BAB II

TINJAUAN PUSTAKA

2.1 Profil Tempat Kerja Praktek

2.1.1 Sejarah *.Ore*™

Berawal dari ide – ide pokok dari impian masing – masing kami untuk membuat usaha yang lebih memprioritaskan segi kesenangan dan pembelajaran tanpa menghilangkan minat serta ketertarikan kami dalam bidang grafis, yang awalnya diberi nama BOHEMIANS pada tanggal 14 Desember 2008. Lalu setelah ada peristiwa – peristiwa yang sempat membuat semangat kami redup, akhirnya kami memutuskan untuk mengubah nama yang pada akhirnya ***.Ore***™ resmi di bentuk pada tanggal 02.09.2009 jam 21.00 WIB di Coffee Break Cafe Jalan Reog, Bandung Jawa Barat.

Mengambil kata .Ore dari Bahasa Inggris yang dalam Bahasa Indonesia berarti "**Barang Tambang**" berdasarkan itulah maka filosofi .Ore adalah bakat yang dimiliki tiap individu bagaikan barang tambang yang sangat bisa diolah sedemikian rupa dan sejalur dengan misi kami mengembangkan bakat personal individu tersebut, lingkungan sekitar, atau kelompok masyarakat.

A. Visi

Mengembangkan bakat personil individu menjadi lebih baik dari yang di bayangkan

B. Misi

Menciptakan karya yang bernilai, memiliki manfaat bagi pembeli,

lingkungan dan .Ore sendiri.

C. Tugas

***Ore*™** bidang grafis ini mempunyai tugas mengembangkan sumber daya manusia yang sukar untuk di apresiasikan atau salah satu wadah pengembangan bakat di bidang grafis yang di olah secara real yang memiliki nilai jual yang tinggi.

D. Fungsi

Dalam proses menyelenggarakan tugas tersebut, ***..Ore*™** bidang grafis ini mempunyai fungsi:

1. Mengendorse cipta dari para pemilik bakat
2. Penyusunan program pengembangan dan pemberdayaan skill individu (bakat)
3. Pengelolaan data dan informasi peningkatan bakat
4. Fasilitasi dan pelaksanaan peningkatan bakat
5. Evaluasi program dan fasilitasi peningkatan bakat

2.1.2 Logo Instansi



Gambar 1.2 Logo *.Ore*™

Digambarkan hanya dalam bentuk tulisan untuk memepermudah orang – orang dalam memahami ***.Ore*™.**

Dasar warna hitam diambil karena hitam merupakan warna dasar yang

lazim digunakan dimana saja (mencitrakan bahwa pandangan .Ore adalah umum dan sama dengan yang lainnya), sehingga apabila dalam pengolahannya ***.Ore***$^{\textbf{TM.}}$ hanya perlu memasukan warna – warna lain, karena dasar dari ***.Ore***$^{\textbf{TM.}}$ adalah "mencari barang tambang" yang mentah untuk kemudian kami olah hingga mereka menemukan ciri khasnya masing – masing.

Sedangkan penggunaan font **Calibri** dalam tifografi ***.Ore***$^{\textbf{TM.}}$ mengesankan bahwasannya kami terlihat serius tapi santai, titik di belakang huruf O adalah tanda bahwa kami memulai segala sesuatunya dari akhir yang buruk serta mengharapkan bahwa ***.Ore***$^{\textbf{TM.}}$ ini merupakan titik balik yang sangat baik untuk diri pribadi serta orang – orang yang terlibat di dalamnya.

2.1.3 Tagline Instansi

".Ore You Can" adalah susunan kata yang bermakna ambiguitas sesuai dari subject atau object yang dimunculkan pada saat membacanya. Bisa berarti kami bisa dalam segala hal yang kami lakukan atau berupa sebuah pertanyaan : "Apakah Kamu Bisa Seperti Kami ?"

2.1.4 Produk Instansi

Selain dalam bidang graphic service, ***.Ore***$^{\textbf{TM}}$ pun memiliki beberapa sub yang mempunyai fokusnya masing-masing yang menjadi penupang ***.Ore***$^{\textbf{TM}}$ itu sendiri, diantaranya :

1. This Ain't T-shirt

Sebuah line clothes dari ***.Ore***$^{\textbf{TM}}$ yang berbentuk kaos. Nama This Ain't T-shirt dipilih karena konsepnya menjadikan kaos yang lebih dari sekedar kaos

yang biasa di pakai dengan grafis yang menarik, tetapi juga sebagai ungkapan apa yang dirasakan oleh pemakainya.

2. This Ain't T-Bag

Serupa dengan T.A.T hanya produk ini berbentuk tas yang bisa di pesan sesuai dengan keinginan pemesannya.

3. T.T.L *(Trough To The Lens)*

Sebuah service kami dalam bidang fotografi dan video editing.

4. B.O.H.M

Sub kami dalam bidang event ograganizer.

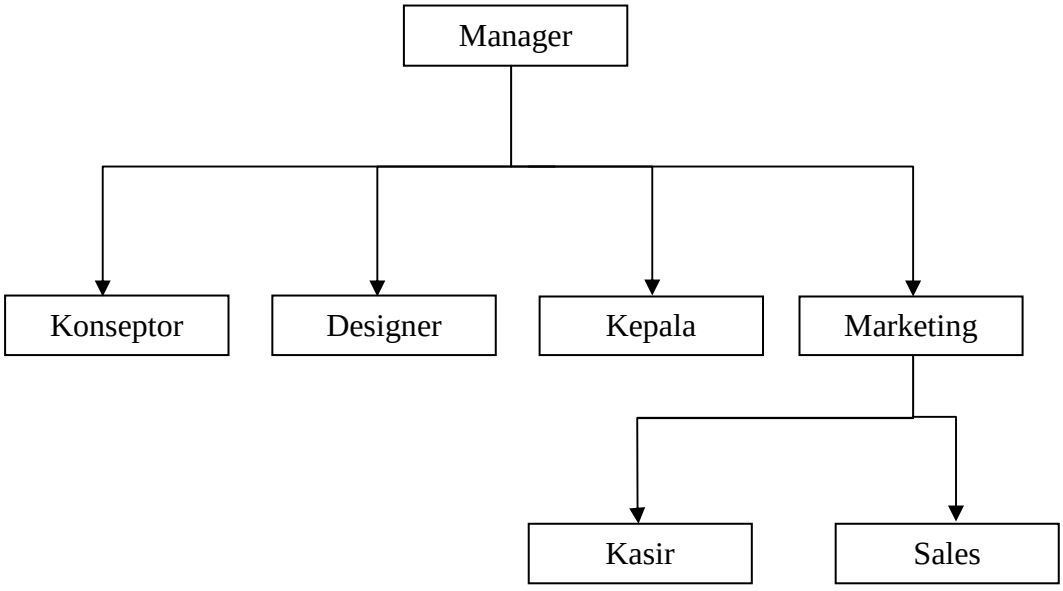
5. D.O.T

Adalah pelayanan kami dalam Graphic Design semisal pembuatan logo atau desain layout.

2.1.5 Badan Hukum Instansi

Badan hukum ***.ore***^{TM} adalah Badan Usaha Milik Pribadi.

2.1.6 Struktur Organisasi



2.1.7 Job Dask

A. Manager

Memonitoring dan memanage kinerja para pegawai agar berjalan sesuai dengan yang direncanakan. Manager juga bertugas mengarahkan para pegawai agar tidak keluar jalur dari aturan – aturan yang sudah ada.

B. Konseptor

Melakukan pengonsepan dari setiap barang atau produk yang akan dipasarkan dan juga menjawab semua ambiguitas dan kompleksifitas yang kemudian dirangkum dalam sebuah konsep.

C. Designer

Mendesain produk – produk yang akan di produksi seperti desain sepatu, jaket, baju, kemeja, celana dan mendesain toolkit – toolkit yang akan dipakai oleh marketing dalam hal memasarkan produk.

D. Kepala produksi

Memonitoring dibagian produksi (proses pembuatan produk) agar hasil produksi sesuai dengan yang diinginkan sehingga bisa menyamakan konsep yang sudah ditentukan.

E. Marketing

Memasarkan barang yang sudah diproduksi dan juga mencari job – job tambahan untuk menghasilkan penghasilan tambahan sehingga penghasilan tidak hanya bergantung dari penjualan produk saja, seperti halnya pemesanan baju dll dalam jumlah yang besar.

a. Sales

Sales adalah orang yang bekerja dibagian pemasaran sebagai bawahan dari marketing, dia bertugas untuk memasarkan produk diluar lapangan berdasarkan perintah dari marketing.

b. Kasir

Kasir bertugas untuk mencatat semua proses yang berhubungan dengan penjualan didalam lingkungan perusahaan.

2.2 Landasan Teori

Landasan teori menjelaskan beberapa teori yang mendukung aplikasi yang dibuat dan berkaitan dengan permasalahan yang dibahas sebagai pemahaman dalam sebuah sistem serta metode yang dipakai untuk kegiatan pengembangan sistem sendiri.

2.2.1 Sistem

Sistem berasal dari bahasa Latin (*systema*) dan bahasa Yunani (*sustema*) adalah suatu kesatuan yang terdiri komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi. Istilah ini sering dipergunakan untuk menggambarkan suatu set entitas yang berinteraksi, di mana suatu model matematika seringkali bisa dibuat.

Sistem juga merupakan kesatuan bagian-bagian yang saling berhubungan yang berada dalam suatu wilayah serta memiliki item-item penggerak, contoh umum misalnya seperti negara. Negara merupakan suatu kumpulan dari beberapa elemen kesatuan lain seperti provinsi yang saling berhubungan sehingga membentuk suatu negara dimana yang berperan sebagai penggeraknya yaitu

rakyat yang berada dinegara tersebut.

Kata "sistem" banyak sekali digunakan dalam percakapan sehari-hari, dalam forum diskusi maupun dokumen ilmiah. Kata ini digunakan untuk banyak hal, dan pada banyak bidang pula, sehingga maknanya menjadi beragam. Dalam pengertian yang paling umum, sebuah sistem adalah sekumpulan benda yang memiliki hubungan di antara mereka.

2.2.2 Aplikasi Kasir

Aplikasi Kasir merupakan perangkat lunak (*software*) yang dibuat untuk memudahkan aktivitas dan pencatatan akuntansi dengan memanfaatkan konsep modularitas atas serangkaian aktivitas yang serupa ke dalam modul-modul spesifik seperti pembelian (Account payable), penjualan (Account receivable), dan lain-lain.

Aplikasi ini bisa merupakan perangkat lunak yang dikembangkan sendiri oleh perusahaan, atau dibeli dari pihak ketiga yang menyediakannya, atau dapat pula merupakan kombinasi dari keduanya. Karena hal tersebut, kompleksitas dan kapabilitas Aplikasi Kasir ini menjadi sangat beragam bergantung pada kondisi lingkungan perusahaan yang akan menggunakannya.

Pada prinsipnya Aplikasi Kasir ditujukan untuk melakukan proses transaksi penjualan. Untuk dapat menghasilkan proses tersebut, langkah mendasar adalah menyusun perkiraan proses terjadi. Dimana setiap proses memiliki kode dan nama (yang lazim digunakan) serta memiliki pos-pos yang jelas dalam posisi transaksi, kode barang untuk form barang dan lain lain.

2.2.3 Sistem Operasi

Sistem operasi (bahasa Inggris: ***operating system*** ; ***OS***) adalah seperangkat program yang mengelola sumber daya perangkat keras komputer, dan menyediakan layanan umum untuk aplikasi perangkat lunak. Sistem operasi adalah jenis yang paling penting dari perangkat lunak sistem dalam sistem komputer. Tanpa sistem operasi, pengguna tidak dapat menjalankan program aplikasi pada komputer mereka, kecuali program aplikasi boot diri.

Waktu-berbagi jadwal tugas sistem operasi untuk penggunaan yang efisien dari sistem dan juga dapat mencakup akuntansi untuk alokasi biaya waktu prosesor, penyimpanan massa, cetak, dan sumber daya lainnya.

Untuk fungsi-fungsi perangkat keras seperti sebagai masukan dan keluaran dan alokasi memori, sistem operasi bertindak sebagai perantara antara program aplikasi dan perangkat keras komputer, meskipun kode aplikasi biasanya dieksekusi langsung oleh perangkat keras dan seringkali akan menghubungi OS atau terputus oleh itu. Sistem operasi yang ditemukan pada hampir semua perangkat yang berisi komputer-dari ponsel dan konsol permainan video untuk superkomputer dan server web.

Contoh populer sistem operasi modern termasuk Linux, Android, iOS, Mac OS X, dan Microsoft Windows.

2.2.4 Bahasa Pemrograman

Bahasa pemrograman, atau sering diistilahkan juga dengan bahasa komputer, adalah teknik komando/instruksi standar untuk memerintah komputer. Bahasa pemrograman ini merupakan suatu set aturan sintaks dan semantik yang

dipakai untuk mendefinisikan program komputer. Bahasa ini memungkinkan seorang programmer dapat menentukan secara persis data mana yang akan diolah oleh komputer, bagaimana data ini akan disimpan/diteruskan, dan jenis langkah apa secara persis yang akan diambil dalam berbagai situasi.

2.2.5 OOP (*object-oriented programming*)

Pemrograman berorientasi objek (Inggris: ***object-oriented programming*** disingkat **OOP**) merupakan paradigma pemrograman yang berorientasikan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus dalam *kelas-kelas* atau *objek-objek*. Bandingkan dengan logika pemrograman terstruktur. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya, Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam teknik piranti lunak skala besar.

Konsep dasar dari Pemrograman Berorientasi Objek Pemrograman orientasi-objek menekankan konsep berikut:

1. Kelas — kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebagai contoh 'class of dog' adalah suatu unit yang terdiri atas definisi-definisi data dan fungsi-fungsi yang menunjuk pada berbagai macam perilaku/turunan dari anjing. Sebuah class adalah dasar dari modularitas dan struktur dalam pemrograman berorientasi object. *Sebuah class secara tipikal sebaiknya dapat dikenali oleh seorang non-programmer sekalipun terkait dengan domain permasalahan yang ada,* dan kode yang

terdapat dalam sebuah class sebaiknya (relatif) bersifat mandiri dan independen (sebagaimana kode tersebut digunakan jika tidak menggunakan OOP). Dengan modularitas, struktur dari sebuah program akan terkait dengan aspek-aspek dalam masalah yang akan diselesaikan melalui program tersebut.

2. Objek - membungkus data dan fungsi bersama menjadi suatu unit dalam sebuah program komputer; objek merupakan dasar dari modularitas dan struktur dalam sebuah program komputer berorientasi objek.
3. Abstraksi - Kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti. Setiap objek dalam sistem melayani sebagai model dari "pelaku" abstrak yang dapat melakukan kerja, laporan dan perubahan keadaannya, dan berkomunikasi dengan objek lainnya dalam sistem, tanpa mengungkapkan bagaimana kelebihan ini diterapkan.
4. Enkapsulasi - Memastikan pengguna sebuah objek tidak dapat mengganti keadaan dalam dari sebuah objek dengan cara yang tidak layak, hanya metode dalam objek tersebut yang diberi izin untuk mengakses keadaannya. Setiap objek mengakses interface yang menyebutkan bagaimana objek lainnya dapat berinteraksi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut.
5. Polimorfisme melalui pengiriman pesan. Tidak bergantung kepada

pemanggilan subrutin, bahasa orientasi objek dapat mengirim pesan; metode tertentu yang berhubungan dengan sebuah pengiriman pesan tergantung kepada objek tertentu di mana pesan tersebut dikirim. Contohnya, bila sebuah burung menerima pesan "gerak cepat", dia akan menggerakan sayapnya dan terbang. Bila seekor singa menerima pesan yang sama, dia akan menggerakkan kakinya dan berlari. Keduanya menjawab sebuah pesan yang sama, namun yang sesuai dengan kemampuan hewan tersebut. Ini disebut polimorfisme karena sebuah variabel tungal dalam program dapat memegang berbagai jenis objek yang berbeda selagi program berjalan, dan teks program yang sama dapat memanggil beberapa metode yang berbeda di saat yang berbeda dalam pemanggilan yang sama. Hal ini berlawanan dengan bahasa fungsional yang mencapai polimorfisme melalui penggunaan fungsi kelas-pertama.

Dengan menggunakan OOP maka dalam melakukan pemecahan suatu masalah kita tidak melihat bagaimana cara menyelesaikan suatu masalah tersebut (terstruktur) tetapi objek-objek apa yang dapat melakukan pemecahan masalah tersebut. Jadi untuk menyelesaikan suatu masalah dengan kolaborasi antar objek-objek yang ada karena setiap objek memiliki deskripsi tugasnya sendiri.

2.2.6 Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian

dari Oracle dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi-aplikasi berbasis java umumnya dikompilasi ke dalam p-code (*bytecode*) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM). Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (*general purpose*), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa platform sistem operasi yang berbeda, java dikenal pula dengan slogannya, "*Tulis sekali, jalankan di mana pun*". Saat ini java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbasis web.

A. Kelebihan

a. ***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 Solaris. Penyebanya adalah setiap sistem operasi menggunakan programnya sendiri-sendiri (yang dapat diunduh dari situs Java) untuk meninterpretasikan *bytecode* tersebut.

b. **Perpustakaan Kelas Yang Lengkap**, Java terkenal dengan kelengkapan *library*/perpustakaan (kumpulan program program yang disertakan dalam pemrograman java) yang sangat memudahkan dalam penggunaan oleh para pemrogram untuk membangun aplikasinya.

c. **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.

Pengumpulan sampah otomatis, memiliki fasilitas pengaturan penggunaan memori sehingga para pemrogram tidak perlu melakukan pengaturan memori secara langsung (seperti halnya dalam bahasa C++ yang dipakai secara luas).

B. Kekurangan

a. **Tulis sekali, jalankan di mana saja** - Masih ada beberapa hal yang tidak kompatibel antara *platform* satu dengan *platform* lain. Untuk J2SE, misalnya *SWT-AWT bridge* yang sampai sekarang tidak berfungsi pada Mac OS X.

b. **Mudah didekompilasi**. Dekompilasi adalah proses membalikkan dari kode jadi menjadi kode sumber. Ini dimungkinkan karena kode jadi

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/di*reverse-engineer*.

c. **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.

2.2.7 Database

Pangkalan data atau **basis data** (bahasa Inggris: *database*), atau sering pula dieja **basisdata**, adalah kumpulan informasi yang disimpan di dalam komputer secara sistematik sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Perangkat lunak yang digunakan untuk mengelola dan memanggil kueri (*query*) basis data disebut sistem manajemen basis data (*database management system*, DBMS). Sistem basis data dipelajari dalam ilmu informasi.

Istilah "basis data" berawal dari ilmu komputer. Meskipun kemudian artinya semakin luas, memasukkan hal-hal di luar bidang elektronika, artikel ini

mengenai basis data komputer. Catatan yang mirip dengan basis data sebenarnya sudah ada sebelum revolusi industri yaitu dalam bentuk buku besar, kuitansi dan kumpulan data yang berhubungan dengan bisnis.

Konsep dasar dari basis data adalah kumpulan dari catatan-catatan, atau potongan dari pengetahuan. Sebuah basis data memiliki penjelasan terstruktur dari jenis fakta yang tersimpan di dalamnya: penjelasan ini disebut skema. Skema menggambarkan obyek yang diwakili suatu basis data, dan hubungan di antara obyek tersebut. Ada banyak cara untuk mengorganisasi skema, atau memodelkan struktur basis data: ini dikenal sebagai model basis data atau model data. Model yang umum digunakan sekarang adalah model relasional, yang menurut istilah layman mewakili semua informasi dalam bentuk tabel-tabel yang saling berhubungan dimana setiap tabel terdiri dari baris dan kolom (definisi yang sebenarnya menggunakan terminologi matematika). Dalam model ini, hubungan antar tabel diwakili denga menggunakan nilai yang sama antar tabel. Model yang lain seperti model hierarkis dan model jaringan menggunakan cara yang lebih eksplisit untuk mewakili hubungan antar tabel.

Istilah *basis data* mengacu pada koleksi dari data-data yang saling berhubungan, dan perangkat lunaknya seharusnya mengacu sebagai *sistem manajemen basis data* (*database management system/DBMS*). Jika konteksnya sudah jelas, banyak administrator dan programer menggunakan istilah basis data untuk kedua arti tersebut.

Perangkat lunak basis data yang banyak digunakan dalam pemrograman dan merupakan perangkat basis data tingkat tinggi (*high level*): Microsoft SQL

Server, Oracle, Sybase, Interbase, XBase, Firebird, MySQL, PostgreSQL, Microsoft Access, dBase III, Paradox, FoxPro, Visual FoxPro, Arago, Force, Recital, dbFast, dbXL, Quicksilver, Clipper, FlagShip, Harbour, Visual dBase, Lotus Smart Suite Approach, db2.

Selain perangkat lunak di atas, terdapat juga perangkat lunak pemrograman basis data aras rendah (*low level*), diantaranya: Btrieve, Tsunami Record Manager

2.2.8 UML

Unified Modelling Language (UML) adalah sebuah "bahasa" yang 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 termasuk bahasa JAVA.

Adapun bagian – bagian yang termasuk kedalam UML yang d Laporan ini:

a. 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.

b. Squence 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 atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

c. Activity Diagram

Activity diagram 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 oleh selesainya *state* sebelumnya.

d. Class Diagram

Class diagram merupakan diagram yang selalu ada di permodelan sistem berorientasi objek. Class diagram menunjukkan hubungan antar *class* dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan.

2.2.9 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 memiliki beberapa keistimewaan, antara lain :

1. **Portabilitas**.

   MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.

2. **Perangkat lunak sumber terbuka**

   MySQL didistribusikan sebagai perangkat lunak sumber terbuka, dibawah lisensi GPL sehingga dapat digunakan secara gratis.

3. **Multi-user**

   MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.

4. '**Performance Tuning**'

   MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

5. **Ragam Tipe Data**

   MySQL memiliki ragam tipe data yang sangat kaya, seperti *signed / unsigned integer, float, double*, *char, text, date, timestamp*, dan lain-lain.

6. **Perintah dan Fungsi**

   MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam perintah (*query*).

7. **Keamanan**

MySQL memiliki beberapa lapisan keamanan seperti level subnetmask, nama host, dan izin akses *user* dengan sistem perizinan yang mendetail serta sandi terenkripsi.

8. **Skalabilitas dan Pembatasan**

   MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (records) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

9. **Konektivitas**

   MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix soket (UNIX), atau Named Pipes (NT).

10. **Lokalisasi**

    MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.

11. **Antar Muka**

    MySQL memiliki antar muka (interface) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi *API* (*Application Programming Interface*).

12. **Klien dan Peralatan**

    MySQL dilengkapi dengan berbagai peralatan (tools) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk *online*.

13. Struktur tabel

MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle.

2.2.10 Netbeans

NetBeans adalah *Integrated Development Environment* (IDE) berbasiskan Java dari Sun Microsystems yang berjalan di atas Swing. Swing sebuah teknologi Java untuk pengembangan aplikasi Desktop yang dapat bejalan di berbagai macam platforms seperti Windows, Linux, Mac OS X and Solaris.

NetBeans mengacu pada kedua platform kerangka untuk aplikasi desktop Java, dan sebuah lingkungan pengembangan terpadu (IDE) untuk pengembangan dengan Java , JavaScript , PHP , Python , Ruby , Groovy , C , C + + , Scala , Clojure , dan lain-lain.