

Bab 2

LANDASAN TEORI

2.1 Sumber Daya Manusia

Sumber daya manusia (SDM) di dalam sebuah perusahaan memiliki peran penting dalam menentukan kemajuan suatu perusahaan. Dalam hal ini yang menjadi faktor penentu bukanlah kuantitas sumber daya manusia tersebut, melainkan kualitasnya sebagai individu. Karena sumber daya manusia yang berkualitas akan menunjang kinerja perusahaan dan menjadi penentu apakah sebuah perusahaan tersebut akan mengalami kemajuan atau hanya menjadi perusahaan biasa yang pada akhirnya tidak dapat bersaing di pasar dan kemudian mengalami kebangkrutan.

Karyawan atau SDM yang bekerja dalam perusahaan umumnya diterima melalui proses seleksi terlebih dulu. Dalam proses ini, data-data karyawan dikumpulkan sebagai catatan internal. Nama karyawan, tempat dan tanggal lahir, alamat rumah, status pernikahan, pengalaman kerja, dan latar belakang pendidikan merupakan contoh sebagian data yang biasa disimpan bagian SDM. Data-data ini menggambarkan profil karyawan dan biasanya diberikan sendiri oleh karyawan. Selanjutnya, data-data seleksi, seperti tingkat kecerdasan (IQ), tingkat emosional (EQ), aspek psikologis, dan kesehatan [9].

Tuntutan terhadap kebutuhan SDM dipengaruhi oleh keadaan dan pengaruh teknologi. Teknologi yang berubah dan masih terus akan berubah dalam arti akan ditemukan berbagai alat yang dapat menggantikan tenaga kerja manusia yang lebih baik, efektif, dan efisien. Hasil ini dapat dilihat dari sejarah peradaban manusia, pada saat ditemukan mesin uap hingga komputer dan robot yang dikendalikan oleh komputer, dan kemudian teknologi komunikasi yang sangat canggih. Pada masa yang

akan datang tentu akan ditemukan berbagai teknologi baru yang lebih efisien dan efektif untuk melakukan pekerjaan.

Temuan-temuan di atas akan sangat mempengaruhi dunia kerja, dalam hal semakin mengecilnya kebutuhan perusahaan terhadap tenaga kerja manusia karena telah digantikan oleh mesin-mesin canggih. Jenis keahlian dan keterampilan SDM yang dibutuhkan juga akan berubah ke arah penguasaan teknologi, dalam hal bagaimana mengoperasikan dan memelihara berbagai teknologi canggih itu untuk proses produksi. Dengan keadaan ini tentu saja program-program dan kegiatan SDM harus diarahkan untuk meningkatkan keterampilan yang sesuai dengan tuntutan teknologi [3].

2.2 Internet

2.2.1 Penjelasan Umum Seputar Internet

Istilah Internet berasal dari singkatan dua suku kata yaitu, *interconnected* dan *network*. Interconnected berarti hubungan yang sangat luas, sedangkan network berarti jaringan kerja. Maka internet merupakan adalah sebuah sistem komunikasi global yang menghubungkan komputer-komputer dan jaringan-jaringan komputer di seluruh dunia. Komputer dan jaringan dengan berbagai platform yang mempunyai perbedaan dan ciri khas masing-masing (Unix, Linux, Windows, Mac, dll) bertukar informasi dengan sebuah protokol standar yang dikenal dengan nama TCP/IP (Transmission Control Protocol/Internet Protocol). Komputer yang dahulunya berdiri sendiri kini dapat berhubungan langsung komputer-komputer lainnya di seluruh dunia. Jaringan ini tercipta melalui saluran komunikasi yang meliputi telepon, satelit, dan jalur telekomunikasi lainnya. Agar komunikasi dapat terhubung serta dapat menggunakan fasilitas internet, maka harus berlangganan ke salah satu ISP (Internet Service Provider) atau penyelenggara layanan internet. fasilitas-fasilitas yang tersedia dalam layanan internet antara lain :

1. WWW (*World Wide Web*) adalah layanan yang paling sering digunakan dan memiliki perkembangan yang sangat cepat karena dengan layanan ini kita bisa menerima informasi dalam berbagai format (multimedia). Informasi dalam

web disimpan ke bentuk dokumen yang disebut halaman web (*web page*). Untuk dapat mengakses layanan www dari sebuah komputer digunakan program web client yang disebut *web browser* atau browser saja. Jenis-jenis browser yang sering digunakan adalah: *Netscape Navigator/Communicator*, *Internet Explorer*, *Mozilla*, dan lain sebagainya.

2. Search engine adalah salah satu fasilitas internet yang dijalankan melalui browser untuk mencari informasi yang kita inginkan. Search engine menampung database situs-situs dari seluruh dunia yang jumlahnya milyaran halaman web, cukup dengan memasukkan kata kuncinya maka search engine akan menampilkan beberapa link situs yang disertai dengan keterangan singkat. Search engine yang sangat terkenal adalah *Google*, namun masih banyak search engine lainnya seperti *Altavista*, *MSN*, *Yahoo Search* dan sebagainya.
3. E-mail (Electronic Mail) merupakan aplikasi surat menyurat melalui Internet. Pengguna dapat saling bertukar berita. Berita-berita itu akan dikumpulkan dalam sebuah file untuk pengalamatan yang berupa mailbox, sehingga pengguna dapat membaca berita yang ditujukan kepadanya kapan saja. Namun sebelum menggunakan fasilitas dalam e-mail, kita harus terlebih dahulu terdaftar dan memiliki *account* di salah satu penyedia layanan e-mail seperti *Yahoo*.
4. FTP (File Transfer Protocol) adalah suatu protokol yang digunakan untuk melakukan pemindahan (*transfer*) baik *upload* atau *download* dari satu atau lebih file (dokumen) antara sebuah komputer dan sembarang server FTP.

2.2.2 Internet sebagai Client Server

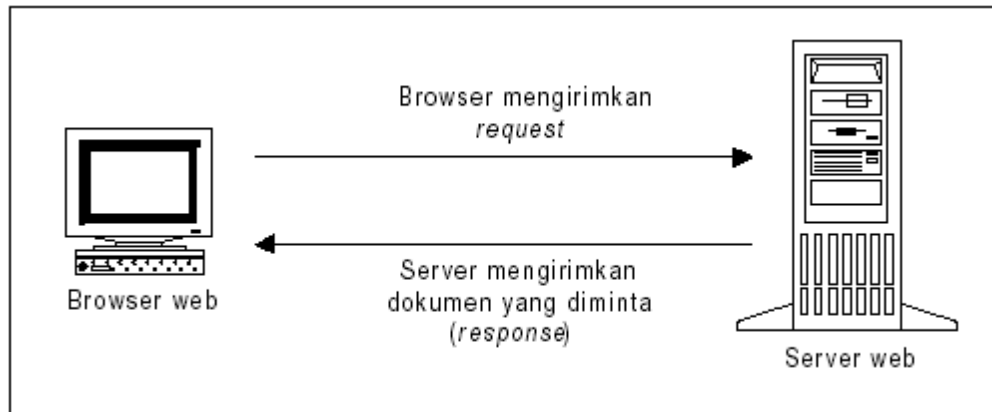
Istilah *client-server* dewasa ini telah demikian populer. Keuntungan utama dari sistem berbasis *client-server* adalah bahwa perangkat keras dan perangkat lunak bisa ditempatkan di mana saja sehingga pengguna bisa bekerja secara lebih optimal. Dahulu, di zaman komputer *mainframe*, komputer *mainframe*-lah yang menjadi pusat kendali dan mengerjakan semua proses komputasi. Pengguna berinteraksi dengan sistem *mainframe* melalui terminal-terminal yang dibutuhkan secara langsung ke

komputer *mainframe*. Terminal-terminal ini tidak punya kemampuan pemrosesan sama sekali, dan oleh karena itu disebut “terminal dungu” (dumb terminal). Terminal dungu tidak lebih dari sekedar perpanjangan kabel untuk keyboard dan layar monitor, dan hanya berfungsi sebagai alat untuk memasukkan dan melihat data saja.

Definisi yang banyak dipergunakan untuk menjelaskan sistem berbasis *client-server* adalah “sistem yang memisahkan antara tugas-tugas komputasi antara proses-proses *client* dan *server*”. Dengan sistem *client-server*, kekuatan pemrosesan bisa disebar (didistribusikan) ke banyak mesin *client* dan mesin *server* yang terpisah secara fisik (itu sebabnya disebut *distributed system*).

Misalnya sebuah *server* web yang mengambil informasi dari database menampilkan hasilnya pada *client* dengan menggunakan *web browser*. *Server web* dan *database* bisa saja ditempatkan pada satu mesin, namun apabila jumlah *client* yang melakukan akses ke *server* semakin banyak dan melebihi kapasitas mesin *server*, perangkat lunak database dan *server web* bisa saja dipisahkan dan ditempatkan di mesin kedua, ketiga atau bahkan lebih. Dengan begitu pemrosesan pada sisi *server* dapat disebar ke beberapa mesin, yang memungkinkan efisiensi komputasi. Begitu pula, dengan cara ini, kapasitas *server* bisa dikembangkan dan ditingkatkan sesuai dengan kebutuhan.

Dari sudut pandang lain, sistem berbasis *client-server* juga bisa memanfaatkan *browser* web untuk meringankan kerja *server*. Tugas menampilkan informasi dan menyediakan tampilan pengguna (*user interface*) tidak perlu dilakukan secara langsung oleh *server*, namun diserahkan sepenuhnya kepada *browser web*. Dengan hadirnya teknologi pemrograman *client-side* (bahasa pemrograman yang dijalankan di sisi *client*) seperti Javascript dan VBScript, maka fungsi-fungsi khusus seperti pemeriksaan/validasi *input* bisa dilakukan oleh *browser* sebelum data dikirimkan kepada *server*, sehingga dapat menjamin data yang dikirimkan ke *server* tidak keliru. Misalnya untuk memastikan bahwa *input* yang diisikan oleh pengguna adalah sebuah angka dapat dilakukan oleh program *client-side* sehingga nantinya *server* tidak perlu lagi memeriksa input tersebut. Hal ini mempercepat kerja *server*, karena hanya mengerjakan tugas-tugas yang berguna.



Gambar 2.1 Konsep Dasar Web Browser dan Web Server

Pada saat ini, aplikasi-aplikasi besar cenderung mengacu pada arsitektur n -tier ($n > 3$) karena lebih *scalable*. Misalkan saja untuk *3-tier* akan terdapat layer presentasi, aplikasi, dan layer *database*. Contohnya adalah aplikasi berbasis web, dimana *layer* presentasi memakai bahasa pemrograman PHP, *layer* aplikasi memakai Apache sebagai *Web Server*, dan *layer database* memakai MySQL.

Jadi, setidaknya dalam teori, sistem berbasis *client-server* memberikan keuntungan yang banyak seperti penggunaan sumber data secara lebih efisien, penyimpanan data terpusat, serta lalu lintas di dalam jaringan menjadi lebih rendah (dibandingkan dengan sistem yang seluruhnya terpusat). Satu-satunya kelemahan utama dari sistem berbasis *client-server* yang bersifat *stand alone* adalah manajemen dan perawatan mesin-mesin *client* serta proses konfigurasi yang memakan waktu dan tenaga. Semua program di sisi *client*, misalnya *front-end* untuk suatu *database*, harus dipasang satu persatu di setiap komputer *client*, dan apabila suatu saat program *client* tersebut harus diubah atau dikembangkan, maka prosesnya harus dilakukan di setiap komputer *client* sehingga hal ini akan sangat menyulitkan dan memakan waktu. Lain halnya jika sistem tersebut berbasis *web*, karena program untuk *client* hanya menggunakan *browser web* yang tidak perlu *update* jika suatu saat sistem tersebut membutuhkan *upgrading*.

Dari sisi manajemen, sistem berbasis *web* bisa mengatasi kekurangan-kekurangan sistem tradisional. Adapun keuntungan mengembangkan sistem *client-server* berbasis *web* adalah sebagai berikut:

1. Tidak ada masalah distribusi program

Pendistribusian berlangsung secara sendirinya, karena setiap salinan dokumen (sebagai satu komponen aplikasi) di-*download* ke mesin *client* setiap saat mesin *client* membutuhkan dan meminta *update* atau salinan yang lebih baru. Tidak perlu lagi seorang *administrator* jaringan meng-*install* perangkat lunak *client* untuk setiap komputer yang ada di organisasinya, suatu pekerjaan yang melelahkan dan memboroskan waktu.

2. Efisien

Distribusi otomatis dan tidak perlunya instalasi untuk setiap *client* jelas mempermudah perawatan dan *updating* aplikasi. Perubahan-perubahan pada aplikasi bisa dikerjakan secara terpusat dan bisa langsung diterapkan tanpa perlu melakukan penyesuaian di semua *client*. Bahkan tampilan pada pengguna bisa diubah-ubah secara berkala disesuaikan dengan waktu dan situasi.

3. Fleksibel

Web browser tersedia untuk hampir semua macam *platform* mesin dan sistem operasi, baik itu mesin Windows berbasis prosesor Intel, mesin UNIX berbasis prosesor RISC, ataupun komputer Macintosh. Fleksibilitas aplikasi berbasis *web* lebih terjamin, karena tidak perlu lagi mengembangkan program-program *client* yang berbeda untuk tiap macam *platform*.

2.3 Short Message Service (SMS)

2.3.1 Pengertian SMS

SMS merupakan fasilitas standar dari *Global System for Mobile Communication* (GSM) yang dikembangkan dan distandarisasi oleh *European Telecommunication Standard Institute* (ETSI) [18]. Fasilitas ini dipakai untuk mengirim dan menerima pesan dalam bentuk teks ke dan dari sebuah ponsel. SMS pertama kali diujicobakan Desember 1992 melalui sebuah komputer ke ponsel di jaringan GSM Vodafone di Inggris. Selanjutnya teknologi SMS berkembang dan jenis aplikasi yang dapat

digunakan bertambah. Panjang pesan yang dapat dikirimkan dalam satu kali pengiriman mencapai 160 karakter atau 70 karakter jika menggunakan karakter *non latin*, seperti Arab atau Cina [15].

Pada saat mengirim SMS dari sebuah ponsel, pesan tersebut tidak langsung dikirim ke ponsel tujuan, akan tetapi terlebih dahulu dikirim ke *SMS Center (SMSC)* dengan prinsip *Store and Forward*, setelah itu dikirimkan ke ponsel yang dituju [11]. Artinya SMSC berperan sebagai penghubung antara pengirim dan penerima.

Terdapat dua *mode* dalam pengiriman dan penerimaan SMS, yaitu *mode teks* dan *mode Protocol Data Unit (PDU)*. *Mode teks* merupakan format teks asli yang dituliskan pada saat akan mengirim pesan sedangkan *mode PDU* adalah *format* pesan dalam bentuk oktet heksadesimal dan oktet semidesimal dengan panjang mencapai 160 (7 bit) atau 140 (8 bit) karakter. Di Indonesia tidak semua operator GSM ataupun ponsel mendukung *mode teks*, sehingga *mode* yang digunakan adalah *mode PDU*. Pada pengiriman pesan terdapat dua jenis *mobile*, yaitu *Mobile Originated (Ponsel Pengirim)* dan *Mobile Terminated (Ponsel Penerima)* [15].

2.3.2 Cara Kerja SMS

Dibalik kemudahan yang diberikan layanan SMS, terdapat suatu sistem yang sangat kompleks. Pesan yang telah diketik di ponsel tidak langsung begitu saja dikirim ke SMSC (*SMS Center*), namun terlebih dahulu diubah kedalam suatu bentuk yang dikenal dengan istilah PDU (*Protocol Data Unit*). PDU berisi bilangan heksadesimal yang mencerminkan bahasa *Input Output*. PDU terdiri dari beberapa *Header* atau potongan informasi. *Header* untuk mengirim SMS atau pesan ke SMSC berbeda dengan SMS yang diterima dari SMSC.

2.3.2.1 Header Pengiriman SMS

Jika sebuah ponsel akan mengirim pesan “hello” ke nomor 081376493629 melalui SMS center 081100000, maka ponsel akan mengirimkan sederetan bilangan heksa dalam bentuk PDU yang terdiri atas delapan header, yaitu :

1. Nomor SMS-Centre

Header pertama ini terbagi atas tiga *subheader*, yaitu :

- a. Jumlah pasangan Heksadesimal SMSC dalam bilangan heksa.
- b. *National/International Code*
Untuk *National Code* atau kode nasional , kode *subheader*nya adalah 81. Sedangkan untuk *International Code* atau kode internasional kode *subheader*nya adalah 91.
- c. Nomor SMSC itu sendiri disusun dalam pasangan heksa yang dibalik-balik. Jika tertinggal satu angka heksa yang tidak memiliki pasangan, angka tersebut akan dipasangkan dengan huruf F di depannya.

Misalnya untuk nomor SMSC Telkomsel dapat diubah ke bentuk PDU dengan cara sebagai berikut :

- a. SMSC yaitu 081100000 menggunakan kode nasional sehingga *subheader*nya adalah 81.
- b. Nomor SMSC itu sendiri disusun terbalik-balik sehingga menjadi 80110000F0

Jumlah pasangan header adalah 6 pasang (1 pasang *National Code* dan 1 5 pasang untuk SMSC itu sendiri, sehingga header untuk SMSC menjadi 068180110000F0

2. Tipe SMS

Untuk proses pengiriman, maka tipe SMS adalah 1. Jadi bilangan heksanya adalah 01.

3. Nomor Referensi SMS

Nomor referensi ini dibiarkan saja bernilai 0, jadi bilangan heksanya adalah 00. Nanti nomor referensi ini akan diberikan secara otomatis oleh ponsel.

4. Nomor Tujuan

Cara penulisan nomor tujuan hampir sama seperti cara penulisan untuk SMS-Centre. Misalnya nomor tujuannya adalah 081376493629, maka nomor ini akan dikonversikan ke dalam bentuk PDU dengan cara sebagai berikut :

- a. Jumlah nomor tujuan adalah 12 angka sehingga bilangannya adalah 0C.
- b. Nomor tujuan 081376493629 menggunakan kode nasional sehingga subheadernya adalah 81.
- c. Nomor tujuan tersebut lalu disusun dibalik-balik sehingga menjadi 803167946392

Sehingga nomor tujuan dalam bentuk PDU adalah 0C81803167946392

5. Bentuk SMS

Jika SMS dikirim sebagai bentuk SMS biasa, maka kodenya adalah 00. Namun jika SMS dikirim sebagai telex, maka kodenya adalah 01. Sedangkan jika dikirim sebagai fax, maka kodenya adalah 02.

6. Skema Enkoding

Kebanyakan ponsel yang ada dipasaran sekarang menggunakan skema 7 bit sehingga kode skema *encoding*nya adalah 00.

7. Jangka Waktu Pengiriman SMS

Jika bagian ini dilewatkan, berarti kita tidak membatasi waktu berlakunya SMS. Sedangkan jika kita isi dengan suatu bilangan integer yang kemudian diubah ke pasangan bilangan heksa tertentu, bilangan yang telah kita berikan tersebut akan mewakili jumlah waktu validitas SMS tersebut.

Adapun Rumus untuk menghitung jangka waktu validitas SMS adalah sebagai berikut :

Tabel 2.1 Tabel Jangka Waktu Validitas SMS

Integer(INT)	Jangka waktu validitas SMS
0 -143	$(Int + 1) \times 5$ menit (Berarti 5 menit s/d 12 jam)
144 -167	12 Jam + $((Int - 143) \times 30)$ menit)
168 - 196	$(Int-166) \times 1$ hari
197 - 255	$(Int-192) \times 1$ minggu

8. Isi SMS

Jika hendak mengirim kata “hello”, maka kata ini harus diubah ke dalam bentuk PDU dengan langkah sebagai berikut:

- a. Mengubahnya menjadi kode biner 7 bit

h 110 1000
e 110 0101
l 110 1100
l 110 1100
o 110 1111

- b. Mengubah 7 bit ke 8 bit yang diwakili oleh pasangan bilangan heksa

 E 8
h 1 110 1000
 3 2
 1
e 00 110 010
 9 B
l 100 1 1011 00
 F D
l 1111 1101 100
 0 6
o 0000 0 101 1111

Karena total 7 bit x 5 huruf = 35 bit, sedangkan yang diperlukan adalah 8 bit x 5 huruf = 40 bit, maka diperlukan 5 bit tambahan yang diisi dengan bilangan bit 0. Setiap 8 bit mewakili pasangan heksa sehingga tiap 4 bit mewakili sebuah bilangan heksa. Dengan demikian kata “hello” telah diubah menjadi 05E8329BFD06. angka “05” pada awal header tersebut berdasarkan jumlah karakter isi pesan yang berarti 5 karakter.

Setelah diperoleh delapan header untuk mengirim SMS, lalu kedelapan header tersebut digabung menjadi sebuah PDU yang lengkap. Sehingga untuk mengirim pesan “hello” ke nomor 081376493629 melalui SMSC 081100000, maka bentuk PDU lengkap untuk proses tersebut adalah :

068180110000F001000C81803167946392000005E8329BFD06

2.3.2.2 Header Pembacaan SMS

Setelah Data berupa PDU dikirimkan dari ponsel ke SMSC, maka SMSC akan meneruskan SMS tersebut ke nomor tujuan, Namun sebelumnya SMSC akan menambahkan beberapa informasi ke dalam header tersebut. Data dalam bentuk PDU yang diterima oleh ponsel tujuan juga terdiri dari delapan header, yaitu :

1. Nomor SMS-Center
2. Tipe SMS, untuk SMS-terima kodenya adalah 04
3. Nomor ponsel pengirim
4. Bentuk SMS
5. Skema *encoding*
6. Tanggal dan waktu SMS sampai di SMSC
7. Batas waktu validitas SMS
8. Isi SMS

Jika ponsel menerima data yang diterima adalah dalam bentuk PDU, misalnya 068180110000F0040C818031563625840000704022512380005E8329BFD06, maka ponsel akan menguraikan PDU tersebut sebagai berikut:

1. Nomor SMS-Center pengirim adalah 081100000.
2. SMS tersebut adalah SMS-terima.
3. Nomor Ponsel pengirim adalah 08126037983.
4. Pesan diterima dalam bentuk SMS, bukan dalam bentuk fax.
5. SMS tersebut memiliki skema *encoding* 7 bit.
6. SMS tersebut sampai di SMS-Center pada tanggal 22 April 2011 pada pukul 15:32:08.
7. SMS tersebut tidak memiliki batas waktu valid.

8. SMS tersebut isinya adalah “hello”

2.3.3 SMS Gateway

2.3.3.1 Pengertian SMS Gateway

Istilah *gateway* dapat diartikan sebagai pintu gerbang. Namun pada dunia komputer, *gateway* dapat diartikan sebagai jembatan penghubung antara satu sistem dengan sistem yang lain, sehingga dapat terjadi pertukaran data antar sistem tersebut. Dengan demikian, SMS *gateway* dapat diartikan sebagai penghubung untuk lalu lintas data-data SMS.

Pada awalnya, SMS *gateway* dibutuhkan untuk menjembatani antar SMSC. Hal ini dikarenakan SMSC yang dibangun oleh perusahaan yang berbeda memiliki protokol komunikasi sendiri, dan protokol tersebut bersifat pribadi. SMS *gateway* ini kemudian ditempatkan di antara kedua SMSC yang berbeda protokol tersebut, yang akan menerjemahkan data dari protokol SMSC satu ke protokol SMSC lainnya yang dituju.

Namun seiring perkembangan teknologi komputer dan perkembangan teknologi komunikasi, masyarakat lebih mengartikan SMS *gateway* sebagai suatu jembatan komunikasi yang menghubungkan perangkat komunikasi (dalam hal ini ponsel) dengan perangkat komputer.

SMS *gateway* kemudian lebih mengarah kepada sebuah program yang mengkomunikasikan sistem operasi komputer dengan perangkat komunikasi yang terpasang untuk mengirim atau menerima SMS.

2.3.3.2 Gammu Software

Komunikasi yang terjadi antara komputer dan handphone tidak dapat terjadi begitu saja. Diperlukan sebuah *software* pendukung yang menjadi perantara ataupun

penghubung agar komunikasi tersebut dapat terjadi. Beberapa *software* yang tersedia di pasaran bersifat *open source*, seperti Kannel, GNOKII, dan Gammu. Dalam hal ini penulis akan membahas mengenai Gammu versi 1.21.90-Windows yang dipergunakan sebagai *software* penghubung [16].

Gammu merupakan *software* yang paling banyak dipakai karena kelebihan yang dimilikinya. Kelebihan Gammu dari tool sms gateway lainnya adalah :

1. Gammu bisa di jalankan di Windows maupun Linux
2. Banyak device yang kompatibel oleh gammu
3. Gammu menggunakan database MySql
4. Baik kabel data USB maupun SERIAL, semuanya kompatibel di Gammu.

Selain kelebihan di atas, Gammu telah didukung banyak jenis dari berbagai *merk handphone*, seperti yang tertera pada tabel di bawah [21]:

Tabel 2.2 Merk-Merk Handphone yang Didukung Gammu

Merk Handphone	Merk Handphone
Alcatel	Nokia
Apple	Onda
BenQ-Siemens	Openmoko
CECT	Option
Daxian	PalmOne
Elson	Philips
Emgeton	Sagem
Falcom	Samsung
Fly	Sandshine
Foston	Sanyo
Gionee	SciPhone
HTC	Sharp
Huawei	Siemens
Hughes	Sierra Wireless
Jinpeng	Simcom
Lenovo	Sony Ericson
LG	Teltonika
Matsunichi	Toshiba
Mitsubishi	Vodafone
Motorola	dll.

2.4 Web Server

Web server adalah komputer yang mengirimkan halaman-halaman *web* [10]. Setiap *web server* mempunyai alamat IP dan nama. Komputer dapat dijadikan sebagai *Web server* dengan diinstal software server dan menghubungkannya dengan internet. Terdapat banyak software server, baik yang gratis, seperti yang dibuat oleh NCSA dan Apache maupun yang komersial, seperti yang dibuat oleh Microsoft dan Netscape.

2.5 Database

Database merupakan kumpulan dari *item* data yang saling berhubungan satu dengan yang lainnya yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, tersimpan di *hardware* komputer dan dengan *software* untuk melakukan manipulasi untuk kegunaan tertentu. Dalam pembuatan aplikasi, pangkalan data mempunyai peranan yang sangat penting karena :

1. Salah satu komponen penting dalam sistem informasi
2. Menentukan kualitas informasi
3. Mengurangi duplikasi data (*data redudancy*)
4. Hubungan data dapat ditingkatkan (*data relatability*)
5. Mengurangi pemborosan tempat simpanan luar

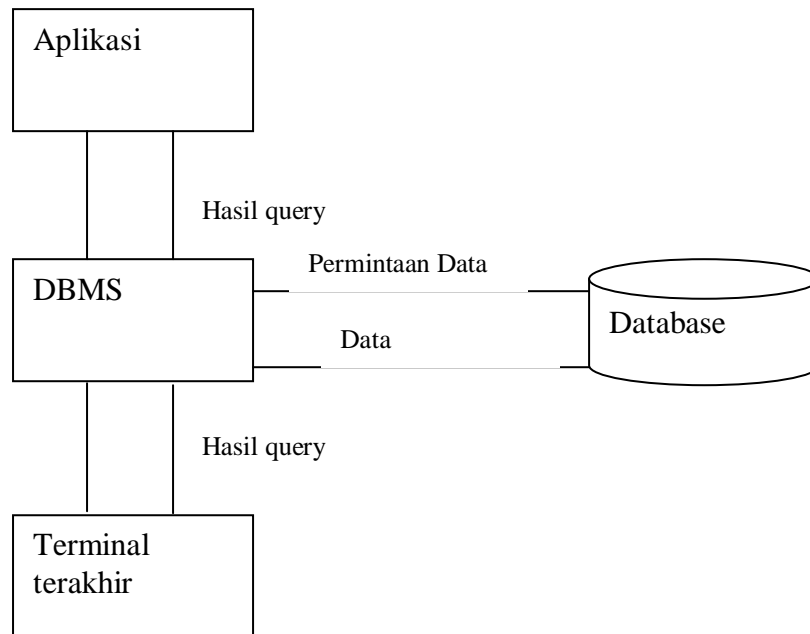
Tujuan utama dibangunnya *database* adalah untuk kemudahan dan kecepatan dalam pengambilan kembali data/arsip.

2.5.1 DBMS (*Data Base Management System*)

Database Management Sistem (DBMS) secara umum diartikan sebagai suatu program komputer yang digunakan untuk memasukkan, mengubah, menghapus, memanipulasi dan memperoleh data informasi dengan praktis dan efisien. Semua operasi input dan output yang berhubungan dengan database harus menggunakan DBMS. Bila pemakai

akan mengakses pangkalan data, DBMS menyediakan *interface* antara pemakai dengan *database*. Hubungan pemakai dengan *database* dapat dilakukan dengan dua cara:

1. Secara interaktif menggunakan bahasa pertanyaan (*query language*).
2. Dengan menggunakan program aplikasi



Gambar 2.2 Hubungan Pemakai dengan Database

Dibandingkan dengan sistem yang berbasis kertas, DBMS memiliki 4 keunggulan :

1. Kepraktisan

Sistem yang berbasis kertas akan menggunakan kertas untuk menyimpan informasi, sedangkan DBMS menggunakan penyimpan sekunder yang berukuran kecil tetapi padat informasi.

2. Kecepatan

Mesin dapat mengambil atau mengubah data jauh lebih cepat daripada manusia.

3. Mengurangi kejenuhan

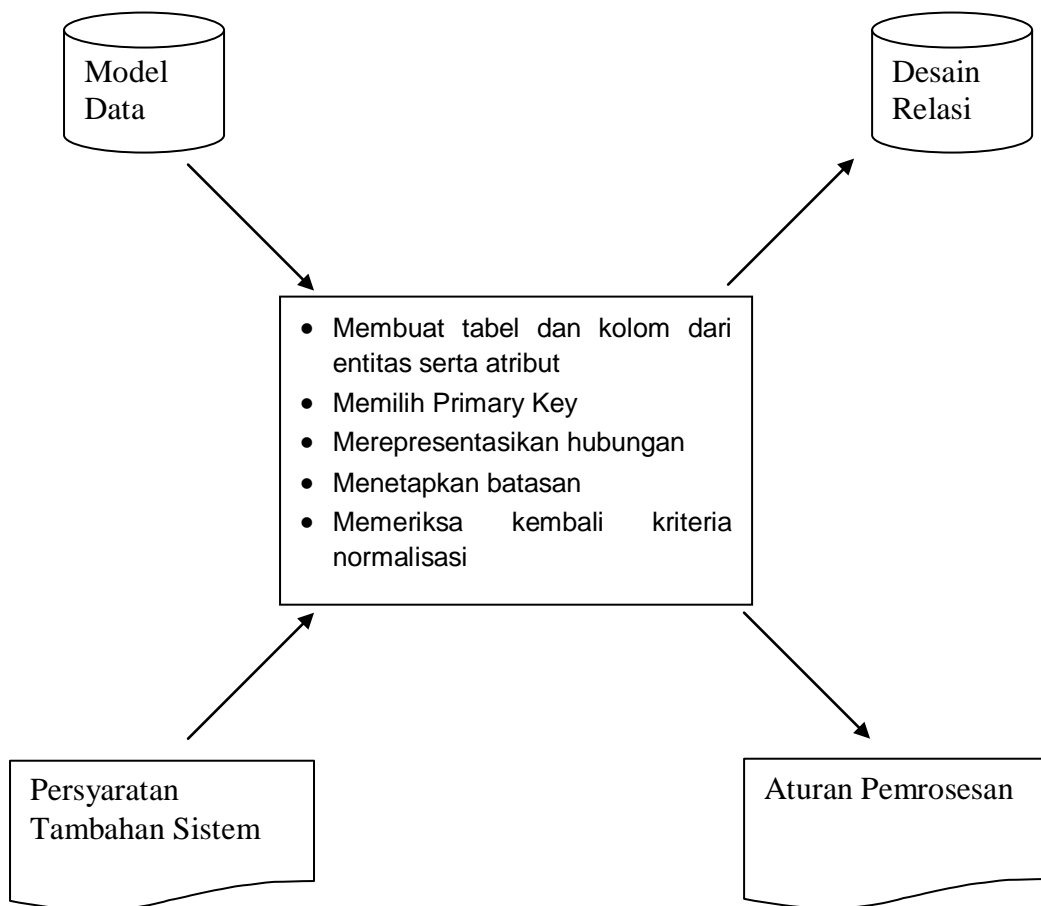
Orang cenderung menjadi bosan kalau melakukan tindakan-tindakan berulang menggunakan tangan.

4. Kekinian

Informasi yang tersedia pada DBMS akan bersifat mutakhir dan akurat setiap saat.

2.5.2 Proses Desain Pangkalan Data

Gambar di bawah memperlihatkan komponen-komponen dalam proses desain pangkalan data.



Gambar 2.3 Komponen-Komponen Desain Pangkalan Data

Adapun yang menjadi tahapan dalam membangun sebuah desain pangkalan data adalah sebagai berikut :

1. Membuat tabel dan kolom dari entitas dan atribut

Untuk mentransformasikan sebuah model entity-relationship ke dalam desain database relasional, masing-masing entitas direpresentasikan sebagai sebuah tabel dan semua atribut entitas akan menjadi kolom tabel tersebut.

2. Memilih *Primary key* (Kunci Utama)

Pemilihan *primary key* merupakan hal yang sangat penting dan hampir semua produk DBMS (*Database Management System*) menyusun indeks pada kolom *primary key* sehingga nilai *key* dapat digunakan untuk mengatur penyimpanan fisik dan memfasilitasi pencarian serta penyeleksian dengan menggunakan *primary key*. Salah satu ciri *primary key* adalah unik sehingga bisa digunakan sebagai pengenal atau *identifier*.

3. Merepresentasikan hubungan

Untuk model relasional, semua hubungan dinyatakan dengan menempatkan *primary key* dari suatu tabel yang satu dengan tabel yang ke dua.

4. Menetapkan batasan

Penetapan batasan dimaksudkan untuk tujuan yang lebih luas. Batasan didefinisikan sebagai suatu aturan yang mengatur nilai-nilai statis atribut yang cukup tepat sehingga dapat memastikan apakah nilai itu benar atau tidak.

5. Memeriksa kembali kriteria normalisasi

Normalisasi bertujuan untuk menghilangkan anomali modifikasi.

2.6 Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah sebuah bahasa grafis yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak [1]. UML merupakan dasar fundamental dari teknik analisis berorientasi objek, berbentuk diagram – diagram yang digunakan untuk menampilkan konstruksi dari sistem berorientasi objek, seperti cetak biru (*blue print*) suatu pembangunan gedung yang menggambarkan konstruksi bangunan tersebut [6]. UML juga telah menjadi sebuah standar pemodelan berorientasi objek. Notasi grafisnya memberi kemudahan pemahaman selama proses pengembangan [8]. UML mendefinisikan diagram – diagram berikut:

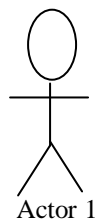
2.6.1 Use Case Diagram

Use case diagram menggambarkan secara grafis perilaku perangkat lunak. Diagram ini memberikan gambaran menurut perspektif pengguna perangkat lunak. Sebuah *use*

case diagram mengandung *actor*, *use case* dan interaksi antara *actor* dengan *use case* [13].

2.6.1.1 Actor

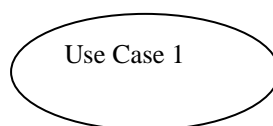
Actor merupakan segala sesuatu yang perlu berinteraksi dengan sistem untuk pertukaran informasi. *Actor* memberikan suatu gambaran jelas tentang apa yang harus dilakukan perangkat lunak. *Actor* dinotasikan seperti pada gambar berikut:



Gambar 2.4 Actor

2.6.1.2 Use case

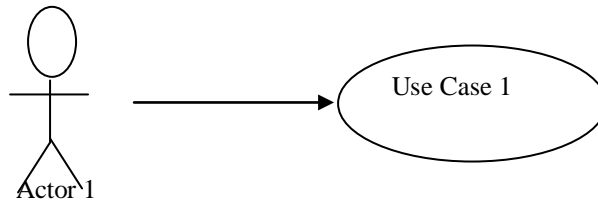
Use case merupakan hasil penyusunan kembali lingkup fungsionalitas sistem menjadi banyak pernyataan fungsionalitas sistem yang lebih kecil. Sebuah *use case* merepresentasikan satu tujuan tunggal dari sistem dan menggambarkan satu rangkaian kegiatan dan interaksi pengguna untuk mencapai tujuan [17]. *Use case* menggambarkan fungsi – fungsi sistem dari sudut pandang pengguna eksternal. Diagram ini juga dapat diartikan sebagai urutan transaksi berkaitan yang dilakukan satu *actor* dengan perangkat lunak. *Use case* dinotasikan seperti pada gambar berikut:



Gambar 2.5 Use case

2.6.1.3 Interaksi Actor dengan Use case

Interaksi *Actor* dengan *Use case* dinotasikan seperti pada gambar berikut:

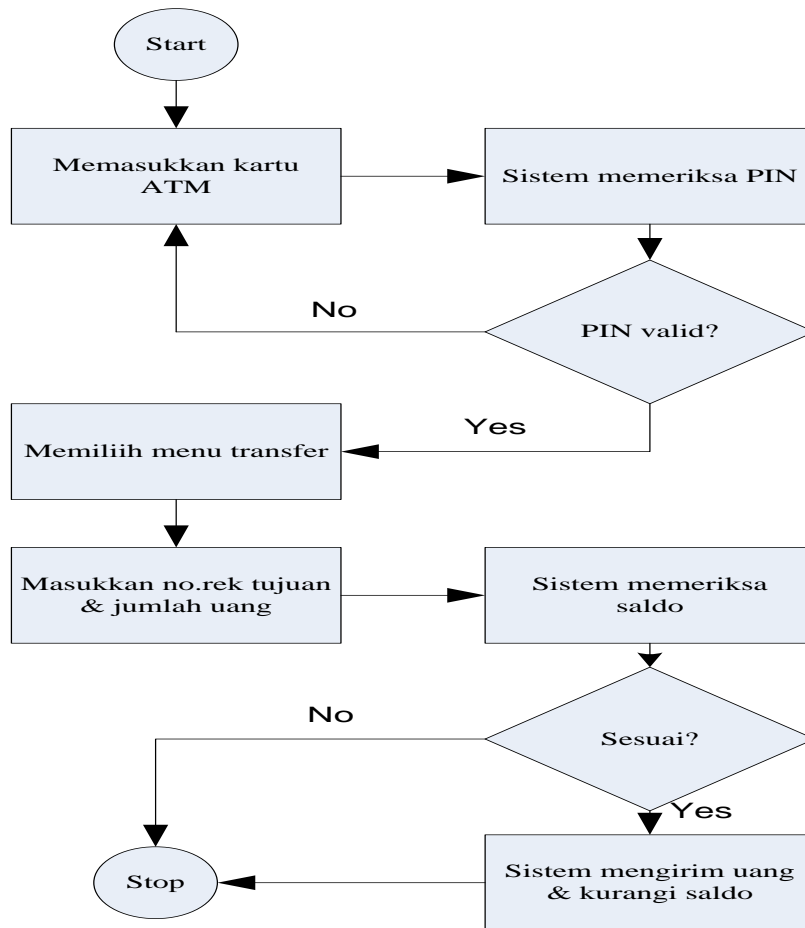


Gambar 2.6 Use case Diagram

2.6.2 Activity Diagram

Activity diagram memodelkan alur kerja (*workflow*) sebuah proses bisnis dan urutan aktifitas dalam suatu proses. Diagram ini sangat mirip dengan sebuah *flowchart* karena kita dapat memodelkan sebuah alur kerja dari satu aktifitas ke aktifitas lainnya atau dari satu aktifitas ke dalam keadaan sesaat (*state*) [13]. *Activity* diagram menggambarkan aliran aktifitas dari sistem yang sedang dirancang, bagaimanapun masing – masing aliran berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. Diagram ini juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Contoh *activity* diagram diperlihatkan pada gambar berikut:

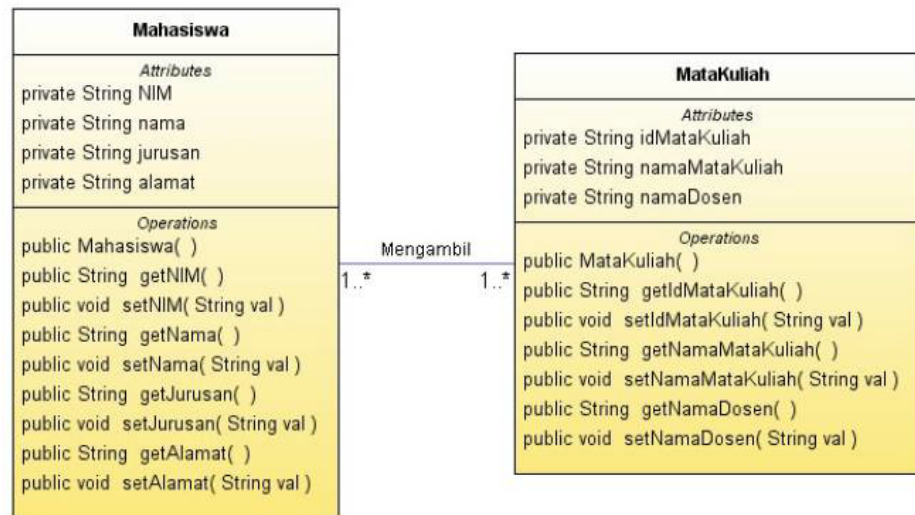


Gambar 2.7 Activity Diagram

2.6.3 Class Diagram

Class diagram merupakan struktur kelas – kelas dari suatu sistem yang memperlihatkan hubungan antar kelas dan penjelasan tiap – tiap kelas. Pada diagram ini terdapat nama kelas, atribut dan operasi kelas tersebut. Selama proses analisis, *class* diagram memperlihatkan aturan – aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Selama tahap desain, diagram ini berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat [13].

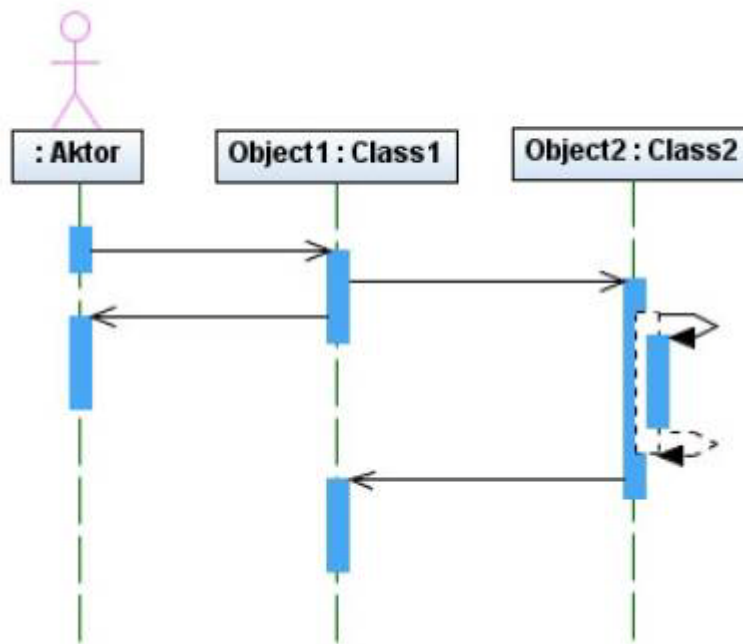
Contoh *class* diagram diperlihatkan pada gambar berikut:



Gambar 2.8 Class Diagram

2.6.4 Sequence Diagram

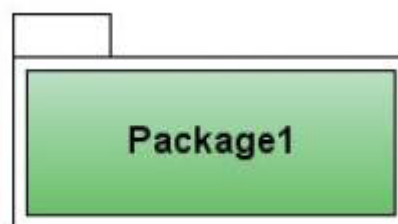
Sequence diagram menjelaskan interaksi objek yang disusun dalam suatu urutan waktu. Diagram ini memperlihatkan tahap demi tahap apa yang harus terjadi untuk menghasilkan sesuatu di dalam *use-case*. *Sequence* diagram secara khusus berinteraksi dengan *use-case*. Masing-masing *sequence diagram* menggambarkan aliran pada suatu *use case*. *Sequence diagram* dapat dibaca dengan melihat pada objek-objek dan pesan-pesan (*message*). Objek-objek yang berperan dalam aliran diperlihatkan pada kotak empat persegi panjang yang melintas pada bagian atas diagram. Setiap objek memiliki garis hidup (*lifeline*), yang digambarkan sebagai garis vertikal di bawah nama suatu objek. Notasi *sequence* diagram digambarkan sebagai berikut:



Gambar 2.9 Sequence Diagram

2.6.5 Package Diagram

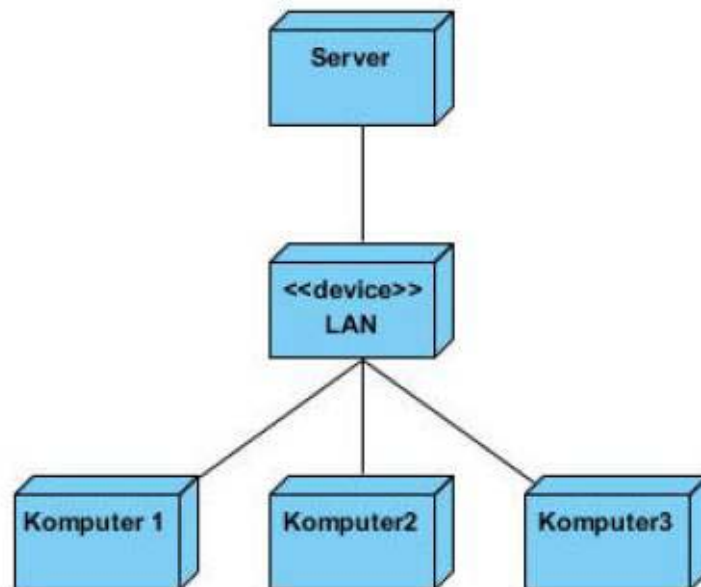
Sebuah *package* adalah sebuah bentuk pengelompokan yang memungkinkan pembangun untuk mengambil setiap bentuk di UML dan mengelompokkan elemen – elemennya dalam tingkatan unit yang lebih tinggi [2]. Diagram ini merupakan mekanisme pengelompokan yang digunakan untuk menandakan pengelompokan elemen – elemen model. Sebuah *package* dapat mengandung beberapa paket lain di dalamnya. Diagram ini digunakan untuk memudahkan mengorganisasi elemen – elemen model. Notasi *package* diagram digambarkan sebagai berikut:



Gambar 2.10 Package Diagram

2.6.6 Deployment Diagram

Deployment diagram menunjukkan susunan fisik sebuah sistem, menunjukkan bagian perangkat lunak mana yang berjalan pada perangkat keras yang mana [2]. Diagram ini adalah diagram dengan tipe implementasi yang digunakan untuk secara grafis menggambarkan arsitektur fisik dari perangkat lunak sistem. Diagram ini dapat digunakan untuk menunjukkan ketergantungan di antara komponen – komponen penyusun sistem. *Deployment* diagram menggambarkan bagaimana komponen dibangun dalam infrastruktur sistem, di mana suatu komponen (pada mesin, server atau perangkat keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server. Sebuah *node* adalah *server*, workstation atau perangkat keras lain yang digunakan untuk membangun komponen dalam lingkungan sebenarnya. Contoh *deployment* diagram diperlihatkan pada gambar berikut:



Gambar 2.11 Deployment Diagram

2.7 Analisis Persyaratan dengan UML

Analisis persyaratan meliputi usaha untuk mengetahui apa kemampuan sebuah sistem yang diinginkan pengguna dan pelanggan dari sebuah pembuat perangkat lunak [2].

Analisis ini dilakukan untuk mendapatkan informasi atau persyaratan cukup untuk mempersiapkan model yang menggambarkan apa yang diperlukan dari perspektif pengguna. Diagram yang digunakan dalam analisis persyaratan yaitu:

1. *Use case diagram* yang digunakan untuk menunjukkan fungsionalitas suatu sistem dan bagaimana sistem berinteraksi dengan dunia luar.
2. *Activity diagram* yang menunjukkan alur kerja (*work flow*) sebuah proses bisnis dan urutan aktivitas dalam suatu proses.
3. *Class diagram* yang membantu dalam visualisasi struktur sistem yang mendeskripsikan jenis-jenis objek dalam suatu sistem dan hubungan yang terdapat diantara objek tersebut.
4. *Package diagram* yang digunakan untuk mengelompokkan elemen-elemen model atau kelas.

2.8 Desain dengan UML

Saat membuat desain adalah saat untuk berpikir secara teknis dalam menggambarkan diagram – diagram UML. Diagram yang digunakan dalam mendesain sistem yaitu:

1. *Class diagram* dalam sudut pandang perangkat lunak, untuk menunjukkan *class* yang terdapat di dalam perangkat lunak dan bagaimana mereka saling berhubungan.
2. *Sequence diagram* untuk menjelaskan interaksi objek yang disusun dalam suatu urutan waktu.
3. *Package diagram* yang digunakan untuk mengelompokkan elemen-elemen model atau kelas.
4. *Deployment diagram* yang menunjukkan arsitektur fisik sebuah sistem.