BAB II

LANDASAN TEORI

2.1 Sejarah GSM

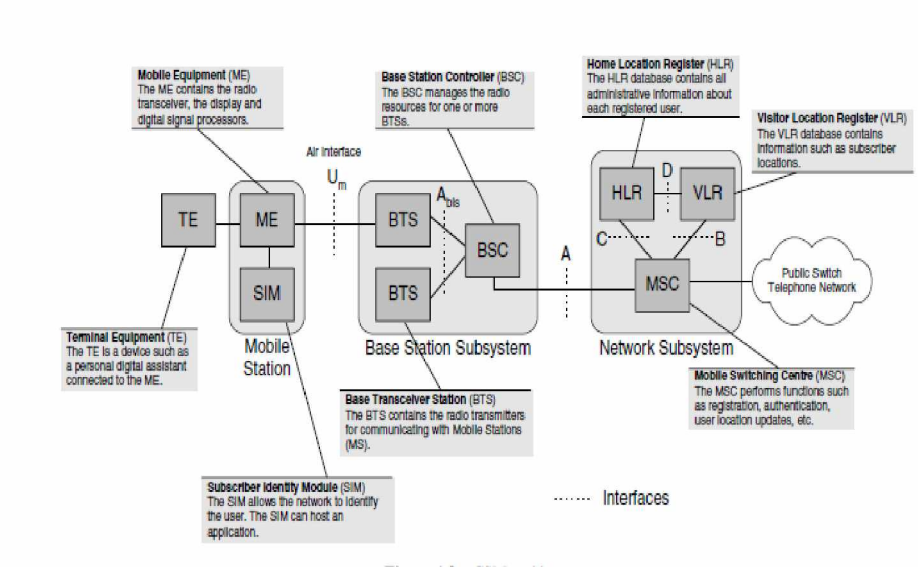
Di Indonesia, liberalisasi bisnis seluler dimulai sejak tahun 1995, saat pemerintah mulai membuka kesempatan kepada swasta untuk berbisnis telepon seluler dengan cara kompetisi penuh. Bisa diperhatikan, bagaimana ketika teknologi *GSM (global system for mobile)* datang dan menggantikan teknologi seluler generasi pertama yang sudah masuk sebelumnya ke Indonesia seperti *NMT (nordic mobile telephone)* dan *AMPS (advance mobile phone system).*

Di tahun 1980-an, teknologi *GSM* datang ke Indonesia, maka para operator pemakai teknologi *AMPS (Advanced Mobile Phone System)* menghilang. Lalu, muncul Satelindo sebagai pemenang, yang kemudian disusul oleh Telkomsel.

Dan pada akhirnya teknologi *GSM* lebih unggul dan berkembang dengan pesat, ini dikarenakan kapasitas jaringan lebih tinggi, karena efisiensi di spektrum frekuensi dari pada teknologi *NMT* dan *AMPS.*

2.1.1 Arsitektur Jaringan GSM

Unsur-unsur yang utama pada arsitektur *GSM* [3GPP-23.002] ditunjukkan pada Gambar 2.1. Jaringan *GSM* terdiri atas tiga subsistem: *Base Station Subsystem (BSS), Network Subsystem (NSS)* dan *Operation Subsystem (OSS). OSS* tidak dijelaskan lebih lanjut, unsur-unsur *BSS* dan *NSS* akan diuraikan lebih lanjut.



Gambar 2.1 Arsitektur GSM

a. Mobile Station (MS)

Merupakan perangkat yang mengirim dan menerima sinyal radio. *MS* dapat berupa mobile handset atau *Personal Digital Assistant (PDA)*. MS terdiri dari *Mobile Equipment (ME)* dan *Subscriber Identity Module (SIM)*. *ME* berisi *transceiver radio*, display dan *Digital Signal Processor*. SIM digunakan agar *network* dapat mengenali user.

b. Base Transceiver Station (BTS)

Base transceiver Station (BTS) berfungsi sebagai interface komunikasi semua MS yang aktif dan berada dalam coverage area *BTS* tersebut. Di dalamnya termasuk modulasi signal, demodulasi, equalize signal dan error coding. Beberapa BTS terhubung pada satu *Base Station Controller (BSC)*. Satu BTS biasanya mampu menghandle 20-40 komunikasi serentak.

c. Base Station Controller (BSC)

BSC berfungsi mengatur konkesi *BTS-BTS* yang berada dalam kendalinya. Fungsi tersebut memungkinkan operasi seperti handover, cell site configuration, management of radio resources dan menyetel power level dari frekuensi radio BTS. Pada jaringan *GSM BSC* mengatur lebih dari 70 *BTS.*

d. Mobile Switching Centre (MSC) dan Visitor Location Register (VLR)

Mobile Switching Centre (MSC) melakukan fungsi registrasi, authentikasi, update lokasi user, billing service dan sebagai interface dengan jaringan lain. Selain itu MSC juga bertanggung jawab untuk call set-up, release dan routing.

e. Visitor Location Register (VLR)

Visitor Location Register (VLR) berisi informasi dinamis tentang user yang terkoneksi dengan mobile network termasuk lokasi user tersebut. VLR biasanya terintegrasi dengan MSC.Melalui MSC, mobile network terhubung dengan jaringan lain seperti *PSTN (Public Switched Telephone Network), ISDN (Integrated Service Digital Network), CSPDN (Circuit Switched Public Data Network)* dan *PSPDN (Packet Switched Public Data Network).*

f. Home Location Register (HLR)

Home Location Register (HLR) adalah elemen jaringan yang berisi detil dari setiap subscriber. Sebuah HLR biasanya mampu mengatur ratusan bahkan ribuan subscriber. Pada jaringan GSM, signaling berbasis pada protokol Signaling System Number 7 (SS7). Penggunaan SS7 dilengkapi

dengan penggunaan protokol *Mobile Application Part (MAP). MAP* digunakan untuk pertukaran informasi lokasi dan subscriber antara HLR dan elemen jaringan lainnya seerti *MSC.*

Untuk setiap subscriber, *HLR* mengatur pemetaan antara *International Mobile Subscriber Identity (IMSI)* dan Mobile Station ISDN Number (MSISDN). Untuk alasan keamanan, IMSI jarang ditaransmisikan melalui perantara radio dan hanya dikenali pada jaringan GSM yang ditentukan. IMSI menggunakan format [ITU-E.212]. Tidak seperti *IMSI, MSISDN* mengidentifikasi subscriber di luar jaringan *GSM, MSISDN* menggunakan format [ITU-E.164].

2.2 Sejarah SMS

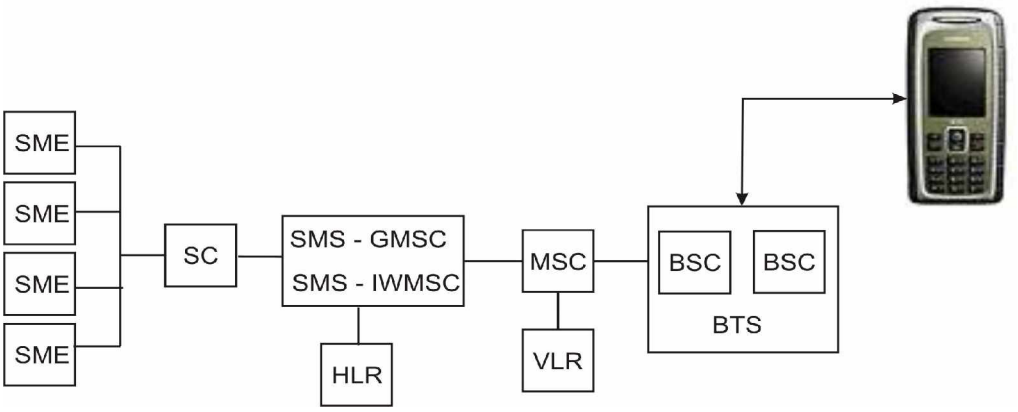
Short Message Sevice (*SMS*) adalah salah satu fasilitas dari teknologi GSM yang memungkinkan mengirim dan menerima pesan pesan singkat berupa text dari *Mobile Station (MS).* Layanan *SMS* juga memungkinkan pengiriman pesan dalam bentuk *alphanumeric,* layanan *SMS* ini banyak diaplikasikan pada sistem komunikasi tanpa kabel (*wireless).* Teknologi wireless dipelopori dari kawasan Eropa yang diawali pada kebutuhan bersama terhadap satu sistem jaringan baru yang dapat menjadi standard jaringan yang berlaku dan dapat diterapkan di seluruh kawasan Eropa. Dalam sistem baru juga harus terdapat kemampuan yang dapat mengantisipasi mobilitas pengguna serta kemampuan melayani lebih banyak pengguna untuk menampung penambahan jumlah pelanggan baru.

Karena hal ini tidak dapat dilakukan dengan mempertahankan sistem analog, maka kemudian diputuskan untuk merombak sistem dan menggantinya dengan sistem digital. Standard baru diperkenalkan dengan nama *Global Standard for Mobile Communications* (GSM). GSM pada awalnya adalah kepanjangan dari *Groupe Speciale Mobile*, sebuah badan gabungan dari para ahli yang melakukan studi bersama untuk menciptakan standard GSM tersebut.

Pada bulan Desember 1992, dilakukan pengiriman pesan menggunakan SMS dari sebuah *Personal Computer* (PC) ke telepon *mobile* (bergerak) dalam jaringan GSM milik Vodafone Inggris, kemudian merambah ke benua Amerika yang dipelopori oleh beberapa operator komunikasi bergerak berbasis digital seperti BellSouth Mobility, PrimeCo dan operator lainnya. Teknologi yang digunakan dari pengiriman SMS yaitu *Store* and *forward service*, jadi SMS yang di kirim akan simpan sementara di server SMS center kemudian dialihkan ke nomor tujuan.

2.2.1 Arsitektur dan Elemen Jaringan SMS

Layanan SMS dibangun dari berbagai entitas yang saling terkait dan mempunyai fungsi dan tugas masing-masing. Tidak ada satupun dalam sistem SMS yang dapat bekerja secara parsial. Entitas dalam jaringan SMS ini disebut juga elemen SMS. Pada gambar Gambar 2.*2* berikut merupakan arsitektur SMS dengan beberapa elemen-elemen yang saling terkait :



Gambar 2.2 Arsitektur SMS

Elemen-elemen dasar pada jaringan SMS :

a. *SME (Short Message Entity)*, merupakan tempat penyimpanan dan pengiriman message yang akan dikirimkan ke MS tertentu.

b. *SC (Service Centre)*, bertugas untuk menerima message dari SME dan melakukan forwarding ke alamat MS yang dituju.

c. *SMS-GMSC (Short Message Service Gateway SMC),* melakukan penerimaan message dari SC dan memeriksa parameter yang ada. Selain itu *GMSC* juga mencari alamat *MS* yang dituju dangan bantuan *HLR*, dan mengirimkannya kembali ke *MSC* yang dimaksud.

d. *SMS IWMSC (Short Message Service Interworking MSC*), berperan dalam SMS Message Origiating, yaitu menerima pesan dari MSC

e. *Hom Location Register (HLR)* merupakan sebuah database yang digunakan sebagai tempat penyimpanan permanen data dan profil pelanggan. Bila diminta oleh SMSC , maka HLR dapat memberikan informasi routing dari pelanggan tertentu. HLR juga dapat memberikan informasi status tujuan apakah aktif atau tidak.

f. *Visitor Location Register (VLR*) merupakan sebuah database tempat menyimpan informasi sementara yang berisi data pelanggan dari sebuah *HLR* yang roaming pada *HLR* lain.

g. *MSC* merupakan sebuah sistem yang melakukan fungsi switching dan mengontrol panggilan telepon dalam sebuah jaringan komunikasi bergerak. *MSC* inilah yang akan mengirimkan sebuah short message ke suatu tujuan tertentu melalui base station yang sesuai.

h. *Base Station Sistem (BSS)* Merupakan kesatuan sistem yang bertanggung jawab mengatur transmisi sinyal elektromagnetik untuk membawa data dari MSC ke perangkat telepon bergerak. Base Station terdiri dari *Base Station Controler (BSC)* dan *Base Tranceiver Station (BTS)* dan juga dikenal dengan nama *cell cite* atau sederhananya cell. Sebuah *BSC* biasanya menangani satu atau lebih *BTS* dan bertanggung jawab menangani pelanggan saat berpindah dari satu cell ke cell lainnya.

i. *Mobile Device* merupakan perangkat yang mempunyai kemampuan mengirimkan dan menerima short message, biasanya berupa telepon seluler dengan teknologi digital. Akan tetapi, saat ini jenis terminal berkembang sesuai aplikasi dan kebutuhan seperti *POS, laptop* dan *Personal Digital Assistant (PDA).*

2.2.2 Basic Features SMS

SMS mempunyai beberapa basic feature, diantaranya :

a. Message Submission and Delivery

Terdiri dari message sending dan message delivery. Pada message sending, pesan dikirm dari MS ke SMSC, dialamatkan ke SME lain sebagai mobile user lain atau host internet. Originator (asal) SME menentukan validity period dari pesan tersebut, pesan yang sudah tidak valid lagi akan dihapus oleh SMSC sepanjang pengiriman pesan. Fitur ini dikenal sebagai *Short Message-Mobile Originated (SM-MO).*

Pada message delivery, pesan disampaikan oleh SMSC ke MS. Dikenal sebagai Short Message Mobile Terminated (SM-MT). SM-MO dan SM-MT dapat dikirim / diterima saat voice call atau koneksi data sedang berlangsung. Pada GSM pesan dikirim pada channel SDCCH/SACCH, pada GPRS pesan dikirim pada channel PDTCH.

b. Status Report

SME asal (originator) meminta status report pada pengiriman pesan singkat ke *SME* penerima (recipient). Status report memberikan indikasi pada user asal apakah pesan terkirim dengan sukses atau tidak kepada SME penerima.

c. Reply Path

Reply Path dapat diatur oleh *SME* asal (atau *SMSC serving*) untuk mengindikasikan bahwa *SMSC* serving dan mampu untuk menghandle secara langsung *reply* dari *SME* penerima.

d. Addressing Mode

Addressing mode menggunakan *MSISDN* pada format [ITU-E.164]. Email address ditentukan oleh *IETF* pada format [RFC-2822] atau operator specific numbering.

e. Validity Period

Pesan originator dimungkinkan untuk menentukan validity period sebuah pesan. Validity period ini menentukan batas waktu sebuah pesan harus dikirim sebelum akhirnya dihilangkan oleh jaringan.

2.2.3 Protokol Layer

SMS protocol layer terdiri dari 4 layer yaitu *application layer, transfer layer, reply layer* dan link layer.

a. Application Layer

Diterapkan pada *SME* dalam bentuk software aplikasi yang berfungsi untuk mengirim, menerima dan menginterpretasikan isi pesan (seperti : editor pesan, games, dll). Application layer disebut juga *SM-AL (Short Message Application Layer).*

b. Transfer Layer

Pesan yang dianggap sebagai serangkaian bilangan oktet yang mengandung informasi seperti panjang pesan, pengirim atau penerima pesan, tanggal penerimaan pesan. Transfer layer disebut juga *SM-TL* (*Short Message Transfer Layer).*

c. Relay Layer

Relay layer mengizinkan pengiriman pesan antar elemen network yang berbeda. Sebuah elemen network menyimpan pesan sementara jika elemen berikutnya tidak menyediakan pesan yang akan diforward. Pada relay layer MSC menangani 2 fungsi switching : SMS-GMSC dan SMS-IWMSC.

1. *SMS-GMSC (SMS-Gateway Mobile Switching Center)*

   Merupakan *MSC* yg dapat menerima *SMS* dari *SMSC*, menanyakan info routing ke *HLR*, dan mengirim pesan ke *MSC* dari *MS* penerima.

2. *SMS-IWMSC (SMS-Interworking Mobile Switching Center)*

   Merupakan *MSC* yg dapat menerima *SMS* dari jaringan mobile dan mengirimkannya ke *SMS* yg tepat.

SMS-GMSC/SMS-IMSC biasanya diintegrasikan dgn *SMSC. Relay* layer disebut juga *SM-RL (Short Message Relay Layer).*

d. Link Layer

Link layer mengizinkan pengiriman pesan pada level physical. Untuk tujuan ini, pesan diprotek untuk mengatasi kesalahan low level channel. Link layer disebut juga *SM-LL (Short Message Link Layer).*

2.2.4 SMS Protocol Data Unit

protocol connectionless, terdiri dari 6 PDU :

a. *SMS-DELIVER* : Mengirimkan pesan dan info terkait dari SMSC ke MS
b. *SMS-DELIVER-REPORT* : Mengirimkan pesan berisi penyebab kegagalan dari MS ke SMSC

c. *SMS-SUBMIT* : Mengirimkan pesan dan info terkait dari *MS* ke *SMSC*

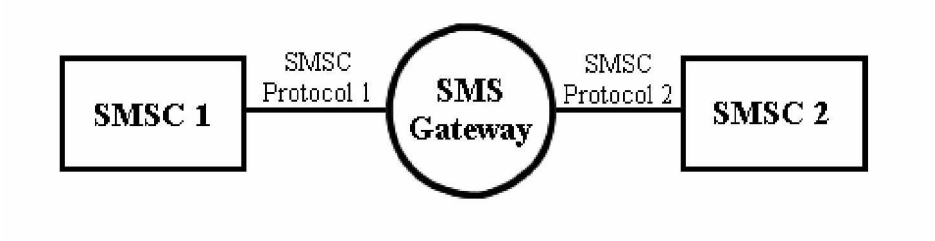
d. *SMS-SUBMIT-REPORT* : Mengirimkan pesan berisi penyebab kegagalan dari SMSC ke MS

e. *SMS-STATUS-REPORT* : Mengirim laporan status

f. *SMS-COMMAND* : Mengirim perintah yang akan dieksekusi untuk SMS-*SUBMIT* yg dikirim dari MS ke *SMSC*

2.2.5 SMS Gateway

Satu permasalahan dalam SMS messaging adalah *SMSCs* yang dikembangkan oleh perusahaan berbeda menggunakan protokol komunikasi yang berbeda pula. Sebagai contoh, Nokia mempunyai suatu protokol SMSC yaitu *CIMD* sedangkan *SMSC* penjual yang lain, *CMG*, mempunyai suatu protocol SMSC yaitu EMI. dua *SMSCs* tidak bisa dihubungkan jika mereka tidak mendukung suatu *SMSC* protokol umum. Untuk mengatasi masalah ini, SMS Gateway ditempatkan diantara kedua *SMSCs.* Pada gambar 2.3, SMS Gateway bertindak sebagai relay antara kedua *SMSCs.* Tugasnya menterjemahkan protocol *SMSC* yang satu protokol *SMSC* yang lain. Ini dapat digunakan sebagai penghubung antar *SMSCs* dengan tujuan sebagai pertukaran antar operator SMS.



Gambar 2.3 SMS Gateway menghubungkan SMS Center 1 dan SMS Center 2

2.2.6 Layanan Aplikasi SMS

Layanan aplikasi SMS merupakan sebuah layanan yang bersifat *none real time* dengan sebuah short message dapat di *submit* ke suatu tujuan, tanpa memperdulikan apakah tujuan tersebut aktif atau tidak. Bila dideteksi bahwa tujuan tidak aktif, maka sistem akan menunda pengiriman ke tujuan hingga tujuan aktif kembali. Pada dasarnya sistem SMS akan menjamin *delivery* dari suatu short message hingga sampai tujuan. Kegagalan pengiriman yang bersifat sementara, seperti tujuan tidak aktif, akan selalu teridentifikasi sehingga pengiriman ulang short message akan selalu dilakukan, kecuali apabila diberlakukuan aturan bahwa short message yang telah melampaui batas

waktu tertentu harus dihapus dan dinyatakan gagal kirim. Berdasarkan mekanisme distribusi pesan *SMS*, terdapat 4 macam mekanisme pengantaran pesan 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

2.2.7 Prinsip Kerja SMS

a. Prinsip kerja *SMS* ini adalah bahwa setiap jaringan mempunyai suatu *service center (SC).* Pesan tidak langsung dikirimkan ke tempat tujuan melainkan di simpan terlebih dahulu di SC sebagai *interface* dalam *Public Land Mobile Network (PLNM).*

b. Transmisi *SMS* dapat terjadi meskipun MS sedang melakukan komunikasi dengan mobile station (MS) yang lain. Hal ini di mungkinkan karena kanal radio untuk transmisi voice telah ditentukan selama durasi pemanggilan sedangkan SMS merambat pada kanal radio dengan memanfaatkan jalur sinyal.

c. Pengiriman SMS yang menggunakan kanal sinyal memiliki dua tipe :

   1. *SMS Point To point* Menyediakan mekanisme untuk mengirimkan pesan hanya dari satu MS ke MS tertentu, berupa pesan pendek ke dan dari piranti bergerak
   2. *SMS broadcast (Point to multipoint)* pengiriman SMS ke beberapa MS sekaligus

d. Pesan yang tidak terkirim, akan memunculkan informasi report yang menyatakan pesan SMS gagal terkirim

e. Walaupun ponsel tidak aktif , sms akan tetap masuk dan disimpan di SMS dengan waktu tertentu, apabila ponsel aktif sebelum batas waktu, maka *SMS* akan dikirimkan.

2.3 Perangkat Lunak

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 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 implentasi pengemasan, dan modifikasi algoritma serta struktur data, dituliskan dalam sebuah bahasa pemrograman tertentu.

Pembuatan sebuah perangkat lunak melalui beberapa tahap atau fase yang menggambarkan sebuah kegiatan yang akan dilakukan untuk memudahkan dalam mendefinisikan, mengembangkan, menguji, mengantarkan, mengoperasikan, dan memelihara produk perangkat lunak. Setiap fase membutuhkan informasi masukan, proses, dan produk yang terdefinisi dengan baik. Deretan fase tersebut adalah :

a. Analisa, terdiri dari dua fase yaitu fase perencanaan yang menghasilkan dua produk yaitu Pendifinisian Sistem (*System Definition)* dan Perencanaan Proyek (*Poject Plan)* serta fase penetapan persyaratan yang menghasilkan sebuah produk spesifikasi kebutuhan perangkat lunak (*Software Requirements Specifications).*

b. Perancangan, yaitu 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 berbagai 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 *Database*

Database merupakan kumpulan data yang saling berhubungan satu sama lainnya yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu. Beberapa hal yang harus di perhatikan dalam membangun suatu *database*, yaitu:

a. Entitas

   Merupakan suatu *object* yang memiliki karakteristik tertentu sebagai suatu atribute, misalnya entitas mahasiswa memiliki atribute nama dan alamat

b. Field

Mempresentasikan suatu atribut dari record yang menunjukkan suatu item dari data, seperti misalnya nama, alamat dan lain sebagainya. Kumpulan dari field membentuk suatu record.

c. Record

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

d. Atribute

Merupakan karakteristik dari sebuah entitas. Nama, alamat merupakan atribute dari entitas mahasiswa.

2.5 *Database Management System* (*DBMS*)

DBMS merupakan kumpulan *database* (basis data) beserta aplikasi yang digunakan untuk mengolah basis data tersebut. Berdasarkan orientasi pemakainya, DBMS dapat dikelompokan dalam 2 kategori yaitu :

1. DBMS berorientasi untuk satu atau sedikit pemakai , contoh dari DBMS yang tersebut antara lain MS-Access, dBase, foxbase, borland paradox.DBMS ini dapat dengan mudah di pasang di komputer pribadi (PC).
2. DBMS yang berorientasi untuk banyak pemakai Oracle, borland interbase, MSSQL-Server, sysbase dan infomix merupakan contoh-contoh DBMS yang lebih berorientasi untuk banyak pemakai karena itu lebih ditujukan untuk pemakaian pada sistem jaringan komputer (LAN ataupun WAN). DBMS

pada umumnya untuk meyediakan beberapa fasilitas untuk operasi pengolahan data atau sering dikenal operasi basis data, antara lain :

1. Penambahan Data.
2. Pencarian Data.
3. Pengubahan Data.
4. Penghapusan data.
5. Pengurutan Data.
6. Penggabungan Data.
7. Dan lain-lain.

2.6 *Protocol Data Unit* (PDU)

Langkah-langkah untuk menentukan kode PDU yang dibutuhkan untuk mengirimkan *(send)* SMS ke SMS *Center* adalah:

a. Nomor SMS Center Terdiri dari tiga *subheader*, yaitu:
   1. Jumlah pasangan heksa desimal SMS *Center* dalam bilangan heksa.
   2. Kode nasional dan kode internasional, nomor kode nasional yang digunakan adalah 81 sedangkan nomor kode internasional adalah 91.
   3. Nomor SMS *Center*, dalam pasangan bilangan heksa yang saling dipertukarkan dengan aturan tertentu, jika tersisa satu angka heksa yang tak memiliki pasangan maka angka tersebut dipasangkan dengan huruf F didepannya.

Contoh konversi nomor ponsel kedalam kode PDU dengan menggunakan :

1. Kode Nasional

   SMS Center menggunakan nomor 0856000000 maka :

   a. Terdapat 6 pasang angka heksa desimal.

   b. Kode nasional 81, ada 1 pasang

   c. 80-56-00-00-00

   Maka kode PDU yang didapat : 06818056000000

2. Kode Internasional

   SMS Center menggunakan nomor 62811000000, maka:

   a. Terdapat 7 pasang angka heksa desimal

   b. Kode internasional 91, ada 1 pasang

   c. 26-18-01-00-00-F0

   Maka kode PDU yang didapat : 07912618010000F0

Pada tabel 2.1 dan tabel 2.2 Berikut tertera beberapa nomor SMS-*Center* operator selular di Indonesia :

Tabel 2.1 Tabel Nomor *SMS-Center*
Operator Selular dan Kode PDU (Cara I)

No	**Operator Selular**	**SMS-*Center* No**	**Kode PDU**
1	Telkomsel	0811000000	06818011000000
2	Satelindo	0816125	0581806121F5
3	Excelcom	0818445009	06818081440590
4	Indosat-M3	0855000000	06818055000000

Tabel 2.2 Tabel Nomor SMS-Center
Operator Selular dan Kode PDU (Cara II)

No	**Operator Selular**	**SMS-*Center* No**	**Kode PDU**
1	Telkomsel	6281000000	07912618010000F0
2	Satelindo	62816125	059126181652
3	Excelcom	62818445009	07912618485400F9
4	Indosat-M3	62855000000	07912658050000F0

b. Tipe SMS, untuk pengiriman atau *send* tipe SMS adalah 1 dalam heksa 01

c. Nomor referensi SMS, nomor ini dibiarkan dulu 0 atau dalam heksa 00, ponsel secara otomatis akan memberi nilai nomor referensinya.

d. Nomor ponsel penerima, terdiri atas tiga bagian *subheader* yaitu:

   1. Jumlah bilangan desimal ponsel yang dituju dalam bilangan heksa
   2. Kode nasional atau kode internasional, kode *subheader* nasional adalah 81 dan internasional adalah 91.
   3. Nomor ponsel yang dituju dalam pasangan heksa yang saling dipertukarkan menurut aturan tertentu.

   Contoh untuk nomor ponsel yang dituju adalah : 085221960301, maka konversi dalam heksa desimal adalah dengan cara seperti yang dipaparkan berikut.

1. Kode Nasional:

   a. terdapat 12 angka, berarti dalam heksa adalah 0C

   b. kode nasional adalah 81

   c 80-25-12-69-03-01

   Maka kode *subheader* PDU yang didapat adalah : 0C81802512690301

2. Kode Internasional

   a 62 81 32 12 17 32 6

   b. Terdapat 13 angka, berarti dalam heksa adalah 0D

   c. Kode internasional adalah 91

   d. 261823212732F6

   Maka kode *subheader* PDU yang didapat adalah : 0D91261823212732F6

a. Bentuk SMS

   00 : dikirim sebagai SMS

   01 : dikirim sebagai Telekomunikasi

b. Skema *encoding* data *Input* dan *Output* (I/O)

   Terdapat dua skema yaitu ;

   1. Skema 7 bit, ditandai dengan angka 00
   2. Skema 8 bit, ditandai lebih besar dari nol (dalam heksa)

c. Jangka Waktu atau validasi sebelum SMS *expired* atau kadaluarsa, jika bagian ini diloncat, berarti berlakunya SMS tidak dibatasi, sedangkan jika diisi dengan suatu bilangan integer yang kemudian diubah ke dalam bilangan heksa, maka bilangan tersebut mewakili jumlah waktu validasi SMS tersebut.

Berikut Tabel 2.3 merupakan validitas waktu SMS

Tabel 2.3 Tabel Validitas Waktu SMS

Integer (INT)	**Jangka Waktu Validasi**
0-143	(INT+1)x 5 menit (berarti 5 sampai dengan 12 jam)
144-167	12 jam + (INT-143) x 30 menit
168-196	(INT-166) x 1 hari
197-255	(INT-192) x 1 minggu

d. Isi SMS

Terdiri dari dua subheader yaitu:

1. Panjang Isi (Jumlah huruf dalam SMS), misal untuk kata " hello terdiri dari 5 huruf.
2. Isi berupa pasangan bilangan heksa desimal, skema 7 bit dapat dilihat pada Tabel 2.4 dibawah ini.

e. Tanggal dan waktu SMS di-*stamp* dalam SMS *Center* diwakili oleh 12 bilangan

heksa (6 pasang) yang berarti YY/MM/DD hh:mm:ss

Contoh:

207022512308 02/07/22 12:32:08 22 Juli2002 15:23:08 WIB

Ada 2 langkah yang harus kita lakukan untuk mengkonversikan isi SMS , yaitu :

Langkah pertama : mengubahnya menjadi kode 7 bit

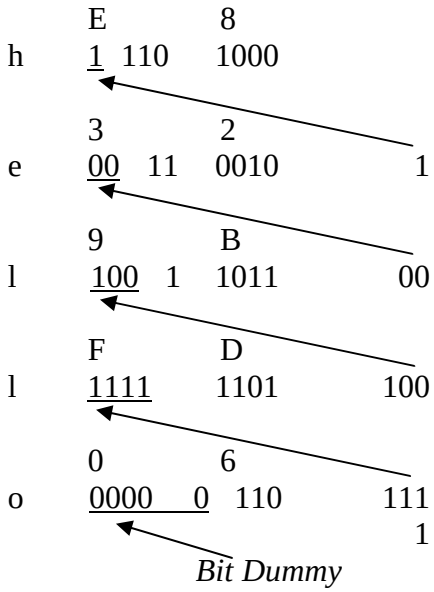
Langkah kedua : mengubah kode 7 bit menjadi 8 bit yang diwakili oleh pasangan heksa.

Contoh untuk kata " hello

Langkah pertama

Bit	7	1
h	110	100
e	110	0101
l	110	1100
l	110	1100
o	110	1111

Langkah kedua



Sehingga hasil konversi kata " hello ke bilangan heksa adalah **E8329BFD06** Oleh karena total 7 bit x 5 huruf = 35 bit, sedangkan yang kita perlukan untuk mengubah ke 8 bit adalah 8 bit x 5 huruf = 40 bit, maka diperlukan 5 bit *dummy* yang diisi bilangan 0.

Setelah masing-masing *header* dan *subheader* untuk mengirim pesan pesan dipecahkan, maka *header-header* diatas digabung menjadi sebuah PDU lengkap.

Seperti contoh untuk mengirim kata Hello SMS ke nomor 628129573337 lewat SMS Center Exelcom tanpa membatasi jangka waktu valid, maka PDU lengkapnya:

07912618485400F901000C9126I892753373000005E8329BFD06.

2.7 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 baterai, kekuatan sinyal, nama operator, lokasi dan cell ID. Pada tabel 2.4 berikut diperlihatkan beberapa jenis perintah Hayes yang berhubungan dengan penanganan pesan-pesan AT-Command GSM versi 07.07.

Tabel 2.4 Perintah AT-Command pada GSM Versi 07.07

AT-Command	Singkatan	Fungsi
ATE1	*Activate Command*	Mengaktivasi At-Command 07.07
AT+C.....	*Attention + Command*	Awal perintah hayes untuk....
AT+COPS	*Operator Selection*	Memilih operator jaringan
AT+CLCK	*Lock*	Menampilkan penguncian telepon
AT+CBC	*Battery Charge*	Menampilkan level batere
AT+CSQ	*Signal Quality*	Menampilkan kualitas sinyal
AT+CCLK	*Clock Mode*	Pengaturan Jam
AT+CPBR	*Phone Book Read*	Membaca daftar buku telepon
AT+CPBS	*Phne Book Selectiono*	Menampilkan isi buku telepon
AT+CCSA	*Service Center Address*	Menampilkan Lokasi *Service Center*
AT+CMGS	*Message Get Send*	Mengirimkan SMS
AT+CMGL	*Message Get List*	Membaca isi *inbox* dan *outbox* SMS

2.8 MySQL

MySQL dikembangkan sekitar sekitar tahun 1994 oleh sebuah perusahaan pengembang *software* sekaligus konsultan basis data bernama MySQL AB yang bertempat di Swedia. Waktu itu perusahaan tersebut masih bernama TcX *DataConsult* AB, dan tujuan awal dikembangkannya MySQL adalah untuk mengembangkan aplikasi berbasis *website* pada *client.*

MySQL merupakan basis data yang dikembangkan dari bahasa SQL. Ada beberapa alasan mengapa MySQL menjadi program basis data yang sangat populer dan digunakan oleh banyak orang. Alasan-alasan tersebut diantaranya sebagai berikut :

1. MySQL adalah basis data yang memiliki kecepatan yang tinggi dalam melakukan pemrosesan data, dapat diandalkan, dan mudah digunakan serta mudah dipelajari.
2. MySQL didukung banyak bahasa pemrograman seperti C, C++, Perl, Python, Java dan PHP. Pengguna dapat menggunakan bahasa pemrograman tersebut untuk berinteraksi maupun berkomunikasi dengan MySQL *server*, atau dapat juga digunakan sebagai komponen pembentuk antarmuka (*interface*) dari suatu basis data MySQL.
3. MySQL dapat melakukan koneksi dengan *client* menggunakan protokol TCP/IP, *Unix socket* (Unix), atau *Named Pipes* (NT).
4. Dalam hal menampung data, MySQL dapat menangani basis data dengan skala yang cukup besar dengan jumlah *record* mencapai lebih dari 50 juta, dapat menampung 60 ribu tabel, dan juga bisa menampung 5 milyar baris data. Selain itu, batas *index* pada tiap tabel menampung mencapai 32 *index*.
5. Dalam hal relasi antartabel pada suatu basis data, MySQL menerapkan metode yang sangat cepat, yaitu dengan menggunakan metode *one-sweep multijoin* yaitu efisiensi pengelolaan informasi yang diminta oleh user dari beberapa tabel sekaligus.

6. *Multiuser*, yaitu dalam satu basis data *server* pada MySQL dapat diakses oleh beberapa user dalam waktu yang sama tanpa mengalami konflik atau kemacetan sistem.
7. Keamanan yang dimiliki basis data MySQL dikenal baik, karena memiliki lapisan sekuritas seperti *level subnetmask*, nama *host* dan izin akses user dengan sistem perizinan yang khusus serta *password* yang dimiliki setiap user dalam bentuk terenkripsi.
8. MySQL adalah *software* basis data yang bersifat *free* atau gratis.

MySQL termasuk DBMS yang memiliki pengembangan tipe data yang cukup beraneka ragam, mulai dari pengembangan tipe data angka, karakter maupun tanggal. Berbagai tipe data yang didukung oleh MySQL untuk lebih lengkapnya dapat dilihat pada Tabel 2.5 dengan deskripsi pada masing-masing tipe data.

Tabel 2.5 Tipe data pada MySQL

Tipe Data	Deskripsi
TinyInt	Ukuran 1 *byte*. Bilangan bulat terkecil, dengan jangkauan untuk bilangan bertanda: -128 sampai dengan 127 dan untuk yang tidak bertanda: 0 sampai dengan 255. Bilangan tak bertanda ditandai dengan kata *Unsigned*.
SmallInt	Ukuran 2 *byte*. Bilangan bulat dengan jangkauan untuk bilangan bertanda: -32768 sampai dengan 32767 dan untuk yang tidak bertanda: 0 sampai dengan 65535.
MediumInt	Ukuran 3 *byte*. Bilangan bulat dengan jangkauan untuk bilangan bertanda: -8388608 sampai dengan 8388607 dan untuk yang tidak bertanda: 0 sampai dengan 16777215.

Tabel 2.5 Tipe data pada MySQL (Lanjutan)

Tipe Data	Deskripsi
Int	Ukuran 4 *byte*. Bilangan bulat dengan jangkauan untuk bilangan bertanda: -2147483648 sampai dengan 2147483647 dan untuk yang tidak bertanda: 0 sampai dengan 4294967295
Integer	Ukuran 4 *byte*. Sinonim dari Int.
BigInt	Ukuran 8 *byte*. Bilangan bulat terbesar dengan jangkauan untuk bilangan bertanda: -9223372036854775808 sampai dengan 9223372036854758087 dan untuk yang tidak bertanda: 0 sampai dengan 18446744073709551615.
Float	Ukuran 6 *byte*. Bilangan pecahan.
Double	Ukuran 8 *byte*. Bilangan pecahan.
Double Precision	Ukuran 8 *byte*. Bilangan pecahan berpresesi ganda.
Real	Ukuran 8 *byte*. Sinonim dari Double.
Decimal(m,d)	Ukuran M *byte*. Bilangan pecahan. Misalnya, Decimal(5,2) dapat digunakan untuk menyimpan bilangan -99,99 sampai dengan 99,99.
Timestamp	Sebuah timestamp otomatis
Date	Tipe data untuk menyimpan informasi tanggal dengan menggunakan "YYYY-MM-DD" sintaksis, tapi dapat diperbaharui dengan sebuah angka atau sebuah huruf.
Time	Tipe data untuk menyimpan informasi jam dengan menggunakan "HH:MM:SS" sintaksis, tapi dapat diperbaharui dengan sebuah angka atau sebuah huruf.
DateTime	Tipe data untuk menyimpan data tanggal dan jam dengan format "YYYY-MM-DD HH:MM:SS".
Numeric(m,d)	Ukuran M *byte*. Sinonim dari Decimal.
MediumBlob, MediumText	L+3 *byte*, dengan $L<2^{24}$. Tipe teks atau Blob dengan panjang maksimum 16777215 karakter.

Tabel 2.5 Tipe data pada MySQL (Lanjutan)

Tipe Data	Deskripsi
Varchar	Sebuah *variable* huruf yang dapat menyimpan 1-255 karakter.
LongBlob, LongText	L+4 *byte*, dengan $L<2^{32}$. Tipe Text atau Blob dengan panjang maksimum 4294967295 karakter
Enum('nilai1', 'nilai2',...)	Ukuran 1 atau 2 *byte* tergantung jumlah nilai enumerasinya (maksimum 65535 nilai)
Set('nilai1', 'nilai2',...)	1, 2, 3, 4, atau 8 *byte*, tergantung jumlah anggota himpunan (maksimum 64 anggota)

2.9 UML *(Unified Modelling Language)*

2.9.1 Sejarah UML

Sejarah UML sendiri cukup panjang. HIngga era tahun 1990, seperti telah diketahui, puluhan metodologi pemodelan berorientasi objek telah bermunculan di dunia. Masa itu terkenal dengan masa perang metodologi (*method war*) dalam pendesainan berorientasi objek. Masing-masing metodologi membawa notasi sendiri-sendiri, yang mengakibatkan timbul masalah baru apabila bekerjasama dengan group atau perusahaan lain yang menggunakan metodologi yang berlainan.

Metodologi yang paling banyak digunakan dalam OOAD adalah OMT *(Object modeling Technique)* dari Rumbaugh, OOAD *(Object Oriented Analysis/Design)* dari Shlaer-Mellor, Metode Booch, *Responsiblity-Driven Design/-Class/Responsibility Coad/Yourdon dan Jacobson* OOSE *(Object Oriented Software Engineering).*

2.9.2 Definisi UML

Unified modeling language adalah bahasa standar yang digunakan untuk menuliskan *blueprint* (cetak-biru) perangkat lunak. UML dapat digunakan untuk memvisualisasikan, menentukan, membangun, dan mendokumentasikan artifak dari sebuah sistem perangkat lunak intensif. Artifak adalah sepotong informasi yang digunakan atau dihasilkan dalam suatu proses rekayasa perangkat lunak.

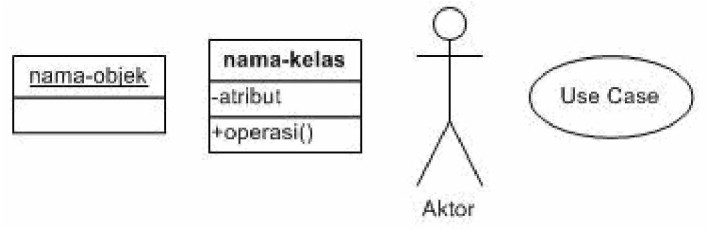
Dalam konteks ini, menentukan berarti membuat model yang tepat, tidak ambigu, dan lengkap. Biasanya, UML berfokus pada semua spesifikasi keputusan analisis, desain dan implementasi penting yang harus dibuat dalam membangun dan menerapkan sistem perangkat lunak.

UML merupakan bahasa yang sangat ekspresif, membuat semua *view* (tampilan) yang dibutuhkan dalam membangun sistem yang berskala besar. UML hanyalah sebuah bahasa dan juga hanya merupakan salah satu bagian dari metode pembangunan perangkat lunak. UML tidak tergantung pada proses, walaupun optimalnya, UML harus digunakan dalam sebuah proses yang bersifat *use-case driven*, *architecture-centric*, iteratif, dan inkremental.

Use-case driven berarti bahwa proses dalam *use case* digunakan sebagai artifak primer untuk membuat *behavior* yang diinginkan dalam sistem, memastikan dan memvalidasi arsitektur sistem, menguji, dan berkomunikasi antar pihak dalam proyek. *Architecture-centric* berarti bahwa proses yang berfokus pada pembangunan sebelumnya dan berdasar pada arsitektur perangkat lunak menggunakan arsitektur sistem sebagai artifak primer untuk mengonsep, membangun, mengatur, dan mengubah sistem yang sedang dibangun.

2.9.3 Notasi UML

UML Merupakan notasi-notasi baku untuk melakukan pemodelan visual. Pemodelan visual merupakan proses penggambaran informasi-informasi secara grafis. Notasi-notasi UML terbentuk atas kerjasama dan upaya Graddy Booch, DR. James Rumbaugh, serta Ivar Jacobson.



Gambar 2.4 Beberapa Notasi UML

Beberapa notasi UML yang sering digunakan antara lain ditunjukkan pada Gambar 2.4 Aktor merupakan entitas yang berada di luar sistem, bersifat eksternal. Salah satu aktor sistem adalah pengguna (*user*).

Use case berfokus pada perilaku sistem, misalnya *use case* "Pengaktifan *modem*". Kelas merupakan prototipe suatu objek sehingga memiliki atribut (variabel) dan operasi (fungsi atau prosedur).

Dalam UML terdapat beberapa jenis diagram yang dapat digunakan dalam pemodelan sistem, diantaranya adalah:

1. *Activity Diagram*. Bersifat dinamis, memodelkan fungsi-fungsi dalam suatu sistem dan menekankan pada aliran kendali antar objek.
2. *Use Case Diagram*. Bersifat statis, menggambarkan hubungan (*relation*) antara aktor dan *use case*.

3. *Sequence Diagram*. Bersifat dinamis, merupakan diagram interaksi yang menekankan pada pengiriman pesan (*message*) secara berurutan berdasarkan waktu.
4. *Class Diagram*. Bersifat statis, menggambarkan himpunan kelas-kelas, antarmuka dan relasi antar kelas.

2.10 Borland Delphi 7

Borland Delphi 7 adalah salah satu bahasa pemrograman yang bekerja dalam lingkup *Microsoft windows*. *Borland Delphi 7* dapat memanfaatkan kemampuan yang dimilikinya untuk dapat bekerja dalam lingkungan *windows* dengan baik dan. kemampuannya dapat dipakai untuk merancang program aplikasi yang berpenampilan seperti program aplikasi lainnya yang berbasis *Microsoft windows.*

Borland Delphi 7 dapat memamfaatkan fasilitas dan kecanggihan yang ada pada sistem operasi *Microsoft windows*. Kemampuan *Borland Delphi 7* ditunjang dengan ketersediaan komponen yang lengkap baik komponen yang umum maupun komponen khusus yang dapat di gunakan setelah diinstalasi sehingga membuat program aplikasi yang dibuat lebih menarik baik dari segi tampilan maupun kehandalanya terutama pada lingkungan system operasi *Microsoft windows.*

Komponen standar yang ada pada *Borland Delphi 7* adalah sebagai berikut:

1. *Project*

   Project adalah sekumpulan modul. Jadi *project* (proyek) adalah program aplikasi itu sendiri. *Project* disimpan dalam *file* berakhiran .DPR. *File* ini menyimpan seluruh komponen program, termasuk pilihan proyek, pilihan *Environment*, pilihan file EXE dan segala sesuatu yang berhubungan dengan proyek.

2. *Form Designer*

   Form Designer adalah suatu objek yang dipakai sebagai tempat bekerja program aplikasi. *form* berbentuk jendela dan dapat dibayangkan sebagai kertas atau meja kerja yang dapat dilukisi atau diletakkan ke dalamnya objek-objek lain.

3. *Object Inspector*

   Object Inspector adalah kotak alat yang berisi *icon-icon* untuk memasukkan objek tertentu ke dalam jendela *form*. Ketika anda mulai membuat suatu proyek , *Delphi 7* akan otomatis menyediakan *icon-icon* objek yang sering dipakai, sesuai kategori aplikasi yang anda buat, misalnya DLL dsb.

4. *Properties*

   Properties digunakan untuk menentukan konfigurasi suatu objek. Suatu objek biasanya mempunyai beberapa properti yang dapat diatur langsung dari jendela *properties* atau lewat kode program. Setting properti akan menentukan cara kerja dari objek yang bersangkutan saat program aplikasi dijalankan, misalnya menentukan warna objek, bingkai objek, pengambilan data dan lain-lain.

5. *Code Editor*

   Code Editor adalah serangkaian tulisan perintah yang akan dilaksanakan jika suatu objek dijalankan. Kode program ini akan mengontrol dan menentukan jalannya suatu objek.

6. *Event*

   Event adalah peristiwa atau kejadian yang diterima oleh suatu objek, misalnya klik kanan tunjuk dan lain-lain. *Event* yang diterima objek akan memicu *Borland Delphi 7* menjalankan kode program yang ada didalamnya.

7. *Metoda*

   Metoda adalah suatu set perintah seperti halnya fungsi dan prosedur, tetapi sudah tersedia didalam suatu objek. Seperti halnya *property* (yang juga terdapat pada suatu objek), suatu metoda dapat dipanggil dengan menyebut nama objek diikuti tanda titik dan nama metodanya.

8. *Module*

   Module dapat disejajarkan dengan *form*, tetapi tidak mengandung objek dan bentuk standar. *Module* dapat berisi beberapa kode program atau prosedur yang dapat digunakan dalam program aplikasi. Sebenarnya suatu form juga adalah sebuah *module* (disebut sebagai *Form Module*), tetapi mempunyai objek berupa *form*.