BAB III

OBJEK DAN METODE PENELITIAN

3.1. Objek Penelitian

Penulis melakukan penelitian di Bengkel Trijaya Motor Bandung yang berlokasi di Jl. Leuwi Panjang No. 111 Bandung Telpon 022-70221812

3.1.1. Sejarah Singkat Perusahaan

Bengkel Trijaya Motor sudah hampir 6 tahun berjalan usaha bisinis bengkel motor, mulai berdiri tanggal 25 juli tahun 2005 di Jl. Leuwi Panjang No. 111 Bandung.

3.1.2. Visi Dan Misi

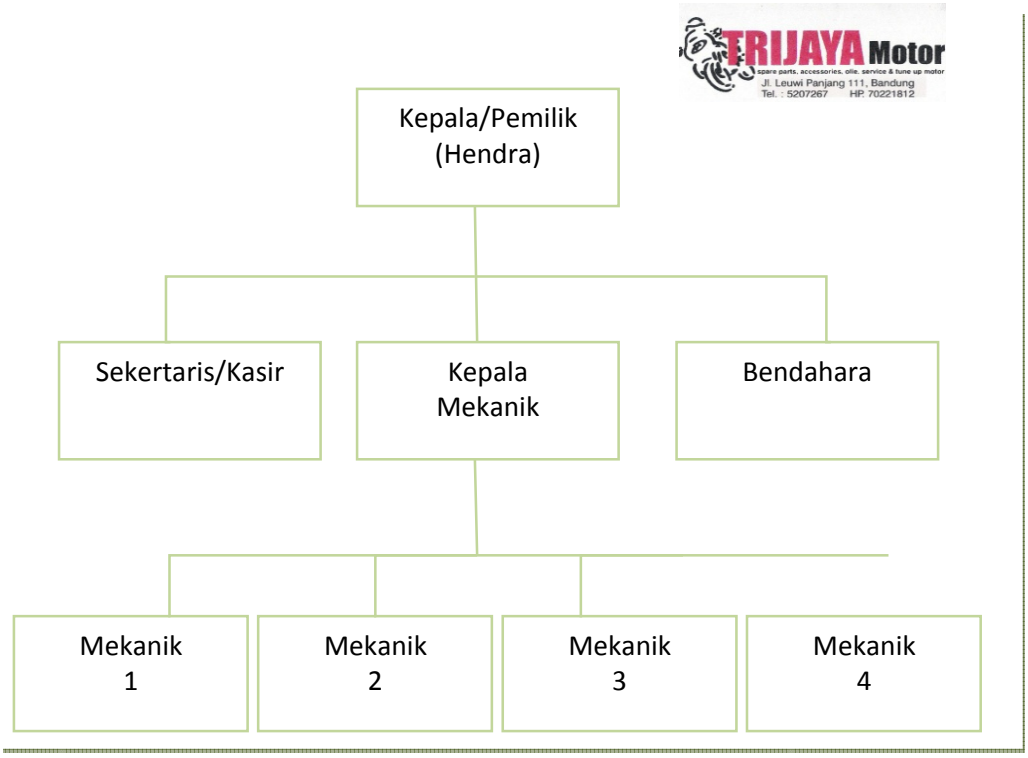
a. Visi

1. Terbaik dalam pelayanan servis di bengkel.
2. Terkenal dalam menangani segala macam problem motor.
3. Terkemuka dan nomor satu di kota Bandung.

b. Misi

1. Mewujudkan pelayanan servis yang profesional.
2. Mewujudkan keahlian mekanik dalam menangani masalah
3. Mewujudkan ketepatan analisis dalam menentukan suatu kerusakan
4. Mewujudkan daya tarik bagi masyarakat di Bengkel Trijaya Motor.

3.1.3. Struktur Organisasi



Gambar 3.1 *Struktur Organisasi Trijaya Motor*

3.1.4. Deskripsi Tugas

Deskripsi tugas *(Job Description)* adalah suatu rincian yang menunjukan posisi, tanggung jawab, wewenang, fungsi dan tugas yang harus dilakukan. Deskripsi tugas perlu dibuat agar masing-masing mengerti tugas dan tanggung jawabnya. Adapun deskripsi tugas *(Job Description)* pada Bengkel Trijaya Motor adalah sebagai berikut:

1. Pemilik
    a. Memiliki wewenang dan pengambil keputusan sepenuhnya atas kegiatan di bengkel.
    b. Bertanggung Jawab penuh terhadap semua kegiatan di bengkel.

c. Mengontrol seluruh kinerja pegawai.

d. Dan memberikan arahan terhadap pegawai untuk kemajuan usaha.

2. Kepala Mekanik

Bertanggung jawab dan mengarahkan bawahannya/mekanik lain terutama terhadap pelayanan service.

3. Bagian Bendahara

Bagian Keuangan mempunyai tugas untuk mengatur semua lalu lintas keuangan dan membuat laporan keuangan.

4. Mekanik

a. Memberikan layanan pelanggan terutama dalam pelayanan service.

b. Mengecek dan memperbaiki sepeda motor.

5. Kasir

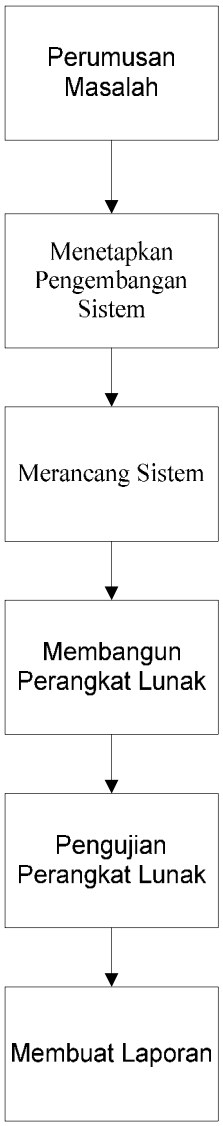
a. Menerima dan melayani pelanggan.

b. Mencatat seluruh transaksi penjualan maupun pelayanan service, Serta membuat laporan pemasukan harian dan laporan jasa service harian dan pembelian barang.

3.2. Metode Penelitian

Metode penelitian merupakan cara ilmiah untuk mendapatkan data dengan tujuan dan kegunaan tertentu.

Berikut adalah hasil metode penelitian secara struktur dalam merencanakan membangun sistem informasi.



Gambar 3.2 Metode penelitian secara struktur

3.2.1. Perumusan Masalah

Perumusan masalah yaitu menetapkan permasalahan yang ada pada di Bengkel Trijaya Motor, peneliti dapatkan dengan cara pengumpulan data. Metode pengumpulan data yang di gunakan penulis adalah dengan memadukan data dari sumber data primer dan data sekunder, sumber data primer dengan beberapa metode pengumpulan data.

3.2.1.1. Sumber Data Primer

Adapun sumber data primer yang dibutuhkan dalam pembuatan aplikasi pelayanan service, penjulan dan pembelian pada Bengkel Trijaya Motor, adalah sebagai berikut:

1. Wawancara (*Interview*).

   Penelitian dalam hal ini penulis mengumpulkan data dengan cara mengajukan tanya-jawab kepada pihak bengkel atau pegawai pada Bengkel Trijaya Motor, yang berhubungan mengenai pelayanan service, penjualan dan pembelian.

2. Pengamatan (*Observasi*).

   Pengamatan dilakukan untuk mengetahui secara langsung alur yang ada pada bangkel tersebut dan mengamati bagaimana sistem yang sedang berjalan yang mengenai pelayanan service, penjualan dan pembelian. Dari hasil pengamatan ini penulis dapat mengetahui kelemahan-kelemahan sistem yang sedang berjalan sehingga memudahkan penulis dalam pembuatan aplikasi pelayanan service, penjualan dan pembelian yang diharapkan dapat dipakai di bengkel tersebut dan memudahkan dalam mengerjakan suatu pekerjaan.

3.2.1.2. Sumber Data Sekunder

Adapun sumber data sekunder yang dibutuhkan dalam pembuatan aplikasi pelayanan service, penjualan dan pembelian pada Bengkel Trijaya Motor, adalah sebagai berikut:

1) Data-data yang berupa tanya-jawab mengenai sejarah perusahaan, visi dan misi, struktur organisasi dan deskripsi jabatan.
2) Data-data yang mengenai pelayanan service, penjualan dan pembelian pada Bengkel Trijaya Motor.

3.2.2. Menetapkan Pengembangan Sistem

Metode pengembangan sistem digunakan untuk memenuhi kebutuhan pengembangan sistem sehingga sistem yang dihasilkan akan sesuai dengan yang diharapkan. Dalam hal ini penulis menggunakan metode pendekatan terstruktur dan metode pengembangan sistem menggunakan *Sequensial linear*. Untuk lebih jelasnya berikut penjelasan dari masing-masing metode :

3.2.2.1. Metode Pendekatan Sistem

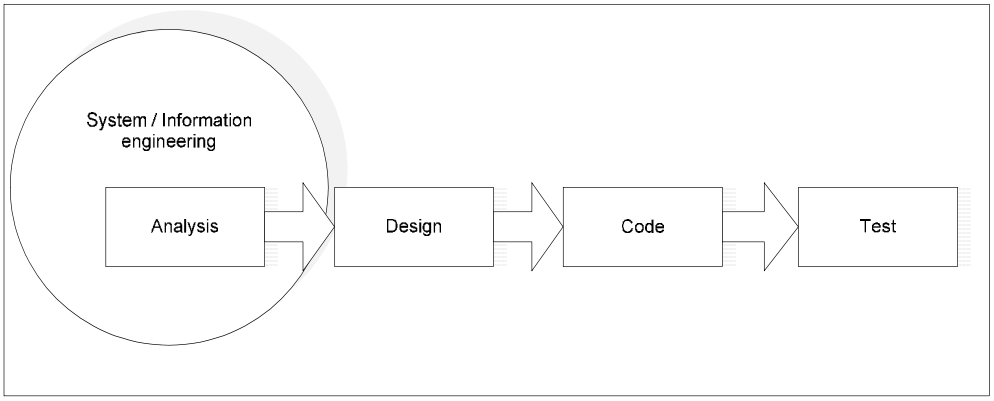
Metode pendekatan sistem yang penulis buat dalam pembuatan skripsi adalah metode terstruktur. Metode pendekatan perancangan terstruktur dimulai dari awal tahun 1970. Pendekatan terstruktur dilengkapi dengan alat-alat (tools) dan teknik-teknik (techniques) yang dibutuhkan dalam pengembangan sistem, sehingga hasil akhir dari sistem yang dikembangkan akan diperoleh sistem yang strukturnya didefinisikan dengan baik dan jelas.

Melalui pendekatan terstruktur, permasalahan yang komplek di organisasi dapat dipecahkan dan hasil dari sistem akan mudah untuk dipelihara, fleksibel, lebih memuaskan pemakainya, mempunyai dokumentasi yang baik, tepat waktu, sesuai dengan anggaran biaya pengembangan, dapat meningkatkan produktivitas dan kualitasnya akan lebih baik (bebas kesalahan).

3.2.2.2. Metode Pengembangan Sistem

Metode pengembangan sistem yang penulis gunakan dalam pembuatan skripsi adalah *"Linear Sequential Model"*. Model ini sering disebut dengan *"classic life cycle"* atau model waterfall. Model ini adalah model yang muncul pertama kali yaitu sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai didalam *Software Engineering (SE)*.

Sequensial linear mengusulkan sebuah pendekatan kepada perkembangan perangkat lunak yang sistematik dan *Sequensial* yang mulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian dan pemeliharaan. Dimodelkan setelah siklus rekayasa konvensional, model *Sequensial linear* melingkupi aktifitas-aktifitas:



Gambar 3.3 Model *Sequensial linear*

[Sumber : Roger S. Pressman (2002 : 37)]

1. Rekayasa dan permodelan sistem/informasi.

   Karena perangkat lunak selalu merupakan bagian dari sebuah sistem (bisbis) yang lebih besar, kerja dimulai dengan membangun syarat dari semua elemen

sistem dan mengalokasikan beberapa subset dan kebutuhan perangkat lunak tersebut. Pandangan sistem ini penting ketika perangkat lunak harus berhubungan dengan elemen-elemen yang lain, seperti perangkat lunak, manusia, dan database. Rekayasa dan analisis menyangkut pengumpulan kebutuhan pada tingkat sistem dengan sejumlah kecil analisis serta desain tingkat puncak. Rekayasa informasi mencakup juga pengumpulan kebutuhan pada tingkat bisnis strategi dan tingkat area bisnis.

2. Analisis kebutuhan perangkat lunak

   Proses pengumpulan kebutuhan diintensifkan dan difokuskan, khususnya pada perangkat lunak. Untuk memahami sifat program yang dibangun, rekayasa perangkat lunak (analis) harus memahami domain informasi, tingkah laku, unjuk kerja dan antar muka *(interface)* yang diperlukan.kebutuhan baik untuk sistem maupun perangkat lunak didokumentasikan dan dilihat lagi dengan pelanggan.

3. Desain

   Desain perangkat lunak sebenarnya adalah proses multi langkah yang berfokus pada empat atribut sebuah program yang berbeda; struktur data, arsitektur perangkat lunak, representasi interface, dan detail (algoritma) procedural. Proses desain menerjemahkan syarat/kebutuhan kedalam sebuah representasi perangkat lunak yang dapat diperkirakan demi kualitas sebelum dimulai pemunculan kode. Sebagaimana persyaratan, desain didokumentasikan dan menjadi bagian dari konfigurasi perangkat lunak.

4. Geneasi kode

Desain harus direjemahkan ke dalam bentuk mesin yang bisa dibaca. Langkah pembuatan kode melakukan tugas ini. Jika desain dilakukan dengan cara yang lengkap, pembuatan kode diselesaikan secara mekanis.

5. Pengujian

   Sekali kode dibuat, pengujian program dimulai. Proses pengujian berfokus pada logika internal perangkat lunak, memastikan bahwa semua pernyataan sudah diuji, dan pada fungsional ekstenal fungsional – yaitu mengarahkan pengujian untuk menemukan kesalahan-kesalahan dan memastikan bahwa input yang dibatasi akan memberikan hasil aktual yang sesuai dengan hasil yang dibutuhkan. Roger S. Pressman (2002 : 36-38).

3.2.3. Merancang Sistem

1. **Bagan Alir Dokumen (*Flow Map*)**

   Flow Map merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan. Flow Map digunakan untuk menganalisis bagaimana hubungan antara sub kerja yang akan menggerakkan sistem. Setelah diketahui bagian-bagian yang terlibat dalam sistem, maka akan diketahui berapa jumlah entitas yang terkait dengan sistem yang dianalisis dan dirancang.

2. **Diagram Konteks**

   Merupakan model grafis yang memperlihatkan sistem dalam bentuk paling umum / global dan digunakan untuk mendefinisikan serta memperlihatkan lingkup atau batas sistem yang akan ditelaah.

Diagram Konteks selalu mengandung satu proses saja. Proses ini mewakili proses dari seluruh sistem. Diagram Konteks menggambarkan hubungan input / output antara sistem dengan kesatuan luar (eksternal entity).

3. *Data Flow Diagram* (DFD)

DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur. DFD dapat menggambarkan arus data di dalam sistem dengan terstruktur dan jelas. DFD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir.

4. Kamus Data

Salah satu komponen kunci dalam sistem manajemen database adalah file khusus yang disebut kamus data *(data dictionary*). Kamus data merupakan katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Dengan menggunakan kamus data, analisis sistem dapat mendefinisikan data yang mengalir berisi informasi tentang struktur database. Kamus data biasanya dipelihara secara otomatis oleh sistem manajemen database.

5. Perancangan Basis Data

Bentuk normalisasi yang biasanya digunakan pada normalisasi adalah bentuk:

a. Normalisasi

1. Bentuk normalisasi I/1-NF *(first normal form)*

Bentuk Normal tahap pertama (1NF) terpenuhi jika sebuah tabel tidak memiliki atribut bernilai banyak *(Multivalued Attribute)* atau lebih dari satu atribut dengan domain nilai yang sama.

2. Bentuk normalisasi II / 2-NF *(Second-Normal Form)*

   Bentuk tahap normal kedua (2NF) terpenuhi jika pada sebuah tabel, semua atribut yang tidak termasuk dalam *key primer* memiliki ketergantungan fungsional (KF) pada *key primer* secara utuh. Sebuah table dikatakan tidak memenuhi 2NF, jika ketergantungannya hanya besifat parsial (hanya tergantung pada sebagian dari *key primer*).

3. Bentuk normalisasi 3-NF*(Third –Normal Form)*

   Suatu relasi memenuhi bentuk III (3-NF) jika dan hanya jika relasi tersebut memenuhi 2-NF, dan setiap kolom bukan kunci tidak tergantung secara fungsional kepada kolom bukan kunci yang lain dalam relasi tersebut. Dengan kata lain setiap kolom bukan kunci primer tidak memiliki ketergantungan secara transitif terhadap kunci primer.

b. Tabel Relasi

   Relasi tabel adalah gambaran tentang hubungan yang terjadi antar tabel-tabel yang akan digunakan dalam program aplikasi pemecahan dari *flat file* yang menurut teknik normalisasi sehingga pemecahan tersebut memiliki sebuah kunci yang menghubungkan relasi datanya.

Dengan teori sistem informasi serta teknik-teknik atau peralatan untuk membuat atau menganalisa suatu sistem maka dapat dibuat suatu aplikasi untuk

mempermudah dalam pengolahan data-data yang akan dialirkan di dalam sistem tersebut.

3.2.4. Membangun Perangkat Lunak

Membangun perangkat lunak yaitu menuangkan hasil analisis kedalam bentuk fisik yaitu perangkat lunak. Dalam mebangun perangkat lunak tersebut, penulis menggunakan bahasa Visual Basic 6.0, dan menggunakan SQL Server 2000 sebagai pennyimpanan basis data.

3.2.5. Pengujian Perangkat Lunak

Faktor pengujian *software* adalah teknik untuk menguji perangkat lunak, mempunyai mekanisme untuk menentukan data uji yang dapat menguji perangkat lunak secara lengkap dan mempunyai kemungkinan tinggi untuk menemukan kesalahan. Perangkat lunak dapat diuji dengan dua cara, yaitu :

1. *White Box Testing*

   Pengujian *White Box* (*glass box*) adalah pengujian yang didasarkan pada pengecekan terhadap detail perancangan, menggunakan struktur kontrol dari desain program secara prosedural untuk membagi pengujian ke dalam beberapa kasus pengujian.

2. *Black Box Testing*

   Pengujian *Black Box* adalah pengujian aspek *fundamental* sistem tanpa memperhatikan struktur logika internal perangka lunak. Metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan

benar. Pengujian *Black Box* merupakan metode perancangan data uji yang didasarkan pada spesifikasi perangkat lunak. Data uji dibangkitkan, dieksekusi pada perangkat lunak dan kemudian keluaran dari perangkat lunak dicek apakah telah sesuai dengan yang diharapkan.

Agar *software* dapat dikatakan layak atau sesuai dengan kebutuhan pengguna perangkat lunak maka dibutuhkan pengujian yang dapat menilai suatu *software* itu layak atau tidak digunakan. Dalam hal ini pengujian menggunakan *Black Box Testing* karena untuk mengetahui perangkat lunak berfungsi dengan benar atau tidak.

Ujicoba blackbox berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya :

1. Fungsi-fungsi yang salah atau hilang
2. Kesalahan Interface
3. Kesalahan dalam struktur data atau akses database eksternal
4. Kesalahan performa
5. Kesalahan inisialisasi dan terminasi.

3.2.6. Membuat Laporan

Membuat Laporan adalah hasil secara tertulis, sehingga hasil uji yang diharapkan dari mulai perumusahan masalah sampai akhir menjadi terstruktur dan secara ilmiah untuk mendapatkan data dengan tujuan dan kegunaan tertentu. Yang digunakan penulis dalam pembuatan aplikasi pelayanan service, penjualan dan pembelian sparepart motor.