# BAB II

# LANDASAN TEORI

## 2.1 Object Oriented

Object oriented merupakan paradigma baru dalam rekayasa software yang didasarkan pada objek dan kelas. Object oriented mencakup bidang aplikasi yang sangat luas. Para pengguna sistem komputer dan sitem lain yang didasarkan tekhnologi komputer merasakan efek object oriented dalam bentuk meningkatkan aplikasi software yang mudah digunakan dan servis yang fleksibel.. Karena luasnya cakupan *Object Oriented* maka terdapat beberapa hal yang membingungkan berkenaan dengan istilah dan konsep *Object Oriented* . Istilah dan konsep yang berkenaan dengan *Object Oriented* adalah :

1. Object Oriented analysis

   Metode analisis yang memeriksa requirements (syarat/keperluan yang harus dipenuhi suatu sistem) dari sudut pandang kelas-kelas dan objek-objek yang ditemui dalam ruang lingkup permasalahan.

2. Object Oriented design

   Metode untuk mengarahkan arsitektur software yang didasarkan pada manipulasi objek-objek sistem atau subsistem.

Beberapa konsep dasar OOAD :

1. Objek

   Objek (object) adalah “benda”, secara fisik atau konseptual, yang dapat kita temui disekeliling kita. Hardware, software , dokumen,

manusia, dan bahkan konsep semuanya adalah objek. Sebuah objek memiliki keadaan sesaat (*state*) dan prilaku (*behavior*). *State* dari sebuah objek adalah kondisi objek tersebut atau himpunan dari keadaan yang mengambarkan objek tersebut. State dinyatakan dengan nilai dari *atribut* objeknya. *Atribut* adalah nilai internal suatu objek yang mencerminkan antara lain karekteristik objek. Perubahan *state* dicerminkan oleh prilaku objek tersebut.

*Behavior* suatu objek mendefenisikan bagaimana sebuah objek bertindak dan memberi reaksi. Behavior dari sebuah objek dicerminkan oleh *interface, service, dan method* dari objek tersebut. *Interface* adalah pintu untuk mengakses service objek. *Service* adalah fungsi yang bisa ditambah objek. *Method* adalah mekanisme internal objek yang mencerminkan prilaku objek tersebut.

2. Kelas

   Kelas (*Class*) adalah defenisi umum (pola, template atau cetak biru) untuk himpunan objek sejenis. Kelas menetapkan spesifikasi prilaku dan atribut objek-objek tersebut. *Class* adalah keniskalan (abstraksi) dari entitas dalam dunia nyata.

3. Kotak Hitam dan Interface

   Encapsulation adalah proses menyembunyikan detil implementasi sebuah objek. Satu-satunya jalan untuk mengakses data objek tersebut adalah melalui interface melindungi internal state sebuah objek dari “campur tangan” pihak luar. Oleh karena itu objek sering digambarkan sebagai kotak hitam (black box) yang menerima

dan mengirim pesan-pesan. Dalam OO programming kotak hitam tersebut berisi kode (himpunan instruksi dengan bahasa yang dipahami komputer).

4. Association dan Aggregation

Association (asosiasi) adalah hubungan antar objek yang saling membutuhkan, sedangkan Aggregation (aggregasi) adalah bentuk khusus dari asosiasi yang mengambarkan seluruh bagian suatu objek merupakan bagian dari objek lainnya. Sebagai contah objek tanggal dapat disusun dari objek hari, objek bulan, dan objek tahun.

### 2.1.1 Konsep Berorientasi Objek

Metode/metodologi berorientasi objek didasarkan pada konsep :

1. Pembungkusan (encapsulation), mengumpulkan data dengan pemanupulasiannya menjadi saru entitas.
    - Terkait pada konsep penyembuyian informasi (information hiding) :
        a. Pemanipulasian rinci informasi dilakukan secara internal.
        b. Dunia luar hanya mengetahui informasi secukupnya dengan cara yang sudah ditentukan sebelumnya
    - Objek : Suatu entitas yang memiliki identitas, status dan prilaku.
    - Kelas : Abstraksi dari sejumlah objek dengan karakteristik yang serupa.

2. Pewarisan (inherintence) : pembentukan suatu hirarki atau jejaring akibat pengumuman atau pengkhususan (bergantung pada fokus perhatian).
3. Kebanyakrupaan (polymorphisme) :
    - Pensubstitusian dinamis supertype oleh subtype
    - Pemanipulasian objek secara seragam dengan tipe yang berbeda-beda.

### 2.1.2 Metode Jacobson

Metode Jacobson OOSE (rekayasa perangkat lunak berorientasi objek), metode Jacobson versi sederhana dari metode Objectory sebelumnya, yang juga dikembangkan oleh Jacobson. Metode ini dibedakan dari metode yang lain karena sangat menekankan *use case* , deskripsi atau skenario yang mengambarkan bagaimana pemakia berinteraksi dengan produk atau sistem. Outline singkat mengenai proses OOA Jacobson adalah sebagai berikut :

- Identifikasi pemakai sistem dan seluruh tanggung jawab mereka.
- Bangun model persyaratan
    a. Tentukan aktor dan tanggung jawab mereka
    b. Identifikasi use case untuk setiap tingkah laku
    c. Persiapkan pandangan awal mengenai objek dan hubungan sistem
    d. Kaji model dengan menggunakan use case sebagai

skenario untuk menentukan vasilitas.

- Bangun model analisis
  a. Identifikasi objek interface dengan menggunakan informasi interaksi aktor.
  b. Ciptakan pandangan struktural mengenai objek interface.
  c. Representasikan tingkah laku objek
  d. Isolasi subsistem dan model untuk masing-masing
  e. Kaji model dengan menggunakan use case sebagai skenario untuk menentukan validitas.

## 2.2 Konsep Dasar Sistem

Istilah “Sistem” sekarang ini banyak dipakai, konsep–konsep yang berhubungan dengan sistem telah diterapkan dalam berbagai rancangan sistem terhadap pemecahan masalah dan manajemen. Sebuah sistem terdiri atas bagian-bagian yang bergabung untuk satu tujuan. Model dasarnya masukan, pengolahan dan keluaran.

Sistem dapat didefinisikan dua pendekatan, yaitu sistem yang menekankan pada prosedur dan sistem yang menekankan pada elemen-elemen atau komponennnya**.** Menurut JOG[5] pengertian Sistem adalah:

*“Suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan berkumpul bersama-sama untuk melakukan suatu kegiatan atau penyelesain suatu sasaran tertentu “*.

## 2.3 Konsep Dasar Informasi

Menurut DAV[2] pengertian Informasi adalah :

*"Data yang telah diolah menjadi suatu bentuk yang berarti bagi penerima dan bermanfaat dalam mengambil keputusan saat ini atau mendatang ".*

Kualitas dari suatu informasi tergantung dari tiga hal , yaitu :

1. Akurat

   Informasi harus bebas dari kesalahan-kesalahan yang biasanya terjadi dan selain itu harus jelas maksud dan tujuannya, sehingga *output* (keluaran) bisa dipertanggung jawabkan.

2. Tepat waktu

   Informasi datang pada sipemakai tidak boleh terlambat karena informasi yang terlambat tidak akan mempunyai nilai lagi dalam pengambilan keputusan.

3. Relevan

   Informasi tersebut mempunyai manfaat dan informasi yang diterima sipemakai dengan yang lainnya bisa berbeda-beda.

## 2.4 Konsep Dasar Sistem Informasi

Sebuah informasi dapat dihasilkan melalui proses dari sebuah sistem. Sistem informasi adalah sebuah sistem yang bertujuan untk menghasilkan informasi yang diperlukan.

Menurut JOG[5] pengertian sistem informasi adalah :

"*Suatu sistem didalam organisasi yang mempertemukan kebutuhan pengelolaan transaksi harian, mendukung operasi, bersifat manajerial,*

*dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan".*

Berdasarkan pengertian diatas, maka dapat disimpulkan bahwa pengertian sistem informasi adalah elemen-elemen yang saling terkait dan berkerj sama untuk menghasilkan informasi bagi suatu organisasi atau perusahaan.

## 2.5 Pengenalan UML

*Unified Modelling Language* (UML) merupkan pengganti dari metode analisis beroreintasi objek dan desain beroroentasi objek (OOA&D)yang dimunculkan sekitar akhir tahun 80-an dan awal 90-an.

*Unified Modelling Language* (UML) merupakan bagian gabungan dari metode Booch, Rumbaugh (OMT) dan Jacobsan tetapi (UML) ini akan mencakup lebih luas dari pada (OOA&D). Pada pertengahan pengembangan (UML) dilakukan standarisasi proses dengan (OMG) *(Object Management Group)* dengan harapan (UML) akan menjadi bahasa standar pemodelan dimasa yang akan datang.

### 2.5.1 Pengertian UML

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

Dengan menggunakan (UML) kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan

pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena (UML) juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET.

Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C. Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *synta*x/semantik.

### 2.5.2 Bagian-Bagian UML

Bagian-bagian utama dari UML adalah view, diagram, model,element, dan general mechanism. Dibawah ini adalah penjelasan mengenai bagian-bagian tersebut :

#### 2.5.2.1 *View*

*View* digunkan untuk melihat sistem yang dimodelkan dari beberapa aspek yang berbeda. *View* bukan melihat grafik, tapi merupakan suatu abstraksi berisi sejumlah diagram.

Beberapa jenis view dalam UML antara lain :

1. *Use Case View*

   Mendeskripsikan fungsionalis sistem yang seharusnya dilakukan sesuai yang dinginkan .

2. *Logical View*

   Mendeskrifsikan bagaimana funsinalis dari sistem. Struktur

statis *class, object* dan *relationship* dan kolaborasi dinamis yang terjadi ketika *object* mengirm pesan ke *object* lain dalam satu fungsi tertentu.

3. *ComponentView*

   Mendeskripsikan implemetasi dan ketergantungan modul. Komponen yang merupakan tipe lainnya dari code module diperlihatkan dengan struktur dan ketergantungannya juga alokasi sumberdaya komponen dan inforamsi dan administrative lainnya.

4. *Concurrency View*

   Membagi sistem ke dalam dan prosesor

5. *Deployment View*

   Mendeskripsikan fisik dari sistem seperti computer dan perangkat (*nodes)* dan bagaimana hubiungannya dengan lainnya.

#### 2.5.2.2 Diagram

Diagram berbentuk grafik yang menunjukan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah diagram merupakan bagian dari suatu view tertentu.

Adapun jenis diagram antara lain :

1. *Use Case Diagram*

   Menggambarkan sejumlah *external actors* dan hubungannya ke

*use case* yang diberikan oleh sistem. Digram ini menunjukan fungsionalitas suatu sistem atau kelas dan bagaimana sistem beritegrasi dengan dunia luar.

2. *Class Diagram*

   Menggambarkan struktur statis *class* di dalam sistem. Class mempresentasikan sesuatu yang ditangani oleh sistem. Sebuah sistem biasanya mempunyai beberapa class diagram.

3. *State Diagram*

   Menggambarkan semua state (kondisi) yang dimiliki oleh suatu *object* dari suatu *class* dan keadaan yang menyebabkan *state* berubah. Kejadian dapat berupa object lain yang mengirim pesan.

4. *Sequence Diagram*

   Menggambarkan kolaborasi dinamis antara sejumlah *object*. Kegunaannya untuk menunjukan rangkaian pesan yang dikirim antara object juga interaksi antara object, sesuatu yang terjadi pada titik tertentu dalam eksekusi system.

5. *Collaboratin Diagram*

   Menggambarkan kolaborasi dinamis seperti *sequence diagram.* Dalam menunjukan pertukaran pesan *Collaboratin Diagram* mengambarkan object dan hubungannya (mengacu ke konteks).

6. *Activity Diagram*

   Menggambarkan rangkain aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu

operasi sehingga dapat digunakan untuk aktifitas lainnya seperti use case atau interaksi.

7. *Component Diagram*

   Menggambarkan struktur fisik kode dari komponen, komponen dapat berupa *source code*, komponen *biner*, atau *executable component.*

8. *Deployment Diagram*

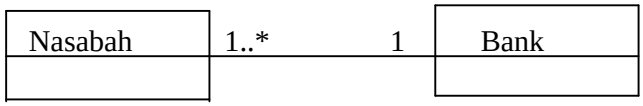
   Menggambarkan arsitektur fisik dari perangkat keras dan perangkat lunak sistem. Digram ini menunjukan hubungan komponen dengan perangkat (nodes) satu sama lain dan jenis hubungannya.

#### 2.5.2.3 Indikator Multiplicity

Indikator *Multiplicity* ditentukan utuk *class***,** *Multiplicity* menentukan banyak *object* yang terlibat dalam relasi. *Multiplicity* menentukan banyak *object* yang terhubung satu dengan yang lainnya. Indikator *Multiplicity* terdapat pada masing-masing akhir garis relasi, baik pada asosiasi maupun agregasi. Beberapa contoh *Multiplicity* berserta gambarnya adalah :

| | |
|---|---|
| 1 | Tepat waktu |
| 1..* | Nol atau lebih |
| 1..* | Satu atau lebih |
| 0..1 | Nol atau Satu |
| 5..8 | Range 5 s.d 8 |

4..6,9 Range 4 s.d 6 dan 9




**Gambar 2.1**
**Indikator multiplicity**

## 2.6 Basis Data

Basis data adalah suatu kumpulan data yang diorganisasikan dan disajikan untuk memenuhi kebutuhan pembentukan informasi. Informasi basis data digunakan untuk menunjang proses pengambilan keputusan.

Sistem pengolahan data merupakan perangkat lunak sistem basis data. Ada 2 hal yang penting dalam menggunakan komputer yang mempunyai daya tampung penyimpanan yang besar serta mempunyai kemampuan pengolahan data yang cepat, tepat dan akurat.

Basis data mempunyai kriteria yang penting yaitu :

1. Dapat digunakan oleh beberapa program aplikasi tanpa perlu mengubah basis data.
2. Dapat berkembang dengan mudah.
3. Dapat memenuhi kebutuhan sistem baru secara mudah.
4. Dapat digunakan dengan cara yang berbeda-beda.

Dari keempat kriteria tersebut diatas telah membedakan secara nyata atau jelas antara file basis data dan file tradisional yang bersifat *prooriented.*

## 2.7 Internet

Menurut Rober J. Verzello yang diterjemahkan oleh Jogiyanto HM (2001) "internet adalah suatu jaringan komputer global yang terbentuk dari jaringan-jaringan komputer lokal dan regional, dengan adanya jaringa ini memungkinkan komunikasi data antar komputer-komputer yang terhubung kejaringan tersebut".

### 2.7.1 Sejarah Internet

Pada awalnya internet merupakan suatu jaringan komputer yang dibentuk oleh Departemen Pertahanan Amerika diawal tahun enam puluhan, melalui proyek ARPA *(Advenced Research Project Egency)* yang disebut ARPANET, dimana meraka mendomonstrasikan bagaimana *hardware* dan *software* komputer yang berbasis UNIX, bisa melakukan komunikasi dalam jarak yang tak terhingga melalui saluran telepon.

Proyek ARPANET merancang bentuk jaringan, kehandalan, seberapa besar informasi dapat dipindahkan. Dan akhirnya standar yang meraka tentukan menjadi cikal bakal untuk pengembangan protokol baru yang dikenal TCP/IP

*( Transmission Control Protocol/Internet Protocol ).*

ARPANET dibentuk oleh empat Universitas besar di Amerika yaitu : University of California at Los Angles, Stanford Research Institute, University of California at Santa Barbara, dan University of Utah. Mereka membentuk suatu jaringan terpadu ditahun 1969, ARPANET diperkenalkan secara umum pada bulan Oktober 1972.

Pada awal tahun delapan puluhan, seluruh jaringan yang tergabung dalam ARPANET diubah menjadi TCP/IP, karena proyeknya sendiri sudah dihentikan, dan jaringan ARPANET inilah yang merupakan koneksi utama *(backbone)* dari internet. Proyek percobaan tersebut tersebut dilanjutkan dan dibiayai oleh NFS *(National Science Foundation).* NSF mengubah nama jaringan ARPANET menjadi NFSNET dimana backbonenya berkecepatan tinggi dihubungkan ke komputer-komputer yang ada di Universitas dan lembaga penelitian terbesar di Amerika, pemerintah Amerika memberi ijin kearah komersial pada tahun 1990.

### 2.7.2 Istilah-istilah Internet

1. *Browser*, merupakan *software* pelengkap yang digunakan untuk melihat *web page.*
2. *Browsing,* adalah metode untuk menjelajah alamat internet.
3. *Downoad,* adalah nenyalin file dari komputer satu kekomputer lainya.
4. *Hyperlink,* adalah bagian dari web page yang saling berhubungan.
5. *Home Page,* adalah informasi yang tersedia dalam bentuk, teks, gambar, maupun suara yang diletakan dikomputer.
6. *Web,* adalah sitem pada internet yang ditampilkan dalam bentuk teks, gambar, maupun suara.
7. *Website,* adalah lokasi pada web.

## 2.8 Perangkat Lunak Pendukung

### 2.8.1 HTML

HTML *(hyper text markup language)* adalah merupakan suatu format yang digunakan dalam pembuatan dokumen dan aplikasi yang berjalan dihalaman web.

HTTP *(hypertext transfer protocol)* merupakan salah satu format yang digunakan untuk mentransfer data antar web server ke web browser, protokol ini menstransfer dokumen-dokumen web yang diatas atau berformat HTML. Dikatakan *markup language* karena HTML berfungsi untuk memformat dokumen teks biasa yang disimpan dengan eksekusi.htm atau.html.

### 2.8.2 PHP

PHP adalah bahasa *scripting server side*, dimana seluruh proses dikerjakan di *server* kemudian hasil yang dikirim ke klien yang melakukan permintaan, tempat pemakai menggunakan *browser* (lebih dikenal dengan istilah *server side scripting*).

Beberapa hal yang membedakan PHP dengan bahasa pemrograman lain adalah adanya tag penentu yaitu diawali dengan “<?” atau “<?php” dan diakhiri dengan “?>”. Jadi, kita bebas menempatkan skrip PHP dimanapun dalam dokumen HTML yang telah kita buat.

#### 2.8.2.1 Sejarah Singkat PHP

Perjalanan PHP sangat panjang dalam beberapa tahun terakhir ini. Berkembang menjadi sebuah bahasa pemrograman berbasis web yang sangat menjadi prioritas utama para pengembang web seperti yang kita lihat sekarang. Mulanya Rasmus Lerdorf penemu PHP, membangun bahasa pemrograman berbasiskan web untuk digunakan pada websitenya sendiri agar dapat mengetahui siapa saja yang melihat biodatanya, dan bahasa pemrograman ini diberi nama PHP Personal Home Page Tools

Dia menggabungkan kefleksibelan bahasa C dan memudahkannya, sehingga banyak yang berminat untuk ikut menggunakan PHP tersebut. Rasmus pun lebih mendalami implementasi bahasa C dan menciptakan PHP/FI dan PHP/FI 2.0 *"Form Interface*", yang dapat menerima inputan data dari *Form* kedalam Variabel dalam PHP. Kali ini bukan hanya personal saja yang menggunakannya, tetapi badan komersialpun mulai melirik bahasa pemrograman ini.

Pada tahun 1997, PHP ditulis ulang oleh Andi Gutmans dan Zeev Suraski, dan terilis lah PHP 3.0 Versi pertama yang bahasa-nya lebih komitmen dan paling mendekati dengan PHP yang kita kenal sekarang. Dikarenakan tidak sesuai lagi dengan penggunaannya, maka PHP pun berganti *ancronym* menjadi PHP *Hypertext Preprocessor*. Pada saat itu telah terdapat 10% webserver di seluruh dunia yang menggunakan PHP.

Tidak lama setelah PHP 3.0, Andi Gutmans and Zeev Suraski memulai kembali pekerjaan untuk menulis ulang *Core* dari PHP. Hasilnya menjadikan performa PHP menjadi jauh lebih baik, dan *Engine* tersebut diberi nama *'Zend Engin"* (dari nama mereka, Zeev and Andi). Diselesaikan pada tahun 1999 dan menghadirkan PHP baru yang kita kenal sekarang, yaitu PHP 4. Dengan banyak *features* baru dan fasilitas yang lebih baik. Lebih dari 20% *Account Domain* di Internet menggunakan PHP 4 ini, ribuan pengembang *(developer)* dan juga jutaan website yang memberikan report bahwa telah terinstall PHP pada website mereka.
PHP Group sedang mengembangkan lagi versi terbaru dari PHP, yaitu PHP 5, dengan engine baru Zend 2.0, yang akan menambah lagi features baru dan keunggulan baru lainnya.

**2.8.2.2 Keistimewaan PHP**

Sebagai bahsa skrip yang bersipat server side PHP memiliki beberapa keunggulan antara lain :

1. Tidak diperlukan kompabilitas browser atau harus menggunakan browser tertentu karena serverlah yang akan mengerjakan skrip PHP
   tersebut.
2. *Freeware* dapat didistribusikan dengan bebas
3. Memiliki kemampuan koneksi berbagai macam database seperti MySQL, PostgreSQL, dBase, SyBase dan banyak lagi.

### 2.8.3 MySQL

MySQL merupakan salah satu *database server*, dimana *database server* adalah suatu perangkat lunak yang mampu mengelola data dengan baik, sehingga data yang tersimpan dapat digunakan kembali. Ketangguhan serta stabilitas MySQL adalah mampu berjalan pada berbagai *platform* sistem operasi. MySQL termasuk jenis RDBMS (*Relational Database Management Sistem*). Pada MySQL, sebuah database mengandung satu atau sejumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau bebarapa kolom.

Secara garis besar, MySQL mengenal tiga kelompok tipe data, yaitu tipe data numerik atau angka, tipe data tanggal dan waktu, serta tipe data *string*. Berikut adalah tipe-tipe data, yaitu:

a. *CHAR* (M)

   Tipe data CHAR digunakan untuk menyimpan data *string* dengan jumlah karakter tertentu. Jumlah karakter pada tipe data CHAR dapat berkisar antara 1-255 karakter.

b. *VARCHAR* (M)

   Tipe data ini lebih fleksibel daripada tipe data *CHAR* (M). Tipe data *VARCHAR* dapat digunakan untuk menyimpan data *string* dengan panjang yang bervariasi, tergantung data yang digunakan hanya panjang karakter 1-255.

c. *INT* (N)

   Tipe data *INT* digunakan untuk menyimpan data *integer* dengan

nilai berkisar antara -2147483648 sampai 2147483647.

d. *DATE*

Tipe data *DATE* digunakan untuk menyimpan informasi seputar penanggalan. Format tipe data *DATE* yaituYYYY-MM-DD, dimana YYYY adalah format untuk tahun, MM adalah format untuk bulan yang dimulai dari 01 (Januari) sampai 12 (Desember), DD adalah format untuk tanggal.

e. *DATETIME*

Tipe data ini digunakan untuk menyimpan informasi penanggalan yang lebih detail daripada tipe data *DATE* karena format tipe data *DATETIME* ditambah dengan format waktu.

f. *TEXT dan BLOB*

Tipe data *TEXT dan BLOB* digunakan untuk menyimpan data *string* dengan jumlah karakter antara 255 sampai 65535.

#### 2.8.3.1 Sejarah Singkat MySQl

MySQL dikembangkan sekitar tahun 1994 oleh sebuah perusahaan pengembang *software* dan konsultan database MySQL AB yang bertempat di Swedia. Waktu perusahaan tersebut bernama TcX DataKonsult AB, dan tujuan awal dikembangkannya MySQL adalah untuk mengembangkan aplikasi berbasisis web pada *client.*

Awalnya Micheal Widenius “ Monty “, pengembang satu-satunya TcX, memeliki UNIREG dan rutin ISAM buatannya sendiri dan sedang mencari antarmuka SQl yang cocok untuk

diimplementasikan kedalamnya. Mula-mula Monty memakai MiniSQl (mSQL) pada percobaanya itu, tetapi mSQL dirasa kurang sesauai, karena terlalu lambat dalam pemrosessan query data.

Akhirnya Monty menghubungi David Hughes, pembaut mSQL yang sedang merilis versi kedua daari mSQL. Kemudian Monty mencoba membuat sendiri mesin SQL yang memiliki antarmuka mirip dengan SQL, tetapi dengan kemampuan yang lebih sesuai, dan lahirlah MySQL.

#### 2.8.3.2 Keistimewaan MySQl

Sebagai database server  ayng memiliki database modern, MySQL memiliki beberapa keistimewaan, diantaranya :

1. *Portability*

   MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, MaC OS X Server, dab lain-lainnya.

2. *Open Source*

   MySQL didistribusikan secara *open source*, sehingga dapat digunakan secara bebeas.

3. *Multi-User*

   MySQL dapat digunakan oleh bebrapa user dalam waktu yang bersamaan tanpa mengalami masalah. Hal ini memungkinkan sebuah database  server MySQL dapat

diakses *client* secara bersamaan.

4. *Performance Tuning*

   MySQL memiliki kecepatan yang tinggi dalam menangani query, dengan kata lain dapat memproses lebih banyak SQL persatuan waktu.

5. *Column Type*

   MySQL memiliki tipe kolom yang sangat kompleks seperti *signed/unsigned, float, double, char, varchar, text, blob, date, time, time stamp, year, set dan error.*

6. *Command dan Function*

   MySQL memiliki operator fungsi secara penuh yang mendukung perintah SELECT dan WHERE data query.

7. *Security*

   MySQL memiliki lapisan-lapisan sekuritas sever subnet maju, nama host, dan izin akses user dengan sistem yang mendetail serta *password.*

8. *Scalability dan Limits*

   MySQL mampu menangani database dengan skala besar, dengan jumlah records lebih dari 50 juta dan 60 juta tabel serta 5 miliar baris, selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada setiap tabelnya.

9. *Connectivity*

   MySQL dapat melakukan koneksi dengan client mrnggunakan protokol TCP/IP, Unix Socet (UNIX), atau

Named Ripes (NT).

10. *Localisation*

    Deteksi pesan kesalahanpada client dengan menggunakan lebih dari 20 bahasa.

11. *Interface*

    Interface terhadap berbagai aplikasi dan bahasa pemograman dengan menggunakan fungsi API (*Aplication Programming Interface*)

12. *Client dan Tool*

    Dilengkapi dengan berbagai tool yang dapat digunakan untuk administrasi, dan pada setiap tool yang ada disertakan petunjuk online.

13. Struktur Tabel

    Struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibanding database lainnya semacam PostgreeSQL ataupun Oracle.

### 2.8.4 Apache

Untuk menjalankan PHP dan MySQL, kita membutuhkan sebuah *Web Server*. *Apache* merupakan salah satu *web server* yang ketangguhannya telah teruji serta sifat dari *Apache* yang *free* dan *open source*. *Web server* adalah suatu program yang dijalankan pada komputer yang terkoneksi dengan internet. *Web server* melihat hubungan dengan internet dan menunggu perintah atau permintaan dari *web browser* akan

HTML atau dokumen.

Ketika ia menerima suatu permintaan dan menemukan suatu dokumen HTML, ia akan mengirimnya kembali ke *browser* yang mengadakan permintaan tersebut. Saat ini *Apache* merupakan *web server* yang paling populer.

## 2.9 Kasus Yang Dianalisa/dirancang

Perkembangan Teknologi web telah digunakan sebagai Sistem Informasi Akademik STMIK SAMARINDA. Sebagai salah satu Perguruan Tinggi dalam pengunaan tehnologi web belum dapat memberikan informasi secara optimal dalam hal penilaian, mengingat web yang ada sebatas pada tampilan logo dan selamat datang saja, sehingga dari web yang ada tidak didapatkan informasi secara lengkap

Dari informasi yang ada, maka dirasakan perlu untuk membangun sebuah web yang dapat memberikan informasi penilaian secara lengkap dan dinamis. Oleh sebab itu penulis mengambil tema Tugas Akhir ini dengan judul "**Perancangan Sistem Informasi Akademik Pada STMIK SAMARINDA**".

This document was created with Win2PDF available at http://www.daneprairie.com.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.