

BAB 2

LANDASAN TEORI

2.1 Teori Umum

2.1.1 GPRS

A. Definisi GPRS

GPRS (General Packet Radio Service) adalah suatu teknologi yang memungkinkan pengiriman dan penerimaan data lebih cepat dibandingkan dengan penggunaan teknologi *Circuit Switch Data* atau CSD. Penggabungan layanan telepon seluler dengan *GPRS (General Packet Radio Service)* menghasilkan generasi baru yang disebut 2.5G. Sistem *GPRS* dapat digunakan untuk *transfer* data (dalam bentuk paket data) yang berkaitan dengan *e-mail*, *data gambar (MMS)*, *Wireless Application Protocol (WAP)*, dan *World Wide Web (WWW)*.

GPRS merupakan sistem transmisi berbasis paket untuk GSM yang menggunakan prinsip '*tunnelling*'. Ia menawarkan laju data yang lebih tinggi. Laju datanya secara kasar sampai 160 kbps dibandingkan dengan 9,6 kbps yang dapat disediakan oleh rangkaian tersakelar GSM. Kanal-kanal radio ganda dapat *dialokasikan* bagi seorang pengguna dan kanal yang sama dapat pula digunakan dengan berbagi antar pengguna sehingga menjadi sangat efisien. Dari segi biaya, harga mengacu pada volume

penggunaan. Penggunaanya ditarik biaya dalam kaitannya dengan banyaknya *byte* yang dikirim atau diterima, tanpa memperdulikan panggilan, dengan demikian dimungkinkan *GPRS* akan menjadi lebih cenderung dipilih oleh pelanggan untuk mengaksesnya daripada layanan-layanan IP.

GPRS merupakan teknologi baru yang memungkinkan para operator jaringan komunikasi bergerak menawarkan layanan data dengan laju *bit* yang lebih tinggi dengan tarif rendah, sehingga membuat layanan data menjadi menarik bagi pasar massal. Para operator jaringan komunikasi bergerak di luar negeri kini melihat *GPRS* sebagai kunci untuk mengembangkan pasar komunikasi bergerak menjadi pesaing baru di lahan yang pernah menjadi milik jaringan kabel, yakni layanan internet. Kondisi ini dimungkinkan karena ledakan penggunaan internet melalui jaringan kabel (telepon) dapat pula dilakukan melalui jaringan bergerak. Layanan bergerak yang kini sukses di pasar adalah, laporan cuaca, pemesanan makanan, berita olah raga sampai ke berita-berita penting harian. Dari perkembangan tersebut, dapat dirasakan dampaknya pada kemunculan berbagai provider HP yang bersaing menawarkan tarif *GPRS* yang semakin terjangkau.

Dalam teorinya *GPRS* menjanjikan kecepatan mulai dari 56 kbps sampai 115 kbps, sehingga memungkinkan akses internet, pengiriman data *multimedia* ke komputer, (*notebook*) dan (*handheld computer*).

Namun, dalam implementasinya, hal tersebut sangat tergantung faktor-faktor sebagai berikut:

- Konfigurasi dan alokasi time slot pada *level* BTS.
- Software* yang dipergunakan.
- Dukungan fitur dan aplikasi ponsel yang di gunakan.

(Sumber : <http://www.ceptelefoncunuz.net/pengertian-wap-dan-gprs/>)

B. Perbedaan GPRS dan WAP

WAP merupakan kependekan dari *Wireless Application Protocol* adalah teknologi seperti WWW dan merupakan protokol untuk mengakses internet melalui HP, sedangkan *GPRS (General Packet Radio Service)*. adalah teknologi koneksi yang digunakan oleh HP tersebut menuju jalur internet. Misalnya, digunakannya *broadband* pada PC yang terkoneksi ke *Speedy*.

C. Komponen Utama

Komponen-komponen utama jaringan *GPRS* adalah:

- GGSN (*Gateway GPRS Support Node*): gerbang penghubung jaringan *GPRS* ke jaringan internet. Fungsi dari komponen ini adalah sebagai *interface* ke PDN (*Public Data Network*), *information routing*, *network screening*, *user screening*, *address mapping*.

- SGSN (*Serving GPRS Support Node*): gerbang penghubung jaringan BSS/BTS ke jaringan *GPRS*. Komponen ini berfungsi untuk mengantarkan paket data ke MS, *update* pelanggan ke HLR, registrasi pelanggan baru.
- PCU : komponen di *level* BSS yang menghubungkan terminal ke jaringan *GPRS*.

2.1.2 J2ME

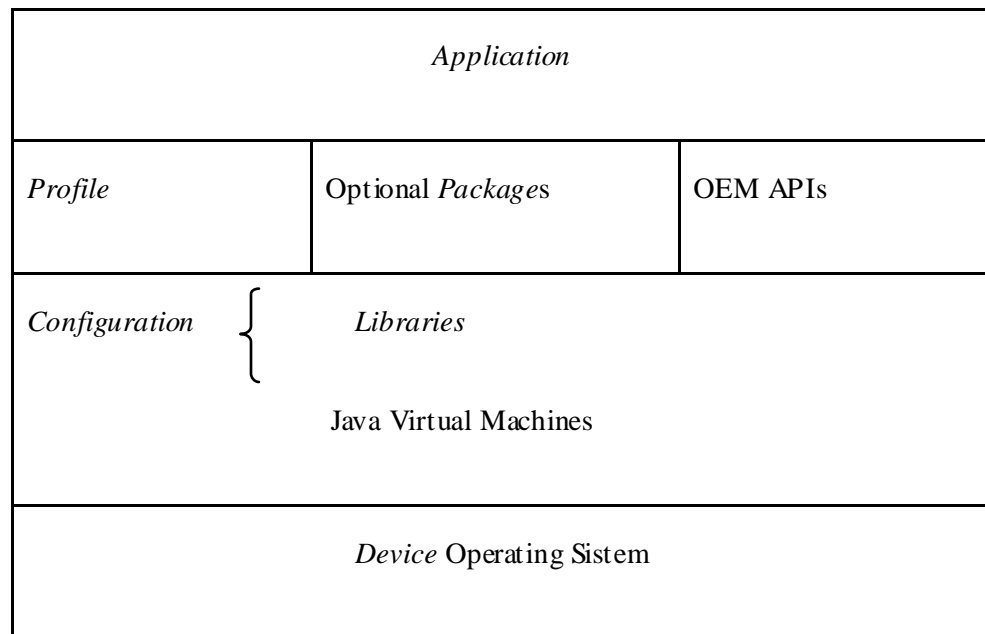
A. Gambaran *J2ME*

Java 2 Micro Edition (J2ME) adalah *Java* versi Sun yang ditujukan untuk mesin-mesin dengan sumber daya *hardware* yang terbatas seperti *PDA*, telepon seluler, dan elektronik konsumen dan perangkat *embedded*. Jadi bisa dikatakan *J2ME* adalah satu set spesifikasi dan teknologi yang fokus kepada perangkat konsumen. Perangkat ini memiliki jumlah memori yang terbatas, menghabiskan sedikit daya dari baterai, layar yang kecil dan bandwidth jaringan yang rendah.

Dengan perkembangbiakan perangkat *mobile* konsumer dari telepon, *PDA*, kotak permainan ke peralatan-peralatan rumah, *Java* menyediakan suatu lingkungan yang portable untuk mengembangkan dan menjalankan aplikasi pada perangkat ini. Program *J2ME* ini sendiri sama seperti semua program *JAVA*

yaitu diterjemahkan oleh VM (*Virtual Machine*). Program-program tersebut dikompil ke dalam *bytecode* dan diterjemahkan dengan *Java Virtual Machine*(JVM).Ini berarti bahwa program-program tersebut tidak berhubungan langsung dengan perangkat.

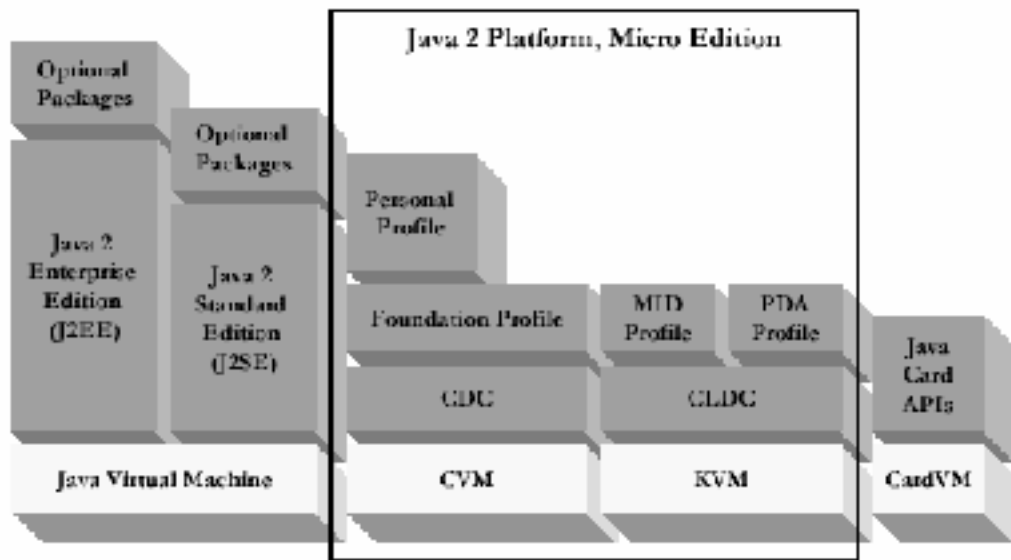
J2ME menyediakan suatu *interface* yang sesuai dengan perangkat. Aplikasi-aplikasi tersebut tidak harus dikompil ulang supaya mampu dijalankan pada mesin yang berbeda. Inti dari *J2ME* terletak pada *configuration* dan *profile-profile*. Suatu *configuration* menggambarkan lingkungan *runtime* dasar dari suatu sistem *J2ME*. Ia menggambarkan *core library*, *virtual machine*, fitur keamanan dan jaringan.



Gambar 2.1 Interface pada J2ME

Sebuah *profile* memberikan *library* tambahan untuk suatu kelas tertentu pada sebuah perangkat. profil-profil menyediakan *user interface(UI)* API, *persistence*, *messaging library*, dan sebagainya.

Satu set *library* tambahan atau *package* tambahan menyediakan kemampuan program tambahan. Pemasukan *package* ini ke dalam perangkat *J2ME* dapat berubah-ubah karena tergantung pada kemampuan sebuah perangkat. Sebagai contoh, beberapa perangkat *MIDP* tidak memiliki *Bluetooth built-in*, sehingga *Bluetooth API* tidak disediakan dalam perangkat ini.



Gambar 2.2 Lapisan J2ME

Dilihat dari Gambar diatas maka *J2ME* (*Java 2 Micro Edition*) mempunyai lapisan konfigurasi dan *profile* yang

didukung oleh *Java Virtual Machine (C-Virtual Machine dan K-Virtual Machine)*.

(Sumber : <http://java.sun.com/javame/technology/index.jsp>)

B. Lapisan J2ME

J2ME sendiri ini dibagi menjadi dua buah bagian diantaranya ialah bagian *configuration* dan *profile*.

1. Lapisan Konfigurasi (*Configuration Layer*)

Configuration sendiri ditentukan perkembangannya oleh JCP (*Java Community Process*), inilah badan non-profit yang berkuat dengan perkembangan teknologi *Java*. *J2ME* mempunyai dua konfigurasi yaitu *Connected Limited Device Configuration (CLDC)* dan *Connected Device Configuration (CDC)*. Berikut ini adalah perbandingan antara CLDC dan CDC.

Tabel 2–1 Tabel Perbandingan CLDC dan CDC

CLDC (<i>Connected Limited Device Configuration</i>)	CDC (<i>Connected Device Configuration</i>)
Mengimplementasikan subset dari J2ME.	Mengimplementasikan seluruh fitur dari J2ME.

JVM yang digunakan adalah KVM.	JVM yang digunakan adalah CVM.
Digunakan pada perangkat handheld (handphone, PDA, two way pager) dengan memory terbatas (160-512 kb).	Digunakan pada perangkat handheld (internet TV, Nokia Communicator, car TV) dengan memory minimal 2 Mb.
Prosesor : 16/ 32 bit.	Prosesor : 32 bit.

2. Lapisan Profil (*Profile Layer*)

Profile merupakan kebalikan dari *configuration* yaitu mengatur hal-hal yang spesifik untuk sebuah *device* atau tipe *market*. Misalkan pada *profile* ini diatur tentang *persistent storage* dan UI. Saat ini JCP telah mendefinisikan lima buah *profile*, salah satunya yaitu *MIDP* yaitu *profile* yang digunakan pada banyak *mobile devices* seperti Siemens M50 dan SL45i.

J2ME mempunyai beberapa profil antara lain :

1. *Mobile Information Device Profile (midp)*
2. *Foundation Profile (FP)*
3. *Personal Profile*
4. *Personal Digital Assistance (PDA) Profile*

2.1.3 PHP

A. Definisi PHP

PHP (akronim dari *PHP Hypertext Preprocessor*) ini bahasa pemrograman *script* yang paling banyak dipakai saat ini. PHP banyak dipakai untuk memrogram situs *web* dinamis, walaupun tidak tertutup kemungkinan digunakan untuk pemakaian lain.

Contoh terkenal dari aplikasi PHP adalah phpBB dan MediaWiki (*software* di belakang Wikipedia). PHP juga dapat dilihat sebagai pilihan lain dari ASP.NET/C#/VB.NET Microsoft, ColdFusion Macromedia, JSP/Java Sun Microsystems, dan CGI/Perl. Contoh aplikasi lain yang lebih kompleks berupa CMS yang dibangun menggunakan PHP adalah Mambo, Joomla!, Postnuke, Xaraya, dan lain-lain.

PHP dikatakan sebagai sebuah *server-side embedded script language* artinya sintaks-sintaks dan perintah yang diberikan akan sepenuhnya dijalankan oleh *server* tetapi disertakan pada halaman HTML biasa. Aplikasi-aplikasi yang dibangun oleh PHP pada umumnya akan memberikan hasil pada *web browser*, tetapi prosesnya secara keseluruhan dijalankan di *server*.

Pada prinsipnya *server* akan bekerja apabila ada permintaan dari *client*. Dalam hal ini *client* menggunakan kode-kode PHP untuk mengirimkan permintaan ke *server* (dapat dilihat pada gambar dibawah). Ketika menggunakan PHP sebagai *server-side embedded script language* maka *server* akan melakukan hal-hal sebagai berikut :

- ☐ Membaca permintaan dari *client/browser*
- ☐ Menca ri halaman/*page* di *server*
- ☐ Melakukan instruksi yang diberikan oleh PHP untuk melakukan modifikasi pada halaman/*page*.
- ☐ Mengirim kembali halaman tersebut kepada *client* melalui internet atau intranet.

(Sumber : <http://www.tizag.com/phpT/>)

B. Kelebihan PHP dari Bahasa Pemrograman lain

- Bahasa pemrograman PHP adalah sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaanya.
- *Web Server* yang mendukung PHP dapat ditemukan dimana - mana dari mulai apache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relatif mudah.

- Dalam sisi pengembangan lebih mudah, karena banyaknya milis - milis dan *developer* yang siap membantu dalam pengembangan.
- Dalam sisi pemahaman, PHP adalah bahasa scripting yang paling mudah karena memiliki referensi yang banyak.
- PHP adalah bahasa open source yang dapat digunakan di berbagai mesin (Linux, Unix, Macintosh, *Windows*) dan dapat dijalankan secara *runtime* melalui console serta juga dapat menjalankan perintah-perintah sistem.

C. Instalasi

Setelah berhasil mendownload Appserv versi 2.5.4a. *Software* ini bentuknya *application*, sehingga bisa langsung melakukan *instalasi* seperti *software* lainnya.

Setelah berhasil melakukan *instalasi*, jika tidak dilakukan perubahan *directory* ketika menginstalnya maka *software* tersebut akan otomatis tersimpan di *directory* "c:\appserv\", pilih typical, *server name* pilih *localhost*, *username* isikan *root*, *password* kosongkan.

2.1.4 Database

Menurut Connolly dan Begg (2002, p.14-15), database adalah kumpulan data yang terhubung secara logis (dan deskripsinya), dirancang untuk memenuhi kebutuhan informasi suatu organisasi. Semua data terintegrasi dengan jumlah duplikasi yang minimum. Database adalah

tempat penyimpanan data tunggal yang besar, didefinisikan sekali dan digunakan berulang kali oleh banyak pengguna dan departemen.

Dengan adanya database yang memisahkan struktur data dengan program aplikasi, apabila terjadi perubahan pada struktur data tidak mempengaruhi program aplikasi.

2.1.5 Entity Relationship Diagram (ERD)

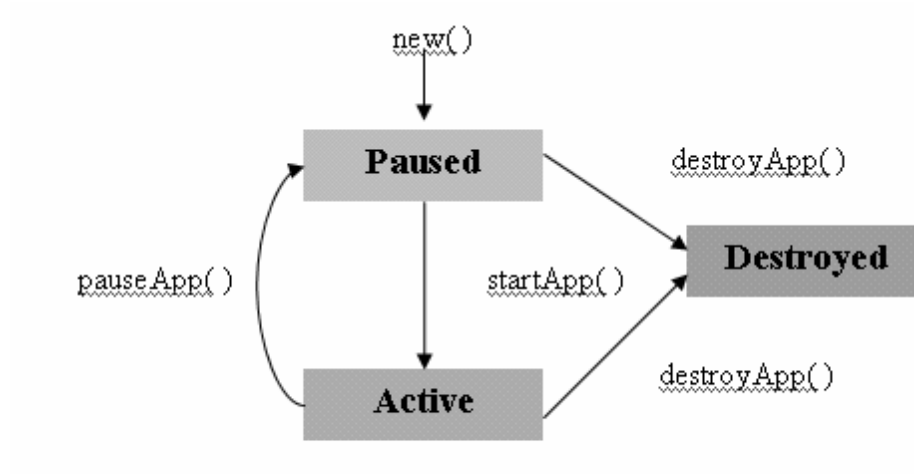
Menurut Conolly dan Begg (2002, p.425), akan lebih mudah menggambarkan sebuah sistem yang kompleks daripada memahami user requirement specification yang berupa text yang panjang. Entity Relationship Diagram digunakan untuk mempermudah dalam merepresentasikan entity dan bagaimana mereka berhubungan satu sama lainnya. Selama tahap perancangan database, ERD sebaiknya digunakan kapanpun bila diperlukan untuk membangun suatu gambaran mengenai sistem yang kita buat.

2.2 Teori Khusus

2.2.1 Daur Hidup (Lifecycle) Midlet

Lifecycle dari sebuah MIDlet ditangani oleh *Application Management Software* (AMS). AMS adalah sebuah lingkungan tempat siklus dari sebuah MIDlet, mampu untuk diciptakan, dijalankan, dihentikan maupun dihilangkan. AMS sering pula disebut dengan *Java Application Manager* (JAM). MIDlet memiliki beberapa *state*, yaitu *Pause*, *Active* dan *Destroy*. Ketika masing-masing *state* dipanggil,

beberapa *method* yang bersesuaian dipanggil. *Method-method* tersebut merupakan bawaan dari *J2ME*.



Gambar 2.3 LifeCycle dan perubahan status MIDlet

A. High Level API

Kelas-kelas yang menyediakan fungsionalitas untuk pembuatan GUI pada *MIDP* ada pada paket *javax.microedition.lcdui*. Pada paket tersebut terdapat tiga *interface* dan 21 kelas. *Interface* tersebut adalah *Display*, *Screen* dan *Form*.

B. Low Level API

Pada *level* pemrograman yang lebih rendah (*low level*), akan ditemukan fungsionalitas yang lebih spesifik ke jenis *handheld* yang digunakan. Kelas-kelas untuk pemrograman GUI pada *level* yang

lebih rendah ini diimplementasikan oleh kelas *javax.microedition.lcdui.Canvas* dan *javax.microedition.lcdui.Graphics*. Kelas *Canvas* ini memungkinkan pengguna untuk menggambar.

2.2.2 Perancangan Dan Implementasi pada *J2ME*

Pada *Java 2 Platform* aplikasi yang digunakan bertujuan untuk menampilkan tampilan *Interface* Pengisian KRS dengan menggunakan program *Java* yang dihubungkan dengan PHP. Dengan menggunakan Aplikasi *J2ME* program dapat ditampilkan dalam bentuk *screen* tampilan yang baik. Jika aplikasi ditingkatkan *level* yang lebih tinggi seperti *client server* pada *Java 2* khususnya *J2ME* bisa dengan menghubungkan ke dalam Basis data.

Dalam pembuatan ini digunakan PHP sebagai tempat penyimpanan data. Karena dalam *Java 2* peran PHP tidak dapat begitu saja ditinggalkan, maka dibuat satu aplikasi PHP yang dapat melakukan *respon-request* terhadap aplikasi *J2ME*.

Untuk *server* ini dibuat dengan menggunakan tomcat dengan alamat *localhost* untuk menjalankan aplikasi *Java*. Sedangkan bentuk simulasinya dapat menggunakan *emulator-emulator* yang sudah ada dalam *J2ME-toolkit* atau *emulator* handphone *symbian* yang digunakan. Percobaan pengisian KRS akan dilakukan pada aplikasi yang berhasil dijalankan dalam *emulator J2ME* ke dalam *emulator* simulasi handphone dengan tidak menghapus dan menghilangkan keaslian data.

Aplikasi MIDlet ini dijalankan dalam program berbasis *J2ME* untuk *handheld device* memerlukan JDK 1.3 atau versi yang lebih tinggi, kemudian *J2ME Wireless Toolkit* untuk pengembangan aplikasinya. Selain itu juga menggunakan *software* pendukung lainnya, seperti J2SDKEE versi 1.3.1, Apache-Tomcat-4.0.1 dan *Nokia Mobility Toolkit*.

A. *J2ME Wireless Toolkit*

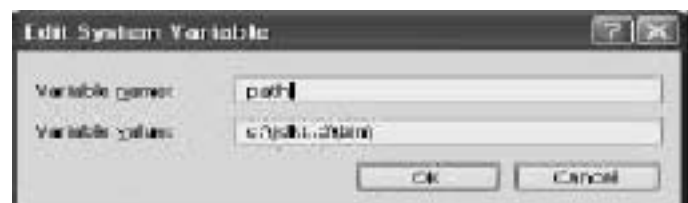
J2ME Wireless Toolkit ini digunakan untuk mengembangkan aplikasi *Java* yang telah dibuat. *J2ME Wireless Toolkit* yang digunakan dalam pembuatan MIDlet ini adalah *J2ME Wireless Toolkit* versi 1.0.4.

B. Setting Instalasi

Setting Instalasi yang perlu dilakukan ialah melakukan *setPATH* dan *Classpath* pada *Java Development Kit* (JDK), *Java 2 Microedition* (*J2ME*), *Java 2 Enterprise Edition* (*J2EE*), dan Tomcat. Tanpa melakukan *setPATH* program *Java* tidak dapat dijalankan pada *Platform*.

(<http://developers.sun.com/mobility/learn/midp/lifecycle/#jadsnjars>)

Setting -> Control Panel -> Sistem -> Advanced -> *Environment Variables*



Gambar 2.4 *setPATH* JDK Sistem Variabel

2.2.3 Kuesioner

Menurut **Sugiyono (2007, p.199)**, kuesioner merupakan teknik pengumpulan data yang dilakukan dengan cara memberi seperangkat pertanyaan atau pernyataan tertulis kepada responden untuk dijawab. Kuesioner merupakan teknik pengumpulan data yang efisien bila peneliti tahu dengan pasti *variabel* yang akan diukur dan tahu apa yang bisa diharapkan dari responden. Selain itu, kuesioner dapat berupa pernyataan/pertanyaan tertutup dan terbuka, dapat diberikan kepada responden secara langsung, maupun melalui internet (tidak langsung).

Bila penelitian dilakukan pada lingkup yang tidak terlalu luas, sehingga kuesioner dapat diantarkan langsung dalam waktu tidak terlalu lama, maka penerimaan angket tidak perlu melalui pos, dengan adanya kontak langsung peneliti dengan responden akan menciptakan suatu kondisi yang cukup baik, sehingga responden dengan sukarela akan memberikan data objektif dengan cepat.

2.2.4 Populasi dan Sampel

A. Populasi

Menurut **Sugiyono (2007, p.61)**, populasi adalah wilayah generalisasi yang terdiri atas : objek/subjek yang mempunyai kualitas

dan karakteristik tertentu yang ditetapkan oleh peneliti untuk dipelajari dan kemudian ditarik kesimpulannya.

Jadi populasi bukan hanya orang, tetapi juga objek dan benda-benda alam yang lain. Populasi juga bukan sekedar jumlah yang ada pada objek/subjek yang dipelajari, tetapi meliputi seluruh karakteristik/sifat yang dimiliki oleh subjek/objek itu sendiri.

Misalnya akan melakukan penelitian di perusahaan A, maka perusahaan A itu merupakan populasi. Perusahaan A ini mempunyai sejumlah orang/subjek yang lain. Hal ini berarti populasi dalam arti jumlah/kuantitas. Tetapi perusahaan X juga mempunyai karakteristik karyawan-karyawannya, misalnya motivasi kerja karyawan, disiplin kerja karyawan, kepemimpinan, dan lain-lain. Dan juga mempunyai objek yang lain, misalnya prosedur kerja, kebijakan dalam perusahaan, dan lain-lain. Yang terakhir berarti populasi dalam arti karakteristik.

Penggunaan satu orang saja sebagai populasi juga dapat dilakukan, karena satu orang itu mempunyai berbagai karakteristik, misalnya gaya bicaranya, disiplin pribadi, hobi, cara bergaul, kepemimpinannya. Misalnya akan melakukan penelitian tentang kepemimpinan Direktur SY maka kepemimpinan itu merupakan sampel dari semua karakteristik yang dimiliki Direktur SY.

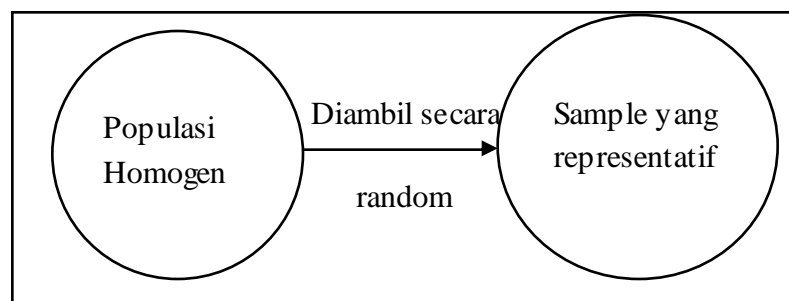
B. Sample

Menurut **Sugiyono (2007, p.73)**, sampel adalah bagian dari jumlah dan karakteristik yang dimiliki populasi tersebut. Bila populasi besar, dan peneliti tidak mungkin mempelajari semua yang ada pada populasi, misalnya karena keterbatasan dana, tenaga dan waktu, maka penelitian dapat menggunakan sampel yang diambil dari populasi itu. Apa yang dipelajari dari sampel itu, kesimpulannya akan diberlakukan pada populasi. Untuk itu sampel yang diambil dari populasi harus betul-betul mewakili populasi itu sendiri.

Bila sampel tidak mewakili populasi itu sendiri, maka orang awam akan sulit menyimpulkan karakteristik populasi itu.

A. Simple Random Sampling

Dikatakan simple menurut **Sugiyono (2004, p.108)**, karena pengambilan sampel anggota populasi dilakukan secara acak tanpa memperhatikan strata yang ada dalam populasi itu. Cara demikian dilakukan bila anggota populasi dianggap homogen.



Gambar 2.5 Teknik Random Sampling

2.2.5 Menentukan ukuran sampel

Jumlah anggota sampel sering dinyatakan dengan ukuran sampel. Jumlah sampel yang 100% mewakili populasi adalah sama dengan jumlah anggota populasi. Jadi bila jumlah populasi dan hasil penelitian itu diberlakukan untuk 1000 orang tersebut tanpa ada kesalahan, maka jumlah sampel yang diambil sama dengan jumlah populasi tersebut, yaitu 1000 orang. Makin besar sampel mendekati populasi, maka peluang kesalahan *generalisasi* semakin kecil dan sebaiknya makin kecil jumlah sampel menjauhi populasi, maka semakin besar kesalahan *generalisasi* (diberlakukan umum).

2.3 Analisis Dan Perancangan Sistem

2.3.1 Model Proses *Waterfall*

Menurut Pressman (2001, p.28), sekuensial linier atau lebih dikenal dengan sebutan *waterfall* model menawarkan sebuah pendekatan yang sistematis dan sekuensial dalam pengembangan piranti lunak yang dimulai dari lebel sistem dan perkembangannya dengan melalui tahap analisis, desain, coding dan testing.

Model *waterfall* adalah versi paling populer dari *system development lifecycle model* untuk *software engineering*. Model ini sering dianggap sebagai pendekatan klasik dalam daur hidup pengembangan sistem. Ada empat tahapan utama dalam model ini, yang

sering dikembangkan oleh orang sesuai kebutuhan pada saat penelitian dilakukan. Keempat cakupan utama dalam model *waterfall* adalah:

1. Analisis

Analisis adalah sebuah proses pengumpulan kebutuhan yang dikhususkan dan difokuskan dalam pembuatan piranti lunak.

2. Desain

Desain adalah sebuah proses yang menerjemahkan hasil dari analisis dalam bentuk representasi piranti lunak sehingga dapat dinilai kualitasnya sebelum proses coding dimulai.

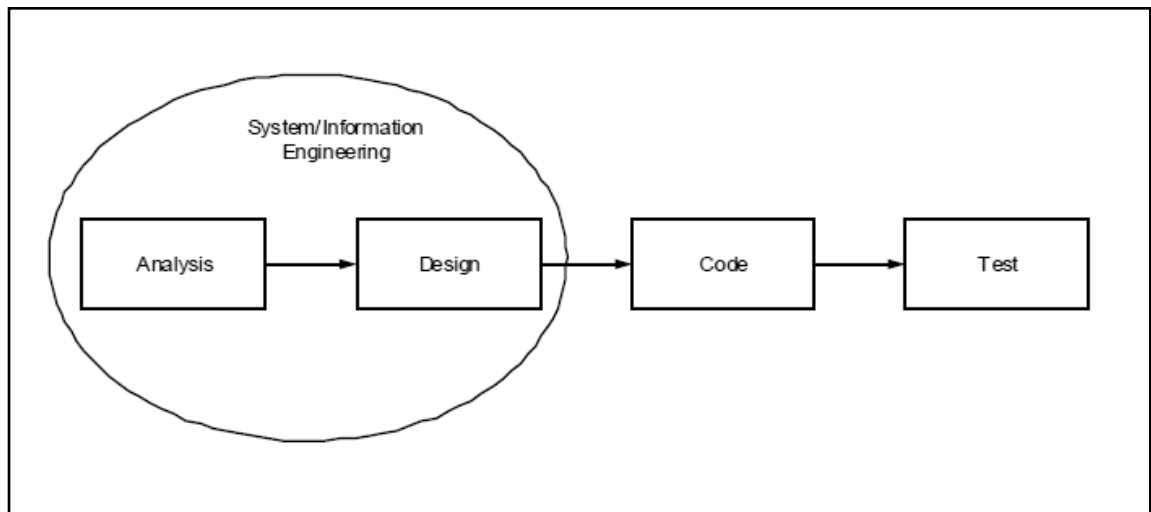
3. Coding

Coding adalah proses dimana hasil dari desain diterjemahkan kembali dalam bentuk bahasa pemrograman yang dapat dimengerti oleh mesin.

4. Testing

Setelah coding dilakukan, maka dilakukan tahap testing untuk mengetahui kesalahan yang tidak terdeteksi sebelumnya atau hasil dari proses yang tidak diinginkan.

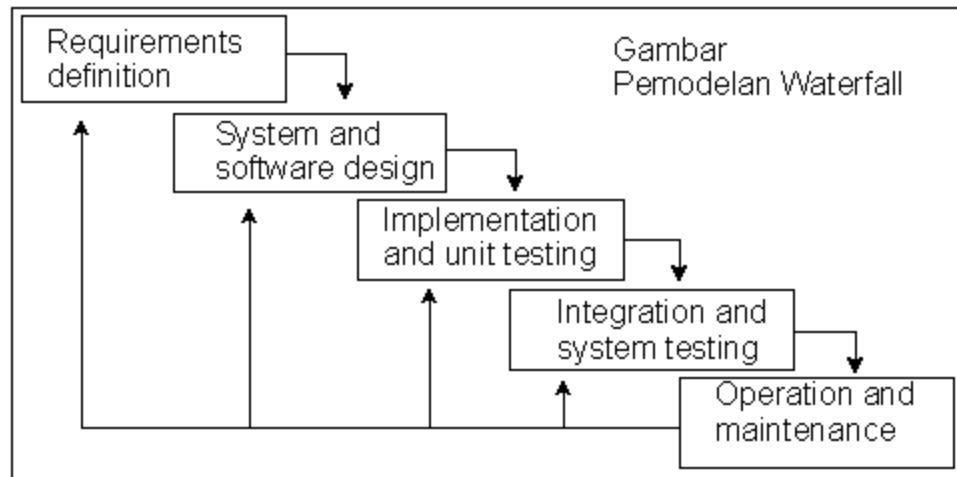
Adapun gambaran dari metodologi *waterfall* ialah sebagai berikut:



Gambar 2.6 Gambar Model Metodologi *Linear Sequential* atau *Waterfall*

(Sumber : Pressman, Software Engineering, 2001, p.29)

Waterfall model juga sering disebut sebagai *classic lifecycle*, yang pada intinya memiliki alir proses utama yang sama. Proses ini dapat dilihat pada gambar 2-7 berikut:



Gambar 2.7 Gambar Model Metodologi *Classic Lifecycle* atau *Waterfall*

(Sumber : <http://tonyjustinus.wordpress.com/2007/11/11/waterfall-process-model/>)

2.3.2 Unified Modelling Language

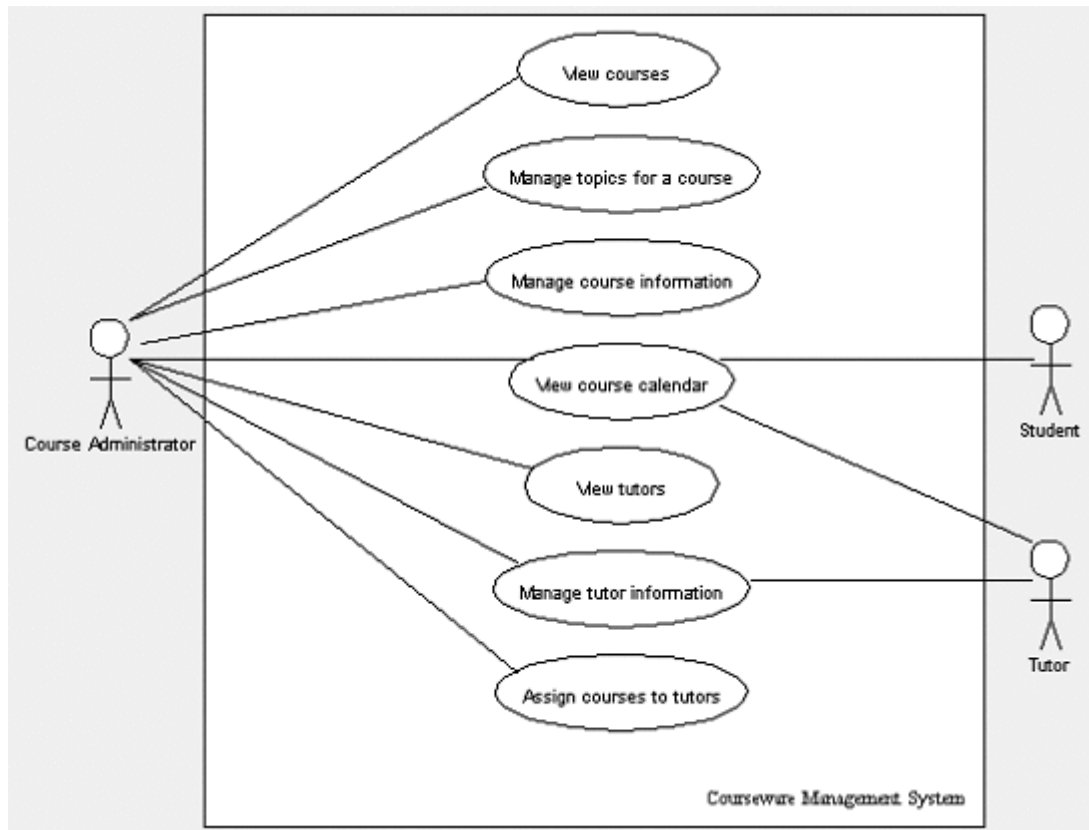
Unified Modeling Language (UML) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. UML tidak berdasarkan pada bahasa pemrograman tertentu. Standar spesifikasi UML dijadikan standar *defacto* oleh OMG (*Object Management Group*) pada tahun 1997. UML yang berorientasikan *object* mempunyai beberapa notasi standar.

Spesifikasi ini menjadi populer dan standar karena sebelum adanya UML, telah ada berbagai macam spesifikasi yang berbeda. Hal ini menyulitkan komunikasi antar pengembang perangkat lunak. Untuk itu beberapa pengembang spesifikasi yang sangat berpengaruh berkumpul untuk membuat standar baru. UML dirintis oleh Grady Booch, James Rumbaugh pada tahun 1994 dan kemudian Ivar Jacobson. (Sumber : <http://www.uml.org/>)

UML mendeskripsikan OOP (*Object Oriented Programming*) dengan beberapa diagram, diantaranya:

A. Use Case diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem.



Gambar 2.8 Contoh Use Case Diagram

B. Class diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (*atribut*/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (*metoda*/fungsi).

Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.

Class memiliki tiga area pokok :

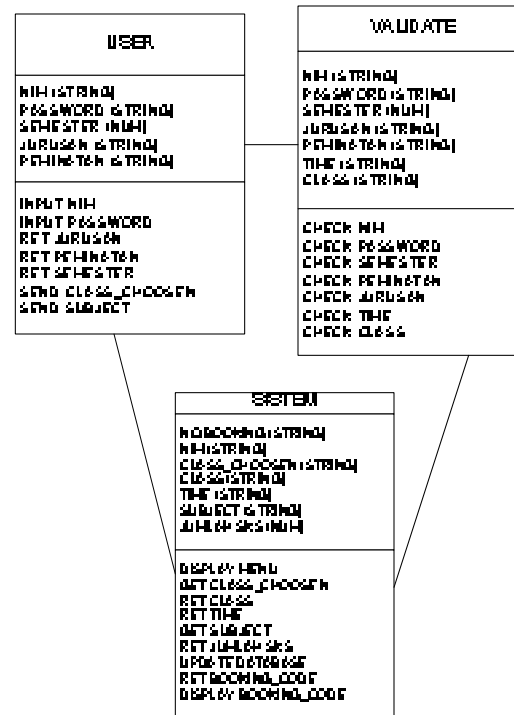
B. Nama dan *stereotype*

C. Atribut

D. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :

- *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan.
- *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya.
- *Public*, dapat dipanggil oleh siapa saja.

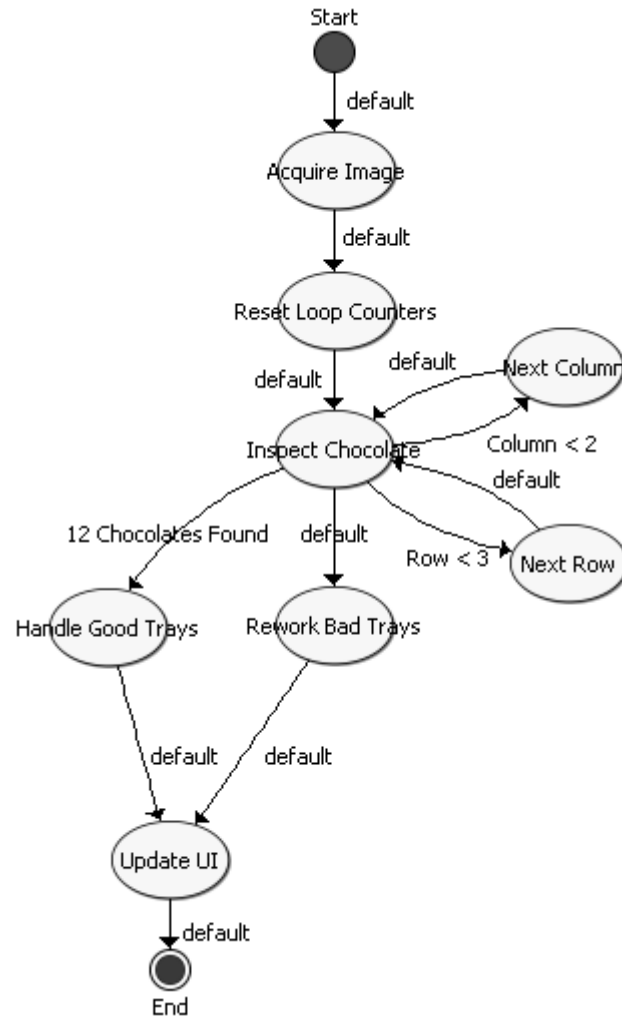


Gambar 2.9 Contoh Class Diagram

C. Behavior diagram :

1. Statechart diagram

Statechart diagram menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari stimuli yang diterima. Pada umumnya *statechart* diagram menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart* diagram).

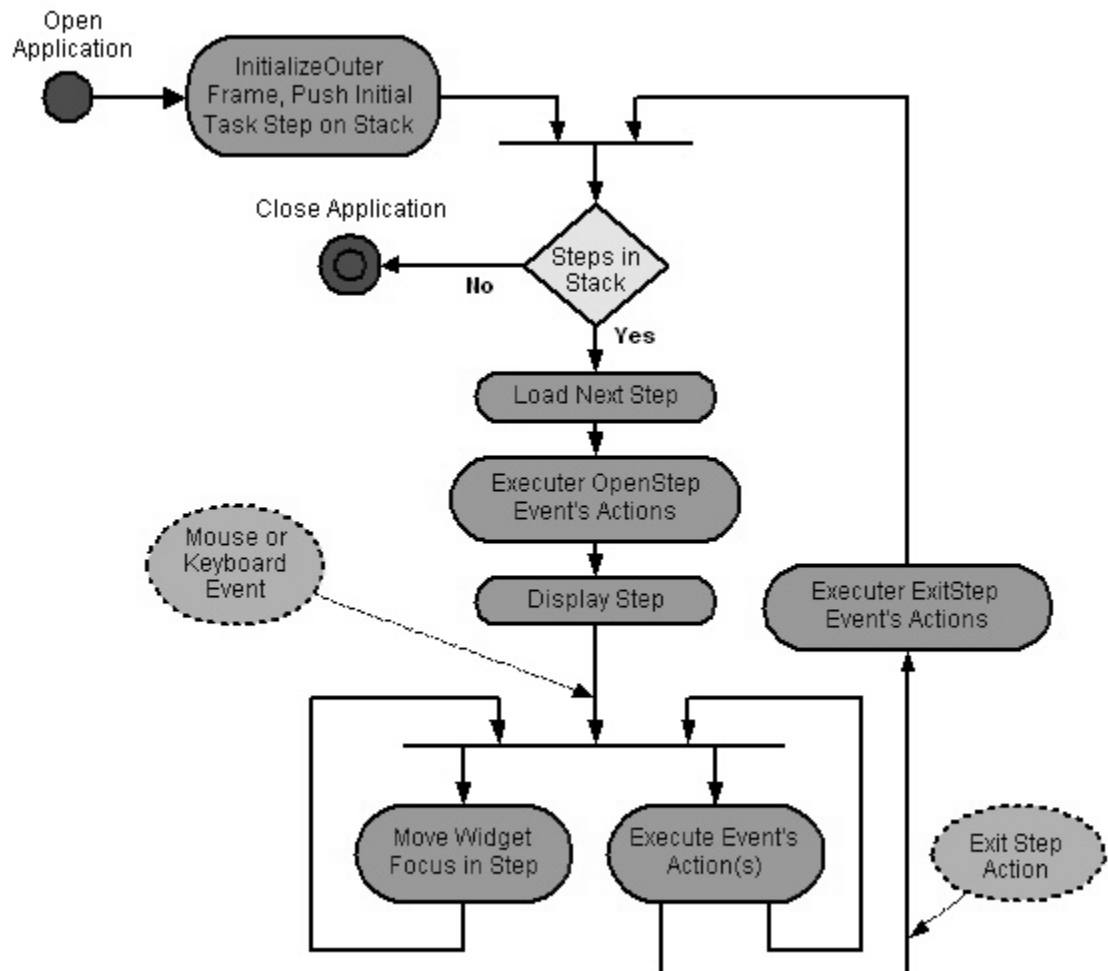


Gambar 2.10 Contoh State Diagram

2. Activity diagram

Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity* diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Activity diagram merupakan *state* diagram khusus, di mana sebagian besar *state* adalah action dan sebagian besar transisi di-trigger oleh selesainya *state* sebelumnya (internal processing). Oleh karena itu *activity* diagram tidak menggambarkan *behaviour* internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari *level* atas secara umum.



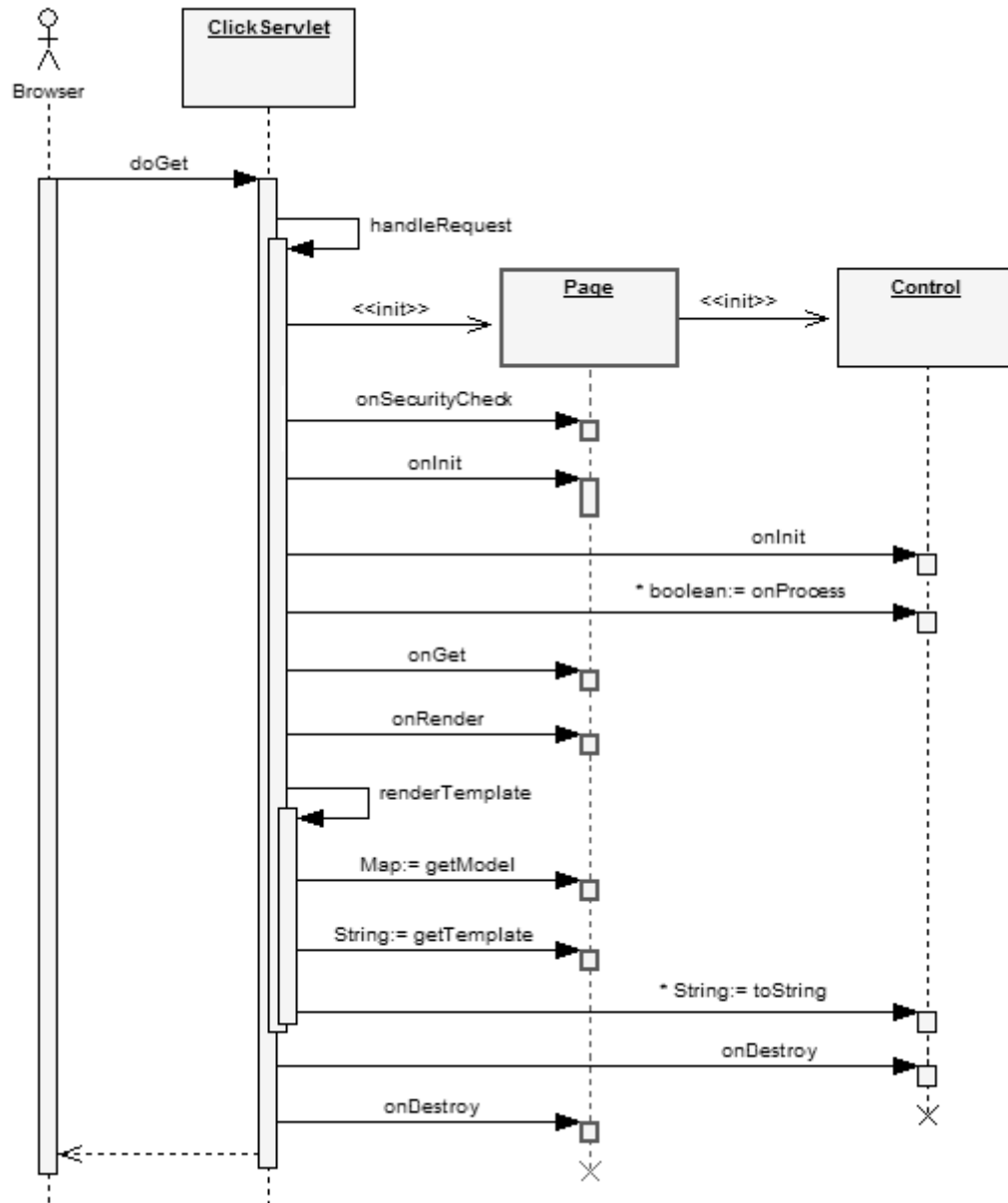
Gambar 2.11 Contoh Activity Diagram

D. Interaction diagram :

1. sequence diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu. *Sequence* diagram terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Sequence diagram biasa digunakan untuk menggambarkan *skenario* atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang men-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan.



Gambar 2.12 Contoh *Sequence Diagram*

2.3.3 Analisis Aplikasi Sejenis

Dalam melakukan riset pengembangan aplikasi *mobile* KRS ini, dilakukan survey terlebih dahulu apakah adanya aplikasi yang sejenis dalam mengisi KRS. Adapun dari kebutuhan pengisian KRS di berbagai universitas menyebabkan banyaknya pengembang yang ingin menciptakan suatu hal yang baru dari pengisian KRS itu sendiri.

Dari survey yang dilakukan, didapatkan beberapa jenis aplikasi yang mirip dalam arti satu konsep yaitu pemesanan tempat. Aplikasi tersebut ialah aplikasi M-ticketing yang telah diterapkan pada bioskop-bioskop.

Melalui aplikasi M-ticketing diharapkan dapat membantu memudahkan pelanggan untuk melakukan pemesanan tiket bioskop melalui telepon genggam dimana saja dan kapan saja, dan juga memberikan informasi mengenai jadwal pemutaran film dan jumlah tiket yang masih tersisa. Selain itu aplikasi ini juga bertujuan untuk memberikan kenyamanan bagi pelanggan sehingga tidak perlu mengantri jika ingin melihat film yang diinginkan. Agar dapat melakukan pemesanan tiket, pelanggan dapat melakukan pendaftaran melalui telepon genggam secara online.

A. Spesifikasi Produk

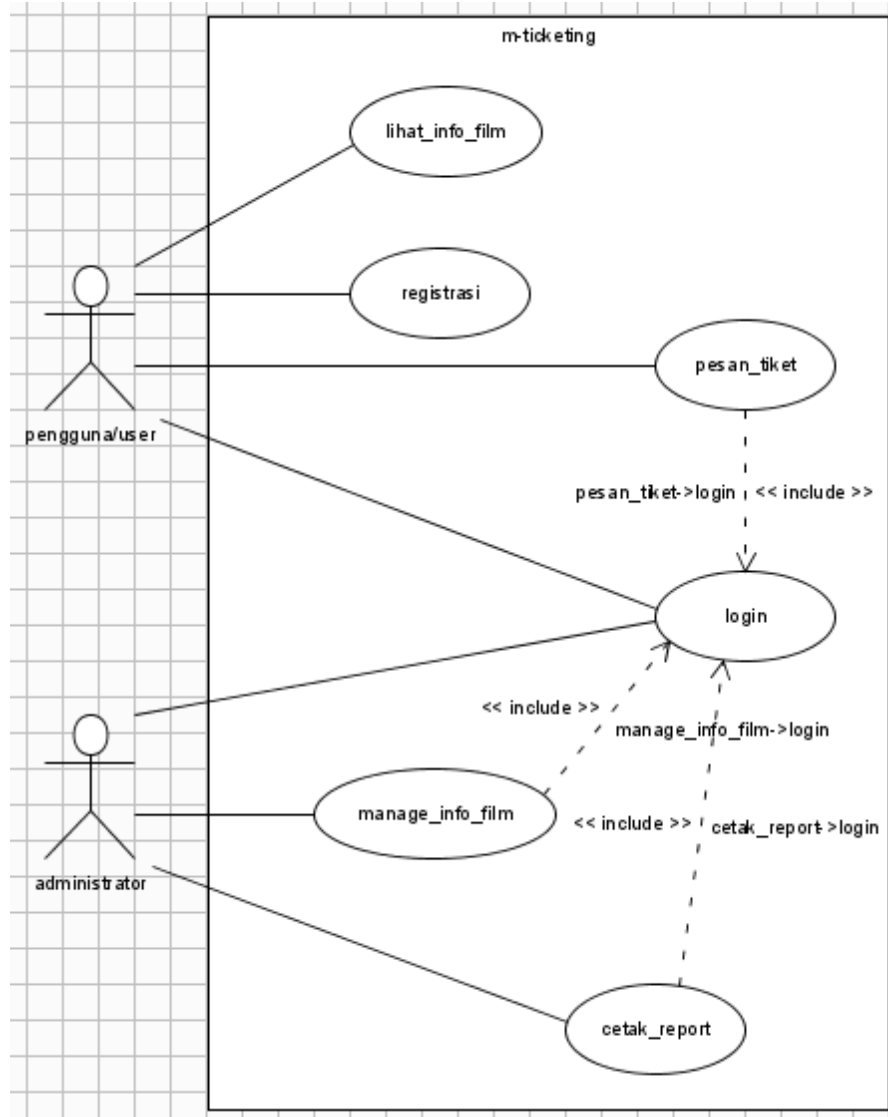
Aplikasi pemesanan tiket bioskop ini dirancang dengan menggunakan bahasa pemrograman WML dan PHP untuk aplikasi dalam telepon genggam, dan bahasa pemrograman C# untuk aplikasi *administrator*. Aplikasi ini juga memanfaatkan *Apache* sebagai web service, dan

MySQL sebagai program untuk penyimpanan data. Komunikasi dilakukan dengan menggunakan saluran GPRS (*General Pocket Radio System*), sehingga agar dapat mengakses aplikasi pemesanan tiket bioskop ini diperlukan telepon genggam yang dilengkapi dengan fasilitas WAP.

Aplikasi pemesanan tiket melalui telepon genggam mencakup:

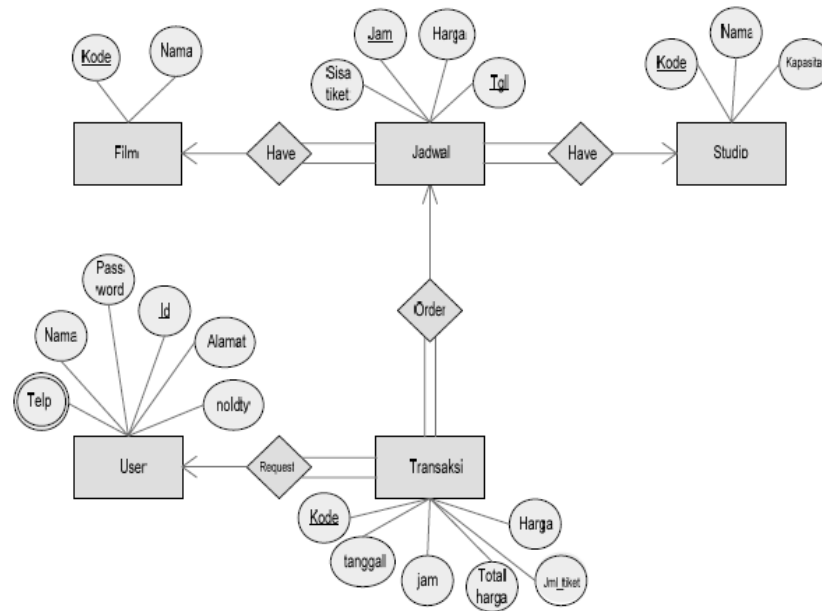
1. Memberi informasi kepada pelanggan tentang jadwal pemutaran film yang mencakup tanggal dan jam pemutaran film, nama film, tempat pemutaran film, dan sisa tiket.
2. Transaksi pemesanan tiket bioskop, dengan batas maksimal jumlah tiket yang dapat dipesan sejumlah 4 tiket. Melalui proses ini pelanggan dapat memilih film yang diinginkan berdasarkan tanggal pemutaran film, jam tayang, jumlah tiket, dan nomor kursi.
3. Proses *login* yang harus dilakukan sebelum melakukan transaksi pemesanan. Pelanggan yang hendak memesan tiket harus sudah terdaftar terlebih dahulu.
4. Proses pendaftaran melalui telepon genggam, dimana pelanggan akan memasukkan data yang diperlukan agar dapat melakukan transaksi pemesanan tiket.

B. Penggambaran Sistem dengan Use Case Diagram



Gambar 2.13 Gambar use case diagram M-Ticketing

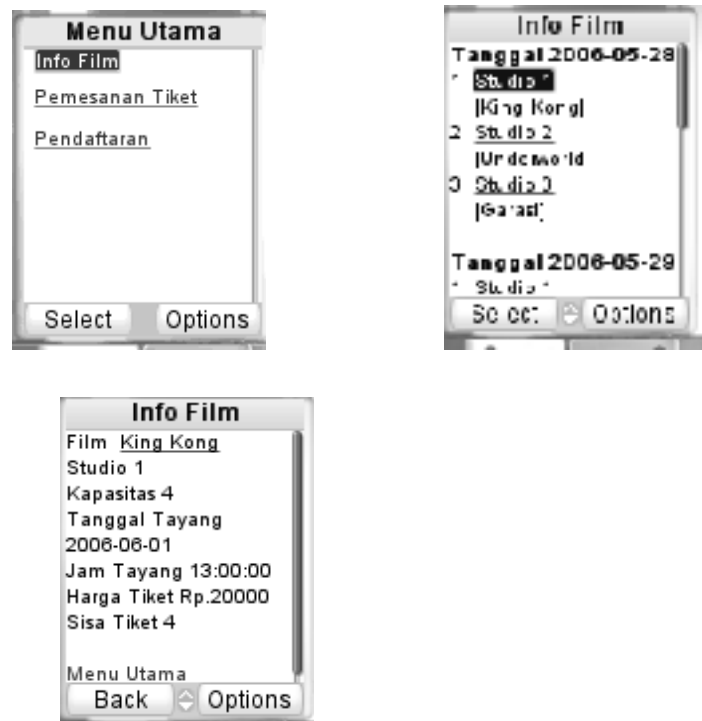
C. Penggambaran Hubungan antar Entitas



Gambar 2.14 Gambar ER diagram Aplikasi M-Ticketing

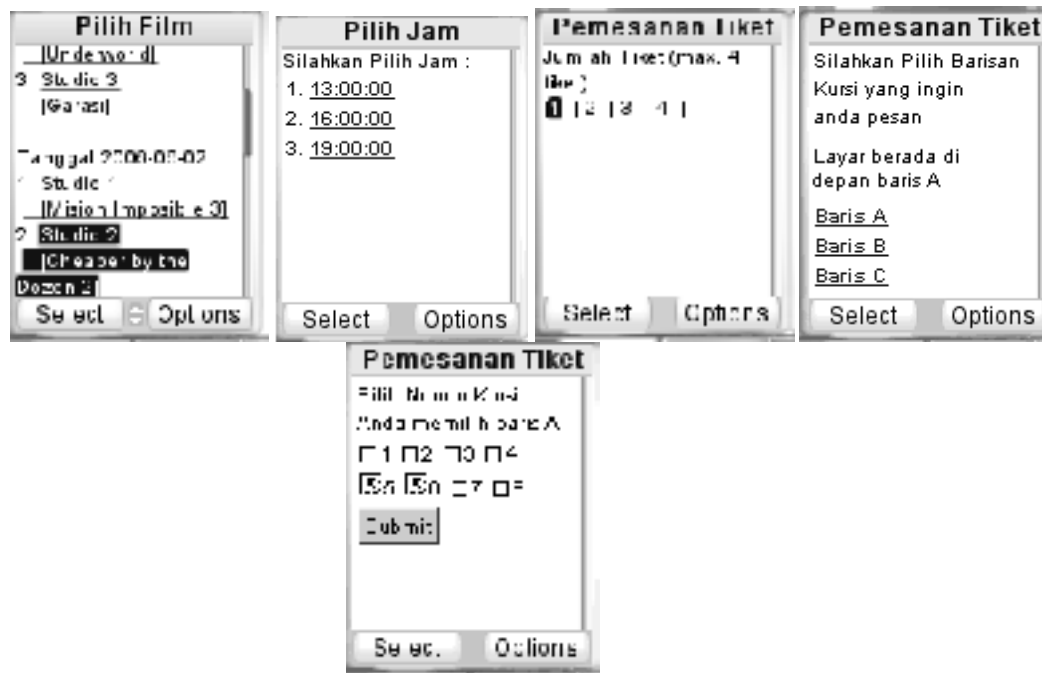
D. Antar Muka Aplikasi

Rancangan layar yang terdapat pada aplikasi M-ticketing ini ialah :



Gambar 2.15 Gambar Tampilan menu info film

Melalui telepon genggam yang memiliki fasilitas WAP *user* akan mengakses *server* dan memperoleh informasi mengenai film-film yang sedang diputar, jadwal pemutaran film, kapasitas kursi, dan sisa tiket yang tersedia.



Gambar 2.16 Gambar Tampilan Menu Pemesanan Tiket

Pelanggan memilih data-data yang diinginkan untuk memesan tiket bioskop. Setelah melalui proses *login*, pelanggan akan diminta untuk memilih film yang diinginkan berdasarkan tanggal pemutaran film, kemudian pelanggan akan diminta untuk memilih jam tayang, dan jumlah tiket. Setelah itu akan muncul konfirmasi berisikan kode transaksi, nama film, tempat pemutaran film, tanggal dan jam tayang, jumlah tiket, harga dan total harga yang harus dibayar. Semua data yang telah diinput *user* akan dimasukkan ke dalam Basis data.

The image shows a screenshot of a web form titled "Pendaftaran User". The form has three input fields: "User ID:", "Password:", and "Re Password:". Below the fields are two buttons: "Edit" and "Options".

Gambar 2.17 Gambar Tampilan menu pendaftaran

Pelanggan memasukkan data-data berupa *user id*, *password*, nama, alamat, nomor telepon, dan nomor identitas sehingga *user* dapat melakukan transaksi pemesanan tiket bioskop. Pada fitur ini *user* memilih menu pendaftaran *user*. Kemudian akan muncul *form* pendaftaran. *User* akan diminta untuk memasukkan *user id*, *password*, nama, alamat, nomor telepon, dan nomor identitas. Data yang diinput *user* akan dikirimkan ke *server* dan data akan disimpan dalam Basis data.

Kelebihan dan kekurangan dari aplikasi M-ticketing ini adalah :

- A. Penonton tidak perlu melakukan antri untuk membeli tiket.
- B. Penonton dapat memesan tempat lebih cepat dari jam bukanya loket yang menjual tiket.
- C. Perusahaan dapat menghemat kertas tiket.
- D. Pelanggan hanya dapat memesan maksimal 4 tiket.