

**SISTEM INFORMASI ADMINISTRASI KEPEGAWAIAN PADA UPT DINAS
PEMUDA DAN OLAHRAGA PAKIS AJI KABUPATEN JEPARA**

Rifki Verianto

A12.2008.02985

Sistem Informatika S1

Fakultas Ilmu Komputer

Universitas Dian Nuswantoro Semarang

ABSTRAK

Pada UPTD Dinas Pemuda Dan Olah raga Kota Jepara pengisian data pegawai masih menggunakan microsoft acces ,sehingga dalam proses pengisian lama dan seringkali terjadi adanya inkonsistensi data karena tidak adanya fasilitas pengaman data yang menyebabkan duplikasi data.berdasarkan masalah diatas penulis meminimalkan kesalahan pengimputan data .dalam penelitian ini penulis menggunakan metode wawancara secara langsung kepada pihak bappeda provinsi jateng serta observasi system secara langsung pada system yang sudah digunakan oleh UPTD dinas pemuda olahraga kota jepara .hasil dari penelitian fungsi trigger dan constraint dapat mengatasi permasalahan redudansi data,inkonsistensi data,dan mampu mempercepat pengaksesan proses penginputan data pegawai. Perancangan sistem yang digunakan mulai dari Context Diagram, Dekomposisi Diagram, Data Flow Diagram (DFD), Entity Relationship Diagram (ERD), Data Dictionary (DD), dan Normalisasi Data menghasilkan suatu konsep untuk system informasi administrasi sistem kepegawaian menggunakan bahasa pemrograman visual basic dan database mysql.

The objectives to be achieved in this thesis is trying to produce a database design that is really able to connect the data that are separate and anticipate the duplicate data. Research methodology include the primary data and literature study. Literature study conducted by the research literature relevant to the issue. With the inventory of information systems can facilitate the administration of the store or to make a report. And can cope with redundancy, and disintegration, data inconsistencies in the transaction process. Can also help reduce errors in the recording process procurement transactions. Additionally, it will make it easier to control the data collection of goods.

Kata kunci : Inventory and Data Base Design of Spare Parts

I. PENDAHULUAN

1.1 Latar Belakang Masalah

Bengkel merupakan suatu jenis wirausaha kecil atau menengah yang bergerak dalam bidang jasa pelayanan perbaikan kendaraan 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 jasa service kendaraan dan modifikasi motor adalah Bengkel Lancar Jaya Motor. Bengkel ini merupakan salah satu toko penyedia *sparepart* kendaraan bermotor yang menjual berbagai jenis *sparepart* motor dan *sparepart* variasi untuk modifikasi motor dengan jumlah persediaan barang yang cukup banyak. Puluhan dan bahkan ratusan transaksi baik itu barang masuk dan barang keluar terjadi setiap harinya, maka tidak heran jika setiap harinya terjadi penumpukan berkas, dari dokumen yang sebanyak itu seringkali terjadi kesalahan maka harus dicari dokumen asli untuk kemudian dilakukan pencatatan kembali, yang terjadi adalah harus mencari dokumen yang dimaksud secara satu persatu diantara tumpukan data yang ada. Pengolahan data pada Bengkel Lancar Jaya Motor masih menggunakan buku manual, sehingga harus menulis satu persatu sampai dengan jumlah transaksi yang terjadi pada hari itu.

Setiap barang yang ada mempunyai kode barang, banyaknya kode barang yang ditulis memungkinkan terjadinya kesamaan. Kesamaan kunci yang berupa kode ini mengakibatkan *redundancy* data sehingga berdampak pula pada laporan yang dihasilkan. Akibat lain yang dihasilkan dari *redundancy* data adalah ketika melakukan pencarian, maka data yang ketemu tidak seperti yang dicari.

Pengaruh lain dari *redundancy* data adalah integrasi data yang kurang baik. Sebagai contoh transaksi dari komputer satu mengambil data barang dengan kode 101, tetapi karena dikomputer lain menyimpan kode barang 101 lebih dari satu maka yang dihasilkan adalah nama barang yang berbeda.

Dengan adanya permasalahan-permasalahan diatas diperlukan suatu perancangan basis data untuk persediaan *sparepart* yang mampu mengatur *sparepart* masuk dan *sparepart* keluar. Sehingga penulis membuat tugas akhir dengan judul “Perancangan Basis Data Persediaan Sparepart Pada Bengkel Lancar Jaya Motor”.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas maka dapat dirumuskan permasalahan sebagai berikut:

1. Bagaimana merancang sebuah basis data persediaan *sparepart* yang mampu merelasikan data-data yang ada sehingga akan memudahkan dalam pembuatan laporan.

2. Bagaimana mengatasi terjadinya kehilangan informasi yang diakibatkan seringnya ada perubahan harga, sehingga sulit menampilkan *history*/harga – harga barang yang sudah lama terjadi.

1.3 Tujuan Penelitian

Didalam aktivitasnya menulis laporan Tugas Akhir ini penulis mempunyai tujuan:

1. Menghasilkan sebuah rancangan basis data persediaan *sparepart* yang mampu merelasikan data-data yang ada sehingga akan memudahkan dalam pembuatan laporan.
2. Mampu mengatasi terjadinya kehilangan informasi yang diakibatkan seringnya ada perubahan harga, sehingga sulit menampilkan *history*/harga – harga barang yang sudah lama terjadi.

II. LANDASAN TEORI

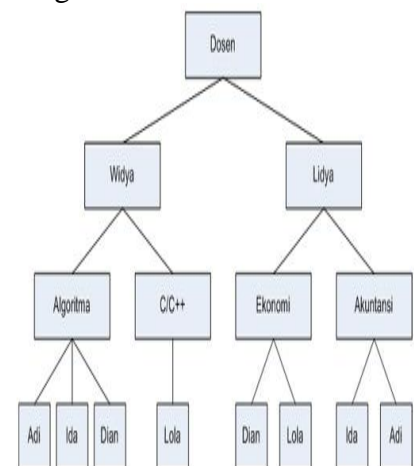
2.1 Model Basis Data

Pada basis data relasional ini, basis data akan dinyatakan dalam bentuk tabel-tabel dua dimensi. Setiap tabel terdiri atas lajur mendatar yang disebut dengan baris (row/record) dan lajur vertikal yang disebut kolom (column/field). Baris-baris ini akan tersusun membentuk satu tabel, yang biasanya tersimpan dalam satu file. Tabel-tabel ini secara keseluruhan merupakan penyajian dari atribut data yang saling berhubungan[8].

1. Model Database Hirarki (*Hierarchical Database Model*)

Model data hirarkis adalah model data paling tua yang pernah diterapkan dalam suatu DBMS. Model ini mengikuti pola hirarki pada suatu organisasi atau pada suatu keluarga, dimana terdapat rekaman data yang berfungsi sebagai “bapak” (parent-record) ada yang berfungsi sebagai “anak” (child-record), atau sebagai “pimpinan” dan “anak-buah”. Dalam model ini seorang “bapak” bisa memiliki lebih dari satu “anak” tetapi seorang “anak” hanya boleh memiliki satu “bapak”.

Contoh model hirarkis yang menunjukkan hubungan Dosen-MataKuliah-Mahasiswa dapat pula digambarkan dalam bentuk diagram sebagai berikut



Gambar 2.2 : Diagram Hirarkis Dosen-Matakuliah-Mahasiswa

Perhatikan dalam model hirarki diatas, data mahasiswa yang sama terpaksa direkam berulang ketika mahasiswa tersebut memprogramkan lebih dari satu matakuliah.

Model hirarkis ini pernah diimplementasikan oleh IBM pada sebuah DBMS yang diberi nama

IMS (Information Management System) namun kemudian dikalahkan oleh implementasi model yang lebih mutakhir seperti model Data Relasional seperti Rbase, dan sebagainya[6].

1. Model Database Relasi (*Relational Database Model*)

Model database relasi merupakan model database yang paling banyak digunakan saat ini, karena paling sederhana dan mudah digunakan serta yang paling penting adalah kemampuannya dalam mengakomodasi berbagai kebutuhan pengelolaan database. Sebuah database dalam model ini disusun dalam bentuk tabel dua dimensi yang terdiri dari baris (*record*) dan kolom (*field*), pertemuan antara baris dengan kolom disebut item data (*data value*), tabel-tabel yang ada di hubungkan (*relationship*) sedemikian rupa menggunakan field-field kunci (*key field*) sehingga dapat meminimalkan duplikasi data[10].

2.2 Integritas Data, Redundansi Data, Inkonsistensi Data

2.2.1 Integritas Data

Basis data berisi *file / table* yang saling terkait, masalah utama adalah bagaimana kaitan antar *table* itu terjadi. Meskipun kita mengetahui *table* A berkaitan dengan *table* B, namun secara teknis ada *field / atribut* kunci yang mengaitkan atau merelasikan *table* tersebut.

2.2.2 Redundansi data

Redundansi (kelebihan) ialah penyimpanan data yang sama secara berulang, atau jika data yang dapat diperoleh dari

data lain disimpan tersendiri. Redundansi menyebabkan masalah pada waktu memperbarui (*update*) data, ruang penyimpanan yang boros dan dapat menimbulkan tidak konsistennya data. Untuk membuat suatu basis data yang memberikan manfaat optimal, suatu *inventory* data harus dibuat. Data dan informasi yang diperlukan harus dianalisis, file basis data yang diperlukan harus dirancang, dan prosedur untuk memelihara basis data harus diadakan[16].

2.2.3 Inkonsistensi Data

Penyimpanan data yang sama berulang-ulang di beberapa tempat dalam *database* dapat mengakibatkan juga *inkonsistensi* (tidak konsisten). Hal ini terjadi, bila suatu ketika mahasiswa tersebut pindah alamat berubah, maka seluruh *table* yang memuat data tersebut harus diubah / *update*. Bila salah satu saja yang *diupdate*, maka menjadi tidak konsisten.

2.2 Relasi

Relasi adalah hubungan antara suatu himpunan dengan himpunan entitas yang lainnya. Pada penggambaran diagram hubungan entitas, relasi adalah perekat yang menghubungkan suatu entitas dengan entitas lainnya. Relasi merupakan hubungan yang berarti antara suatu entitas dengan entitas lainnya. Frasa ini berimplikasi bahwa relasi mengijinkan untuk menjawab pertanyaan-pertanyaan yang berkaitan dengan hubungan suatu entitas dengan lainnya. Hubungan dibedakan antar bentuk hubungan antar entitas dengan isi. Misalnya kasus hubungan antara

entitas pegawai dan entitas bagian adalah jam kerja, sedangkan isi hubungannya dapat berupa total jam kerja, gaji lembur. Relasi digambarkan dalam bentuk intan. Pada model data relasi hubungan antar data dihubungkan dengan kunci relasi. Tipe hubungan diantara beberapa buah tipe entitas adalah kumpulan dari relasi di antara entitas-entitas dari tipe entitas tersebut[2].

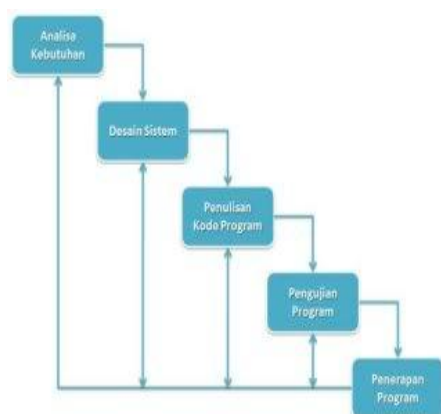
2.3 Mysql

MySQL adalah Relational database Management System (RDBMS) yang didistribusikan secara gratis dibawah license GPL (General Public License) [11].

III. METODE PENELITIAN

3.1 Metode Pengembangan

Metode pengembangan sistem yang digunakan adalah dengan menggunakan metode Model *Waterfall* yang merupakan metode yang berfungsi sebagai sebuah mekanisme untuk mengidentifikasi kebutuhan perangkat lunak. Metode ini merupakan metode yang sering digunakan oleh penganalisa sistem pada umumnya. Inti dari metode *waterfall* adalah pengerjaan dari suatu sistem dilakukan secara berurutan atau secara linear.



1. Analisa

Langkah ini merupakan analisa terhadap kebutuhan sistem. Pengumpulan data dalam tahap ini bisa malakukan sebuah penelitian, wawancara atau study literatur.

2. Design

Proses desain akan menerjemahkan syarat kebutuhan ke sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat coding. Proses ini berfokus pada : struktur data, arsitektur perangkat lunak, representasi interface, dan detail (algoritma) prosedural.

3. Coding & Testing

Coding merupakan penerjemahan *design* dalam bahasa yang bisa dikenali oleh komputer. Dilakukan oleh *programmer* yang akan meterjemahkan transaksi yang diminta oleh *user*. Tahapan ini lah yang merupakan tahapan secara nyata dalam mengerjakan suatu sistem.

4. Penerapan

Tahapan ini bisa dikatakan *final* dalam pembuatan sebuah sistem. Setelah melakukan analisa, design dan pengkodean maka sistem yang sudah jadi akan digunakan oleh *user*.

5. Pemeliharaan

Perangkat lunak yang sudah disampaikan kepada pelanggan pasti akan mengalami perubahan.

Perubahan tersebut bisa karena mengalami kesalahan karena perangkat lunak harus menyesuaikan dengan lingkungan (peripheral atau sistem operasi baru) baru, atau karena pelanggan membutuhkan perkembangan fungsional.

Keuntungan Metode Waterfall

- Kualitas dari sistem yang dihasilkan akan baik. Ini dikarenakan oleh pelaksanaannya secara bertahap. Sehingga tidak terfokus pada tahapan tertentu.
- Document pengembangan sistem sangat terorganisir, karena setiap fase harus terselesaikan dengan lengkap sebelum melangkah ke fase berikutnya. Jadi setiap fase atau tahapan akan mempunyai dokumen tertentu.

Kelemahan waterfall

- Diperlukan manajemen yang baik, karena proses pengembangan tidak dapat dilakukan secara berulang sebelum terjadinya suatu produk.
- Kesalahan kecil akan menjadi masalah besar jika tidak diketahui sejak awal pengembangan.

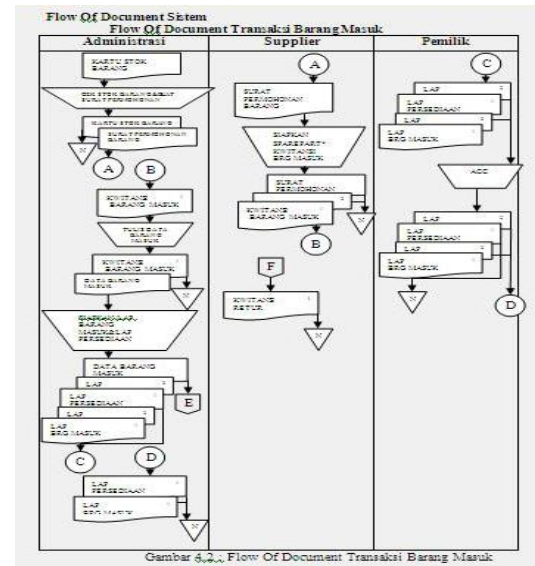
Pelanggan sulit menyatakan kebutuhan secara eksplisit sehingga tidak dapat mengakomodasi

ketidakpastian pada saat awal pengembangan

IV. ANALISIS DAN PERANCANGAN SISTEM

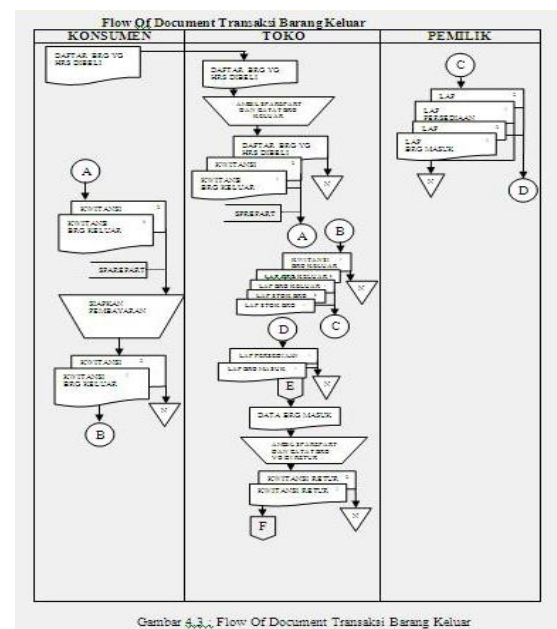
4.1 Flowchart

Berikut adalah flowchart barang masuk



Gambar 4.2., Flow Of Document Transaksi Barang Masuk

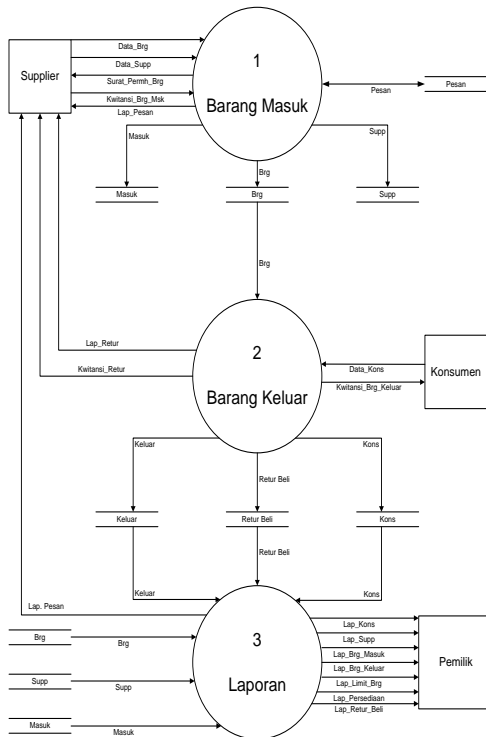
Berikut adalah flowchart barang keluar



Gambar 4.3., Flow Of Document Transaksi Barang Keluar

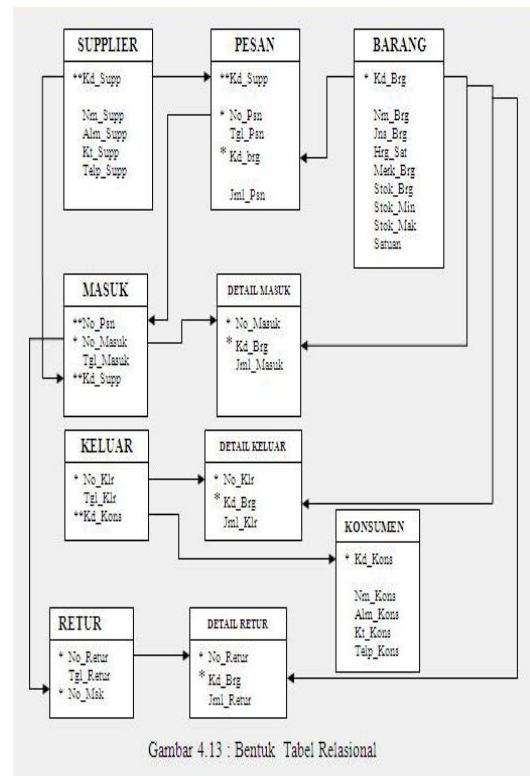
4.2 DFD Level 0

Berikut adalah gambar DFD Level 0



4.3 Tabel Relasional

Berikut adalah gambar tabel relasional



Gambar 4.13 : Bentuk Tabel Relasional

V. PENUTUP

5.1 Kesimpulan

Dari hasil penelitian dan pengamatan yang telah penulis lakukan di Bengkel LANCAR JAYA MOTOR, maka dapat diberikan beberapa kesimpulan sebagai berikut :

- Memudahkan bagian toko atau administrasi untuk membuat laporan
- Dapat mengatasi *redundancy*, *disintegrasi* dan *inkonsistensi* data dalam proses transaksi pada Bengkel Lancar Jaya Motor.
- Dapat membantu mengurangi kesalahan dalam proses pencatatan transaksi pengadaan barang.

- d. Akan semakin mudah dalam mengontrol pendataan barang.

5.2 Saran-Saran

Sistem yang baru ini tentunya masih terdapat banyak kekurangan dan masih perlu upaya pengembangan, untuk itu saran yang ada yaitu :

- a. Perlunya dikembangkan ke arah sistem yang lebih baik, misalnya bisa dikontrol secara online.

DAFTAR PUSTAKA

- [1] Arbie. (2004). *Manajemen Database dengan MySQL*. Yogyakarta : Penerbit ANDI offset
- [2] Kadir, Abdul. (2009). *Dasar Perancangan Dan Implementasi Database Relational*. Yogyakarta: Andi Offset.
- [3] Kusriani. (2007). *strategi Perancangan dan Pengelolaan Basis Data*. Yogyakarta: Andi Offset. 39
- [4] Nugroho, Adi. (2010). *Mengembangkan Aplikasi Basis Data Menggunakan C# dan SQL Server*. Yogyakarta: Andi Offset. 29
- [5] Rudyanto, M. Arief.(2006). *Pemrograman Basis Data Menggunakan Transact-SQL Dengan Microsoft SQL Server* 2000. Yogyakarta : Andi Offset
- [6] Simarmata, Janner, Imam paryudi. (2005). *Basis Data*. Yogyakarta: Andi Offset. 1
- [7] <http://adynu.wordpress.com/2011/05/10/perancangan-diagram-alir-data/data> , diakses tanggal 19 September 2011
- [8] <http://blog.re.or.id/model-basis-data.htm> , diakses tanggal 14 September 2011
- [9] <http://dewiar.staff.gunadarma.ac.id/Downloads/files/425/M2+-+DBMS.pdf> , diakses tanggal 20 September 2011
- [10] http://eprints.undip.ac.id/6053/1/DESAIN_DATA_BASE_-_CAHYA_TRI_PURNAMI.pdf , diakses tanggal 15 September 2011
- [11] <http://id.wikipedia.org/wiki/MySQL>, Diakses pada Tanggal 20 September 2011
- [12] <http://krida85.wordpress.com/2008/04/16/pengertian-basis-data/> , diakses tanggal 14 September 2011
- [13] <http://lunarphue.wordpress.com/data/ddl-dan-dml/constraint-in-database> , diakses tanggal 21 September 2011

- [14] <http://nurmanto.com/model-model-basis-data/> ,diakses tanggal 8 Mei 2011
- [15] <http://ridho-thesoftware.blogspot.com/2011/03/apa-itu-sql.html> , diakses tanggal 20 September 2011
- [16] <http://tembilahanitgenerations.blogspot.com/2010/10/materi-kuliah-si-c-perancangan-basis.html> , diakses tanggal 17 September 2011
- [17] <http://www.docstoc.com/docs/12217588/operasi-basis-data> , diakses tanggal 21 September 2011
- [18] <http://dansite.wordpress.com/2009/03/31/pengertian-persediaan-inventory/> , Diakses pada 20 Februari 2011