

BAB 2

LANDASAN TEORI

2.1 Teori Umum

Teori umum berisi teori-teori pokok yang digunakan sebagai landasan bagi teori-teori lainnya.

2.1.1 Internet

Menurut Yuhefizal (2008, p2) *Internet* adalah rangkaian hubungan jaringan komputer yang dapat diakses secara umum diseluruh dunia, yang mengirimkan data dalam bentuk paket data berdasarkan standar *Internet Protocol* (IP). Lebih dalam lagi, *Internet* adalah kumpulan jaringan dari jaringan-jaringan komputer dunia yang terdiri dari jutaan unit-unit kecil, seperti jaringan pendidikan, jaringan bisnis, jaringan pemerintahan dan lain-lain, yang secara bersama menyediakan layanan informasi seperti e-mail, online chat, transfer *file* dan saling keterhubungan (*linked*) antara satu halaman *web* dengan sumber halaman *web* yang lainnya.

2.1.2 Sejarah Internet

Menurut Simarmata (2010, p50) *Internet* pertama kali digunakan sebagai proyek penelitian yang ditemukan oleh *Advanced Research Project Agency* (ARPA) *Departement of Defense* (DOD) di Amerika Serikat. Pada dasarnya, *Internet* digunakan untuk menghubungkan komputer. Versi yang pertama disebut

ARPANET. Pada tahun 1972, ARPA berubah menjadi DARPA dengan tetap mempromosikan proyek ARPANET.

Pengembangan *Internet* dengan jenis peralatan yang berbeda, namun bisa saling berhubungan satu sama lain merupakan tantangan yang besar pada saat itu. Pada tahun 1973-1974, peneliti merancang sebuah *Transmission Control Protocol/ Internet Protocol* (TCP/IP). Pada awalnya, TCP/IP dimaksudkan untuk menyediakan dukungan untuk kebutuhan berikut:

- interoperabilitas antar sistem heterogen.
- komunikasi end-to-end berbagai jaringan yang berbeda, dan
- operasi otomatis dan sempurna di dalam menghadapi terjadinya kegagalan hubungan data.

Pada saat itu, aplikasi yang digunakan masih sangat sederhana daripada yang digunakan saat ini. Aplikasi yang paling banyak yang digunakan adalah *Telnet* untuk login remote dan FTP untuk perpindahan *file* dan e-mail.

Pada tahun 1980-an, ARPANET dipecah menjadi dua bagian yaitu MILNET dan ARPANET karena pertimbangan keamanan. Pihak militer berjalan terus dengan MILNET, sedangkan penelitian, pengembangan, dan sektor lain tetap memakai ARPANET. Pada pertengahan tahun 1980-an, *National Science Foundation* (NTF) di Washington, DC. Mendistribusikan teknologi *Internet* kepada beberapa universitas (Berkeley, MIT, Stanford, dan UCLA). Selanjutnya, *Internet* pun mulai menyebar di dunia.

Pada tahun 1990, DOD memutuskan untuk membubarkan ARPANET dan menggantinya dengan pendukung (*backbone*) NSFNET, bekerja sama

dengan agen jaringan *Internet*. Tabel berikut adalah statistik populasi dan penggunaan *Internet* dunia.

Tabel 2.1 Tabel populasi pengguna *internet*

Wilayah	Populasi (2008)	Pengguna Internet, 31 Desember 2000	Pemakaian Internet, Data terakhir	% Populasi (Penetra-si)	Pemakaian % dunia	Pertumbuhan pemakaian 2000-2008
Afrika	955.206.348	4.514.400	51.065.630	5,3 %	3,5 %	1.031,2 %
Asia	3.776.181.949	114.304.000	578.538.257	15,3 %	39,5 %	406,1 %
Eropa	800.401.065	105.096.093	384.633.765	48,1 %	26,3 %	266,0 %
Timur Tengah	197.090.443	3.284.800	41.939.200	21,3 %	2,9 %	1.176,8 %
Amerika Utara	337.167.248	108.096.800	248.241.969	73,6 %	17 %	129,6 %
Amerika Latin/Karibia	576.091.673	18.068.919	139.009.209	24,1 %	9,5 %	669,3 %
Oceania/Australia	33.981.562	7.620.480	20.204.331	59,5 %	1,4 %	165,1 %
TOTAL DUNIA	6.676.120.288	360.985.492	1.463.632.361	21,9 %	100 %	305,5 %

(Sumber: <http://www.internetworldstats.com/stats.htm>)

2.1.3 World Wide Web (www)

Menurut Mooduto dan Hidayat (2009,p1) mengatakan sebuah sistem penyebaran informasi melalui *internet*. Informasi yang dikirimkan dapat berupa teks, suara (audio), animasi, gambar dan bahkan dalam format video yang dapat diakses melalui sebuah software yang disebut *browser*, seperti *internet explorer*, mozilla firefox, opera dan lain-lain.

2.1.4 Pengertian *Website*

Menurut Mooduto dan Hidayat (2009, p2) mengatakan *website* adalah keseluruhan halaman-halaman *web* yang terdapat dalam sebuah domain yang mengandung informasi. Sebuah *website* biasanya dibangun atas banyak halaman *web* yang saling berhubungan, Hubungan antara satu halaman *web* dengan halaman *web* yang lainnya disebut dengan hyperlink, sedangkan teks yang dijadikan media penghubung disebut *hypertext*.

Istilah lain yang sering disebut ditemui sehubungan dengan *website* adalah *homepage*. *Homepage* adalah halaman awal sebuah domain. Misalnya membuka *website* www.binus.ac.id, halaman pertama yang muncul disebut dengan *homepage*, jika meng-klik menu-menu yang ada dan meloncat ke lokasi yang lainnya, disebut *web page*, sedangkan keseluruhan isi/conten domain disebut *website*.

2.1.5 Jenis-Jenis *Website*

Seiringan dengan perkembangan teknologi informasi yang begitu cepat, *website* juga mengalami perkembangan yang sangat ber-arti. Dalam pengelompokan jenis *web*, lebih diarahkan berdasarkan pada fungsi, sifat dan bahasa pemrograman yang digunakan.

Menurut Mooduto dan Hidayat (2009,p3) mengatakan jenis-jenis *web* berdasarkan sifatnya adalah:

- *Website dinamis*, merupakan sebuah *website* yang menyediakan content atau isi yang selalu berubah-ubah setiap saat. Misalnya *website* berita, seperti, www.kompas.com, www.detik.com, dan lain-lain.

- -*Website statis*, merupakan *website* yang contentnya sangat jarang diubah. Misalnya, *web profile* organisasi, dan lain-lain.

Menurut Mooduto dan Hidayat (2009,p3) mengatakan berdasarkan tujuannya, *website* dibagi atas:

- *Personal web*, *website* yang berisi informasi pribadi seseorang.
- *Corporate web*, *website* yang dimiliki oleh sebuah perusahaan.
- *Portal web*, *website* yang mempunyai banyak layanan, mulai dari layanan berita, email, dan jasa-jasa lain.
- *Forum web*, sebuah *web* yang bertujuan sebagai media diskusi.
- Di samping itu juga ada *website* e-Government, e-Banking, e-Payment, e-Procurement, dan sebagainya.

Menurut Mooduto dan Hidayat (2009,p3) mengatakan ditinjau dari segi bahasa pemrograman yang digunakan, *website* terbagi atas:

- ✓ *Server side*, merupakan *website* yang menggunakan bahasa pemrograman yang tergantung kepada tersedia server. Seperti PHP,ASP dan lain sebagainya. Jika tidak ada server, *website* yang dibangun menggunakan bahasa pemrograman di atas tidak dapat berfungsi sebagaimana mestinya.
- ✓ *Client Side*, adalah *website* yang tidak membutuhkan server dalam menjalankannya, cukup diakses melalui *browser* saja. Misalnya, html.

2.1.6 Interaksi Manusia Komputer (IMK)

Interaksi manusia dan komputer (*human-computer interaction*, HCI) adalah disiplin ilmu yang mempelajari hubungan antara manusia dan komputer yang meliputi perancangan, evaluasi, dan implementasi antarmuka pengguna komputer agar mudah digunakan oleh manusia. Sedangkan interaksi manusia dan komputer sendiri adalah serangkaian proses, dialog dan kegiatan yang dilakukan oleh manusia untuk berinteraksi dengan komputer yang keduanya saling memberikan masukan dan umpan balik melalui sebuah antarmuka untuk memperoleh hasil akhir yang diharapkan.

Menurut Shneiderman (2010, p88-89), sebuah sistem berbasis komputer selalu mempunyai elemen manusia yang terkait di dalamnya. Elemen manusia ini berinteraksi langsung dengan perangkat keras dan perangkat lunak melalui sebuah antarmuka (*interface*).

Penggunaan delapan aturan emas berasal dari prinsip – prinsip yang didapatkan dari pengalaman – pengalaman, dan direvisi selama lebih dari 2 dekade. Delapan aturan emas tersebut adalah :

- **Strive for consistent**

Urutan aksis yang konsisten diperlukan pada situasi yang sama. Konsistensi juga harus digunakan pada *prompt*, *menu*, layar bantu. Warna tampilan, kapitalisasi, huruf dan sebagainya juga harus konsisten.

- **Cater to universal usability**

Mengenali dan mengkategorikan kebutuhan pengguna yang beragam, memfasilitasi transformasi konten. Menambahkan *feature* untuk pemula seperti penjelasan dan untuk pemakai yang sudah ahli seperti shortcuts

dan faster pacing sehingga bisa memperkaya design *interface* dan meningkatkan kualitas sistem.

- **Offer informative feedback**

Untuk setiap tindakan yang dilakukan pemakai, diharapkan adanya respon dari sistem. Untuk tindakan yang sering terjadi dan tidak membutuhkan banyak aksi, respon yang ada bisa dibuat secara sederhana, sedangkan tindakan yang jarang dilakukan dan membutuhkan banyak aksi harus lebih ditonjolkan.

- **Design dialogs to yield closure**

Urutan aksi harus disusun ke dalam kelompok awal, tengah dan akhir. Suatu umpan balik yang informatif pada akhir pekerjaan sebaiknya dibuat untuk mengindikasikan bahwa pekerjaan telah selesai dan siap untuk melanjutkan ke aksi berikutnya.

- **Prevent errors**

Sebisa mungkin dibuat sistem yang tidak akan menghasilkan kesalahan yang serius jika digunakan oleh pemakai. Jika pemakai membuat kesalahan, sistem tersebut harus dapat mendeteksi kesalahan dan menawarkan instruksi yang sederhana, konstruktif, dan spesifik untuk perbaikan.

- **Permit easy reversal of actions**

Aksi yang telah dilakukan harus dapat dikembalikan ke keadaan awal.

- **Support internal locus of control**

Menjadikan pemakai sebagai pemberi aksi, bukan sebagai orang yang merespon aksi.

- **Reduce short-term memory load**

Karena keterbatasan manusia dalam pemrosesan informasi pada ingatan jangka pendek, sebaiknya tampilan dibuat sederhana, beberapa halaman dijadikan satu, frekuensi pergerakan window dikurangi, dan harus ada waktu yang cukup untuk mempelajari kode-kode, singkatan, serta urutan aksi. Informasi seperti singkatan atau kode sebaiknya juga tersedia.

2.1.7 Basis Data

Basis data (*database*), atau sering pula dieja basisdata, adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Perangkat lunak yang digunakan untuk mengelola dan memanggil kueri (*query*) basis data disebut sistem manajemen basis data (DBMS).

Istilah "basis data" berawal dari ilmu komputer. Meskipun kemudian artinya semakin luas, memasukkan hal-hal di luar bidang elektronika, artikel ini mengenai basis data komputer. Catatan yang mirip dengan basis data sebenarnya sudah ada sebelum revolusi industri yaitu dalam bentuk buku besar, kuitansi dan kumpulan data yang berhubungan dengan bisnis.

Menurut Connolly (2010, p15) basis data adalah sekumpulan data yang terhubung satu sama lain secara logika dan suatu deskripsi data yang dirancang untuk memenuhi kebutuhan informasi suatu organisasi.

Keuntungan basis data, yaitu:

- Kemampuan untuk menggunakan data yang sama di banyak aplikasi dan sistem.
- Penyimpanan data dalam format yang fleksibel. Hal ini memungkinkan karena basis data didefinisikan secara terpisah dari sistem informasi dan program – program aplikasi yang akan menggunakan basis data.
- Teknologi basis data menyediakan skalabilitas superior, dalam arti basis data dan sistem yang menggunakannya dapat ditingkatkan atau dikembangkan untuk memenuhi kebutuhan – kebutuhan perubahan pada sebuah organisasi.
- Kemajuan independensi data yang sangat mengurangi redundansi data, telah meningkatkan fleksibilitas.

a) Data Definition Language (DDL)

Definisi dari *Data Definition Language* menurut Connolly (2010, p40) adalah suatu bahasa yang memungkinkan *Database Administrator* (DBA) atau pengguna untuk mendeskripsikan nama dari suatu entitas, atribut, dan relasi data yang diminta oleh aplikasi, bersamaan dengan integritas data dan batasan keamanan datanya.

b) Data Manipulation Language (DML)

DML menurut Connolly (2010, p40) adalah suatu bahasa yang memberikan fasilitas pengoperasian data yang ada di dalam basis data.

Pengoperasian data yang akan dimanipulasi meliputi :

1. Penambahan data baru ke dalam basis data.
2. Modifikasi data yang disimpan ke dalam basis data.
3. Pemanggilan data yang terdapat di dalam basis data.
4. Penghapusan data dari basis data.

Sedangkan definisi *Procedural* DML menurut Connolly (2010,p41) adalah suatu bahasa yang memperbolehkan pengguna untuk mendeskripsikan ke dalam sistem data apa yang dibutuhkan dan bagaimana mendapatkan data tersebut.

2.1.8 Unified Modeling Language (UML)

Menurut Whitten (2007, p408) UML adalah suatu konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem piranti lunak yang terkait dengan objek.

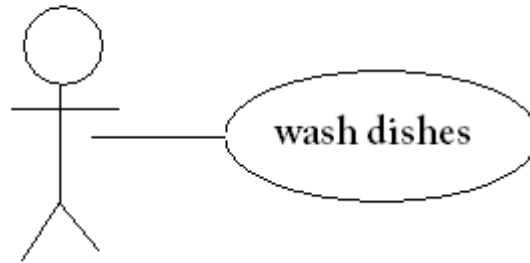
UML terdiri dari berbagai tipe diagram, antara lain : (Joseph Schmuller, 2004)

a) Use Case Diagram

Use Case adalah sebuah deskripsi dari tingkah laku sebuah sistem yang dilihat dari sudut pandang pengguna. Bagi para pengembang ini adalah *tool* yang sangat berharga : sebuah teknik *tried – and – true* yang dipakai untuk mengumpulkan persyaratan sebuah sistem yang dilihat dari sudut pandang pengguna. Teknik pengumpulan tersebut penting jika

tujuannya adalah untuk membangun sebuah sistem yang bisa digunakan semua orang (bukan hanya orang yang terbiasa dengan komputer).

Berikut adalah sebuah contoh sederhana *Use Case* diagram dari, penggunaan mesin pencuci piring



Penggunaan Mesin Cuci Piring

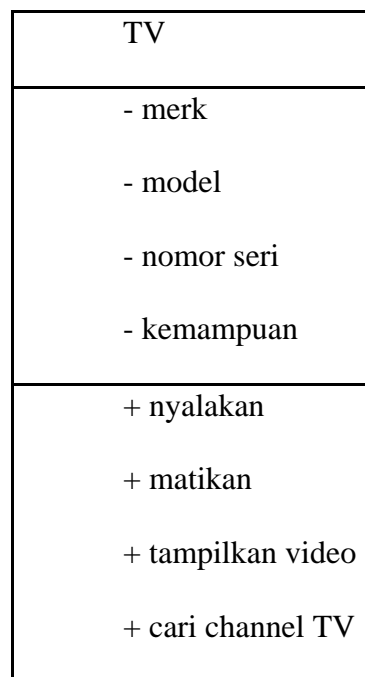
Gambar 2.1 Contoh USE CASE Diagram

b) *Class Diagram*

Class Diagram merupakan presentasi grafik dari *static view* yang menampilkan sekumpulan model elemen deklaratif, seperti *class*, *type*, beserta dengan isi dan hubungannya. *Class Diagram* bisa berisi tampilan dari suatu *package* dan bisa berisi simbol untuk *nested package*. *Class Diagram* berisi elemen tingkah laku tertentu, seperti operasi, Sebuah *class* dapat berhubungan dengan *class* lain melalui garis penghubung yang dikenal sebagai asosiasi. Sebuah *class* adalah sebuah kategori atau grup atas beberapa hal yang mempunyai atribut yang mirip dan tingkah laku yang umum. Berikut ini adalah satu contoh : semua yang ada di dalam *class* dari sebuah TV mempunyai atribut seperti merk, model,

nomor seri, dan kemampuan. Tingkah laku *class* ini meliputi operasi “nyalakan”, “matikan”, “menampilkan video”, dan “cari channel TV”.

Gambar berikut ini menunjukkan sebuah contoh yang menggambarkan atribut dan tingkah laku dari sebuah TV. Sebuah segi empat adalah ikon yang merepresentasikan *class* dan terbagi menjadi tiga area. Area teratas terdiri dari nama. Area di tengah berisi atribut – atribut. Area terbawah menunjukkan operasi – operasi. Sebuah *Class Diagram* terdiri dari beberapa segi empat tersebut, yang dihubungkan dengan garis yang menunjukkan bagaimana *class-class* tersebut saling berhubungan.



Gambar 2.2 Contoh Class Diagram

Untuk berhubungan dengan dunia yang kompleks ini, kebanyakan *software* modern mensimulasikan beberapa aspek dari dunia. Pengalaman puluhan tahun menyarankan bahwa hal termudah untuk mengembangkan

software untuk *Class Diagram* adalah saat *software* tersebut merepresentasikan *class –class* dari hal – hal di dunia nyata. *Class Diagram* menyediakan representasi dari tempat kerja pengembang.

c) *Activity Diagram*

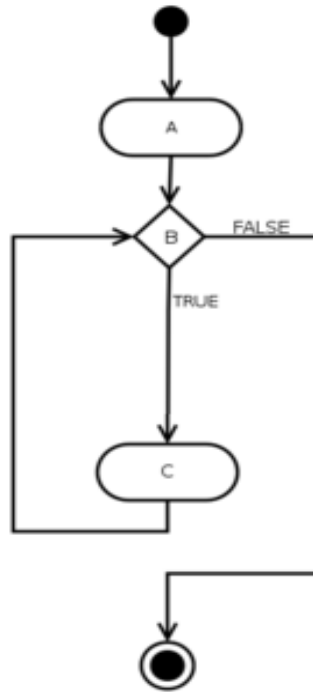
Activity Diagram merupakan notasi untuk *activity graph*, yang mencakup beberapa simbol *shorthand* yang mudah digunakan. Simbol – simbol ini biasa digunakan pada *Statechart Diagram* manapun, walaupun kadang simbol – simbol yang tercampur tidak enak dipandang mata.

State dari *Activity Diagram* digambarkan dengan kotak berakhiran bundar atau membulat yang berisi deskripsi dari aktivitas yang ada. Transisi lengkap digambarkan dengan anak panah (*arrow*). *Branch* digambarkan sebagai kondisi penjaga pada transisi atau dengan belah ketupat (*diamond*) yang mempunyai banyak anak panah sebagai petunjuk akhir. *Fork* atau *join* digambarkan sama dengan yang ada di *Statechart*, yaitu dengan *multiple arrow* yang memasuki atau meninggalkan bar sinkronisasi.

Untuk situasi di mana *event* eksternal harus dimasukkan, penerimaan *event* biasa digambarkan sebagai *trigger* (pemicu) pada transisi atau sebagai simbol *inline* khusus yang menandakan adanya sinyal yang ditunggu. Notasi yang hampir sama digunakan untuk mengirim sinyal tersebut. Jika terdapat beberapa transisi yang dipicu oleh *event*, *Statechart Diagram* umum biasanya lebih cocok untuk dipakai.

Kadangkala pengaturan aktivitas dalam model disesuaikan dengan penggunaan yang bertanggung jawab, misalnya dengan mengelompokkan semua aktivitas yang ditangani oleh 1 organisasi secara bersama – sama. Hal seperti ini bisa digambarkan dengan mengatur aktivitas - aktivitas ke dalam wilayah terpisah yang dipisahkan oleh sebuah garis lurus di dalam diagram. Pada tampilan ini, tiap wilayah disebut *swimlane*.

Sebuah *Activity Diagram* bisa menunjukkan aliran nilai objek, seperti halnya dengan aliran control. *Object flow state* (aliran state object) menggambarkan objek yang merupakan *input* atau *output* dari suatu aktivitas. Untuk nilai *output*, anak panah putus – putus digambarkan dari *activity* ke *object flow state*. Untuk nilai *input*, garis putus – putus digambarkan dari *object flow state* ke *activity*. Jika aktivitas punya lebih dari satu nilai *output* atau penerus dari *control flow*, maka anak panah digambarkan dari simbol *fork*. Sama halnya dengan *multiple input* yang digambarkan menuju simbol *join*.



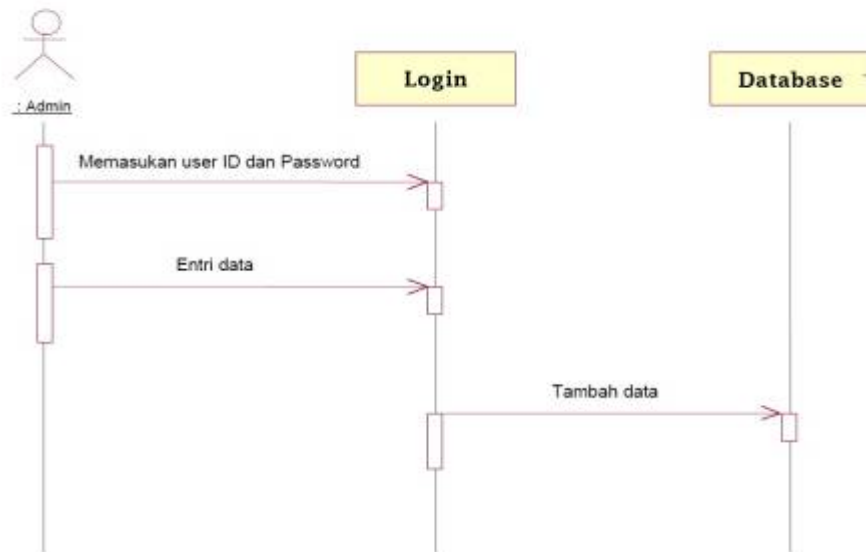
Gambar 2.3 Contoh Activity Diagram

d) Sequence Diagram

Sequence Diagram digambarkan sebagai interaksi dari 2 grafik dimensional. Dimensi vertikal merupakan sumbu waktu, yang berjalan makin ke bawah. Dimensi horizontal menggambarkan peranan *classifier* yang menampilkan objek individual di dalam kolaborasi. Tiap peranan *classifier* digambarkan dengan kolom vertikal – sebagai *lifeline*. Dalam waktu selama objek ada, peranan dari objek tersebut digambarkan dengan garis putus – putus. Selama aktivitas prosedur pada sebuah objek masih aktif, *lifeline* digambarkan dengan garis ganda (*double line*).

Pesan ditunjukkan dengan gambar anak panah (*arrow*) dari *lifeline* suatu objek menuju objek yang lainnya. Anak panah tersebut diatur menurun sesuai urutan waktu di dalam diagram.

Gambar berikut ini menunjukkan sebuah *Sequence Diagram* yang menjelaskan seorang admin memasukkan data ke dalam *database*. Waktu dalam diagram ini berlangsung dari atas ke bawah.



Gambar 2.4 Contoh Sequence Diagram

2.1.9 PHP Hypertext Preprocessor (PHP)

Menurut Welling dan Thomson (2008, p4) PHP (PHP *Hypertext Preprocessor*) adalah bahasa pemrograman berbasis server yang dirancang khusus untuk *web*. Dalam halaman HTML, dapat dimasukkan kode PHP yang akan dieksekusi setiap kali halaman *web* tersebut diakses. Kode PHP ini akan diterjemahkan oleh *web* server dan akan dijalankan bersamaan dengan HTML atau *output* lainnya, yang akan dilihat oleh pengunjung situs *web*.

Beberapa kelebihan PHP dibandingkan dengan bahasa pemrograman sejenis seperti Perl, Microsoft Active Server *Pages* (ASP), Java Server *Pages* (JSP), dan Allaire Cold Fusion adalah:

- Kemampuan yang tinggi.
- Kemampuan untuk dapat terhubung dengan banyak sistem basis data, seperti : MySQL, PostgreSQL, mSQL, Oracle, dbm, *filePro*, Hyperwave Informix, dan Interbase.
- Tidak dibutuhkan biaya untuk mendapatkan PHP.
- Mudah dipelajari dan digunakan, karena PHP dibuat berdasarkan bahasa pemrograman dasar, yaitu bahasa C dan Perl.
- Dapat berjalan pada berbagai system operasi, seperti Linux, Solaris, dan berbagai versi Microsoft Windows.

2.1.10 Framework PRADO

Di dalam pengembangan perangkat lunak, suatu framework digambarkan suatu struktur pendukung dimana perancangan lunak yang lain dapat terorganisir dan dikembangkan. Suatu framework dapat dilengkapi program pendukung, kumpulan kode-kode program (libraries), suatu bahasa scripting, atau perangkat lunak lain untuk membantu mengembangkan dan menggabungkan komponen-komponen yang berbeda menjadi satu dari suatu perancangan perangkat lunak. Prado adalah sebuah framework pemrograman berbasis komponen dan event driven untuk pengembangan aplikasi *web* pada PHP 5. PRADO merupakan singkatan dari PHP Rapid Application Development Object-oriented.

Framework ini dibuat oleh Qiang Xue dan telah menjadi pemenang dalam Zend PHP 5 Coding Contest.

Konsep Prado yang component-based dan event-driven member banyak keuntungan bagi programmer *web*. Berikut keuntungan dengan menggunakan Prado :

- a) *Reusability*, komponen-komponen dalam Prado dapat digunakan ulang.
- b) *Ease of Use*, komponen-komponen dalam Prado sangat mudah
- c) digunakan. Komponen juga dapat dibuat sendiri dengan menurunkan kelas yang sudah ada sesuai dengan kebutuhan.
- d) *Robustness*, Prado membebaskan pengembangan program (developer) dari kejenuhan dalam menulis kode-kode. Semua kode ditulis dalam format objek, method, dan property. Tidak seperti pemrograman PHP yang biasa digunakan.
- e) *Performance*, Prado menggunakan teknik cache untuk menjamin performa aplikasi. Dengan adanya cache ini, ia tidak perlu mem-parser ulang kode XML (eXtensible Markup Language) yang dibuat.
- f) *Team Integration*, Prado memisahkan business logic dan presentation logic. Yang dimaksudkan adalah pembuatan layout tampilan (template) dengan kode program (class). Pembuatan keduanya dilakukan pada *file* yang terpisah. Dengan demikian, aplikasi berbasis Prado dapat dilakukan dalam sebuah tim dengan personal yang berbeda.

Untuk melakukan koneksi ke database Prado memanfaatkan database abstract layer, ADOdb. ADOdb adalah class yang ditulis menggunakan bahasa PHP yang berfungsi sebagai data tier, dan akan membantu mengatasi perbedaan

antara penggunaan database. Cukup dengan menuliskan sebuah kode, maka koneksi dapat dilakukan ke berbagai macam database seperti MySQL, SQLite, SQL Server, Oracle, DB2, Interbase, PostgreSQL, dan sebagainya.

2.1.11 MySQL

Menurut Welling dan Thomson (2008, p3 - 4) MySQL berarti sistem manajemen hubungan antar basis data yang sangat cepat dan sempurna. MySQL merupakan alat bantu untuk manipulasi basis data, sehingga basis data dapat dengan mudah diisi, diambil, disusun dan diubah datnya. Server MySQL pun dapat mengatur kontrol akses dari data, sehingga beberapa pengguna dapat sekaligus bekerja pada waktu yang bersamaan.

Beberapa kelebihan MySQL, dibandingkan dengan sistem basis data sejenis seperti Microsoft SQL Server, Oracle:

- Kemampuan yang tinggi.
- Tidak dibutuhkan biaya untuk mendapatkan MySQL.
- Mudah untuk konfigurasi dan dipelajari.
- Dapat dijalankan pada beberapa sistem operasi seperti sistem Unix dan Microsoft Windows.

2.1.12 Model Proses Software

Menurut Pressman (2010, p78), ada beberapa model proses software yang umum digunakan, salah satunya adalah Model Sekuensial Linear Model

Sekuensial Linear ini juga dikenal dengan nama “*ClassicLife Cycle*” atau “*Waterfall Model*”. Model ini melingkupi aktivitas-aktivitas sebagai berikut :

1. Communication

Proses pengumpulan kebutuhan diintensifkan dan difokuskan, khususnya pada perangkat lunak. Untuk memahami sifat program yang dibangun, seorang perancang perangkat lunak harus memahami kebutuhan informasi, fungsi-fungsi, unjuk kerja, dan interface yang diperlukan.

2. Planning

Design perangkat lunak sebenarnya adalah proses multi langkah yang berfokus pada empat attribute sebuah program yang berbeda : struktur data, arsitektur perangkat lunak, representasi *interface*, dan detail (algoritma).

3. Modelling.

Design harus diterjemahkan menjadi bentuk yang bisa dibaca atau dimengerti oleh komputer, biasanya dalam bentuk bahasa pemrograman.

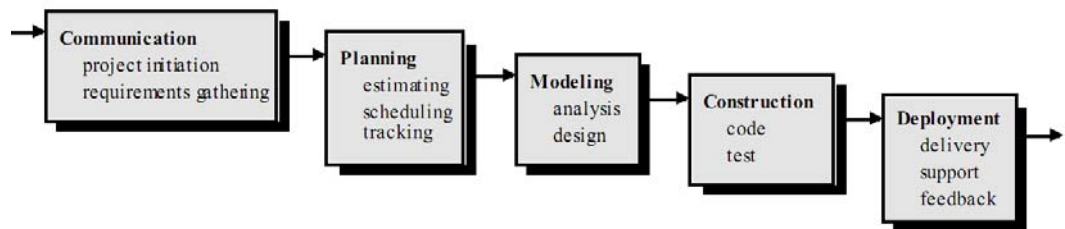
4. Construction.

Sekali kode dibuat, pengujian program juga dimulai. Proses pengujian berfokus pada logika internal perangkat lunak, memastikan bahwa semua pernyataan sudah diuji, dan pada eksternal fungsional yaitu mengarahkan pengujian untuk menemukan kesalahan-kesalahan dan memastikan bahwa input yang dibatasi akan memberikan hasil aktual yang sesuai dengan hasil yang dibutuhkan.

5. Deployment.

Perangkat lunak akan mengalami perubahan setelah disampaikan kepada pelanggan. Perubahan akan terjadi karena kesalahan²⁸ kesalahan ditentukan,

karena perangkat lunak harus disesuaikan untuk mengakomodasikan perubahan-perubahan didalam lingkungan eksternalnya, atau pelanggan membutuhkan perkembangan fungsional atau untuk kerja. Pemeliharaan perangkat lunak mengaplikasi lagi setiap fase program sebelumnya dan tidak membuat yang baru lagi.

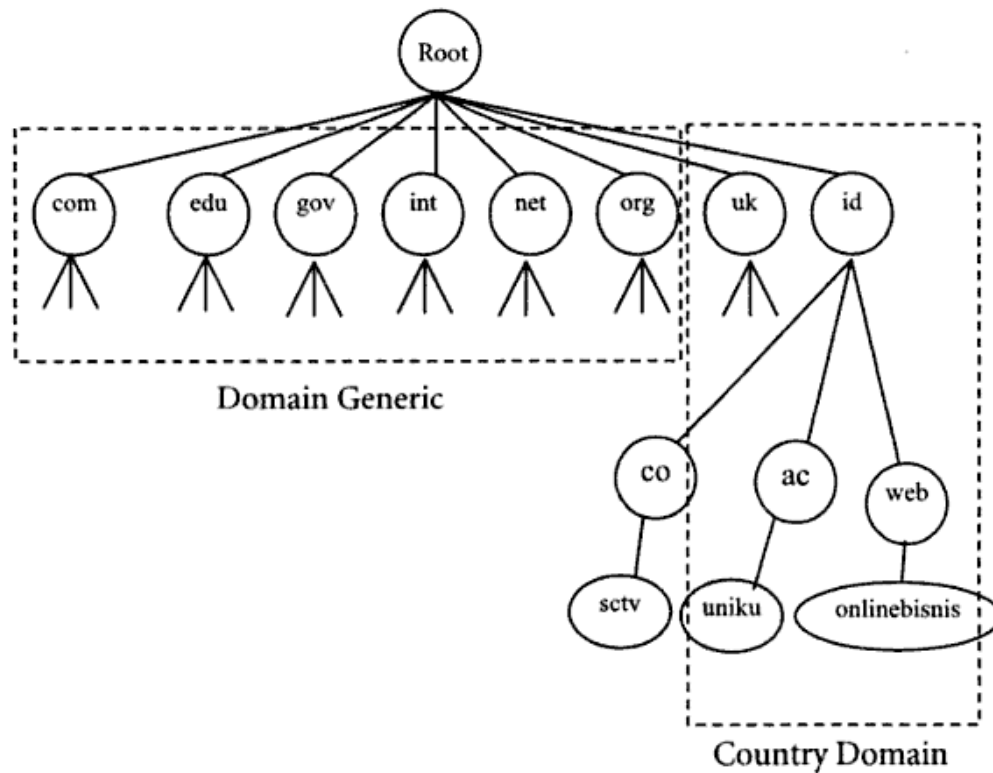


Gambar 2.5 System Developing Lifecycle

2.2 Teori Khusus

2.2.1 Pengertian *domain* dan *subdomain*

Menurut Suryana (2008,p9) *domain* pada umumnya dikenal identik dengan alamat *url* atau *website*, sebagai contoh *domain blog* bisnis penulis: <http://onlinebisnis.web.id>, *domain* MataMaya studio beralamat di <http://mata-maya.com>, dan seterusnya. Tetapi sebenarnya secara teoritis *domain* dapat berupa nama *host*, *subdomain*, atau *top level domain* (TLD). Penamaan *host* (komputer yang terhubung ke jaringan) memiliki format dan aturan tersendiri yang menggambarkan suatu hierarki, kumpulan hierarki ini disebut *domain name space*.



Gambar 2.6 *Domain name space*

Dari gambar di atas, *domain* paling atas tingkatannya adalah *root domain* dilambangkan dengan tanda titik, dan semua titik tepat di bawah *root domain* disebut *top level domain* atau sering dikenal dengan istilah TLD, pada gambar di atas yang tergolong TLD yaitu com, edu, gov, int, net, uk, id, dan seterusnya. *Top level domain* digunakan untuk menunjukkan jenis perusahaan atau lembaga atau negara tempat *host/komputer server* berada. TLD terdiri dari TLD Generik dan TLD Negara

Tabel 2.2 *Tabel domain*

<i>Domain</i>	<i>Keterangan</i>

go.id	Untuk lembaga pemerintahan, misalnya: kuningankab.go.id
co.id	Untuk lembaga komersil, misalnya: sctv.co.id
ac.id	Untuk lembaga pendidikan tinggi dan universitas misalnya: binus.ac.id
sch.id	Untuk lembaga pendidikan dasar dan menengah, misalnya: smkn2-kng.sch.id
net.id	Untuk perusahaan penyedia jasa <i>internet</i> , misalnya: andalan.net
or.id	Untuk lembaga non komersial dan LSM
web.id	Untuk perorangan maupun kelompok yang sifatnya lebih luas, misalnya: onlinebisnis.web.id

Untuk memperoleh *domain* bisa dilakukan melalui pendaftaran *domain* melalui pengelola *domain*.

Menurut Suryana (2008,p12), selain *domain*, dikenal juga *subdomain* adalah *domain* yang berada di bawah satu *domain*, jadi satu *domain* bisa memiliki beberapa *subdomain*. Kalau dianalogikan dalam *file manager*, *domain* merupakan sebuah *direktori*, dan *subdomain* merupakan *subdirektori*.

Tabel 2.3 Tabel *subdomain*

<i>Domain</i>	onlinebisnis.web.id
<i>Subdomain</i>	
kios_pulsa.onlinebisnis.web.id	<i>Subdomain</i> dari onlinebisnis.web.id, dimana kontennya tersebut mengkhususkan mempromosikan bisnis pulsa online.

simse.onlinebisnis.web.id	<i>Subdomain</i> dari onlinebisnis.web.id, dimana kontennya tersebut mengkhususkan mempromosikan jual beli Sistem Informasi Manajemen Sekolah.
jual_software.onlinebisnis.web.id	<i>Subdomain</i> dari onlinebisnis.web.id, dimana kontennya tersebut mengkhususkan mempromosikan penjualan program.

Untuk memperoleh *subdomain* tidak seperti pada saat ingin memiliki *domain*, dimana kalau ingin memiliki *domain*, maka kita bisa sesuka hati membuat *subdomain* sesuai dengan kebutuhan.

2.2.2 Pengertian Web Hosting

Menurut Suryana (2008,p13) *Web hosting* adalah jasa penyedia *hosting* di *internet*, tugas mereka adalah menyediakan dan mengelola ruang *harddisk* yang akan disewa oleh pemilik suatu *domain* untuk menyimpan *file-filenya* agar halaman *web* si pemilik *domain* dapat ditampilkan di *browser internet*. Pada dasarnya perusahaan *web hosting* adalah lembaga bisnis yang menyediakan komputer server yang melayani banyak pemilik *domain*.

2.2.3 Definisi Software Engineer

Menurut Timothy dan Robert (2005,P5) *Software Engineer* adalah seseorang yang memiliki pendidikan, dan pengalaman kinerja dalam proses pemecahan masalah yang dialami oleh pelanggan dengan pembangunan secara

sistematis dan evolusi besar, aplikasi berkualitas tinggi membutuhkan waktu, biaya dan kendala lainnya

2.2.4 Stakeholders di software engineering

Stakeholder di *software engineering* dibagi menjadi 4 kategori yaitu (Timothy and Robert,2005) :

- Pemakai (*Users*)

Seseorang yang menggunakan aplikasi (perangkat lunak).

- Konsumen (*Customers*)

Seseorang yang membuat diskusi tentang pemesanan dan pembayaran tentang aplikasi (perangkat lunak). konsumen membeli perangkat lunak untuk menambah pendapatan atau membuat simpel untuk menjalankan bisnis mereka supaya lebih efektif. Konsumen mungkin juga bisa dikatakan sebagai pemakai.

- *Software developers*

Seseorang yang mengembangkan dan membuat aplikasi (perangkat lunak). Beberapa dari mereka dipanggil dengan nama *software engineers*. Dalam tim *software developers* terdapat perancang desain, spesialis database, manajemen, spesialis coding, dan lain-lain. Mereka bekerja sama dalam tim untuk mendapatkan hasil maksimal yang diminta oleh konsumen.

- *Development managers*.

Seseorang yang bekerja di organisasi tersebut untuk mengembangkan aplikasi yang diminta oleh konsumen

2.2.5 Electronic commerce

Menurut Standart International yang ditentukan oleh Internarional Trade Centre (2009,p12) menyatakan *Electronic commerce*, yang diketahui juga sebagai EC, *e-Commerce* ataupun *ecommerce*, terdiri dari beberapa bagian utama, yaitu distribusi, pembelian, penjualan, pemasaran pelayanan jasa produk atau jasa terhadap sistem elektronik seperti *internet* atau jaringan komputer lainnya. Industri Teknologi Informasi melihat EC sebagai aplikasi bisnis elektronik yang ditujukan kepada transaksi bisnis yang bersifat komersial. Yang mana tentunya melibatkan *electronic funds transfer, supply chain management, e-marketing, online marketing, online transaction processing, electronic data interchange (EDI), automated inventory management systems* dan *automated data collection systems*. Dan biasanya menggunakan teknologi komunikasi elektronik seperti *internet, ekstranet, e-mail, e-books, databases*, dan bahkan *mobile phones*.

Sebuah transaksi komersial dapat dibagi menjadi tiga tahap utama : *advertising and searching, ordering and payment, and delivery*. Dan semua tahapan ini dapat dilakukan di *internet* dengan menggunakan konsep dari *e-Commerce*.

Electronic commerce dapat dikategorikan berdasarkan pihak yang berpartisipasi di dalam transaksi atau bisnis prosesnya. Beberapa kategori *e-Commerce* yang sudah banyak dikenal adalah *Business-to-Consumer (B2C)*, *Business-to-Business(B2B)*, *Consumer-to-Consumer (C2C)* dan *Business-to-Government (B2G)*.

Dan dua kategori yang paling sering digunakan adalah :

- Bisnis yang menjual secara langsung kepada end-consumers dan menggunakan cara online untuk menerima order dan pembayaran atau yang dikenal sebagai B2C *e-Commerce*. Pembayaran di muka via sistem online yang aman menawarkan keamanan yang lebih kepada pembeli dimana sebagian processing company, menyetujui untuk mengembalikan uang (refund) customer apabila order tidak dapat dipenuhi dengan benar .
- B2B *e-Commerce* adalah istilah yang digunakan untuk menunjuk kepada bisnis yang menjual kepada bisnis lainnya. Transaksi bisnis terhadap bisnis secara online saat ini sedang mengalami perkembangan yang sangat pesat. Sistem seperti ini biasanya dilakukan antara sebuah perusahaan besar dengan supplier-nya. Tentu saja biasanya terdapat standar - standar tertentu yang ditetapkan untuk proses ini seperti format pemesanan secara elektronik menggunakan aplikasi yang akan diberikan oleh perusahaan tersebut, dll.