

Implementasi Desktop Sistem Inventasi Pada Hudi Motor Karangrayung Grobogan

Alfian Helmi Muhib

A11.2008.04268

Program Studi Teknik Informatika

Universitas Dian Nuswantoro

2013

ABSTRAK

Pengelolaan data sparepart pada Hudi Motor Karangrayung Grobogan yang dilakukan saat ini masih ditulis secara manual dalam buku stok. Dengan banyaknya item dan jumlah serta kegunaan untuk jenis-jenis sepeda motor yang sangat banyak, maka sering terjadi kesalahan dan kekeliruan dalam mendatanya. Belum lagi bentuk fisik barang yang hampir sama dan banyak yang berukuran kecil, maka pada saat proses masuk dan keluar barang sering terjadi kesalahan pendataan. Oleh karena itu diperlukan sebuah sistem yang mampu mengolah data sparepart baik stok, maupun proses penambahan dan pengeluaran barang dengan menggunakan komputer. Hal ini akan memudahkan dalam proses pengontrolan stok dan pembuatan laporan pemasukan dan pengeluaran barang. Disamping itu Software yang dibangun akan memudahkan manajemen dalam menganalisa kebutuhan stok sparepart sehingga persediaan barang menjadi lebih sehat, tidak ada barang yang habis dan tidak ada barang yang melebihi kebutuhan.

Kata kunci : aplikasi, stok, sparepart, data, software.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Bengkel merupakan suatu usaha jenis wirausaha kecil dan menengah yang bergerak dalam bidang jasa pelayanan perbaikan baik itu sepeda motor atau mobil, lebih dari itu bengkel juga melakukan usaha penjualan sparepart guna melengkapi kebutuhan penggantian sparepart kendaraan yang rusak. Salah satu jenis bengkel yang bergerak dalam bidang jasa service sepeda motor adalah bengkel Hudi Motor Karangrayung Grobogan.

Pada saat ini HUDI MOTOR Karangrayung Grobogan dalam menyajikan data yang dibutuhkan oleh bengkel masih manual, dalam hal ini dalam

mengendalikan persediaan stok sparepart atau keluar masuknya jumlah sparepart masih kurang efisien sehingga banyak sekali kesulitan yang ada apalagi data yang harus diolah banyak dikarenakan stok sparepart yang terdiri dari berbagai macam sparepart dengan berbagai macam jenisnya. Maka untuk mengatasi masalah tersebut HUDI MOTOR Karangrayung Grobogan memerlukan suatu sistem pengolahan data yang lebih baik dari sistem yang sebelumnya yaitu dengan sistem informasi persediaan stok sparepart yang akan memberikan hasil pengolahan data yang optimal dan didukung dengan sumber daya manusia yang berkualitas sehingga HUDI MOTOR Karangrayung Grobogan dapat mengontrol kegiatannya dengan mudah cepat akurat dan dapat

memberikan laporan yang diinginkan secara tepat dan cepat setiap saat bila dibutuhkan.

HUDI MOTOR Karangrayung Grobogan pada pelaksanaan tugasnya harus bisa mengatur atau mengelola persediaan sparepart yang ada, sehingga mengetahui jika hendak kehabisaan suatu jenis sparepart sebelum pemesanan dari pelanggan terjadi. Misalnya untuk mengetahui stok sparepart yang masuk maupun sparepart yang keluar, order pembelian dan lain-lain. Masalah ini akan dapat berkurang jika pencatatan tersebut menggunakan alat bantu komputer yang mengolah data secara tepat dan mudah sehingga diperoleh informasi yang cepat dan akurat sesuai yang dibutuhkan.

1.2 Perumusan Masalah

Sehubungan hal yang diatas yang menjadi permasalahan adalah bagaimana HUDI MOTOR Karangrayung Grobogan menyajikan laporan yang relevan sehingga dapat digunakan sebagai acuan untuk perencanaan dan pengendalian jumlah sparepart. Oleh maka itu masalah yang ada dalam jumlah sparepart pada HUDI MOTOR Karangrayung Grobogan dapat dirumuskan sebagai berikut :

Bagaimana mengolahan persediaan sparepart pada HUDI MOTOR Karangrayung Grobogan supaya informasi yang disajikan menjadi lebih cepat, tepat dan akurat.

1.3 Pembatasan Masalah

Dalam pembatasan masalah tentang proses persediaan khususnya pengadaan dan pengeluaran sparepart banyak masalah yang dijumpai. Oleh karena itu penulis membatasi masalah yang akan dianalisis mengingatkan adanya keterbatasan waktu, tenaga dan pikiran dalam penyusunannya. Pembatasan masalah tersebut adalah sebagai berikut :

1. Prosedur pemasukan dan pengeluaran jumlah sparepart (*Stock*).
2. Perancangan aplikasi perangkat lunak terhadap pemasukan dan pengeluaran sparepart.
3. Dengan menggunakan aplikasi Visual FoxPro 9.0.

1.4 Tujuan Tugas Akhir

Tujuan dari penulisan tugas akhir ini adalah untuk merancang sistem informasi yang mampu mendata arus masuk dan keluarnya sparepart pada HUDI MOTOR Karangrayung Grobogan serta stok sparepart dengan cepat dan akurat. Untuk dapat membantu mengefesienkan pekerjaan pada bagian persediaan sparepart dengan menggunakan sistem komputer.

1.5 Manfaat Tugas Akhir

1.5.1 Manfaat Bagi Penulis

Dapat mengembangkan dan menerapkan ilmu yang diperoleh pada perkuliahan dalam kehidupan yang nyata di

masyarakat dan menambah pengetahuan penulis tentang apa yang sebenarnya persediaan barang. Untuk memenuhi salah satu syarat dalam menyelesaikan program strata satu fakultas ilmu komputer khususnya sistem informasi di Universitas Dian Nuswantoro Semarang.

1.5.2 Manfaat bagi Perusahaan

Dapat memberikan sumbangan pikiran dan pertimbangan dalam membuat perencanaan dalam mengambil keputusan pada masa yang akan datang yang didasarkan pada hasil laporan persediaan barang.

Menambahkan kecepatan serta ketepatan dalam pengolahan data sparepart sehingga dapat menyediakan informasi yang dibutuhkan setiap saat. Mempermudah dan mengefesienkan pekerjaan dengan menggunakan sistem komputer.

Memperoleh sistem informasi secara tepat dan akurat, mengetahui distribusi atau perputaran sparepart secara menyeluruh dengan digunakan sistem komputer. Mengurangi resiko kesalahan yang terjadi dalam pengolahan data.

1.5.3 Manfaat Bagi Akademik

Sebagai tambahan referensi perpustakaan yang dapat dimanfaatkan

sebagai media untuk menambah pengetahuan pembaca. Dapat menjadi acuan dan dorongan sebagai tolak ukur dalam mendidik dan membekali ilmu bagi mahasiswa sebelum terjun ke masyarakat.

BAB II

LANDASAN TEORI

2.1 Aplikasi Perangkat Lunak

2.1.1 Pengertian Aplikasi Perangkat Lunak

Aplikasi perangkat lunak atau *software application* adalah *software program* yang memiliki aktivitas pemrosesan perintah yang diperlukan untuk melaksanakan permintaan pengguna dengan tujuan tertentu. *software application* terdiri dari :

1. Bahasa pemrograman (programming language)
2. Program aplikasi (*Application Program*)
3. Program paket / paket aplikasi (*packet program*)
4. Program utilitas (*utility program*)
5. Games dan entertainment dan lain-lain

Untuk mendukung operasi *software application* di atas pengguna komputer dibagi menjadi :

- a. Analisis *system*
- b. Programmer
- c. Operator
- d. Administrator database, yaitu orang yang bertugas melakukan definisi dan manipulasi *software* database. Ini dibutuhkan jika *software* yang dioperasikan akan menggunakan aplikasi database.
- e. Administrator jaringan, yaitu orang yang bertugas mengelola *software system* operasi, *software* aplikasi yang digunakan dalam sebuah jaringan komputer.

2.2 Langkah-Langkah Analisis Sistem

Menurut Jogyanto HM,(2009), langkah-langkah dasar yang digunakan dalam analisis sistem adalah

1. Mengidentifikasi Masalah (*Identify*)

Merupakan langkah pertama yang dilakukan dalam analisis sistem. Masalah dapat didefinisikan sebagai suatu pertanyaan yang di inginkan untuk dipecahkan. Masalah inilah yang menyebabkan sasaran dari sistem tidak dapat dicapai, oleh karena itu langkah pertama yang harus dilakukan analisis adalah

mengidentifikasi dahulu masalah yang terjadi.

Tahap-Tahap yang harus ditempuh :

a. Mengidentifikasi penyebab masalah.

Mengkaji ulang terlebih dahulu subjek-subjek permasalahan yang telah diutarakan oleh manajemen atau yang telah di temukan oleh analisis sistem di tahap perencanaan sistem.

b. Mengidentifikasi titik keputusan.

Titik keputusan menunjukan suatu kondisi yang menyebabkan suatu kejadian lain.

c. Mengidentifikasi personil-personil kunci.

Dilakukan dengan mengacu pada bagan alir dokumen yang ada di perusahaan.

2. Memahami kerja dari sistem yang ada (*Understand*)

Yaitu memahami kerja sisten dengan mempelajari secara rinci bagaimana sistem yang ada berjalan dengan cara melakukan penelitian. Tujuannya yaitu untuk mendeskripsikan sistem yang berjalan dengan penekanan pada area dimana masalah tersebut timbul.

Hasilnya yaitu Deskripsi sistem yang berjalan. Langkah-Langkahnya adalah sebagai berikut :

- a. Menentukan jenis penelitian
- b. Merencanakan jadwal penelitian
- c. Membuat penugasan penelitian
- d. Membuat agenda wawancara
- e. Mengumpulkan hasil penelitian

3. Menganalisis Sistem (*Analyze*)

Dilakukan berdasarkan data yang telah di peroleh dari hasil penelitian yang dilakukan :

- a. Menganalisis kelemahan sistem
- b. Menganalisis kebutuhan informasi pegawai

4. Membuat laporan hasil analisis (*Report*)

Proses pelaporan dilakukan setelah menganalisa hasil-hasil penelitian, yang meliputi :

- a. Daftar masalah yang ditemukan oleh analisis sistem.
- b. Suatu pernyataan asumsi penting yang dibuat oleh analisis sistem selama pelaksanaan analisa sistem.
- c. Alasan yang mendasari dan luas analisa sistem yang dilaksanakan.
- d. Rekomendasi yang bersangkutan dengan sistem yang telah diusulkan dengan persyaratan-persyaratan yang harus

dipenuhi oleh sistem yang diusulkan tersebut.

- e. Proyeksi sumber daya yang diperlukan dan biaya yang dibutuhkan dalam perancangan sistem baru.

2.3 Alat Bantu Analisis Sistem

Alat bantu analisa yang digunakan yaitu : *Flow Of Document*.

Merupakan bagan alir yang menunjukkan arus dari formulir dan laporan termasuk tembusan-tembusannya dengan menggunakan simbol-simbol yang sama dengan digunakan didalam bagan alir system. Berikut ini adalah simbol-simbol bagan alir system :

Tabel 2.1. Tabel Bagan Alir Sistem

Simbol	Keterangan
	DOKUMEN Menunjukkan dokumen input atau output baik untuk proses manual atau komputer.
	KEGIATAN MANUAL Menunjukkan pekerjaan manual.
	SIMPANAN Menunjukkan pengarsipan file.
	KARTU PLONG Menunjukkan input atau output yang menggunakan kartu plong.
	PROSES Menunjukkan operasi kegiatan proses dari operasi program komputer.
	OPERASI LUAR Menunjukkan operasi yang dilakukan diluar komputer.
	PENGLURUTAN OFFLINE Menunjukkan proses pengurutan data diluar

	proses komputer.
	PITA MAGNETIC Menunjukkan input atau output menggunakan pita magnetik.
	HARDDISK Menunjukkan input atau output menggunakan harddisk.
	DISKETTE Menunjukkan input output menggunakan disket.
	PITA KERTAS BERLUBANG Menunjukkan input output menggunakan pita kertas berlubang.
	KEYBOARD Menunjukkan input yang menggunakan on line keyboard.
	DISPLAY Menunjukkan output yang ditampilkan di monitor.

	PENGHUBUNG Menunjukkan penghubung ke halaman yang masih sama. Menunjukkan penghubungan beda halaman.
	PROSES ALIH Simbol garis alir menunjukkan arus dari proses.
	DRUM MAGNETIC Menunjukkan input output menggunakan disket.
	KOMENTATOR Menunjukkan keterangan atau komentator untuk memperjelas maksud isi dari simbol flow chard satu dengan yang lain

Sumber : Jogiyanto HM, 2009

2.4 Tahap Perancangan Sistem (Desain Sistem)

2.4.1 Pengertian Perancangan Sistem

Menurut Jogiyanto HM, (2009), desain sistem dapat diartikan sebagai berikut :

1. Tahap setelah analisis dari siklus pengembangan sistem.
2. Pendefinisian dari kebutuhan-kebutuhan fungsional.

3. Persiapan untuk rancang bangun.
4. Menggambarkan bagaimana suatu sistem dibentuk yang berupa penggambaran, rancangan dan pembuatan sketsa atau pengaturan dari beberapa elemn yang terpisah kedalam satu kesatuan yang utuh.

Tahap perancangan sistem ini mempunyai 2 (dua) tujuan utama yaitu :

1. Untuk memenuhi kebutuhan pemakai sistem.
2. Untuk memberikan gambaran yang jelas rancang bangun yang lengkap kepada pemrogram komponen yang terlibat.

Tujuan kedua diatas lebih condong pada desain sistem yang terinci, yaitu pembuatan rancang bangun yang jelas dan lengkap untuk nantinya digunakan untuk pembuatan program komputernya. Untuk mencapai tujuan ini, analis sistem harus dapat mencapai sasaran-sasaran sebagai berikut :

1. Desain sistem harus berguna, mudah dipahami dan nantinya muda digunakan. Ini berarti data harus mudah ditangkap, metode-metode harus mudah diterapkan dan

informasi harus mudah dihasilkan serta mudah dipahami dan digunakan.

2. Desain sistem harus dapat mendukung tujuan utama perusahaan sesuai dengan yang telah didefinisikan pada tahap perencanaan sistem yang dilanjutkan pada tahap analisis sistem.

Adapun langkah-langkah dari perancangan sistem yaitu :

1. Menentukan entitas yang akan digunakan
Entitas adalah objek, jadi dalam sebuah perancangan haruslah jelas siapa saja yang memberi data dan menerima informasi.
2. Menentukan tabel-tabel yang akan digunakan dalam perancangan. Untuk kelancaran dari sebuah sistem informasi yang dihasilkan maka penentuan tabel haruslah cermat dan harus dapat menghemat memori.
3. Menentukan relasi antar tabel sehingga antara kunci tabel yang satu dengan tabel yang lain saling berkaitan. Hal ini berguna untuk pencarian data dari beberapa tabel.

2.4.2 Alat Bantu Perancangan Sistem

2.4.2.1 Context Diagram

Context diagram adalah *Data Flow Diagram* (DFD) tingkat atas, yaitu diagram yang paling tidak detail dari sebuah sistem informasi yang menggambarkan aliran-aliran data ke dalam dan ke luar sistem dan ke dalam dan keluar eksternal.

Context diagram mempunyai sejumlah karakteristik dalam sistem, yaitu :

- a. Kelompok pemakai, organisasi atau sistem lain dimana sistem melakukan komunikasi (sebagai terminator).
- b. Data masuk, yaitu data yang diterima sistem dari lingkungan dan harus diproses dengan cara tertentu.
- c. Data keluar, yaitu data yang dihasilkan sistem dan diberikan ke dunia luar.
- d. Penyimpanan data (*storage*), yaitu digunakan secara bersama antara sistem

dengan terminator. Data ini dapat dibuat oleh sistem dan digunakan oleh lingkungan atau sebaliknya dibuat oleh lingkungan dan digunakan oleh sistem. Hal ini berarti pembuatan simbol data *storage* dalam CD dibenarkan, dengan syarat simbol tersebut.

- e. Batasan antara sistem dan lingkungan.

Tabel 2.2 :Tabel Simbol-Simbol Context Diagram

Simbol	Nama Simbol dan Keterangan
	<p>TERMINATOR</p> <p>Menggambaran asal data atau tujuan data.</p>
	<p>PROSES</p> <p>Menggambaran entitas atau proses aliran data masuk yang ditransformasikan ke aliran data.</p>
	<p>ALIRAN DATA</p> <p>Menggambaran aliran data atau informasi dari atau ke system.</p>
	<p>PENYIMPANAN</p> <p>Dapat digunakan untuk mendefinisikan basis data atau seringkali mendefinisikan bagaimana penyimpanan di implementasikan dalam sistem komputer.</p>

Sumber : Jogiyanto HM, 2009

2.4.2.2 Data Flow Diagram (DFD)

DFD adalah Diagram yang menggambarkan suatu sistem yang telah ada atau sistem baru

yang akan dikembangkan secara logika, tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir atau dimana data tersebut akan disimpan. Hal yang harus diperhatikan dalam menggambarkan diagram alir :

- a. Bagan alir sebaiknya digunakan dari atas ke bawah mulai dari bagian kiri suatu halaman.
- b. Kegiatan didalam bagan alir harus ditunjukkan dengan jelas.
- c. Harus ditunjukkan dimana kegiatan dimulai dan dimana kegiatan berakhir.
- d. Masing-masing kegiatan didalam suatu bagan alir sebaiknya digunakan suatu kata untuk mengawali suatu kegiatan.
- e. Gunakan simbol-simbol bagan alir dalam *Context Diagram*. Simbol yang digunakan dalam DFD

Leveled sama dengan simbol

dalam *Context Diagram*.

Tabel 2.3 :Tabel Simbol-Simbol
Data Flow Diagram (DFD)

Simbol	Nama Simbol dan Keterangan
	KESATUAN LUAR Memberikan input atau menerima output dari sistem.
	PROSES Menggambarakan logika dari hasil suatu arus data yang masuk ke dalam proses ,untuk menghasilkan arus data yang akan keluar dari proses.
	ALIRAN DATA Menggambarakan aliran data atau informasi dari satu ke system.
	SIMPAN DATA Dapat digunakan sebagai sasaran untuk mengumpulkan data, penyimpan data ini direpresentasikan dengan dua garis paralel, penyimpanan data ini biasanya dihubungkan dengan penyimpanan file-file database.

2.4.3 Tahap Perancangan Basis Data

Basis data merupakan kumpulan data yang saling berhubungan satu dengan lainnya, yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redundansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan .

Penggunaan basis data untuk mengelola data mempunyai beberapa keuntungan yaitu :

1. Kebebasan data dan akses efisien.
2. Mereduksi waktu pengembangan aplikasi.
3. Integritas dan keamanan data.

4. Administrasi keseragaman data.

5. Akses kebersamaan dan perbaikan dari terjadinya *crashes* (tabrakan dari proses serentak).

Alat bantu perancangan basis data :

2.4.3.1 Entity Relationship Diagram (ERD)

Simbol	Nama Simbol
	ENTITY Digunakan untuk menggambarakan objek. Yang dapat didefinisikan dalam lingkaran pemakai.
	ATRIBUT Digunakan untuk menggambarakan elemen-elemen dari suatu entity, yang menggambarakan karakter entity.
	HUBUNGAN RELASI Entity dapat berhubungan satu sama lain. Hubungan ini disebut relationship.
	GARIS Digunakan untuk menghubungkan entity dan entity dengan atribut.

Model entity-relationship yang berisi komponen-komponen himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan atribut-atribut yang mempresentasikan seluruh fakta dari dunia nyata yang ditinjau, dapat digambarkan dengan lebih sistematis dengan menggunakan *Digram Entity-Relationship* (Diagram E-R).

Adapun simbol-simbol yang digunakan dalam ERD (*Entity*

Relationship Diagram) adalah sebagai berikut

Tabel 2.4 : Tabel Simbol ERD (*Entity Relationship Diagram*)

Simbol	Nama Simbol
	ENTITY Digunakan untuk menggambarkan objek. Yang dapat didefinisikan dalam lingkaran pemakai.
	ATRIBUT Digunakan untuk menggambarkan elemen-elemen dari suatu entity, yang menggambarkan karakter entity.
	HUBUNGAN / RELASI Entity dapat berhubungan satu sama lain. Hubungan ini disebut relationship.
	GARIS Digunakan untuk menghubungkan entity dan entity dengan atribut.

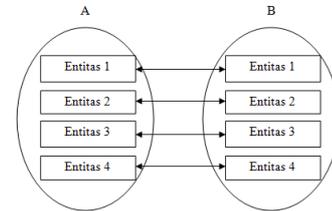
2.5 Kardinalitas atau Derajat Relasi

Kardinalitas relasi menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain. Kardinalitas yang terjadi diantara dua himpunan entitas (misal A dan B) dapat berupa :

1. Satu ke Satu (*One to One*).

yang berarti setiap entitas pada himpunan entitas A dapat berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas B, dan begitu juga sebaliknya, setiap entitas pada himpunan entitas B

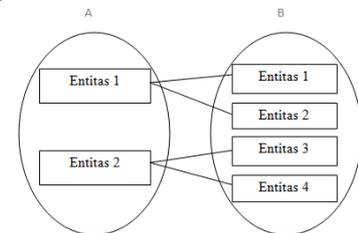
berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas A.



Gambar 2.1 : Kardinalitas relasi satu ke satu

2. Satu ke Banyak (*One to Many*).

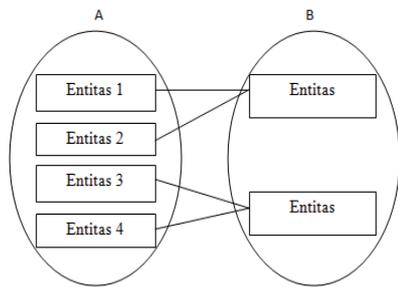
Yang berarti setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi tidak sebaliknya, dimana setiap entitas pada himpunan entitas B berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas A.



Gambar 2.2 : Kardinalitas relasi satu ke banyak

3. Banyak ke Satu (*Many to One*).

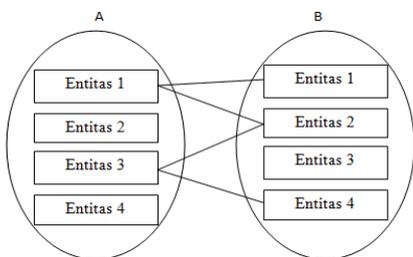
Setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas B, tetapi tidak sebaliknya, dimana setiap entitas pada himpunan entitas A berhubungan dengan paling banyak satu entitas pada himpunan entitas B.



Gambar 2.3 : Kardinalitas relasi banyak ke satu

4. Banyak ke Banyak (*Many to Many*).

Yang berarti setiap entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, dan demikian juga sebaliknya, dimana setiap entitas pada himpunan B dapat berhubungan dengan banyak entitas pada himpunan entitas A.



Gambar 2.4 : Kardinalitas relasi banyak ke banyak

2.6 Implementasi Tabel

Aturan umum dalam pemetaan Model Data (*Level Konseptual* dalam Abstraksi Data) yang kita gambarkan dengan Diagram E-R menjadi basis data fisik adalah :

1. Setiap himpunan entitas akan diimplementasikan sebagai sebuah table (*file data*).



Tabel Barang

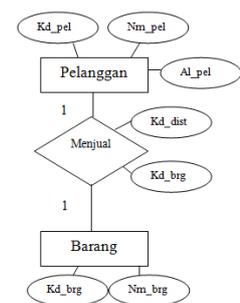
Tabel 2.5 : Implementasi tabel pada

ERD

Kd_brg	Nm_brg	Jml_brg	Satuan

1. Relasi dengan derajat relasi

1-1 (*one to one*) yang menghubungkan dua buah himpunan entitas akan dipresentasikan dalam bentuk penambahan atau penyertaan atribut-atribut relasi ke tabel yang mewakili salah satu dari kedua himpunan entitas.



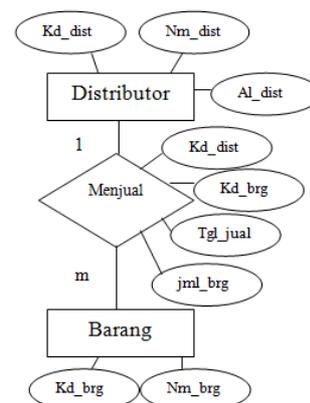
Tabel Distributor

Kd_dist	Nm_dist	Al_dist

Tabel Barang

Kd_brg	Nm_brg	Kd_dist

2. Relasi dengan derajat relasi 1-M (*one to many*) yang menghubungkan dua buah himpunan entitas juga akan dipresentasikan dalam bentuk pemberian atau pencantuman atribut kunci dari himpunan entitas pertama (yang berderajat 1) ke table yang mewakili himpunan entitas kedua (yang berderajat M). Atribut kunci dari himpunan pertama ini menjadi atribut tambahan bagi himpunan entitas kedua.



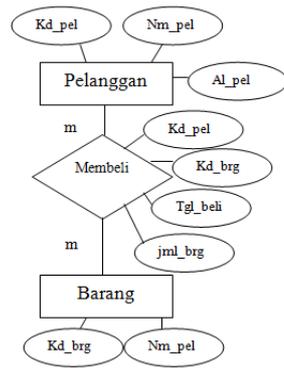
Tabel Distributor

Kd_dist	Nm_dist	Al_dist

Tabel Menjual

Kd_dist	Tgl_Jual	Kd_brg	Nm_brg	Jml_brg

3. Relasi dengan derajat relasi M-M (*many to many*) yang menghubungkan dua buah himpunan entitas akan diwujudkan dalam bentuk table khusus memiliki field yang berasal dari kunci-kunci dari himpunan entitas yang dihubungkannya.



Tabel Konsumen

Kd_pel	Nm_pel	Alm_pel

Tabel Membeli

Kd_pel	Kd_brg	Tgl_Beli	Jml_brg

Tabel barang

Kd_brg	Nm_brg

2.7 Ketergantungan Fungsional

- Sebuah tabel berisi paling sedikit 2 buah atribut, yaitu A dan B, notasinya $A \rightarrow B$. Yang berarti A secara fungsional menentukan B atau B secara fungsional tergantung pada A, hanya jika untuk setiap kumpulan baris data (*row*) yang ada pada tabel, pasti ada 2 baris (*row*) di tabel dengan nilai untuk A yang sama, maka nilai untuk B pasti juga sama.

Definisinya :

Diberikan 2 *row* r1 dan r2 dalam tabel dimana $A \rightarrow B$

Jika $r1(A) = r2(A)$, maka $r1(B) = r2(B)$

Pada data pelanggan atribut yang digunakan, yaitu Kd_pel, Nm_pel, Alm_pel, Telp_pel.

Berikut data pada table data pelanggan :

Tabel 2.6 : Ketergantungan Fungsional

	Kd_pel	Nm_pel	Alm_pel	Telp_pel
Row 1	A0001	PT.Jateng Top	Jl.Gajah 87	024-8769090
Row 2	A0002	PT.Astra Motor	Jl.Durian Raya 9	024-3547879
Row 3	A0003	PT.Yam.aha Motor	Jl.Brotojoyo 56	024-3567898

Maka ketergantungan fungsional yang dapat kita ajukan adalah :

- $Kd_pel \rightarrow Nm_pel$

Yang berarti bahwa atribut Nm_pel hanya tergantung pada atribut Kd_pel. Tanpa memperhatikan pengertian ketergantungan secara alamiah terhadap tabel tersebut, kita juga dapat mengajukan sejumlah ketidaktergantungan (non KF).

- $Nm_pel \not\rightarrow Alm_pel$

Yang artinya atribut Alm_pel tidak tergantung pada atribut Nm_pel.

c. Alm_pel ↘ Telp_pel

Yang artinya atribut Alm_pel tidak tergantung pada pada atribut Telp_pel..

3.4 Normalisasi Data

Proses normalisasi adalah suatu proses dimana elemen-elemen data dikelompokkan menjadi tabel-tabel, dimana didalam tabel tersebut terdapat entity-entity dan relasi antar entity tersebut.

Ada beberapa hal yang perlu diperhatikan dalam normalisasi suatu data yaitu:

1. *Field* atau Atribut Kunci.

Field kunci merupakan satu *field* atau set *field* yang terdapat dalam satu file yang merupakan kunci dan mewakili *record*. *Field* yang merupakan kunci akan menjadi penentu dalam pencairan program.

2. Macam-macam kunci :

a. *Candidat Key* (Kunci Calon)

Adalah satu atribut atau *field* yang mengidentifikasi secara unik dari suatu kejadian yang sifatnya khusus dari suatu entity.

b. *Primary Key* (Kunci Primer)

Adalah kunci kandidat yang dipilih untuk mewakili setiap kejadian dari suatu entity.

Kunci primer ini sifatnya unik, tidak mungkin sama dan tidak mungkin ganda.

c. *Alternate Key* (Kunci Alternatif)

Adalah kunci kandidat yang tidak dipakai sebagai kunci primer.

d. *Foregin Key* (Kunci Tamu)

Adalah kunci primer yang ditempatkan pada file lain dan biasanya menunjukkan dan melengkapi suatu hubungan antara file satu dengan file lainnya.

Tahap-tahap dalam normalisasi data adalah sebagai berikut :

1. Bentuk Tidak Normal

Adalah suatu bentuk dimana semua data dikumpulkan apa adanya tanpa mengikuti aturan-aturan tertentu. Bisa jadi data yang dikumpulkan akan tidak lengkap dan terjadi duplikasi data.

Contoh :

File Ob.at (Kd_brg, Nm_brg1, Nm_brg2).

Bentuk adalah bentuk tidak normal karena dalam file barang yang mempunyai Kd_brg, Nm_brg, Hrg_brg mengambil 2 nama barang, sehingga terjadi perulangan nama barang 2 kali.

Kd_brg	Nm_brg1	Nm_brg2
0001	Baut	Oli

0002	Lampu	Kamprat
------	-------	---------

2. Bentuk Normal Pertama

Bentuk normal pertama merupakan suatu bentuk dimana data yang dikumpulkan menjadi satu field yang sifatnya tidak akan berulang dan tiap field hanya mempunyai satu pengertian.

Contoh :

Dari tabel barang diatas diubah menjadi bentuk normal pertama.

Kd_brg	Nm_brg1
0001	Baut
0002	Lampu
0003	Oli
.0004	Kamprat

3. Bentuk Normal Kedua

Bentuk normal kedua adalah bentuk yang memenuhi syarat-syarat sebagai berikut:

- Sudah memenuhi kriteria bentuk normal pertama.
- Field yang bukan kunci tergantung secara fungsi pada kunci primer.

Contoh :

Dari file barang, dapat dilihat bahwa kunci primernya adalah field Kd_brg. Field Nm_brg tergantung pada field Kd_brg.

4. Bentuk Normal Ketiga

Bentuk normal ketiga adalah suatu bentuk yang memenuhi syarat-syarat:

- Relasi antar file sudah merupakan bentuk normal kedua.
- Field yang bukan kunci tergantung secara fungsi pada kunci primer.

Contoh :

File Barang dan file Distributor sudah merupakan bentuk normal ketiga karena seluruh file yang bukan kunci sudah tergantung pada field kunci yaitu kunci primernya.

3.5 Data Dictionary (DD)

Kamus data atau *Data Dictionary* (selanjutnya disebut DD) adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. (Jogiyanto HM, 2005).

DD dapat mengidentifikasi data yang mengalir di sistem dengan lengkap. Dalam perancangan sistem kamus data di gunakan untuk merancang input, laporan-laporan dan database.

Tabel 2.7. Tabel *Data*

Dictionary

Notasi	Arti
=	Terbentuk dari atau terdiri dari atau sama dengan
+	And
[]	Salah satu dari (memilih salah satu dari elemen-elemen data di dalam kurung bracket ini)
	Sama dengan simbol []
N { } M	Interaksi (elemen data dalam kurung brace berinteraksi mulai N kali dan maksimum M kali)
()	Optimal (elemen data di dalam kurung parenthesis sifatnya optional, dapat ada dan dapat tidak ada)
*	Keterangan setelah tanda ini adalah komentar

Sumber : Jogiyanto HM, 2009

3.6 Perancangan *Input Output Design*

Yaitu untuk menggambarkan tampilan program input maupun output.

1. Perancangan Masukan (*Input Design*)

Dimulai dari dasar sebagai penangkap input pertama kali, karena bila dokumen dasar tidak di desain dengan baik, kemungkinan input yang tercatat dapat salah bahkan kurang,. Dokumen dasar biasanya terbentuk formulir yang digunakan untuk menangkap (*capture*) data yang terjadi (Jogiyanto HM, 2005).

2. Perancangan Keluaran (*Ouput Design*)

Dimaksudkan untuk menentukan kebutuhan ouput dari sistem yang baru. (Jogiyanto HM, 2009).

3.7 Tahap Implementasi Sistem

Implementasi sistem merupakan tahap meletakkan sistem supaya siap untuk dioperasikan. Tahap implementasi sistem terdiri dari langkah-langkah sebagai berikut :

1. Menerapkan Rencana Implementasi

Merupakan kegiatan awal dari tahap implementasi sistem. Rencana implementasi dimaksudkan terutama untuk mengatur biaya dan waktu yang dibutuhkan selama tahap implementasi. Dalam rencana implementasi semua biaya yang akan dikeluarkan untuk kegiatan implementasi perlu dianggarkan dalam bentuk anggaran biaya. Waktu yang diperlukan untuk kegiatan implementasi juga perlu diatur dalam rencana implementasi dalam bentuk skedul waktu.

2. Melakukan Kegiatan Implementasi

Kegiatan implementasi dilakukan dengan dasar kegiatan yang telah direncanakan dalam rencana implementasi. Kegiatan-kegiatan yang dapat dilakukan dalam tahap implementasi ini adalah :

1. Pemilihan dan pelatihan personil
2. Pemilihan tempat dan instalasi perangkat keras dan perangkat lunak

3. Pemrograman dan pengetesan program
 4. Pengetesan sistem
 5. Konversi sistem.
3. Tindak Lanjut Implementasi

Tindak lanjut implementasi merupakan pengetesan penerimaan sistem (*system acceptance test*). Pengetesan ini dilakukan dengan menggunakan data sesungguhnya dalam jangka waktu tertentu yang dilakukan oleh analisis sistem bersama-sama dengan *user*.

3.8 Visual Foxpro 9.0

2.12.1 Pengenalan Microsoft Visual Foxpro

Visual foxpro merupakan *database management system* (aplikasi pengolah database) atau disingkat DBMS yang merupakan pengembangan lebih lanjut dari foxpro 2.x. Kelebihan visual foxpro dibandingkan foxpro 2.x terletak pada kemampuannya untuk mendukung *object oriented programming* (pemrograman berorientasi objek) atau disingkat OOP.

OOP merupakan program yang diciptakan menjadi efisien, terstruktur dengan baik dan yang terpenting mampu memanfaatkan kembali kelas-kelas yang sudah diciptakan sebelumnya untuk digunakan dalam program lain. Dengan

kemampuan ini pembuatan program menjadi lebih singkat dan mudah.

Manfaat OOP paling dirasakan untuk *graphical user interface* (pemrograman berorientasi grafik) atau disingkat GUI. Berorientasi grafis berarti bahwa program yang dibuat akan mempunyai tampilan seperti program-program *windows*. Masing-masing objek grafis yang ada pada layar (kotak teks, jendela, *spinner* atau tombol perintah) dapat di representasikan sebagai objek visual foxpro dengan hanya mengambil definisi objek tersebut dari kelas template bawaan visual foxpro. Puluhan kelas template telah disediakan oleh visual foxpro yang dapat digunakan untuk membuat tombol perintah, kotak edit, kotak cek dan objek visual lainnya.

Visual foxpro dapat juga menyediakan sejumlah *developer application* (aplikasi pengembang) untuk membantu mengembangkan program, antara lain berupa aplikasi *tabel designer, database designer, form designer, menu designer, report designer* dan *project manager*. Dengan aplikasi-aplikasi tersebut program dapat diciptakan tanpa banyak

melakukan pemrograman kode baris demi baris yang sangatlah menyita waktu.

2.12.2 Developer Application (Aplikasi-Aplikasi Pengembang)

Microsoft visual fo.xpro sebenarnya bukan merupakan suatu perangkat lunak tunggal, melainkan terdiri dari sejumlah *developer application* (aplikasi pengembang) yang digabungkan menjadi satu. Dibawah ini penjelasan singkat mengenai masing-masing aplikasi pengembang yang terdapat pada Microsoft Visual Foxpro :

✓ *Database Designer*

Database designer merupakan aplikasi Visual Foxpro yang berfungsi untuk mendesain database. Dalam visual foxpro, database adalah gabungan dari beberapa tabel (*file-file* yang berekstensi.DBF) yang berelasi atau berhubungan satu nama lain.

✓ *Tabel Designer*

Tabel designer merupakan aplikasi visual foxpro yang berguna untuk merancang tabel. Dengan tabel *designer* dapat dibuat struktur field dari suatu tabel, menspesifikasikan indeks tabel tersebut dan seterusnya.

Tabel dalam visual foxpro dapat berbentuk tabel bebas yang berdiri sendiri maupun tabel yang berada di dalam suatu *file database* (dalam foxpro *file database* berekstensi .DBC).

✓ *Query Designer*

Query designer merupakan aplikasi visual foxpro yang berfungsi untuk menyusun *query*. *Query* adalah seperangkat aturan yang menspesifikasikan .cara mengambil informasi yang terkandung dalam sebuah atau beberapa tabel. *File-file query* disimpan dengan ekstensi .QPR. Penting untuk diingat bahwa hasil dari sebuah *query* tidak bisa diedit.

✓ *View Designer*

View designer berfungsi untuk membuat *view*. *View* mirip dengan *query* tetapi mempunyai satu kelebihan yaitu selain dapat menampilkan isi tabel sumber, *view* juga dapat mengedit tabel sumber. *View* disimpan dengan nama file yang berekstensi .VUE atau disimpan dengan *file database* (.DBC). *View*

diperlakukan seperti tabel oleh visual foxpro dengan sedikit perbedaan. Jika tabel biasa merupakan se.buah data yang sifatnya fisik (artinya yang tersimpan, itulah yang ditampilkan), maka *view* merupakan sebuah sumber data logical, artinya isi tabel yang ditampilkan tergantung pada aturan-aturan yang dispesifikasikan oleh *view* bersangkutan. Jadi, *view* dapat dibuka dengan perintah USE, dapat ditampilkan dengan perintah BROWSE dan sebagainya.

✓ *Form Designer*

Form designer berfungsi untuk merancang tampilan program atau form langsung dilayar tanpa harus melakukan pemrograman baris demi baris seperti pada foxpro biasa. Hasil rancangan form ini disimpan dalam sebuah file yang berekstensi .SCX.

✓ *Menu Designer*

Menu designer digunakan untuk merancang menu. Hasil rancangan menu dapat disimpan ke dalam berekstensi .MNX.

✓ *Class Designer*

Class designer digunakan untuk menciptakan kelas yang berhubungan dengan tampilan. Dengan class designer wujud fisik objek dari kelas yang dirancang dapat dilihat langsung di layar. Kelas yang diciptakan dapat disimpan ke dalam file berekstensi .VCX..

✓ *Program Editor*

Program editor digunakan untuk menciptakan file program. Meskipun visual foxpro sudah menyediakan berbagai class, seperti *class designer* dan *form designer* guna mengurangi kebutuhan untuk menulis program, tetapi sering program yang dibuat membutuhkan sedikit pemrograman agar aplikasi menjadi lebih hidup. Hasil penulisan program dapat disimpan ke dalam file berekstensi .PRG. File ini dapat dijalankan dan dapat pulan dikompilasi menjadi file .EXE.

✓ *Report Designer*

Report designer digunakan untuk menciptakan laporan. Dengan aplikasi ini dapat ditentukan jenis keluaran yang muncul disetiap

halaman cetak, jenis keluaran yang muncul disetiap pergantian baris dan seterusnya. Hasil rancangan dengan report designer dapat disimpan dalam file ekstensi .FRX serta dapat dicetak dengan perintah *REPORT FORM*.

✓ *Project Manajer*

Project manajer digunakan untuk mengorganisir aplikasi. Aplikasi yang terdiri dari sejumlah *tabel, view, query, database, form*, laporan, kelas dan program dapat diorganisir dalam sebuah proyek menggunakan project manajer. Hasil pengorganisasian dengan project manajer dapat disimpan dengan file berektensi .PRJ. File ini kemudian dapat dijalankan atau kompilasi ke dalam file .EXE untuk nantinya didistribusikan ke klien.

BAB III

METODE PENELITIAN

3.1 Obyek Penelitian

Dalam penyusunan tugas akhir ini dilakukan penelitian dibagian administrasi gudang pada HUDI MOTOR Karangrayung Grobogan,

yang beralamat di JL.Raya Karangrayung 86, Grobogan.

3.2 Jenis dan Sumber Data

Dalam usaha untuk mendapatkan data-data yang benar sehingga tercapai maksud dan tujuan penyusunan tugas akhir ini, penulis menggunakan metode pengumpulan data dari jenis data dengan cara sebagai berikut:

A. Data Primer

Data Primer yaitu data yang diperoleh secara langsung dapat dilakukan melalui wawancara.

Secara langsung dengan bagian yang terkait.

Data Primer dapat berupa:

- Wawancara dengan pimpinan tentang bagaimana sistem persediaan sparepart yang dijalankan pada HUDI MOTOR Karangrayung Grobogan
- Wawancara tentang sejarah berdirinya HUDI MOTOR Karangrayung Grobogan

B. Data Sekunder

Data sekunder yaitu data yang diperoleh secara tidak langsung yang dapat berupa catatan-catatan, laporan-laporan tertulis, dokumen-dokumen dan makalah-makalah serta daftar pustaka.

Data sekunder dapat berupa :

- Data sparepart
- Data pembelian dan penjualan sparepart

3.3 Metode Pengumpulan Data

Dalam penyusunan tugas akhir ini akan perlu adanya suatu metode tertentu yang akan digunakan dalam pengumpulan data yang diperoleh dengan cara sebagai berikut :

a. Studi Lapangan

Yaitu data yang secara langsung dapat diperoleh dengan meninjau dan mengamati secara langsung terhadap objek yang diteliti melalui :

1. Wawancara (*Interview*)

Yaitu metode pengumpulan data yang mengadakan tanya jawab langsung dengan responden sehingga dapat mengetahui masalah secara jelas.

2. Observasi (*Observation*)

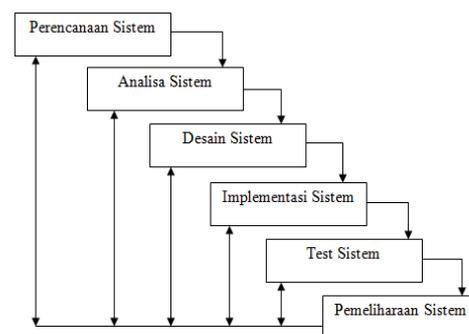
Yaitu pengumpulan data dengan pencatatan secara cermat terhadap objek yang diamati.

b. Studi Pustaka

Yaitu penelitian dengan menggunakan dan mempelajari buku-buku maupun literatur-literatur yang berhubungan dengan masalah yang diteliti sebagai landasan teori bagi penulis.

3.4 Metode Pengembangan Sistem

Metode pengembangan sistem menggunakan *System Development Life Cycle Model* (SDLC Model) atau juga terkenal dengan model *Waterfall*. Metode pengembangan *System Development Life Cycle Model* (SDLC Model) ini mengusulkan pendekatan pengembangan perangkat lunak yang sistematis dan sekuensial, metode ini didasarkan pada beberapa aktifitas berikut :



Gambar 3.1 : Pengembangan Sistem Model Waterfall

1. Perencanaan Sistem

Pengembangan sistem informasi dimulai dengan mengadakan penelitian terhadap elemen-elemen kebutuhan sistem bersangkutan dan mendefinisikan kebutuhan-kebutuhan tersebut dan menjabarkannya kedalam panduan bagi pengembangan sistem ditahap berikutnya. Aspek-aspek yang berkaitan berupa elemen-elemen yang berkaitan dengan sistem baik itu sumber daya manusia, peraturan perundang-undangan,

perangkat keras (*hardware*), prosedur kerja organisasi maupun beragam aspek lainnya, baik yang terkait secara langsung maupun tidak dengan sistem komputerisasi yang akan dibangun. Fase ini merupakan fase yang sangat penting (*essential*) untuk mendapatkan gambaran utuh sistem guna pengembangan sistem bersangkutan kedalam bentuk penerapan sistem yang berbasis komputerisasi.

2. Analisa Sistem

Tugas yang paling penting dalam tahap ini adalah proses menemukan permasalahan dan menghasilkan alternatif pemecah masalah. Dan diharapkan untuk memahami sistem yang ada serta menentukan kebutuhan-kebutuhan pemakai dan hambatan-hambatan pada suatu sistem baru.

3. Desain Sistem

Tahap setelah analisis dari siklus pengembangan sistem adalah desain sistem. Desain sistem menggambarkan bagaimana suatu sistem dibentuk yang dapat berupa penggambaran, perencanaan dan pembuatan sketsa. Tujuan dari desain sistem ini adalah memenuhi kebutuhan pemakai sistem serta memberikan gambaran yang jelas dan lengkap kepada pemrograman komputer dan ahli-ahli teknik lainnya yang terlibat.

4. Implementasi Sistem

Tahap implementasi sistem merupakan tahap meletakkan sistem supaya siap untuk dioperasikan. Tahap ini termasuk juga kegiatan menulis kode program dengan menggunakan bahasa pemrograman komputer yang telah ditentukan dalam tahap sebelumnya.

5. *Testing* (Uji Coba)

Setelah proses penulisan kode pemrograman langkah berikutnya berupa proses pengujian atau test sistem. Pengetesan sistem termasuk juga pengetesan program secara menyeluruh. Pengetesan sistem ini adalah untuk memastikan bahwa elemen-elemen atau komponen dari sistem telah berfungsi sesuai dengan yang diharapkan. Pengetesan dilakukan untuk mencari kesalahan-kesalahan atau kelemahan-kelemahan yang mungkin masih terjadi.

6. *Maintenance* (Pemeliharaan)

Fase ini merupakan fase perawatan terhadap sistem yang telah dikembangkan dan diimplementasikan. Cakupan fase ini berupa proses perawatan terhadap sistem yang berkaitan dengan perawatan berkala dari sistem maupun proses terhadap perbaikan sistem manakala sistem menghadapi kendala

dalam operasionalnya akibat masalah teknis dan non teknis yang tidak terindikasi dalam proses pengembangan sistem. Proses *maintenance* ini juga meliputi upaya-upaya pengembangan terhadap sistem yang telah dikembangkan sebelumnya dalam menghadapi mengantisipasi perkembangan maupun perubahan sistem bersangkutan.

3.5 Metode Analisis Data

Metode analisis data yang digunakan adalah Metode *Kualitatif*. Merupakan serangkaian observasi dimana tiap observasi yang terdapat dalam sample tergolong pada salah satu dari kelas-kelas yang eksklusif secara bersama-sama dan yang kemungkinan tidak dapat dinyatakan dalam angka-angka. Data-data yang telah diperoleh diolah dengan cara merancang sistem secara terperinci agar diperoleh alternative pemecahan masalah alat-alat yang digunakan, misalnya :

- HIPO (*Hirarchy Plus Input Proses Output*)
HIPO Sebenarnya adalah alat yang sekarang banyak digunakan sebagai teknik dokumen dalam siklus pengembangan sistem.
- DFD (*Data Flow Diagram*)
Merupakan gambaran sistem secara logical, gambaran ini tidak tergantung pada perangkat

keras, perangkat lunak, struktur data atau organisasi file.

- Kamus Data (*Data Dictionary*)
Merupakan gambaran dari seluruh atribut yang ada dan diambil secara langsung dari formulir atau slip yang ada pada objek penelitian, misalnya data tamu, data kamar dan sebagainya.
- ERD (*Entity Relational Diagram*)
Merupakan diagram yang menunjukkan hubungan antara entitas-entitas yang ada dalam suatu sistem entitas tentang apa data itu berbicara.

3.6 Implementasi Sistem

3.6.1 Tahap-Tahap Implementasi

Implementasi Sistem terdiri atas:

- a. Tahap Pengembangan
Pada tahap ini penulis mempergunakan bahasa pemrograman untuk implementasi dari logika sistem yang dirancang. Bahasa pemrograman yang digunakan adalah Visual Foxpro.
- b. Tahap Evaluasi
Untuk mengetahui adanya kesalahan pada logika yang dirancang maka penulis melakukan uji coba dan bila

terjadi kesalahan dilakukan perbaikan. Tahap ini meliputi :

1. Pemilihan dan penelitian *personil*
2. Pemilihan tempat dan instalasi perangkat lunak
3. Pengetesan program
4. Pengetesan sistem
5. Konversi sistem

3.6.2 Program dan Testing

a. Pemrograman

Pemrograman merupakan kegiatan menulis kode program yang akan dieksekusi oleh komputer. Kode program yang akan ditulis oleh programmer harus berdasarkan dokumentasi yang disediakan oleh analis *system* hasil dari desain secara rinci. Hasil program yang sesuai dengan yang dibutuhkan oleh pemakai sistem.

b. Testing

Sebelum program ditetapkan, maka program harus bebas terlebih dahulu dari kesalahan-kesalahan. Oleh karena itu program harus dites untuk menentukan kesalahan-kesalahan yang mungkin dapat terjadi.

Kesalahan dari program yang mungkin terjadi dapat di klasifikasi dalam bentuk tiga kesalahan, yaitu :

1. Kesalahan Bahasa (*Language Error*)

Adalah kesalahan di dalam penulisan source program yang tidak sesuai dengan yang telah diisyaratkan.

2. Kesalahan Sewaktu Proses (*Runtime Error*)

Kesalahan yang terjadi sewaktu executable program dijalankan.

3. Kesalahan Logika (*Logical Error*)

Kesalahan dari logika program yang dibuat.

3.6.3 Menu Design

Merupakan suatu tampilan program yang menampilkan menu utama dari suatu aplikasi yang diterapkan. Sistem menu merupakan komunikasi awal antara pengguna atau user dengan komputer.

3.6.4 Input Design

Merupakan suatu tampilan program yang menampilkan tampilan input yang berguna untuk memasukkan data yang diperlukan

atau akan diproses ke dalam suatu keluaran atau output yang dikehendaki.

3.6.5 *Output Design*

Merupakan suatu tampilan dari hasil proses yang ada yaitu tampilan yang keluar baik ke layar monitor maupun ke printer sebagai media keluaran.

3.6.6 *Training*

Pada tahap implementasi perlu dilakukan terhadap karyawan yang terkait dalam pelaksanaan sistem informasi pengolahan data inventori.

3.6.7 *Maintenance*

Setelah semua sistem terselesaikan, maka sistem membutuhkan perawatan supaya sistem dapat berjalan dengan baik dan data dapat tersimpan lebih aman.

Yang perlu dilakukan untuk merawat sistem adalah :

1. *Back Up* Secara Periodik

Back Up data ini dapat dilakukan dengan cara menyimpan ulang data pada disket atau pada CD. Data yang harus di *back up* misalnya data pasien, laporan keuangan, dan mungkin dapat dilakukan dalam satu bulan sekali.

2. Indeks Ulang (*Posting*)

Indeks ulang itu dapat dilakukan dengan cara pengurutan data sesuai dengan yang diinginkan (urut nomer, alphabet,urut tanggal).

3. *Update Database*

Update database dilakukan jika proses ini akan ditambahkan sistem baru, misalnya dalam sistem hanya terdapat *Delete, Add*, kemudian disesuaikan dengan kebutuhan ditambahkan *field* yang berfungsi untuk pencarian.

4. Menghapus data yang sudah tidak dipakai

Karena kapasitas komputer terbatas, maka untuk data-data yang sudah tidak terpakai harus dibuang.

BAB IV

ANALISIS DAN PERANCANGAN SISTEM

4.1 Tinjauan Umum Perusahaan

Bagian ini menjelaskan tentang, sejarah singkat HUDI MOTOR Karangrayung Grobogan, Struk tur Organisasi, Job Description.

4.1.1 Sejarah Singkat HUDI MOTOR Karangrayung Grobogan

Toko ini didirikan dengan nama HUDI MOTOR Karangrayung Grobogan yang bergerak dalam usaha penjualan spare part sepeda motor. HUDI MOTOR Karangrayung Grobogan ini didirikan pada tahun 2005 oleh Hudi Awang Nugroho, seorang pria kelahiran Semarang. Pada awalnya hanya seorang bengkel yang kecil dan spare part yang disediakan tidak lengkap. Usaha ini pernah mengalami kembang kempisnya dalam usaha dan pernah mengalami jatuhnya usaha, tetapi Hudi Awang Nugroho ini tidak pantang menyerah dan terus maju. Akhirnya dengan keberanian diri Hudi Awang Nugroho mengusahakan modal yang cukup besar untuk membangun usaha kembali.

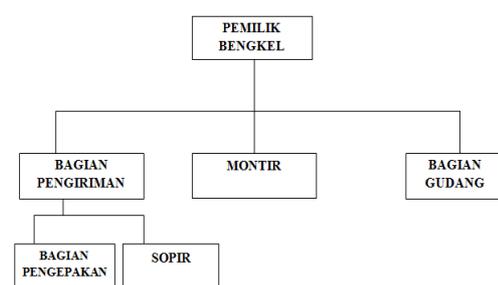
Akhirnya usaha yang didirikannya mengalami peningkatan dari tahun ke tahun modalnya bertambah dan sampai sekarang. Sehingga usaha yang didirikannya lebih meningkat dan mempunyai banyak pelanggan, baik pelanggan eceran maupun grosir.

Usaha ini didirikan pada daerah Purwodadi, karena di daerah sana usaha perbengkelan tersebut sangat langka kita

jumpai apalagi bengkel dengan spare part lengkap. Dengan terjangkaunya tempat usaha merupakan pendukung suksesnya dari usaha yang kita bangun. Sekarang bengkel ini mempunyai 15 montir dan 4 karyawan yang mendukung usaha ini.

4.1.2 Struktur Organisasi

Untuk dapat menjalankan kegiatan usaha dengan baik dan lancar suatu toko memerlukan suatu bentuk struktur organisasi toko yang baik. Struktur organisasi adalah hubungan antara komponen-komponen bagian dan posisi dalam HUDI MOTOR Karangrayung Grobogan maka perlu mempunyai suatu bentuk struktur organisasi yang dapat digambarkan sebagai berikut:



Gambar 4.1 : Bagan Struktur Organisasi

Sumber : HUDI MOTOR Karangrayung Grobogan

4.1.3 Job Description

Berdasarkan struktur organisasi diatas, maka berikut ini merupakan tugas dan wewenang dari masing-masing bagian :

A. Pemilik Bengkel

Mempunyai tugas dan wewenang :

1. Menentukan jumlah spare part yang harus di beli setiap periode.
2. Menentukan tugas dan tanggung jawab dari masing-masing bagian yang berada dibawahnya.
3. Mengatur jumlah biaya yang beli dan jual untuk jalannya usaha.
4. Memberikan gaji kepada karyawan sesuai dengan keahlian dan bidangnya.
5. Memberikan otorisasi pada setiap bagian, apabila bagian tersebut memberikan laporannya.
6. Memberikan arahan kepada setiap bagian sesuai dengan tugas dan

tanggung jawab masing-masing bagian.

7. Meminta laporan-laporan yang seharusnya ada dari bagian tertentu.

B. Bagian Gudang

Mempunyai tugas dan wewenang :

1. Mencatat data spare part yang dibeli dan dijual.
2. Menentukan stok minimal untuk setiap spare part.
3. Membuat laporan spare part, laporan spare part beli dan laporan spare part jual untuk kemudian diserahkan ke pimpinan bengkel.

C. Bagian Pengepakan

Mempunyai tugas dan wewenang :

1. Membaca setiap pesanan yang akan dikirim.
2. Melakukan pembungkusan atau pengepakan spare part yang akan dikirim.

D. Sopir

Mempunyai tugas dan wewenang :

1. Melakukan pengiriman ke tempat tujuan.
2. Melakukan laporan pengiriman spare part yang telah dikirim.

4.2 Pengertian Rancangan Sistem

Pengertian dari rancangan sistem adalah merancang suatu bentuk sistem agar dapat mempermudah dalam pembuatan suatu program. Rancangan sistem ini dapat kita buat dengan cara terlebih dahulu mengkaji data-data yang ada dalam suatu organisasi dari kegiatan ini biasanya disebut studi kelayakan terhadap organisasi.

Dapat dikatakan bahwa suatu rancangan adalah membuat perancangan program tersebut dapat seefisien mungkin, di dalam pelaksanaannya selalu melibatkan beberapa unsur yang menjadi dasar pembuatan sistem satu dengan yang lainnya seperti : *Desain System, Desain Database, Desain Input dan Desain Output* yang berupa laporan.

4.2.1 Rancangan Sistem Aplikasi Komputer yang Dirancang (New System)

Penulis hanya melengkapi pengolahan data administrasi dari manual ke komputerisasi dan juga merubah format laporan yang sudah ada. Perubahan yang ada penulis buat dengan sesederhana mungkin untuk memudahkan user memahami informasi yang dihasilkan oleh sistem baru. Dengan sistem yang baru kita dapat mengetahui berapa banyak administrasi penjualan yang terjadi setiap harinya.

Sebelum kita menguraikan lebih jauh tentang sistem yang baru, penulis akan uraikan alur sistem dan data barang dari pembelian, penjualan sampai laporan-laporan yang dirancang.

a. Distributor dan Program Pembelian

Meski Distributor dan Pembelian Barang guna penambahan stok tidak menjadi bahan permasalahan utama yang penulis mencoba membuat bagian tersebut menjadi pelengkap yang baik dan perlu disertakan. Guna menambahkan stok yang berasal dari transaksi pembelian secara garis besar operator hanya perlu memasukkan kode distributor serta kode-kode barang yang dibeli berikut

jumlah dan harga satuannya, apabila selesai maka stok akan secara otomatis bertambah.

b. Pembeli dan Program Penjualan

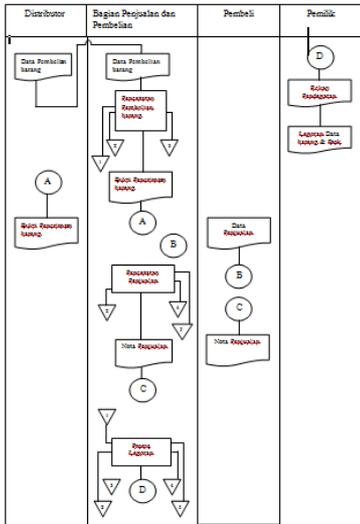
Program penjualan adalah salah satu inti masalah yang penulis angkat dalam penyusunan TA ini. Untuk melayani pembeli yang beragam maka penulis mencoba membuat program penjualan sedemikian rupa sehingga memudahkan operator dalam mengoperasikannya. Sedangkan untuk pelanggan penulis telah menyediakan file khusus untuk pelanggan dengan menggunakan kode untuk masing-masing pelanggan.

- c. Dalam kaitan prgram penjualan, penulis mencoba menerapkan sistem FIFO (*First In First Out*) seperti yang diminta oleh perusahaan. Dalam suatu pembelian dengan pelayanan sistem FIFO ini bisa mengakibatkan harga jual yang berbeda, dikarenakan jika ada perubahan harga pembelian baik naik maupun turun maka harga

jual memakai harga baru tetapi barang yang dikeluarkan tetap barang lama.

- d. Laporan-laporan dan menu. Laporan-laporan yang penulis buat hanya sebagian dari seluruh sistem yang penulis buat. Pada intinya laporan yang penulis buat hanya meliputi laporan posisi stok serta laporan penjualan guna mempermudah pengecekan barang keluar serta posisi stok yang ada apabila pimpinan menginginkan stok obnam secara mendadak. Sedangkan untuk menu penulis membuat sesederhana mungkin agar supaya operator dapat dengan segera memahami kegunaan-kegunaannya.
- e. Re-index. Dalam penyusunan script menggunakan Foxpro 9.0 memang dirasakan sempurna untuk kebutuhan database.

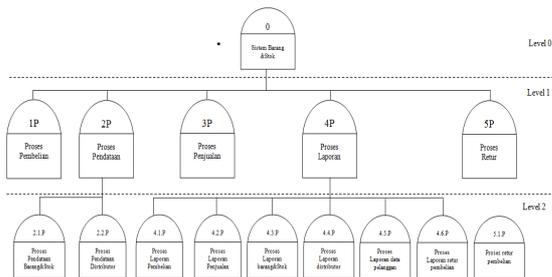
4.3 Flow Of Document (FOD) New System



.Gambar 4.2.FOD New System

4.4 Bagan Berjenjang.

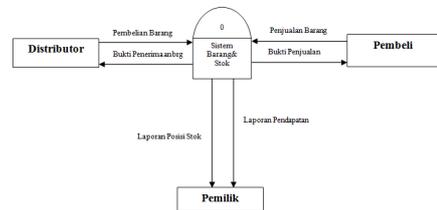
Bagan berjenjang atau disebut juga hirachi chart digunakan untuk mempersiapkan penggambaran DFD dari level 0 (nol) ke level-level yang lebih bawah. Bagan berjenjang dapat digambarkan dengan menggunakan notasi proses yang digunakan DFD. Untuk sistem stok dan penjualan sistem .FIFO ini proses-proses akan tampak seperti bagan berjenjang dibawah ini :



.Gambar 4.3. Bagan Berjenjang

4.5 Data Flow Diagram (DFD)

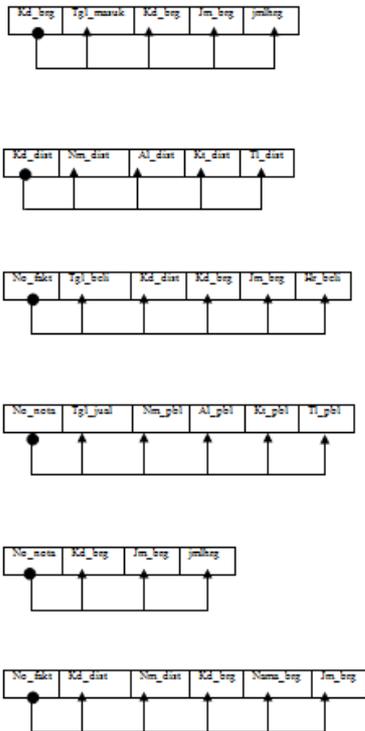
DFD digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik data tersebut mengalir dan akan disimpan. DFD Pengolahan Data Penjualan Sistem FIFO sebagai berikut :



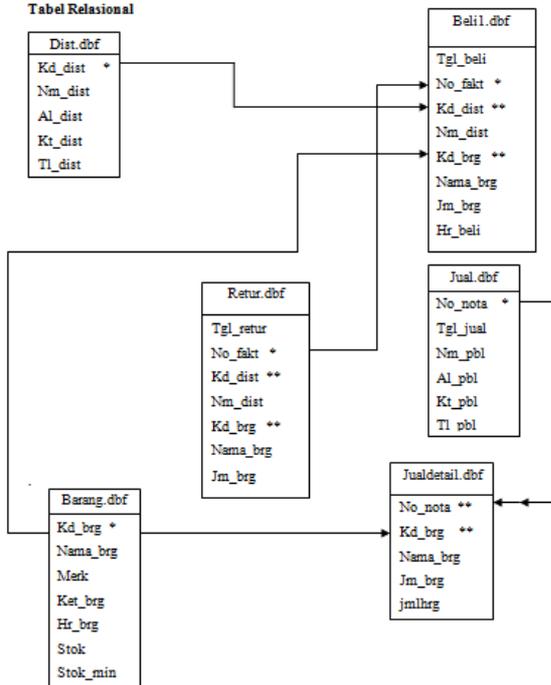
Gambar 4.4. Contex Diagram

Keterangan :

Pembelian barang dilakukan apabila stok habis atau transaksi besar yang tidak melebihi jumlah stok yang ada. Untuk memudahkan proses pembelian penulis menyediakan file khusus yang menyimpan data-data distributor yang telah diberi kode. Pembeli eceran maka kode tidak perlu dimasukkan saat penginputan transaksi komputer.



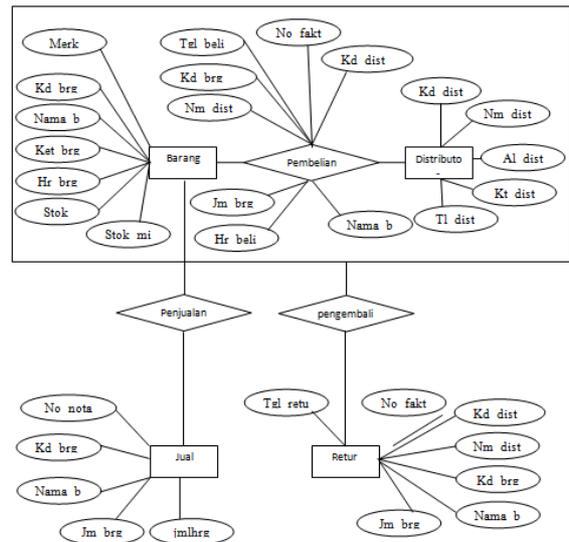
Tabel Relasional



Gambar 4.7. Bentuk Tabel Relasional

4.7 Entity Relationship Diagram (ERD)

Entity relationship diagram menggambarkan bentuk hubungan (relasi) antar file database yang satu dengan file database yang lain. Adapun bentuk ERD persediaan stok sparepart yang penulis buat adalah :



Gambar 4.8 Entity Relationship Diagram

4.8 Kamus Data

Kamus data (KD) atau *Data Dictionary* (DD) adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Pada tahap perancangan sistem, KD digunakan untuk merancang input, laporan dan database. KD dibuat berdasarkan arus data yang ada di DFD yang sifatnya global, sedangkan kete.ran.gan lebih lanjut tentang struktur dari arus data DFD diatas adalah berikut :

Tabel 4.1 Kamus data pembelian barang

Nama arus data	: Pembelian Barang		
Alias	: Data Pembelian		
Bentuk data	: Dokumen, file		
Arus data	: Distributor → P1 dan P1 → F.Beli		
Penjelasan	: Mencatat data pembelian barang		
Periode	: Saat transaksi pembelian		
No	Nama	Type/Lebar	Keterangan
1	Tgl_beli	D/8	Tanggal pembelian
2	No_Fakt	C/5	Nomor bukti faktur pembelian
3	Kd_dist	C/4	Kode distributor
4	Nm_dist	C/15	Nama distributor
5	Kd_brg	C/4	Kode barang
6	Nm_brg	C/15	Nama barang
7	Jm_beli	N/3	Jumlah barang yang dibeli
8	Hr_beli	N/8	Harga beli barang

Tabel 4.2. Kamus data barang

Nama arus data	: Data Barang & Stok		
Alias	: Kode-kode Barang		
Bentuk data	: File		
Arus data	: D2 → P1, P2 → D2, D2 → P3 dan Distributor → P2		
Penjelasan	: Menetapkan/membaca data barang berdasarkan kode barang		
Periode	: Sewaktu transaksi pembelian/penjualan atau pendataan		
No	Nama	Type/Lebar	Keterangan
1	Kd_brg	C/4	Kode barang
2	Nama_brg	C/15	Nama barang
3	Merk	C/15	Merk barang
4	Ket_brg	C/25	Keterangan barang
5	Hr_brg	N/8	Harga Barang
6	Stok	N/4	Stok barang
7	Stok_min	N/4	Stok minimal barang

Tabel 4.3. Kamus data distributor

Nama arus data	: Data Distributor		
Alias	: Kode Distributor		
Bentuk data	: File		
Arus data	: D4 → P2 dan D4 → P1		
Penjelasan	: Menetapkan/membaca kode distributor		
Periode	: Saat transaksi pembelian atau saat pendataan		
No	Nama	Type/Lebar	Keterangan
1	Kd_dist	C/4	Kode distributor
2	Nm_dist	C/15	Nama perusahaan distributor
3	Al_dist	C/25	Alamat distributor
4	Kt_dist	C/15	Kota distributor
5	Tl_dist	C/15	Nomor distributor

Tabel 4.4. Kamus Penjualan

Nama arus data	: Penjualan Barang		
Alias	: Data Penjualan		
Bentuk data	: File		
Arus data	: Pembeli → P3, P3 → D5 dan D5 → P4		
Penjelasan	: Memproses data penjualan barang		
Periode	: Saat transaksi penjualan atau saat pembuatan laporan		
No	Nama	Type/Lebar	Keterangan
1	Tgl_jual	D/8	Tanggal transaksi penjualan
2	No_nota	C/5	Nomor nota / bukti penjualan
3	Nm_pbl	C/15	Nama pembeli
4	Al_pbl	C/15	Alamat pembeli
5	Kt_pbl	C/15	Kota pembeli
6	Tl_pbl	C/15	Nomor telepon pembeli

Tabel 4.5. Kamus data Penjualan Detail

Nama arus data	: Detil Data Penjualan		
Alias	: Data Penjualan secara terperinci		
Bentuk data	: File		
Arus data	: P3 → D6 dan D6 → P4		
Penjelasan	: Menyimpan atau membaca detil data penjualan		
Periode	: Saat transaksi penjualan atau saat pembuatan laporan		
No	Nama	Type/Lebar	Keterangan
1	No_nota	C/5	Nomor nota / bukti penjualan
2	Kd_brg	C/4	Kode barang
3	Nama_brg	C/15	Nama Barang
4	Jm_brg	N/3	Jumlah penjualan barang
5	Jmlhrng	N/8	Harga jual barang

Tabel 4.6. Kamus data penambahan stok

Nama arus data	: Penambahan Stok		
Alias	: Stok Barang Bertambah		
Bentuk data	: File		
Arus data	: P1 → D3		
Penjelasan	: Menambah stok barang dari proses pembelian		
Periode	: Saat transaksi pembelian		
No	Nama	Type/Lebar	Keterangan
1	Tgl_masuk	D/8	Tanggal masuk barang
2	Kd_brg	C/4	Kode barang
3	Jm_brg	N/3	Jumlah barang
4	Hr_beli	N/10	Harga beli menurut tanggal masuk

Tabel 4.7. Kamus data pengurangan stok

Nama arus data	: Pengurangan Stok		
Alias	: Stok Barang Berkurang		
Bentuk data	: File		
Arus data	: P3 → D3		
Penjelasan	: Mengurangi stok barang dari proses pembelian		
Periode	: Saat transaksi penjualan		
No	Nama	Type/Lebar	Keterangan
1	Tgl_masuk	D/8	Tanggal masuk barang
2	Kd_brg	C/4	Kode barang
3	Jm_brg	N/3	Jumlah barang
4	Hr_brg	N/8	Harga barang

4.9 Struktur Database File

Dalam mendesain database harus diperhatikan penempatan data dalam file dan kebutuhan data yang akan digunakan dalam sistem. Adapun rancangan file dan struktur database yang digunakan pada perancangan sistem pengolahan

No	Nama	Type / Lebar	Keterangan
1	Kd_dist	C/4	Kode distributor
2	Nm_dist	C/15	Nama perusahaan distributor
3	Al_dist	C/25	Alamat distributor
4	Kt_dist	C/15	Kota distributor
5	Tl_dist	C/15	Nomor distributor

data penjualan, pembelian dan posisi stok adalah sebagai berikut :

1. FILE DISTRIBUTOR.DBF

File ini berfungsi untuk menyimpan data identitas distributor. Kunci primer (*primary key*) yang digunakan adalah **Kd_dist**.

2. FILE BARANG.DBF

File ini berfungsi untuk menyimpan data identitas barang. Kunci primer (*primary key*) yang digunakan adalah **Kd_brg**.

No	Nama	Type / Lebar	Keterangan
1	Kd_brg	C/4	Kode barang
2	Nama_brg	C/15	Nama barang
3	Merk	C/15	Merk barang
4	Ket_brg	C/25	Keterangan barang
5	Hr_brg	N/8	Harga Barang
6	Stok	N/4	Stok barang
7	Stok_min	N/4	Stok minimal barang

3. FILE BELI.DBF

File ini berfungsi untuk menyimpan data transaksi pembelian barang yang diterima HUDI MOTOR Karangrayung Grobogan dari distributor. Kunci primer (*primary key*) yang digunakan adalah **No_fakt**. Sedangkan Kd_dist berfungsi sebagai penghubung (*relasi*) dengan file induk distributor (Dist.dbf), Kd_brg sebagai penghubungan dengan file induk barang.dbf dan file stok.dbf.

No	Nama	Type / Lebar	Keterangan
1	Tgl_beli	D/8	Tanggal pembelian
2	No_Fakt	C/5	Nomor bukti faktur pembelian
3	Kd_dist	C/4	Kode distributor
4	Nm_dist	C/15	Nama distributor
5	Kd_brg	C/4	Kode barang
6	Nm_brg	C/15	Nama barang
7	Jm_beli	N/3	Jumlah barang yang dibeli
8	Hr_beli	N/8	Harga beli barang

4. FILE JUAL.DBF

File ini berfungsi untuk menyimpan data transaksi penjualan barang, khususnya mencatat data pembeli. Sedangkan data barang, jumlah serta harga dicatat apada file terpisah guna menghemat tempat penyimpanan. Kunci primer (*primary key*) yang digunakan adalah **No_nota**.

Struktur File JUAL.DBF			
No	Nama	Type / Lebar	Keterangan
1	Tgl_jual	D/8	Tanggal transaksi penjualan
2	No_nota	C/5	Nomor nota / bukti penjualan
3	Nm_pbl	C/15	Nama pembeli
4	Al_pbl	C/15	Alamat pembeli
5	Kt_pbl	C/15	Kota pembeli
6	Tl_pbl	C/15	Nomor telepon pembeli

5. FILE JUALDETAIL.DBF

File ini berfungsi untuk menyimpan detil data transaksi penjualan barang, khususnya mencatat data barang, jumlah serta harga. File ini termasuk file lemah yang tidak mempunyai kunci primer (*primary key*). Sedangkan **No_nota** digunakan untuk penghubung dengan file jual.dbf. Sedangkan Kd_brg berfungsi sebagai penghubung (*relasi*) dengan file induk barang.dbf dan file stok.dbf.

Struktur File JUALDETAIL.DBF			
No	Nama	Type / Lebar	Keterangan
1	No_nota	C/5	Nomor nota / bukti penjualan
2	Kd_brg	C/4	Kode barang
3	Nama_brg	C/15	Nama Barang
4	Jm_brg	N/3	Jumlah penjualan barang
5	Jmlhr	N/8	Harga jual barang

6. FILE RETUR.DBF

File ini berfungsi untuk menyimpan detil data transaksi retur pembelian barang, khususnya mencatat data barang, jumlah retur barang. Kunci primer (*primary key*) yang digunakan adalah **No_nota**.

Struktur File RETUR.DBF			
No	Nama	Type / Lebar	Keterangan
1	Tgl_rektur	D/8	Tanggal Retur pembelian barang
2	No_fakt	C/5	Nomor faktur
3	Kd_dist	C/4	Kode Distributor
4	Nm_dist	C/15	Nama Distributor
5	Kd_brg	C/4	Kode barang
6	Nama_brg	C/15	Nama Barang
7	Jm_brg	N/3	Jumlah barang

4.10 Desain Keluaran

4.10.1 Desain Tampilan Menu Utama

Batas Kota Motor Sukorejo		Date :	
		Timer :	
Aplikasi Persediaan Stok Barang			
Data	Transaksi	Laporan	Retur
Data Barang & Stok	Penjualan	Laporan Penjualan	Pembelian
Data Distributor	Pembelian	Laporan Pembelian	
		Laporan Data Barang & Stok	
		Laporan Data Distributor	
		Laporan Data Pelanggan	
		Laporan Retur Pembelian	
<small>@CopyRight.A11.2008.04089</small>			

Gambar 4.9 Desain Tampilan Menu

Utama

4.10.2 Desain Tampilan Data Distributor

Date :

Kode Distributor

Nama

Kota

Nomor Telepon

Data Distributor

Kode Dist	Nama Dist	Alamat Dist	Kota Dist	Telepon Dist

Add Edit Delete

Gambar 4.10 Desain Tampilan Data Distributor

4.10.5 Desain Tampilan Pembelian

Pembelian

Tanggal Beli

No Faktur

Kode Dist

Nama Dist

Kode Brg Nama Brg Sat HrgSat

Stok

Kd_brg	Nm_brg	Jm_brg	Hrg_beli
Total			

Add Delete

Gambar 4.13 Desain Tampilan Pembelian

4.10.3 Desain Tampilan Penjualan

Penjualan

Tanggal Jual

No Nota

Nama Pembeli

Alamat

Kota

Telepon

Kode Barang Nama brg Satuan Hrg sat

Stok

Kd_brg	Nama_brg	Jml_brg	Jmlharga
Total			

Add Delete

Gambar 4.11 Desain Tampilan Penjualan

4.10.6 Desain Tampilan Retur Pembelian

Rektur Pembelian

Tanggal Retur

No Faktur

Kode Dist

Nama Dist

Kode Brg Nama Brg Sat

Kd_brg	Nm_brg	Jm_brg

Retur

Gambar 4.14 Desain Tampilan Rektur Pembelian

4.10.4 Desain Tampilan Data Barang & Stok

Kode Brg

Nama Brg

Merk

Ket Brg

Hrg Brg

Stok

Stok:Min

Data Barang & Stok

Kode Brg	Nama Brg	Merk	Ket Brg	Hrg Brg	Stok	Stok Min

Add Edit Delete

Gambar 4.12 Desain Tampilan Data Barang & Stok

4.10.7 Desain Tampilan Laporan Penjualan

Laporan Penjualan

No Nota	Kode Barang	Nama Barang	Satuan	Harga Jual
N0001	O003	OLI	1	26.000
N0002	B001	BAUT	5	800

Gambar 4.15 Desain Tampilan Laporan Penjualan

4.10.8 Desain Tampilan Laporan Pembelian

Laporan Pembelian

Tgl beli	No Faktur	Kode Dist	Kode Brg	Nama Brg	Satuan	Harga beli
20-09-12	F0001	0003	O006	OLI	20	500.000
20-09-12	F0002	0001	B002	BAUT	200	20.000

Gambar 4.16 Desain Tampilan Laporan Pembelian

4.10.9 Desain Tampilan Laporan Data

Barang & Stok

Laporan Data Barang & Stok						
Kode brg	Nama brg	Merk	Ket brg	Harga	Stok	Stok Min
B001	BODY	ZENKU	SAMPING	40.000	13	5
B002	BAUT		KECIL	200	150	50

Gambar 4.17 Desain Tampilan Laporan Data Barang & Stok

4.10.10 Desain Tampilan Laporan Data

Distributor

Laporan Distributor					
Kode Dist	Nama Dist	Alamat Dist	Kota Dist	Telp Dist	
0001	INDAH LESTARI	Jl.Sujono 110	Sukorejo	0899124567	
0002	TARI GUNAWAN	Jl.Pasar kletek	Sukorejo	0856789012	

Gambar 4.18 Desain Tampilan Laporan Distributor

4.10.11 Desain Tampilan Laporan Data

Pelanggan

Laporan Data Pelanggan						
Tgl Jual	No Nota	Nama Pel	Alamat Pel	Kota Pel	Tel Pel	
20-09-12	N0001	Pak Widodo	Pasar Sujono	Sukorejo	0895612678	
20-09-12	N0002	Bu Rejeki	Jl.glentik	Sukorejo	0812234889	

Gambar 4.19 Desain Tampilan Laporan Data Pelanggan

4.10.12 Desain Tampilan Laporan Retur

Pembelian

Laporan Retur Barang						
Tgl Retur	No Faktur	Kode Dist	Kode Dist	Kode Brg	Nama Brg	Jumlah
20-09-12	F0001	0003	PT.JATENG TOP	O006	OLI	2
20-09-12	F0003	0002	TARI GUNAWAN	H001	HELM	1

Gambar 4.20 Desain Tampilan Laporan Retur Barang

4.11 Hasil Implementasi

Merupakan tampilan untuk mengisi data sesuai dengan nama fieldnya. Dalam program ini masukan terdiri dari data distributor, data penjualan, data barang dan data pembelian.

4.11.1 Tampilan Program

4.11.1.1 Tampilan Menu Utama



Gambar 4.21 Tampilan Menu Utama

4.11.1.2 Tampilan Data Distributor

Sesuai dengan filenya makan fungsinya adalah untuk mencatat segala hal yang berkaitan dengan data-data distributor.



Gambar 4.22 Tampilan Input,Edit

dan Delete Data Distributor

4.11.1.3 Tampilan Input Data

Penjualan

Masukan ini berfungsi untuk mencatat data yang berkaitan dengan penjualan. Hal ini untuk mempermudah dalam mengetahui perkiraan stok barang yang ada pada gudang jika pembeli memberikan pesan pembelian.



Gambat 4.23. Tampilan Input & Delete Penjualan

4.11.1.4 Tampilan Input Data Barang

Masukan ini berfungsi untuk mencatat data-data barang yang ada dalam perusahaan. Hal ini untuk mempermudah dalam mengetahui perkiraan stok barang yang ada dalam gudang jika pembeli memberikan pesan pembelian.



Gambar 4.24. Tampilan Input/edit/hapus data barang

4.11.1.5 Tampilan Input Data

Pembelian

Masukan ini berfungsi untuk mencatat transaksi pembelian yang dilakukan oleh perusahaan dengan distributor.



Gambar 4.25. Tampilan Input & Delete pembelian

4.11.1.6 Tampilan Input Data Retur

Pembelian

Masukan ini berfungsi untuk mencatat transaksi retur pembelian yang dilakukan oleh perusahaan dengan distributor.



Gambar 4.26. Tampilan Retur Pembelian

4.11.2 Tampilan Keluran Laporan

Proses ini menghasilkan informasi akhir dari pengolahan data masukan. Beberapa yang dihasilkan oleh program ini adalah laporan penjualan, laporan pembelian dan laporan barang & stok barang.

4.11.2.1 Laporan Penjualan

Laporan ini berfungsi untuk mengetahui berapa penjualan yang telah dicapai perusahaan, dengan demikian target penjualan bisa dilihat sudah tercapai atau belum.

No Nota	Kode Brg	Nama Barang	Satuan	Harga Jual
N0001	O003	OLI	2	58.000
N0001	O004	OLI	1	29.000
N0002	B001	Body supra	1	40.000
N0003	K001	Kamprat	2	60.000
N0004	O008	OLI	2	84.000

Gambar 4.27. Tampilan Laporan Penjuala

4.11.2.2 Laporan Pembelian Barang

Laporan ini berfungsi untuk mengetahui berapa pembelian yang telah dilakukan distributor, dari sini akan diketahui apakah pesanan barang sudah terpenuhi atau belum.

Tgl Beli	No Faktur	Kode Dist	Kode Brg	Nama Brg	Satuan	Harga Beli
20-09-12	F0001	0003	O006	OLI	20	500.000
20-09-12	F0002	0001	B002	Baut	200	20.000
20-09-12	F0003	0002	H001	Helm	5	400.000
20-09-12	F0004	0003	O007	OLI	10	330.000

Gambar 4.28. Tampilan Laporan Pembelian

4.11.2.3 Laporan Data Barang & Stok

Laporan ini berfungsi untuk mengetahui macam-macam barang yang diproduksi dan dapat mengetahui jumlah barang yang mendekati stok minimal.

Kode Brg	Nama Barang	Merk	Keterangan Brg	Harga	Stok	Stok Min
B001	Body supra	zarisu	samping	40.000	13	5
B002	Baut	keca	keca	200	160	50
H001	Helm	Azadi	ukuran L	80.000	10	4
K001	Kamprat	Supra		30.000	11	4
O001	Os	Meyran	0,8 Mer	25.000	30	10
O002	OLI	CASTROL GOLD	0,8 LTR	25.000	30	5
O003	OLI	CASTROL GOLD	1 LTR	29.000	31	5

Gambar 4.29. Tampilan Data Laporan Barang & Stok

4.11.2.4 Laporan Data Distributor

Laporan ini berfungsi untuk mengetahui data distributor secara terperinci.

Kode Distributor	Nama Distributor	Alamat Distributor	Kota Distributor	Telepon Distributor
0001	Indah Iszari	J. Ingono 110	Sukoraga	099124567
0002	Tari Gunawan	J. Pasa Babak	Sukoraga	095779002
0003	PT JATENG TOP	Jl. MAJAPAHIT NO 84 B	SEMARANG	024673018
0004	ASTRA OTOPART	Jl. MAJAPAHIT NO 84	SEMARANG	024672564
0005	TGP WAHYU	Jl. UPAJWA NO 3	SURABAYA	0023991899
0006	CASTROL	Jl. LET. JEN TB KAV 88	JAKARTA	02178843879

Gambar 4.30. Tampilan Laporan Distributor

4.11.2.5 Laporan Data Pelanggan

Laporan ini berfungsi untuk mengetahui data Pelanggan secara lebih detail.

Tgl Jual	No Nota	Nama Pelanggan	Alamat Pelanggan	Kota Pelanggan	Telepon Pelanggan
20-09-12	N0001	pak widodo	pasar sujono	sukoraga	0990612679
20-09-12	N0002	bu rjpti	j. glantik	sukoraga	0812234889
20-09-12	N0003	pak widjan	pasar teratak	wetan	0806780123
20-09-12	N0004	bu susilo	j. kedon	sukoraga	0879911150
20-09-12	N0005	bu kongwar	cedak pasar	sukoraga	029167991

Gambar 4.31. Tampilan Laporan Pelanggan

4.11.2.6 Laporan Retur Pembelian

Laporan ini berfungsi untuk mengetahui data retur pembelian secara terperinci.

Tgl Retur	No Faktur	Kode Dist	Nama Dist	Kode dist	Nama Barang	Jumlah
20-09-12	F9001	0003	PT JATENG TOP	O006	OLI	2
20-09-12	F9003	0002	Tari Gunawan	H001	Helm	1

Gambar 4.32. Tampilan Laporan Retur

Pembelian

BAB V

PENUTUP

5.1 Kesimpulan

Setelah melalui tahap perencanaan, analisis dan desain aplikasi perangkat lunak penulis mengambil beberapa kesimpulan diantaranya adalah dalam pembuatan suatu sistem pengolahan data hendaknya mempertimbangkan masalah efisien waktu, tenaga pelaksana dan biaya pengoperasian. Karena bila mengingat sistem pengolahan data administrasi stok ini dilakukan secara manual untuk mengatasi hal tersebut diatas maka perlu diambil suatu langkah baru yaitu pengolahan data dengan sistem komputerisasi sehingga sistem kerja menjadi lebih efisien, cepat dan tidak menekan biaya.

Pengolahan data administrasi stok yang dikembangkan adalah sistem yang baru yang

merupakan pengimplementasian sistem lama melalui komputerisasi. Untuk memperjelas pemahaman tentang *system* yang dirancang digunakan alat-alat pendukung analisa dan desain *Data Flow Diagram* (DFD), *Entity Relationship Diagram* (ERD), *Flow Of Document* (FOD) dan Normalisasi. Untuk mengefektifkan sistem komputerisasi tersebut digunakan bahasa pemrograman yang berorientasi pada manajemen basis data (Foxpro) karena data pengolahan administrasi stok adalah data yang kompleks dan memerlukan penanganan yang cermat untuk hasil yang cepat dan akurat. Selain itu Foxpro juga memberikan kemudahan dalam pencarian data dan integrasi data.

5.2 Saran - Saran

Agar sistem baru yang dirancang bisa diimplementasikan sebagaimana yang diharapkan dan memberikan hasil yang maksimal, maka penulis mengajukan beberapa masukan berupa saran yang mungkin berguna bagi pemakai.

1. Ditempatkan seorang operator yang benar-benar paham pada sistem yang dijalankan dan dalam pengolahan data.
2. Penggunaan komputer dituntut kedisiplinan yang tinggi para pemakai tersebut dalam penyimpanan data masukan yang sesuai sehingga dapat menghasilkan keluaran sesuai

dengan apa yang diharapkan. Untuk itu perlu diadakan kerjasama yang baik untuk setiap bagiannya.

3. Dengan penggunaan sistem komputerisasi yang tidak terlepas dari gangguan *software* yang ada, untuk itu perlu ditekankan pada para pemakai hendaknya mengikuti struktur program yang telah ada sehingga terhindar dari kesalahan yang diharapkan.