

BAB II

LANDASAN TEORI

2.1 Sekilas Mengenai Website

Website adalah kumpulan dari halaman-halaman situs, yang biasanya terangkum dalam sebuah *domain* (nama unik yang diberikan untuk mengidentifikasi nama *server* komputer seperti *web server* atau *email server* di jaringan komputer ataupun internet) atau *subdomain* (bagian dari sebuah nama *domain* induk), yang tempatnya berada di dalam *World Wide Web* (WWW) di *Internet*. Sebuah halaman *web* adalah dokumen yang ditulis dalam format HTML (*Hyper Text Markup Language*) yang selalu bisa diakses melalui HTTP (*Hypertext Transfer Protokol*), yaitu protokol (sebuah aturan atau standar yang mengatur atau mengijinkan terjadinya hubungan, komunikasi, dan perpindahan data antara dua atau lebih titik komputer) yang menyampaikan informasi dari *server website* untuk ditampilkan kepada para pemakai melalui *web browser*. Semua publikasi dari *website-website* tersebut dapat membentuk sebuah jaringan informasi yang sangat besar. (Abdul Kadir, 2009 : 3).

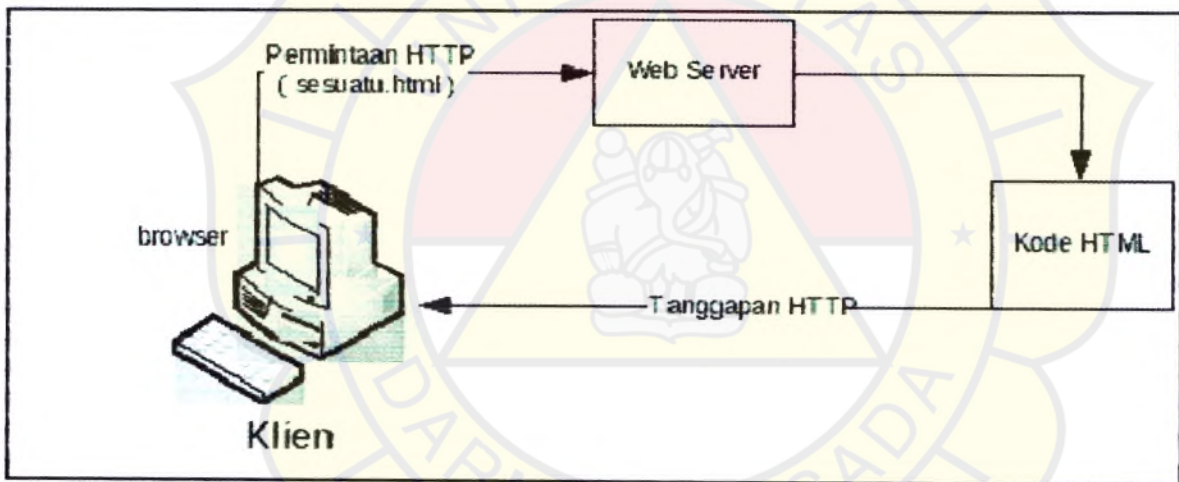
2.1.1 Arsitektur Website

Arsitektur dasar dari aplikasi web ini adalah arsitektur *client-server*. Artinya pemrosesan aplikasi ini dijalankan melibatkan kedua sisi yakni sisi mesin *server* pusat dan sisi *client*.

Client-server merupakan sebuah metode dalam teknologi informasi yang merujuk kepada cara untuk mendistribusikan aplikasi ke dalam dua pihak, pihak *client* dan pihak *server*.

Dalam model *client-server*, sebuah aplikasi dibagi menjadi dua bagian yang terpisah, tapi masih merupakan sebuah kesatuan yakni komponen *client* dan komponen *server*.

Komponen *client* juga sering disebut sebagai *front-end*, sementara komponen *server* disebut sebagai *back-end*. Komponen *client* dari aplikasi tersebut dijalankan dalam sebuah *workstation* dan menerima masukan data dari pengguna. Komponen *client* tersebut akan menyiapkan data yang dimasukkan oleh pengguna dengan menggunakan teknologi pemrosesan tertentu dan mengirimkannya kepada komponen *server* yang dijalankan di atas mesin *server*, umumnya dalam bentuk *request* terhadap beberapa layanan yang dimiliki oleh *server*. Komponen *server* akan menerima *request* dari *client*, dan langsung memrosesnya dan mengembalikan hasil pemrosesan tersebut kepada *client*. *Client* pun menerima informasi hasil pemrosesan data yang dilakukan *server* dan menampilkannya kepada pengguna, dengan menggunakan aplikasi yang berinteraksi dengan pengguna.



Gambar 2.1. Arsitektur *Client/Server* (Abdul Kadir, 2008:5)

Sebuah contoh dari aplikasi *client-server* sederhana adalah aplikasi *web* yang didesain dengan menggunakan *Active Server Pages* (ASP) atau PHP. *Script* PHP atau ASP akan dijalankan di dalam *web server* (Apache atau Internet Information Services), sementara *script* yang berjalan di pihak *client* akan dijalankan oleh *web browser* pada komputer *client*. *Client-server* merupakan penyelesaian masalah pada *software* yang menggunakan *database* sehingga setiap komputer tidak perlu diinstall *database*, dengan metode *client-server*

database dapat diinstal pada suatu komputer sebagai *server* dan aplikasinya diinstal pada *client*.

2.1.1.1 Web Server

Server web adalah sebuah perangkat lunak *server* yang berfungsi menerima permintaan HTTP atau HTTPS dari *client* yang dikenal dengan *browser* web dan mengirimkan kembali hasilnya dalam bentuk halaman web yang umumnya berbentuk dokumen HTML (http://id.wikipedia.org/wiki/Web_server, 28 Maret 2012). *Server web* yang terkenal diantaranya adalah Apache dan Microsoft Internet Information Service (IIS). Apache merupakan *server web* antar platform dimana dapat beroperasi di system operasi Windows dan Linux/Unix. IIS hanya dapat beroperasi di sistem operasi Windows.

2.1.1.2 Database Server

Server basis data adalah sebuah program komputer yang menyediakan layanan pengelolaan basis data dan melayani komputer atau program aplikasi basis data yang menggunakan model *client/server*.

(http://id.wikipedia.org/wiki/Server_basis_data, 28 Maret 2012). Istilah ini juga merujuk kepada sebuah komputer (umumnya merupakan *server*) yang didedikasikan untuk menjalankan program yang bersangkutan. Sistem manajemen basis data (SMBD) pada umumnya menyediakan fungsi-fungsi *server* basis data, dan beberapa SMBD (seperti halnya MySQL atau Microsoft SQL Server) sangat bergantung kepada model *client-server* untuk mengakses basis datanya.

2.1.1.3 Web Browser

Browser adalah program aplikasi yang menterjemahkan kode *HTML* dan merepresentasikan halaman web site. Aplikasi inilah yang paling sering kita gunakan setiap hari untuk melakukan *browsing* di dunia maya. Macam browser sekarang semakin banyak saja diantaranya adalah: *Internet Explorer*, *Netscape*, *Mozilla*, *Chrome*, *Safari*, dll. Dalam Bahasa Indonesia *Browser* diterjemahkan sebagai Peramban web, peramban ramatraya (*web browser*) disebut juga penjelajah web, adalah software/perangkat lunak yang berfungsi menampilkan dan melakukan interaksi dengan dokumen-dokumen yang disediakan oleh server web. Peramban web adalah jenis perantara pengguna yang paling sering digunakan. *Ramat jembar* (*wide web*) sendiri adalah kumpulan jaringan berisi dokumen dan tersambung satu dengan yang lain, yang dikenal sebagai *www* (*World Wide Web* / *Bhs.Indonesia: Waring Wera Wanua*).

Sejarah *Web Browser/Peramban Web* Peramban web pertama pertama bernama *WorldWideWeb* berbasis teks, seperti halnya *Lynx* yang populer hingga sekarang, karena memang sistem dengan antarmuka grafis belum umum digunakan pada saat itu. Baru setelah sistem berbasis grafis mulai banyak digunakan, seorang mahasiswa bernama Marc Andressen di University of Illinois di Urbana-Champaign, Amerika Serikat, membuat sebuah penjelajah web berbasis grafis pertama yang berjalan di atas sistem operasi Windows dan UNIX (berbasis Motif). Peramban web tersebut dinamai *Mosaic*. Selanjutnya, setelah lulus dari universitas, Marc ditawari oleh Jim Clarc, salah seorang petinggi Silicon Graphics Incorporated (SGI), untuk membuat perusahaan dengan nama *Mosaic Communication*, yang kemudian berubah menjadi Netscape Communication. Marc membuat sebuah penjelajah web populer pertama yang digunakan oleh umum, yang disebut dengan *Netscape Navigator*. Pada saat pengembangannya, Navigator memiliki nama kode *Mozilla*. Navigator merupakan penjelajah web komersial, akan tetapi seiring dengan waktu akhirnya Navigator pun menjadi

dapat diperoleh secara gratis. Sementara itu, penjelajah web Mosaic yang gratis tidak diteruskan lagi pengembangannya, dan diserahkan kepada NCSA (National Computing for Supercomputer Application). Karena NCSA tidak memiliki izin untuk mengomersialkan produk Mosaic, NCSA akhirnya menjual Mosaic kepada Spyglass, yang akhirnya membuat Mosaic menjadi penjelajah web komersial. Microsoft, yang terlambat turun di pasar peramban web, pun membeli lisensi Mosaic dari Spyglass, sehingga pada akhirnya merilis produk peramban web-nya, yang disebut sebagai *Internet Explorer*. Dengan dirilisnya Internet Explorer, muncullah perang besar antara peramban web populer, yakni antara netscape Navigator, dan Microsoft Internet Explorer. (<http://www.daishz.com/2009/01/browser-adalah.html>, 28 Maret 2012).

2.2 Sekilas Mengenai Short Message Service (SMS)

Short Message Service atau yang lebih dikenal orang dengan istilah SMS merupakan fitur yang digunakan untuk berkirim pesan dalam format teks. SMS ini dapat dinikmati oleh seluruh pengguna handphone. Handphone yang dulu merupakan barang mewah, saat ini sudah berubah status menjadi suatu kebutuhan utama yang harus dipenuhi. Tiap waktu, handphone pasti sangat diperlukan untuk menunjang aktivitas sehari-hari, baik itu pekerjaan, mereka yang bergelut dalam dunia bisnis, atau kegiatan lainnya.

Dengan adanya SMS, dapat dipastikan bahwa tiap pesan yang masuk itu pasti terbaca oleh pemilik handphone tersebut ditambah sekarang tarif operator yang lagi gencar-gencarnya mempromosikan tarif SMS murah bahkan gratis. Ini yang menyebabkan SMS menjadi salah satu andalan dalam komunikasi antar sesama. Saat ini SMS digunakan tidak terbatas untuk komunikasi antar pengguna saja, namun adakalanya SMS dibuat secara otomatis menggunakan komputer, terlebih lagi jika menyangkut pengiriman dalam jumlah banyak (massal). Komputer dapat mengirimkan pesan secara otomatis ke nomor yang dituju,

bandingkan jika Anda mengetikkan secara manual, kemudian memilih-milih nomor tujuan. Tentu itu akan memakan waktu lebih lama dibandingkan menggunakan komputer.

Layanan SMS lebih diminati masyarakat karena beberapa keunggulan, diantaranya:

- a. Biaya relatif murah, pengiriman terjamin sampai ke nomor tujuan dengan catatan nomor dalam keadaan aktif. Selain itu, waktu pengiriman juga cepat, bandingkan jika kita menggunakan pak pos untuk mengirimkan pesan
- b. Dengan layanan ini juga pengguna dapat mengirimkan pesan secara fleksibel, dalam artian pengguna dapat mengirim pesan kapan pun dan di mana saja.
- c. Layanan SMS ini mudah digunakan, dapat dipastikan orang bukan dari latar belakang IT (Information Technology) pun dapat memahami cara penggunaannya.

SMS atau *Short Message Service* pada awal diciptakan adalah bagian dari layanan pada sistem GSM. SMS semula hanyalah merupakan layanan yang bersifat komplementer terhadap dua layanan utama sistem GSM (atau sistem 2G pada umumnya) yaitu layanan *voice* dan *switched data*. Namun karena keberhasilan SMS yang tidak terduga, dengan ledakan pelanggan yang mempergunakannya, menjadikan SMS sebagai bagian integral dari layanan sistem.

SMS merupakan layanan messaging yang pada umumnya terdapat pada setiap sistem jaringan wireless digital. SMS adalah layanan untuk mengirim dan menerima pesan tertulis (teks) dari maupun kepada perangkat bergerak (*mobile device*). Pesan teks yang dimaksud tersusun dari huruf, angka, atau karakter alfanumerik. Pesan teks dikemas dalam satu paket/frame yang berkapasitas maksimal 160 byte yang dapat direpresentasikan berupa 160 karakter huruf latin atau 70 karakter alfabet non-latin seperti alfabet Arab atau Cina. (Agus Saputra, Membangun Aplikasi SMS dengan php dan MySQL, 2011:4)

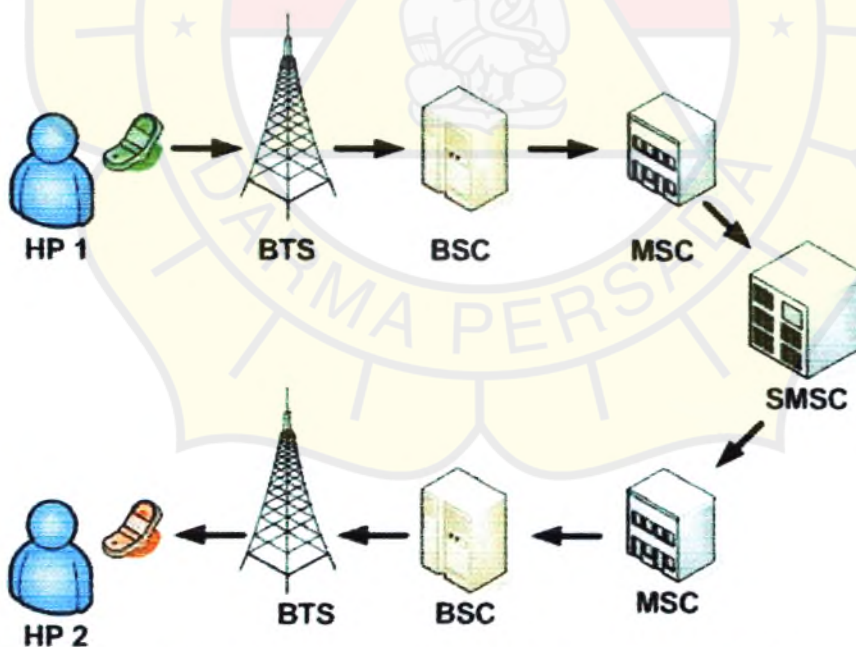
2.2.1 SMS Gateway

Merupakan salah satu penghubung dari provider ke pelanggan dengan tujuan tertentu. Biasanya SMS Gateway digunakan untuk kepentingan *Entertainment* atau *Advertising* dari *Content Provider*.

Dalam lingkup yang lebih kecil, SMS Gateway berfungsi untuk mengatur keluar masuknya pesan dari provider ke SIM Card yang diatur oleh *Server* dengan perangkat modem (GSM-Modem atau Telepon Genggam). Pemakaian disesuaikan dengan kebutuhan Admin atau user seperti:

- Pendaftaran melalui SMS (contoh: Reg<spasi>Nama)
- Auto Reply SMS Content (contoh: Mengetahui nilai ujian/kelulusan)
- Penggunaan Website SMS oleh user dengan sistem ticketing.

2.2.2 Alur Pengiriman SMS



Gambar 2.2 Alur Pengiriman SMS (Agus Saputra, 2011:5)

Keterangan:

- a. BTS (Base Transceiver Station)
- b. BSC (Base Station Controller)
- c. MSC (Mobile Switching Center)
- d. SMSC (Short Message Service Center)

Setiap kita mengirimkan pesan melalui SMS, Pesan tersebut tidak langsung sampai ke nomor handphone tujuan, tapi melewati beberapa proses terlebih dahulu. Yaitu pesan akan ditangkap oleh BTS terlebih dahulu dilanjutkan ke BSC kemudian akan sampai ke tahap MSC. MSC selanjutnya akan meneruskan atau mem-*forward* pesan tersebut kepada SMSC. Pada tahap inilah pesan tersebut disimpan untuk sementara jika nomor tujuan yang ditunjukkan sedang tidak aktif atau berada di luar jangkauan. Jika nomor tujuan sudah aktif maka akan diteruskan melewati MSC, BSC kemudian diterima oleh jaringan BTS nomor tujuan, kemudian dikirimkan kepada pengguna nomor handphone tersebut. (Agus Saputra, Membangun Aplikasi SMS dengan php dan MySQL, 2011 : 5).

2.3 Bahasa Pemrograman

2.3.1 HTML

HyperText Markup Language (HTML) adalah sebuah bahasa *markup* yang digunakan untuk membuat sebuah halaman *web* dan menampilkan berbagai informasi di dalam sebuah *browser internet*. Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut dengan SGML (*Standard Generalized Markup Language*), HTML adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman *web*. HTML saat ini merupakan standar *internet* yang didefinisikan dan dikendalikan penggunaannya oleh *World Wide Web Consortium* (W3C).

HTML berupa kode-kode *tag* yang menginstruksikan *browser* untuk menghasilkan tampilan sesuai dengan yang diinginkan. Sebuah file yang merupakan file HTML dapat dibuka dengan menggunakan *browser web* seperti Mozilla Firefox atau Microsoft Internet Explorer. HTML juga dapat dikenali oleh aplikasi pembuka *email* ataupun dari PDA dan program lain yang memiliki kemampuan *browser*. (http://id.wikipedia.org/wiki/Hypertext_markup_language, 30 Maret 2012).

2.3.2 CSS

CSS merupakan singkatan dari *Cascading Style Sheet*. Kegunaannya adalah untuk mengatur tampilan dokumen HTML, contohnya seperti pengaturan jarak antar baris, teks, warna dan format border bahkan penampilan file gambar.

CSS dikembangkan oleh W3C, organisasi yang mengembangkan teknologi internet. Tujuannya tak lain untuk mempermudah proses penataan halaman web.

Perlu diingat, CSS hanyalah berupa kumpulan skrip yang tujuannya bukan untuk menggantikan HTML, melainkan pelengkap agar dokumen HTML bisa tampil lebih cantik dan dinamis.

Sejak awal ditemukannya CSS pada awal dekadean 90an, CSS terus dikembangkan dan diserap oleh web developer. Hingga saat ini versi CSS telah mencapai versi ke-3.

Kode CSS bersifat lintas platform, yang berarti skrip ini dapat dibaca oleh berbagai macam sistem operasi dan *browser*. Hanya saja *browser* seperti Internet Explorer, seringkali salah mengartikan skrip CSS yang menyebabkan tidak sempurnanya tampilan dokumen HTML. Skrip CSS perlu dioptimalkan agar tampil maksimal pada *browser* Internet Explorer. (Jayan, CSS untuk Orang Awam, 2010 : 2).

2.3.3 JavaScript

JavaScript adalah bahasa skrip yang populer di internet dan dapat bekerja di sebagian besar penjelajah web populer seperti Internet Explorer (IE), Mozilla Firefox, Netscape dan Opera. Kode JavaScript dapat disisipkan dalam halaman web menggunakan tag SCRIPT.

JavaScript pertama kali dikembangkan oleh Brendan Eich dari Netscape dibawah nama *Mocha*. Yang berganti nama menjadi *LiveScript*, dan akhirnya menjadi *JavaScript*.

Navigator sebelumnya telah mendukung Java untuk lebih bisa dimanfaatkan para programmer yang non-Java. Maka dikembangkanlah bahasa pemrograman bernama *LiveScript* untuk mengakomodasi hal tersebut. Bahasa pemrograman inilah yang akhirnya berkembang dan diberi nama *JavaScript*, walaupun tidak ada hubungan bahasa antara Java dengan JavaScript.

JavaScript bisa digunakan untuk banyak tujuan, misalnya untuk membuat efek *rollover* baik di gambar maupun teks, dan yang penting juga adalah untuk membuat AJAX. JavaScript adalah bahasa yang digunakan untuk AJAX. (<http://id.wikipedia.org/wiki/JavaScript>, 5 Mei 2012).

2.3.4 PHP

Menurut dokumen resmi PHP, PHP merupakan singkatan dari *PHP Hypertext Preprocessor*. Ia merupakan bahasa berbentuk skrip yang ditempatkan dalam server dan diproses di server. Hasilnyalah yang dikirimkan ke klien tempat pemakai menggunakan *browser*.

Secara khusus, PHP dirancang untuk membentuk aplikasi web dinamis, Artinya, ia dapat membentuk suatu tampilan berdasarkan permintaan terkini. Misalnya, Anda bisa menampilkan isi *database* ke halaman web. Pada prinsipnya PHP mempunyai fungsi yang sama dengan skrip – skrip seperti ASP (*Active Server Page*), Cold Fusion ataupun Perl.

Namun, perlu diketahui bahwa PHP sebenarnya bisa dipakai secara *command line*. Artinya, skrip PHP dapat dijalankan tanpa melibatkan *webserver* maupun *browser*.

Kelahiran PHP bermula saat Rasmus Lerdorf membuat sejumlah skrip Perl yang dapat mengamati siapa saja yang melihat – lihat daftar riwayat hidupnya, yakni pada tahun 1994. Skrip – skrip ini selanjutnya dikemas menjadi *tool* yang disebut “Personal Home Page”. Paket inilah yang menjadi cikal – bakal PHP. Pada tahun 1995, Rasmus menciptakan PHP/FI Versi 2. Pada versi inilah pemrogram dapat menempelkan kode terstruktur di dalam tag HTML. Yang menarik, kode PHP juga bisa berkomunikasi dengan *database* dan melakukan perhitungan-perhitungan yang kompleks sambil jalan.

Pada saat ini PHP cukup populer sebagai peranti pemrograman Web, terutama di lingkungan Linux. Walaupun demikian, PHP sebenarnya juga dapat berfungsi pada *server-server* yang berbasis UNIX, Windows, dan Macintosh.

Pada awalnya, PHP dirancang untuk integrasikan dengan web server Apache. Namun, belakangan PHP juga dapat bekerja dengan web server seperti PWS (Personal Web Server), IIS (Internet Information Server) dan Xitami.

Untuk mencoba PHP, Anda tidak perlu menggunakan komputer berkelas server. Hanya dengan sebuah komputer biasa, Anda bisa mempelajari dan mempraktikkan PHP.

PHP bersifat bebas pakai. Anda tidak perlu membayar apa pun untuk menggunakan perangkat lunak ini alias *free*. Anda dapat men-*download*-nya melalui situs www.php.net. PHP tersedia dalam bentuk kode biner maupun kode sumber yang lengkap. (Abdul Kadir, Dasar Pemrograman WEB Dinamis Menggunakan PHP, 2008 : 2).

2.4 DBMS

2.4.1 Basis Data

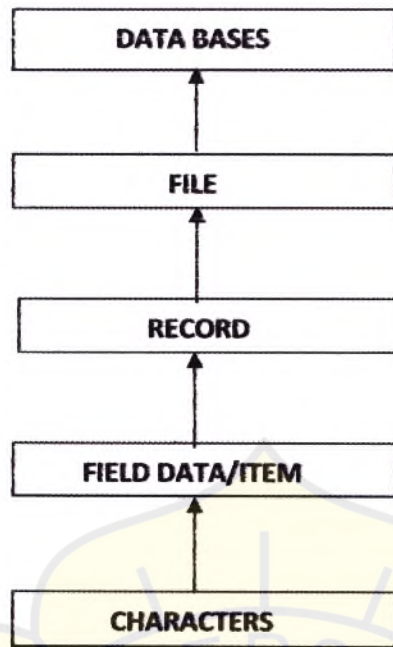
Basis data (database) adalah kumpulan file-file yang mempunyai kaitan antara satu file dengan file yang lain sehingga membentuk satu bangunan data untuk menginformasikan satu perusahaan, instansi, dalam batasan tertentu. (Edhy Sutanta, 2004).

Sistem basis data (database system) adalah system yang terdiri dari kumpulan file (table) yang saling berhubungan (dalam sebuah basis data di sebuah sistem komputer) dan sekumpulan program (DBMS) yang memungkinkan beberapa pemakai atau program lain untuk mengakses dan memanipulasi file-file (table-tabel) tersebut.

Sistem basis data yaitu behubungan antara basis data dengan komponen-komponen lain yang mendukung berfungsinya basis data tersebut.

Komponen-komponen sistem basis data antara lain :

- a. Hardware, contoh : teknologi storage, CPU, dan lain-lain.
- b. OS
- c. DBMS
- d. User
- e. Aplikasi



Gambar 2.3 Jenjang Data (Edhy Sutanta, 2004)

- a. **Characters** : merupakan bagian data yang terkecil, dapat berupa karakter numerik, huruf ataupun karakter-karakter khusus (special characters) yang membentuk suatu item data / field.
- b. **field Data / Item** : mempersentasikan suatu atribut dari record yang menunjukkan suatu item dari data, seperti nama, alamat, dan lain sebagainya.

Kumpulan dari field membentuk suatu record.

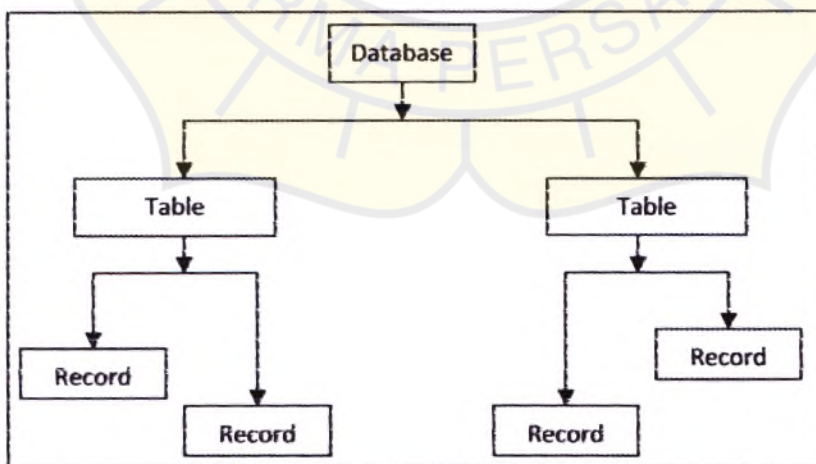
- field name : harus diberi nama untuk membedakan field yang satu dengan lainnya.
- field representation : tipe field (karakter, teks, tanggal, angka, dsb), lebar field (ruang maksimum yang dapat diisi dengan karakter-karakter data).
- field value : isi dari field untuk masing-masing record.

populer di kalangan programmer *web*. MySQL dapat dijalankan dalam 2 *operating system* yang sangat populer saat ini, yaitu : Windows dan Linux. Menurut perusahaan pengembangnya, MySQL telah terpasang di sekitar 3 juta komputer, dan puluhan hingga ratusan ribu situs sangat mengandalkan MySQL sebagai *databasenya*. (Mohamad Sukarno, *Membangun Website Dinamis Interaktif dengan PHP-MySQL*, 2006).

MySQL dikembangkan oleh sebuah perusahaan Swedia yang pada saat itu bernama TcX DataKonsult AB, dan pada akhirnya berubah nama menjadi MySQL AB. Sekitar tahun 1994-1995, TcX membuat MySQL untuk mengembangkan aplikasi *web* bagi klien-nya. TcX merupakan perusahaan pengembangan *software* dan konsultan *database*.

Dalam pembentukan sistem *website* yang dinamis sudah tentu akan membutuhkan pengolahan data secara otomatis pada *server*, hal itu membutuhkan suatu program *database engine* yang digunakan untuk membantu manajemen data pada *server*. *Database engine* seperti MySQL yang berdasar pada konsep *database relasional*.

Manajemen dasar *database relasional* adalah pengarsipan data-data dalam *record-record* tabel. Sebuah *database* dapat berisi tabel-tabel, dan *record* adalah bagian terkecil dari sebuah tabel. Seperti terlihat pada gambar dibawah ini :



Gambar 2.4 Struktur Basis Data Relasional (Bunafit Nugroho, 2008)

2.5.3 PHP MyAdmin

Pada awalnya para programmer *web* dalam membuat atau mengelola suatu *database* dilakukan pada *console* (Linux) atau *dos-prompt* (Windows). Semenjak adanya paket yang menyatukan antara Apache-MySQL-PHP, yaitu salah satunya PHPTriad, sehingga untuk melakukan pembuatan dan pengolahan *database* dapat dilakukan pada *browser*, yaitu dikenal dengan *phpmyadmin*. Xampp pun menyediakan folder *phpmyadmin* yang berfungsi dalam pembentukan dan pengolahan *database* pada *browser*. Adapun alamat URL (*Uniform Resource Locator*) *phpmyadmin* pada *xampp* di *operating system* Linux ataupun Windows, adalah <http://localhost/phpmyadmin>.

phpMyAdmin adalah suatu *tool* perangkat lunak yang tidak berbayar untuk PHP yang diharapkan untuk menangani administrasi MySQL didalam *Website*. *phpMyAdmin* merupakan *software* yang sering digunakan sebagai antarmuka dalam mengatur (*database*, tabel-tabel, *field-field*, relasi, dan lain-lain), bisa juga melaksanakan secara setiap perintah SQL. (Mohamad Sukarno, *Membangun Website Dinamis Interaktif dengan PHP-MySQL*, 2006).

2.5.4 NetBeans IDE 7.1

NetBeans merupakan sebuah proyek kode terbuka (*Open Source*) yang sukses dengan pengguna yang sangat luas, komunitas yang terus tumbuh, dan memiliki hampir 100 mitra (dan terus bertambah). *Sun Microsystems* mendirikan proyek kode terbuka NetBeans pada bulan Juni 2000 dan terus menjadi sponsor utama. Saat ini terdapat dua produk : NetBeans IDE dan NetBeans Platform.

The NetBeans IDE adalah sebuah lingkungan pengembangan - sebuah kakas untuk pemrogram menulis, mengompilasi, mencari kesalahan dan menyebarkan program. Netbeans IDE ditulis dalam Java - namun dapat mendukung bahasa pemrograman lain. Terdapat

banyak modul untuk memperluas Netbeans IDE. Netbeans IDE adalah sebuah produk bebas dengan tanpa batasan bagaimana digunakan.

Tersedia juga NetBeans Platform; sebuah fondasi yang modular dan dapat diperluas yang dapat digunakan sebagai perangkat lunak dasar untuk membuat aplikasi desktop yang besar. Mitra ISV menyediakan plug-in bernilai tambah yang dapat dengan mudah diintegrasikan ke dalam Platform dan dapat juga digunakan untuk membuat kaskas dan solusi sendiri.

Kedua produk adalah kode terbuka (*Open Source*) dan bebas (*free*) untuk penggunaan komersial dan non komersial. Kode sumber tersedia untuk guna ulang dengan lisensi *Common Development and Distribution License (CDDL)*. (www.netbeans.org, 29 Maret 2012).

2.5.5 Rational Rose Enterprise Edition

Rational Rose adalah *tools* pemodelan visual untuk pengembangan sistem berbasis obyek yang baik untuk digunakan sebagai bantuan bagi para pengembang dalam melakukan analisis dan perancangan sistem. *Rational rose* mendukung pemodelan bisnis yang membantu para pengembang memahami sistem secara komprehensif. *Rational rose* juga membantu analisis sistem dengan cara pengembang membuat *use case diagram* untuk melihat fungsionalitas sistem secara keseluruhan sesuai dengan harapan dan keinginan pengguna. *Rational rose* juga menuntut pengembang untuk mengembangkan *Interaction Diagram* untuk melihat bagaimana obyek-obyek saling bekerjasama dalam menyediakan fungsionalitas yang diperlukan.

Dalam *Rational rose*, pemodelan adalah cara melihat sistem dari berbagai sudut pandang. *Rational Rose* mencakup semua diagram yang dikenal dalam UML, *actor-actor* yang terlibat dalam sistem, *use-case*, obyek-obyek, *class-class*, komponen-komponen.

Model juga mendeskripsikan rincian yang diperlukan sistem dan bagaimana model akan bekerja, sehingga para pengembang dapat menggunakan model itu sebagai *blue print* untuk sistem yang akan dikembangkan.

Pemodelan yang dapat dibuat dengan rational rose adalah : *Use Case Diagram, Class Diagram, Sequence Diagram, Collaboration Diagram, State Diagram.*

2.5.6 Gammu 1.31.90

Gammu adalah nama proyek sama seperti utilitas untuk baris perintah, yang dapat Anda gunakan untuk mengontrol telepon / modem Anda. Gammu dibuat dalam Bahasa pemrograman C dan dibangun di atas *libGammu*.

Utilitas perintah Gammu menyediakan akses ke berbagai fitur telepon, namun tiap-tiap ponsel berbeda, Anda bisa cek di daftar telepon yang didukung oleh Gammu di (www.wammu.eu/phones). Umumnya mendukung fitur berikut ini:

- a. Daftar panggilan, memulai dan penanganannya
- b. Pengambilan SMS, backup dan pengiriman
- c. Pengambilan MMS
- d. Daftar buku telepon, ekspor dan impor (format standar seperti vCard)
- e. Kalender dan daftar tugas, ekspor dan impor (format standar seperti vCalendar atau iCalendar)
- f. Pengambilan informasi telepon dan jaringan
- g. Akses ke file sistem telepon (catatan bahwa beberapa ponsel bekerja sebagai perangkat penyimpanan USB dan tidak diakses melalui Gammu).

Proyek ini mulanya dari Gnokii yang telah mencapai ke versi 0,58 dengan sebutan MyGnokii2. Kemudian berubah nama menjadi Gammu yang dipilih sebagai singkatan dari (*Gammu All Mobile Management Utilities*), tanpa mengetahui bahwa Gammu dari "*Heretics*

of Dune" buku yang ditulis oleh Frank Herbert. Dalam Paket Gammu tidak hanya kode biner didalamnya, tetapi terdapat juga *Gammu SMS Daemon*, *Gammu library* dan *Python bindings* yang dapat anda gunakan untuk mengembangkan aplikasi anda untuk mengakses ponsel. (www.wammu.eu/gammu, 29 Maret 2012).

2.6 Pemodelan Sistem Menggunakan UML

UML merupakan salah satu alat bantu yang sangat handal dalam bidang pengembangan sistem berorientasi objek karena UML menyediakan bahasa pemodelan visual yang memungkinkan pengembang sistem membuat *blue print* atas visinya dalam bentuk yang baku. UML berfungsi sebagai jembatan dalam mengkomunikasikan beberapa aspek dalam sistem melalui sejumlah elemen grafis yang bisa dikombinasikan menjadi diagram. UML mempunyai banyak diagram yang dapat mengakomodasi berbagai sudut pandang dari suatu perangkat lunak yang akan dibangun. Diagram-diagram tersebut digunakan untuk (Henderi et al, 2008:71) :

- a. Mengkomunikasikan ide
- b. Melahirkan ide-ide baru dan peluang-peluang baru
- c. Menguji ide dan membuat prediksi
- d. Memahami struktur dan relasi-relasinya.

2.6.1 Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem, yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah *use case* merepresentasikan sebuah interaksi antara *actor* dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya *login* ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Seorang/sebuah *actor* adalah sebuah entitas manusia atau mesin yang

berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. (Dharwiyanti, Sri & W Satria, Romi. Ilmukomputer.com, 2003).

Use case mendeskripsikan interaksi antara para pengguna sistem itu sendiri, dengan memberi sebuah narasi tentang bagaimana sistem tersebut digunakan. Dua hal (atau istilah) yang tidak dapat dipisahkan dari *use case*, yakni *scenario* dan *actor*. *Scenario* adalah rangkaian langkah-langkah yang menjabarkan sebuah interaksi antara seorang pengguna dengan sebuah sistem.

Dimulai dengan mengambil salah satu *scenario* sebagai sekenario keberhasilan utama (MSS – *main success scenario*). Isi *use case* dibuat dengan menuliskan sekenario keberhasilan utama sebagai serangkaian langkah-langkah bernomor, kemudian diambil *scenario* lain dan ditulis sebagai ekstensi, hasilnya adalah variasi sekenario keberhasilan utama. Ekstensi-ekstensi tersebut dapat berupa keberhasilan pengguna mencapai keberhasilan.

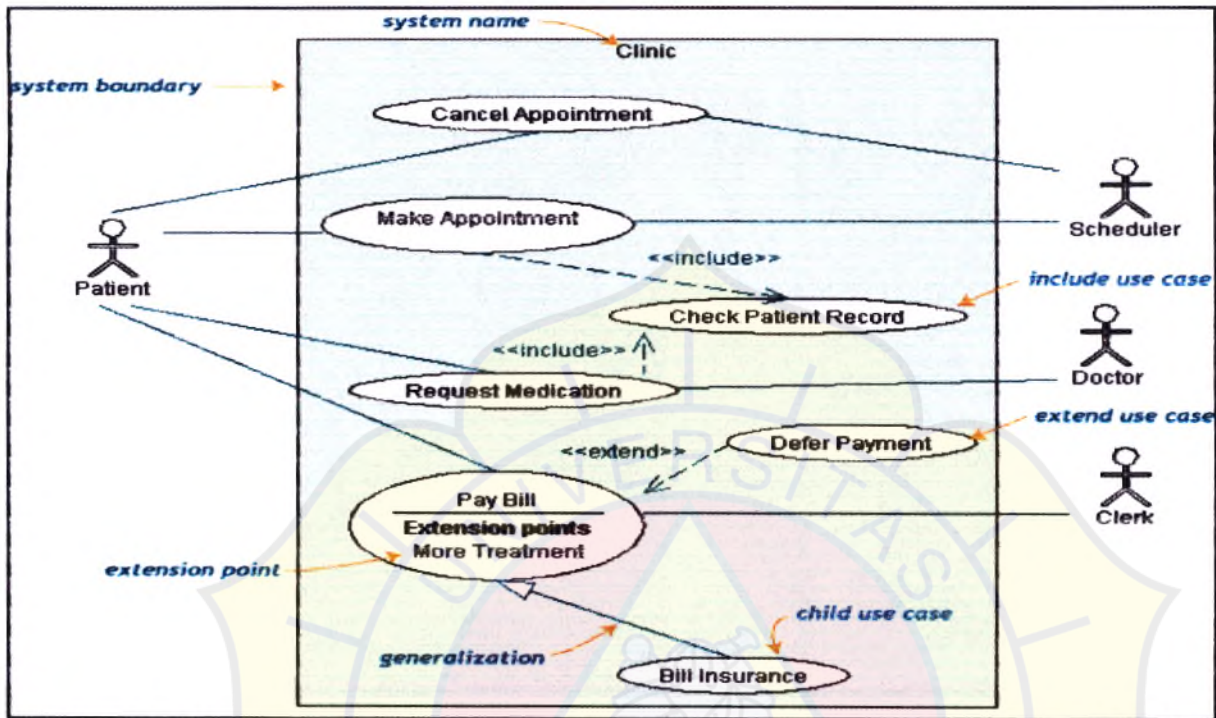
Sebuah ekstensi dalam sebuah *use case* menyebutkan sebuah kondisi yang menghasilkan interaksi berbeda dari yang disebutkan dalam MSS dan menyebutkan apa perbedaannya. Sebuah ekstensi dimulai dengan menunjuk langkah di mana kondisi tersebut terdeteksi dan tersedia sebuah gambaran singkat tentang kondisi tersebut. Kondisi tersebut kemudian diurutkan dengan langkah-langkah bernomor menggunakan model yang sama dengan MSS. Langkah-langkah ini diakhiri dengan kembali ke MSS selanjutnya.

Use case diagram dapat sangat membantu bila sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem.

Sebuah *use case* dapat meng-include fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-include akan dipanggil setiap kali *use case* yang meng-include dieksekusi secara normal.

Sebuah *use case* juga dapat meng-*extend use case* lain dengan *behaviour*-nya sendiri.

Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.



Gambar 2.5 Contoh *use case diagram* UML (Dharwiyanti,Sri & W Satria, Romi, 2003)

2.6.2 Activity Diagram

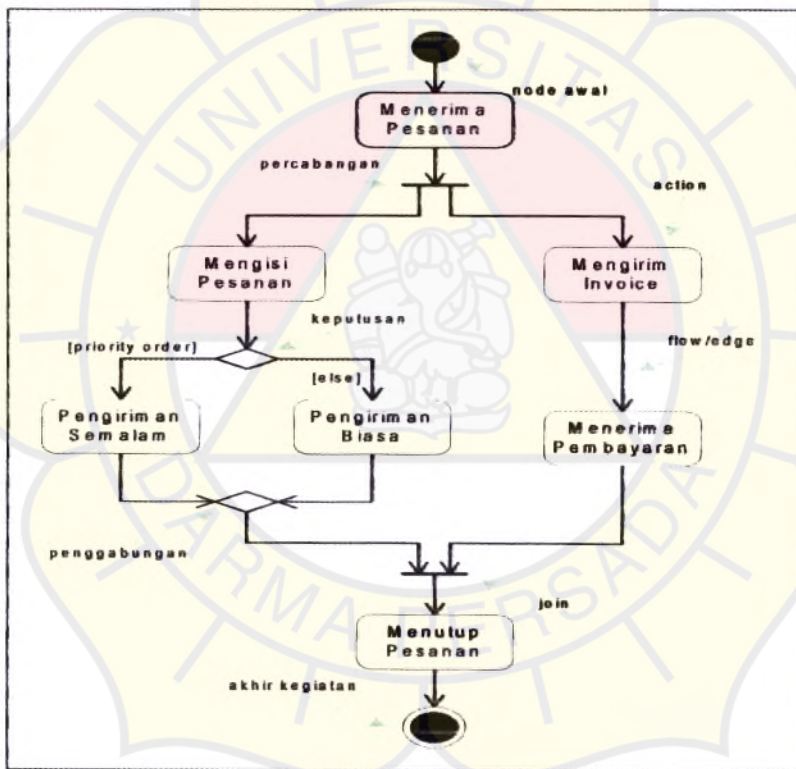
Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Activity diagram merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). *Activity diagram* lebih menggambarkan proses-proses dan jalur-jalur aktivitas secara umum. Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih.

Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana *actor* menggunakan sistem untuk melakukan aktivitas.

Standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Activity diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

UML 1, *activity diagram* dianggap sebagai kasus khusus *state diagram*. UML 2, *activity diagram* telah dikembangkan sehingga menjadi diagram yang benar-benar berdiri sendiri, tidak ditemukan lagi keterikatan dengan *state diagram*.



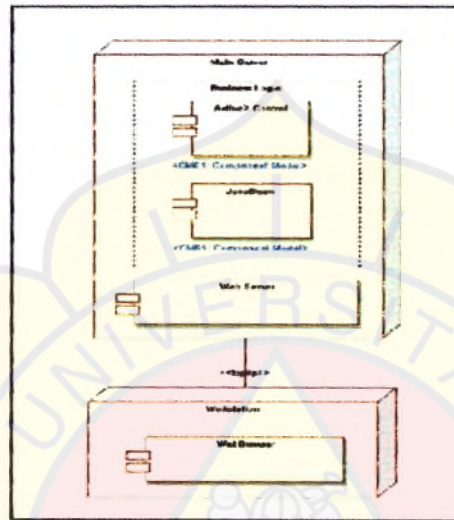
Gambar 2.6 Contoh *Activity diagram* UML (Dharwiyanti,Sri & W Satria, Romi, 2003)

2.6.3 Deployment Diagram

Deployment/physical diagram menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti

keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisik.

Sebuah *node* adalah *server*, *workstation*, atau piranti keras lain yang digunakan untuk *men-deploy* komponen dalam lingkungan sebenarnya. Hubungan antar *node* (misalnya TCP/IP) dan *requirement* dapat juga didefinisikan dalam diagram ini.

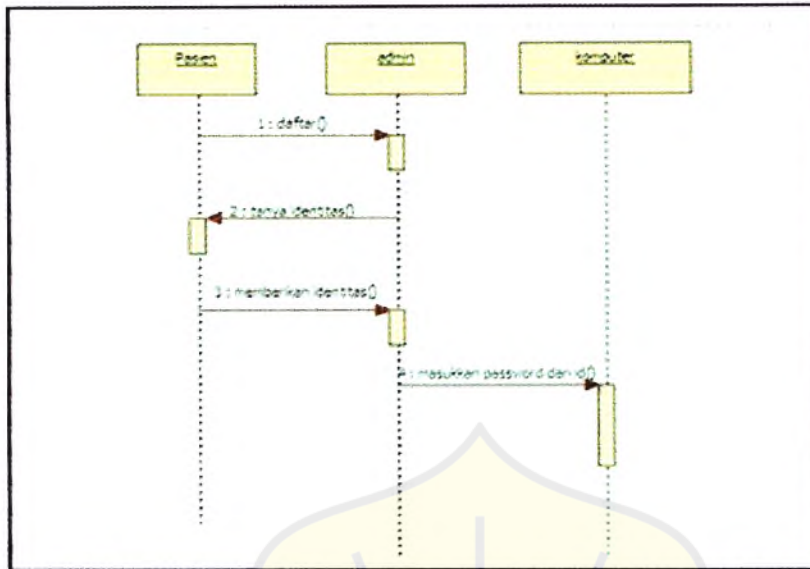


Gambar 2.7 Contoh *Deployment diagram* UML (Dharwiyanti,Sri & W Satria, Romi, 2003)

2.6.4 Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu. Sequence diagram terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang men-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan.



Gambar 2.8 Contoh *Sequence Diagram* UML (Dharwiyanti,Sri & W Satria, Romi, 2003)

Dari gambar diatas dapat disimpulkan bahwa sequence diagram tersebut merupakan proses pendataan pasien yang ada dalam suatu tempat. Tetapi si pasien ini harus mendaftarkan diri kepada admin yang menjaga tempat itu, setelah itu admin pun meminta untuk identitas diri yang kemudian admin memberikan password dan id untuk memproses suatu data pasiennya.