BAB II

LANDASAN TEORI

2.1 Internet

Menurut Oxford Dictionary, *internet is a global computer network providing a variety of information and communication facilities, consisting of interconnected networks using standardized communication protocols.* Press (2012)

Atau dapat diartikan bahwa internet adalah jaringan komputer global yang menyediakan fasilitas informasi dan komunikasi menggunakan protokol komunikasi standar.

2.2 *Website*

Pengertian *website* atau situs menurut Hidayat (2010) adalah kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman. Hubungan antara satu halaman *web* dengan halaman *web* yang lainnya disebtu *hyperlink,* sedangkan teks yang dijadikan media penghubung disebut *hypertext.*

Seiring dengan perkembangan teknologi informasi yang begitu cepat, *website* juga mengalami perkembangan yang sangat berarti. Dalam pengelompokan jenis *web*, lebih diarahkan berdasarkan kepada fungsi, sifat atau *style* dan bahasa pemrograman yang digunakan.

Menurut Hidayat (2010), jenis-jenis *web* berdasarkan sifat atau *style*-nya :

a. *Website* dinamis, merupakan sebuah *website* yang menyediakan content atau isi yang selalu berubah-ubah setiap saat. Bahasa pemrograman yang digunakan antara lain PHP, ASP, .NET dan memanfaatkan *database* MySQL atau MS SQL.
b. *Website* statis, merupakan *website* yang *content*-nya sangat jarang diubah. Bahasa pemrograman yang digunakan adalah HTML dan belum memanfaatkan *database*. Misalnya: *web profile* organisasi, dan lain-lain

Berdasarkan pada fungsinya, *website* terbagi atas:

a. *Personal website*, *website* yang berisi informasi pribadi seseorang
b. *Commercial website, website* yang dimiliki oleh sebuah perusahaan yang bersifat bisnis.
c. *Government website, website* yang dimiliki oleh instansi pemerintahan, pendidikan yang bertujuan memberikan pelayanan kepada pengguna.
d. *Non-Profit Organization website, website* yang dimiliki oleh organisasi yang bersifat *non-profit* atau tidak bersifat bisnis.

Berdasarkan dari segi bahasa pemrograman yang digunakan, *website* terbagi atas:

a. *Server side*, merupakan *website* yang menggunakan bahasa pemrograman yang tergantung kepada tersedianya server. Speerti PHP, ASP, .NET dan lain sebagainya. Jika tidak ada server, *website* yang dibangun menggunakan bahsa pemrograman di atas tidak akan berfungsi sebagaimana mestinya.
b. *Client side*, adalah *website* yang tidak membutuhkan server dalam menjalankannya, cukup diakses melalui browser saja. Misalnya, HTML.

2.3 *Unified Modeling Language (UML)*

Menurut Widodo (2011) UML merupakan bahasa pemrograman standar yang memiliki sintak dan semantik. Ketika kita membuat model menggunakan konsep UML ada aturan-aturan yang harus diikuti. Bagaimana elemen pada model-model yang kita buat berhubungan satu dengan lainnya harus mengikuti standar yang ada. UML bukan hanya sekedar diagram, tetapi juga menceritakan konteksnya.

Widodo menyebutkan ada 9 jenis diagram UML. Jenis diagram itu antara lain:

a. Diagram kelas. Bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Diagram ini umum dijumpai pada pemodelan sistem berorientasi objek. Meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas aktif.
b. Diagram *package*. Bersifat statis. Diagram ini memperlihatkan kumpulan kelas-kelas, merupakan bagian dari diagram komponen.
c. Diagram *Use-Case.* Bersifat statis. Diagram ini memperlihatkan himpunan *use-case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna.
d. Diagram interaksi dan *sequence.* Bersifat dinamis. Diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu.
e. Diagram *communication*. Bersifat dinamis. Diagram sebagai pengganti diagram kolaborasi UML 1.4 yang menekankan organisasi struktural dari objek-objek yang menerima serta mengirim pesan.
f. Diagram *Statechart*. Bersifat dinamis. Diagram status memperlihatkan keadaan-keadaan pada sistem, memuat status (*state*), transisi kejadian serta aktifitas. Diagram ini terutama penting untuk memperlihatkan

sifat dinamis dari antarmuka (interface), kelas, kolaborasi, dan terutama penting pada pemodelan sistem-sistem yang reaktif.

g. Diagram *Activity*. Bersifat dinamis. Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi suatu sistem dan memberi tekanan pada aliran kendali antar objek.

h. Diagram komponen. Bersifat statis. Diagram komponen ini memperlihatkan organisasi serta kebergantungan sistem/perangkat lunak pada komponen-komponen yang telah ada sebelumnya. Diagram ini berhubungan dengan diagram kelas dimana komponen secara tipikal dipetakan ke dalam satu atau lebih kelas-kelas antarmuka-antarmuka serta kolaborasi-kolaborasi.

i. Diagram *Deployment.* Bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (*run-time).* Memuat simpul-simpul beserta komponen-komponen yang ada di dalamnya. Diagram *deployment* berhubungan erat dengan diagram komponen diaman diagram ini memuat satu atau lebih komponen-komponen. Diagram ini sangat berguna saat aplikasi kita berlaku sebagai aplikasi yang dijalankan pada banyak mesin *(distributed computing).*

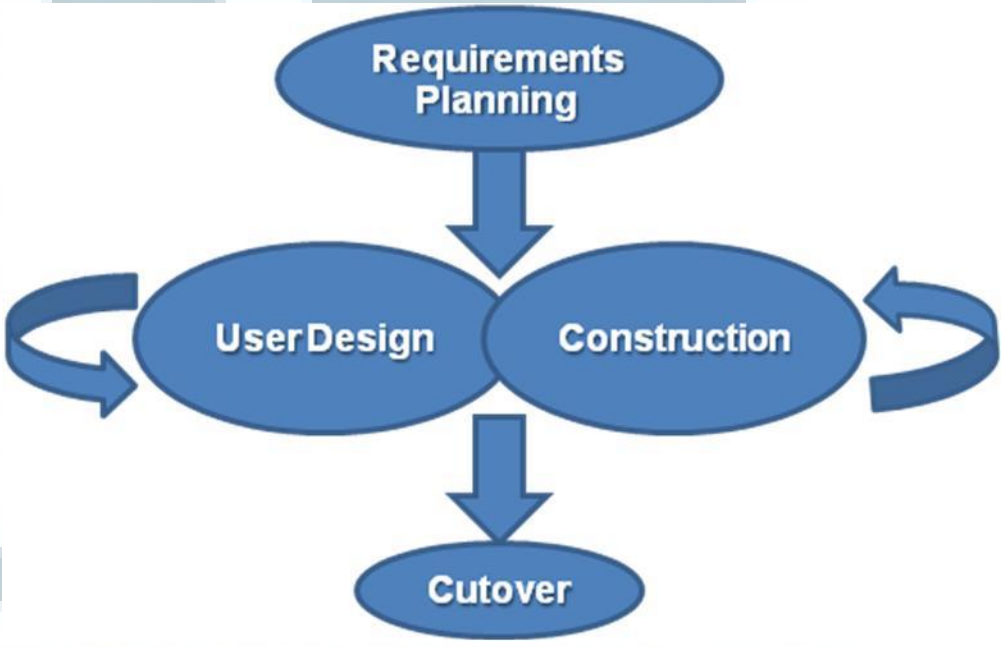
Kesembilan diagram ini tidak mutlak harus digunakan dalam pengembangan perangkat lunak, semuanya dibuat sesuai kebutuhan. UML memungkinkan kita menggunakan diagram-diagram lainnya (misalnya *Data Flow Diagram, Entity Relationship Diagram* dan sebagainya).

2.4 *Rapid Application Development (RAD)*

Menurut Pressman (2012) RAD adalah proses model perangkat lunak inkremental yang menekankan siklus pengembangan yang singkat. Model RAD adalah sebuah adaptasi “kecepatan tinggi” dari model waterfall, di mana perkembangan pesat dicapai dengan menggunakan pendekatan konstruksi berbasis

komponen. Jika tiap-tiap kebutuhan dan batasan ruang lingkup projek telah diketahui dengan baik, proses RAD memungkinkan tim pengembang untuk menciptakan sebuah "sistem yang berfungsi penuh" dalam jangka waktu yang sangat singkat. Dari penjelasan Pressman (2012) ini, satu perhatian khusus mengenai metodologi RAD dapat diketahui, yakni implementasi metode RAD akan berjalan maksimal jika pengembang aplikasi telah merumuskan kebutuhan dan ruang lingkup pengembangan aplikasi dengan baik.

Sedangkan menurut Kendall (2010), RAD adalah suatu pendekatan berorientasi objek terhadap pengembangan sistem yang mencakup suatu metode pengembangan serta perangkat-perangkat lunak. RAD bertujuan mempersingkat waktu yang biasanya diperlukan dalam siklus hidup pengembangan sistem tradisional antara perancangan dan penerapan suatu sistem informasi. Pada akhirnya, RAD sama-sama berusaha memenuhi syarat-syarat bisnis yang berubah secara cepat.



Gambar 2.1 Siklus RAD

(Sumber: Cashman, 2013)

Menurut Cashman (2013), terdapat 4 fase yang terjadi dalam siklus *Rapid Application Development*:

1. *Requirements Planning*

   Pada fase ini *pengguna dan penganalisis bertemu* untuk menentukan tujuan-tujuan aplikasi atau sistem serta untuk mengidentifikasikan syarta-syarat informasi yang ditimbulkan dari tujuan tersebut. Orientasi dalam fase ini adalah menyelesaikan masalah-masalah perusahaan. Meskipun teknologi informasi dan sistem bisa mengarah sebagian dari sistem yang diajukan, fokusnya akan selalu tetap pada upaya pencapaian tujuan-tujuan perushaan.
2. *User Design*

   Fase ini adalah fase untuk merancang dan memperbaiki yang bisa digambarkan sebagai *workshop.* Penganalisis dan pemrogram dapat bekerja membangun dan menunjukan representasi visual desain dan pola kerja kepada pengguna.
3. *Construction*

   Di fase *construction,* penganalisis dan pemrogram membuat sistem dari hasil desain yang telah disetujui oleh pengguna. Selama *construction,* pengguna merespon prototipe yang ada dan penganalisis memperbaiki modul-modul yang dirancang berdasarkan respon pengguna.
4. *Cutover*

   Pada fase ini sistem disetujui lalu kemudian diimplementasikan. Sistem kemudian dilakukan uji coba dan kemudian diperkenalkan kepada organisasi.

Menurut Pressman (2012), seperti semua proses model, pendekatan RAD mempunyai beberapa kelemahan, yaitu:

a. Untuk proyek yang berskala besar, RAD membutuhkan sumber daya manusia yang cukup untuk membuat tim RAD dengan jumlah yang tepat.

b. RAD membutuhkan *developer* dan *customer* yang *commit* terhadap aktifitas gerak cepat yang dibutuhkan untuk membuat sistem selesai pada jangka waktu terbatas. Jika komitmen yang dimiliki terbatas maka proyek RAD akan gagal.
c. Tidak semua jenis pengaplikasian cocok dengan RAD. Jika sistem tidak dapat dimodularisasi, membangun komponen yang penting bagi RAD dapat merepotkan. Jika performa tinggi menjadi masalah dan performa mau dicapai melalui antarmuka kepada komponen sistem, maka pendekatan RAD berkemungkinan tidak berjalan.
d. RAD tidak pantas dipakai ketika resiko teknis tinggi. Hal ini dapat terjadi ketika sebuah aplikasi baru menggunakan teknologi baru secara besar atau ketika *software* baru membutuhkan kerja sama yang tinggi dengan program komputer yang ada.

2.5 *Database*

Perkembangan jaman yang pesat kini membuat semakin meningkatnya minat pengguna dalam menggunakan sistem untuk aktivitas sehari-harinya, tentu itu sangat dibutuhkan sebuah *database* dalam menyimpan data-data dalam jumlah yang sangat besar. Pada umumnya *database* digunakan pada perusahaan untuk menyimpan data-data seperti penjualan, pembelian, produk pada sebuah barang, hingga data tentang konsumen mereka.

Menurut Connolly dan Begg (2010), mendeskripsikan *database* merupakan kumpulan dari data yang terkait secara logis dan merupakan deskripsi dari data yang dibangun agar dapat memenuhi kebutuhan perusahaan.

2.6 Normalisasi

Normalisasi Menurut Carlos Coronel (2013) adalah suatu proses untuk melakukan evaluasi dan memperbaiki struktur tabel di dalam *database* dengan tujuan untuk meminimalisir redudansi data dan mengurangi anomali data.

Normalisasi bekerja melalui beberapa rangkaian tahapan yang disebut dengan *Normal Forms*. Ada 3 tahapan normalisasi yaitu *First Normal Form (1NF), Second Normal Form (2NF),* dan *Third Normal Form (3NF).*

Tahapan 1NF:

a. Hilangkan kelompok yang berulang
b. Identifikasi *Primary Key*
c. Identifikasi semua variabel dependensi

Tahapan 2NF:

a. Hilangkan *partial dependencies*
b. Atur kembali variabel dependen yang berhubungan

Tahapan 3NF:

a. Hilangkan *transitive dependencies*

2.7 *Entity Relationship Diagram (ERD)*

Menurut Brady dan Loonam (2010), *entity relationship diagram* (ERD) merupakan teknik yang digunakan untuk memodelkan kebutuhan data dari suatu organisasi, biasanya oleh sistem analis dalam tahap analisis persyaratan proyek pengembangan sistem. Sementara seolah-olah teknik diagram atau alat peraga memberikan dasar untuk desain *database* relasional yang mendasari sistem informasi yang dikembangkan. ERD bersama-sama dengan detail pendukung merupakan model data yang pada gilirannya digunakan sebagai spesifikasi untuk *database*.

Dalam pembentukan ERD terdapat 3 komponen yang akan dibentuk yaitu:

a. Entitas

Pada post sebelumnya mengenai basis data telah dijelaskan sedikit tentang pengertian *entity* (entitas) yaitu suatu obyek yang dapat dibedakan dari lainnya yang dapat diwujudkan dalam basis data. Pengertian lainnya menurut Brady dan Loonam (2010), entitas adalah objek yang menarik di bidang organisasi yang dimodelkan.

b. Hubungan (relasi/*relationship*)

Suatu hubungan adalah hubungan antara dua jenis entitas dan direpresentasikan sebagai garis lurus yang menghubungkan dua entitas.

c. Atribut

Atribut memberikan informasi lebih rinci tentang jenis entitas. Atribut memiliki struktur internal berupa tipe data. Atribut yang dimaksud disebut dengan *key*, terdapat beberapa jenis *key* seperti berikut ini:

1. *Primary key*, merupakan suatu atribut yang tidak hanya dapat mengidentifikasi secara unik suatu kejadian yang spesifik, akan tetapi juga dapat mewakili setiap kejadian dari suatu *entity*.
2. *Secondary key*, disebut juga *alternate key* karena *key* yang tidak terpilih menjadi *primary key.*
3. *Composite key*, suatu kelompok dari atribut yang dapat mengidentifikasi suatu entitas.
4. *Foreign key*, merupakan suatu atribut yang melengkapi satu *relationship* yang menunjukan keinduknya

Menurut Brady dan Loonam (2010), derajat relasi atau kardinalitas rasio menjelaskan jumlah maksimum hubungan antara satu entitas dengan entitas lainnya.

1. *One to One* (1:1)

   Setiap anggota entitas A hanya boleh berhubungan dengan satu anggota entitas B, begitu pula sebaliknya.

2. *One to many* (1:M / *Many*)

   Setiap anggota entitas A dapat berhubungan dengan lebih dari satu anggota entitas B tetapi tidak sebaliknya.

3. *Many to Many* (M:M)

   Setiap entitas A dapat berhubungan dengan banyak entitas himpunan entitas B dan demikian pula sebaliknya

2.8 *PHP*

PHP pertama kali dibuat oleh Rasmus Lerdrof pada tahun 1995. Pada saat itu, PHP merupakan sebuah kumpulan skrip yang digunakan untuk mengolah data dari *web*. Pada saat ini PHP dikenal sebagai *Hypertext preprocessor* yang dipakai untuk membangun sebuah aplikasi *web* yang mendukung *database.*

Sedangkan menurut Kevin Yank (2012), PHP merupakan *server-side scripting language* yang dapat digunakan pengguna untuk memasukan macam-macam intruksi ke dalam halaman *web* yang nantinya akan dieksekusi oleh software pada umumnya seperti apache sebelum halam tersebut dikirim ke browser yang memintanya.

2.9 *MySQL*

Menurut Anhar (2010) MySQL adalah salah satu *databases management system (DBMS)* dari sekian banyak DBMS seperti *Oracle*, *MS SQL*, *Postagre SQL*, dan lainya. *MySQL* berfungsi untuk mengolah *database* menggunakan bahasa *SQL*. *MySQL* bersifat *open source* sehingga kita bisa menggunakannya secara gratis. Pemograman *PHP* juga sangat mendukung / *support* dengan *database* MySQL.