BAB II

LANDASAN TEORI

Pada bab ini berisi penjelasan tentang kategori yang menjadi landasan teori dalam penyusunan Penelitian dan secara garis besar akan di jelaskan pengertian-pengertian dan teori-teori relevan yang akan digunakan dalam perancangan sistem yang akan dibuat dalam laporan ini.

2.1. Sistem

Sistem berasal dari bahasa latin (*systema*) yang artinya adalah kesatuan yang terdiri dari komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi. Istilah ini sering dipergunakan untuk menggambarkan suatu set entitas yang berinteraksi ,dimana suatu model matematika sringkali bisa dibuat.

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu (Jogyanto:"Analisis dan Desain",2005 hal 1).

2.1.1. Elemen Sistem

Ada beberapa elemen yang membentuk sebuah sistem, yaitu : tujuan, masukan, proses, keluaran, batas, mekanisme pengendalian dan umpan balik serta lingkungan. Berikut penjelasan mengenai elemen-elemen yang membentuk sebuah sistem :

1. Tujuan

Setiap sistem memiliki tujuan (*Goal*), entah hanya satu atau mungkin banyak. Tujuan inilah yang menjadi pemotivasi yang mengarahkan sistem. Tanpa tujuan, sistem menjadi tak terarah dan tak terkendali. Tentu saja, tujuan antara satu sistem dengan sistem yang lain berbeda.

2. Masukan

Masukan (*input*) sistem adalah segala sesuatu yang masuk ke dalam sistem dan selanjutnya menjadi bahan yang diproses. Masukan dapat berupa hal-hal yang berwujud (tampak secara fisik) maupun yang tidak tampak. Contoh masukan yang berwujud adalah bahan mentah, sedangkan contoh yang tidak berwujud adalah informasi (misalnya permintaan jasa pelanggan).

3. Proses

Proses merupakan bagian yang melakukan perubahan atau transformasi dari masukan menjadi keluaran yang berguna dan lbih bernilai, misalnya berupa informasi dan produk, tetapi juga bisa berupa hal-hal yang tidak berguna, misalnya saja sisa pembuangan atau limbah. Pada pabrik kimia, proses dapat berupa bahan mentah. Pada rumah sakit, proses dapat berupa aktivitas pembedahan pasien.

4. Keluaran

Keluaran (*output*) merupakan hasil dari pemrosesan. Pada sistem informasi, keluaran bisa berupa suatu informasi, saran, cetakan laporan, dan sebagainya.

5. Batas

Yang disebut batas (*boundary*) sistem adalah pemisah antara sistem dan daerah di luar sistem (lingkungan). Batas sistem menentukan konfigurasi, ruang lingkup, atau kemampuan sistem. Sebagai contoh, tim sepakbola mempunyai aturan permainan dan keterbatasan kemampuan pemain. Pertumbuhan sebuah toko kelontong dipengaruhi oleh pembelian pelanggan, gerakan pesaing dan keterbatasan dana dari bank. Tentu saja batas sebuah sistem dapat dikurangi atau dimodifikasi sehingga akan mengubah perilaku sistem. Sebagai contoh, dengan menjual saham ke publik, sebuah perusahaan dapat mengurangi keterbasatan dana.

6. Mekanisme Pengendalian dan Umpan Balik

Mekanisme pengendalian (*control mechanism*) diwujudkan dengan menggunakan umpan balik (*feedback*), yang mencuplik keluaran. Umpan balik ini digunakan untuk mengendalikan baik masukan maupun proses. Tujuannya adalah untuk mengatur agar sistem berjalan sesuai dengan tujuan.

7. Lingkungan

Lingkungan adalah segala sesuatu yang berada diluar sistem. Lingkungan bisa berpengaruh terhadap operasi sistem dalam arti bisa merugikan atau menguntungkan sistem itu sendiri. Lingkungan yang merugikan tentu saja harus ditahan dan dikendalikan supaya tidak mengganggu kelangsungan operasi sistem, sedangkan yang menguntungkan tetap harus terus dijaga, karena akan memacu terhadap kelangsungan hidup sistem.

2.1.2 Klasifikasi Sistem

Ada beberapa tipe sistem berdasarkan kategori:

1. Atas dasar keterbukaan:
   - Sistem terbuka, dimana pihak luar dapat mempengaruhinya.
   - Sistem tertutup.
2. Atas dasar komponen:
   - Sistem fisik, dengan komponen materi energy.
   - Sisstem non fisilk atau konsep, berisikan ide-ide.

2.2. Informasi

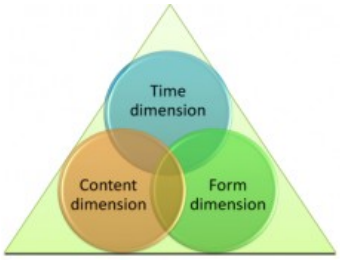
Informasi mempunyai manfaat dan peranan yang sangat dominan dalam suatu organisasi atau perusahaan. Tanpa adanya informasi dalam suatu organisasi, para manajer atau pimpinan tidak dapat bekerja efisiendan efektif. Tanpa tersedia informasi pula para atasan tidak dapat mengambil keputusan dengan cepat dan mencapai tujuan dengan efektif dan efisien.

Adapun pengertian informasi menurut beberapa pakar atau ahli, diantaranya sebagai berikut:

Menurut George R Terry, Ph. D, informasi adalah data yang penting yang memberikan pengetahuan yang berguna.

Menurut Gordon B. Davis, Informasi adalah data yang telah diolah menjadi suatu bentuk yang penting bagi penerima dan mempunyai nilai yang nyata yang dapat dirasakan dalam keputusan-keputusan yang sekarang dan yang akan datang (http://www.sisilain.net/2011/08/pengertian-informasi-menurut-para-ahli.html).

Adapun pada saat menganalisa dan merancang Sistem Informasi sering kali ditanya tentang kualitas informasi, Bahkan termasuk dalam tujuan utama yang ingin di capai. Sistem yang canggih, *interface* yang menarik dan kompleks akan menjadi percuma jika infrmasi yang di hasilkan tidak berkualitas. Maka kualitas sebuah informasi dibagi menjadi tiga : *Time Dimension*, *Content Dimension*, *Form Dimension*.



Gamabar 2.1. Kualitas Informasi

Sumber : http://biginaict.wordpress.com/tag/kualitas-informasi/

1. *Time Dimension* (Dimensi Waktu Informasi)

- *Currency* alias *Up To date.* Informasi yang disampaikan tepat waktu. Buat sistem informasi yang menyajikan informasi basi. Tidak bisa digunakan apalagi untuk mengambil keputusan. Informasi yang tersaji cepat akan memuaskan pengguna dan mendukung pengambilan keputusan. Akan lebih baik lagi jika real time

- *Timeliness* alias tersedia kapan saja user membutuhkan. Artinya informasi tersedia kapan pun user menginginkannya. Pagi, siang, sore, bahkan tengah malam. Mungkin yang saat ini lagi dikembangkan adalah aplikasi sistem informasi via handphone (*mobile application*).

- *Frequency* yang berarti informasi tersedia dalam periode waktu tertentu. Agak mirip-mirip dengan kategori up to date.

2. *Content Dimension* (Dimensi Konten Informasi)

- *Accuracy*. Jelas bahwa informasi yang tersedia akurat, bebas dari kesalahan sehingga tidak menjerumuskan user dan berakibat salah dalam mengambil keputusan.

- *Conciseness*. Dimaksudkan bahwa informasi yang disajikan diperlukan oleh user. Misal informasi prakiran cuaca, user membutuhkan suhu sekarang berapa, akan hujan atau tidak, berapa kecepatan angin, layak tidak untuk berlayar. Jangan diberi informasi kepadatan lalu lintas.

- *Relevance*. Informasi yang tersedia haruslah sesuai dengan kebutuhan *user*.

3. *Form Dimension* (Dimensi Bentuk Informasi)

Kalau bentuk informasi adalah cara bagaimana informasi tersebut sampai ke user. Media apa yang sebaiknya digunakan. Apakah sistem informasi stand alone atau yang online. Bisa diakses melalui apa, televisi, radio, komputer, layar lebar (seperti di jalan-jalan), atau melalui ponsel. Pilihan-pilihan ini dikembalikan lagi pada kebutuhan sistem berdasarkan hasil analisis permasalahan saat ini.

2.3. Sistem Informasi

Sistem Informasi (SI) adalah kombinasi dari teknologi informasi dan aktivitas orang yang menggunakan teknologi itu untuk mendukung operasi dan manajemen. Dalam arti yang sangat luas, istilah sistem informasi yang sering digunakan merujuk kepada interaksi antara orang, proses algoritmik, data, dan teknologi. Dalam pengertian ini, istilah ini digunakan untuk merujuk tidak hanya pada penggunaan organisasi teknologi informasi dan komunikasi (TIK), tetapi juga untuk cara di mana orang berinteraksi dengan teknologi ini dalam mendukung proses bisnis.

Ada yang membuat perbedaan yang jelas antara sistem informasi, dan komputer sistem TIK, dan proses bisnis. Sistem informasi yang berbeda dari teknologi informasi dalam sistem informasi biasanya terlihat seperti memiliki komponen TIK. Hal ini terutama berkaitan dengan tujuan pemanfaatan teknologi informasi. Sistem informasi juga berbeda dari proses bisnis. Sistem informasi membantu untuk mengontrol kinerja proses bisnis.

Alter berpendapat untuk sistem informasi sebagai tipe khusus dari sistem kerja. Sistem kerja adalah suatu sistem di mana manusia dan mesin melakukan pekerjaan dengan menggunakan sumber daya untuk memproduksi produk tertentu atau jasa bagi pelanggan. Sistem informasi adalah suatu sistem kerja yang kegiatannya ditujukan untuk pengolahan (menangkap, transmisi, menyimpan, mengambil, memanipulasi dan menampilkan) informasi.

Dengan demikian, sistem informasi berhubungan dengan sistem data di satu sisi dan sistem aktivitas di sisi lain. Sistem informasi adalah suatu bentuk komunikasi sistem di mana data yang mewakili dan diproses sebagai bentuk dari memori sosial. Sistem informasi juga dapat dianggap sebagai bahasa semi formal yang mendukung manusia dalam pengambilan keputusan dan tindakan. Sistem informasi merupakan fokus utama dari studi untuk disiplin sistem informasi dan organisasi informatika.

Sistem informasi adalah gabungan yang terorganisasi dari manusia, perangkat lunak, perangkat keras, jaringan komunikasi dan sumber data dalam mengumpulkan, mengubah, dan menyebarkan informasi dalam organisasi.

2.4 Java

2.4.1 Pengenalan Java Language

Bahasa Java dikembangkan oleh *Sun Microsystem* tahun 1991 sebagai bagian dari suatu proyek penelitian untuk mengembangkan software bagi konsumer barang-barang elektronik seperti televisi, *VCR, toaster* dan mesin –

mesin lainnya yang dapat dibeli di swalayan. Tujuan penciptaan Java pada waktu itu adalah menjadi suatu program yang berukuran kecil, efisien, dan *portable* di segala jenis *hardware*. Tujuan yang sama ini membuat Java menjadi satu bahasa yang ideal untuk mendistribusikan program-program yang dapat dijalankan melalui *www* dan juga suatu bahasa pemrograman untuk segala tujuan untuk mengembangkan program- program yang dapat digunakan dengan mudah dan *portable* di berbagai *platform* yang berbeda.

Sekarang, *Sun* telah mengeluarkan berbagai program Java yang dapat digunakan seperti *Java API,* atau JDK atau *JAVA Developer Kit* . Selain itu, banyak juga program-program lain yang dapat digunakan untuk membuat program Java, seperti *NetBeans, JBuilder, JCreator, J++*, dan sebagainya.

Saat ini distribusi Java dan kelas pendukungnya dibagi dalam tiga bagian yang masing-masing memiliki konsentrasi tersendiri yaitu:

1. *Java 2 Standart Edition* (J2SE), untuk aplikasi *desktop*
2. *Java 2 Enterprise Edition* (J2EE), untuk aplikasi *server*
3. *Java 2 Micro Edition* (J2ME), untuk piranti dengan kemampuan terbatas.

J2SE diutamakan untuk membuat java *Applet* dan aplikasi yang *stand alone*. J2EE digunakan untuk membuat sistem *enterprise*, dimana sebuah aplikasi dapat disebarkan pada jaringan computer dan dapat diakses dari jauh. J2EE mengandung semua fungsi J2SE dan diutamakan untuk membuat aplikasi server. Pada aplikasi ini penulis akan memakai platform J2SE.

2.4.2 Sejarah Java

Bahasa pemrograman Java pertama lahir dari *The Green Project*, yang berjalan selama 18 bulan, dari awal tahun 1991 hingga musim panas 1992. Proyek tersebut belum menggunakan versi yang dinamakan Oak. Proyek ini dimotori oleh *Patrick Naughton*, beserta sembilan programmer lainnya dari *Sun Microsystem.*Sekitar musim panas 1992 proyek ini ditutup dengan menghasilkan sebuah program Java Oak pertama, yang ditujukan sebagai pengendali sebuah peralatan dengan teknologi layar sentuh (*touch screen*), seperti pada PDA sekarang ini. Teknologi baru ini dinamai "*7" (Star Seven).Setelah era *Star Seven* selesai, sebuah anak perusahaan TV kabel tertarik ditambah beberapa orang dari proyek *The Green Project*.

Mereka memusatkan kegiatannya pada sebuah ruangan kantor di 100 Hamilton Avenue, Palo Alto Perusahaan baru ini bertambah maju: jumlah karyawan meningkat dalam waktu singkat dari 13 menjadi 70 orang. Pada rentang waktu ini juga ditetapkan pemakaian Internet sebagai medium yang menjembatani kerja dan ide di antara mereka. Pada awal tahun 1990-an, Internet masih merupakan rintisan, yang dipakai hanya di kalangan akademisi dan militer.

Pada sekitar bulan Maret 1995, untuk pertama kali kode sumber Java versi 1.0a2 dibuka. Sayang terjadi perpecahan di antara mereka suatu hari. Tiga dari pimpinan utama proyek,dari *Sun Microsystems* bersama Marc Andreessen, membentuk *Netscape.*

Nama Oak, diambil dari pohon oak yang tumbuh di depan jendela ruangan kerja "bapak java", James Gosling. Nama Oak ini tidak dipakai untuk versi release

Java karena sebuah perangkat lunak sudah terdaftar dengan merek dagang tersebut, sehingga diambil nama penggantinya menjadi "Java".

2.5 Kelebihan Dan Kekurangan Java

2.5.1 Kelebihan Java

- Multiplatform

Kelebihan utama dari *Java* ialah dapat dijalankan di beberapa *platform* / sistem operasi komputer, sesuai dengan prinsip tulis sekali, jalankan di mana saja. Dengan kelebihan ini pemrogram cukup menulis sebuah program *Java* dan dikompilasi (diubah, dari bahasa yang dimengerti manusia menjadi bahasa mesin / *bytecode*) sekali lalu hasilnya dapat dijalankan di atas beberapa *platform* tanpa perubahan. Kelebihan ini memungkinkan sebuah program berbasis *Java* dikerjakan diatas *operating system Linux* tetapi dijalankan dengan baik di atas *Microsoft Windows*. *Platform* yang didukung sampai saat ini adalah *Microsoft Windows*, *Linux*, *Mac OS*, dan *Sun Solari*. Penyebabnya adalah setiap sistem operasi menggunakan programnya sendiri-sendiri (yang dapat diunduh dari situs *Java*) untuk menginterpretasikan *bytecode* tersebut.

- OOP (Object Oriented Programming)

Semua aspek yang terdapat di *Java* adalah Objek. *Java* merupakan salah satu bahasa pemrograman berbasis objek secara murni. Semua tipe data diturunkan dari kelas dasar yang disebut Object. Hal ini sangat memudahkan pemrogram untuk mendesain, membuat, mengembangkan dan mengalokasi kesalahan sebuah program dengan basis *Java* secara cepat, tepat, mudah dan

terorganisir. Kelebihan ini menjadikan *Java* sebagai salah satu bahasa pemograman termudah, bahkan untuk fungsi fungsi yang *advance* seperti komunikasi antara komputer sekalipun.

- Perpustakaan yang lengkap

  Java terkenal dengan kelengkapan *library* (kumpulan program program yang disertakan dalam pemrograman java) yang sangat memudahkan dalam penggunaan oleh para pemrogram untuk membangun aplikasinya. Kelengkapan perpustakaan ini ditambah dengan keberadaan komunitas *Java* yang besar yang terus menerus membuat perpustakaan-perpustakaan baru untuk melingkupi seluruh kebutuhan pembangunan aplikasi

- Bergaya C++

  Memiliki sintaks seperti bahasa pemrograman [C++] sehingga menarik banyak pemrogram C++ untuk pindah ke *Java*. Saat ini pengguna Java sangat banyak, sebagian besar adalah pemrogram C++ yang pindah ke *Java*. Universitas-universitas di Amerika juga mulai berpindah dengan mengajarkan Java kepada murid-murid yang baru karena lebih mudah dipahami oleh murid dan dapat berguna juga bagi mereka yang bukan mengambil jurusan komputer.

2.5.2 Kelemahan Java

- Mudah Didekompilasi

Dekompilasi adalah proses membalikan dari kode jadi menjadi kode sumber. Ini dimungkinkan karena kode di *Java* merupakan *bytecode* yang menyimpan banyak atribut bahasa tingkat tinggi, seperti nama-nama kelas, metode, dan tipe data. Hal yang sama juga terjadi pada *Microsoft .NET Platform*.

Dengan demikian, algoritma yang digunakan program akan lebih sulit disembunyikan dan mudah dibajak atau *direverse-engineer*.

- Penggunaan Memori yang banyak

Penggunaan memori untuk program berbasis *Java* jauh lebih besar daripada bahasa tingkat tinggi generasi sebelumnya seperti C/C++ dan Pascal (lebih spesifik lagi, Delphi dan Object Pascal). Biasanya ini bukan merupakan masalah bagi pihak yang menggunakan teknologi terbaru (karena trend memori terpasang makin murah), tetapi menjadi masalah bagi mereka yang masih harus berkutat dengan mesin komputer berumur lebih dari 4 tahun.

2.6 Interface Java

Dalam pembuatan project ini, saya hanya menggunakan hal-hal yang sederhana, misalkan penggunaan *button, label, textfield*, dan lain-lain. Dalam aplikasi ini, saya menggunakan awt yang merupakan komponen yang saling terkait, mulai dari jendela yang paling luar sampai komponen-komponen kecil dalam interface pengguna (*user*). Komponen bisa seperti apa saja yang tampak dilayar monitor nantinya, seperti *button, label, textfield,* dan lain-lain.

Salah satu yang menarik dari *Java* adalah tersedianya sedemekian banyak *class* yang siap pakai dan tinggal diturunkan untuk membuat *applet* maupun aplikasi baru. Kumpulan *class-class* ini disimpan dalam satu paket yang diakses dengan memanggil nama paket yang bersangkutan. Paket-paket ini telah meliputi paket bahasa *Java* dasar (java.lang), paket abstraksi window dan objek-objeknya (java.awt), sampai dengan paket networking TCP/IP (java.net).

2.7 Penjualan

Penjualan adalah sebuah usaha atau langkah konnkrit yang dilakukan untuk memindahkan suatu produk, baik berupa barang atau pun jasa, dari produsen kepada konsumensebagai sasarannya. Tujuan utama penjualan yaitu mendatanngkan keuntungan atau laba daru produk ataupun barang yang dihasilkan produsennya dengan penglolaan yang baik. Dalam pelaksanaannya, penjualan sendiri tidak akan dapat dilakukan tanpa adanya pelaku yang bekerja di dalamnya seperti agen, pedagang dan tenaga pemasaran.

2.8 Tools Perancangan

NetBeans mengacu pada dua hal, yakni *platform* untuk pengembangan aplikasi desktop java, dan sebuah *Integrated Development Environment* (IDE) yang dibangun menggunakan *platform* NetBeans.

Platform NetBeans memungkinkan aplikasi dibangun dari sekumpulan komponen perangkat lunak moduler yang disebut ‘modul’. Sebuah modul adalah suatu arsip java yang memuat kelas-kelas java untuk berinteraksi dengan NetBeans Open API dan file manifestasi yang mengidentifikasinya sebagai modul. Aplikasi yang dibangun dengan modul-modul dapat dikembangkan dengan menambahkan modul-modul baru. Karena modul dapat dikembangkan secara independen, aplikasi berbasis platforms NetBeans dapat dengan mudah dikembangkan oleh pihak ketfa secara mudah dan powerful.

2.8.1 Sejarah Netbeans

Pengembangan NetBeans diawali sari Xelfi, sebuah proyek mahasiswa tahun 1997 dibawah bimbingan Fakultas Matematika dan Fisika Universitas Charles, Praha. Sebuah perusahaan kemudian dibentuk untuk proyek tersebut

dan menghasilkan versi komersial NetBeans IDE hingga kemudian dibeli *Sun Microsystem* pada tahun 1999. Sun kemudian menjadikan NetBeans open source pada bulan juni tahun 2000. sejak itu komunitas NetBeans terus berkembang

2.8.2 Platform NetBeans

Platform NetBeans adalah dapat digunakan kembali kerangka untuk menyederhanakan pengembangan aplikasi *Java Swing* desktop.NetBeans IDE Java SE dukungan berisi apa yang dibutuhkan untuk mulai mengembangkan NetBeans NetBeans plugin dan aplikasi berbasis Platform, SDK ada tambahan diperlukan.

Ketika sebuah aplikasi yang didasarkan pada Platform NetBeans dijalankan, platform's Main kelas dijalankan. Tersedia modul berada, ditempatkan dalam sebuah memori registri, dan modul 'startup tugas-tugas yang dilaksanakan. Secara umum, kode modul di-load ke memori hanya ketika diperlukan.

Aplikasi dapat menginstal modul secara dinamis. Setiap aplikasi dapat memasukkan modul *Update Center* untuk mengijinkan pengguna aplikasi untuk men-download ditandatangani *digital upgrade* dan fitur-fitur baru secara langsung ke aplikasi yang dijalankan.Layanan menawarkan platform umum untuk aplikasi desktop, mengijinkan pengembang untuk fokus pada logika aplikasi yang spesifik untuk mereka. Di antara fitur platform adalah:

1. Manajemen antarmuka pengguna (misalnya menu dan toolbar)
2. Pengaturan pengguna manajemen

3. *Storage management* (tabungan dan memuat segala jenis data)
4. Pengelolaan jendela
5. *Wizard framework* (mendukung langkah-demi-langkah dialog)

NetBeans IDE adalah sebuah aplikasi open source lingkungan pengembangan terintegrasi seluruhnya ditulis dalam Java menggunakan NetBeans Platform. NetBeans IDE mendukung pengembangan semua tipe aplikasi *Java (Java SE,EJB dan mobile aplikasi)* di luar kotak. Di antara fitur-fitur lainnya adalah Ant proyek berbasis sistem control Versi.

Modularitas**:** Semua fungsi IDE disediakan oleh modul-modul. Setiap modul menyediakan fungsi didefinisikan dengan baik, seperti dukungan untuk bahasa Java, mengedit, atau dukungan bagi CVS *sistem versioning*, dan SVN. NetBeans memuat semua modul yang diperlukan untuk pembangunan Jawa dalam satu download, memungkinkan pengguna untuk mulai bekerja segera. Modul-modul juga mengijinkan NetBeans untuk diperpanjang. Fitur-fitur baru, seperti dukungan untuk bahasa pemrograman lain, dapat ditambahkan dengan menginstal modul tambahan.Sebagai contoh, *Sun Studio, Sun Java Studio Enterprise*, dan *Sun Java Studio Creator dari Su nMicrosystems* semua didasarkan pada NetBeans IDE.

2.9 MySQL

MySQL adalah suatu perangkat lunak database relasi (*Relational Database Management System atau RDBMS*), seperti halnya ORACLE, Postgresql, MSSQL, dan sebaginya. MySQL AB menyebut produknya sebagai database open source terpopuler di dunia. Berdasarkan riset dinyatakan bahwa di

platfrom web, dan baik untuk kategori open source maupun umum, MySQL adalah database yang paling banyak dipakai. Menurut perusahaan pengembangannya, MySQL telah terpasang di sekitar 3 juta komputer. Puluhan hingga ratusan ribu situs mengandalkan MySQL bekerja siang malam memompa data bagi para pengunjungnya.

2.9.1 Fitur MySQL

Kalau di seri 3.22 MySQL mulai diadopsi banyak orang dan meningkat populasi penggunanya, maka di seri 3.23 dan 4.0-lah terjadi banyak peningkatan dari sisi teknologi. Ini tidak terlepas dari tuntutan pemakai yang semakin mengandalkan MySQL, namun membutuhkan fitur-fitur yang lebih banyak lagi. Seri 3.23. Di seri 3.23 MySQL menambahkan tiga jenis tabel baru: pertama MyISAM, yang sampai sekarang menjadi tipe tabel default; kedua BerkeleyDB, yang pertama kali menambahkan kemampuan transaksi pada MySQL; dan ketiga InnoDB, primadona baru yang potensial. Seri 4.x.

Di seri yang baru berjalan hingga 4.0 tahap alfa ini, pengembang MySQL berjanji akan menjadikan MySQL satu derajat lebih tinggi lagi. Fitur-fitur yang sejak dulu diminta akan dikabulkan, seperti subselek (di 4.1), union (4.0), foreign key constraint (4.0 atau 4.1—meski InnoDB sudah menyediakan ini di 3.23.x), stored procedure (4.1), view (4.2), cursor (4.1 atau 4.2), trigger (4.1). MySQL AB tetap berdedikasi mengembangkan dan memperbaiki MySQL, serta mempertahankan MySQL sebagai database open source terpopuler.

Keunggulan MySQL :

1. MySQL begitu popular di kalangan Web adalah karena ia memang cocok bekerja di lingkungan tersebut. Pertama, MySQL tersedia di berbagai platform Linux dan berbagai varian Unix. Sesuatu yang tidak dimiliki Access, misalnya—padahal Access amat popular di platform Windows. Banyak server Web berbasiskan Unix, ini menjadikan Access otomatis tidak dapat dipakai karena ia pun tidak memiliki kemampuan client-server/networking.
2. fitur-fitur yang dimiliki MySQL memang yang biasanya banyak dibutuhkan dalam aplikasi Web. Misalnya, klausa LIMIT SQL-nya, praktis untuk melakukan paging. Atau jenis indeks field FULLTEXT, untuk full text searching. Atau sebutlah kekayaaan fungsi-fungsi builtinnya, mulai dari memformat dan memanipulasi tanggal, mengolah string, regex, enkripsi dan hashing. Yang terakhir misalnya, praktis untuk melakukan penyimpanan password anggota situs.
3. MySQL memiliki overhead koneksi yang rendah. Soal kecepatan melakukan transaksi atau kinerja di kondisi load tinggi mungkin bisa diperdebatkan dengan berbagai benchmark berbeda, tapi kalau soal yang satu ini MySQL-lah juaranya. Karakteristik ini membuat MySQL cocok bekerja dengan aplikasi CGI, di mana di setiap request skrip akan melakukan koneksi, mengirimkan satu atau lebih perintah SQL, lalu memutuskan koneksi lagi. Cobalah melakukan hal ini dengan Interbase atau bahkan Oracle. Maka dengan load beberapa request per detik saja server Web/database Anda mungkin akan segera menyerah karena tidak bisa mengimbangi beban ini.

2.10 Perangkat Pendukung

Perangkat lunak pendukung adalah perangkat lunak yang digunakan dalam merancang sistem dari mulai *coding* sampai implementasinya, berikut ini adalah perangkat lunak yang akan digunakan dalam membangun Sistem Informasi:

UML menurut Boch et al, (1999,p14) dalam Gintoro, Andreyus, Emilia, Richard William (2010; 10) adalah UML adalah sebuah bahasa yang telah menjadi standar dalam industri untuk menvisualisasi, menspesifikasi, merancang dan mendokumentasi sistem piranti lunak

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 *syntax*/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna

tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (Object-Oriented Design), Jim Rumbaugh OMT (Object Modeling Technique), dan Ivar Jacobson OOSE (Object-Oriented Software Engineering).

2.10.1 Use Case Diagram

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

2.10.2 Class Diagram

Class diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut

2.10.3 Activity Diagram

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

Activity diagram merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak.

2.10.4 Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar Sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu

2.10.5 Collaboration Diagram

Collaboration diagram juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*. Setiap *message* memiliki *sequence number*, di mana *message* dari level tertinggi memiliki nomor 1. *Messages* dari level yang sama memiliki prefiks yang sama.