BAB II

TINJAUAN PUSTAKA

2.1 Sejarah Telekomunikasi

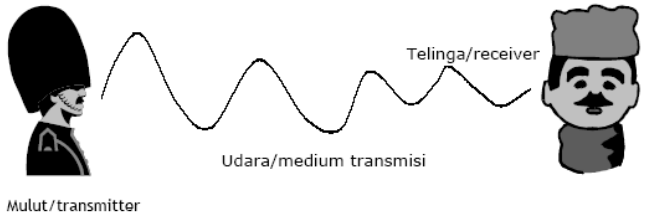
Telepon pertama kali ditemukan oleh Alexander Graham Bell pada tahun 1876 dan beliau memegang hak paten atas penemuannya tersebut. Sejak masa lalu sampai sekarang, nampak bahwa trafik telepon (suara) merupakan pengguna fasilitas komunikasi yang utama. Namun, sejak sekitar 25 tahun terakhir, penyediaan fasilitas pengiriman informasi yang bukan suara telah mengalami pertumbuhan yang pesat. Oleh karena itu perlu dikembangkan teknologi baru yang sesuai dengan munculnya layanan yang baru.

2.2 Konsep dasar komunikasi

Kata komunikasi berasal dari kata berbahasa Inggris *communications*. Kata *communications* berasal dari bahasa latin *communicare* yang berarti saling berbagi (*share*). Komunikasi dapat diartikan sebagai proses menampilkan, merubah, menginterpretasikan, atau mengolah informasi antara manusia atau mesin.

Proses ini melibatkan suatu pengirim (*transmitter*), penerima (*receiver*), dan sebuah medium transmisi untuk tempat mengalirnya informasi. Gambar 2.1 memperlihatkan bentuk komunikasi yang paling sederhana, yaitu perbincangan antara dua orang. Pada contoh ini, manusia bertukar informasi dengan cara berbicara. Suara yang berasal dari orang-orang tersebut akan merubah tekanan

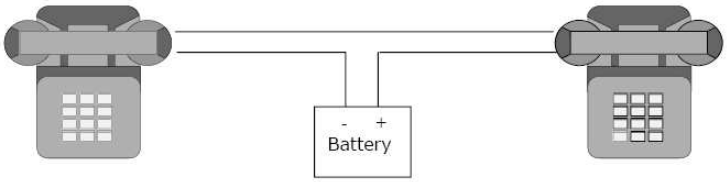
udara yang pada akhirnya ketika sampai di telinga pendengar akan menggetarkan gendang telinga si pendengar tersebut.



Gambar 2.1 Bentuk Dasar Komunikasi

Masalah yang timbul pada bentuk dasar komunikasi tersebut di atas adalah jarak. Jika jarak antara kedua orang yang sedang berbicara tersebut bertambah, informasi yang dibawa oleh udara akan kehilangan kekuatannya sebelum dapat mencapai tujuan. Komunikasi jarak jauh ini disebut telekomunikasi.

Bentuk telekomunikasi yang paling pertama muncul adalah telepon. Komunikasi lewat telepon menggunakan arus listrik yang dialirkan melalui kabel yang memungkinkan transfer informasi pada jarak yang berjauhan. Gambar 2.2 memperlihatkan suatu koneksi telepon yang sederhana.



Gambar 2.2 Koneksi Telepon yang Sederhana

Koneksi seperti yang diperlihatkan pada gambar 2.2 tidak mencukupi bila manusia yang akan berkomunikasi jumlahnya lebih dari satu pasang. Kebutuhan ini melahirkan apa yang disebut jaringan komunikasi. Jaringan komunikasi

menyediakan wahana untuk menghubungkan banyak orang secara efisien menggunakan *link* dan *node*.

2.3 Konsep Dasar Sistem

Suatu sistem adalah jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu.

Suatu sistem yang baik harus mempunyai tujuan dan sasaran yang tepat karena hal ini akan sangat menentukan dalam mendefinisikan masukan yang dibutuhkan sistem dan juga keluaran yang dihasilkan.

Terdapat dua kelompok pendekatan di dalam pendefinisian sistem, yaitu yang menekankan pada prosedur dan yang menekankan pada komponen atau elemen–elemen. Pendekatan sistem yang lebih menekankan pada prosedur mendefinisikan sistem sebagai berikut :

> *"Suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu"*. **[1]**

Pendekatan sistem yang lebih menekankan pada komponen atau elemen-elemennya mendefinisikan sistem sebagai berikut :

> *"Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu"*. **[1]**

2.4 Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat–sifat tertentu antara lain sebagai berikut :

1. Komponen-komponen Sistem (*Components*)

   Merupakan salah satu karakteristik sistem yang berupa sub sistem atau gagasan sistem.

2. Batas Sistem (*Boundary*)

   Merupakan daerah yang membatasi antara suatu sistem dengan sistem lainnya atau dengan lingkungan lainnya.

3. Lingkungan Luar Sistem (*Environment*)

   Segala sesuatu yang berada di luar batas dari sistem yang mempengaruhi operasi baik itu menguntungkan operasi sistem yang merupakan energi dari sistem yang harus dipelihara atau ditahan sehingga tidak mengganggu atau merusak sistem.

4. Penghubung Sistem (*Interface*)

   Merupakan media penghubung antara satu sub sistem dengan sub sistem lainnya.

5. Masukan Sistem (*Input*)

   Masukan sistem adalah data yang dimasukkan ke dalam sistem yang diproses untuk mendapat satu kesatuan atau informasi.

6. Keluaran Sistem (*Output*)

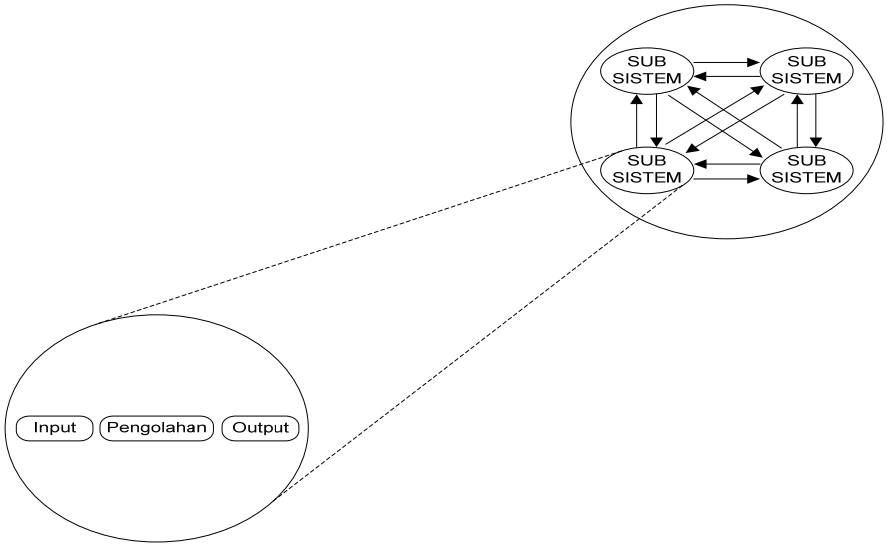
Keluaran dari sistem adalah data yang diolah serta diklasifikasikan menjadi keluaran yang berguna atau menjadi informasi yang dibutuhkan.

7. Pengolah Sistem (*Process*)

   Suatu sistem dapat mempunyai suatu bagian pengolahan yang akan merubah suatu masukan menjadi keluaran yang dibentuk.

8. Sasaran Sistem (*Goal*)

   Suatu sistem mempunyai maksud tertentu yaitu tujuan atas sasaran, dimana yang menentukan sekali masukan serta keluaran sistem yang mengena pada sasaran atau tujuan yang dimaksudkan.



Gambar 2.3 Karakteristik Sistem

2.5 Klasifikasi Sistem

Dari berbagai sudut pandang, sistem dapat diklasifikasikan menjadi beberapa bagian yaitu :

2 Sistem abstrak dan sistem fisik

Sistem abstrak merupakan sistem yang tidak bisa dilihat secara mata biasa dan biasanya sistem ini berupa pemikiran atau ide-ide. Contoh dari sistem abstrak ini adalah filsafat. Sistem fisik merupakan sistem yang bisa dilihat secara mata biasa dan biasanya sering digunakan oleh manusia. Contoh dari sistem fisik ini adalah sistem akuntansi, sistem komputer dan sebagainya.

3 Sistem alamiah dan sistem buatan

Sistem alamiah merupakan sistem yang terjadi karena pengaruh alam. Misalnya sistem rotasi bumi, sistem grafitasi dan sebagainya. Sistem buatan merupakan sistem yang dirancang dan dibuat oleh manusia. Misalnya sistem pengolahan gaji.

4 Sistem tertutup dan sistem terbuka

Sistem tertutup merupakan sistem yang tidak berhubungan dengan bagian luar sistem dan biasanya tidak terpengaruh oleh kondisi diluar sistem. Sedangkan sistem terbuka merupakan sistem yang berhubungan dengan bagian luar sistem.

2.6 Analisis Sistem

Untuk mencapai tujuan dari suatu sistem yang dibuat, dibutuhkan 3 perangkat atau alat bantu yang dapat meningkatkan kinerja dari sebuah sistem.

Tiga perangkat tersebut meliputi perangkat keras (*hardware*), perangkat lunak (*software*) dan perangkat manusia (*brainware*). Perangkat keras dapat berupa komputer, ponsel, sedangkan perangkat lunak adalah program. Perangkat manusia dapat berupa manager, analisis sistem, *programmer* dan sebagainya.

2.7 GSM (Global System for Mobile Communication)

Komunikasi sekarang ini tidak mengenal jarak dan waktu. Seseorang dapat melakukan komunikasi dari tempat yang berbeda tampa mengalami kesulitan. Ini semua karena adanya salah satu teknologi yang popular GMS (*Global System for Mobile Communication*) yaitu standardisasi untuk komunikasi mobile.

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.

2.8 SMS (Short Message Service)

SMS adalah sebuah layanan yang banyak diaplikasikan pada sistem komunikasi tampa kabel yang merupakan salah satu fasilitas dari GMS yang 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. Karakteristik utama SMS adalah 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.

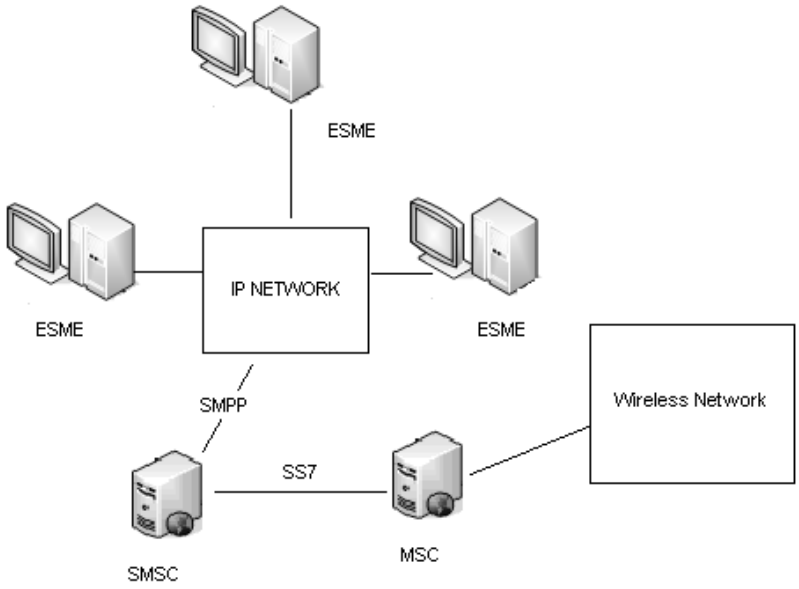
Dalam SMS mekanisme utama yang dilakukan dalam sistem adalah sistem melakukan pengiriman short message dari satu terminal pelanggan ke terminal yan lain. Hal ini dapat dilakukan berkat adanaya sebuah Entitas dalam sistem SMS yang bernama Short Message Service Center (SMSC), disebut juga Message

Center (MC). SMSC merupakan sebuah perangkat yang melakukan tugas *store and forward* trafik short message. Di dalamnya termasuk penentuan atau pencarian rute tujuan akhir dari short message. Sebuah SMSC biasanya didesain untuk dapat menangani short message dari berbagai sumber sperti Voice mail System(VMC), Web-based message, Email Integration, External Short Messaging Entities (ESME), dan lain-lain. Dalam interkoneksi dengan entitas dalam jaringan komunikasi wireless seperti Home location Register (HLR) dan mobile Switching Center (MSC), SMSC biasanya selalu menggunakan Signal Transfer Point (STP).

2.9 Arsitektur dan Elemen Jaringan SMS

2.9.1 Arsitektur Jaringan

Layanan SMS dibangun dari berbagai entitas yang saling terkait dan mempunyai fungsi dan tugas masing-masing. Tidak ada satu pun dalam sistem SMS yang dapat bekerja secara parsial. Entitas dalam jaringan SMS ini disebut juga elemen jaringan SMS. Secara umum yang dintegrasikan dengan jaringan wireless, adalah sebagai berikut :



Gambar 2.4 Arsitektur Dasar Jaringan SMS

2.9.2 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

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.

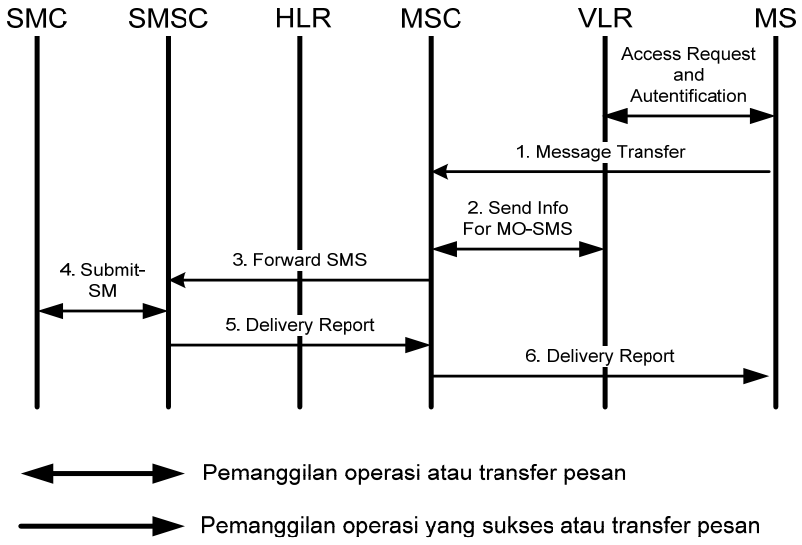
2.9.3 Elemen Layanan

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.



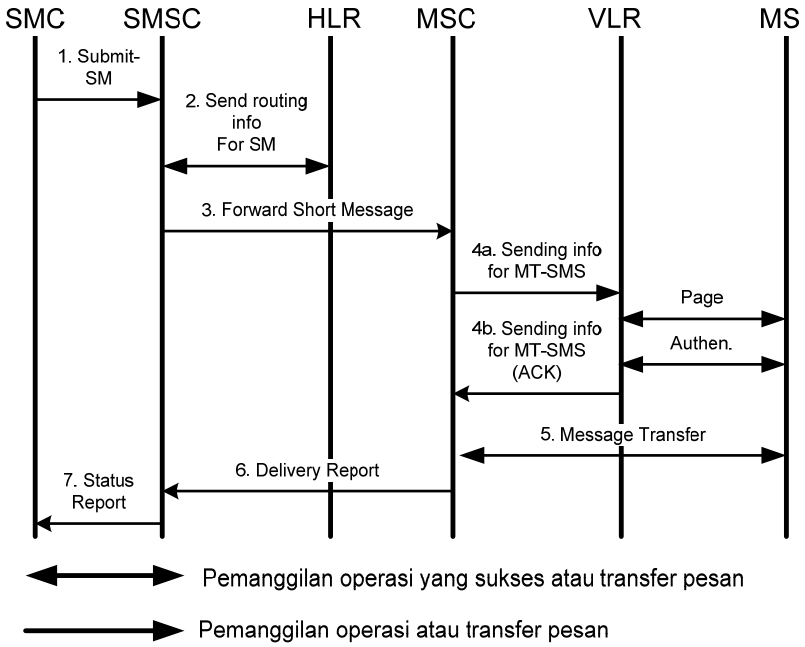
Gambar 2.5 Skenario pengiriman MO-SM

Keterangan :

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 2.6 Skenario pengiriman MT-SM

Keterangan :

a. Pesan pendek dikirimkan dari SME ke SMSC.
b. Setelah menyelesaikan pengolahan internalnya, SMSC menginterogasi HLR dan menerima informasi *routing* untuk pelanggan *mobile*.
c. SMSC mengirimkan pesan pendek ke MSC dengan menggunakan operasi *Forward Short Message.*.
d. MSC mengambil informasi pelanggan dari VLR. Operasi ini dapat melibatkan prosedur autentifikasi.

e. MSC mengirimkan pesan pendek ke MS.

f. MSC mengembalikan hasil dari operasi *Forward Short Message* ke SMSC.

g. Jika diminta oleh SME, SMSC akan mengembalikan laporan status yang mengindikaikan pengiriman pesan pendek.

2.9.4 Protocol Data Unit (PDU)

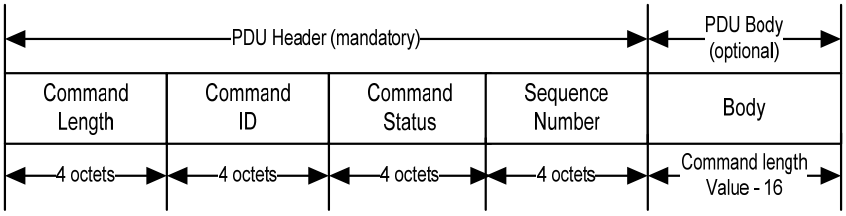
Satuan paket data yang dipertukarkan pada lapisan aplikasi dalam protokol SMS Manager disebut *Protocol Data Unit* (PDU). Dalam protokol tersebut terdapat beberapa macam format PDU dimana penggunaan masing-masing PDU terebut harus sesuai dengan fungsinya, sebagai contoh untuk mengirim sebuah pesan, harus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

2.9.5 Format Umum PDU

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



Gambar 2.7 Format umum PDU

Tabel 2.1 Format PDU

	SMPP PDU Field	**Size (Octets)**	**Type**	**Description**
PDU HEADER	*Command_length*	4	Integer	Overall size of PDU Including header and body
	Command_id	4	Integer	Identifies the PDU
	Command_status	4	Integer	Use to carry an error code
	Sequence_number	4	Integer	Use to uniquely identified PDU session
BODY	Standard Parameters	Var.	Mixed	The body part of PDU
	TLV Parameter	Var.	mixed	

Enam belas oktet pertama disebut PDU *header*, merupakan bagian yang bersifat mandatori atau harus selalu ada dalam setiap PDU. Sisanya disebut PDU *Body*, merupakan bagian yang bersifat opsional (tergantung jenis PDU). Body PDU terdiri dari parameter standar dan parameter tambahan yang disebut parameter *Tag-Length_Value* (TLV), berikut penjelasan masing-masing bagian tersebut :

1. ***command_Length***

   Command_length menyatakan panjang yang sesungguhnya dari PDU meliputi PDU Header dan PDU Body, termasuk didalamnya oktet dari *command_length* sendiri. Posisinya terletak pada awal atau field pertama dari setiap PDU.

2. ***command_id***

   command_id merupakan parameter yang mendgindikasikan jenis operasi dari protokol SMS Manager yang sedang dilakukan. Parameter ini dikodekan dalam 4 oktet dengan tipe data integer.Nilai parameter ini berkisar antara 0x00000000 hingga 0x00000001FF untuk parameter yang termasuk kategori request, kategori respons nilainya berkisar antara 0x80000000 hingga 0x800001FF, berikut contoh beberapa command_id :

Tabel 2.2 Contoh *command_id*

Command_id	**Nilai**
Bind_receiver	0x00000001
Bind_transmitter	0x00000002
Query_sm	0x00000003
Submit_sm	0x00000004

Deliver_sm	0x00000005
Unbind	0x00000006
Replace_sm	0x00000007
Cancel_sm	0x00000008
Bind_transceiver	0x00000009
Outbind	0x0000000B
Data_sm	0x00000103
Broadcast_sm	0x00000111
Bind_receiver_resp	0x80000001
Query_sm_resp	0x80000003
Submit_sm_resp	0x80000004
Deliver_sm_resp	0x80000005
Unbind_resp	0x80000006
Data_sm_resp	0x80000103
Broadcast_sm_resp	0x80000111
Query_broadcast_sm_resp	0x80000112

3. ***command_status***

   command_satus merupakan paramater yang menunjukkan status operasi dalam protokol dan hanya relevan untuk PDU-PDU kategori *respons*. Untuk PDU-PDU kategori *request*, parameter ini diabaikan.

4. ***sequence_number***

   sequence_number merupakan parameter yang digunakan untuk menandai sebuah PDU secara unik agar dapat dibedakan dengan PDU lain.

5. ***Parameter Standar***

Parameter standar merupakan parameter yang terdapat pada PDU *Body* pada PDU-PDU tertentu (PDU yang non-header only, yaitu PDU *submission* dan *delivery*). Formatnya sendiri bervariasi tergantung PDU-nya dan merupakan kombinasi data dengan tipe integer.

6. Parameter TLV

Tag-Length-Value, merupakan parameter dalam protokol yang dapat dibongkar-pasang sesuai dengan kebutuhan. Satu-satunya syarat hanyalah parameter ini, bila disertakan, harus diletakkan setelah paramater standar.

2.9.6 SMS PDU Pengirim (Mobile Originated)

SMS PDU Pengirim adalah pesan yang dikirim dari Handphone ke terminal yang kemudian dikirimkan ke SMSC. Pada prinsipnya apabila kita mengirimkan pesan ke nomor tujuan, pesan itu akan melalui SMSC.

Pesan yang dikirimkan oleh terminal masih dalam bentuk teks, sedangkan dalam pengiriman ke SMSC harus dalam bentuk PDU. Untuk itu sebelum dikirim, terminal atau handphone akan melakukan perubahan dari format teks menjadi format PDU. Proses ini sering disebut proses *encodec*. Adapun skema dari format PDU pengirim adalah diatur dan ditetapkan oleh ETSI sebagai berikut :

SCA	PDU Type	MR	DA	PID	DCS	VP	UDL	UD

Gambar 2.8 Skema Format SMS PDU pengirim

1. **SCA (Service Center Address**)

SCA adalah informasi dari alamat (nomor) SMSC. SCA memiliki tiga komponen utama yaitu *len, type of number, service center number.*

2. PDU Type

Nilai default dari PDU *type* untuk SMS pengirim adalah 11 hexa.

3. MR (Message Reference)

MR adalah acuan dari pengaturan pesan SMS. Untuk membiarkan pengaturan pesan SMS dilakukan sendiri oleh Handphone tujuan, maka nilai yang diberikan adalah " 00".

4. DA (Destination Address)

DA adalah alamat (nomor) tujuan, yang terdiri atas panjangnya nomor tujuan (*LEN*), format dari nomor tujuan (*type number)* dan nomor tujuan (D*estination Number*).

5. PID (Protocol Indentifier)

PID adalah tipe atau fprmat dari cara pengiriman pesan yang biasanya diatur dari handphone pengirim. Untuk pesan SMS yang berbentuk teks maka nilainya adalah "00".

6. DCS (Data Coding Scheme)

DCS adalah rencana dari pengkodean data untuk menentukan kelas dari pesan tersebut apakah berupa SMS teks standar, *Flash SMS* atau *Blinking SMS*. Apabila berupa teks standar maka nilainya "00".

7. VP (Validity Period)

VP adalah lama waktu pesan SMS disimpan di SMSC

8. UDL (User Data Length)

UDL adalah panjang pesan SMS yang akan dikirimkan dalam bentuk teks standar.

9. UD (User data)

UD adalah isi pesan yang akan dikirimkan dalam format heksadesimal.

Contoh :

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

Keterangan:

Tabel 2.3 Keterangan format PDU Pengirim

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 (SMS SUBMIT)
00	TP-Message-Reference. Diisi "00" agar diisi otomatis oleh handphone.

0C	Panjang digit dari nomor penerima (0C hex = 12 desimal)
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'

2.9.7 SMS PDU Penerima (Mobile Terminated)

SMS PDU Penerima adalah terminal menerima pesan yang datang atau masuk dari SMSC ke Handphone dalam format PDU. Pada prinsipnya pesan yan gkita terima dari SMSC masih dalam format PDU setelah itu terminal handphone yang menerima pesan akan melakukan pengkodekan menjadi teks, proses ini sering disebut proses *decodec*. Cara pengkodean format PDU sudah diatur dan distandarkan oleh ETSI. Format PDU dari SMS penerima adalah :

SCA	**PDU Type**	**OA**	**PID**	**DCS**	**VP**	**SCTS**	**UDL**	**UD**

Gambar 2.9 Skema Format SMS PDU penerima

1. OA (Originator Address)

OA adalah alamat (nomor) dari pengirim, yang terdiri atas panjangnya nomor pengirim (*Len)*, format dari nomor pengirim (*type Number)*, dan nomor pengirim (Originator Number).

2. SCTS adalah (Service Center Time Stamp)

SCTS adalah waktu dari penerimaan pesan oleh SMSC penerimaan. SCTS terdiri atas tahun, bulan, tanggal, jam, menit, dan detik, serta zona waktu. Contoh :

07 91 2658050000F0 04 0C 91 265836164900 00 00 506020 31133180 04 C830FB0D

Dengan keterangan sebagai berikut:

Tabel 2.4 Keterangan format PDU Penerima

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)
04	Oktet pertama untuk pesan SMS yang diterima
0B	Panjang digit dari nomor pengirim (0C hex = 12 desimal)
91	Jenis nomor pengirim (sama dengan jenis nomor SMSC)
265836164900	Nomor pengirim SMS, yang jika diterjemahkan adalah +628563619400
00	Pengenal protokol, dalam hal ini adalah 0
00	Skema pengkodean SMS, juga bernilai 0
506020 311331 80	Waktu pengiriman, yang berarti 05-06-02 (2 Juni 2005), dan jam 13:31:13. Sedangkan 80 adalah Timezone yang digunakan.
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'

2.10 Bahasa pemograman

2.10.1 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. Pada tabel 2.5 diperlihatkan beberapa jenis perintah Hayes yang berhubungan dengan penanganan pesan-pesan AT-Command GSM.

Tabel 2.5 Contoh Perintah AT-Command GSM

AT-Command	**Singkatan**	**Fungsi**
ATE1	*Activate Command*	Mengaktivasi At-Command
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	*Phone Book Selection*	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
AT+CMGD	*Message Get Deleted*	Menghapus *inbox* dan *outbox* SMS
AT+CGMI	*Get Manufactured Identification*	Menampilkan merk *handphone*
AT-CGMM	*Get Mobile Model*	Menampilkan ID model Handphone
AT+CGSN	*Get Serial Number*	Menampilkan serial nomor IMEI
AT+CPWD	*Change Password*	Merubah *password*
AT+CLIP	*List Incoming Phone*	Menampilkan nomor telepon masuk
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+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+CLCC	*List Curent Call*	Menampilkan penelepon terakhir
AT+CGMR	*GSM Mobile Version*	Menampilkan versi ponsel GSM
AT+CHLD	*Call Hold and Multiparty*	Menahan telepon masuk
AT+CPAS	*Query the telephone status*	Status telepon
AT+CPIN	*Enter PIN and query lock*	Memasukan PIN PUK atau SIM
AT+CHUP	*Terminate Call*	Untuk mengakhiri telepon masuk
AT+CEER	*Query the reason for disconnection of last call*	Menanyakan alasan pemutusan telepon terakhir.
AT+CREG	*Network Registration*	Registrasi Jaringan
AT+CCFC	*Call Forwarding*	Menampilkan telepon masuk
AT+CMEE	*Expanded Error Messages*	Menampilkan Pesan Error pada telepon
AT+CSCS	*Select TE character set*	Memilih set karakter TE mana yang dipakai.
AT+CSSN	*Supplementary service notifications*	Pemberitahuan penambahan service
AT+CRSM	*Restricted SIM Acces*	Membatasi akses SIM
AT+CACM	*Accumulated call meter*	Menghitung banyaknya panggilan

AT+CAMM	*Accumulated call meter maximum*	Menghitung maksimum banyaknya panggilan
AT+COPN	*Read operator names*	Membaca nama operator
AT+CPUC	*Price per unit and currency table*	Harga per unit dan skema mata uang.

2.11 E-MAIL

Email adalah singkatan dari Electronic Mail. Pesan, atau surat secara elektronik, baik berupa teks maupun gabungan dengan gambar, yang dikirimkan dari satu alamat ke alamat lain di jaringan internet.Sebuah alamat e-mail yang mewakili banyak alamat email sekaligus disebut sebagai mailing list. Sebuah alamat email biasanya memiliki format semacam username@host.domain, misalnya: namasaya@yahoo.com. Email ada 2 jenis yakni : Email berbasis web dan POP3 Pengertian : Email berbasis web adalah email yang dapat digunakan pada saat kita sedang on-line, contohnya : yahoo, plasa, hotmail, dll. Sedangkan email berbasis POP3 adalah email yang dapat kita gunakan pada saat kita sedang off-line, contohnya :email lokal smansa. Perbedaan :

Email berbasis web :

a. Hanya dapat digunakan pada saat on-line

b. Jangkauan seluruh dunia

Email berbasis POP3 :

a. Dapat digunakan meskipun dalam keadaan off-line

b. Jangkauannya hanya berkisar pada suatu area tertentu

ISTILAH ISTILAH DALAM EMAIL

1. Address : Alamat email yang dimiliki
2. Address book atau contacts : Istilah untuk daftar nama dan alamat email pada sebuah program email client
3. Attachment : istilah untuk melampirkan dokumen-dokumen tertentu yang akan dibawa pada saat email itu dikirim
4. BCC : system email yang memungkinkan penggunanya melakukan pengiriman email yang sama ke banyak alamat
5. Bounce email : Email yang tidak terkirim ke penerimanya dan masuk ke mailbox anda
6. cc : melakukan pengiriman email yang sama ke banyak alamat. Beda dengan bcc, cc in tidak menyembunyikan alamat penerima lainnya.
7. Delivery Receipt : Pengirim menerima info bahwa penerima email telah mendowload email yang dikirim dan telah disimpan di mail clientnya
8. Domain : Penamaan dalam dunia internet
9. Draft : Tulisan yang belum dikirim
10. Filters : Penyaring email yang masuk. Seperti email porno, penipuan, promosi
11. Folder : Fartisi tempat penyimpanan email yang diterima
12. Forward : Melakukan penerusan sebuah email ke penerima lain
13. Header : Kepala email, berisi informasi penting mengenai server apa, jam berapa, darimana email itu, dll.
14. IMAP : Metode pengaturan email yang anda terima

15. Inbox : Kotak surat
16. Junk mail : Email sampah, email yang dianggap tidak perlu bahkan menggangu sipenerima
17. Mailbox : Untuk menyatakan sebuah account email dengan kafasitas penampungan tertentu
18. Mail Client : Program penarik dan pengirim email
19. Mail Server: Perangkat server atau pc untuk melayani dan proses pengiriman dan penerimaan email
20. Mail Transper Agent (MTA) : Sistem yang bertugas meneruskan dan menghantarkan pesan email
21. Outbox : Tempat penampungan sementara email yang mau dikirim
22. Outlook : Nama program mail client yang ada pada operating system windows
23. Plain-text mail : Email yang hanya teks saja
24. POP3 : Protokol yang digunakan untuk proses penerimaan email
25. Preview Pane: Halaman yang digunakan untuk membaca email tanpa harus membuka satu halaman baru
26. Sent items : Nama dari sebuah folder pada program mail client
27. Signature : Identitas pengirim yang harus selalu dituliskan di akhir pesan email
28. SMTP : Sebagai protocol yang mengatur pengiriman pesan email
29. Vacation messege: Untuk menginformasikan kepada pengirim email bahwa anda sedang tidak bisa membaca email

30. Web based Email : Halaman web khusus untuk melayani anda untuk mengirim dan menerima email. WBE ini Milik yahoo, Gmail dll

2.12 Perangkat Lunak

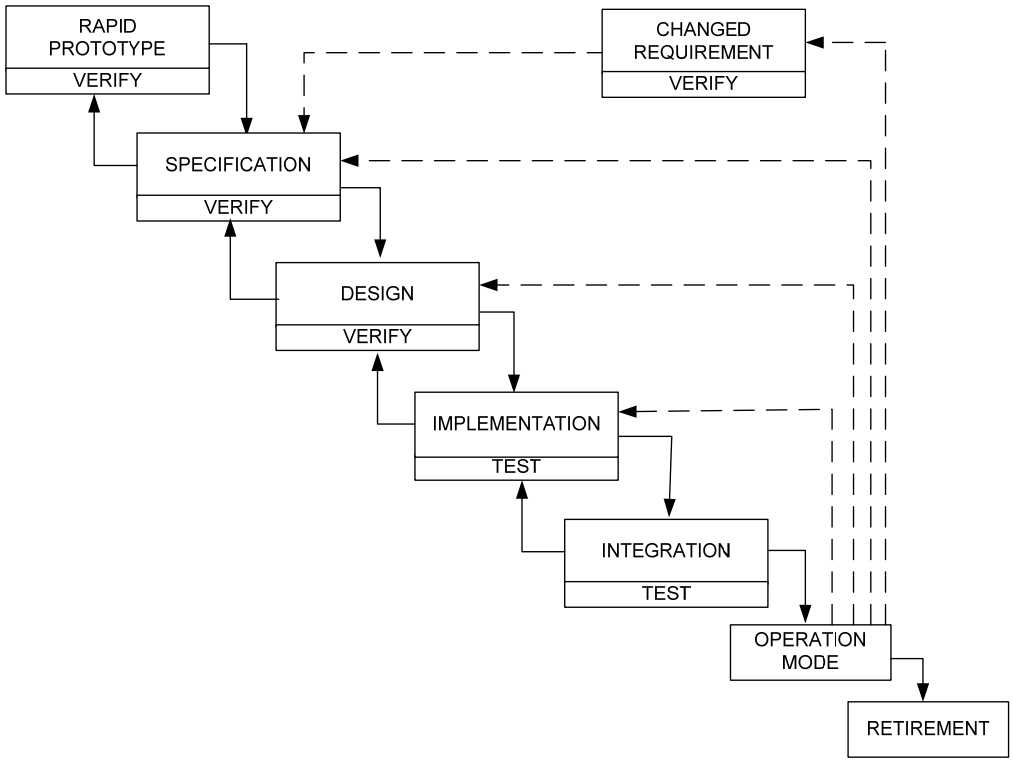
Perangkat Lunak adalah instruksi (program komputer) yang ketika dijalankan menyediakan fungsi dan tampilan yang diinginkan, struktur data yang memberikan kesempatan program untuk memanipulasi informasi dan dokumen yang mendeskripsikan operasi dan penggunaan program

2.12.1 Model Proses Perangkat Lunak

Terdapat beberapa model proses rekayasa perangkat lunak, diantaranya :

1. **Model Waterfall**

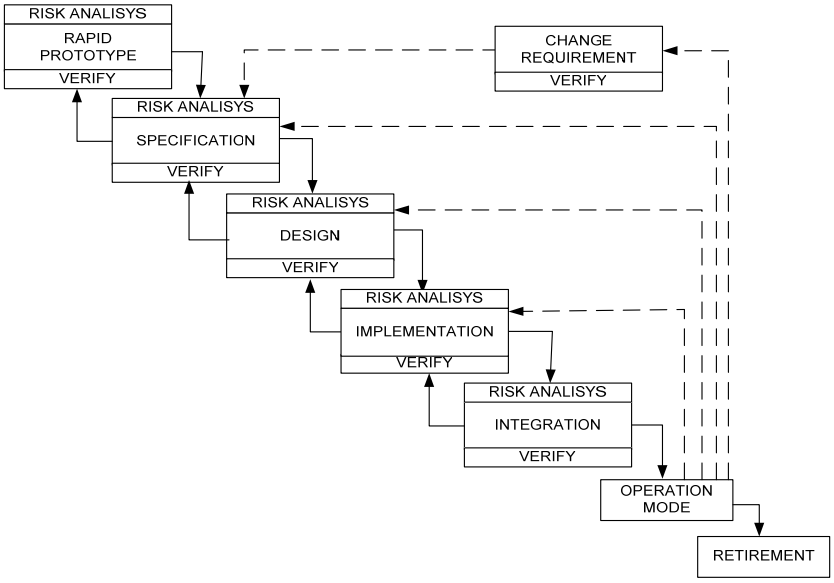
   Model Waterfall berisi rangkaian aktivitas proses seperti spesifikasi kebutuhan, implementasi desain perangkat lunak, uji coba dan sebagainya diuraikan dan disajikan dalam proses yang terpisah. Setelah setiap langkah didefinisikan, langkah tersebut dihentikan, dan pengembangan dilanjutkan pada langkah berikutnya. Gambar ilustrasi dari Model Waterfall adalah sebagai berikut :



Gambar 2.8 Model *Waterfall*

2. Model *Spiral* Boehm

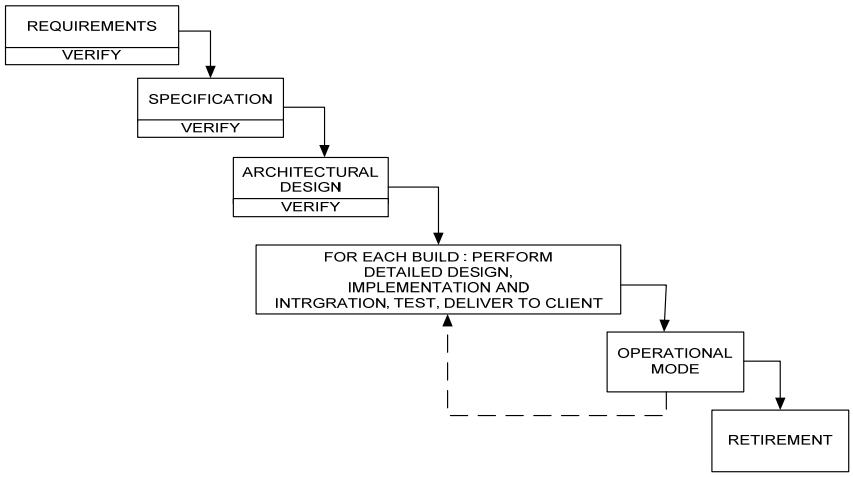
Model Spiral Boehm merupakan kombinasi antara *prototyping* model dan waterfall model. Setiap tahapan model ini selalu dilakukan *Risk Analisys* dan verifikasi atau testing. Gambar ilustrasi model Spiral Boehm adalah sebagai berikut :



Gambar 2.9 Model *Spiral* Boehm

3. Model Incremental

Model Incremental merupakan rekayasa perangkat lunak pembagian, hingga menghasilkan perangkat lunak yang lengkap. Proses membangun berhenti jika produk telah mencapai fungsi yang diharapkan. Gambar ilustrasi model Incremental adalah sebagai berikut :



Gambar 2.9 Model Incremental

2.13 Alat Pengembangan Sistem

Alat pengembangan sistem adalah unsur yang sangat penting dalam melakukan langkah–langkah metodologi pengembangan sistem yang terstruktur. Alat–alat yang digunakan umumnya berupa gambar, diagram atau grafik serta kamus data .

2.13.1 UML (Unified Modeling Language)

Unified Modelling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UMLmenawarkan sebuah standar untuk merancang model sebuah sistem.

Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasabahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB, Delphi atau C.

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax*/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut

dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (Object-Oriented Design), Jim Rumbaugh OMT (Object Modeling Technique), dan Ivar Jacobson OOSE (Object-Oriented Software Engineering).

2.13.2 Konsepsi Dasar UML

Konsepsi dasar UML digambarkan seperti dibawah ini :

Tabel 2.6 Konsep Dasar UML

Area Utama	Tampilan	Diagram	Konsep Utama
struktural	Tampilan Statis	*Class* diagram	*Class*, association, *gerenalisasi*, *dependensi*, *realization*, *interface*
	Tampilan *Use Case*	*Use case* diagram	*Use case*, aktor, *association*, *extend*, *include*, *use case* generalisasi
	Tampilan implementasi Tampilan *deployment*	*Component* diagram	Komponen, tampilan muka, depedensi, *realization*
Dinamis	Tampilan mesin state	*Deployment* diagram	*State*, *event*, transisi, *action*
		Statechart diagram	*State*, *activity*, penyelesaian transisi, *fork*, *join*
	Tampilan aktifitas	*Activity* diagram	*Interaction*, objek,

			pesan, aktifasi
		Collaboration diagram	Kolaborasi, berinteraksi, pesan
Manajemen model	Tampilan Model manajemen	*Class* diagram	*Package*, bagian *system*, model
extendsibility	semua	semua	*Constrain*, *stereo type*, nilai.

Abstraksi konsep dasar UML yang terdiri dari *structural classification*, *dynamic behavior*, dan *model management*, bisa kita pahami dengan mudah apabila kita melihat gambar diatas dari *Diagrams*. *Main concepts* bisa kita pandang sebagai term yang akan muncul pada saat kita membuat diagram.

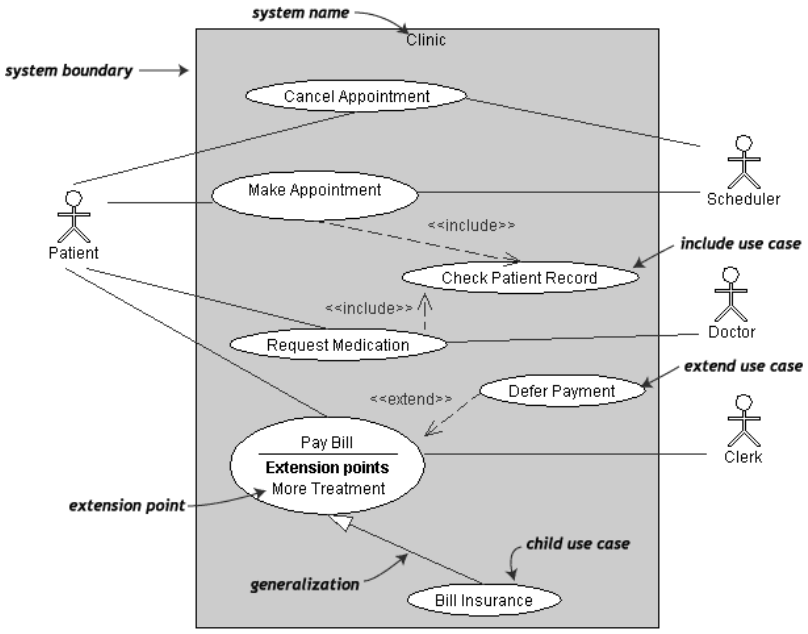
UML mendefinisikan diagram-diagram sebagai berikut:

1. *Use Case Diagram*

   Use case diagram adalah deskripsi dari sebuah sistem dari deskripsikan tipikal interaksi antara pengguna (penguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem disebut skenario.

   Dalam pembicaraan tentang *use case*, pengguna biasanya disebut dengan *actor*. *Actor* adalah sebuah peran yang bisa dimainkan oleh pengguna dalam interaksinya dengan sistem.

   Contoh *use case diagram* :



Gambar 2.10 Use Case Diagram

2. Class Diagram

Class Diagram adalah sebuah spesifikasi yang jika diintansi akan menghasilak sebuah objek dan merupakan inti dari pengembangan denagn desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tiga area pokok :

a. Nama

b. Atribut

c. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :

a. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan

b. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya

c. *Public*, dapat dipanggil oleh siapa saja

Akun
+ Notes: string + order: ordered + orderBalance: currency + Orderstatus: string
+ Getitembalance():currency + getorderID(): orderID

Class dapat merupakan implementasi dari sebuah *interface*, yaitu *class* abstrak yang hanya memiliki metoda. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class*. Dengan demikian *interface* mendukung resolusi metoda pada saat *run-time*.

3. *statechart diagram*

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

4. *Activity Diagram*

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

   Activity diagram merupakan *state diagram* khusus, dimana sebagian besar *state* adalah aktion dan sebagian besar transisi di *trigger* oleh selesainya *state* sebelum *(internal processing).* Oleh karena itu *activity diagram* tidak menggambarkan *behavior internal* sebuah sistem (dan interaksi antar

subsistem) secara eksak, tetapi lebih menggambarkan prose-proses dan jalur-jalur aktifitas dari level atas secara umum.

5. *Sequence Diagram*

   Sequence diagram menggambarkan interaksi antar objek didalam dan disekitar sistem (termasuk pengguna, *display* dan sebagainya) berupa *message* terhadap waktu. Sequence diagram terdiri atas dimensi vertikal (waktu) dan dimensi hotizontal (objek-objek) yang terkait.

   Sequence diagram bisa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah entitas *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang mentrigger aktifitas tersebut, proses dan perubahan apa saja yang terjadi secara *internal* dan *output* apa yang dihasilkan.

6. *Collaboration Diagram*

   Collaboration diagram juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message* setiap *message* memiliki *sequence number*, dimana message dari level tertinggi memiliki nomor satu. *Message* dari level yang sama memiliki prefix yang sama.

7. *Component Diagram*

   Component diagram menggambarkan struktur dan hubungan antar piranti lunak, termasuk ketergantungan *(dependency)* diantaranya. Komponen piranti lunak adalah modul berisi *code*, baik berisi *souce code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time*,

link time maupun *run time*. Umumnya komponen terbentuk dari beberapa *class* atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil.

8. *Deployment Diagram*

   Deployment diagram menggambarkan detail bagaimana komponen di *deploy* dalam infrastruktur sistem dimana komponen akan terletak (pada mesin, server atau piranti keras), bagaiman kemampuan jaringan pada lokasi tersebut, seperti *server* dan lain-lain yang bersifat fisikal.

2.14 Perangkat Lunak Pendukung

2.14.1 Borland Delphi 7

Borland Delphi 7 adalah paket bahasa pemrograman yang bekerja dalam sistem operasi windows. Delphi merupakan bahasa pemrograman yang mempunyai cakupan kemampuan yang luas dan sangat canggih. Berbagai jenis aplikasi dapat dibuat dengan Delphi, termasuk aplikasi untuk mengolah teks, grafik, angka, database dan aplikasi web. Secara umum, kemampuan Delphi adalah menyediakan komponen-komponen dan bahasa pemrograman yang andal, sehingga memungkinkan untuk dapat membuat program aplikasi sesuai dengan keinginan, dengan tampilan dan kemampuan yang canggih.

Untuk mempermudah *programmer* dalam membuat program aplikasi, Delphi menyediakan fasilitas pemrograman yang sangat lengkap. Fasilitas pemrograman tersebut dibagi dalam 2 kelompok, yaitu *object* dan bahasa pemrograman. Secara ringkas, *object* adalah suatu komponen yang

mempunyai bentuk fisik dan biasanya dapaat dilihat (visual). *Object* biasanya dipakai untuk melakukan tugas tertentu dan mempunyai batasan-batasan tertentu. Sedangkan bahasa pemrograman secara singkat dapat disebut sebagai sekumpulan teks yang mempunyai arti tertentu dan disusun dengan aturan tertentu serta untuk menjalankan tugas tertentu. Delphi menggunakan struktur bahasa pemrograman *Object* Pascal yang sudah sangat dikenal di kalangan pemrogram profesional. Gabungan dari *object* dan bahasa pemrograman ini sering disebut sebagai bahasa pemrograman berorientasi *object* atau *Object Oriented Programing* (OOP).

Khusus untuk pemrograman database, Delphi menyediakan *object* yang sangat kuat, canggih dan lengkap, sehingga memudahkan pemrogram dalam merancang, membuat dan meyelesaikan aplikasi database yang diinginkan. Selain itu Delphi juga dapat menangani data dalam berbagai format database, misalnya format MS-Acces, SyBase, Oracle, FoxPro, Informix, DB2 dan lain-lain.

2.14.2 Komponen Pendukung Borland Delphi 7

a. DSPACK
   DSPACK merupakan komponen tambahan Delphi untuk mengakses kamera
b. ZEOS
   ZEOS merupakan komponen tambahan Delphi sebagai konektor antara database MYSQL dan Delphi 7
c. ChilkatMail_v7_9.dll

ChilkatMail_v7_9.dll merupakan komponen tambahan Delphi untuk pengiriman email ke server SMTP.

d. Cport

Cport merupakan komponen tambahan Delphi untuk memilih port yang digunakan untuk modem GSM.

2.14.3 MySQL

MySQL adalah sebuah program *database server* yang mampu menerima dn mengirimkan datannya dengan sangat cepat, multi user serta menggunakan perintah standar SQL (Structured Query Language).

MySQL memiliki dua bentuk lisensi, yaitu FreeSoftware dan Shareware. MySQL yang biasa kita gunakan adalah MySQL *Free Software* yang berada di bawah lisensi GNU/GPL (General Public Lisence). Keterangan lengkap dapat dilihat pada http://www.gnu.org/lisences/. Selain itu kita juga dapat memilki produk MySQL yang sifatnya komersial, biasa disebut dengan MySQL AB.

MySQL merupakan sebuah *database serv*er yang *free*, artinya kita bebas menggunakan database ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensinya. MySQL pertama kali dirintis oleh seorang *programmer database* bernama *Michael Widenius*. Selain sebagai *database server*, MySQL yang berposisi sebagai *server*. Pada saat itu berarti program kita berposisi sebagai *client.* Jadi MySQL adalah sebuah *database* yang dapat digunakan baik sebagai client ataupun client sever.

MySQL adalah sebuah *database server,* dapat juga berperan sebagai *client* sehingga sering disebut *database client/server*, yang *open source*

dengan kemampuan dapat berjalan baik di OS *(Operating Sistem)* manapun, dengan Platform windows maupun linux.

Selain itu database ini memiliki beberapa kelebihan dibanding database lain, diantaranya :

1. MySQL sebagai *Database Managemet System (DBMS)*.
2. MySQL sebagai *Relation Database Management System (RDBMS)*.
3. MySQL adalah sebuah *software database yang open source*, artinya program ini dapat bersifat *free* atau bebas digunakan siapa saja tanpa harus membeli dan membayar lisensi kepada pembuatnya.
4. MySQL adalah sebuah *database server*, jadi dengan menggunakan database ini anda dapat menghubungkannya ke media internet sehingga dapat diakses dari jauh.
5. MySQL merupakan sebuah *database client*, selain menjadi server yang melayani permintaan, MySQL juga dapat melakukan query yang mengakses database pada server. Jadi MySQL dapat juga berperan sebagai client.
6. MySQL mampu menerima query yang bertumpuk dalam satu permintaan atau yang disebut Multi-Threading.
7. MySQL merupakan sebuah database yang mampu menyimpan data berkapasitas sangat besar hingga berukuran Gigabyte sekalipun.
8. MySQL didukung oleh driver ODBC, artinya database MySQL dapat diakses menggunakan aplikasi apa saja termasuk berupa visual seperti Delphi ataupun Visual Basic.

9. MySQL didukung oleh sebuah komponen C dan perl API, sehingga database MySQL dapat diakses melalui sebuah program aplikasi yang berada dibawah *protokol internet* berupa web. Biasanya aplikasi yang sering digunakan adalah php dan perl.
10. MySQL menggunakan suatu bahasa permintaan standar yang bernama SQL *(Structure Query Language)* yaitu sebuah bahasa permintaan yang distandarkan pada beberapa database server seperti Oracle,PostGreSQl dan SQL Server.

2.14.4 SQL (Structure Query Language)

SQL adalah sebuah bahasa permintaan *database* yang terstruktur. Bahasa SQL dibuat sebagai bahasa yang dapat merelasikan beberapa tabel dalam database maupun merealisasikan antar database. Bahasa SQL ditulis langsung dalam sebuah *program database* sehingga seorang pengguna dapat melihat langsung permintaan yang diinginkan, sekaligus melihat hasilnya. MySQL sendiri adalah sebuah *program database server* yang memerlukan sebuah bahasa permintaan dalam melayani permintaan user.

SQL dibagi menjadi tiga bentuk query, yaitu :

2.14.4.1 DDL (Data Definition Language)

DDL adalah sebuah metode query SQL yang berupa untuk mendefinisikan data pada sebuah database, adapun query yang dimiliki adalah*Create* adalah digunakan untuk melakukan pembuatan tabel dan database. *Drop* adalah digunakan untuk melakukan penghapusan tabel maupun database. *Alter* adalah digunakan untuk

pengubahan struktur tabel yang telah dibuat, baik menambah Field *(add)*, menganti nama Field *(change)* ataupun menamakannya kembali *(rename)*, serta menghapus *(drop).*

2.14.4.2 DML (Data Manipulation Language)

DML adalah sebuah metode *query* yang dapat digunakan apabila DDL telah terjadi, sehingga fungsi dan *query* ini adalah untuk melakukan pemanipulasian database yang telah ada atau telah dibuat sebelumnya. Adapun *query* yang termasuk di dalamnya adalah :

1. *Insert* adalah digunakan untuk melakukan penginputan / pemasukan data pada tabel database.
2. *Update* adalah digunakan untuk melakukan pengubahan atau peremajaan terhadap data yang ada pada tabel.
3. *Delete* adalah digunakan untuk melakukan penghapusan data pada tabel. Penghapusan ini dapat dilakukan secara sekaligus (seluruh isi tabel) maupun hanya beberapa *recordset.*

2.14.4.3 DCL (Data Control Language)

Data Control Language terdiri atas sekelompok perintah SQL untuk memberikan hak otorisasi mengakses database, mengalokasikan space, pendefinisian space, dan pengauditan penggunaan database. Beberapa dari perintah ini merupakan fungsi dari DBA (Database Administrator). Perintah SQL DCL ini meliputi :Commit, Rollback, Grant, dan Revoke. Secara umum DCL merupakan bahasa yang digunakan untuk melakukan pengelolaan pemakai yang dapat

melakukan akses dan manipulasi database, terutama perintah GRANT dan REVOKE. Perintah COMMIT dan ROLLBACK merupakan kelengkapan fasilitas dalam pembuatan aplikasi yang memungkinkan suatu transaksi yang terjadi untuk dapat segera disimpan atau dibatalkan transaksinya.