BAB II

TINJAUAN PUSTAKA

2.1 Sejarah Perusahaan

Giancanibbio Scooteroom Bandung didirikan oleh Berry Joe Agatha pada tahun 2005 bersama sang istri tercinta. Asal nama Giancanibio Scooteroom diambil dari putri beliau yang bernama Meidi Fikih Giancanibbio dengan ditambah kata Scooter dibelakangnya, alhasil terbentuklah nama Giancanibbio Scooteroom.



Gambar 2.1 Logo Giancanibbio Scooteroom

Produk yang dijual bermacam-macam *accessories* dan *sparepart* khusus untuk kendaraan *scooter* yang biasa disebut dengan Vespa. *Work and Shop* Giancanibbio Scooteroom ini berada di Jalan Pasir impun No.05 Sukamiskin Bandung. Selama kurang lebih 7 tahun berdiri,Giancanibbio Scooteroom masih di kelola langsung oleh Berry Joe Agatha dan Fikih Novianty selaku istri tercinta yang juga dibantu beberapa teman-teman dekat sampai sekarang.

2.1.1 VISI DAN MISI

VISI

Menjalin tali persahabatan dan persaudaraan antar sesama penggemar motor vespa.

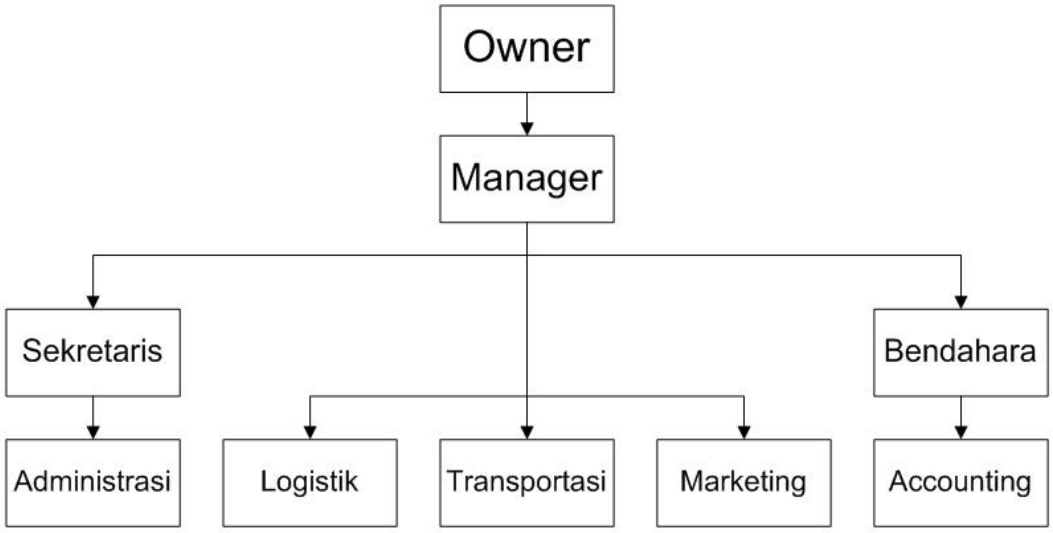
MISI

1. Melestarikan dan mensosialisasikan motor vespa di seluruh Indonesia sebagai sarana transportasi yang klasik dan bersahabat.
2. Memberikan pelayanan terbaik kepada setiap pengguna vespa seluruh tanah air.
3. Meningkatkan mutu dan kualitas produk *accessoriess* dan *sparepart* dalam negeri agar tidak kalah bersaing dengan produksi dari mancanegara.

2.1.2 Struktur Organisasi

Organisasi dalam suatu perusahaan merupakan suatu wadah kerjasama antara tenaga kerja manusia dengan menggunakan alat-alat produksi dan sumber daya yang lainnya untuk mencapai suatu tujuan yang efisien sesuai dengan prinsip-prinsip ekonomis. Untuk mengatur organisasi, maka diperlukan struktur organisasi yang dibuat sedemikian rupa sehingga dapat menjamin kelancaran dalam melaksanakan kegiatan dan untuk mencapai tujuan organisasi.

Struktur organisasi merupakan kerangka yang menghubungkan fungsi-fungsi dalam suatu organisasi dalam suatu usaha dan menetapkan hubungan diantara anggota yang melaksanakan fungsi masing-masing.



Gambar 2.2 Struktur Organisasi

Berikut ini adalah penjelasan tugas yang ada di Giancanibbio Scooteroom Bandung :

1. *Owner*

   Owner adalah yang mendirikan sekaligus pemilik perusahaan, dimana semua kebijakan dan pengawasan jalannya usaha berada di tangan *owner*. Memiliki tugas :

   a. Mengatur, mengawasi, dan mengkoordinasi setiap kegiatan yang akan dilakukan perusahaan agar setiap aktivitas perusahaan dapat berjalan dengan baik.
   b. Mengevaluasi laporan baik mengenai penjualan produk ataupun pengadaan produk.

2. *Manager*

   Manager bertanggung jawab atas pengadaan produk, baik yang masuk maupun yang keluar yang sekaligus membuat laporan mengenai pengadaan produk kepada *owner.*

3. *Sekretaris*

   Sekretaris merupakan asisten atau tangan kanan manager dalam mengatur aktifitas perusahaan, mulai dari administratif sampai *human relations*. Selain itu sekretaris juga sebagai perantara bagi manager dan pihak-pihak yang ingin berhubungan dengan manager.

4. *Bendahara*

   Bendahara mempunyai tugas sebagai berikut :

   1. Bertanggung jawab atas administrasi keuangan yang ada di perusahaan.
   2. Membuat laporan keuangan perusahaan secara berkala sehingga dapat diperiksa jika diperlukan.
   3. Menandatangani bukti-bukti pengeluaran dan penerimaan uang.
   4. Menerima dan menyimpan uang milik perusahaan.
   5. Mengeluarkan uang sesuai dengan keperluan berdasarkan persetujuan dari manager.

5. *Administrasi*

   Fungi umum dari administrasi adalah membantu sekretaris dalam merencanakan dan melaksanakan kegiatan sesuai dengan visi, misi, dan tujuan perusahaan, yang mempunyai tugas sebagai berikut :

   1. Mengagendakan surat masuk dan surat keluar.

2. Membuat arsip surat masuk dan surat keluar.

6. *Accounting*

   Accounting membantu bendahara dalam mengatur administrasi perusahaan, antara lain :

   1. Menyusun laporan keuangan.
   2. Menyusun anggaran pengeluaran maupun pemasukan.
   3. Membayar gaji pegawai.

7. *Logistik*

   Logistik merupakan bagian yang bertugas dalam pengadaan barang, bagian produksi, supplay dan jasa perusahaan. Tugas lain dari Logistik antara lain :

   1. Mengelola dan mengirimkan barang yang dipesan oleh pelanggan.
   2. Membeli barang dari luar.
   3. Mengecek persedian barang dalam perusahaan.

8. *Transportasi*

   Fungsi transportasi secara umum adalah membantu staf lain dalam pengiriman barang. Contohnya antara lain :

   1. Mengirimkan barang yang telah dipesan oleh pembeli.
   2. Mengambil barang yang di beli oleh perusahaan.
   3. Menyediakan transportasi untuk perusahaan.
   4. Merawat alat transportasi yang ada di perusahaan.

9. Marketing

Marketing bertugas sebagai berikut :

1. Merencanakan, mengorganisir, dan mengendalikan seluruh aktifitas dari penjualan perusahaan.
2. Merencanakan kegiatan perusahaan dalam jangka pendek dan jangka panjang.
3. Menerima pesanan penjualan atau purchase order dari perusahaan lain.
4. Membuat laporan distribusi untuk ditijukan kepada manager.
5. Mengarsip semua dokumen-dokumen penjualan secara sistematis.

2.2 LANDASAN TEORI

Berikut akan dijelaskan beberapa landasan teori yang digunakan dalam membangun program aplikasi.

2.2.1 Java Programming Language

Java Programming Language adalah bahasa pemrograman yang berorientasi objek (OOP) dan dapat dijalankan pada berbagai *platform* system operasi. Perkembangan Java tidak hanya terfokus pada satu sistem operasi, tetapi dikembangkan untuk berbagai sistem operasi dan bersifat *open source.*

Ada banyak kelebihan yang ditawarkan oleh Java itu sendiri, selain dari sifatnya yang multi *platform* (artinya dapat berjalan di berbagai macam sistem operasi), java juga sangat mudah dalam memaintenance program yang telah dibuat, karena bertumpu pada pemrograman berorientasi objek. Sampai saat ini

ada banyak edisi java yang sudah mengalami modifikasi. Namun, dari segi kategori nya java terbagi ke dalam tiga kelompok besar, yakni :

a. Java 2 Standard Edition (J2SE), lebih terkonsentrasi pada PC.
b. Java 2 Enterprise Edition (J2EE) untuk konsentrasi pada aplikasi server besar.
c. Java 2 Micro Edition (J2ME) untuk konsentrasi pada mobile.

2.2.2 IDE Netbeans 6.9

IDE Netbeans adalah IDE open source yang ditulis sepenuhnya dengan bahasa pemrograman Java menggunakan platform Netbeans. IDE Netbeans mendukung pengembangan semua tipe aplikasi Java (J@SE, web, EJB, dan aplikasi mobile). Fitur lainnya adalah sistem proyek berbasis Ant, Kontrolversi, dan refactoring. IDE Netbeans versi 6.9 adalah mengembangkan fitur - fitur Java EE yang sudah ada (termasuk Java Persistance support, EJB-3 dan JAX-WS).

Platform Netbeans memungkinkan aplikasi dibangun dari sekumpulan komponen perangkat lunak moduler yang disebut (modul). Sebuah modul adalah suatu arsip Java (Java Archive) 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 platform Netbeans dapat dengan mudah dikembangkan oleh pihak ketiga secara mudah dan powerfull.

2.2.3 AppServ

Appserv merupakan aplikasi yang berfungsi untuk install beberapa program antara lain Apache, PHP, MySQL dalam waktu yang singkat. Dengan adanya AppServ, mempermudahkan orang untuk membuat web server dan database. AppServ ini mempunyai beberapa versi antara lain, AppServ versi 2.5.9, 2.5.10, 2.6.0 (saat ini).

Tujuan aplikasi AppServ antara lain adalah mempermudah dalam membuat database dan membuat web server. Dalam membuat database biasanya yang digunakan adalah MySql dan dalam pembuatan skrip menggunakan pHp. Antara pHp dan MySql saling berhubungan disebabkan skrip pada pHp akan muncul pada halaman web dan dimasukan dalam MySql (database). Jadi awal pembuatan web, dilakukan menggunakan skrip html (untuk penampilan) dan skrip pHp untuk proses dari skrip html. Apabila kita mengetikan atau akan memasukan data kedalam database yang terhubung dengan web, pada halaman web browser kita masukan yang akan diinput dan kemudian akan diproses oleh pHp dan akan dimasukan kedalam database.

2.2.4 Database MySQL

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 komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu

SQL (Structured Query Language). 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. *Open Source*.

   MySQL didistribusikan secara open source, dibawah lisensi GPL sehingga dapat digunakan secara cuma-cuma.

3. *Multiuser*.

   MySQL dapat digunakan oleh beberapa user 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. Jenis Kolom.

   MySQL memiliki tipe kolom yang sangat kompleks, seperti signed atau 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 sekuritas 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 interface (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming Interface).

12. 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.

13. Struktur tabel.

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

2.2.5 PhpMyAdmin

PhpMyAdmin adalah perangkat lunak bebas yang ditulis dalam bahasa pemrograman Php yang digunakan untuk menangani administrasi MySql melalui Word Wide Web (WWW). PhpMyAdmin mendukung operasi MySql diantaranya mengelola basis data, tabel-tabel, bidang, relations, index, pengguna atau user, perijinan dan lain-lain.

Pada dasarnya mengelola basis data pada MySql harus dilakukan dengan cara mengetikan baris-baris yang sesuai (command line) untuk setiap maksud tertentu. Jika seseorang ingin membuat basis data atau database ketikan baris perintah yang sesuai untuk membuat basis data.

Fitur dalam PhpMyAdmin antara lain :

1. Antar muka berbasis web, yaitu :
    a. Menelusuri dan drop basis data (database), tabel, pandangan (view), bidang (fields) dan index.

b. Membuat, menyalin, drop, dan mengubah nama basis data, tabel, kolom dan index.
c. Pemeliharaan server, basis data dan tabel, dengan server konfigurasi.
d. Melaksanakan, mengedit dan penunjuk pernyataan sql, bahkan batch-queries.
e. Mengelola pengguna MySql dan hak istimewa.
f. Mengelola prosedur penyimpanan.

2. Import data dari CSV dan SQL.
3. Export data ke berbagai format.
4. Membuat grafik PDF dari tampilan basis data.
5. Membuat kompleks query dengan menggunakan query by example (QBE)
6. Pencarian global dalam basis data.
7. Transformasi data disimpan dalam format yang menggunakan satu set fungsi yang telah ditetapkan, seperti menampilkan data blog-data atau download-link.

2.3 UML*(Unified Modeling Language)*

UML (*Unified Modeling Language*) adalah bahasa standar yang digunakan untuk menentukan, menggambarkan, membangun dan mendokumentasikan sistem dari sebuah software. UML mewakili suatu kumpulan pengerjaan teknik yang baik dan telah dibuktikan dalam melakukan pemodelan terhadap sistem yang

besar dan kompleks. UML menggunakan notasi-notasi secara graphis untuk mengekspresikan desain dari sebuah *software.*

2.3.1 Tujuan Mendesain dengan UML

Tujuan pengembang mendesain dengan menggunakan UML adalah :

1. UML memberikan kemudahan terhadap user karena memakai model visual. Oleh karena itu user bisa mengembangkan dan melakukan perubahan yang berarti terhadap model yang sudah ada.
2. UML memisahkan antara bahasa pemrograman dan proses pengembangan.
3. UML memberikan dasar yang formal untuk pengertian model bahasa
4. UML mendukung perkembangan tools market OO.

2.3.2 Macam - Macam Diagram UML

Penjelasan dari masing-masing diagram tersebut akan dijelaskan dalam bentuk sub bab berikut ini :

2.3.2.1 Use Case Diagram

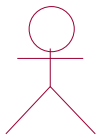
Use Case Diagram adalah gambar dari beberapa atau seluruh aktor dan *Use Case* dengan tujuan mengenali interaksi mereka dalam suatu sistem yang dapat membantu menganalisa kebutuhan suatu sistem.

Berikut adalah penjelasan dari istilah-istilah yang terdapat dalam *Use Case*:

a. Aktor

Aktor mewakili siapa pun atau apa saja yang harus berinteraksi dengan sistem.Aktor bisa didefinisikan sebagai berikut :

1. Aktor hanya memberikan informasi kepada sistem.
2. Aktor hanya menerima informasi dari sistem.
3. Aktor memberikan dan menerima informasi ke dan dari sistem.

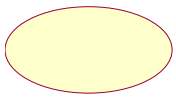


Aktor

Gambar 2.3 Notasi Aktor

b. *Use Case* Model

Use Case Model adalah dialog antara aktor dengan sistem yang akan menggambarkan fungsi yang diberikan oleh sistem.



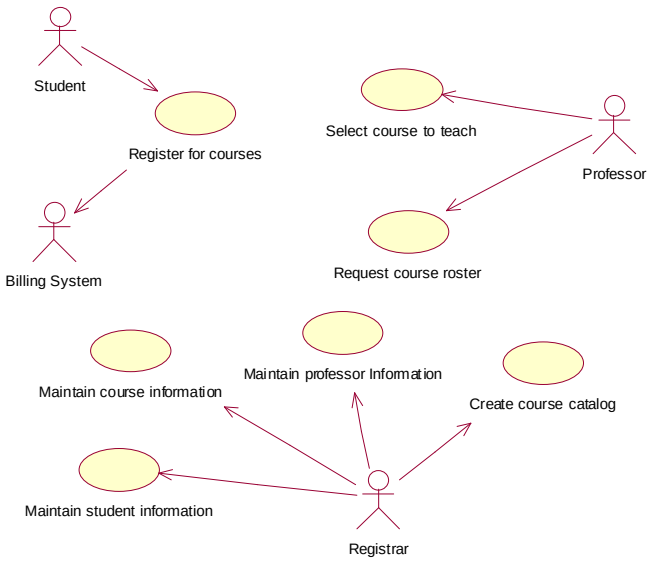
UseCase

Gambar 2.4 Notasi *Use Case*

c. *Use Case* Relationship

Use Case Relationship adalah suatu hubungan baik itu antara aktor dan *Use Case* atau antara *Use Case* dan *Use Case*. Hubungan antara aktor dan *Use Case* disebut dengan *communicate association*. Berikut adalah contoh gambar *Use Case*

diagram dengan communicate association, dimana *Use Case* ini memperlihatkan dialog antara aktor dan sistem, sebagai contoh : registrasi mata pelajaran di ESU



Gambar 2.5 *Use Case* Diagram Registrasi Mata Kuliah

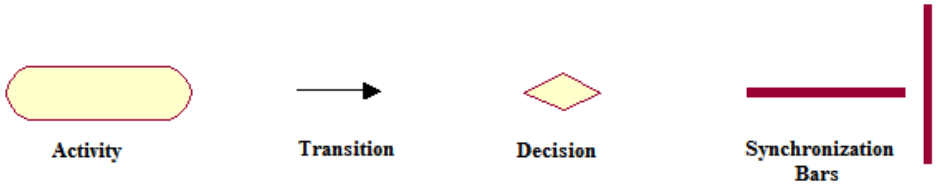
2.3.2.2 Activity Diagram

Activity Diagram berupa *flow chart* yang digunakan untuk memperlihatkan aliran kerja dari sistem. Notasi yang digunakan dalam *Activity Diagram* adalah sebagai berikut:

1. Activity adalah notasi yang menggambarkan pelaksanaan dari beberapa proses dalam aliran pekerjaan.
2. *Transition* adalah notasi yang digunakan untuk memperlihatkan jalan aliran kontrol dari *activity* ke *activity*.

3. *Decision* adalah notasi yang menandakan kontrol cabang aliran berdasarkan decision point.
4. *Synchronization Bars* adalah aliran kerja notasi ini menandakan bahwa beberapa aktivitas dapat diselesaikan secara bersamaan (pararel).

Berikut adalah gambar dari masing-masing notasi yang disebutkan diatas:



Gambar 2.6 Notasi UML Untuk *Activity Diagram*

2.3.2.3 Kelas Diagram

Kelas Diagram berfungsi untuk menjelaskan tipe dari *object* sistem dan hubungannya dengan *object* yang lain. *Object* adalah nilai tertentu dari setiap *attribute* kelas entity. Pada penggambaran kelas diagram ada dikenal dengan kelas analisis yaitu kelas ber-*stereotype*.

Jenis kelas ber-*stereotype* adalah sebagai berikut:

1. Kelas *Boundary*

Kelas *Boundary* adalah kelas yang memodelkan interaksi antara satu aktor atau lebih dengan sistem. Kelas *Boundary* dapat berupa:

a. User *interface* yang merupakan sarana komunikasi antara sistem dengan user. Misalkan window dalam GUI.

b. Sistem *interface* yang merupakan sarana komunikasi antara sistem dengan informasi lainnya misalkan *communication protocol.*

c. Device *interface* yang merupakan sarana komunikasi antara sistem dengan device seperti printer, sensor dan lain sebagainya.

2. Kelas Control

Digunakan untuk memodelkan perilaku mengatur khusus untuk satu atau beberapa *Use Case* saja. Kelas ini biasanya digunakan untuk membuat isi kelas *entity* dan biasanya memasangkan kelas *Boundary* dengan kelas entity.

3. Kelas Entity

Memodelkan informasi yang harus disimpan oleh sistem. Kelas *entity* memperlihatkan struktur data dari sebuah sistem.