BAB III

OBJEK DAN METODE PENELITIAN

3.1 Objek Penelitian

Objek penelitian ini adalah penelitian deskriptif analitik yakni suatu penelitian yang bertujuan untuk memberikan gambaran tentang realitas pada objek yang di teliti secara objektif. Penelitian ini menekankan pada satu variabel yakni pembangunan sistem informasi penjualan dan pembelian.

3.1.1 Sejarah Singkat Perusahaan

CV. Abadi Mitra Utama adalah salah satu perusahaan yang bergerak dibidang hanya bergerak dalam bidang perbengkelan pembuatan *sparepart* industri terutama industri texstile yang didirika oleh Rito Samsunurrohmat, BA. Tanggal 5 Januari 1997 merupakan momentum sejarah berdirinya CV. Abadi Mitra Utama dengan Surat Izin Usaha Perdagangan (SIUP) Kecil dengan izin nomor 510/1-0796/2004/1003-DISKUKM&PERINDAG/2008 yang diperoleh dari Pemerintahan Kota Bandung, Dinas Perindustrian dan Perdagangan.

Seiring dengan waktu CV ABADI MITRA UTAMA terus berkembang, dengan dibukanya divisi electric pada tahun 1999, dengan menghimpun dan mengerahkan tenaga-tenaga ahli dan berpengalaman yang berpotensi dalam

bidangnya yang berkeinginan untuk berkarya dan berinovasi dalam mengisi pembangunan di tanah air ini. Dan

3.1.2 Visi dan Misi Perusahaan

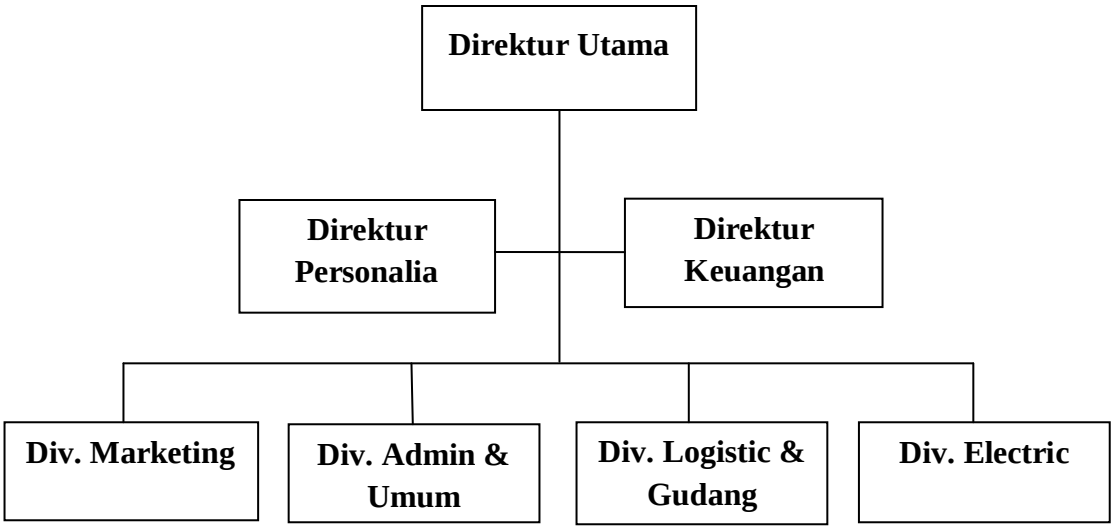
CV. Abadi Mitra Utama mempunyai visi dan misi sebagai berikut :

Visi : Tegas, jujur, konsekwen dan bijaksana dalam mengambil keputusan, menjalankan praturan dan prosedur yang sudah ditentukan serta siap menanggung resiko yang akan diterima.

Misi : Dalam menjalankan tugas/kegiatan senantiasa terencana, teliti dan memanfaatkan waktu semaksimal mungkin.

3.1.3 Struktur Organisasi Perusahaan

Struktur organisasi merupakan suatu kerangka dasar dan pengenalan dalam manajemen sebuah organisasi/perusahaan, karena dengan mempelajari struktur organisasi, perusahaan akan dengan mudah untuk dapat mempelajari kehidupan perusahaan yang sedang berjalan baik dari segi otorisasi, tanggung jawab pada tugas masing - masing bagian. Maka dari itu dengaan adanya pengorganisasian dalam suatu perusahaan dapat menghindari penyalahgunaan tugas dan fungsi dari unit - unit organisasi yang ada di dalamnya. Adapun struktur organisasi pada CV. Abadi Mitra Utama dapat di gambarkan sebagai berikut:



Gambar 3.1 Struktur Organisasi CV. Abadi Mitra Utama

(Sumber : CV. Abadi Mitra Utama)

3.1.4 Deskripsi Tugas

Dengan melihat struktur organisasi di atas, maka dapat diuraikan mengenai tugas dan fungsi dari masing – masing bagian yang terlibat dalam organisasi:

1. Direktur Utama

   Bertanggung jawab kepada dewan direksi/pemegang saham. Berwenang memerintah direktur umum, direktur keuangan dan operational.

2. Direktur Personalia

   Bertanggung jawab kepada direktur utama. Berwenang memerintah seluruh manager, staf dan pelaksana sesuai dengan kapasitas jabatan.

3. Direktur Keuangan

   Bertanggung jawab kepada direktur utama. Berwenang memerintah seluruh manager, staf dan pelaksana sesuai dengan kapasitas jabatan dan mengecek keuangan baik pengeluaran dan pemasukan yang terjadi di CV. Abadi Mitra Utama tiap bulanya.

4. Div. Marketing

   Bertanggung jawab kepada marketing manager dan berkoordinasi dengan departemen lain.

5. Div. Admin & Umum

   Bertanggung jawa kepada direktur umum dan personalia. Berwenang memerintah staf administrasi dan umum, pelaksanaan dan berkoordinasi dengan departemen lain.

6. Div. Logistic & Gudang

   Bertanggung jawa kepada direktur operational. Berwenang memerintah dan berkoordinasi dengan departemen lain.

7. Div. Electric

   Bertanggung jawab kepada direktur umum berkoordinasi dengan departemen lain.

3.2 Metode Penelitian

Metode penelitian merupakan cara ilmiah untuk, mendapatkan data dengan tujuan dan kegunaan tertentu. Cara ilmiah berarti kegiatan penelitian itu didasarkan pada ciri – ciri keilmuan, yaitu : rasional, empiris, dan sistematis.

a. Rasional berarti kegiatan penelitian dilakukan dengan cara – cara yang masuk akal, sehingga terjangkau nalar manusia.
b. Empiris berarti cara – cara yang dilakukan itu dapat diamati oleh indera manusia, sehingga orang lain dapat mengamati dan mengetahui cara – cara yang digunakan.
c. Sistematis artinya proses yang digunakan dalam penelitian menggunakan langkah – langkah tertentu yang bersifat logis.

3.2.1 Desain Penelitian

Dalam melakukan penelitian, diperlukan suatu desain. Desain penelitian yang meliputi perencanaan dan kajian yang sesungguhnya seperti lokasi kajian, pemilihan sampel, serta pengumpulan dan analisis data.

3.2.2 Jenis dan Metode Pengumpulan Data

Dalam penelitian ini metode pengumpulan data yang akan digunakan meliputi :

1. Wawancara (interview) pemilik perusahaan dan karyawan, yaitu teknik mengumpulkan data yang dilakukan melalui tatap muka dan wawancara antara pengumpul data (pencatat) dengan responden. Wawancara dilakukan baik secara langsung maupun dengan menggunakan pedoman ’’daftar pertanyaan” dari kuesioner sebagai instrumen penelitian.
2. Pengamatan (observasi), yaitu teknik pengumpulan data melalui pengamatan langsungkepada objek penelitian. Teknik observasi biasanya dilakukan

bersamaan dengan teknik lain untuk mengamati keadaan fisik, lokasi atau daerah penelitian secara sepintas *(on the spot)* dan dengan melakukan pencatatan seperlunya.

3.2.2.1 Sumber Data Primer

1. Obsevasi

   Penulis melakukan pencarian data dan informasi dengan cara meninjau dan meneliti secara langsung data-data yang berkaitan dengan bahan laporan.

2. Wawancara

   Penulis melakukan tatap muka dan tanya jawab secara langsung antara penulis sebagai pewawancara sedangkan nara sumber adalah Bagian Penjualan dan Pembelian pada CV. Abadi Mitra Utama Bandung.

3.2.2.2 Sumber Data Sekunder

Studi literatur

Penulis melakukan penelitian melalui data-data yang diperoleh dari buku-buku yang berupa dokumen - dokumen perusahaan yang bersifat rahasia yang berkaitan dengan persediaan dan penjualan barang yang menjadi akar permasalahan, catatan sebagai teoritis yang memadai dalam upaya melakukan analisa yang berhubungan dengan masalah yang diteliti.

3.2.3 Metode Pendekatan Dan Pengembangan Sistem

Adapun beberapa metode pendekatan / pengembangan yang digunakan diantaranya :

3.2.3.1 Metode Pendekatan Sistem

Prototype merupakan suatu metode dalam pengembangan sistem yang menggunakan pendekatan untuk membuat sesuatu program dengan cepat dan berharap sehingga segera dapat dievaluasi oleh pemakai. Hal ini berbeda dengan pendekatan SDLC tradisional (konvensional) yang lebih banyak menghabiskan waktu untuk menghasilkan spesifikasi yang sangat rinci sebelum pemakai mengalami kesulitan dalam memahami spesifikasi sistem berakibat bahwa pemakai tidak begitu paham sampai pengujian dilakukan. Selain itu, *prototype* membuat proses pengembangan sistem informasi menjadi lebih cepat dan lebih mudah, terutama pada keadaan kebutuhan pemakai sulit untuk di identifikasi.

3.2.3.2 Metode Pengembangan Sistem

Metode yang digunakan dalam pengembangan untuk membangun sistem informasi ini yaitu metode *prototype* yang dimana *prototype* merupakan suatu metode dalam pemgembangan sistem yang menggunakan pendekatan untuk membuat suatu program dengan cepat dan bertahap dan *prototype* juga membuat proses pengembangan sistem informasi menjadi lebih baik cepat dan lebih mudah. Dimana tahapan-tahapan yang harus dilaksanakan adalah sebagai berikut:

1. Identifikasi Kebutuhan Sistem

   Pada tahap ini merupakan tahap awal dalam membangun sebuah sistem informasi, dimana antara pemakai sistem (*user*) dan pengembang sistem bertemu. User dijelaskan tentang kebutuhan sistem yang akan dibangun oleh pengembang sistem.

2. Membuat *Prototype*

   Setelah menganalisa sistem yang akan dibangun, pengembangan sistem mulai membuat *prototype*.

3. Menguji *Prototype*

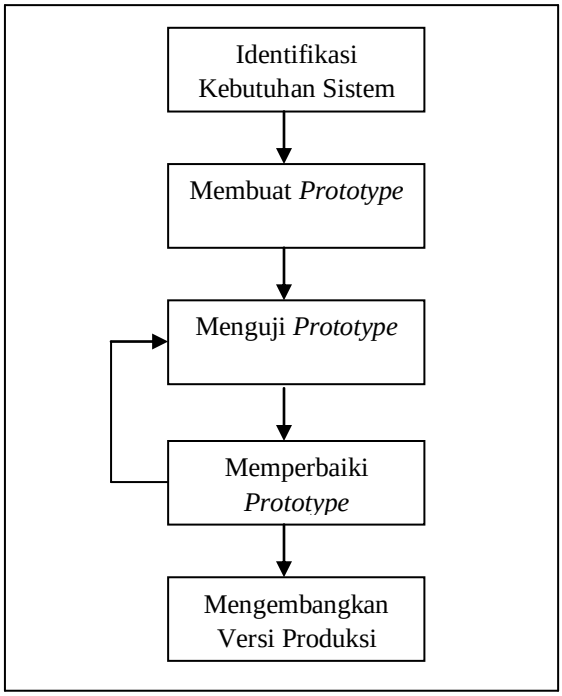
   Tahapan ini adalah proses penilaian terhadap *prototype* yang telah dibuat, apakah sesuai dengan kebutuhan atau tidak, jika tidak maka *prototype* akan diperbaiki.

4. Memperbaiki *Prototype*

   Setelah ditemukan letak kesalahan dari *prototype* yang dirancanga, pada tahapan ini penyusun akan memperbaiki *prototype* yang ada. Setelah itu akan diuji kembali sehingga *prototype* sesuai dengan keinginan pengguna.

5. Mengembangkan Versi Produksi

   Setelah aplikasi dapat berjalan dan memenuhi kebutuhan sistem, maka aplikasi siap dipakai dan jika kebutuhan mendatang bertambah, maka sistem siap dikembangkan sesuai dengan apa yang diharapkan.



Gambar 3.2 Mekanisme Pengembangan Sistem Dengan *Prototype*

(Sumber : Abdul Kadir, 2003 : 417)

Penulis mengambil model proses secara *prototyping* karena :

a. Penyajian aspek-aspek perangkat lunak yang akan dibangun akan nampak bagi pelanggan/pemakai secara cepat (contohnya pendekatan input dan format output). Perancangan kilat membawa kepada konstruksi sebuah *prototype*.
b. *Prototype* dievaluasi oleh kedua belah pihak (*programmer* dan calon *user*) sehingga penyaringan kebutuhan pengembangan perangkat lunak dapat dengan cepat dilakukan sesuai dengan keinginan *user*.
c. *Programmer* akan memahami apa yang harus dilakukan agar kebutuhan *user* dapat dipenuhi.

3.2.3.3 Alat Bantu Analisis dan Perancangan

1. Flow Map

Flow Map memberikan gambaran tentang sebuat sistem, yang akan mempermudah penganalisisan tentang masalah yanga akan di identifikasi, serta menjadi tolak ukur sejauh mana sistem akan dikembangkan. Dari sistem yang sudah dibuat maka di temukan beberapa data dan fakta yang akan dijadikan bahan untuk pengembangan dan penerapan sebuah aplikasi sistem yang akan diuslkan. Maka Flow map merupakan salah satu langkah awal untuk membangun atau mengembangkan sebuah sistem.

2. Diagram Konteks

Diagram konteks adalah level tertinggi dari DFD yang menggambarkan seluruh input ke sistem atau output dari sistem. Ia akan memberikan gambaran tentang keseluruhan sistem, dalam diagram konteks hanya ada satu proses. Tidak boleh ada *store* dalam diagram konteks.

3. Data Flow Diagram

Data Flow Diagram (DFD) adalah suatu network yang menggambarkan suatu sistem automat/komputerisasi, manualisasi atau gabungan dari keduanya, yang penggambaranya disusun dalam bentuk kumpulan komponen sistem yang saling berhubungan sesuai aturan. Keuntungan dari DFD adalah memungkinkan untuk menggambarkan sistem dari level yang paling tinggi kemudian menguraikan menjadi level yang lebih rendah.

4. Kamus Data

Kamus data adalah kumpulan elemen – elemen atau simbol – simbol yang digunakan untuk membantu dalam penggambaran atau pengidentifikasian setiap field atau file dalam sistem.

Kamus data dibuat berdasarkan arus data yang ada pada data flow diagram. Arus data yang berada di DFD bersifat global dan hanya menunjukan nama arus datanya saja. Keterangan lebih lanjut tentang struktur dari suatu arus data di DFD dapat dilihat pada kamus data. Kamus data atau data *dictionary* harus dapat mencerminkan keterangan yang jelas tentang data yang dicatatnya. Untuk keperluan ini maka kamus data harus memuat hal – hal sebagai berikut :

1. Arus Data

   Arus data menunjukan dari mana data mengalir dan kemana data menuju. Keterangan arus data ini perlu dicatat di kamus data untuk memudahkan mencari arus data di dalam Data Flow Diagram (DFD).

2. Nama arus data

   Nama arus data memberikan penjelasan lebih lanjut tentang suatu arus data sehingga dapat langsung mencarinya dengan mudah di kamus data.

3. Alias

   Alias atau nama lain dari data, untuk menyatakan nama lain dari data elemen atau data *store* yang sebenarnya sama dengan data elemen atau data *store* yang telah ada.

4. Bentuk data

   Bentuk data dapat dipergunakan untuk mengelompokan kamus data kedalam kegunaanya sewaktu perancangan sistem.

5. Penjelasan

   Untuk memperjelas tentang makna dari arus data yang dicatat di kamus data, penjelasan dapat di isi dengan keterangan-keterangan tentang arus data tersebut.

5. Perancangan Basis Data

   Perancangan basis data diperlukan agar kita membuat suatu basis data yang lengkap dan efisien.

Langkah – langkah dalam perancangan basis data, diantaranya :

1. Normalisasi

Proses normalisasi merupakan proses pengelompokan elemen data menjadi tabel-tabel yang menunjukan *entity* dan relasinya. Pada dasarnya normalisasi adalah suatu teknik menstrukturkan data dengan cara-cara tertentu untuk membantu mengurangi atau mencegah timbulnya masalah yang berhubungan dengan pengolahan data dalam database. Normalisasi dilakukan sebagai suatu uji coba pada suatu relasi secara berkelanjutan untuk menentukan apakah relasi tersebut sudah baik atau masih melanggar aturan-aturan standar yang diberlakukan pada suatu relasi yang normal.

Dalam perspektif normalisasi, sebuah basis data dikatakan baik jika tabel yang menjadi unsur pembentuk basis data harus dalam keadaan baik dan normal. Sebuah tabel dikatakan normal dan efisien, jika telah memenuhi 3 kategori, yaitu

1. Jika ada dekomposisi (penguraian) tabel, maka dekomposisinya harus dijamin aman (*Lossless – Join Decomposition*).
2. Terpeliharanya ketergantungan fungsional pada saat perubahan data (*dependency preservation*).
3. Tidak melanggar Boyce – Code Normal Form (BCNF).

Beberapa bentuk normal yang sering digunakan sebagai berikut :

a. Bentuk normal pertama (1 Normal Form/NF)

   Suatu tabel dapat terpenuhi jika sebuah table tidak memiliki atribut bernilai banyak (*Multivalued Atribute*) atau lebih dari satu atribut.

b. Bentuk normal kedua (2 Normal Form/NF)

   Suatu tabel dikatakan bentuk normal kedua jika sudah memenuhi bentuk normal pertama dan semua atribut bukan kunci memiliki depedensi terhadap kunci primer.

c. Bentuk normal ketiga (3 Normal Form/NF)

   Suatu tabel dikatakan bentuk normal ketiga jika sudah memenuhi normal kedua dan setiap atribut bukan kunci tidak memiliki depedensi transitif. Normalisasi dilakukan untuk menghindari adanya redudansi *field-field* dari tabel yang ada.

2. Tabel Relasi

   Relasi Tabel secara sederhana dapat dikatakan sebagai suatu database yang didalamnya terdapat tabel – tabel yang saling berelasi satu sama lain. Relasi antar satu tabel dengan tabel yang lainya ditentukan berdasarkan aturan – aturan tertentu.

3. Entity-Relationship Diagram

   ERD merupakan bentuk bagan yang menggunakan relasi dan entitas suatu informasi. Entitas relasi dibuat untuk menggunakan persepsi yang terdiri dari sekumpulan objek dasar yaitu entitas dan hubungan antar entitas. *Entity*

Relationship Diagram (ERD) dapat juga dikatakan suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak.

1. *Entity*

   Pada *ERD*, *entity* digambarkan dengan sebuah bentuk persegi panjang. Entitas diberi nama dengan kata benda dan dapat dikelompokkan dalam empat jenis nama, yaitu orang, benda, lokasi, kejadian (terdapat unsur waktu didalamnya).

2. *Relationship*

   Pada *ERD*, *relationship* dapat digambarkan dengan sebuah bentuk belah ketupat. Relationship adalah hubungan alamiah yang terjadi antara entitas.

3.2.4 Pengujian Software

Pengujian menyajikan *anomaly* yang menarik bagi perekayasa perangkat lunak. Pada proses perangkat lunak, perekayasa terutama berusaha membangun perangkat lunak dari konsep *abstrak ke implementasi* yang dapat dilihat, baru kemudian dilakukan pengujian.

3.2.4.1 Sasaran – Sasaran Pengujian

Terdapat sejumlah aturan yang berfungsi sebagai sasaran pengujian :

a. Pengujian adalah proses eksekusi suatu program dengan maksud menemukan masalah.

b. *Test case* yang baik adalah *test case* yang memiliki probabilitas tinggi untuk menemukan kesalahan yang belum pernah ditemukan sebelumya.

c. Pengujian yang sukses adalah pengujian yang menemukan semua kesalahan yang belum pernah ditemukan sebelumnya.

3.2.4.2 Prinsip Pengujian

Sebelum mengaplikasikan metode untuk mendesain test case yang efektif, perekayasa perangkat lunak harus memahami prinsip dasar menuntun pengujian perangkat lunak. Serangkaian prinsip pengujian akan dijelaskan dibawah ini :

a. Pengujian harus direncanakan lama sebelum pengujian itu dimulai.
b. Prinsip pareto berlaku untuk pengujian perangkat lunak.
c. Pengujian harus mulai dari yang kecil dan berkembang ke pengujian yang besar.
d. Pengujian yang mendalam tidak mungkin.
e. Untuk menjadi paling efektif, pengujian harus dilakukan oleh pihak ketiga yang independent.

3.2.4.3 Pengujian Blackbox

Menurut Roger S Pressman dalam bukunya Rekayasa Perangkat Lunak :

" Pengujian Black – Box berfokus pada persyaratan fungsional perangkat lunak. Pengujian Black – Box memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program ".

Pengujian *Black–Box* bukan merupakan alternative dari teknik *White–Box* , tetapi merupakan pendekatan komplementer yang kemungkinan besar mamapu mengungkap kelas kesalahan daripada metode *White – Box*. Pengujian *White – Box* berusaha menemukan kesalahan dalam kategori sebagai berikut :

1. Fungsi – fungsi yang tidak benar atau hilang
2. Kesalahan *interface*
3. Kesalahan dalam struktur data atau akses *database external*
4. Kesalahan kinerja
5. Inisialisasi dan kesalahan terminasi