

BAB II

TINJAUAN PUSTAKA

II.1. Tinjauan Perusahaan

Tinjauan perusahaan yang dilakukan adalah untuk mengetahui sejarah perusahaan, visi dan misi perusahaan, struktur organisasi perusahaan beserta tugas dan tanggung jawab perusahaan.

II.1.1. Sejarah Singkat Perusahaan

Dengan semakin pesatnya perkembangan dunia otomotif dan semakin meningkatnya pengetahuan masyarakat pada dunia otomotif, hal ini mendasari berdirinya Bengkel SEHAT yang pertama pada tahun 1989, tepatnya di Jln. Hasanudin (SEHAT Auto Galery), dengan fasilitas cuci mobil dan ganti oli. Kemudian pada tahun 1993, Bengkel SEHAT menambah 2 outlet lagi yaitu di Jln. Gatot subroto (SEHAT Pusat Perawatan Mobil) dan di Jln. SM Raja (SEHAT Motor). Bersama itu pula Bengkel SEHAT menambah fasilitas layanan bengkel untuk Engine Repair, Sporing, Balancing, dan jasa bengkel lainnya.

Pada tahun 1994, Bengkel SEHAT menambah satu outlet lagi Bengkel SEHAT di Jln. H. Adam Malik (SEHAT Perawatan dan Perbaikan Mobil) dengan tambahan fasilitas layanan OtoShop (aksesoris mobil). Sejak itu pula semua Bengkel SEHAT meningkatkan pelayanannya dengan melengkapi fasilitas layanan bengkel di semua outlet yang ada. Menjawab permintaan pasar maka Bengkel SEHAT mencoba untuk menambah fasilitas yang ada dengan Layanan

Body Repair (SEHAT Jln. Gatot Subroto) yaitu pada tahun 2001.

Dengan motto selalu memberikan layanan yang terbaik dan terpercaya dan dengan kredibilitas yang sudah tidak di ragukan lagi di Wilayah Sumatera Utara, maka Bengkel SEHAT membuka satu outlet lagi di Jln S. Parman tahun 2003. Menimbang perkembangan permintaan terhadap layanan Body repair maka pada tahun 2006, Bengkel Sehat Jl. Adam Malik menambah layanan Body Repair dengan nama divisi Micro Auto Body Repair I, setahun kemudian, pada tahun 2007 di buka Micro Auto Body Repair II di Jl. SM Raja. Kemudian di akhir tahun 2007 di buka Micro Auto Body Repair III di Jl. Gatot Subroto No. 80.

II.1.2. Visi dan Misi Perusahaan

Sebagai salah satu perusahaan yang bergerak di bidang pencucian mobil dan service mobil, Salon mobil SEHAT berkomitmen untuk memberikan layanan dan solusi terbaik bagi pelanggan. Komitmen itu tercermin dalam visi dan misi perusahaan untuk menjadi salon mobil terbaik di Kota Medan. Adapun visi dan misi itu adalah sebagai berikut :

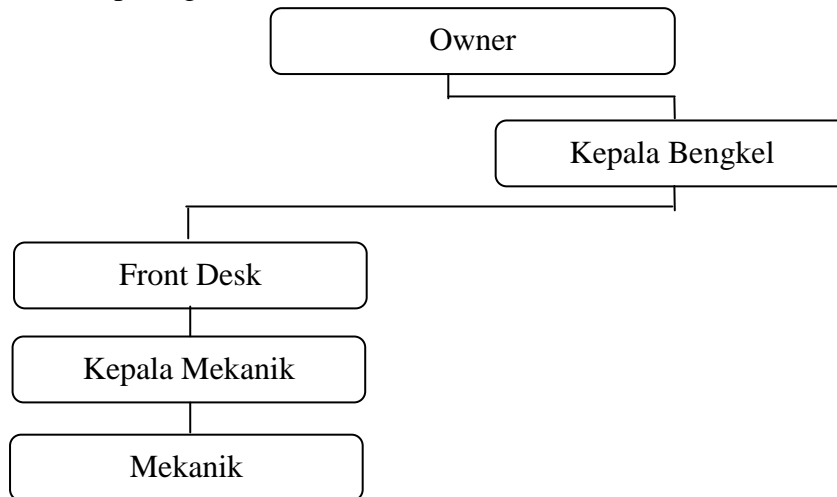
1. **Visi** : Untuk menjadi bengkel mobil terbaik di Kota Medan yang mengutamakan pada kepuasan pelanggan didukung dengan peralatan canggih dan tenaga ahli yang berkompeten dalam memberikan pelayanan jasa pencucian mobil dan service mobil kepada pelanggan, dan masyarakat.
2. **Misi** : Memberikan jasa yang unggul berdasarkan nilai-nilai kejujuran, keadilan dan kehati-hatian, memberikan layanan mobil sehat yang cepat

dan berkualitas serta memberikan solusi tepat dan cepat mulai dari saat pelanggan memperbaiki mobil, proses menunggu sampai dengan mobil selesai diperbaiki.

II.1.3. Struktur Organisasi Perusahaan

struktur organisasi adalah suatu gambar yang menggambarkan tipe organisasi, pendepartemenan organisasi kedudukan dan jenis wewenang pejabat, bidang dan hubungan pekerjaan, garis perintah dan tanggung jawab, rentang kendali dan system pimpinan organisasi. Unsur-unsur organisasi terdiri dari manusia, tempat kedudukan, tujuan, pekerjaan, struktur, teknologi, dan lingkungan.

Berikut ini adalah gambar dari struktur organisasi Salon Mobil SEHAT dapat dilihat pada gambar II.1.



Gambar II.1. Struktur Organisasi Salon Mobil SEHAT

(Sumber : Salon Mobil SEHAT)

II.1.4. Tugas dan Tanggung Jawab

1. Owner, Pemilik sekaligus pengurus dan pengawas segala yang terjadi di bengkel Salon Mobil SEHAT.
2. Kepala Bengkel, bertanggung jawab terhadap keseluruhan yang ada di bengkel atau dibagian operasional bengkel, seperti menerima laporan harian yang diberikan oleh FD (Front Desk) atau kasir
3. FD (Front Desk), Bertugas mencatat secara administrasi kegiatan yang ada di bengkel dan merangkap juga sebagai kasir dan membuat laporan keuangan setiap transaksi yang terjadi didalam bengkel Salon Mobil SEHAT setiap harinya yang kemudian dilaporkan kepada kepala bengkel
4. KM (Kepala Mekanik), Bertugas mengecek hasil kerja mekanik serta mengawasi kegiatan mekanik dan memberikan arahan terhadap mekanik bila mengalami kesulitan dalam pekerjaannya
5. Mekanik, Bertugas memperbaiki mobil yang dimiliki konsumen yang selanjutnya selesai pengerjaan tersebut di laporkan kepada kepala mekanik

II.2. Landasan Teori

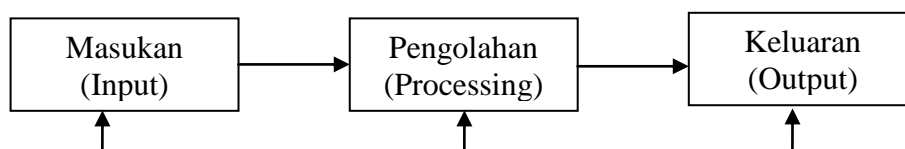
II.2.1. Sistem

Defenisi sistem berkembang sesuai dengan konteks dimana pengertian sistem itu digunakan. Berikut akan diberikan beberapa defenisi sistem secara umum :

1. Kumpulan dari bagian-bagian yang bekerja sama untuk mencapai tujuan yang sama.

Contoh :

- a. Sistem Tatasurya
 - b. Sistem Pencernaan
 - c. Sistem Transportasi Umum
 - d. Sistem Otomotif
 - e. Sistem Komputer
 - f. Sistem Informasi
2. Sekumpulan objek-objek yang saling berelasi dan berinteraksi serta hubungan antar objek bisa dilihat sebagai satu kesatuan yang dirancang untuk mencapai satu tujuan (Hanif Al Fatta; 2007 : 1).
3. Jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu (Kristanto; 2008 : 1).



Gambar II.2: Model Sistem

(Sumber : Hanif Al Fatta ; 2007 : 2)

II.2.2. Karakteristik Sistem

Model umum suatu sistem terdiri dari input, proses, dan output. Adapun karakteristik yang dimaksud adalah sebagai berikut :

1. Komponen sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang bekerjasama membentuk satu kesatuan.

2. Batasan sistem (*Boundary*)

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem lainnya atau sistem dengan lingkungan luarnya.

3. Lingkungan luar sistem (*Environment*)

Bentuk apapun yang diluar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut dengan lingkungan luar sistem.

4. Penghubung sistem (*Interface*)

Media yang menghubungkan sistem dengan subsistem yang lain disebut dengan penghubung sistem atau *interface*

II.2.3. Siklus Hidup Pengembangan Sistem

Merupakan suatu bentuk yang digunakan untuk menggambarkan tahapan utama dan langkah-langkah pada tahapan tersebut dalam proses pengembangan system (Tata Sutabri,2013:56).

Siklus hidup pengembangan sistem dimulain dari fase perencanaan, fase pengembangan (investigasi, analisis, desain, implementasi) dan dievaluasi secara terus menerus untuk menetapkan apakah sistem informasi tersebut masih layak di aplikasikan.

1. Fase Perencanaan

Pengembangan sistem informasi bertujuan untuk mengidentifikasi dan memprioritaskan sistem informasi apa yang akan dikembangkan, sasaran-sasaran yang ingin dicapai, jangka waktu pelaksanaan, serta mempertimbangkan dana tersedia dan siapa yang akan melaksanakan.

2. Fase Pengembangan

Fase pengembangan sistem informasi tersebut juga sebagai siklus hidup pengembangan sistem informasi yang garis besarnya terdiri dari beberapa langkah. Tahapan utama dalam proses pengembangan sistem informasi adalah sebagai berikut :

a. Investigasi Sistem

Manfaat dari fase penyelidikan/ investigasi ini adalah untuk menentukan masalah-masalah atau kebutuhan yang timbul.

b. Analisis Sistem

Tahap analisis bertitik tolak pada kegiatan-kegiatan dan tugas-tugas dimana sistem yang berjalan dipelajari lebih mendalam, konsepsi, dan usulan dibuat untuk menjadi landasan bagi sistem baru yang akan dibangun.

c. Desain Sistem

Pada tahap ini sebagian besar kegiatan yang berorientasi ke computer dilaksanakan.

d. Implementasi Sistem

Tahap ini adalah prosedur yang dilakukan untuk menyelesaikan desain sistem yang ada dalam dokumen desain sistem disetujui dan menguji, menginstal dan memulai penggunaan sistem baru atau sistem yang diperbaiki.

e. Pemeliharaan Sistem

Disarankan adanya dua tahap review yang harus dilakukan, pertama kali tidak terlalu lama setelah penerapan sistem. Review berikutnya dapat dilakukan kira-kira setelah enam bulan berjalan (Tata Sutabri; 2013 : 57).

II.2.4. Informasi

Informasi merupakan proses lebih lanjut dari data yang sudah memiliki nilai tambah, informasi dapat dikelompokkan menjadi 3 bagian, yaitu :

1. Informasi strategis, informasi ini digunakan untuk mengambil keputusan jangka panjang, yang mencakup informasi eksternal, rencana perluasan perusahaan, dan sebagainya.
2. Informasi taktis, informasi ini dibutuhkan untuk mengambil keputusan jangka menengah, seperti informasi tren penjualan yang dapat dimanfaatkan untuk menyusun rencana penjualan.
3. Informasi teknis, informasi ini dibutuhkan untuk keperluan operasional sehari-hari, seperti informasi persediaan *stock*, retur penjualan, dan laporan kas harian.

Informasi adalah data yang telah diklasifikasikan atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan. Sistem pengolahan informasi akan mengolah data menjadi informasi atau mengolah data dari bentuk tidak berguna menjadi berguna bagi yang menerimanya. Nilai informasi berhubungan dengan keputusan. Bila tak ada pilihan atau keputusan maka informasi tidak diperlukan. Keputusan dapat berkisar dari keputusan berulang sederhana sampai keputusan strategis jangka panjang. Nilai informasi dilukiskan paling berarti dalam konteks pengambilan keputusan (Tata Sutabri ; 2012 : 21).

II.2.5. Geografis

Menurut Prof.Bintarto Geografis adalah mempelajari hubungan kausal gejala-gajala dimuka bumi dan peristiwa-peristiwa yang terjadi dimuka bumi baik yang fisik maupun yang menyangkut makhluk hidup serta permasalahannya, melalui pendekatan keruangan, ekologis dan regional untuk kepentingan program, proses dan keberhasilan pembangunnya (id.wikipedia).

II.2.6. Sistem Informasi

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan laporan-laporan yang diperlukan oleh pihak luar tertentu (Tata Sutabri ; 2012 : 38).

II.2.7. Sistem Informasi Geografis

Definisi Sistem Informasi Geografis (SIG) selalu berkembang, bertambah, dan bervariasi. SIG juga merupakan suatu bidang kajian ilmu dan teknologi yang relatif baru, digunakan oleh berbagai bidang disiplin ilmu, dan berkembang dengan cepat. SIG adalah sistem komputer yang digunakan untuk memasukkan (*capturing*), menyimpan, memeriksa, mengintegrasikan, memanipulasi, menganalisa, dan menampilkan data-data yang berhubungan dengan posisi-posisi di permukaan bumi. SIG dapat didefinisikan sebagai kombinasi perangkat keras dan perangkat lunak komputer yang memungkinkan untuk mengelola (*manage*), menganalisa, memetakan informasi spasial berikut data atributnya (data deskriptif) dengan akurasi kartografi (Basic, 2000 dalam Eddy Prahasta, 2009). Dari definisi ini dapat diuraikan menjadi beberapa subsistem yaitu data input, dasar output, data manajemen, dan data manipulasi dan analisis.

II.3. Tahap-tahap Membangun Sistem Informasi Geografis (SIG)

1. Pengumpulan Data

Melalui (GPS, analisa citra atau foto satelit, survei lahan atau topografi dengan theodolit, clinometer untuk menghitung kemiringan suatu tanah, lahan atau lokasi) lalu di digitalisasi dengan cara (scanning peta, georeferencing).

2. Pengolahan dan Manajemen data

Menjadikan kumpulan data tersebut sebuah database dalam bentuk (geodatabase, geocoding, input table).

3. Analisis

Dalam SIG, analisis sangat diperlukan. Karena dalam analisis inilah yang akan membentuk sebuah peta baru yang telah diolah baik secara manual ataupun secara komputerisasi.

4. Output

Setelah melakukan analisis, maka hasil SIG itu akan dicetak dalam bentuk peta, diagram, tabel, webMAP.

5. Masukan Data

merupakan proses pemasukan data pada komputer dari peta (peta topografi dan peta tematik), data statistik, data hasil analisis penginderaan jauh data hasil pengolahan citra digital penginderaan jauh, dan lain-lain. Data-data spasial dan atribut baik dalam bentuk analog maupun data digital tersebut dikonversikan ke dalam format yang diminta oleh perangkat lunak sehingga terbentuk basisdata (database). Menurut Anon (2003) basisdata adalah pengorganisasian data yang tidak berlebihan dalam komputer sehingga dapat dilakukan pengembangan, pembaharuan, pemanggilan, dan dapat digunakan secara bersama oleh pengguna.

6. Penyimpanan Data dan Pemanggilan Kembali (data storage dan retrieval)

Penyimpanan data pada komputer dan pemanggilan kembali dengan cepat (penampilan pada layar monitor dan dapat ditampilkan/cetak pada kertas).

7. Pelaporan Data

Dapat menyajikan data dasar, data hasil pengolahan data dari model menjadi bentuk peta atau data tabular. Menurut Barus dan wiradisastra (2000) Bentuk produk suatu SIG dapat bervariasi baik dalam hal kualitas, keakuratan dan kemudahan pemakainya. Hasil ini dapat dibuat dalam bentuk peta-peta, tabel angka-angka: teks di atas kertas atau media lain (hard copy), atau dalam cetak lunak (seperti file elektronik).

II.3.1. MapInfo Profesional

Mapinfo merupakan program yang digunakan dalam pembuatan peta. Peta yang diperoleh sudah dilengkapi dengan koordianat,data base. Selain pembuatan peta Mapinfo juga dapat diaplikasikan dalam pembuatan analisis dibidang geografi, sumber daya lahan, dan pembuatan jalan.

II.4. Macromedia Dreamweaver 8

Macromedia Dreamweaver 8 adalah sebuah perangkat lunak aplikasi untuk mendesain dan membuat halaman *web*. Dengan menggunakan *Dreamweaver 8*, ketika membuat halaman *web*, tidak perlu lagi mengetik kode-kode *HTML* atau kode-kode lainnya secara manual. Cukup melakukan klik beberapa kali, maka halaman web yang di inginkan sudah jadi. *Macromedia Dreamweaver 8* dapat membuat tabel hanya dengan melakukan dua kali klik saja. *Macromedia Dreamweaver 8* akan menciptakan kode-kode *HTML* yang sesuai untuk membuat tabel yang diinginkan. Selain *HTML*, *Macromedia Dreamweaver 8* juga mendukung *CSS*, *JavaScript*, *PHP*,*ASP* dan bahasa pemograman lainnya untuk membuat *web* (Arief Ramadhan, S.Kom ; 2007 : 2).

II.5. PHP (*Hypertext Preprocessor*)

PHP (Hypertext Preprocessor) adalah kode/skrip yang akan dieksekusi pada *server side*. Skrip *PHP* akan membuat suatu aplikasi dapat diintegrasikan ke dalam *HTML*, sehingga suatu halaman *web* tidak lagi bersifat statis, namun menjadi bersifat dinamis. Sifat *server side* berarti pengerjaan skrip dilakukan di *server*, baru kemudian hasilnya dikirimkan ke *browser* (Deni Sutaji : 2012: 2).

II.6. Database

Database merupakan komponen utama dalam membangun sebuah sistem yang menyangkut pendokumentasian data ke dalam sebuah *database*. Bentuk basis data adalah sebuah aturan yang mengatasi masalah tersebut. Saat ini basis data memiliki peranan yang sangat penting dalam mengolah data yang ada di dalamnya. Validasi juga mencakup di dalamnya karena aturan-aturannya dalam sebuah data yang terdokumentasikan juga memiliki sebuah aturan yang dikenal dengan basis data. *Database* adalah sebuah cara mendokumentasikan berbagai macam data yang kemudian dimanajemen dengan sebuah sistem untuk disimpan dalam sebuah media penyimpanan. Dengan demikian data tersebut dapat diakses dengan mudah dan cepat. Media penyimpanan tersebut dapat diibaratkan sebagai sebuah *storage* penyimpanan, misalnya *Hardisk* (Bunafit Nugroho ; 2005: 71).

II.6.1. Hirarki Database

Merupakan data yang dimana data tersebut diatur dengan struktur data *free*. Adapun beberapa hirarki database adalah sebagai berikut :

1. Field

Merupakan kumpulan dari karakter-karakter yang membentuk suatu arti tertentu.

2. Record

Merupakan kumpulan dari field-field yang membentuk sebuah arti.

3. File

Merupakan kumpulan dari record-record (Bunafit Nugroho; 2009:76)

II.6.2. Perancangan Basis Data

Sebelum melakukan perancangan *database* ada beberapa kriteria yang harus dipenuhi untuk memperoleh validasi data. Beberapa hal yang harus dilakukan untuk memenuhi kriteria tersebut adalah sebagai berikut :

1. Studi Kelayakan : langkah ini menjadi langkah pembuka yang harus dilakukan demi mendapatkan informasi yang valid sehingga dapat diterapkan ke dalam sebuah basis data.
2. Rencana Pendahuluan : merupakan langkah menentukan lingkup sistem yang dikerjakan. Hal ini berguna untuk melakukan pendefinisian waktu yang dibutuhkan dalam proses pembuatan sistem.
3. Menganalisa Sistem : setelah perencanaan selesai, langkah selanjutnya adalah melakukan penganalisaan data.
4. Merancang Sistem : melakukan perancangan sistem.

5. Implementasi : setelah perancangan sistem dilakukan dan muncul hasilnya, maka dapat melanjutkan proses analisis dengan mengimplementasikannya (Bunafit Nugroho ; 2005 : 81).

II.7. MySQL

MySQL adalah sebuah program *database server* yang mampu menerima dan mengirimkan datanya dengan sangat cepat, *multi user* serta menggunakan perintah standar *SQL (Structured Query Language)*. *MySQL* merupakan sebuah *database server* yang *free*, artinya bebas menggunakan *database* ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensinya. *MySQL* adalah sebuah program *database server* yang mampu menerima dan mengirimkan datanya dengan sangat cepat, *multi user*, serta menggunakan standart *SQL* (Bunafit Nugroho ; 2005: 1).

II.7.1. Kelebihan MySQL

MySQL adalah sebuah *database server*, dapat juga berperan sebagai *client /server*. *Database* ini memiliki kelebihan dibanding *database* lain, di antaranya adalah :

1. *MySQL* sebuah *Database Management System (DBMS)*.
2. *MySQL* sebagai *Relation Management System (RDBMS)*.
3. *MySQL* adalah sebuah *software database* yang *OpenSource*, artinya program ini bersifat *free* atau bebas digunakan oleh siapa saja.
4. *MySQL* mampu menerima *query* yang bertumpuk dalam satu permintaan.

5. *MySQL* merupakan sebuah database yang mampu menyimpan data berkapasitas sangat besar hingga berukuran *gigabyte*.
6. *MySQL* dapat menciptakan lebih dari 16 kunci per tabel, dan dalam satu kunci memungkinkan berisi belasan kolom (Bunafit Nugroho ; 2005: 3).

II.8. Teknik Normalisasi

Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basis data relasional. Pada dasarnya, normalisasi adalah proses dua langkah yang meletakkan data dalam bentuk tabulasi dengan menghilangkan kelompok berulang lalu menghilangkan data yang terduplikasi dari tabel rasional.

Teori normalisasi didasarkan pada konsep bentuk normal. Sebuah tabel relasional dikatakan berada pada bentuk normal tertentu jika tabel memenuhi himpunan batasan tertentu. Ada lima bentuk normal yang telah ditemukan.

II.8.1. Bentuk-bentuk Normalisasi

1. Bentuk tidak Normal

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti format tertentu, dapat saja tidak lengkap dan terduplikasi. Data dikumpulkan apa adanya sesuai keadaannya.

2. Bentuk normal tahap pertama (1st Normal Form)

Sebuah tabel disebut 1NF jika : tidak ada baris yang diduplikat dalam tabel tersebut, masing-masing *cell* bernilai tunggal.

3. Bentuk normal tahap kedua (2nd normal form)

Bentuk Normal Kedua (2NF) terpenuhi jika pada sebuah tabel semua atribut yang tidak termasuk dalam *primary key* memiliki ketergantungan fungsional pada *primary key* secara utuh. Sebuah tabel dikatakan tidak memenuhi 2NF jika ketergantungannya hanya bersifat parsial (hanya tergantung pada sebagian dari *primary key*).

4. Bentuk normal tahap ketiga (3rd normal form)

Sebuah tabel dikatakan memenuhi bentuk normal ketiga (3NF), jika untuk setiap ketergantungan fungsional dengan notasi $X < A$, dimana A mewakili semua atribut tunggal di dalam tabel yang tidak ada di dalam X, maka: X haruslah *superkey* pada tabel tersebut, atau A merupakan bagian dari *primary key* pada tabel tersebut.

5. Bentuk Normal Tahap Keempat dan Kelima

Penerapan aturan normalisasi sampai bentuk normal ketiga sudah memadai untuk menghasilkan tabel berkualitas baik. Namun demikian, terdapat pula bentuk normal keempat (4NF) dan kelima (5NF). Bentuk Normal Keempat berkaitan dengan sifat ketergantungan banyak nilai pada suatu tabel yang merupakan pengembangan dari ketergantungan fungsional. Adapun bentuk normal tahap kelima merupakan nama lain dari *Project Join Normal Form* (PJNF).

6. Boyce Code Normal Form (BCNF)

Memenuhi 1NF, relasi harus bergantung fungsi pada atribut *super key* (Kusrini, M. Kom ; 2010 : 41).

II.9. *Unified Modeling Language (UML)*

UML singkatan dari *Unified Modelling Language* yang berarti bahasa pemodelan standart. *UML* memiliki sintaks dan *semantic*. Ketika kita membuat model menggunakan konsep *UML* ada aturan –aturan yang harus diikuti. Bagaimana elemen pada model-model yang kita buat harus berhubungan satu dengan lainnya harus mengikuti standart yang ada. *UML* bukan hanya sekedar diagram, tetapi juga menceritakan konteksnya. Ketika pelanggan memesan sesuatu dari sistem, bagaimana transaksinya? Bagaimana sistem mengatasi error yang terjadi? Bagaimana keamanan terhadap sistem yang ada kita buat? Dan sebagainya dapat dijawab dengan *UML*.

UML diaplikasikan untuk maksud tertentu, biasanya antara lain untuk :

1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat lunak dengan bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasikan sistem yang ada, proses-proses dan organisasinya.

UML telah diaplikasikan dalam investasi perbankan, lembaga kesehatan, departemen pertahanan, sistem terdistribusi, sistem pendukung alat kerja, retail, sales, dan supplier.

Blok pembangunan utama *UML* adalah diagram. Beberapa diagram ada yang rinci (jenis *timing diagram*) dan lainnya ada yang bersifat umum (misalnya diagram kelas). Para pengembang sistem berorientasikan objek menggunakan bahasa model untuk menggambarkan, membangun dan mendokumentasikan sistem yang mereka

rancang. *UML* memungkinkan para anggota team untuk bekerja sama dalam mengaplikasikan beragam sistem. Intinya, *UML* merupakan alat komunikasi yang konsisten dalam mensupport para pengembang sistem saat ini. Sebagai perancang sistem mau tidak mau pasti menjumpai *UML*, baik kita sendiri yang membuat sekedar membaca diagram *UML* buatan orang lain (Prabowo Pudjo Widodo Herlawati ; 2011 ; 6).

II.9.1. Diagram-diagram UML

Beberapa literatur menyebutkan bahwa *UML* menyediakan Sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa yang digabung, misalnya diagram komunikasi, diagram urutan, dan diagram pewaktuan digabung menjadi diagram interaksi. Namun demikian model-model itu dapat dikelompokkan berdasarkan sifatnya yaitu statis atau dinamis. Jenis diagram itu antara lain :

1. Diagram Kelas. Bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi, serta relasi-relasi diagram. Diagram ini umum dijumpai pada pemodelan sistem berorientasi objek. Meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas.
2. Diagram paket (*Package Diagram*) bersifat statis. Diagram ini memperlihatkan kumpulan kelas-kelas merupakan bagian dari diagram komponen.
3. Diagram *Use Case* bersifat statis. Diagram ini memperlihatkan himpunan *use-case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama

sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna.

4. Diagram interaksi dan *Sequence* (urutan). Bersifat dinamis. Diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalam waktu tertentu.
5. Diagram komunikasi (*Communication Diagram*) bersifat dinamis. Diagram sebagai pengganti diagram kolaborasi *UML* yang menekankan organisasi *structural* dari objek-objek yang menerima serta mengirim pesan.
6. Diagram *Statechart* (*Statechart Diagram*) bersifat dinamis. Diagram status memperlihatkan keadaan-keadaan pada sistem, memuat status (*State*), transisi kejadian serta aktifitas. Diagram ini terutama penting untuk memperlihatkan sifat dinamis dari antarmuka (*interface*), kelas, kolaborasi dan terutama penting pada pemodelan sistem-sistem yang reaktif.
7. Diagram aktivitas (*Activity Diagram*) bersifat dinamis. Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi suatu sistem dan member tekanan pada aliran kendali antar objek.
8. Diagram komponen (*Component Diagram*) bersifat statis. Diagram komponen ini memperlihatkan organisasi serta kebergantungan sistem/perangkat lunak pada komponen-komponen yang telah ada sebelumnya. Diagram ini berhubungan diagram kelas dimana komponen dipetakan kedalam satu atau lebih kelas-kelas. Antarmuka-antarmuka serta kolaborasi-kolaborasi.

9. Diagram *Deployment* (*Deployment Diagram*) bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (*run time*). Memuat simpul-simpul beserta komponen-komponen yang ada di dalamnya. Diagram *Deployment* berhubungan erat dengan diagram komponen dimana diagram ini memuat satu atau lebih komponen-komponen. Diagram ini sangat berguna saat aplikasi kita berlaku sebagai aplikasi yang dijalankan pada banyak mesin (*distributed computing*).

Kesembilan diagram ini tidak mutlak harus digunakan dalam pengembangan perangkat lunak, semuanya dibuat sesuai dengan kebutuhan.

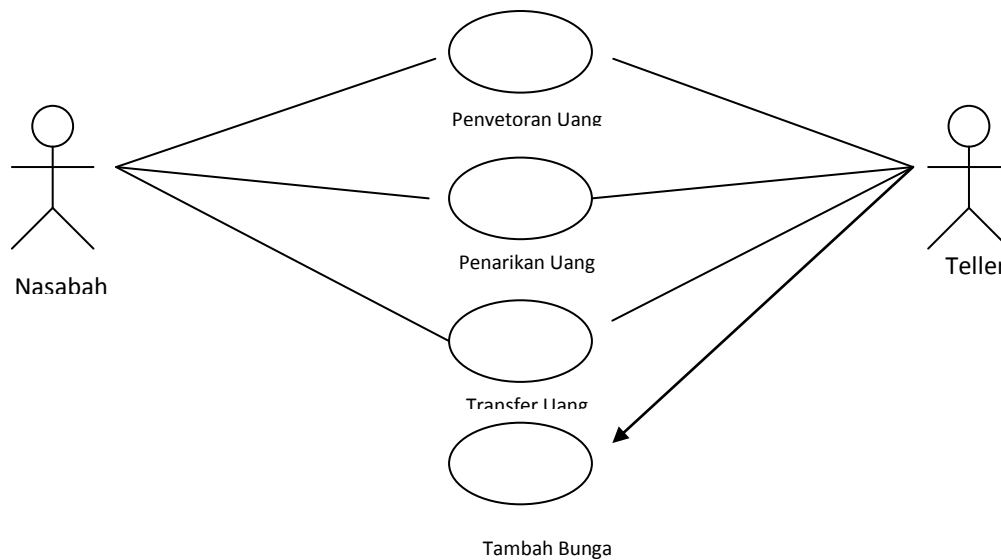
II.9.2. Diagram Use Case (*use case diagram*)

Use Case menggambarkan *external view* dari sistem yang akan kita buat modelnya. Menurut Pooley (2005:15) mengatakan bahwa model *use case* dapat dijabarkan dalam diagram, tetapi yang perlu diingat.

komponen pembentuk diagram *use case* adalah :

1. Aktor (*actor*), menggambarkan pihak-pihak yang berperan dalam sistem.
2. *Use Case*, aktivitas/ sarana yang disiapkan oleh bisnis/sistem.
3. Hubungan (*Link*), aktor mana saja yang terlibat dalam *use case* ini.

Gambar di bawah ini merupakan salah satu contoh bentuk diagram *use case*.



Gambar II.3: Diagram Use Case

(Sumber : Probowo Pudjo Widodo ; 2011:17)

1. Diagram Kelas (*Class Diagram*)

Diagram kelas adalah inti dari proses pemodelan objek. Baik *forward engineering* maupun *reverse engineering* memanfaatkan diagram ini *forward engineering* adalah proses perubahan model menjadi kode program sedangkan *reverse engineering* sebaliknya merubah kode program menjadi model (Probowo Pudji Widodo; 2011 : 37).

2. Diagram Aktivitas (*Activity Diagram*)

Diagram aktivitas lebih memfokuskan diri pada eksekusi dan alur sistem dari pada bagaimana sistem dirakit. Diagram ini tidak hanya memodelkan software melainkan memodelkan bisnis juga. Diagram aktivitas menunjukkan aktivitas sistem dalam kumpulan aksi-aksi. Ketika digunakan dalam pemodelan

software, diagram aktivitas merepresentasikan pemanggilan suatu fungsi tertentu misalnya *call*. Sedangkan bila digunakan dalam pemodelan bisnis, diagram ini menggambarkan aktivitas yang dipicu oleh kejadian-kejadian diluar seperti pemesanan atau kejadian-kejadian internal misalnya penggajian tiap jumat sore (Probowo Pudji Widodo ;2011 : 143-145).

Aktivitas merupakan kumpulan aksi-aksi. Aksi-aksi nelakukan langka sekali saja tidak boleh dipecah menjadi beberapa langkah-langkah lagi. Contoh aksinya yaitu :

- a. Fungsi Matematika
- b. Pemanggilan Perilaku
- c. Pemrosesan Data

Ketika kita menggunakan diagram aktivitas untuk memodelkan perilaku suatu *classifier* dikatakan kontek dari aktivitas. Aktivitas dapat mengakses atribut dan operasi *classifier*, tiap objek yang terhubung dan parameter-parameter jika aktivitas memiliki hubungan dengan perilaku. Ketika digunakan dengan model proses bisnis, informasi itu biasanya disebut *process-relevant data*. Aktivitas diharapkan dapat digunakan ulang dalam suatu aplikasi, sedangkan aksi biasanya *specific* dan digunakan hanya untuk aktivitas tertentu.

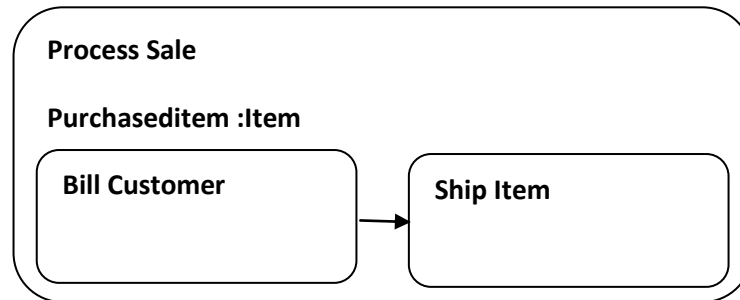
Aktivitas digambarkan dengan persegi panjang tumpul. Namanya ditulis di kiri atas. Parameter yang terlibat dalam aktivitas ditulis dibawahnya.



Gambar II.4: Aktivitas serderhana tanpa rincian

(Sumber : Probowo Pudjo Widodo ; 2011:145)

Detail aktivitas dapat dimasukan di dalam kotak. Aksi diperlihatkan dengan symbol yang sama dengan aktivitas dan namanya diletakkan didalam persegi panjang.



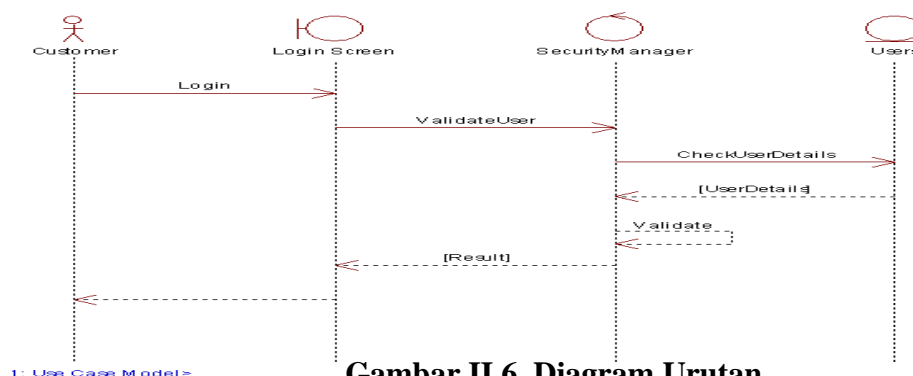
Gambar II.5: Aktivitas dengan detail rincian

(Sumber : Probowo Pudjo Widodo ; 2011:145)

3. *Sequence Diagram*

Ada tiga diagram primer UML dalam memodelkan scenario interaksi, yaitu diagram urutan (*sequence diagram*), diagram waktu (*timing diagram*) dan diagram komunikasi (*communication diagram*).

Diagram yang paling banyak dipakai adalah diagram urutan. Gambar II.6. memperlihatkan contoh diagram urutan dengan notasi-notasinya yang akan dijelaskan nantinya.



Gambar II.6. Diagram Urutan

Sumber : Probowo Pudjo Widodo