BAB II

LANDASAN TEORI

Pada bab ini akan dikemukakan teori-teori yang ada hubungannya dengan pembahasan dalam tugas akhir. Teori-teori ini nantinya akan membantu dalam memecahkan masalah yang ada.

2.1 Antrian

2.1.1 Pengertian Antrian

Proses antrian merupakan suatu proses yang berhubungan dengan kedatangan seorang pelanggan pada suatu fasilitas pelayanan, kemudian menunggu dalam suatu baris (antrian), dan akhirnya meninggalkan fasilitas tersebut. Sistem antrian merupakan suatu himpunan pelanggan, pelayan, dan suatu aturan yang mengatur kedatangan para pelanggan dan pemrosesan masalahnya.

2.2 Administrasi

2.2.1 Pengertian Administrasi

Administrasi adalah usaha dan kegiatan yang berkenaan dengan penyelenggaraan kebijaksanaan untuk mencapai tujuan. Administrasi dalam arti sempit adalah kegiatan yang meliputi: catat-mencatat, surat-menyurat, pembukuan ringan, ketik-mengetik, agenda, dan sebagainya yang bersifat teknis ketatausahaan.Administrasi dalam arti luas adalah seluruh proses kerja sama antara dua orang atau lebih dalam mencapai tujuan dengan memanfaatkan sarana prasarana tertentu secara berdaya guna dan berhasil guna.

2.3 Pengertian Sistem Informasi

2.3.1 Pengertian Sistem

"Sistem merupakan kumpulan elemen yang saling berkaitan yang bertanggung jawab memproses masukan (input) sehingga menghasilkan keluaran(output)."(kusrini 2007:10)

Menurut moekijat (2005:71) "sistem adalah kumpulan bagian-bagian yang saling mempengaruhi dan berkerjasama untuk mencapai suatu tujuan ".

Sistem dibagi menjadi tiga berbeda yaitu input, proses dan output.bagian-bagian tersebut dikelilingi oleh sebuah lingkungan dan sering melibatkan sebuah mekanisme umpan balik.

a. input adalah elemen yang masuk kedalam sistem. Contoh input adalah bahan mentah yang dimasukan seperti pabrik kimia dan input data ke dalam halaman Web untuk query database.
b. Proses adalah semua elemen yang diperlukan untuk mengonversi atau mengtransformasi input ke dalam output .Contoh sebuah proses pabrik kimia dapat dimasukan pemanasan material, penggunaan prosedur penelitian, penggunaan karyawan dan mesin
c. Output adalah produk kosekuensi yang ada pada sistem. Sebagai contoh *fertilizer* adalah output dari sebuah pabrik kimia, orang yang berpendidikan adalah orang output dari sebuah sistem komputer.
d. Umpan balik adalah aliran informasi dari komponen output ke pengambil keputusan berkenaan dengan output atau performa sistem. Berdasarkan output, pengambilan keputusan yang bertindak sebagai

kontrol, dapat memutuskan untuk memodifikasi input, proses atau keduanya.

Dari definisi diatas dapat diambil suatu kesimpulan bahwa sistem adalah kumpulan dari berbagai elemen yang terkait yang saling mempengaruhi dan bekerja sama untuk mencapai suatu tujuan.

2.3.2 Pengertian Informasi

Informasi adalah data yang telah di proses menjadi bentuk yang memiliki arti bagi penerima dan dapat berupa fakta dan menjadi suatu nilai yang bermanfaat adasuatu proses transformasi data menjadi suatu informasi sama dengan input,proses,dan output yang mengolah data yang merupakan data mentah (raw material)untuk kemudian menjadi suatu informasi.

Menurur Hasta Dewa Putranta (2004:22) “informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berati bagi penerimanya

Dari definisi diatas informasi dapat diartikan sebagai pengetahuan yang didapat dari sebuah data yang diolah menjadi bentuk yang lebih berguna bagi penerimanya,yang mennggambarkan suatu kejadian nyata dan digunakan untuk mengambil keputusan.

2.4 Website

2.4.1 Pengertian Web

Website adalah sebuah penyebarah informasi melalui internet yang interaktif dan digunakan pada suatu jaringan komputer. *Web* mempunyai suatu metode untuk menampilkan informasi diinternet, baik berupa text, gambar, suara maupun video yang interaktif dan mempunyai kelebihan untuk menghubungkan

(link) satu dokumen dengan dokumen lain (*hypertest*) yang dapat diakses melalui sebuah *browser.*

2.4.2 Jenis Website

Secara garis besar,website bisa digolongkan menjadi 3 bagian yaitu:

- Website statis

  Website statis adalah web yang mempunyai halaman tidak berubah. Artinya adalah untuk melakukan perubahan pada suatu halaman dilakukan secara manual dengan mengedit *code* yang menjadi struktur dari website tersebut

- Website Dinamis

  Website Dinamis merupakan *website* yang secara strukturdiperuntukan untuk *update* sesering mungkin. Biasanya selain utama yang bisa diakses oleh *user* pada umumnya,juga disediakan halaman backend untuk mengedit kontent, dari web portal yang didalamnya terdapat fasilitas berita,polling dan sebagainya

- Website interaktif

  Website Interaktif adalah web yang saat ini memang sedang dbooming.salah satunya contoh *website* interaktif adalah blog dan forum.di *website ini user* bisa berinteraksi dan beradu argument mengenai apa yang menjadi permikiran mereka. Biasanya *website* seperti memiliki moderator untuk mengatur supaya topik yang diperbincangkan tidak keluar jalur.

2.4.3 Manfaat Website

Website mempunyai beberapa manfaat atau kegunaan, antara lain:

1. Fungsi Komunikasi

   Situs web yang mempunyai fungsi komunikasi pada umumnya adalah situs web dinamis. Karena dibuat menggunakan pemograman web *(server side*) maka dilengkapi fasilitas yang memberikan fungsi-fungsi komunikasi, seperti *web mail*, *form contact, chatting form*, dan yang lainnya.

2. *Fungsi Informasi*

   Situs *web* yang memiliki fungsi informasi pada umumnya lebih menekan pada kualitas bagian kontennnya, karena tujuan situs tersebut adalah menyampaikan isinya. Situs ini sebaliknya berisi teks dan grafik yang dapat di *download* dengan cepat.pembatasan penggunaan animasi *gambar* dan elemen bergerak seperti *shockwave* dan java diyakini sebagai langkah yang tepat,diganti dengan fasilitas yang memberikan fungsi informasi seperti *news,profile compsny, libary, reference,dll.*

3. Fungsi entertaintment

   Situs *web* juga dapat memiliki fusi *entertiment/* hiburan. Bila situs *web* kita berfungsi sebagai sarana hiburan maska penggunaan animadi gambar dan elemen bergerak dapat meningkatan mutu prestasi desainya, meski tetap harus memperytimbangkan kecepatan *downloadnya*. Beberapa fasilitas yang memberikan

fungsi hiburan adalah *game online*, film *online*, music *online*, dan sebagainya.

4. Fungsi transaksi

   Situs *web* dapat dijadikan sarana transaksi bisnis, baik barang, jasa, atau lainnya. Situs web ini menghubungkan perusahaan, konsumen, dan komunitas tertentu melalui transaksi elektronik. Pembayarannya bisa menggunakan kartu kredit, *transfer*, atau dengan membayaran secara langsung.

2.5 UML *(Unified Modeling Language)*

2.5.1 Pengertian UML

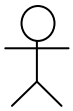
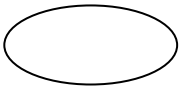
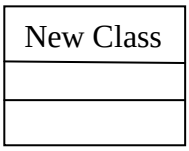
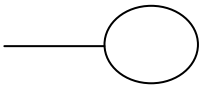
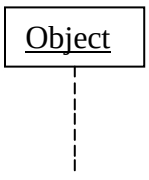
Model adalah bentuk sederhana dari kenyataan yang menyediakan penjelasan lengkap dari beberapa sudut pandang Sommerville (2011:6). membangun sebuah model untuk dapat lebih memahami sistem yang akan dibuat. *Modeling* sangat penting karena membantu tim dalam memvisualisasikan, menspesifikasikan, membangun, dan mendokumentasikan struktur dan *behavior* dari arsitektur sistem. Menggunakan bahasa *modeling* standard seperti UML (*Unified Modeling Language*), dapat membantu komunikasi antar anggota tim sehingga menghasilkan keputusan yang tidak ambigu.

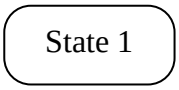
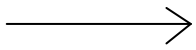
Menggunakan alat *visual modeling* dapat memfasilitasi manajemen model, dan merawat konsistensi antara sistem artefak: *requirements*, *designs*, dan *implementations*. Kesimpulannya, *visual modeling* membantu meningkatkan kemampuan tim untuk mengatur kompleksitas perangkat lunak.

2.5.2 Elemen UML

Tabel berikut berisi notasi , deskripsi dan simbol yang digunakan dalam UML.

Tabel 2.1 Tabel Notasi,deskripsi dan simbol UML

Notasi	**Deskripsi**	**Simbol**
Aktor	Segala pengguna aplikasi	
Use case	Menjelaskan kegiatan yang dilakukan aktor dan sistem untuk mencapai tujuan tertentu.	
Class	Menunjukkan kumpulan objek yang memilki atribut dan operasi yang sama. Digunakan untuk mengabstraksikan elemen – elemen yang sedang di bangun	
Interface	Kumpulan operasi tanpa implementasi dari suatu class. Merupakan suatu cara mewujudkan prinsip enkapsulasi dalam objek.	
Objek lifeline	Suatu objek dengan keterangan waktu (lifeline) dieksekusinya , objek tersebut	
Decision	Keadaan kondisional yang menyebabkan lebih dari suau state.	

Initial state	Keadaan awal untuk memulai aksi	
Final state	Keadaan akhir jika proses aksi tersebut selesai dilakukan	
State	Menunjukkan kadaan suatu objek bergantung pada kegiatan dan keadaan yang berlaku saat ini.	
Transition	Peralihan yang merupakan event (peristiwa) atau condition (keadaan) yang menyebakan perubahan pada state	

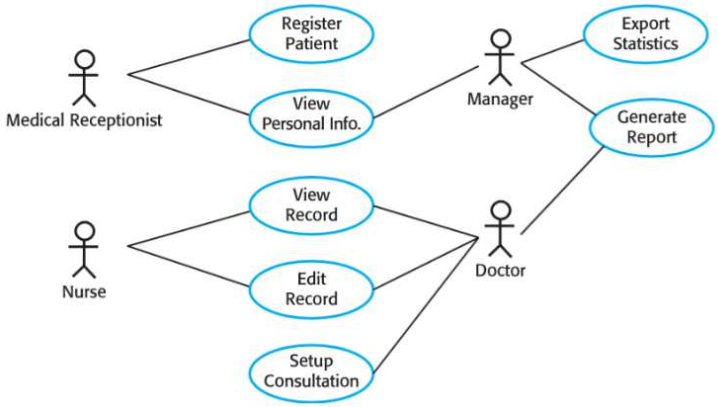
2.5.3 Tipe-tipe UML

2.5.3.1 Use Case

Sebuah *Use Case* adalah salah satu cara formal dalam menggambarkan bagaimana sistem bisnis berinteraksi dengan lingkungannya. *Use Case* mengilustrasikan aktifitas yang dilakukan oleh pengguna system. *Use Case Modeling* seringkali disebut sebagai sudut pandang dari bisnis proses, karena menunjukkan bagaimana pengguna melihat proses bukan mekanisme internal.

Use case didokumentasikan *high-level use case diagram*. Sekumpulan use case menggambarkan semua interaksi yang dijelaskan dalam kebutuhan sistem. Aktor dalam proses, bisa sebagai manusia atau sistem lain yang digambarkan dalam bentuk *stick figures*. Setiap kelas dalam interaksi dilambangkan sebagai

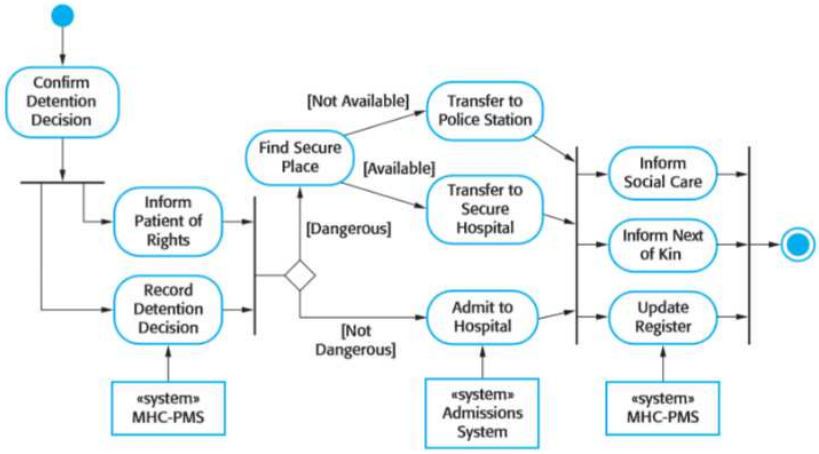
ellipse. Garis menghubungkan aktor dengan interaksi. Terkadang garis panah digambarkan untuk menunjukkan darimana interaksi bermula.*Use Case* mengidentifikasi interaksi individu antara sistem dengan pengguna atau dengan sistem lain. Setiap *Use Case* harus didokumentasikan dengan penjelasan tekstual. Sehingga, dapat dihubungkan dengan model lain dalam UML yang akan mengembangkan scenario lebih detil. Sebagai contoh, penjelasan dari *use case* konsultasi Gambar 2.1



Gambar 2.1 *Use Case* Konsultasi

Menurut Sommerville (2011:107) Aturan konsultasi mengijinkan dua atau lebih dokter, bekerja di kantor berbeda, melihat data tersimpan yang sama di waktu bersamaan. Satu dokter berinisiatif untuk membuka konsultasi dengan memilih orang yang terlibat dari menu pilihan dokter yang *on-line*. Data pasien kemudian muncul pada layar mereka namun hanya dokter pengusul yang dapat mengganti data. Sebagai tambahan, terdapat layar *chat* yang dibuat untuk dapat membantu.

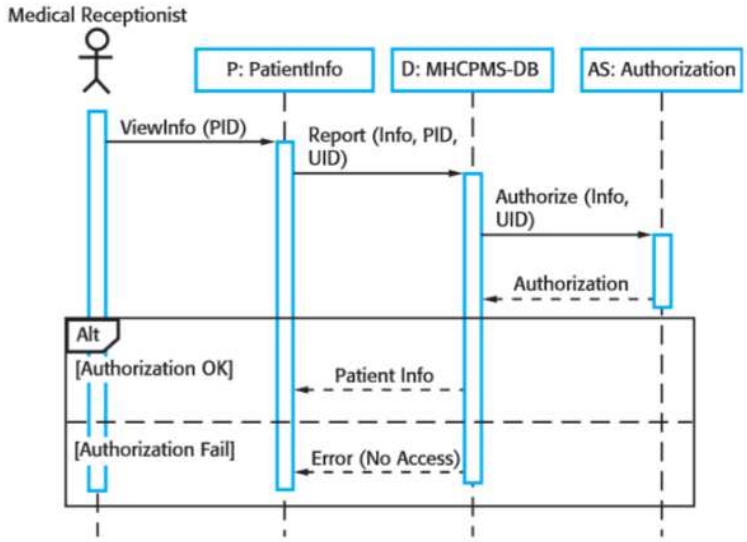
2.5.3.2 Activity Diagram



Gambar 2.2 *Activity Diagram*

Menurut Sommerville (2011:23) *activity diagram* yang ditujukan untuk menunjukkan aktifitas yang membuat sebuah proses sistem dan aliran kontrol dari satu aktifitas ke aktifitas lain. Untuk memulai proses dimulai dengan *filled circle*, dan diakhiri dengan lambing lingkaran dalam lingkaran. Persegi dengan segi tumpul menggambarkan aktifitas. Dalam UML *activity diagram*, garis panah menggambarkan alur kerja dari aktiftas satu ke aktifitas yang lain. Garis tebal digunakna untuk mengindikasi kordinasi aktifitas. Saat alur mulai lebih dari satu aktifitas mengarah ke garis tebal maka semua aktifitas harus dilakukan sebelum melanjutkan ke tahap berikutnya. Saat alur dari garis tebal menuju aktifitas tertentu, maka aktifitas tersebut boleh dikerjakan secara paralel.

2.5.3.3 Sequence Diagram



Gambar 2.3 *Sequence diagram*

Sumber : Sommerville (2011:127)

Sequence diagram dalam UML digunakan untuk memodelkan interaksi antara aktor dan objek dalam sistem dan interaksi dengan dirinya sendiri. UML mempunyai banyak *syntax* untuk *sequence diagram* yang memungkinkan berbagai macam interaksi untuk dimodelkan. Seperti namanya, sebuah *sequence diagram* menunjukkan urutan dari interaksi yang digunakan selama *use case* tertentu. Gambar 2.3 adalah sebuah contoh *sequence diagram* yang menggambarkan notasi sederhana. Diagram ini memodelkan interaksi yang terjadi dalam tampilan use case informasi pasien, dimana resepsionis dapat melihat informasi pasien.

2.5.4 Tujuan UML

1. Memberikan model yang siap pakai, bahasa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.
2. Memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa
3. Memodelkan suatu sistem (bukan hanya perangkat lunak) yang menggunakan konsep berorientasi object

2.6 Software Testing

Pengujian ini dimaksudkan untuk menunjukkan bahwa program melakukan apa yang dimaksudkan untuk dilakukan dan cacat Program discover sebelum mulai digunakan. Ketika menguji perangkat lunak maka menjalankan sebuah program dengan menggunakan data buatan. memeriksa hasil uji coba untuk kesalahan, anom-alies, atau informasi tentang atribut non-fungsional program. Proses pengujian memiliki dua tujuan yang berbeda:

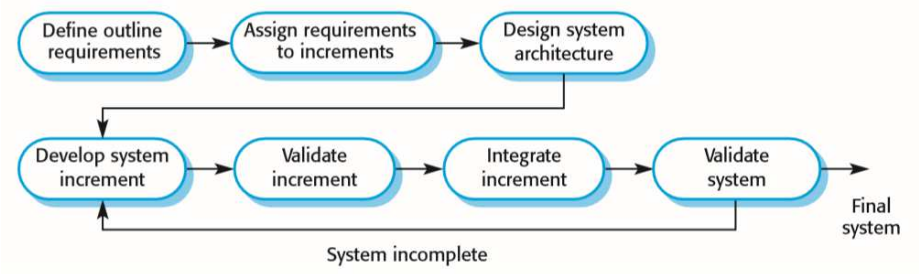
1. Untuk menunjukkan kepada pengembang dan pelanggan bahwa perangkat lunak memenuhi persyaratan. Untuk perangkat lunak kustom, ini berarti bahwa harus ada setidaknya satu tes untuk setiap kebutuhan dalam dokumen persyaratan. Untuk produk perangkat lunak generik, itu berarti bahwa harus ada tes untuk semua fitur

sistem, ditambah kombinasi dari fitur ini, yang akan dimasukkan dalam rilis produk.

2. Untuk menemukan situasi di mana perilaku dari software ini tidak benar atau tidak sesuai dengan spesifikasinya. Ini adalah konsekuensi dari cacat perangkat lunak. Pengujian Cacat berkaitan dengan membasmi diinginkan perilaku system seperti sistem crash, interaksi yang tidak diinginkan dengan sistem lain, perhitungan yang salah, dan korupsi data.

Tujuan pertama mengarah ke pengujian validasi, di mana Anda mengharapkan sistem untuk melakukan dengan benar menggunakan satu set kasus uji yang mencerminkan diharapkan penggunaan sistem. Gol kedua mengarah ke pengujian cacat, di mana uji kasus dirancang untuk mengekspos cacat. Uji kasus dalam pengujian cacat dapat sengaja tidak jelas dan tidak perlu mencerminkan bagaimana sistem biasanya digunakan. Tentu saja, tidak ada batas yang pasti antara dua pendekatan untuk pengujian. Selama pengujian validasi , Anda akan menemukan cacat dalam sistem;selama pengujian cacat, beberapa tes akan menunjukkan bahwa program memenuhi persyaratan.

2.6.1 Incremental delivery



Gambar 2.4 alur incremental delivery

Sumber : Sommerville (2011)

Model waterfall pembangunan mengharuskan pelanggan untuk sistem untuk berkomitmen satu set persyaratan sebelum desain dimulai dan desainer untuk berkomitmen terutama particular strategi desain sebelum pelaksanaan. Perubahan persyaratan memerlukan pengerjaan ulang dari persyaratan, desain dan implementasi. Namun, pemisahan desain dan implementasi harus mengarah pada sistem terdokumentasi dengan baik yang setuju untuk berubah. Sebaliknya, pendekatan evolusioner untuk pengembangan memungkinkan persyaratan dan keputusan desain yang akan ditunda, tetapi juga mengarah ke perangkat lunak yang mungkin kurang terstruktur dan sulit untuk memahami dan memelihara. Dalam proses Incremental delivery, pelanggan mengidentifikasi, secara garis besar, layanan yang akan disediakan oleh sistem. Mereka mengidentifikasi mana dari pelayanan-pelayanan yang paling penting dan yang paling penting bagi mereka. Sejumlah bertahap delivery kemudian didefinisikan, dengan kenaikan masing-masing menyediakan subset dari fungsi sistem. Alokasi layanan kepada bertahap tergantung pada prioritas sebagian besar dengan layanan prioritas

tertinggi disampaikan pertama. Setelah penambahan sistem telah diidentifikasi, persyaratan untuk pelayanan-pelayanan yang akan disampaikan dalam selisih pertama didefinisikan secara rinci, dan kenaikan yang dikembangkan. Selama pembangunan, persyaratan analisis lebih lanjut untuk kemudian bertahap dapat terjadi, tetapi persyaratan perubahan untuk kenaikan saat ini tidak diterima. Setelah kenaikan selesai dan dikirim , pelanggan dapat memasukkannya ke dalam layanan. Ini berarti bahwa mereka menerima pengiriman awal bagian dari fungsi sistem. Mereka dapat melakukan percobaan dengan sistem yang membantu mereka memperjelas persyaratan mereka untuk kenaikan kemudian dan untuk versi kenaikan saat ini. Seperti kenaikan baru selesai, mereka terintegrasi dengan penambahan yang ada sehingga fungsi sistem membaik dengan kenaikan masing-masing disampaikan. Layanan umum dapat diimplementasikan pada awal proses atau dapat diimplementasikan secara bertahap sebagai fungsi yang dibutuhkan oleh sebuah kenaikan. Proses pembangunan inkremental ini memiliki sejumlah keuntungan:

1. Pelanggan tidak harus menunggu sampai seluruh sistem disampaikan sebelum mereka dapat memperoleh nilai dari itu . Kenaikan pertama memenuhi paling penting persyaratan-persyaratan mereka sehingga mereka dapat menggunakan perangkat lunak segera .
2. Pelanggan dapat menggunakan peningkatan awal prototipe dan mendapatkan pengalaman yang menginformasikan kebutuhan mereka untuk peningkatan sistem nanti.

3. Ada risiko yang lebih rendah dari kegagalan proyek secara keseluruhan . Meskipun masalah dapat encoun - yang terdaftar di beberapa bertahap, ada kemungkinan bahwa beberapa akan berhasil dikirim ke pelanggan .
4. Sebagai layanan prioritas tertinggi diserahkan pertama, dan kemudian bertahap inte - parut dengan mereka, tidak dapat dihindari bahwa layanan sistem yang paling penting menerima paling pengujian .Ini berarti bahwa pelanggan cenderung untuk mengalami kegagalan perangkat lunak di bagian yang paling penting dari sistem.

Namun, ada masalah dengan pengiriman tambahan. Bertahap harus relatif kecil (tidak lebih dari 20.000 baris kode), dan setiap kenaikan harus memberikan beberapa fungsi sistem . Ini bisa sulit untuk memetakan persyaratan-persyaratan pelanggan ke penambahan sebesar ukuran yang tepat. Selain itu , kebanyakan sistem memerlukan seperangkat fasilitas dasar yang digunakan oleh berbagai bagian dari sistem. Sebagai persyaratan tidak didefinisikan secara rinci sampai kenaikan akan dilaksanakan , itu akan sulit untuk fasilitas umum iDEN - tify yang dibutuhkan oleh semua bertahap. Hal ini didasarkan di sekitar pengembangan dan pengiriman bertahap sangat kecil fungsionalitas , keterlibatan pelanggan dalam proses , kode perbaikan dan pasangan pemrograman konstan.

2.7 PHP - MySQL

PHP adalah bahasa pemrograman sisi server yang mengijinkan untuk memasukkan instruksi ke halaman web dimana software web server akan

mengeksekusi sebelum dikirim ke halaman web yang diminta. Sebuah database server (MySQL) adalah program yang dapat menyimpan banyak informasi dan format yang terorganisir sehingga memudahkan dalam bahasa pemrograman seperti PHP.

2.8 Database

2.8.1 Pengertian Database

Menurut Neeraj Sharma, Liviu Perniu, Raul F. Chong, Abhishek Iyer, Chaitali Nandan, Adi-Cristina Mitea, Mallarswami Nonvinkere, Mirela Danubianu (2010:23) Database adalah tempat penyimpanan data, yang dirancang untuk mendukung penyimpanan data yang efisien, pengambilan dan pemeliharaan.

2.8.2 Pengertian Database Management System

Menurut Neeraj Sharma, Liviu Perniu, Raul F. Chong, Abhishek Iyer, Chaitali Nandan, Adi-Cristina Mitea, Mallarswami Nonvinkere, Mirela Danubianu (2010:24) Sementara database adalah satu set perangkat lunak yang mengontrol akses , mengatur, menyimpan, mengelola , mengambil dan memelihara data dalam database . Dalam penggunaan praktis , database istilah, server database, sistem database , data server **,** dan sistem manajemen database yang sering digunakan secara bergantian .

2.8.2.1 Hierarki Penyimpanan Database

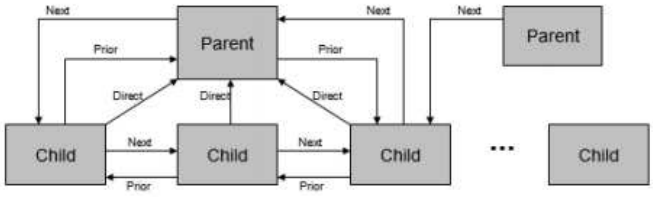
1. Bit komputer seperti atas dasar prinsip bahwa listrik bisa dihidupkan dan dimatikan(prinsip saklar on/off). Jadi bit adalah unit data terkecil yang bisa disimpan komputer. Bit adalah unit data terkecil yang bisa disimpan dalam komputer yang direpresentasikan dengan lambang angka 0 (off) atau 1 (on).
2. Field adalah sebuah unit data yang berisi satu atau lebih karakter (byte). Ia merupakan unit terkecil dari informasi berharga dalam database. setiap field memiliki nama field yang menggambarkan jenis data yang harus dimasukkan ke dalam field. Contoh field adalah nama pertama Anda, alamat jalan, atau jenis kelamin Anda. Field dapat didesain dengan panjang maksimun terntentu. Field juga dapat didesain dengan tipe data berbeda, semisal hanya teks, atau hanya angka, tanggal, waktu, atau bahkan hanya jawaban “ya” dan “tidak”, link web, gambar, suara dan video.
3. Record adalah kumpulan field-field yang berhubungan. Masing-masing record menyimpan data hanya sekitar satu entitas, yang bisa erupa orang, tempat benda dan peristiwa atau gejala. Contoh record bisa saja nama dan alamat Anda dan Nomor Jaminan sosial Anda.
4. File adalah kumpulan record-record yang saling berhubungan. contoh sebuah file adalah data tentang siapa saja yang bekerja di

departemen yang sama dalam sebuah perusahaan, termasuk nama, alamat, dan nomor jaminan sosial. File banyak digunakan karena ia merupakan kumpulan data atau informasi yang diperlakukan sebagai satau unit oleh komputer.dalam hierarki data, file berada dalam bagian atas. Kumpulan file yang berhubungan membentuk suatu database.database sebuah perusahaan mencakup semua file pegawai sebelumnya dan pegawai saat ini di departemen. Setiap pegawai bisa punya beberapa file, sebagai contoh file upah, dana pensiun, kuota penjualan, dan penerimaan, dan sebagainya.

2.8.3 Jenis-jenis information Model

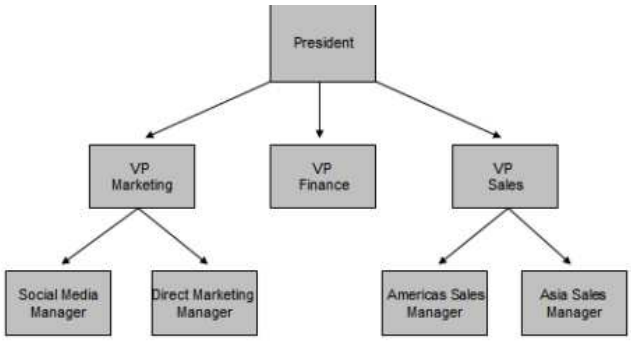
Menurut Neeraj Sharma, Liviu Perniu, Raul F. Chong, Abhishek Iyer, Chaitali Nandan, Adi-Cristina Mitea, Mallarswami Nonvinkere, Mirela Danubianu (2010:27) jenis-jenis information model terdiri dari Network model, Hierarchical model, Relational model, Entity-Relationship model, Object-relational model, Other data models.

- Network model: pada tahun 1969, CODASYL (Committee on Data Systems Languages) merilis pertama spesifikasi tentang model data jaringan. Hal ini diikuti pada tahun 1971 dan 1973 dengan spesifikasi untuk catatan waktu ata bahasa manipulasi data.



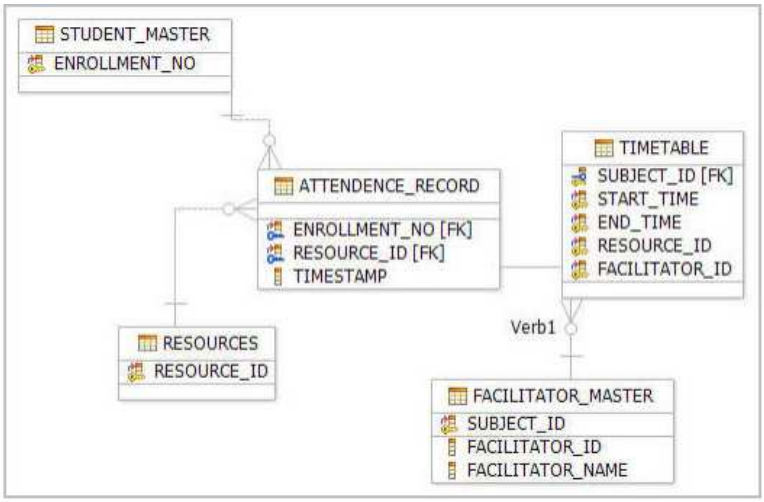
Gambar 2.5 A network model

- Hierarchical model: Model hirarkis mengatur data dengan menggunakan struktur pohon. Akar pohon adalah induk diikuti oleh node anak. Sebuah node anak tidak dapat memiliki lebih dari satu orang tua, meskipun orang tua dapat memiliki banyak node anak.



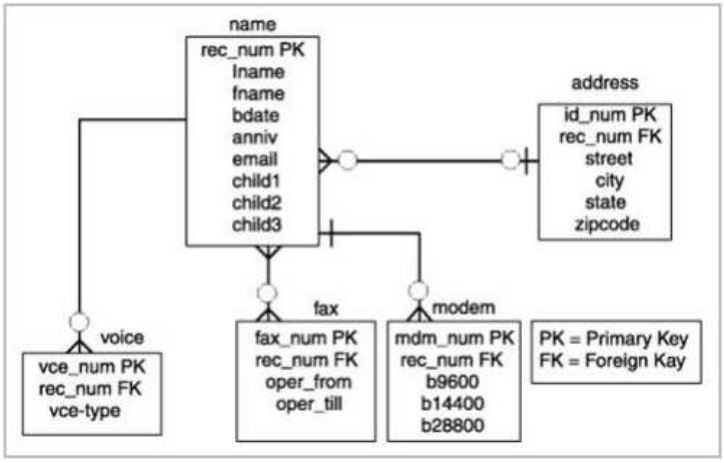
Gambar 2.6 A Hierarchical model

- Relational model: Relational model sederhana dan elegan. Ini memiliki dasar matematika yang kuat berdasarkan teori set dan predikat kalkulus dan merupakan model data yang paling banyak digunakan untuk database saat ini.



Gambar2.7 Relational model

- Pada pertengahan 1970-an, Peter Chen mengusulkan entity-relationship (ER) model data. Ini menjadi alternatif untuk relasion CODASYL, dan hierarchical data models. dia berpikir diusulkan database sebagai kumpulan contoh entitas. Entitas adalah objek yang memiliki independen keberadaan setiap entitas lain dalam database. Entitas memiliki atribut, yang merupakan elemen data yang mencirikan entitas. Satu atau lebih atribut ini dapat ditunjuk untuk menjadi kunci. Terakhir, mungkin ada hubungan antara entitas. Hubungan bisa menjadi 1-to-1, 1-to-n, n-ke-1 atau m-to-n, tergantung pada bagaimana entitas berpartisipasi dalam hubungan. Hubungan juga dapat memiliki atribut yang menggambarkan hubungan.



Gambar 2.8 E-R Diagram for a telephone directory data model

- The Object-Relational (OR) model yang sangat mirip dengan model relasional; Namun, memperlakukan setiap entitas sebagai obyek (instance dari kelas), dan hubungan sebagai warisan.

  Beberapa fitur dan manfaat dari model Object-Relational adalah:

  1. Dukungan untuk kompleks, ditetapkan pengguna jenis
  2. Object inheritance
  3.Extensible objects

  Database Object-Relational memiliki kemampuan untuk menyimpan hubungan objek dalam bentuk relasional.

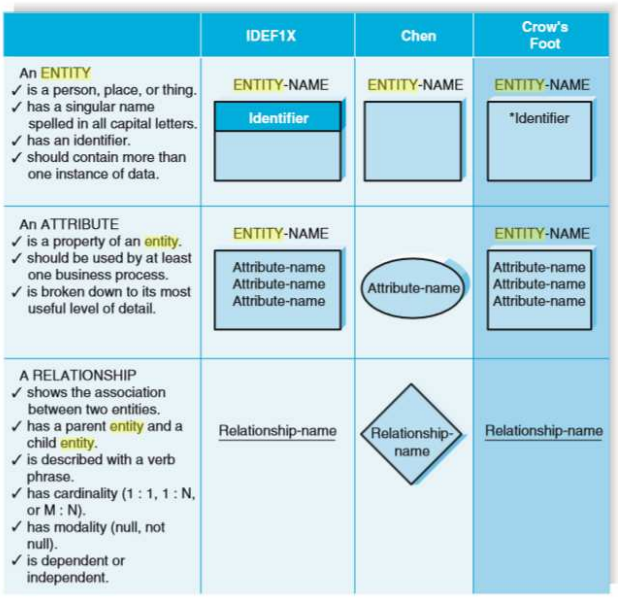
- Other data models: Dekade terakhir telah melihat sejumlah besar bekerja pada semi-terstruktur, model data semantik dan berorientasi objek. XML sangat ideal untuk menyimpan data semi-terstruktur. Model berbasis XML telah mendapatkan banyak popularitas dalam industri berkat Web 2.0 dan arsitektur berorientasi layanan (SOA). Model data berorientasi objek yang

populer di perguruan tinggi, tetapi belum diterima secara luas di industri; Namun, object-relational mapping (ORM) alat yang tersedia yang memungkinkan integrasi program berorientasi objek dengan database relasional.

2.9 Entity Relationship Diagram

2.9.1 Pengertian Entity Relationship Diagram

Menurut alan dennis (2009:210) entity relationship diagram adalah gambar yang menunjukkan informasi yang dibuat, disimpan, dan digunakan oleh sistem bisnis. Seorang analis dapat membaca ERD untuk menemukan bagian-bagian individu dari informasi dalam suatu sistem dan bagaimana mereka terorganisir dan terkait satu sama lain. Pada ERD, jenis yang sama dari informasi yang tercantum bersama-sama dan ditempatkan di dalam kotak yang disebut entitas. Garis ditarik antara entitas untuk mewakili hubungan antar data, dan simbol-simbol khusus itare ditambahkan ke dia-gram untuk berkomunikasi aturan bisnis tingkat tinggi yang perlu didukung oleh sistem. ERD berarti tidak ada order, meskipun entitas yang terkait satu sama lain biasanya ditempatkan dekat bersama-sama.



Gambar 2.9 ERD

2.9.2 Jenis-Jenis Relationships

Menurut Ryan K. Stephens,Ronald R. Plew(2000:162) jenis Relationships terdiri dari One-to-one, One-to-many, Many-to-many, Recursive, Mandatory, Optional.

- One-to-One Relationship: Hubungan satu-ke-satu merupakan hubungan antara entitas yang satu kejadian data dalam satu kesatuan mungkin memiliki satu terjadinya data dalam entitas yang terkait. Entitas A mungkin hanya satu kejadian data terkait dalam entitas B, dan entitas B mungkin hanya memiliki satu kejadian data terkait dalam entitas A.

Gambar 2.10 contoh one-to-one relationship

- One-to-Many Relationship: Dalam kebanyakan database relasional yang telah kita lihat, hubungan one-to-many tampaknya menjadi Hubungan yang paling umum yang ada. Hubungan satu-ke-banyak merupakan relasi antara entitas yang satu kejadian data dalam satu kesatuan mungkin memiliki satu atau lebih kejadian data di entitas terkait. Misalnya, entitas A mungkin memiliki beberapa kejadian data yang terkait dalam entitas B.

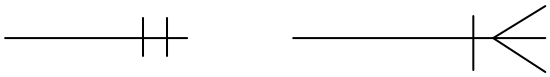
Gambar 2.11 one-to-many relationship

- Many-to-Many Relationship**:** Sebuah Many-to-Many Relationship ada jika beberapa kejadian data terkait diperbolehkan ada antara dua entitas, di kedua arah. Misalnya, entitas A mungkin memiliki banyak kejadian data yang terkait dalam entitas B, dan entitas B mungkin memiliki banyak kejadian data terkait dalam entitas A.

Gambar 2.12 many-to-many relationship

- Recursive Relationships: Kadang-kadang masuk akal untuk berhubungan data ke data lain dalam satu kesatuan. Sebuah Recursive Relationships adalah hubungan melingkar yang ada antara dua atribut dalam entitas yang sama. Recursive Relationships jarang terjadi, tapi berguna. Contoh yang paling umum digunakan untuk menggambarkan Recursive Relationships adalah nama karyawan dan manajer. Setiap karyawan memiliki seorang manajer, yang juga seorang karyawan.
- mandatory relationship: A mandatory relationship mewakili data yang diperlukan telah terkait data, atau Anda bisa mengatakan bahwa hubungan harus ada. mandatory relationship biasanya menggunakan kata harus. Berikut ini adalah contoh dari mandatory relationship:
    1. Sebuah rekor gaji karyawan harus sesuai catatan personel karyawan. (Sebuah catatan gaji karyawan tidak bisa ada tanpa karyawan catatan personel yang sesuai.)
    2. Order harus ditempatkan oleh pelanggan. (Setiap order harus dikaitkan dengan satu pelanggan.)
    3. Perintah harus sesuai dengan produk yang tersedia. (Setiap order harus dengan satu produk.)
    4. Seorang penulis harus dikaitkan dengan satu atau lebih penerbit

5. Sebuah buku harus dikaitkan dengan satu atau lebih penulis dan satu atau lebih penerbit.



Gambar 2.13 Simbol Mandatory Relationship

- Optional Relationships: Sebuah hubungan opsional tidak memerlukan hubungan ada, yang berarti bahwa data mungkin ada yang tidak terkait langsung dengan data dari entitas lain. Sebuah Optional Relationships biasanya menggunakan kata mungkin. Berikut ini adalah contoh dari Optional Relationships satu sisi:
    1. Pelanggan dapat menempatkan satu atau lebih perintah. (Seorang pelanggan mungkin tidak diperlukan untuk memesan, tetapi mungkin berhenti untuk dipertimbangkan pelanggan setelah jangka waktu tertentu tanpa aktivitas account.)
    2. Suatu produk dapat dipesan oleh pelanggan. (Products mungkin ada yang belum pernah diperintahkan.)
    3. Seorang individu diasuransikan dapat menempatkan klaim. (Seorang individu mungkin memiliki asuransi kesehatan dan tidak pernah memiliki klaim kesehatan.)
    4. Seorang karyawan dapat diberikan bonus. (Beberapa karyawan dapat diberikan bonus, sedangkan yang lain tidak.)

Gambar 2.14 Simbol Optional Relationship

2.10 Antrian

2.10.1 Pengertian Antrian

Menurut Hendra, Helfi Nasution(2012:27) Antrian merupakan bagian dari kehidupan manusia sehari-hari. Antrian terbentuk bilamana banyaknya yang akan dilayani melebihi kapasitas layanan yang tersedia. Dalam banyak hal, penambahan jumlah layanan dapat dipenuhi untuk mengurangi antrian atau menghindari antrian yang terus membesar namun demikian, biaya penambahan layanan dapat menyebabkan keuntungan berada di bawah taraf yang dapat diterima. Dipihak lain, antrian yang terlalu panjang dapat mengakibatkan kehilangan penjualan ataupun pelanggan.

2.10.2 Komponen Sistem Antrian

Komponen dasar proses antrian adalah:

1. Kedatangan

   Setiap masalah antrian melibatkan kedatangan, misalnya orang,mobil, panggilan telepon untuk dilayani, dan lain-lain. Unsur ini sering dinamakan proses input. Proses input meliputi sumber kedatangan atau biasa dinamakan calling population, dan cara terjadinya kedatangan yang umumnya merupakan variabel acak.

2. Pelayan

   Pelayan atau mekanisme pelayanan dapat terdiri dari satu atau lebih pelayan, atau satu atau lebih fasilitas pelayanan. Tiap-tiap fasilitas pelayanan kadang-kadang disebut sebagai saluran (channel). Contohnya, jalan tol dapat memiliki beberapa pintu tol. Mekanisme pelayanan dapat hanya terdiri dari satu pelayan dalam satu fasilitas pelayanan yang ditemui pada loket seperti pada penjualan tiket di gedung bioskop.

3. Antri

   Inti dari analisa antrian adalah antri itu sendiri. Timbulnya antrian terutama tergantung dari sifat kedatangan dan proses pelayanan. Jika tak ada antrian berarti terdapat pelayan yang menganggur atau kelebihan fasilitas pelayanan.

2.11 Flowchart

2.11.1 Pengertian Flowchart

Flowchart adalah penyajian yang sistematis tentang proses dan logika dari kegiatan penanganan informasi atau penggambaran secara grafik dari langkah-langkah dan urut-urutan prosedur dari suatu program. Flowchart menolong analis dan programmer untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian. System flowchart adalah urutan proses dalam system dengan menunjukkan alat media input, output serta jenis media penyimpanan dalam

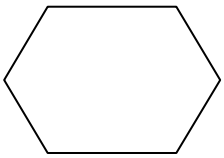
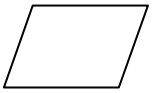
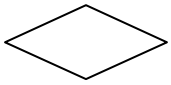
proses pengolahan data. Program flowchart adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses (instruksi) dengan proses lainnya dalam suatu program.

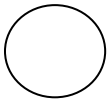
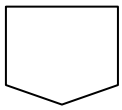
2.11.2 Pedoman-pedoman membuat flowchart

1. Flowchart digambarkan dari halaman atas ke bawah dan dari kiri ke kanan.
2. Aktivitas yang digambarkan harus didefinisikan secara hati-hati dan definisi ini harus dapat dimengerti oleh pembacanya.
3. Kapan aktivitas dimulai dan berakhir harus ditentukan secara jelas.
4. Setiap langkah dari aktivitas harus diuraikan dengan menggunakan deskripsi kata kerja, misalkan Melakukan penggandaan diri.
5. Setiap langkah dari aktivitas harus berada pada urutan yang benar.
6. Lingkup dan range dari aktifitas yang sedang digambarkan harus ditelusuri dengan hati-hati. Percabangan-percabangan yang memotong aktivitas yang sedang digambarkan tidak perlu digambarkan pada flowchart yang sama. Simbol konektor harus digunakan dan percabangannya diletakan pada halaman yang terpisah atau hilangkan seluruhnya bila percabangannya tidak berkaitan dengan sistem.
7. Gunakan simbol-simbol flowchart yang standar.

2.11.3 Simbol-simbol Flowchart

Tabel 2.2 Simbol-simbol Flowchart

	Karakteristik Produk	Deskripsi
	*Terminator*	Permulaan/akhir program
	Garis alir (flowline)	Arah aliran program
	*Preparation*	Proses inilisasi/pemberian harga awal
	*Proses*	Proses perhitungan/proses pengelolahan data
	*Input/output data*	Proses input/output data ,parameter,informasi
	*Predefined process(sub program)*	Permulaan sub program/proses menjalakan sub pogram
	*Decision*	Perbandingan pernyataan,penyeleksian data yang memberikan pilihan untuk langkah selanjutnya

	*On page connector*	Penghubung bagian bagian Flowchart yang berada pada satu halaman
	*Off page conector*	Penghubung bagian-bagian Flowchart yang berada pada halaman berbeda

2.12 SMS *Gateway*

2.12.1 Definisi Sms

Short Message Service (SMS) (Talukder, 2005.) merupakan sebuah layanan yang banyak diaplikasikan pada sistem komunikasi tanpa kabel, memungkinkan dilakukannyapengiriman pesan dalam bentuk teks. SMS didukung oleh GSM (Global System For Mobile Communication), TDMA (Time Division Multiple Access), CDMA (Code Division Multiple Access) yang berbasis pada telepon seluler yang saat ini banyak digunakan. SMS (Short Message Service) adalah merupakan salah satu layanan pesan teks yang dikembangkan dan distandarisasi oleh suatu badan yang bernama ETSI (European Telecommunication Standards Institute) sebagian dari pengembangan GSM (Global System for Mobile Communication) Phase 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 dan menerima pesan-pesan teks dengan panjang sampai dengan 160 karakter melalui jaringan GSM. SMS dapat dikirimkan ke perangkat stasiun

seluler digital lainnya hanya dalam beberapa detik selama berada pada jangkauan pelayanan GSM. Lebih dari sekedar pengiriman pesan biasa, layanan SMS memberikan garansi SMS akan sampai pada tujuan meskipun perangkat yang dituju sedang tidak aktif yang dapat disebabkan karena sedang dalam kondisi mati atau berada di luar jangkauan layanan GSM. Dengan adanya feature seperti ini maka layanan SMS juga cocok untuk dikembangkan sebagai aplikasi-aplikasi seperti: pager, e-mail, dan notifikasi voice mail, serta layanan pesan banyak pemakai (multiple user). Namun pengembangan aplikasi tersebut masih bergantung pada tingkat layanan yang disediakan oleh operator jaringan.

2.12.2 Keuntungan Sms

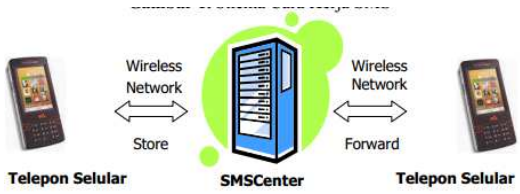
Pada tingkat minimum keuntungan yang dapat diberikan oleh SMS bagi pemakai meliputi pengiriman notifikasi dan peringatan (alert), penyampaian pesan SMS yang terjamin, handal, mekanisme komunikasi dengan biaya rendah, kemampuan untuk menyaring pesan SMS dan menanggapi panggilan secara selektif sehingga meningkatnya produktifitas customer.

Untuk fungsionalitas yang lebih canggih, SMS memberikan beberapa keuntungan tambahan bagi user yaitu pengiriman pesan SMS ke beberapa user sekaligus dalam waktu yang bersamaan, kemampuan menerima informasi yang beragam, dan integrasi dengan aplikasi lain yang berbasis internet dan data.

2.12.3 Cara Kerja SMS

Dalam sistem SMS, mekanisme utama yang dilakukan dalam suatu sistem adalah melakukan pengiriman short message dari satu terminal customer ke terminal yang lain. Hal ini dapat dilakukan berkat adanya sebuah entitas dalam sistem SMS yang bernama Short Message Service Center (SMSC), disebut juga Message Center (MC). Pada saat pesan SMS dikirim dari handphone (mobile orginated) pesan tersebut tidak langsung dikirim ke handphone tujuan (mobile terminated), akan tetapi terlebih dahulu ke SMSC,baru kemudian pesan tersebut dikirimkan ke handphone tujuan.

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 seperti Voice Mail System (VMS), Web-based messaging, Email Integration, External Short Message Entities (ESME), dan lain-lain.



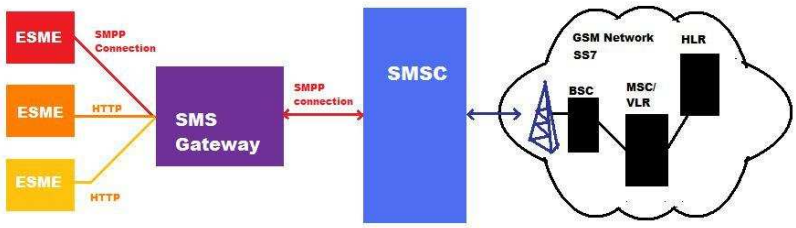
Gambar 2.15 Skema Cara Kerja SMS

2.12.4 Defini *SMS Gateway*

SMS *gateway* adalah sebuah perangkat yang menawarkan layanan transit SMS, mentransformasikan pesan ke jaringan selular dari media lain, atau sebaliknya, sehingga memungkinkan pengiriman atau penerimaan pesan SMS dengan atau tanpa menggunakan ponsel.

2.12.5 Cara Kerja *SMS Gateway*

Berikut ini adalah contoh kedudukan SMS *Gateway* di dalam jaringan milik operator beserta protokol komunikasi yang dipakai.



Gambar 2.16 Skema Cara Kerja SMS *Gateway*

Dari gambar diatas terlihat bahwa SMS *Gateway* berfungsi sebagai penghubung yang melakukan relay sms antara ESME (External Short Message Entitiy) dan SMSC dan sebaliknya. Komunikasi antara ESME dan SMS *Gateway* dapat menggunakan protokol SMPP atau dengan HTTP, sementara ke SMSC menggunakan SMPP.

ESME adalah entitas luar yang dapat berupa server aplikasi penyedia layanan (Application Service Provider) yang dimiliki oleh Content Provider, aplikasi Perbankan, server polling, dan lain-lain yang dapat menerima pesan, memproses pesan dan mengirim respons atas pesan yang masuk , serta perangkat lain seperti email *gateway*, WAP proxy server , Voice mail server.