BAB II

LANDASAN TEORI

2.1 Short Message Service (SMS)

2.1.1 Sejarah *SMS*

Short Message Service (SMS) merupakan sebuah layanan yang banyak diaplikasikan pada sistem komunikasi tanpa kabel, dikembangkan dan distandarisasi oleh suatu badan yang bernama *European Telecomunication Standards Institute* (ETSI) sebagai bagian dari pengembangan GSM fase 2, yang terdapat pada dokumentasi GSM 03.40 dan GSM 03.38. Fitur SMS ini memungkinkan perangkat Stasiun Seluler Digital (*Digital Cellular Terminal*, seperti ponsel) untuk dapat mengirim pesan dalam bentuk *alphanumeric* dan menerima pesan-pesan teks dengan panjang sampai dengan 160 karakter melalui jaringan GSM. (ETSI, 1996)

Isu SMS pertama kali muncul dibelahan Eropa pada sekitar tahun 1991 bersama sebuah teknologi komunikasi *wireless* yang saat ini cukup banyak pengunanya, yaitu *Global System for Mobile comunication* (GSM). Dipercaya bahwa pesan pertama yang dikirimkan menggunakan SMS dilakukan pada bulan Desember 1992, dikirimkan dari sebuah *Personal Computer* (PC) ke telepon *mobile* (bergerak) dalam jaringan GSM milik Vodafone Inggris. Perkembangannya kemudian merambah ke benua Amerika, dipelopori oleh beberapa operator komunikasi bergerak berbasis digital seperti *BellSouth Mobility*, *PrimeCo*, *Nextel*, dan beberapa operator lain.

Layanan SMS merupakan layanan yang bersifat *nonreal time* dimana sebuah *short message* dapat di-*submit* ke suatu tujuan, tidak peduli apakah tujuan tersebut aktif atau tidak. SMS merupakan sebuah sistem pengiriman data dalam paket yang bersifat *out-of-band* dengan *bandwidth* kecil, dengan karakteristik ini pengiriman suatu *burst data* yang pendek dapat dilakukan dengan efisiensi yang sangat tinggi.

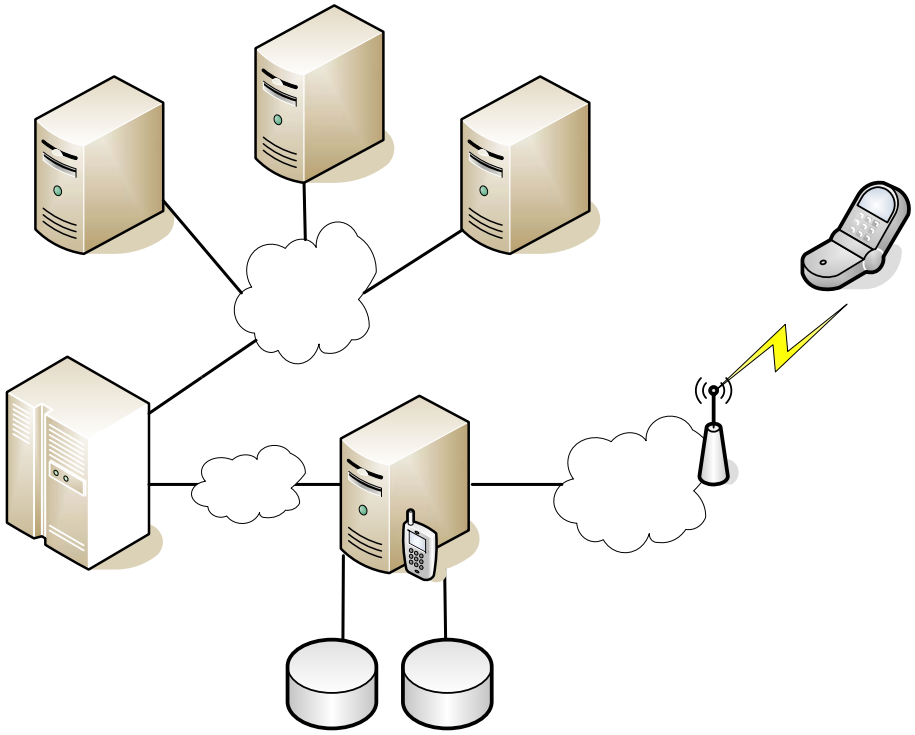
2.1.2 Mekanisme Distribusi Pesan SMS

Terdapat empat macam mekanisme distribusi pesan SMS oleh aplikasi SMS, yaitu:

a. *Pull*, yaitu pesan yang dikirimkan ke pengguna berdasarkan permintaan pengguna.
b. *Push – Event based*, yaitu pesan yang diaktivasi oleh aplikasi berdasarkan kejadian yang berlangsung
c. *Push - Schedule*, yaitu pesan yang diaktivasi oleh aplikasi berdasarkan waktu yang telah terjadwal.
d. *Push – Personal Profile*, yaitu pesan yang diaktivasi oleh aplikasi berdasarkan *profile* dan *preference* dari pengguna

2.1.3 Arsitektur dan Elemen Layanan Jaringan SMS

Untuk implementasi layanan SMS, operator menyediakan apa yang disebut SMS Center (SMSC). Secara fisik SMSC dapat berwujud sebuah PC biasa yang mempunyai interkonektivitas dengan jaringan GMS.



Gambar 2.1 Arsitektur dasar jaringan SMS

Pada gambar 2.1 menjelaskan tentang cara kerja dari SMSC (SMS Center) yang mempunyai interkonektivitas dengan jaringan GSM.

2.1.4 Elemen dasar jaringan SMS

1. *Short Messaging Entities* (SME), suatu piranti yang dapat menerima dan mengirim pesan pendek.
2. *Short Message Service Center* (SMSC), kombinasi perangkat lunak dan perangkat keras yang brtanggung jawab memperkuat, menyimpan dan meneruskan pesan pendek antar SME dan piranti bergerak (*mobile phone).*
3. *Signaling System 7* (SS7), protokol *signalling* yang umum di gunakan dalam jeringan telepon seluler.

4. *Base Station System* (BSS), berfungsi mengendalikan satu atau lebih BTS dan bertanggung jawab dalam pemberian sumber data dan transmisi sinyal radio elektromagnetis antara MSC dan *mobile phone.*
5. *Home Location Register* (HLR), basis data yang digunakan untuk penyimpanan permanen, pengelolaan dan profil layanan.
6. *Visitor Location Register* (VLR), basis data yang berisi informasi temporal mengenai pelanggan yang berasal dari suatu HLR yang roaming ke HLR lainnya.

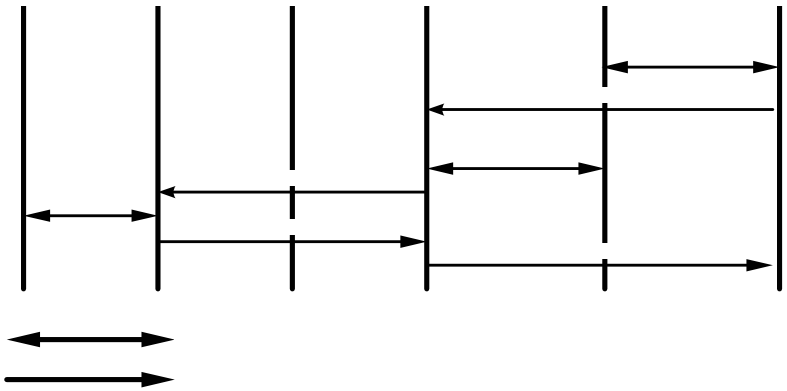
2.1.5 Elemen Layanan

SMS terdiri dari beberapa elemen layanan yang relevan terhadap penerimaan dan pengiriman pesan pendek yaitu:

1. *Message Expiration*, SMSC akan menyimpan dan mencoba mengirimkan kembali pesan yang mengalami kegagalan sampai pengiriman tersebut berhasil.
2. *Priority*, Untuk memberi tanda pesan-pesan yang penting dan membedakannya dari pesan biasa.

Sistem SMS memiliki dua layanan dasar *point-to-point* bagi pelanggan yaitu:

a. *Mobile-Oriented* (MO) *Short Message*, dikirimkan dari *mobile phone* yang MO-*Capable* ke SMSC dan dapat ditujukan ke *mobile phone* lainnya. Pada layanan ini selalu ada laporan yang dikirimkan ke *mobile phone*, baik yang mengkonfirmasikan pengiriman oesan pendek ke SMSC ataupun mengkonfirmasikan kegagalan pengiriman dan mengidentifikasikan penyebabnya.



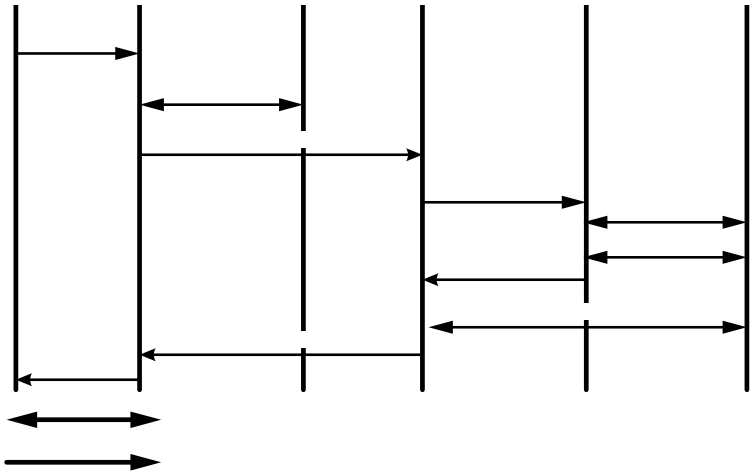
Gambar 2.2 Skenario pengiriman MO-SM

Pada gambar 2.2 diatas menjelaskan skenario pengiriman MO-SM yang dapat diuraikan lebih rinci lagi pada keterangan dibawah ini. yaitu:

1. MS mengirimkan SM ke MSC
2. MSC menginterogasi VLR utuk membuktikan bahwa pengiriman pesan tersebut tidak melanggar permintaan layanan pembatasan yang telah ditetapkan.
3. MSC mengirimkan pesan pendek ke SMSC dengan menggunakan operasi *Forward Short Message.*
4. SMSC mengirimkan pesan pendek ke SME
5. SMSC memberitahu MSC mengenai keberhasilan operasi *Forward Short Message.*
6. MSC mengembalikan hasil dari operasi MO-SM ke MS.

b. *Mobile Terminated* (MT) *Short Message*, dikirimkan dari SMSC ke *mobile phone* dan dapat sampai ke SMSC dari *mobile phone* lainnnya melalui MO-SM, pada layanan ini juga terdapat laporan yang diberikan kepada SMSC

yang isinya bisa berupa konfirmasi pengiriman pesn pendek ke *mobile phone* maupun informasi kegagalan pengiriman pesan.



Gambar 2.3 Skenario pengiriman MT-SM

Pada gambar 2.3 diatas menjelaskan skenario pengiriman MT-SM yang dapat diuraikan lebih rinci lagi pada keterangan dibawah ini. yaitu:

1. Pesan pendek dikirimkan dari SME ke SMSC.
2. Setelah menyelesaikan pengolahan internalnya, SMSC menginterogasi HLR dan menerima informasi *routing* untuk pelanggan *mobile*.
3. SMSC mengirimkan pesan pendek ke MSC dengan menggunakan operasi *Forward Short Message*..
4. MSC mengambil informasi pelanggan dari VLR. Operasi ini dapat melibatkan prosedur autentifikasi.
5. MSC mengirimkan pesan pendek ke MS.
6. MSC mengembalikan hasil dari operasi *Forward Short Message* ke SMSC.
7. Jika diminta oleh SME, SMSC akan mengembalikan laporan status yang mengindikaikan pengiriman pesan pendek.

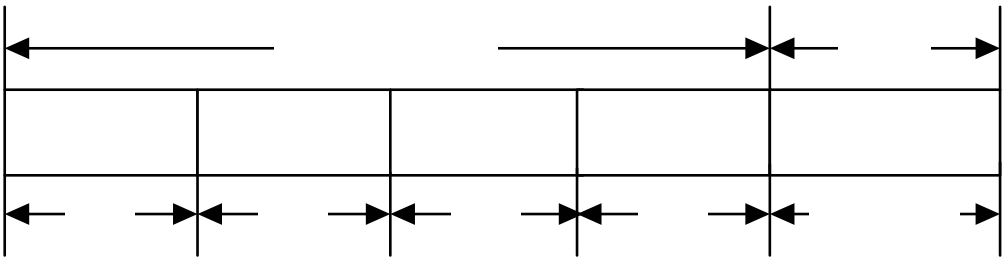
2.1.6 *Protocol Data Unit* (PDU)

Satuan paket data yang dipertukarkan pada lapisan aplikasi dalam protokol SMS Manager disebut *Protocol Data Unit* (PDU). Dalam protokol tersebut terdapat beberapa macam format PDU dimana penggunaan masing-masing PDU terebut harus sesuai dengan fungsinya, sebagai contoh untuk mengirim sebuah pesan, harus digunakan PDU dengan format ***submit_sm, deliver_sm,*** atau ***data_sm.***

Umumnya cara pengiriman paket data dari satu titik ke titik lain, slah satu titik harus bertindak sebagai server dan titik lainnya sebagai *client*, Inisiatif dan pembentukan sebuah *session* (jenis permintaan atau perintah) dilakukan oleh client. Jenis session yang dipilih sepenuhnya diserahkan kepada *client* (otorisasi diterima atau tidak tetap dipegang oleh server), terdapat tiga buah *session* yang dapat dipilih yaitu:

1. *Receiver* (RX), bila *client* ingin dapat menerima paket data.
2. *Transmitter* (TX), bila *client* ingin dapat mengirimkan paket data.
3. *Transceiver* (TRX), bila *client* ingin dapat mengirim dan menerima paket data.

Secara umum format PDU dalam protokol SMS Manager adalah sebagai berikut:



Gambar 2.4 Format umum PDU

Berikut merupakan contoh Strutuktur yang dikirim PDU

```
07 91 2658050000F0 11 00 0C 91 265836164900 00 00 FF
04 C830FB0D
```

Tabel 2.1 Struktur umum PDU

Oktet / Digit Hexa	Keterangan
07	Panjang atau jumlah pasangan digit dari nomor SMSC (service number) yang digunakan, dalam hal ini adalah 7 pasangan (14 digit berikutnya)
91	Jenis nomor SMSC. Angka 91 menandakan format nomor internasional (misal +6281xxx). Untuk 081xxx menggunakan angka 81
2658050000F0	Nomor SMSC yang digunakan. Karena jumlah digit nomor SMS adalah ganjil, maka digit paling belakang dipasangkan dengan huruf F. Kalau diterjemahkan, nomor SMSC yang digunakan adalah +62855000000 (IM3)
11	Oktet pertama untuk PDU SMS untuk dikirim
00	TP-Message-Reference. Diisi "00" agar diisi otomatis oleh handphone.
0C	Panjang digit dari nomor penerima (0C hex = 12 desimal)

Command Length

4 octets

91	Jenis nomor penerima (sama dengan jenis nomor SMSC)
265836164900	Nomor penerima SMS, yang jika diterjemahkan adalah +628563619400
00	Pengenal protokol, dalam hal ini adalah 0
00	Skema pengkodean SMS, juga bernilai 0.
FF	Validitas waktu. FF berarti maksimum
04	Panjang dari pesan SMS, dalam hal ini adalah 4 huruf (dalam mode 7 bit)
C830FB0D	Pesan SMS dalam mode 7 bit. Jika diterjemahkan kedalam 8 bit, lalu dirubah ke ASCII, maka didapat pesan 'Halo'

2.2 *Global System for Mobile communications* (GSM)

2.2.1 Sejarah GSM

Komunikasi bergerak (*mobile communication*) mulai dirasakan perlu sejak orang semakin sibuk pergi ke sana kemari dan memerlukan alat telekomunikasi yang siap dipakai sewaktu-waktu di mana saja ia berada. Kebutuhan ini ternyata tidak dibiarkan begitu saja oleh para *engineer* telekomunikasi. Mereka telah memikirkan standardisasi untuk komunikasi bergerak ini, salah satunya adalah GSM (*Global System for Mobile communications*)

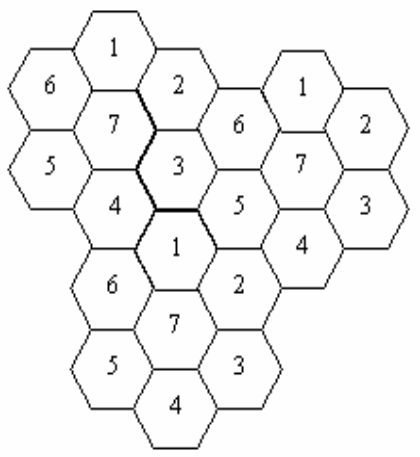
Alokasi spektrum frekuensi untuk GSM awalnya dilakukan pada tahun 1979. Spektrum ini terdiri atas dua buah *sub-band* masing-masing sebesar 25MHz, antara 890MHz - 915MHz dan 935MHz - 960MHz. Sebuah *sub-band*

dialokasikan untuk frekuensi *uplink* dan *sub-band* yang lain sebagai frekuensi *downlink.*

Kedua *sub-band* tersebut dibagi lagi menjadi kanal-kanal, sebuah kanal pada satu *sub-band* memiliki pasangan dengan sebuah kanal pada *sub-band* yang lain. Tiap *sub-band* dibagi menjadi 124 kanal, yang kemudian masing-masing diberi nomor yang dikenal sebagai *Absolute Radio Frequency Channel Number* (ARFCN). Jadi sebuah *handphone* yang dialokasikan pada sebuah ARFCN akan beroperasi pada satu frekuensi untuk mengirim dan satu frekuensi untuk menerima sinyal.

2.2.2 Struktur Sistem Seluler

Pembagian area dalam kumpulan sel-sel merupakan prinsip penting GSM sebagai sistem telekomunikasi selular. Sel-sel tersebut dimodelkan sebagai bentuk heksagonal seperti pada gambar berikut.



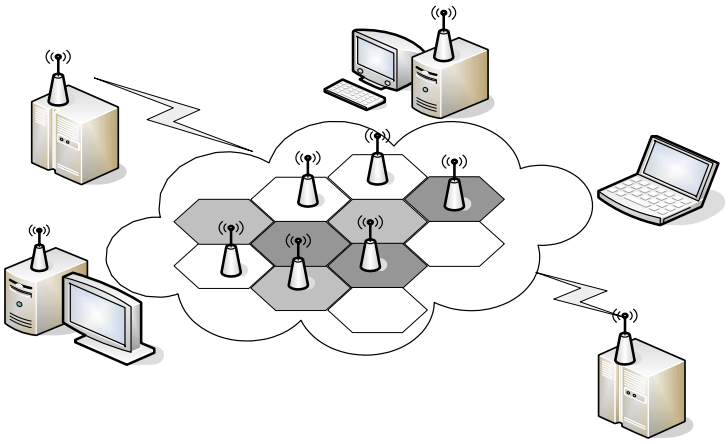
Gambar 2.5 Pembagian area sel frekuensi GSM

Pada gambar 2.2 diatas terlihat tentang pembagian area sel frekuensi GSM. Tiap sel mengacu pada satu frekuensi pembawa atau kanal tertentu. Pada

kenyataannya jumlah kanal yang dialokasikan terbatas, sementara jumlah sel bisa saja berjumlah sangat banyak. Untuk memenuhi hal ini, dilakukan teknik pengulangan frekuensi (*frequency re-use*).

Jelas bahwa semakin besar jumlah himpunan kanal, semakin sedikit jumlah kanal tersedia per-sel dan oleh karenanya kapasitas sistem menurun. Namun, peningkatan jumlah himpunan kanal menyebabkan jarak antara sel yang berdekatan kanal semakin jauh, dan ini mengurangi resiko terjadi interferensi sehingga desain sistem GSM memerlukan kompromi antara kualitas dan kapasitas.

Bagian paling rendah dari sistem GSM adalah *Mobile Station* (MS). Bagian ini berada pada tingkat pelanggan dan *portable*. Pada tiap sel terdapat *Base Transceiver Station* (BTS). BTS ini berfungsi sebagai *station* penghubung dengan MS dan merupakan sistem yang langsung berhubungan dengan *handphone*. BTS pada dasarnya hanya merupakan "pesuruh" saja. Otak yang mengatur lalu-lintas trafik di BTS adalah *Base Station Controller* (BSC). *Location Updating*, penentuan BTS dan proses *handover* pada percakapan ditentukan oleh BSC ini. Beberapa BTS pada satu region diatur oleh sebuah BSC. Seluruh BSC ini dihubungkan dengan *Mobile Switching Center* (MSC). MSC merupakan pusat penyambungan yang mengatur jalur hubungan antar BSC maupun antara BSC dan jenis layanan telekomunikasi lain (PSTN, operator GSM lain, AMPS, dll). Saat ini teknik *switching* terus berkembang, dan begitu pula pada layanan GSM. Beberapa operator GSM di Indonesia telah menerapkan *Intelegent Network* lanjutan dalam teknik *switching*nya.



Gambar 2.6 Jaringan sistem selular

Pada gambar 2.6 diatas terlihat bagaimana carakerja dari jaringan sistem selular.

2.3 Perangkat Lunak

2.3.1 Pengertian Perangkat Lunak

Perangkat lunak terdiri dari program, dokumen dan data yang berisi perintah atau program komputer yang bila dieksekusi akan memberikan fungsi dan unjuk kerja seperti yang diinginkan.

Perancangan perangkat lunak adalah disiplin manajerial dan teknis yang berkaitan dengan pembuatan dan pemeliharaan produk perangkat lunak secara sistematis, termasuk pengembangan dan modifikasinya, yang dilakukan pada waktu yang tepat dan dengan mempertimbangkan faktor biaya.

Software engineer bertugas melakukan analisa, rancangan, uji dan verifikasi, dokumentasi, pemeliharaan perangkat lunak, serta pengelolaan proyek. *Software engineer* harus mempunyai keterampilan dan pengalaman seorang *programmer.*

Programmer adalah individu yang bertugas dalam hal rincian implementasi, pengemasan, dan modifikasi algoritma serta struktur data, dituliskan dalam sebuah bahasa pemrograman tertentu.

2.3.2 *Life Cycle* Perangkat Lunak

Siklus hidup (*Life-cycle)* sebuah perangkat lunak mencakup semua kegiatan yang perlu dilakukan untuk mendefinisikan, mengembangkan, menguji, mengantarkan, mengoperasikan, dan memelihara produk perangkat lunak. Beberapa model tersebut yaitu :

1. *Phased Model* (Model Fase)

Model ini membagi *life cycle* ke dalam sederetan kegiatan (fase). Setiap fase membutuhkan informasi masukan, proses, dan produk yang terdefinisi dengan baik. Deretan fase tersebut adalah :

a. Analisa, terdiri dari fase pertama yaitu perencanaan yang menghasilkan dua produk yaitu Pendifinisian Sistem (*System Definition)* dan Perencanaan Proyek (*Poject Plan)* dan fase kedua yaitu penetapan persyaratan yang menghasilkan sebuah produk spesifikasi kebutuhan perangkat lunak (*Software Requirements Specifications).*
b. Perancangan, melakukan identifikasi terhadap komponen perangkat lunak (fungsi, arus data, penyimpanan data), hubungan antar komponen, struktur perangkat lunak (dekomposisi menjadi modul-modul dan antar muka Perangkat Lunak). Fase ini menghasilkan arsitektur rinci, terutama dalam bentuk algoritma-algoritma.

c. Implementasi, adalah terjemahan langsung arsitektur rinci ke dalam bahasa pemrograman tertentu.
d. Pengujian, terdiri dari fase pertama yaitu uji integrasi dengan melakukan pengujian terhadap semua modul dan pengantarmukaan sehingga pada level sistem dapat beroperasi dengan benar, dan fase kedua yaitu uji penerimaan dengan melakukan berbagi pengujian mengacu kepada berbagai persyaratan yang telah ditentukan.
e. Pemeliharaan, terdiri dari fase peningkatan kemampuan, adaptasi terhadap lingkungan pemrosesan, dan melakukan berbagai koreksi atas kesalahan yang terjadi

2.4 UML *(Unified Modelling Language)*

2.4.1 Pengertian UML *(Unified Modelling Language)*

Merupakan bagian dari OOAD (*Object Oriented Analysis and Design*) yang dimunculkan sekitar akhir tahun 80-an dan awal tahun 90-an. UML adalah sebuah alat untuk menentukan visualisasi, konstruksi dan mendokumentasikan artifact dari system *software,* untuk memodelkan bisnis dan sistem non software lainnya. UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan sistem yang besar dan kompleks.

Maksud dari *artifact* diatas adalah bagian dari informasi yang digunakan atau dihasilkan oleh proses pembuatan perangkat lunak, *artifact* tersebut dapat berupa model, deskripsi atau perangkat lunak.

2.4.2 Bagian-bagian dari UML

Saat kita membuat model untuk membangun suatu *software* juga terdapat berbagai sudut pandang (View). View tersebut dibagi menjadi:

1. Use case view

Melakukan fungsionalitas sistem yang seharusnya dilakukan sesuai dengan yang digunakan *eksternal actors. Use case view* dapat membantu kita untuk memahami dan menggunakan sistem yang kita modelkan. *View* ini melihat pada bagian *actor* dan *use case* berinteraksi. *Actor* menggambarkan pengguna (*user*) system, sedangkan *use case* dapat digambarkan sebagai suatu cara tertentu untuk menggunakan system dari sudut pandang satu pengguna. Terdapat beberapa diagram yang digunakan dalam *use case view* yaitu:

a. Use Case Diagram

b. Sequence Diagram

c. Collaboration Diagram

d. Actifity Diagram

2. Logical View

Mendeskripsikan bagaimana fungsionalitas dari system. *logical view* mengarah pada persyaratan (*requirment*) fungsionalitas system. *View* ini melihat pada kelas-kelas dan hubungan antar kelas-kelas tersebut. Terdapat beberapa diagram yang digunakan dalam *use case view* yaitu:

a. Class Diagram

b. Sequence Diagram

c. Collaboration Diagram

d. State Chart Diagram

3. Component View

Component view mengarah kepada pengaturan *software*. *View* ini mengandung informasi mengenai komponen-komponen *software*. Hanya ada satu jenis diagram yang digunakan pada *view* ini, yaitu *Component Diagram.*

4. Deployment View

Deployment view memperlihatkan pemetaan setiap proses kedalam *hardware. View* ini hanya memiliki satu diagram, yaitu *Deployment Diagram.*

2.4.3 Diagram Grafis UML

Diagram berbentuk yang menunjukan symbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari system.

Sebuah diagram merupakan bagian dari suatu *view* tertentu dan ketika digambarkan biasanya dialokasikan untuk *view* tertentu. Adapun jenis diagram tersebut antara lain:

1. Use case diagram

Menggambarkan sejumlah *external actors* dan hubungannya ke *use case* yang diberikan oleh sistem. *Use case* digambarkan hanya untuk dilihat dari luar *actor* (keadaan lingkungan sistem yang dilihat user) dan bukan bagaimana fungsi yang ada didalam sistem.

2. Class Diagram

Menggambarkan hubungan antar kelas dan penjelasan detail tiap-tiap kelas didalam model desain dari suatu sistem. *Class diagram* juga merupakan pondasi untuk *component diagram* dan *deployment diagram*.

3. Sequence Diagram

Menggambarkan kolaborasi dinamis antar sejumlah *object.* Kegunaanya untuk menunjukan rangkaian pesan yang dikirim antar *object* juga interaksi antar *object.* Sebaiknya digunakan diawal tahap desain atau analisis, karena kesederhanaanya dan mudah untuk dimengerti.

4. Collaboration Diagram

Menggambarkan kolaborasi dinamis seperti *sequence diagram*, akan tetapi *sequence diagram* lebih menekankan pada urutan kejadian. *Collaboration diagram* lebih menekankan pada hubungan antar *object*

5. Activity Diagram

Menggambarkan rangkaian dari alur kerja (*workflow*), digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya, seperti *use case* atau interaksi.

6. State Chart Diagram

Menggambarkan semua *state* (kondisi) yang dmiliki oleh suatu *object* dari suatu *class* dan keadaan yang menyebabkan *state* berubah. *State class* tidak digambarkan untuk semua *class*, hanya yang mempunyai sejumlah *state* yang terdefinisi dengan baik dan kondisi *class* berubah oleh *state* yang berbeda.

7. Component Diagram

Menggambarkan struktur fisik kode dari komponen. Komponen dapat berupa *source code*, komponen biner, *executable componen*. Sebuah komponen berisi informasi tentang *logic class* atau *class* yang di implementasikan sehingga membuat pemetaan dari *logical view* ke *component view*.

8. Deployment Diagram

Deployment diagram memperlihatkan pemetaan *software* kepada *hardware.*

2.5 Database

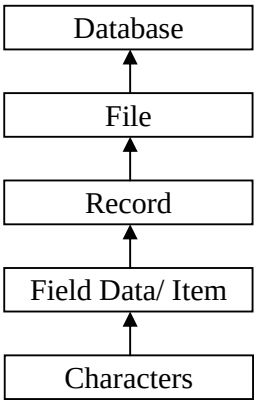
Database merupakan kumpulan dari item data yang saling berhubungan satu dengan yang lainnya yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, tersimpan di hardware komputer dan dengan software untuk melakukan manipulasi untuk kegunaan tertentu.

Beberapa alasan mengapa diperlukan database, yaitu :

1. Salah satu komponen penting dalam *sistem informasi*, karena merupakan dasar dalam menyediakan informasi
2. Menentukan kualitas informasi : akurat, tepat pada waktunya dan relevan. Informasi dapat dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya
3. Mengurangi duplikasi data (data r*edudanc*y)
4. Hubungan data dapat ditingkatkan (data *relatability*)
5. Mengurangi pemborosan tempat simpanan luar

2.5.1 Tingkatan *Database*

Berikut merupakan tingkatan dari dari *Database*



Gambar 2**.7** Tingkatan Database

Pada gambar 2.7 diatas menjelaskan tentang Tingkatan Database yang mana dari Tingkatan Database tersebut dapat diuraikan lebih rinci lagi pada keterangan dibawah ini. yaitu:

a. ***Characters*** : merupakan bagian data yang terkecil, dapat berupa karakter numerik, huruf ataupun karakter-karakter khusus (special characters) yang membentuk suatu item data / field.

b. ***Field*** : merepresentasikan suatu atribut dari record yang menunjukkan suatu item dari data, seperti misalnya nama, alamat dan lain sebagainya. Kumpulan dari field membentuk suatu record.

   a} field name: harus diberi nama untuk membedakan field yang satu dengan lainnya

   b} field representation: tipe field (karakter, teks, tanggal, angka, dsb), lebar field (ruang maksimum yang dapat diisi dengan karakter-karakter data)

c) field value: isi dari field untuk masing-masing record.

c. ***Record*** : Kumpulan dari field membentuk suatu record. Record menggambarkan suatu unit data individu yang tertentu. Kumpulan dari record membentuk suatu file. Misalnya file personalia, tiap-tiap record dapat mewakili data tiap-tiap karyawan.

d. ***File***: File terdiri dari record-record yang menggambarkan satu kesatuan data yang sejenis. Misalnya file mata pelajaran berisi data tentang semua mata pelajaran yang ada.

e. ***Database*** : Kumpulan dari file atau tabel membentuk suatu database

2.5.2 *Database Management System* (DBMS)

DBMS adalah kumpulan database (basis data) beserta aplikasi yang digunakan untuk mengolah basis data tersebut. DBMS biasanya menawarkan kemampuan pengolahan data secara terintegrasi. Kemampuan DBMS antara lain :

1. Membuat, menghapus, menambah, dan memodifikasi data
2. biasanya dilengkapi dengan fitur berbasis windows yang akan memudahkan pengguna dalam pengolahan data
3. memberikan keamanan bagi data sehingga tidak semua orang dapat mengakses data yang ada (login dan password)
4. kemampuan berkomunikasi dengan program aplikasi lain. Misalnya dengan visual basic
5. kemampuan mengakses melalui komunikasi antar komputer (client server)

2.6 AT-*Command*

AT-*Command* atau *ATtention Command* yaitu perintah AT (Hayes AT-*Command)* yang digunakan untuk berkomunikasi dengan terminal (modem) melalui gerbang serial pada komputer. Dengan penggunaan perintah AT, dapat diketahui atau dibaca kondisi dari terminal, seperti mengetahui kondisi sinyal, kondisi baterai, mengirim pesan, membaca pesan, menambah item pada daftar telepon, dan sebagainya. Beberapa jenis ponsel memiliki *extended* AT *Command* yang bisa digunakan untuk mengambil informasi jenis, model hp, nomor *Internasional Mobile Station Equipment Identity* (IMEI) , SIM *Subscriber Identification Number* (IMSI), status batere, kekuatan sinyal, nama operator, lokasi dan cell ID.

AT-Command	**Singkatan**	**Fungsi**
AT+C.....	Attention + Command.....	Awal perintah hayes untuk....
AT+CGMI	Get Manufactured Identification	Menampilkan ID pabrik pembuat
AT-CGMM	Get Mobile Model	Menampilkan ID model Handphone
AT+CGSN	Get Serial Number	Menampilkan serial nomor IMEI
AT+COPS	Operator Selection	Memilih operator jaringan
AT+CLCK	Lock	Merubah penguncian telepon
AT+CPWD	Change Password	Merubah *password*
AT+CLIP	List Incoming Phone	Menampilkan nomor telepon masuk
AT+CPIN	Pin	Memasukan PIN
AT+CBC	Battery Charge	Mengisi batere
AT+CSQ	Signal Quality	Menampilkan kualitas sinyal
AT+CPBW	Phone Book Write	Memasukan data pada buku telepon
AT+CRSM	Restricted SIM Access Mode	Memblokir akses SIM
AT+CAOC	Advice Of Charge	Menampilkan biaya panggilan
AT+CCLK	Clock Mode	Pengaturan Jam

AT+CALM	Alert Sound Mode	Pengaturan jenis nada dering
AT+CRSL	Ringer Sound Level	Pengaturan volume nada dering
AT+CMUT	Mute Control	Pengaturan tanpa suara
AT+CLVL	Level Volume	Pengaturan volume pengeras suara
AT+CVIB	Vibration Mode	Pengaturan nada getar
AT+CPBR	Phone Book Read	Menampilkan daftar buku telepon
AT+CLCC	List Curent Call	Menampilkan penelepon terakhir

Tabel 2.2 Perintah AT-Command GSM 07.07

Pada table 2.2 diatas yaitu menjelaskan tentang perinta-perintah AT-Command yang digunakan untuk berkomunikasi dengan terminal (Modem) melalui gerbang serial pada komputer.