikan lagi di dalam profil. Dengan kata lain, profil akan membantu secara fungsional yaitu dengan menyediakan kelas-kelas yang tidak terdapat di level konfigurasi. Dalam J2ME terdapat dua buah profil yaitu MIDP (*Mobile* Information *Device* Pro*file*) dan *Foundation Profile*. *Mobile Information Device Profile* (MIDP) adalah spesifikasi untuk sebuah profil J2ME. MIDP memiliki lapisan diatas CLDC dan API. *Foundation Profile*, yaitu profil yang digunakan untuk konfigurasi CDC. Profil ini menambahkan beberapa kelas dari J2SE ke dalam konfigurasi CDC, dan berperan juga sebagai pondasi untuk membentuk profil baru lainnya.

KVM adalah paket JVM yang di desain untuk perangkat yang kecil. KVM mendukung sebagian dari fitur-fitur JVM. KVM diimplementasikan dengan menggunakan C sehingga mudah beradaptasi pada tipe *platform* yang berbeda.

CVM adalah paket JVM optimal yang digunakan pada CDC. CVM mempunyai seluruh fitur dari virtual machine yang didesain untuk perangkat yang memerlukan fitur-fitur *Java 2 virtual machine*.

Paket-paket opsional merupakan paket-paket tambahan yang dibutuhkan oleh aplikasi sehingga pada saat proses deployment paket-paket tersebut perlu didistribusikan juga sebagai bagian dari aplikasi bersangkutan. Saat ini terdapat dua jenis aplikasi J2ME yaitu:

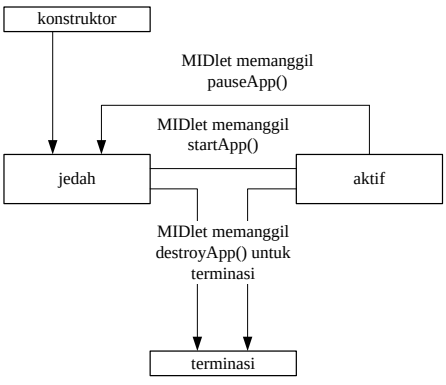
1) Walled Garden Application merupakana aplikasi yang berdiri sendiri atau *standalone* yang berjalan pada *handphone* tanpa perlu mengakses sumber data eksternal melalui jaringan pembawa (*carier network*). Contoh dari aplikasi ini adalah kalkulator atau *single player games.*

2) Network Aware Application merupakan aplikasi yang berinteraksi dengan jaringan. Tidak seperti aplikasi pertama, aplikasi ini memiliki kemampuan untuk mengakses sumber daya eksternal. Contoh dari aplikasi jenis ini adalah aplikasi email yang berada pada *handphone*, aplikasi untuk mendapatkan kembali data alamat-alamat yang tersimpan melalui jaringan dan pengiriman email berbagai alamat melalui jaringan data.[7]

MIDlet adalah aplikasi yang ditulis untuk MIDP. Aplikasi MIDlet adalah bagian dari kelas Javax.microedition .midlet.MIDlet yang didefenisikan pada MIDP. MIDlet berupa sebuah kelas abstrak yang merupakan subkelas dari bentuk dasar aplikasi sehingga antarmuka antara aplikasi J2ME dan aplikasi manajemen pada perangkat dapat terbentuk.

Terdapat tiga buah method yang harus diimplementasikan oleh setiap MIDlet. Dengan kata lain, setiap MIDlet yang dibuat harus memiliki ketiga buah method tersebut. Adapun method-method tersebut adalah startApp (), pauseApp () dan destroyApp ().

Setiap MIDlet dapat berada dalam salah satu keadaan (*state*) berikut: ***Paused, Active*** maupun ***Destroyed***. Berikut ini diilustrasikan ketiga buah keadaan tersebut dan pada saat kapan MIDlet akan berada dalam keadaan tertentu.



Gambar 2 Siklus MIDlet [7]

Seperti yang terlihat pada gambar 2 bahwa pada saat pembuatan MIDlet baru, mula-mula MIDlet akan berada dalam keadaan *Paused.* Apabila proses pembuatan MIDlet gagal atau mengakibatkan kesalahan (menimbulkan eksepsi), maka MIDlet akan langsung berada dalam keadaan *Destroyed*. Namun apabila proses pembuatan MIDlet berjalan dengan baik, maka setelah MIDlet dijalankan secara otomatis akan mengeksekusi method startApp() dan hal ini akan mengubah MIDlet untuk berada dalam keadaan *Active*. MIDlet yang berada dalam keadaan *Active* dapat diubah kembali menjadi keadaan *Paused* melalui pemanggilan method pauseApp() atau diubah menjadi keadaan *Destroyed* melalui pemanggilan method destroyApp(). Sebagai contoh, pada saat MIDlet dijalankan dan kemudian dihentikan oleh *user*, maka MIDlet akan mengalami perubahan keadaan, yaitu dari *Active* menjadi *Destroyed*.

Method startApp() akan dipanggil untuk memerintahkan MIDlet agar memperoleh fokus dan menjadikan MIDlet berada dalam keadaan *Active*. Untuk lebih memudahkan pemahaman, dapat dikatakan pula bahwa method startApp() itu digunakan untuk mengaktifkan MIDlet. Hal ini dapat terjadi ketika MIDlet baru saja dibuat atau MIDlet yang akan kembali diaktifkan dari keadaan *Paused.* Method pauseApp()dipanggil untuk memerintahkan MIDlet agar tidak memiliki fokus dan menjadikan MIDlet berada dalam keadaan *Paused*. Dalam keadaan ini, aplikasi tidak memiliki satu pun tampilan UI (*User* Interface). Apabila aplikasi yang dibuat mengandung objek Thread maupun Timer, maka objek-objek tersebut tidak akan dihentikan secara otomatis. Artinya harus dihentikan secara manual melalui penulisan kode. Aplikasi akan kembali berada dalam keadaan *Active* bila diaktivasi ulang. Method destroyApp() dipanggil untuk memerintahkan MIDlet agar membuang atau membebaskan semua *resource* (biasanya berupa *file*) yang digunakan sekaligus menutup atau menghentikan aplikasi sesegera mungkin. Ini berarti bahwa harus ditutup semua stream yang masih terbuka serta menghentikan semua thread dan timer yang digunakan. Pemanggilan method destroyApp() akan mengakibatkan MIDlet berada dalam keadaan *Destroyed* sehingga pada saat tersebut MIDlet sudah tidak dapat lagi melakukan pengaksesan terhadap objek Display.

JAD (Java Application Descriptor) digunakan untuk mendeskripsikan isi aplikasi untuk keperluan pemetaan. *File* JAD berisi deskripsi *file* JAR (Java Archieve) dan pemetaan atribut MIDlet, sedangkan *file* JAR berisi kumpulan kelas dan *resource*.

C. RMS (Record Management System)

MIDlet tidak menggunakan *file* sistem untuk menyimpan data, tetapi menyimpan semua informasi dalam sebuah memori non-volatile (memori tetap) yang disebut dengan Record Management System. RMS adalah kumpulan record, dan record disimpan sebagai array dari byte dalam sebuah record store. RMS memiliki orientasi record basis data yang sederhana sehingga MIDlet dapat menyimpan informasi dan mengaksesnya. MIDlet yang berbeda dapat mengakses RMS yang sama.[7]

D. SMS (Short Messaging Service)

SMS adalah sebuah layanan yang dilaksanakan dengan ponsel untuk mengirim maupun menerima pesan-pesan. Teks tersebut bisa terdiri dari kata-kata atau nomor atau kombinasi alphanumeric. SMS diciptakan sebagai standar pesan (*message*) oleh ETSI (*Europesan Telecommunication Standards Institute*), yang juga membuat standar GSM yang diimplementasikan oleh semua operator GSM. SMS yang pertama dikirimkan pada Desember 1992 dari PC ke sebuah ponsel melalui jaringan GSM Vodafone di UK. Setiap Pesan maksimal terdiri dari 160 karakter jika menggunakan alphabet Latin, dan 70 karakter jika menggunakan alphabet non-Latin seperti huruf Arab atau China.[2]

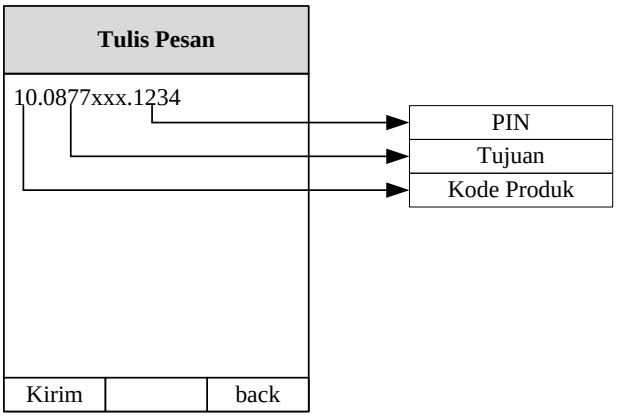
III. ANALISIS DAN PERANCANGAN APLIKASI

A. Analisis Sistem Sedang Berjalan

Analisis sangat diperlukan dalam membangun sebuah perancangan sistem. Dengan analisis akan memudahkan dalam melakukan perancangan. Sebelum melakukan perancangan aplikasi, penulis menganalisis sistem yang digunakan reseller saat ini agar sistem baru yang menggunakan aplikasi ini dapat menggantikan kelemahan yang ada pada sistem yang lama yaitu sistem yang berjalan dengan SMS secara manual.

Pada transaksi pengisian pulsa yang sedang berjalan saat ini, pengisian pulsa yang dilakukan *reseller* melalui media SMS (*Short Message Service*) masih bersifat manual yaitu melalui menu tulis pesan yang terdapat pada aplikasi *handphone* untuk dapat memproses permintaan pengisian dari *reseller*. Server pengisian pulsa telah menentukan format-format penulisan pesan, jadi reseller harus mengetikkan format penulisan yang telah ditentukan oleh server agar transaksi dapat di proses.

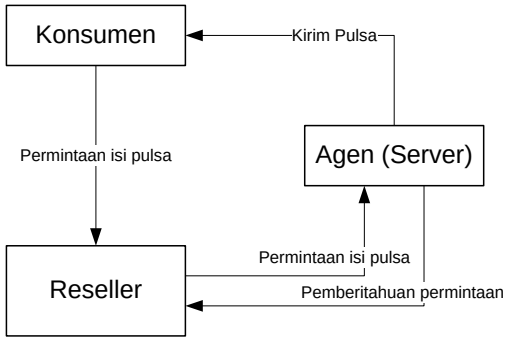
Berikut ini adalah gambar pengisian pulsa yang dilakukan *reseller* melalui media SMS (*Short Message Service*) yaitu menu "tulis pesan atau *create message*".



Gambar 3 Transaksi pengisian pulsa melalui SMS secara manual

Pada gambar 3.1 merupakan bentuk format penulisan pesan isi pulsa melalui sms secara manual dari menu pesan pada handphone. Urutan penulisan pesan yang terdapat pada gambar diatas tidak dapat diubah dengan susunan menginputkan Kode produk, tujuan dan PIN yang dipisahkan dengan titik.

Transaksi pengisian pulsa dengan sistem lama menggunakan SMS secara manual. Berikut ini gambar proses transaksi pulsa secara keseluruhan.



Gambar 4 Proses Transaksi Pulsa

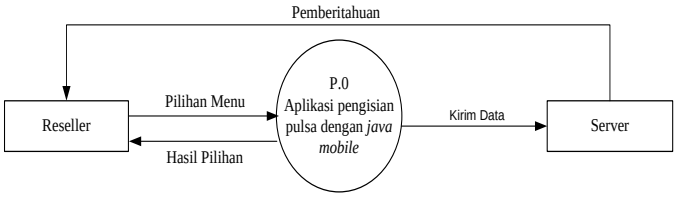
Reseller menerima permintaan pengisian pulsa dari konsumen, kemudian reseller melanjutkan permintaan konsumen ke server. Transaksi pengisian yang dilakukan *reseller* melalui SMS, akan dikirim ke *server* pengisian pulsa (agen). Selanjutnya s*erver* pengisian pulsa akan memproses permintaan dari *reseller*. Setelah memproses permintaan, reseller segera mengirimkan pemberitahuan ke *reseller* dan mengirimkan pulsa ke konsumen.

B. Perancangan Aplikasi Sebagai Sistem Baru

Dalam merancang aplikasi pengisian pulsa ini ada beberapa tahapan yang harus diperhatikan, sehingga aplikasi yang akan dirancang sesuai dengan tujuan yang ingin dicapai. Tahapan-tahapan yang diperlukan dalam merancang aplikasi ini antara lain adalah *Data Flow Diagram* (DFD), *Flowchart* dan rancangan aplikasi.

Data Flow Diagram (DFD) adalah suatu diagram yang menggunakan notasi-notasi untuk menggambarkan arus dari data sistem, yang penggunaannya sangat membantu untuk memahami sistem secara logika, tersruktur dan jelas. DFD merupakan alat bantu dalam menggambarkan atau menjelaskan sistem yang sedang berjalan logis.

DFD terdiri dari diagram konteks dan diagram rinci. Diagram konteks merupakan diagram yang menggambarkan hubungan antar sistem dengan entitas diluar sistem, merupakan sistem secara keseluruhan. Pada aplikasi pengisian pulsa, diagram konteksnya dapat dilihat pada gambar 5 berikut ini :



Gambar 5 Diagram konteks aplikasi pengisian pulsa

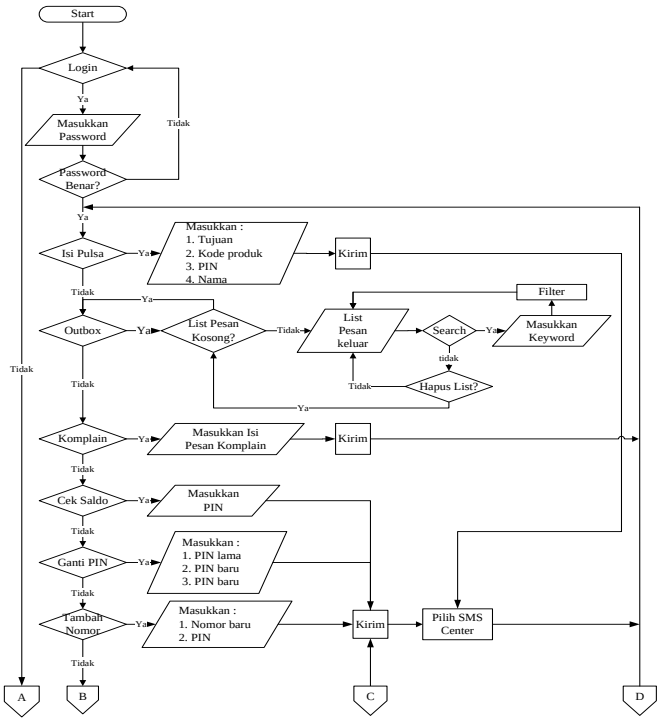
Diagram Konteks di atas menggambarkan sistem secara garis besar yang memperlihatkan masukan, proses, dan keluaran dari sistem yang akan dirancang. Proses yang terjadi pada diagram konteks di atas dapat dijelaskan dengan menggunakan spesifikasi proses pada tabel 2.

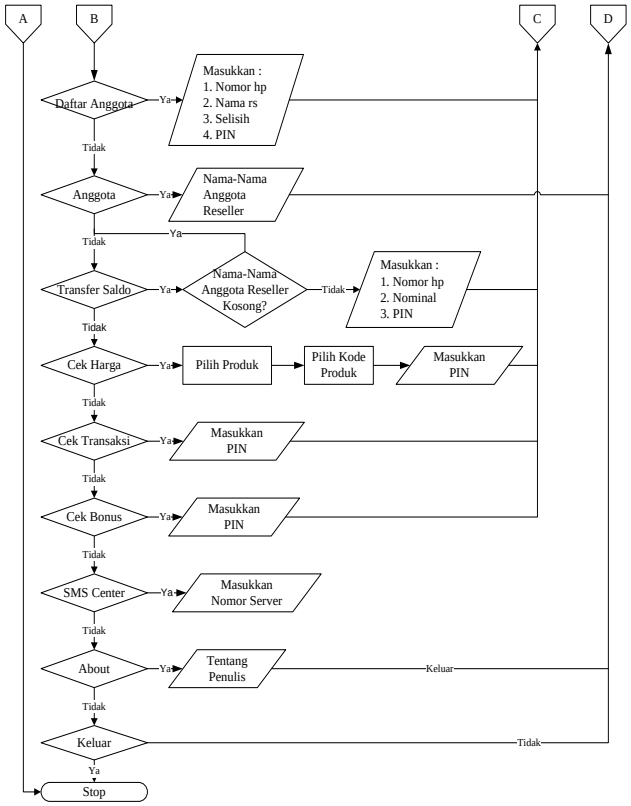
TABEL 2
SPESIFIKASI PROSES DIAGRAM KONTEKS

No Proses	Nama Proses	Input	Keterangan Proses	Output
P.0	Aplikasi pengisian pulsa dengan *java mobile*	Pilihan Menu	Gambaran proses secara keseluruhan	Hasil Pilihan

FlowChart (Bagan Alir Program) adalah suatu bagan yang menggambarkan arus logika dari data yang akan diproses dalam suatu program dari awal sampai akhir.

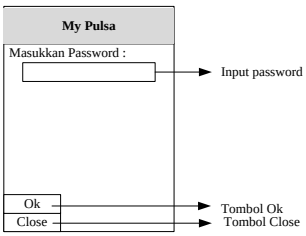
Flowchart dari aplikasi pengisian pulsa dengan menggunakan J2ME dapat dilihat pada gambar 6.





Gambar 6 Flowchart aplikasi pengisian pulsa

Aplikasi pengisian pulsa dirancang sedemikian rupa dengan sederhana sehingga diharapkan dapat memberikan kemudahan bagi *reseller* dalam menggunakan aplikasi ini. Tampilan utama dari aplikasi pengisian pulsa yaitu *form login.* Berikut ini gambar 7 *form login*:



Gambar 7 Rancangan *Form login*

Berdasarkan gambar diatas *reseller* diminta untuk *login* terlebih dahulu sebelum memasuki menu selanjutnya. *Password* dapat diinputkan pada tempat yang telah ditentukan sesuai dengan gambar 7. Tombol *Ok* digunakan jika *reseller* ingin melanjutkan aplikasi ke menu utama seperti yang terlihat pada gambar 7. Sebelum menuju ke menu utama, akan dilakukan proses pemeriksaan pada *password* yang diinputkan dengan *password* yang telah tersimpan pada aplikasi. Jika *password* yang diinputkan sesuai dengan *password* yang tersimpan pada aplikasi maka aplikasi akan melanjutkan ke menu utama. Namun jika *password* salah maka *handphone*

reseller tetap akan menampilkan *form login*. Sedangkan tombol *Close* berfungsi untuk keluar dan menutup aplikasi.

IV. IMPLEMENTASI

A. *Implementasi*

Pada Bab ini, perancangan aplikasi pengisian pulsa dengan menggunakan J2ME yang telah dibuat diimplementasikan dengan menggunakan perangkat lunak *Netbeans IDE 7.1.1* yang menggunakan Java2 Micro Edition sebagai bahasa pemrogramannya dan Nokia SDK 1.0 for java sebagai simulator.

B. *Spesifikasi Perangkat Keras dan Perangkat Lunak yang digunakan*

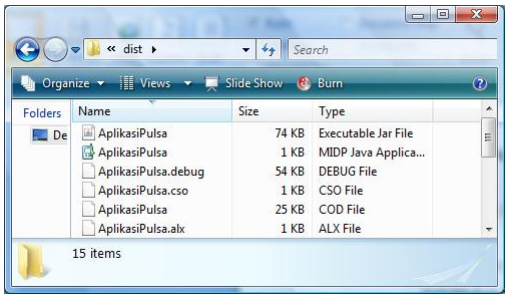
Untuk melakukan pengujian pada aplikasi pengisian pulsa menggunakan J2ME, penulis menggunakan laptop dengan *hardware* dan *software* dengan spesifikasi sebagai berikut:

a. Prosessor Intel(R) Core(TM)2 Duo CPU T5800 @2.00GHz 2.00GHz
b. Sistem Operasi Windows Vista
c. *Memory*(RAM) 4.00GB
d. Perangkat Lunak Netbeans IDE 7.1.1
e. Simulator Nokia SDK 1.0 for java
f. *Handphone* yang mendukung Java MIDP 2.0 dan konfigurasi CLDC 1.1. Program yang berjalan di *emulator* belum tentu dapat berjalan dengan baik di *handphone* yang sesungguhnya.

C. *Distribusi Aplikasi ke Perangkat Mobile*

Distribusi aplikasi ke perangkat *mobile* yaitu dilakukan dengan cara sebagai berikut:

1) Buka folder project aplikasi dengan alamat AplikasiPulsa\dist, maka Anda akan menemukan file berekstensi .jar seperti gambar 8 berikut ini.



Gambar 8 Isi Folder Project AplikasiPulsa

2) Kirimkan file AplikasiPulsa.jar tersebut ke perangkat *mobile* yang diinginkan.
3) Kemudian lakukan instalasi pada perangkat *mobile*.
4) Selesai

D. *Persiapan Teknis*

Untuk menguji aplikasi ini dibutuhkan beberapa komponen yang mendukung agar berjalan dengan baik. Adapun beberapa komponen itu adalah sebagai berikut:

a. Ponsel (*handphone*) yang telah mendukung fasilitas Java dengan *platform* CLDC-1.1 dan MIDP-2.0
b. *Memory* eksternal ponsel
c. Kabel data
d. *Bluetooth*

E. *Pengujian*

Sebelum memasuki menu utama dari aplikasi pengisian pulsa dengan java *mobile*, *reseller* pertama kali akan disajikan dengan pembukaan yaitu *form login*. Untuk masuk ke menu utama, *reseller* diharuskan mengisi *form login* dengan memasukkan *password* pada tempat yang telah disediakan seperti pada gambar 10 berikut ini.

Gambar 10 *Login* Aplikasi

V. KESIMPULAN DAN SARAN

A. *Kesimpulan*

Berdasarkan pembahasan dari bab-bab sebelumnya dapat ditarik beberapa kesimpulan yang terkait dengan perancangan aplikasi pengisian pulsa dengan *java mobile* adalah:

1) Aplikasi ini pengisian pulsa dapat dilakukan dengan mudah oleh reseller tanpa harus mengingat format penulisan pesan, reseller hanya menginputkan kode produk, nomor tujuan dan PIN pada tempat yang yang telah disediakan sehingga aplikasi ini mampu mengurangi kesalahan penulisan format pesan yang sering dilakukan reseller dalam transaksi pengisian pulsa.

2) Selain menu pengisian pulsa masih banyak lagi menu yang dibutuhkan reseller seperti cek saldo, cek bonus, dan lain sebagainya. Pada apliksai ini berbagai menu yang dibutuhkan reseller telah dirangkum dalam menu utama sehingga reseller

dapat dengan mudah menggunakan setiap menu tanpa harus mengingat format penulisan dari tiap menu.

3) Aplikasi ini menyimpan pesan keluar dari pengisian pulsa ke outbox yang terdapat pada menu utama. Pesan keluar ini dapat dimanfaatkan oleh reseller sebagai catatan sehingga reseller tidak perlu mencatat setiap transaksi secara manual.

B. Saran

Berikut adalah saran untuk pengembangan lebih lanjut terhadap aplikasi pengisian pulsa dengan *java mobile*:

Setelah aplikasi pengisian pulsa di coba pada beberapa handphone yang memiliki fitur java. Pada handphone Nokia dengan tipe Nokia X3-02 dapat berjalan dengan baik. Pada handphone Nokia yang menggunakan sistem operasi symbian seperti Nokia 5800 aplikasi tidak dapat instal. Pada saat melakukan penginstalan aplikasi di handphone terdapat pesan kesalahan. Sedangkan pada handphone Blackberry dengan tipe 9220 dan 9320 aplikasi dapat berjalan pada handphone namun aplikasi tidak bisa membaca phonebook pada handphone. Untuk handphone yang menggunakan sistem operasi android seperti Sonyericsson x8 dan Samsung Galaxi w, aplikasi tidak bisa diinstal karena pada android file yang dapat diinstal harus berekstensi .apk sedangkan file aplikasi pengisian pulsa berekstensi .jar. Untuk menginstal file berekstensi .jar pada android dibutuhkan aplikasi pendukung lainnya. Berdasarkan hasil uji coba di atas, diharapkan pengembangan aplikasi dapat kompatibel dengan semua perangkat mobile.

DAFTAR PUSTAKA

[1] Hendri. 2008. *Belajar Otodidak Java dengan Netbeans 6.0*. Jakarta: Elex Media Komputindo.

[2] Imron, R. 2009. *Membuat Sendiri SMS Gateway Berbasis Protokol SMPP.* Yogyakarta: Andi

[3] Irawan. 2008. *Java Mobile untuk Orang Awam*. Palembang: Maxicom.

[4] Parwanto, Heru. 2011. *Perancangan Aplikasi Isi Ulang Pulsa Elektrik dengan Java 2 Micro Edition (J2ME)*. Narotama Collection : Jurnal.

[5] Raharjo dkk. 2007. *Mudah Belajar Java*. Bandung: Informatika.

[6] Riyanto dkk. 2008. *Pengembangan Aplikasi Manajemen Database dengan Java 2 (SE/ME/EE)*. Yogyakarta: Gava Media.

[7] Shalahuddin, M dan Rossa, A.S. 2010. *Pemograman J2ME Belajar Cepat Perangkat Telekomunikasi Mobile Revisi Kedua*. Bandung: Informatika.

[8] Supardi, Yuniar. 2008. *Pemograman Handphone dengan J2ME*. Jakarta: Elex Media Komputindo.

[9] Supardi, Yuniar. 2009. *Belajar Semua Edisi Java2 untuk Segala Tingkat*. Jakarta: Elex Media Komputindo.

[10] http://netbeans.org/kb/trails/mobility.html. Diakses tanggal 18 januari, 2012.

[11] http : / / www. oracle.com/ technetwork/ java / javame/javamobile/ overview/ about/ind ex.html. Diakses tanggal 18 januari, 2012.