

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Pada tinjauan pustaka ini, penulis mendapat referensi dari sebuah jurnal yang berjudul “ PENERAPAN METODE *FORWARD CHAINING* PADA PENJADWALAN MATA KULIAH” oleh Wisnu Yudho Utomo Program Studi Manajemen Informatika, Fakultas Teknologi Informasi Universitas Kanjuruhan Malang.

Penjadwalan diperlukan untuk mengatur waktu kerja, sehingga didapatkan waktu kerja yang efektif dan teratur sesuai dengan rencana. Penjadwalan perkuliahan adalah penjadwalan yang mengatur masalah pelaksanaan perkuliahan yang meliputi dosen pengajar, mata kuliah yang diajarkan, waktu (hari dan jam) perkuliahan, dan ruangan yang dipakai. Penjadwalan perkuliahan melibatkan banyak komponen yang telah disebutkan, sehingga proses penjadwalan menjadi sangat rumit. Penjadwalan mata kuliah ini menerapkan metode *Forward Chaining* dengan pencarian dan pelacakan ke depan.

Inti dari penjadwalan mata kuliah adalah bagaimana menjadwalkan sejumlah komponen yang terdiri atas mahasiswa, dosen, ruang, dan waktu dengan sejumlah aturan dan batasan tertentu.

2.2 Pengertian Sistem

Menurut Jogiyanto (2005:2) dalam buku yang berjudul Analisis dan Desain Sistem Informasi, sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan-tujuan tertentu

2.3 Pengertian Informasi

Menurut Jogiyanto HM., (1999: 692), informasi adalah sekumpulan fakta-fakta yang telah diolah menjadi bentuk data, sehingga dapat menjadi lebih berguna dan dapat digunakan oleh siapa saja yang membutuhkan data-data tersebut sebagai pengetahuan ataupun dapat digunakan dalam pengambilan keputusan.

Sumber informasi adalah data. Data itu berupa fakta kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Yang kemudian data tersebut diolah melalui suatu metode untuk menghasilkan informasi, kemudian penerima menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yang kemudian menghasilkan suatu tindakan yang lain yang akan menimbulkan sejumlah data kembali.

2.4 Sistem Informasi

2.4.1. Definisi Sistem Informasi

Menyangkut pemahaman tentang pengertian sistem informasi ini, dalam bukunya Abdul Kadir (2003) mengutip beberapa pendapat para ahli, diantaranya :

1. Menurut Hall sistem Informasi adalah sebuah rangkaian prosedur formal dimana data dikelompokkan, diproses menjadi informasi dan didistribusikan kepada pemakai.
2. Menurut Bodnar dan Hopwood sistem Informasi adalah kumpulan perangkat keras dan perangkat lunak yang dirancang untuk mentransformasikan data ke dalam bentuk informasi yang berguna.

2.5 Pengembangan Sistem

2.5.1 Definisi Pengembangan Sistem

Terdapat beberapa pendapat yang menjelaskan mengenai definisi dari pengembangan sistem, diantaranya :

1. Pengembangan sistem merupakan suatu proyek yang harus melalui suatu proses pengevaluasian seperti pelaksanaan proyek lainnya. (Amsa, 2008)
2. Pengembangan sistem adalah metode/prosedur/konsep aturan yang digunakan untuk mengembangkan suatu sistem informasi atau pedoman bagaimana dan apa yang harus dikerjakan selama pengembangan sistem (*algorithm*). Metode adalah suatu cara, teknik sistematis untuk mengerjakan sesuatu (dinu, 2008).

2.5.2 Hal Mendasar Dalam Pengembangan Sistem

Dalam pengembangan dan perancangannya, penganalisa sistem merupakan bagian dari tim yang berfungsi mengembangkan sistem yang memiliki daya guna tinggi dan memenuhi kebutuhan pemakai akhir. Pengembangan dipengaruhi sejumlah hal (Okta, 2007), yaitu:

a. Produktifitas

Saat ini dibutuhkan sistem yang lebih banyak, lebih baik dan lebih cepat. Hal ini membutuhkan lebih banyak *programmer* dan penganalisa sistem yang berkualitas, kondisi kerja ekstra, kemampuan pemakai untuk mengembangkan sendiri, bahasa pemrograman yang lebih baik, perawatan sistem yang lebih baik (umumnya 50% s.d 70% sumber daya digunakan untuk perawatan sistem), disiplin teknis pemakaian perangkat lunak, dan perangkat pengembangan sistem yang terotomasi.

b. Reliabilitas

Waktu yang dihabiskan untuk testing sistem secara umum menghabiskan 50% dari waktu total pengembangan sistem. Dalam kurun waktu 30 tahun sejumlah sistem yang digunakan diberbagai perusahaan mengalami kesalahan dan ironisnya tidak mungkin untuk diubah. Sebagai contoh kasus; untuk setiap program yang dihasilkan dari IBM's *superprogramer*

project punya tiga sampai lima kesalahan untuk setiap kesalahan untuk setiap sepuluh *statement* pemrograman.

2.6 Mind Mapping

Mind Mapping atau Peta Pikiran adalah metode mempelajari konsep yang ditemukan pertama kali oleh Tony Buzan. Konsep ini didasarkan pada cara kerja otak kita dalam menyimpan informasi. Hasil penelitian menunjukkan bahwa otak kita tidak menyimpan informasi dalam kotak-kotak sel saraf yang terjejer rapi melainkan dikumpulkan pada sel-sel saraf yang bercabang-cabang yang apabila dilihat sekilas akan tampak seperti cabang-cabang pohon. *Mind Map* merupakan istilah teknik pemetaan pikiran untuk membantu membuka seluruh potensi dan kapasitas otak yang masih "tersembunyi". Pemetaan pikiran ini akan melibatkan kedua sisi otak secara bersamaan, yaitu otak kanan dan otak kiri.

Mind mapping dapat diartikan sebagai proses memetakan pikiran untuk menghubungkan konsep-konsep permasalahan tertentu dari cabang-cabang sel saraf membentuk korelasi konsep menuju pada suatu pemahaman dan hasilnya dituangkan langsung di atas kertas dengan animasi yang disukai dan gampang dimengerti oleh pembuatnya. Sehingga tulisan yang dihasilkan merupakan gambaran langsung dari cara kerja koneksi-koneksi di dalam otak.

Mind mapping adalah cara mengembangkan kegiatan berpikir ke segala arah, menangkap berbagai pikiran dalam berbagai sudut. Mind

mapping mengembangkan cara berpikir divergen dan berpikir kreatif. Mind mapping yang sering kita sebut dengan peta konsep adalah alat berpikir organisasional yang sangat hebat yang juga merupakan cara termudah untuk menempatkan informasi ke dalam otak dan mengambil informasi itu ketika dibutuhkan (Tony Buzan , 2008:4).

Ditinjau dari segi waktu Mind mapping juga dapat mengefisienkan penggunaan waktu dalam mempelajari suatu informasi. Hal ini utamanya disebabkan karena metode ini dapat menyajikan gambaran menyeluruh atas suatu hal, dalam waktu yang lebih singkat. Dengan kata lain, Mind mapping mampu memangkas waktu belajar dengan mengubah pola pencatatan linear yang memakan waktu menjadi pencatatan yang efektif yang sekaligus langsung dapat dipahami oleh individu.



Gambar 2.1 Mind Mapping

2.7 Forward Chaining

2.7.1 Pengertian Forward Chaining

Menurut (Rusel S. 2003) mendefinisikan *forward chaining* ini metode pencarian atau teknik pelacakan ke depan yang dimulai

dengan informasi yang ada dan penggabungan rule untuk menghasilkan suatu kesimpulan dan tujuan. Metode ini akan menganalisa masalah tertentu dan selanjutnya akan mencari jawaban atau kesimpulan yang terbaik. Penalaran dimulai dengan mencocokkan kaidah-kaidah dalam basis pengetahuan dengan fakta-fakta yang ada dalam basis data.

Forward chaining merupakan metode inferensi yang melakukan penalaran dari suatu masalah kepada solusinya. Jika klausa premis sesuai dengan situasi (bernilai TRUE), maka proses akan menyatakan konklusi. Forward chaining adalah data-driven karena inferensi dimulai dengan informasi yang tersedia dan baru konklusi diperoleh. Jika suatu aplikasi menghasilkan tree yang lebar dan tidak dalam, maka gunakan forward chaining.

Tipe sistem yang dapat dicari dengan Forward Chaining :

1. Sistem yang dipersentasikan dengan satu atau beberapa kondisi.
2. Untuk setiap kondisi, sistem mencari rule-rule dalam knowledge base untuk rule-rule yang berkorespondensi dengan kondisi dalam bagian IF
3. Setiap rule dapat menghasilkan kondisi baru dari konklusi yang diminta pada bagian THEN. Kondisi baru ini ditambahkan ke kondisi lain yang sudah ada.

4. Setiap kondisi yang ditambahkan ke sistem akan diproses. Jika ditemui suatu kondisi baru dari konklusi yang diminta, sistem akan kembali ke langkah 2 dan mencari rule-rule dalam knowledge base kembali. Jika tidak ada konklusi baru, sesi ini berakhir.

Terdapat 10 aturan yang tersimpan dalam basis pengetahuan yaitu :

R1 : if A and B then C

R2 : if C then D

R3 : if A and E then F

R4 : if A then G

R5 : if F and G then D

R6 : if G and E then H

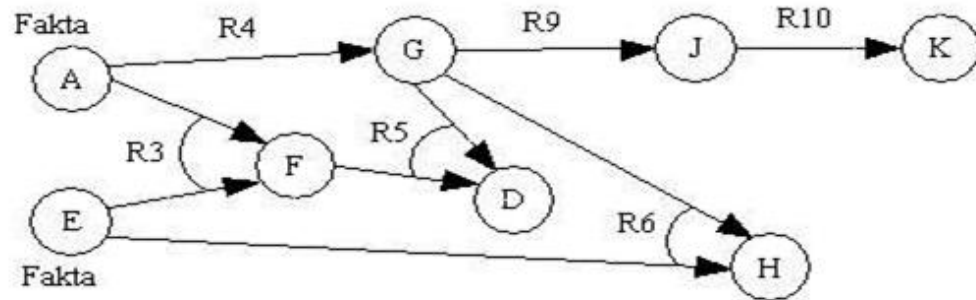
R7 : if C and H then I

R8 : if I and A then J

R9 : if G then J

R10 : if J then K

Fakta awal yang diberikan hanya A dan E, ingin membuktikan apakah K bernilai benar. Proses penalaran forward chaining terlihat pada gambar dibawah :



2.8 Definisi Unified Modeling Language (UML)

UML (*Unified Modeling Language*) adalah metode pemodelan secara visual sebagai sarana untuk merancang dan atau membuat *software* berorientasi objek. Karena UML ini merupakan bahasa visual untuk pemodelan bahasa berorientasi objek, maka semua elemen dan diagram berbasiskan pada paradigma *object oriented*.

UML sendiri juga memberikan standar penulisan sebuah sistem *blue print*, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema *database*, dan komponen - komponen yang diperlukan dalam sistem *software*.

UML tidak hanya merupakan sebuah bahasa pemrograman visual saja, namun juga dapat secara langsung dihubungkan ke berbagai bahasa pemrograman, seperti JAVA, C++, Visual Basic, atau bahkan dihubungkan secara langsung ke dalam sebuah *object-oriented database*.

Untuk dapat memahami UML membutuhkan bentuk konsep dari sebuah bahasa model, dan mempelajari 3 (tiga) elemen utama dari UML, seperti *building block*, aturan-aturan yang menyatakan bagaimana

building block diletakkan secara bersamaan, dan beberapa mekanisme umum (*common*).

2.8.1 Building Block

Tiga macam yang terdapat dalam building block yaitu:

- **Benda / Things**

Adalah hal yang sangat mendasar dalam model UML, juga merupakan bagian paling statik dari sebuah model, serta menjelaskan elemenelemen lainnya dari sebuah konsep dan atau fisik. Bentuk dari beberapa benda / thing adalah *Classes*, *Interfaces*, *Collaboration*, *Use Case*, dan *Nodes*.

- **Hubungan / Relationship**

Ada 4 macam hubungan didalam penggunaan UML, yaitu :

- *Dependency*, adalah hubungan semantik antara dua benda/*things* yang mana sebuah benda berubah mengakibatkan benda satunya akan berubah pula. Umumnya sebuah digambarkan sebuah panah dengan garis terputus-putus.
- *Association*, hubungan antar benda struktural yang terhubung diantara obyek. Kesatuan obyek yang terhubung merupakan hubungan khusus, yang menggambarkan sebuah hubungan struktural diantara seluruh atau sebagian. Umumnya *assosiation* digambarkan dengan sebuah garis yang dilengkapi dengan sebuah label, nama, dan status hubungannya.

- *Generalizations*, adalah menggambarkan hubungan khusus dalam obyek anak/*child* yang menggantikan obyek *parent* / induk. Dalam hal ini, obyek anak memberikan pengaruhnya dalam hal struktur dan tingkah lakunya kepada obyek induk. Digambarkan dengan garis panah.
- *Realizations*, merupakan hubungan semantik antara pengelompokkan yang menjamin adanya ikatan diantaranya. Hubungan ini dapat diwujudkan diantara *interface* dan kelas atau *elements*, serta antara *use cases* dan *collaborations*. Model dari sebuah hubungan *realization*.

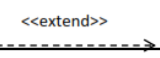
2.9 Diagram UML

Berikut ini adalah macam – macam diagram pada UML :

2.9.1 Use Case

Diagram Use Case menggambarkan apa saja aktifitas yang dilakukan oleh suatu sistem dari sudut pandang pengamatan luar. Yang menjadi persoalan itu apa yang dilakukan *bukan* bagaimana melakukannya.

Tabel 2.1 Simbol Pada Use Case Diagram

Penjelasan	Simbol
<i>Actor</i> : Mewakili peran orang, sistem lain atau alat ketika berkomunikasi dengan <i>use case</i> .	
<i>Use case</i> : Abstraksi dari interaksi antara sistem dan actor	
<i>Association</i> : adalah abstraksi dari penghubung antara actor dan <i>use case</i>	
<i>Include</i> : menunjukkan kebutuhan fungsionalitas dari suatu use case dengan yang lain.	
<i>Extend</i> : menunjukkan perilaku pilihan yang dapat dilibatkan.	
<i>Generalisation</i> : menunjukkan hubungan dari use case khusus ke use case yang lebih umum.	

2.9.2 Class Diagram

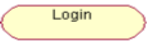
Diagram Class memberikan pandangan secara luas dari suatu sistem dengan menunjukkan kelas-kelasnya dan hubungan mereka. Diagram Class bersifat statis, *menggambarkan hubungan apa yang terjadi bukan apa yang terjadi jika mereka berhubungan.*

2.9.3 Activity

Pada dasarnya diagram Activity sering digunakan oleh *flowchart*. Diagram ini berhubungan dengan diagram Statechart. Diagram Statechart berfokus pada *obyek yang dalam suatu proses* (atau proses menjadi suatu obyek), diagram Activity berfokus pada *aktifitas-aktifitas yang terjadi, yang terkait dalam suatu proses tunggal*. Jadi dengan kata lain, diagram ini

menunjukkan bagaimana aktifitas-aktifitas tersebut bergantung satu sama lain.

Tabel 2.2 Simbol Pada Activity Diagram

Penjelasan	Simbol
Titik Awal atau permulaan.	
Titik Akhir atau akhir dari aktivitas.	
<i>Activity</i> , atau aktivitas yang dilakukan oleh aktor.	
<i>Decision</i> , atau pilihan untuk mengambil keputusan.	
Arah tanda panah alur proses.	

2.9.4 Sequence Diagram

Diagram sequence merupakan salah satu diagram Interaction yang menjelaskan bagaimana suatu operasi itu dilakukan, *message* (pesan) apa yang dikirim dan kapan pelaksanaannya. Diagram ini diatur berdasarkan waktu.

Obyek-obyek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut.

2.10 Perangkat Analisa Sistem

2.10.1 Flowchart

Flowchart adalah bagan – bagan yang mempunyai arus menggambarkan langkah – langkah suatu masalah flowchart. Flowchart merupakan cara penyajian dari suatu algoritma.

2.10.1.1 Fungsi Flowchart

Fungsi Flowchart :

- Bagan-bagan yang mempunyai arus.
- Menggambarkan langkah-langkah penyelesaian suatu masalah.
- Merupakan salah satu cara penyajian algoritma.

2.10.1.2 Tujuan Flowchart

Tujuan Flowchart :

- Menggambarkan suatu tahapan.
- Penyelesaian masalah.
- Secara sederhana, terurai, rapi, dan jelas.
- Mengetahui bagaimana menggambarkan prosedur sistem dengan berbagai jenis bagan alir.

Berdasarkan model atau jenisnya flowchart dibagi menjadidua yaitu Sistem Flowchart dan Program Flowchart.

➤ Sistem Flowchart

Bagan yang memperlihatkan urutan proses dalam sistem dengan menunjukkan alat media input, output, serta jenis media penyimpanan dalam proses pengolahan data.

➤ Program Flowchart

Bagan yang memperlihatkan intruksi yang digambarkan dengan symbol tertentu untuk memecahkan masalah dalam suatu program.

Metode penggambaran program flowchart :

- Conceptual flowchart, menggambarkan alur pemecahan masalah secara global.
- Detail flowchart, menggambarkan alur pemecahan masalah secara rinci.

2.10.1.3 Simbol-simbol Flowchart

Simbol-simbol yang dipakai dalam flowchart dibagi menjadi 3 kelompok :

1. Flow direction symbols

- Digunakan untuk menghubungkan simbol satu dengan yang lain.
- Disebut juga connecting line

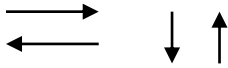
2. Processing symbols

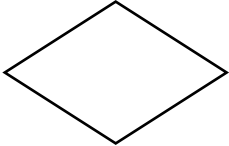
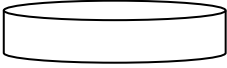
- Menunjukkan jenis operasi pengolahan dalam suatu proses / prosedur.

3. Input / Output symbols

- Menunjukkan jenis peralatan yang digunakan sebagai media input atau output.

Tabel 2.3 Simbol Flowchart

	Symbol arus / flow, yaitu menyatakan jalannya arus suatu proses
	Simbol Proses Untuk menyatakan suatu tindakan (proses) yang dilakukan oleh computer

	Simbol Dokumen Untuk menyatakan input yg berasal dari dokumen dalam bentuk kertas
	Simbol Decision/Logika Untuk menunjukan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban ya/tidak
	Simbol Database Untuk input data
	Simbol Terminal Untuk menyatakan permulaan atau akhir suatu program

2.11 Pengenalan Microsoft Visual Studio 2010

Visual Basic .NET (atau VB.NET) merupakan salah satu bahasa pemrograman yang berpusat pada object (Object Oriented Programming) digunakan dalam pembuatan aplikasi Windows yang berbasis *Graphical User Interface*(GUI), dan bisa digunakan untuk membangun aplikasi-aplikasi .NET di platform Microsoft .NET. Tidak seperti generasi sebelumnya ,Visual Basic versi 6.0 ke bawah yang lebih difokuskan untuk pengembangan aplikasi desktop. Visual Basic .NET memungkinkan para pengembang membangun bermacam aplikasi, baik desktop maupun aplikasi web.

2.11.1 Sejarah

Bill Gates, pendiri Microsoft, memulai bisnis perangkat lunak dengan mengembangkan interpreter bahasa Basic untuk Altair 8800, untuk kemudian ia ubah agar dapat berjalan di atas IBM PC dengan sistem operasi DOS. Perkembangan berikutnya

ialah diluncurkannya BASICA (basic-advanced) untuk DOS. Setelah BASICA, Microsoft meluncurkan Microsoft QuickBasic dan Microsoft Basic (dikenal juga sebagai Basic Compiler).

Visual Basic adalah pengembangan dari bahasa komputer BASIC (Beginner's All-purpose Symbolic Instruction Code). Bahasa BASIC diciptakan oleh Professor John Kemeny dan Thomas Eugene Kurtz dari Perguruan Tinggi Dartmouth pada pertengahan tahun 1960-an. Bahasa program tersebut tersusun mirip dengan bahasa Inggris yang biasa digunakan oleh para programmer untuk menulis program-program komputer sederhana yang berfungsi sebagai pembelajaran bagi konsep dasar pemrograman komputer.

Sejak saat itu, banyak versi BASIC yang dikembangkan untuk digunakan pada berbagai platform komputer, seperti Microsoft QBASIC, QUICKBASIC, GWBASIC, IBM BASICA, Apple BASIC dan lain-lain. Apple BASIC dikembangkan oleh Steve Wozniak, mantan karyawan Hewlett Packard dan teman dekat Steve Jobs (pendiri Apple Inc.). Steve Jobs pernah bekerja dengan Wozniak sebelumnya (mereka membuat game arcade "Breakout" untuk Atari). Mereka mengumpulkan uang dan bersama-sama merakit PC, dan pada tanggal 1 April 1976 mereka secara resmi mendirikan perusahaan komputer Apple. Popularitas dan pemakaian BASIC yang luas dengan berbagai jenis komputer turut berperan dalam mengembangkan dan memperbaiki bahasa

itu sendiri, dan akhirnya berujung pada lahirnya Visual Basic yang berbasis GUI (Graphic User Interface) bersamaan dengan Microsoft Windows. Pemrograman Visual Basic begitu mudah bagi pemula dan programmer musiman karena ia menghemat waktu pemrograman dengan tersedianya komponen-komponen siap pakai.

Hingga akhirnya Visual Basic juga telah berkembang menjadi beberapa versi, sampai yang terbaru, yaitu Visual Basic 2010. Bagaimanapun juga Visual Basic 6.0 tetap menjadi versi yang paling populer karena mudah dalam membuat programnya dan ia tidak menghabiskan banyak memori. Sejarah BASIC di tangan Microsoft sebagai bahasa yang diinterpretasi (BASICA) dan juga bahasa yang dikompilasi (BASCOM) membuat Visual Basic diimplementasikan sebagai gabungan keduanya. Programmer yang menggunakan Visual Basic bisa memilih kode bahasa pemrograman yang dikompilasi atau kode yang harus bahasa pemrograman yang diinterpretasikan sebagai hasil porting dari kode VB. Sayangnya, meskipun sudah terkompilasi jadi bahasa mesin, DLL bernama MSVBVMxx.DLL tetap dibutuhkan.

2.12 Pengertian Basis Data

Definisi basis data adalah kumpulan data logikal yang saling berhubungan dan deskripsi data tersebut dirancang dari suatu organisasi. Berbeda dengan sistem file yang menyimpan data secara terpisah, pada basis data sebuah data tersimpan secara integrasi. Basis

data bukan menjadi milik departemen, melainkan menjadi sumber daya yang dapat digunakan bersama. [Connolly, 2000]

Jika ditinjau dari segi bahasa, basis data terdiri dari dua kata, yaitu basis dan data, yang dapat didefinisikan dalam sejumlah sudut pandang seperti:

- Himpunan Kelompok Data (Arsip) yang saling berhubungan dan diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
- Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (Redundansi) yang tidak perlu.
- Kumpulan File/Table/Arsip yang saling berhubungan yang disimpan dalam media penyimpan Elektronik.

2.12.1 Tujuan Basis Data

Penggunaan sebuah basis data memiliki berbagai macam tujuan, adapun tujuan-tujuan tersebut meliputi [Kristianto, 2000] :

- Kecepatan dan kemudahan (Speed)
- Efisiensi ruang penyimpanan (Space)
- Keakuratan (Accuracy)
- Ketersediaan (Availability)
- Kelengkapan (Completeness)
- Keamanan (Security)
- Kebersamaan (Sharability)

2.12.2 Kegunaan Basis Data

Penyusunan satu basis data digunakan untuk mengatasi masalah-masalah pada penyusunan data yang dialami, yaitu [Kristianto, 2000] :

1. Redudansi dan inkonsistensi data
2. Kesulitan pengaksesan data
3. Isolasi data dan standarisasi
4. Multiuser (Banyak pemakai)
5. Masalah keamanan (Security)

2.13 MYSQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *database management system*) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL. MySQL adalah Relational Database Management System (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat *closed source* atau komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (Structured Query Language).SQL

adalah sebuah konsep pengoperasian database, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Keandalan suatu sistem database (DBMS) dapat diketahui dari cara kerja *optimizer*-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh user maupun program-program aplikasinya. Sebagai database server, MySQL dapat dikatakan lebih unggul dibandingkan database server lainnya dalam query data. Hal ini terbukti untuk query yang dilakukan oleh single user, kecepatan query MySQL bisa sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan Interbase.