BAB 2
TINJAUAN PUSTAKA

2.1 Tinjauan Perusahaan

2.1.1 Sejarah dan Perkembangan Madgoat Strore

Madgoat Store adalah sebuah perusahaan yang membuat produk dibidang *fashion*, perusahaan ini merupakan milik perseorangan. Perusahaan ini didirikan tahun 2009 di Cimahi. Semula perusahaan ini bernama Deadbeat Inc., karena beberapa faktor perusahaan ini berubah nama menjadi Madgoat Store. Nama tersebut dipilih karena sesuai dari karaker pemiliknya. Pertama didirikan perusahaan ini hanya memproduksi *t-shirt*, seiiring dengan kepercayaan dari rekan kerja, perusahaan ini mulai membuat *jacket, sweeter* dan aksesoris lain yang berhubungan dengan dunia *fashion*.Tahun ini Madgoat Store mulai membuka toko mereka sendiri yang diberi nama Madgoat Shop. Disini mereka menjual produk yang mereka buat sendiri, selain menjual di Madgoat Shop, mereka juga mendistribusikan produk yang mereka buat ke beberapa distro atau *outlet* di kota Bandung dan Cimahi.

2.1.2 Visi dan Misi Madgoat Store

2.1.2.1 Visi

Terpecaya dan berstandar dalam membuat produk.

2.1.2.2 Misi

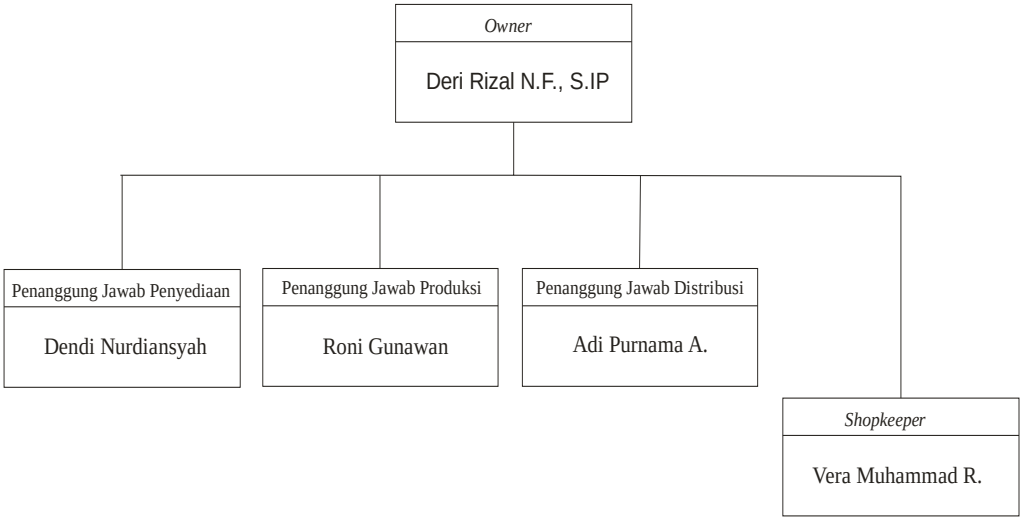
1. Membuat produk lokal lebih dipilih oleh konsumen dibandingkan dengan produk dari luar.
2. Memperluas kejasama dengan perusahaan lain khususnya perusahaan dibidang *fashion*.
3. Mengembangkan produk yang sesuai dengan perkembangan zaman.

2.1.3 Lokasi dan Jam Kerja

Madgoat Store beralamat di Jl. Leuwigajah no.152 Cimahi, dengan nomor telepon 087825747882. Madgoat Store ini beroperasi dari jam 10.00 – 20.00 setiap harinya.

2.1.4 Struktur Organisasi

Adapun struktur organisasi dan tugas atau pekerjaan di masing-masing bagian di Madgoat Store adalah sebagai berikut :



2.1 Gambar Struktur Organisasi

Penjelasan tugas di masing-masing bagian adalah sebagai berikut :

1. *Owner* bertugas untuk mengawasi semua aktivitas yang terjadi di semua bagian.
2. Penanggung jawab penyediaan bertugas untuk melakukan pembelanjaan bahan mentah dan melakukan pendataan hasil belanja.
3. Penanggung jawab produksi bertugas untuk mengawasi proses produksi dan memberi perintah untuk melakukan produsi. Selain itu petugas ini juga melalukan pendataaan hasil produksi.
4. Penanggung jawab distribusi bertugas untuk mengawasi proses distribusi dan memberi perintah untuk melakukan distribusi.

2.2 Landasan Teori

2.2.1 Pengertian Aplikasi

Aplikasi adalah kumpulan perintah program yang dibuat untuk melakukan pekerjaan-pekerjaan tertentu (khusus) [3]. Aplikasi merupakan penerapan, menyimpan hal, data, permasalahan pekerjaan suatu sarana atau media yang digunakan untuk menerapkan atau mengimplikasikan hal atau permasalahan tersebut, sehingga berubah menjadi suatu bentuk yang baru tanpa menghilangkan nilai-nilai dasar dari hal, data, permasalahan atau pekerjaan. Jadi dalam hal ini hanya bentuk dari tampilan data yang berubah, sedangkan isi yang termuat dalam data tersebut tidak mengalami perubahan.

Jadi program aplikasi adalah sekumpulan instruksi yang tersusun menurut urutan yang logis dan dikenal oleh komputer atau program yang dipakai untuk melakukan atau mempermudah suatu pekerjaan.

2.2.2 Pengertian Penyediaan

Penyediaan adalah barang dagangan yang dibeli untuk dijual kembali dan masih ada di gudang berupa stok. Penyediaan di bidang manufaktur dapat dibedakan menjadi 3 bagian, yaitu pemyediaan berupa bahan baku (*raw material*), penyediaan barang berupa proses (*goods in process*), dan penyediaan barang jadi (*finished goods*) [9]. Terdapat enam fungsi penting yang dikandung oleh penyediaan dalam memenuhi kebutuhan perusahaan, antara lain :

a. Menghilangkan resiko keterlambatan pengiriman bahan baku atau barang yang dibutuhkan perusahaan.
b. Menghilangkan resiko jika material yang dipesan tidak baik sehingga harus dikembalikan
c. Menghilangkan resiko terhadap kenaikan harga barang atau inflasi.
d. Untuk menyimpan bahan baku yang dihasilkan secara musiman sehingga perusahaan tidak akan kesulitan bila bahan tersebut tidak tersedia di pasaran.
e. Mendapatkan keuntungan dari pembelian berdasarkan potongan kuantitas (*Quantity discount*).

f. Memberikan pelayanan kepada langganan dengan tersedianya bahan yang diperlukan.

Jenis-jenis penyediaan dalam suatu perusahaan menurut fungsinya dapat dibedakan atas :

1. *Bath Stock/Lot Size Inventory* adalah penyediaan yang diadakan karena kita membeli atau membuat bahan-bahan atau barang-barang dalam jumlah yang lebih besar daripada jumlah yang dibutuhkan pada saat itu.
   Keuntungannya:
   a. Potongan harga pada harga pembelian.
   b. Efisiensi produksi.
   c. Penghematan biaya angkutan.
2. *Fluctuation Stock* adalah penyediaan yang diadakan untuk menghadapi fluktuasi permintaan konsumen yang tidak dapat diramalkan.
3. *Anticipation Stock* adalah penyediaan yang diadakan untuk menghadapi fluktuasi permintaan yang dapat diramalkan, berdasarkan pola musiman yang terdapat dalam satu tahun dan untuk menghadapi penggunaan, penjualan, atau permintaan yang meningkat.

2.2.3 Pengertian Distribusi

Distribusi adalah serangkaian kegiatan menyalurkan barang atau jasa dari produsen kepada konsumen [11]. Proses distribusi tersebut pada dasarnya menciptakan faedah (*utility*) waktu, tempat, dan pengalihan hak milik.

Dalam menciptakan ketiga faedah tersebut, terdapat dua aspek penting yang terlibat didalamnya, yaitu :

1. Lembaga yang berfungsi sebagai saluran distribusi (*Channel of distribution/marketing channel*).
2. Aktivitas yang menyalurkan arus fisik barang (*Physical distribution*).

Adapun tujuan distribusi adalah sebagai berikut :

a. Menyampaikan barang atau jasa dari produsen kepada konsumen.
b. Mempercepat sampainya hasil produsen kepada konsumen.
c. Tercapainya pemerataan produksi.

d. Menjaga kesinambungan produksi.
e. Memperbesar dan meningkatkan kualitas dan kuantitas produksi.
f. Meningkatnya nilai guna barang atau jasa.

Jenis – jenis distribusi barang atau jasa:

1. Produsen ---> Konsumen (Umumnya Jasa)
   Contoh : Bengkel, Rumah Makan, Salon, Panti Pijit, dsb
2. Produsen ---> *Retailer* ---> Konsumen
   Contoh : Koran, Es Krim, dll.
3. Produsen ---> *Wholesaler* ---> *Retailer* ---> Konsumen
   Contoh : Mie Instan, Beras, Sayur-Mayur, Minuman Dalam Kemasan
4. Produsen ---> Agen ---> *Wholesaler* ---> *Retailer* ---> Konsumen
   Contoh : Barang Impor
5. Produsen ---> Industri (Produsen)
   Contoh : Pabrik mie telor menjual produknya ke pedagang mie ayam.
6. Produsen ---> *Wholesaler* ---> Industri (Produsen)
   Contoh : Suatu distributor membeli mesin berat dari luar negeri untuk dijual ke pabrik-pabrik di dalam negeri.

2.2.4 *Eqonomic Order Quantity*

Economic Order Quantity (EOQ) adalah teknik pengendalian permintaan / pemesanan barang dengan biaya *inventory* serendah mungkin [13]. Dalam kegiatan normal *Model Economic Order Quantity* memiliki beberapa karakteristik antara lain :

a. Jumlah barang yang dipesan pada setiap pemesanan selalu konstan,
b. Permintaan konsumen, biaya pemesanan, biaya transportasi dan waktu antara pemesanan barang sampai barang tersebut dikirim dapat diketahui secara pasti, dan bersifat konstan.
c. Harga per unit barang adalah konstan dan tidak mempengaruhi jumlah barang yang akan dipesan nantinya, dengan asumsi ini maka harga beli menjadi tidak relevan untuk menghitung EOQ, karena ditakutkan pada

nantinya harga barang akan ikut dipertimbangkan dalam pemesanan barang,

d. Pada saat pemesanan barang, tidak terjadi kehabisan barang atau *back order* yang menyebabkan perhitungan menjadi tidak tepat. Oleh karena itu, manajemen harus menjaga jumlah pemesanan agar tidak terjadi kehabisan barang,

e. Pada saat penentuan jumlah pemesanan barang kita tidak boleh mempertimbangkan biaya kualitas barang,

f. Biaya penyimpanan per unit konstan.

Perumus EOQ [4] :

- Biaya pemesanan (bp) per tahun :

  bp = F x S

  bp = (D/Q) x S ……..… (2.1)

- Biaya penyimpanan (bpi) per tahun :

  bpi = persediaan rata-rata x biaya penyimpanan

  bpi = (Q/2) x H ……. (2.2)

- EOQ terjadi bila biaya pemesanan = penyimpanan

$$(D/Q) \times S = (Q/2) \times H$$

$$2DS = HQ^2$$

$$Q^2 = (2DS) / H$$

Maka $Q' = (\sqrt{2DS / H})$ …… (2.3)

- Rumus untuk menentukan frekuensi pesanan

$$F = \frac{D}{Q}$$ …… (2.4)

- Rumus untuk menentukan jarak pesanan

$$T = \frac{\text{Jumlah hari satu tahun}}{F}$$ ……..…….. (2.5)

Keterangan :

D = jumlah kebutuhan barang (unit/tahun)

S = biaya pemesanan atau biaya *setup* (rupiah/pesanan)

h = biaya penyimpanan (% terhadap nilai barang)

C = harga barang (rupiah/unit)

H = h x C = biaya penyimpanan (rupiah/unit/tahun)

Q = jumlah pemesanan (unit/pesanan)

F = frekuensi pemesanan (kali/tahun)

T = jarak waktu pesanan (tahun, hari)

TC = biaya total pemesanan

Rumus biaya total pemesanan :

TC = Biaya Pemesanan + Biaya Penyimpanan…….(2.6)

2.2.4.1 *Safety Stock*

Safety stock adalah tingkat penyediaan minimal yang harus selalu ada pada setiap periode, sehingga kita dapat mengantisipasi apabila terjadi lonjakan permintaan atau keterlambatan pengiriman [9]. Penyediaan tambahan yang diadakan untuk melindungi atau menjaga kemungkinan terjadinya kekurangan. Dengan adanya *safety stock* akan mengurangi stockout cost bagi perusahaan.

Rumus *safety stock* (SS) berdasarkan perbedaan pemakaian maksimum dan rata-rata :

Safety Stock = (Pemakaian maksimum – Pemakaian rata-rata) x *lead time*…(2.7)

Ada beberapa faktor yang menentukan besarnya penyediaan pengaman, yaitu:

a. Penggunaan bahan baku rata-rata. Hal ini perlu diperhatikan karena ketika kita mengadakan pemesanan pengganti maka pemenuhan permintaan dari langganan sebelum barang yang dipesan datang harus dapat dipenuhi dari *stock* yang ada atau yang disimpan.
b. Faktor waktu. Lamanya waktu antara mulai dilakukannya pemesanan bahan-bahan yang dipesan sampai pada bahan diterima digudang pesediaan.

Keterangan :

- Pemakaian maksimum adalah jumlah pemakaian barang dalam satu periode, misalkan pemakaian barang dalam satu minggu ataupun satu bulan.
- Pemakaian rata-rata adalah jumlah pemakaian barang rata-rata dalam satu periode tertentu, misalkan pemakaian rata-rata dalam satu minggu ataupun satu bulan.
- *Lead time* adalah waktu tenggang yang diperlukan untuk memesan (membuat) suatu barang sejak saat pesanan (pembuatan) dilakukan sampai barang itu diterima (selesai dibuat).

2.2.5 *Database*

Database adalah kumpulan fakta-fakta sebagai representasi dari dunia nyata yang saling berhubugan dan mempunyai arti tertentu. Secara singkat *database* dipakai untuk menyimpan data agar data tersebut dapat dimanipulasi dengan mudah, terjamin keakuratannya, efektif dalam penyimpanannya, dan tentu saja mudah untuk mengakses kembali [10].

Keuntungan sistem *database* :

1. Mengurangi redundasi (*redundanccy can be reduced*), data yang sama pada beberapa aplikasi cukup disimpan sekali saja.
2. Integrity (*integrity can be maintained*), data yang tersimpan secara akurat (akibatnya tidak akan terjadi *incorrect Information*).
3. Menghindarkan inkonsistensi, karena redundansi berkurang, sehingga umumnya *update* data hanya (karena tidak *reduncancy* dan tidak inkonsistensi maka informasi dijamin akurat).
4. Penggunaan data bersama (*the data can be shared*), data yang sama dapat diakses oleh beberapa *user* pada saat yang bersamaan).
5. Standarisasi (keseragaman penyajian data).
6. Jaminan Sekuriti (DBA mengatur & menjamin *security check* terhadap sensitif data), data hanya dapat diakses oleh yang berhak.

7. Menyeimbangkan kebutuhan (DBA mengatur tentang prioritas), dapat ditentukan prioritas suatu operasi, misal antara update dengan retriefal.

Kerugian sistem *database* :

1. Diperlukan *hardware* tambahan: CPU yang lebih kuat, terminal yang lebih banyak, alat untuk komunikasi.
2. Biaya *performance* yang lebih besar (mahal), listrik, personal yang lebih tinggi klasifikasinya, biaya telekomunikasi antar lokasi/kota.
3. Sistem tampak lebih komplek karena kemampuan *software* yang lebih besar menjadi rumit dan banyaknya aspek yg harus di perhatikan.
4. Rawannya keberhasilan operasi, karena gangguan teknis.

2.2.6 Jaringan *Client Server*

Jaringan *client server* menghubungkan komputer *server* dengan komputer klien/*workstation*. Komputer *server* adalah komputer yang menyediakan fasilitas bagi komputer – komputer *workstation* yang terhubung dalam jaringan, sedangkan komputer klien atau *workstation* adalah komputer yang menggunakan fasilitas dari komputer *server* [7].

Keuntungan dari jaringan *client server* adalah:

1. Terdapat *administrator* yang mengelola sistem keamanan dan administrasi jaringan, sehingga sistem keamanan dan administrasi jaringan akan lebih terkontrol.
2. Komputer *server* difungsikan sebagai pusat data, komputer *client* dapat mengakses data yang ada dari komputer *client* manapun. Apabila terdapat komputer *client* rusak, pengguna masih dapat mengakses data dari komputer *client* yang lain.
3. Pengaksesan data lebih tinggi karena penyediaan dan pengelolaan fasilitas jaringan dilakukan oleh komputer *server*, dan komputer *server* tidak terbebani dengan tugas lain seperti komputer *client*.
4. Pada tipe jaringan *client server*, sistem *backup* data lebih baik, karena *backup* data dapat dilakukan terpusat di komputer *server*. Apabila data di

client mengalami masalah atau kerusakan masih tersedia *backup* pada komputer *server*.

Kelemahan dari tipe jaringan *client server* adalah :

1. Biaya mahal, karenan membutuhkan komputer yang memiliki kemampuan tinggi yang difungsikan sebagai komputer *server*.
2. Kelancaran jaringan tergantung pada komputer *server*, bila komputer *server* mengalami gangguan maka jaringan akn terganggu.

Berikut ini macam-macam topologi jaringan komputer :

1. Topologi Bus

   Topologi bus merupakan topologi dimana semua perangakat keras terhubung melalui kabel tunggal yang kedua ujungnya tidak tertutup dan masing-masing ujungnya menggunakan sebuah perangkat terminator. Jika alamat perangkat sesuai dengan alamat pada informasi yang dikirim, maka informasi akan diterima dan diproses. Jika tidak, maka informasi akan diabaikan.

   Keuntungan topologi bus adalah :

   1. Jarak LAN tidak terbatas
   2. Kecepatan pengiriman tinggi.
   3. Tidak diperlukan pengendali pusat.
   4. Jumlah perangkat yang terhubung dapat dirubah tanpa mengganggu yang lain.
   5. Kemampuan pengembangan tinggi.
   6. Keterandalan jaringan tinggi.
   7. Kondusif untuk jaringan gedung bertingkat.

   Kerugian topologi bus adalah :

   1. Jika tingkat *traffic* tinggi dapat menyebabkan kemacetan.
   2. Diperlukan *repeater* untuk memperkuat sinyal.
   3. Operasional jaringan LAN tergantung tiap perangkat.

2. Topologi *Ring*

   Topologi ring merupakan topologi dimana setiap perangkat dihubungkan sehingga berbentuk lingkaran. Setiap informasi yang diperoleh akan

diperiksa alamatnya oleh perangkat jika sesuai maka informasi akan diproses sedangkan jika tidak maka informasi diabaikan.

Keuntungan topologi *ring* adalah :

1. Kecepatan pengiriman tinggi.
2. Dapat melayani *traffic* yang padat.
3. Tidak diperlukan *host*, relatif murah.
4. Dapat melayani berbagai mesin pengirim.
5. Komunikasi antar terminal mudah.
6. Waktu yang diperlukan untuk pengaksesan data optimal.

Kerugian topologi *ring* adalah :

1. Perubahan jumlah perangkat sulit.
2. Kerusakan pada media pengirim dapat mempengaruhi seluruh jaringan.
3. Harus memiliki kemampuan untuk mendeteksi kesalahan untuk kemudian di isolasi.
4. Kerusakan salah satu perangkat menyebabkan kelumpuhan jaringan.
5. Tidak baik untuk pengiriman suara, *video* dan data.

3. Topologi *Tree*

Topologi *tree* merupakan generalisasi dari topologi bus, media transmisi berupa kabel yang bercabang tanpa *loop* tertutup. Topologi *tree* selalu dimulai pada titik yang disebut headend. Satu atau beberapa kabel berasal dari *headend*.

Keuntungan topologi *tree* adalah :

1. Kontrol manajemen mudah karena bersifat terpusat.
2. Mudah untuk dikembangkan.

Kerugian topologi *tree* adalah :

1. Karena data yang dikirim diterima oleh semua perangkat diperlukan mekanisme untuk mengidentifikasi perangkat yang ingin di tuju.
2. Diperlukan mekanisme transmisi data untuk menghindari *overlapping* sinyal jika 2 perangkat mengirim data secara bersamaan.

4. Topoligi *Star*

   Pada topologi *star* terdapat perangkat pengendali yang berfungsi sebagai pengatur dan pengendali komunikasi data. Sedangkan perangkat lain terhubung dengan perangkat pengendali sehingga pengiriman data akan melalui perangkat pengendali.

   Keuntungan topologi *star* adalah :

   1. Mudah diandalkan.
   2. Mudah dikembangkan.
   3. Keamanan data tinggi.
   4. Kemudahan akses ke jaringan LAN lain

   Kerugian topologi *star* adalah :

   1. Jika trafik padat maka dapat menyebabkan lambatnya jaringan.
   2. Jaringan sangat bergantung pada perangkat pengendali.

2.2.7 Pemodelan Pembangunan Sistem

2.2.7.1 *Database Management System* (DBMS)

Sistem manajemen database (*Database Management System* / DBMS) adalah satu rangkaian program-program yang mengelolas sebuah *database* dan menyediakan mekanisme-mekanisme melalui mana jenis-jenis data dapat disimpan, dicari kembali dan diubah [1]. Terdapat berbagai macam *database management system* yang ada saat ini, dengan beberapa di antaranya didesain untuk memberikan kontrol yang tepat pada database yang digunakan untuk tujuan tertentu. Terdapat empat elemen penting yang ditemukan pada setiap DBMS yang dapat Anda jumpai saat ini. Yang pertama adalah implementasi dari sebuah *modelling language* yang berfungsi untuk mendefiniskan bahasa dari setiap *database* yang dihosting melalui DBMS.

Pada dasarnya, *modelling language* memastikan kemampuan database untuk berkomunikasi dengan DBMS, untuk dapat beroperasi pada sistem. Kedua, struktur data (*data structure*) yang juga diatur oleh DBMS. *Data structure* merupakan apa yang memungkinkan DBMS untuk berinteraksi dengan data tanpa perlu menyebabkan kerusakan integritas data itu sendiri. Komponen ketiga adalah

data *query language*. Data *query language* bekerja dengan data struktur untuk memastikan tidak adanya data yang *irrelevan*t (tidak relevan) masuk ke dalam database. Dan elemen terakhir adalah sebuah mekanisme yang memungkinkan melakukan transaksi merupakan sebuah standar utama untuk setiap DBMS. Hal ini membantu memungkinkan lebih dari satu akses bersamaan untuk dapat mengakses *database* oleh lebih dari satu pengguna, mencegah manipulasi dari satu record yang digunakan oleh dua pengguna dalam waktu yang sama, dan mencegah terjadinya duplikasi *record*.

2.2.7.2 *Object Oriented Programming*

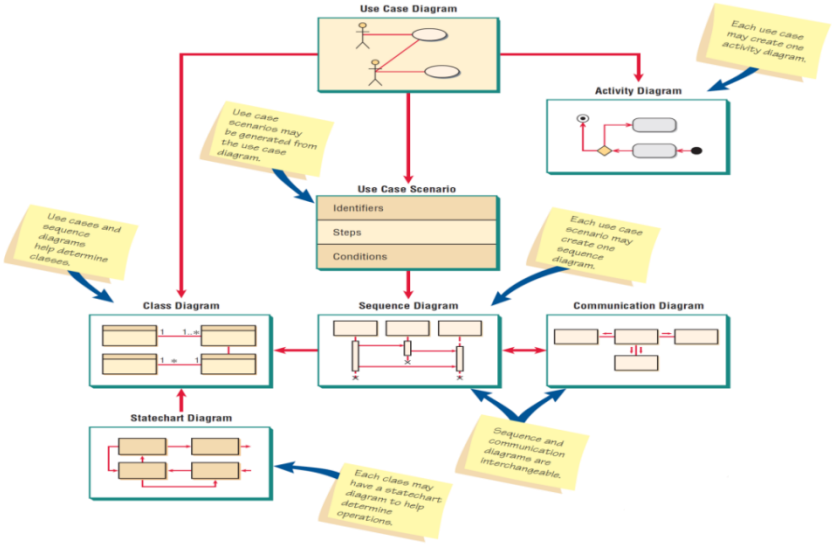
Pemrograman berorientasi objek atau *object oriented programming* adalah suatu cara mengordinasi program dengan memodelkan objek-objek dunia nyata (seperti benda, sifat, sistem, dan yang lainnya) kedalam sebuah bahasa pemrograman dalam bahasa pemrograman pemodelan OOP ini diaplikasikan dengan sebuah kelas, *method*, dan *property/variable* [2]. Dalam pemograman berorientasi objek dikenal ada beberapa pengertian dasar yang harus kita ketahui yaitu:

1. *Class*, dimana *class* adalah merupakan penggambaran dari sebuah objek atau benda,sifat objek, dan juga apa yang bisa dilakukan oleh objek tersebut.
2. *Object*, adalah merupakan suatu bentuk dari *class*.
3. *Inheritance***,** yaitu salah satu bentuk teknis untuk membuat sebuah class lebih spesifik lagi.
4. *Encapsulation*, yaitu mengemas suatu objek sehingga pada saat kita akan memakai objek tersebut kita tidak perlu lagi tahu tentang detail bagaimana suatu *action* itu terjadi.
5. *Polymorphism*, yaitu suatu *object* bisa bertindak lain terhadap *message*/*method* yang sama.

2.2.7.3 *Unified Modeling Language*

UML (*Unified Modeling Language*) adalah himpunan struktur dan teknik untuk pemodelan dan desain program berorientasi objek serta aplikasinya. UML adalah metode untuk mengembangkan sitem OOP dan sekelompok *tool* untuk mendukung pengembangan sistem tersebut [6]. 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*.

Diagram-diagram yang ada dalam UML dapat dikelompokan menjadi 2 kelompok besar, yaitu diagram yang menggambarkan model struktur statis dari sistem (model statis) dan digram yang menggambarkan struktur dinamis dari sistem (model dinamis).



Gambar 2.2 Relasi antar Diagram UML

1. *Use Case Diagram*

   Use case adalah abstraksi dari interaksi antara sistem dan *actor*. *Use case* bekerja dengan cara mendeskripsikan tipe interaksi antara *user* sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. *Use case* merupakan konstruksi untuk mendeskripsikan bagaimana sistem akan terlihat di mata *user*. Sedangkan *use case* diagram memfasilitasi komunikasi diantara analis dan pengguna serta antara analis dan *client*.

2. *Use Case Scenario*

   Use case scenario merupakan hasil instansiasi dari setiap *use case*.

3. *Activity Diagram*

   Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi.

4. *Class Diagram*

   Class adalah dekripsi kelompok obyek-obyek dengan *property*, perilaku (operasi) dan relasi yang sama. Sehingga dengan adanya *class diagram* dapat memberikan pandangan global atas sebuah system. Hal tersebut tercermin dari *class* yang ada dan relasinya satu dengan yang lainnya. Sebuah sistem biasanya mempunyai beberapa *class diagram*. *Class* diagram sangat membantu dalam visualisasi struktur kelas dari suatu sistem.

5. *Statechart 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. *State class* tidak digambarkan untuk semua *class*, hanya yang mempunyai sejumlah *state* yang terdefinisi dengan baik dan kondisi *class* berubah oleh st*ate* yang berbeda.

6. *Sequence Diagram*

   Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara *object* juga interaksi antara *object*, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

2.2.8 Java

Java merupakan bahasa pemrograman sangat andal [2], java dapat membuat seluruh bentuk aplikasi, *desktop*, web dan lainnya, sebagaimana dibuat dengan menggunakan bahasa pemrograman konvensional yang lain. Java merupakan

bahasa pemrograman berorientasi objek (OOP) dan dapat dijalankan pada berbagai platform sistem operasi. Perkembangan java tidak hanya terfokus pada satu sistem operasi, tetapi dikembangkan untuk berbagai sistem operasi dan bersifat open source.java sudah memiliki *library database* yang disebut JDBC.

Kelebihan dari Java adalah :

1. *Multiplatform*, Java dapat dijalankan dalam beberapa platform komputer dan sistem operasi yang berbeda.
2. OOP atau *Object Oriented Programming*.
3. Java memiliki *library* yang lengkap. *Library* disini adalah sebuah kumpulan dari program yang disertakan dalam Java. Hal ini akan memudahkan pemrograman menjadi lebih mudah.

Kekurangan dari Java adalah :

1. penggunaan memori yang cukup banyak.
2. Mudah didekompilasi. Dekompilasi adalah proses membalikkan dari kode jadi menjadi kode sumber.

2.2.9 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*Database Management System*) atau DBMS yang *multithread*, *multi-user* [5]. MySQL merupakan perangkat lunak basis data *open source* yang paling digemari, karena perangkat lunak ini merupakan perangkat lunak basis data yang *powerful* dan cukup stabil untuk digunakan sebagai media penyimpanan data.

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. *Multiuser*. MySQL dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
3. *Performance tuning*. MySQL memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

4. Jenis Kolom. MySQL memiliki tipe kolom yang sangat kompleks, seperti *signed / unsigned integer, float, double, char, text, date, timestamp*, dan lain-lain.
5. Perintah dan Fungsi. MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *select* dan *where* dalam perintah (*query*).
6. Keamanan. MySQL memiliki beberapa lapisan sekuritas seperti *level subnetmask*, nama *host*, dan izin akses *user* dengan sistem perizinan yang mendetail serta sandi terenkripsi.
7. 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.
8. Konektivitas. MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, *Unix soket* (UNIX), atau *Named Pipes* (NT).
9. Lokalisasi. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.
10. Antar Muka. MySQL memiliki *interface* (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).
11. Klien dan Peralatan. MySQL dilengkapi dengan berbagai peralatan (*tool*) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk *online*.
12. 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 IDE merupakan salah satu IDE yang dikembangkan dengan bahasa pemrograman java[2]. Netbeans juga dapat mendukung bahasa pemrograman lain (JavaScript , PHP , Python , Ruby , Groovy , C , C + + , Scala , Clojure , dan lain-lain) NetBeans juga mengacu pada kedua *platform* kerangka untuk aplikasi *desktop* Java, dan sebuah lingkungan pengembangan terpadu untuk pengembangan dengan Java.

NetBeans IDE dapat digunakan jika JVM diinstal, termasuk Windows, Mac OS, Linux, dan Solaris. Sebuah JDK diperlukan untuk pengembangan fungsionalitas java, tetapi tidak diperlukan untuk pembangunan di bahasa pemrograman lain.

Berikut ini keuntungan dari Netbeans :

1. Gratis, dengan ribuan *plug in* yang bisa kita *download* langsung di *website* resminya, maupun dari pihak ketiga
2. Kompetebel dengan *swing* karena memang langsung dikembangkan oleh Sun yang notabenenya sebagai pengembang *swing*. *Swing* merupakan sebuah teknologi Java untuk pengembangan aplikasi *desktop* yang dapat berjalan pada berbagai macam platform seperti windows, linux, Mac OS dan Solaris.
3. Netbeans tidak hanya dapat digunakan buat java saja, karena Netbeans dapat di gunakan untuk bahasa pemograman lain seperti C/C++, Ruby, dan PHP.
4. NetBeans sangat cocok untuk digunakan dalam pengembangan sistem berskala *enterprise*.
5. Pada paket tertentu, Netbeans juga menyertakan GlassFish V2 UR2 dan Apache Tomcat 6.0.16.

Adapun kekurangan dari Netbeans adalah sebagai berikut :

1. NetBeans hanya mensupport 1 pengembangan Java GUI, yaitu *swing*, yang padahal ada Java GUI yang dikembangkan oleh eclipse yang bernama SWT dan JFace yang sudah cukup populer.

2. NetBeans mempatenkan *source* untuk Java GUI yang sedang dikerjakan dalam sebuah *Generated Code*, sehingga programmer tak dapat mengeditnya secara manual.
3. Dari segi sumber daya, Netbeans memerlukan sumber daya yang besar, seperti *memory* dan ruang *hard disk*.
4. Netbeans memerlukan dukungan prosesor yang cukup handal untuk mendapatkan performa maksimalnya.