

# APLIKASI SISTEM JUAL BELI SEPEDA MOTOR BERBASIS WEB DI USAHA DAGANG (UD) RUHMA MOTOR

**Sujito, Lesa Ahsanti Ruma<sup>1)</sup>**

Program Studi Sistem Informasi STMIK Pradnya Paramita<sup>1)</sup>

Jl. Laksda Adi Sucipto No. 249A Malang (0341) 412699

E-mail: sujito@pradnya.ac.id

## **ABSTRACT**

Business competition is increasingly fierce to make entrepreneurs find a way to continue to develop and advance their business. Efficiency in the field of administration and finance of particular concern to employers. UD. Ruhma Motors is one company that handles business in buying and selling motorcycles. Currently, efforts UD. Ruhma Motor is developing that require the accuracy of data in the process of buying and selling. This study aims to establish information system selling motorcycles in UD. Ruhma Motor Web-based using php. Methods of research on the development of this information system using the System Development Life Cycle.

*This research produces information systems web-based trading that can help business owners to continue to monitor the development efforts. Data motors and customers can always controlled and stored neatly in the administrative database that helps officers to make reports.*

*Keywords: systems, information, buying and selling, UD Ruhma Motor.*

## **PENDAHULUAN**

Perkembangan teknologi saat ini semakin meningkat, hal ini terbukti dengan banyaknya perkembangan peralatan dan pengembangan aplikasi berbasis komputer. Rutinitas pekerjaan yang biasanya dikerjakan oleh tenaga manusia dapat digantikan dengan memanfaatkan teknologi komputer.

Selain itu, dengan berkembangnya teknologi komputer juga menyebabkan masuknya budaya asing semakin mudah. Hal ini mengakibatkan budaya tradisional semakin ditinggalkan. Padahal banyak potensi yang dapat digali dari kebudayaan tradisional. Contoh konkret dari permasalahan ini adalah telah banyak orang yang enggan mempelajari Aksara Jawa. Aturan Aksara Jawa yang tergolong sulit

karena banyak pola membuat orang enggan untuk mempelajarinya.

Dari fakta tersebut, perlu dilakukan inovasi agar dalam belajar Aksara Jawa lebih mudah sesuai dengan perkembangan teknologi saat ini. Hal ini dapat diwujudkan dengan memanfaatkan teknologi untuk pembelajaran Aksara Jawa. Wujud nyata yang dapat dilakukan

## **Rumusan Masalah**

Rumusan masalah dalam penelitian ini adalah “Bagaimana membangun aplikasi sistem informasi penjualan dan pembelian sepeda motor berbasis web menggunakan Php dan MySql untuk layanan informasi administrasi dan keuangan di UD. Ruhma Motor ?

## Tujuan Penelitian

Tujuan dari penelitian ini adalah membangun aplikasi sistem informasi penjualan dan pembelian sepeda motor berbasis web menggunakan Php dan MySql untuk layanan informasi administrasi dan keuangan di UD. Ruhma Motor ?

## Manfaat Penelitian

Manfaat dari penelitian ini adalah:

- a. Mendukung proses jual beli sepeda motor di UD. Ruhma Motor sehingga menjadi lebih mudah dan efisien
- b. Mengantisipasi adanya kesalahan pembiayaan yang tidak diinginkan untuk memperoleh data yang lebih akurat

## KAJIAN TEORI

### 1. Sistem Informasi

Menurut (Jogiyanto,2005), sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.

Berikut ini tentang komponen fisik sistem informasi:

1. Perangkat keras komputer: CPU, *storage*, perangkat *input/output*, terminal untuk interaksi, media komunikasi data.
2. Perangkat lunak komputer: perangkat lunak sistem (sistem operasi dan perangkatnya),

perangkat lunak umum aplikasi (bahasa pemrograman), perangkat lunak aplikasi (aplikasi akuntansi dan lain-lain).

3. Basis data: penyimpanan data pada media penyimpan komputer.
4. Prosedur: langkah-langkah penggunaan sistem.
5. Personil untuk pengelolaan operasi (SDM), meliputi:
  - a. *Clerical personnel* (untuk menangani transaksi dan pemrosesan data dan melakukan inquiry = operator)
  - b. *First level manager*: untuk mengelola pemrosesan data didukung dengan perencanaan, penjadwalan, identifikasi situasi *out-of-control* dan pengambilan keputusan *level* menengah ke bawah
  - c. *Staff specialist*: digunakan untuk analisis perencanaan dan pelaporan
  - d. *Management*: untuk pembuatan laporan berkala, permintaan khusus, analisis khusus, laporan khusus, pendukung identifikasi masalah dan peluang

### 2. PHP

PHP merupakan *script* untuk pemrograman *script web server side*, *script* yang membuat dokumen HTML secara *on the fly*, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan *editor* teks atau *editor* HTML.

Dengan menggunakan PHP maka *maintenance* suatu situs *web* menjadi lebih mudah. Proses *update* data dapat dilakukan dengan menggunakan aplikasi yang dibuat dengan menggunakan *script* PHP.

PHP/FI merupakan nama awal dari PHP. PHP – *Personal Home Page*, FI adalah *Form Interface*. Dibuat pertama kali oleh Rasmus Lerdoff. PHP, awalnya merupakan program CGI yang dikhususkan untuk menerima masukan melalui *form* yang ditampilkan dalam *browser web*.

Software ini disebarkan dan dilisensikan sebagai perangkat lunak *open source*. PHP secara resmi merupakan kependekan dari PHP: *Hypertext Preprocessor*, merupakan bahasa *script server-side* yang disisipkan pada HTML.

### 3. Database

(Kadir, 2004) *Database* adalah kumpulan data (elementer) yang secara logik berkaitan dalam merepresentasikan fenomena/fakta secara terstruktur dalam domain tertentu untuk mendukung aplikasi pada sistem tertentu. *Database* adalah kumpulan data yang saling berhubungan yang merefleksikan fakta-fakta yang terdapat di organisasi. *Database* mendeskripsikan *state* organisasi/perusahaan/sistem. Saat satu kejadian muncul di dunia nyata mengubah *state* organisasi/perusahaan/sistem maka satu perubahanpun harus dilakukan terhadap data yang disimpan di *database*. *Database* merupakan komponen utama sistem informasi karena semua informasi untuk pengambilan keputusan berasal dari data di *database*. Pengelolaan *database* yang

buruk dapat mengakibatkan ketidaktersediaan data yang penting yang digunakan untuk menghasilkan informasi yang diperlukan dalam pengambilan keputusan.

Sistem manajemen *database* atau *Database Management System (DBMS)* adalah perangkat lunak untuk mendefinisikan, menciptakan, mengelola dan mengendalikan pengaksesan *database*. Fungsi sistem manajemen *database* saat ini yang paling penting adalah menyediakan basis untuk sistem informasi manajemen.

### 4. Arsitektur Aplikasi Database

(Kadir, 2004) Aplikasi-aplikasi *database* terdiri dari beberapa fungsi, seperti:

1. *Database management system (DBMS)*
2. *Application Logic (business rules)*
3. *Presentation Logic (interface)*

Pada arsitektur *client-server* fungsi-fungsi tersebut dapat diletakkan antara *client* dan *server* dalam beberapa cara. Berikut ini merupakan dua (2) arsitektur *client-server* yang mungkin untuk sesuatu aplikasi *database*.

#### 1. Arsitektur *two-tier*

Pada arsitektur *two-tier*, *application logic* dan *database driver* terletak di sisi *client* sedangkan *DBMS* terletak di sisi *server*. Arsitektur *two-tier* dikenal dengan *fat-client architecture*, di mana kode dan pemrosesan banyak dilakukan di sisi *client*. Diagram arsitektur *two-tier* disajikan pada gambar 1.

### Gambar 1 Arsitektur 2-tier

#### 2. Arsitektur *three-tier*

Pada arsitektur *three-tier*, hanya *presentation logic* atau *interface* yang terletak di sisi *client*. *Application* atau *business rules* terletak di sisi *server* sedangkan *DBMS* dapat diletakkan di sisi *server* atau komputer ketiga. Arsitektur *three-tier* sering dikenal sebagai *thin-client*, di mana kode dan pemrosesan di sisi *client* lebih sedikit dibandingkan di *server*. *Presentation code* dikirim ke *client* berupa HTML, sedangkan *business logic* pada yang menghubungkan ke *database system* pada *server*. Diagram arsitektur *three-tier* disajikan pada gambar 2.

### Gambar 2 Arsitektur 3-tier

## 5. SQL

(Kadir, 2004) SQL (dibaca "ess-que-el") merupakan singkatan dari *Structure Query Language*. SQL digunakan untuk berkomunikasi

dengan suatu *database*. Berdasarkan ANSI (*American National Standards Institute*), SQL merupakan bahasa standar untuk *relational database management systems*. Pernyataan-pernyataan SQL digunakan untuk melakukan fungsi-fungsi seperti *update* data pada *database*, atau pengambilan data dari *database*. Beberapa *relational* DBMS yang menggunakan SQL, antara lain : Oracle, Sysbase, Microsoft Sql Server, Access, Ingress, MySQL, Postgre dan masih banyak lagi. Meskipun hampir semua DBMS menggunakan SQL, banyak DBMS yang memiliki tambahan kemampuan dalam SQL-nya dan biasanya digunakan khusus untuk DBMS tersebut.

SQL memiliki beberapa versi antara lain : SQL1 atau dikenal dengan SQL-86, SQL-2 atau dikenal dengan SQL-92 dan SQL-3 yang dikenal dengan SQL-99.

Pernyataan-pernyataan SQL dapat dibagi menjadi dua (2) kategori, yaitu pernyataan-pernyataan *Data Manipulation Language* (DML) dan *Data Definition Language* (DDL). Pernyataan-pernyataan DML berkaitan dengan data, baik pengambilan atau pemodifikasian data supaya tetap *up-to-date*. Sedangkan pernyataan-pernyataan DDL berkaitan dengan pembuatan atau perubahan *table* dan objek-objek *database* lainnya seperti *view* dan *index*.

Berikut ini merupakan pernyataan-pernyataan *DML*, antara lain :

#### 1. *SELECT*

Digunakan untuk *query* dan menampilkan data dari sebuah *database*. Pernyataan *SELECT* menspesifikasikan

kolom mana yang termasuk dalam hasil tes (hasil *query*). Pernyataan *SELECT* ini merupakan pernyataan yang paling sering digunakan dalam aplikasi-aplikasi *database*.

#### 2. *INSERT*

Digunakan untuk mengisi *table* yang baru dibuat atau menambahkan baris/*record* baru pada *table* yang sudah dibuat.

#### 3. *DELETE*

Digunakan untuk menghapus sebuah baris tertentu atau sekumpulan baris dari sebuah *table*.

#### 4. *UPDATE*

Digunakan untuk mengubah nilai suatu kolom atau sekumpulan kolom pada suatu *table*.

Berikut ini merupakan pernyataan-pernyataan DDL, antara lain :

##### 1. *CREATE TABLE*

Digunakan untuk membuat *table* dengan nama kolom yang ditentukan oleh pengguna. Pengguna juga perlu mendefinisikan tipe data untuk setiap kolom. Tipe-tipe data tersebut sangat bervariasi, setiap DBMS memiliki tipe data yang berbeda dengan DBMS lainnya. Dengan demikian pengguna perlu mengetahui terlebih dahulu tipe data yang didukung oleh DBMS tersebut.

##### 2. *DROP*

Digunakan untuk menghapus semua baris/*record* dan menghapus definisi *table* dari *database*.

##### 3. *TABLE*

Digunakan untuk menghapus semua baris/*record* dan menghapus definisi *table* dari *database*.

##### 4. *ALTER TABLE*

Digunakan untuk menambah atau menghapus sebuah kolom dari suatu *table*.

## 6. *MySQL*

Fadlan (2010) MySQL adalah suatu software sistem manajemen basis data. Basis data adalah suatu koleksi data yang terstruktur. Basis data ini bisa berupa daftar belanja sederhana sampai informasi sangat besar dari suatu perusahaan internasional. Untuk menambahkan, mengakses, dan memproses data yang disimpan di komputer, diperlukan sistem manajemen basis data seperti MySQL.

Alasan penggunaan MySQL sebagai software basis data *server* adalah MySQL sangat cepat, tangguh, dan mudah digunakan. MySQL dari awal didesain untuk menangani basis data yang besar, lebih cepat dari solusi yang sudah ada dan sudah terbukti sukses beberapa tahun ini. Melalui pengembangan yang terus menerus, MySQL memiliki fungsi yang kaya dan berguna. Konektivitas, kecepatan, dan sekuriti yang baik membuat MySQL sangat cocok digunakan di internet. Cara yang benar mengucapkan MySQL adalah "My Ess Que Ell" (bukan My-SEQUEL).


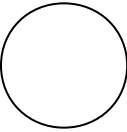
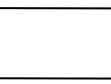
## 7. DFD

*Data Flow Diagram* yang nantinya akan disingkat dengan DFD adalah representasi dari

sebuah sistem secara grafis yang digambarkan dengan sejumlah simbol tertentu untuk menunjukkan perpindahan data dalam proses-proses suatu sistem (Jogiyanto,2005).

DFD terdiri dari empat (4) simbol.

**Tabel 1 Simbol DFD**

No	Simbol	Fungsi
1		<i>Terminator</i> / entitas luar, terminator mewakili entitas eksternal yang berkomunikasi dengan sistem yang sedang dikembangkan.
2		Proses, menggambarkan bagian dari sistem yang mentransformasikan input menjadi output.
3		<i>Data Store</i> , simpanan data yang dapat berupa file atau database, arsip, tabel acuan, agenda atau buku.

## METODE PENELITIAN

Penelitian ini menggunakan beberapa metode pengumpulan data:

1. Interview, peneliti terjun langsung ke lapangan dan melakukan wawancara kepada pemilik perusahaan serta petugas yang mengurus urusan administrasi mengenai data-data yang diperlukan dalam penelitian ini.
2. Observasi, melalui pengamatan dilapangan.
3. Studi Pustaka, dengan mempelajari literature mapun penelusuran melalui

media internet yang berkaitan dengan penelitian ini

## Metode Pengembangan Sistem

Pengembangan sistem dalam penelitian ini menggunakan *System Development Life Cycle*. Turban dalam Puranto dan Shu (2008) SDLC (*System Development Life Cycle*)

Secara konseptual siklus pengembangan sebuah sistem informasi adalah sebagai berikut :

1. Analisis Sistem: menganalisis dan mendefinisikan masalah dan kemungkinan solusinya untuk sistem informasi dan proses organisasi.
2. Perancangan Sistem: merancang *output*, *input*, struktur file, program, prosedur, perangkat keras dan perangkat lunak yang diperlukan untuk mendukung sistem informasi.
3. Pembangunan dan *Testing* Sistem: membangun perangkat lunak yang diperlukan untuk mendukung sistem dan melakukan *testing* secara akurat. Melakukan instalasi dan *testing* terhadap perangkat keras dan mengoperasikan perangkat lunak.
4. Implementasi Sistem: beralih dari sistem lama ke sistem baru, melakukan pelatihan dan panduan seperlunya.
5. Operasi dan Perawatan: mendukung operasi sistem informasi dan melakukan perubahan atau tambahan fasilitas.
6. Evaluasi Sistem: mengevaluasi sejauh mana sistem telah dibangun dan seberapa bagus sistem telah dioperasikan.

## PEMODELAN

## Bagan Alir Dokumen Sistem Berjalan

Bagan Alir Dokumen (*document flowchart*) atau disebut juga dengan Bagan Alir Formulir merupakan suatu bagan alir yang menunjukkan arus dari pelaporan dan formulir termasuk tembusan-tembusannya.

Bagan alir dokumen dari sistem yang sedang berjalan disajikan pada gambar 3

**Gambar 3 BAD yang sedang berjalan**

## Pemecahan Masalah

### Konfigurasi User Akses yang Diusulkan

*Custom user* yang dirancang untuk sistem informasi ini hanya 1 (satu) *user* akses, yang mempunyai hak akses sebagai berikut :

**Tabel 2 Level Hak Akses**

- *Level Administrator* memiliki hak akses penuh untuk mengontrol manajemen data sistem *update* data, transaksi jual beli, dan cetak laporan.

### Bagan Alir Dokumen yang Diusulkan

Bagan Alir Dokumen sistem yang diusulkan disajikan pada gambar 4.

**Gambar 4 BAD yang diusulkan**

### DFD (*Data Flow Diagram*)

DFD *Top Level* untuk sistem jual beli yang akan dibangun disajikan pada gambar 5.

**Gambar 5 DFD Top Level Sistem Jual Beli Sepeda Motor**

Keterangan DFD *top level* di atas, sebagai berikut:

1. *Order* motor, terima kuitansi
2. *Login*, *input order*, cek motor, *input* pembayaran, *input* laporan
3. Terima kuitansi, terima laporan
4. Terima laporan

Perancangan DFD di atas adalah perancangan yang mengakomodasikan kebutuhan sistem yang akan dibuat untuk DFD *level* selanjutnya.

DFD *level 0* disajikan pada gambar 6.

**Gambar 6 DFD Level 0 Sistem Jual Beli Sepeda Motor**

Level User	Hak Akses				
	C-Panel	Master Table	Master Product	Transaction	Report
Administrator	□	□	□	□	□

Keterangan proses pada gambar 4 sebagai berikut:

1. Proses 1: admin melakukan *login* aplikasi
2. Proses 2: *customer* melakukan pemesanan, admin melakukan *input* pemesanan dan mencetak kuitansi

3. Proses 3: admin melakukan cek *stock* motor dan mencetak laporan *stock* motor
4. Proses 4: admin membuat laporan dan mencetak laporan yang digunakan untuk pertanggungjawaban kepada *owner*

Pada DFD *Level 0* yang dilakukan *decomposition* ( pemilahan ) akan mendapat DFD *Level 1*. DFD *Level 1* menggambarkan tujuan kemana arah tujuan data akan dilanjutkan dan menghasilkan entitas yang dipakai dalam *database*.

Proses yang terdapat pada DFD *level 0* didekomposisikan yang kemudian masing-masing akan menjadi DFD *level 1* untuk proses transaksi, *stock*, dan laporan, sebagai berikut:

1. DFD *Level 1* Transaksi, yang memuat aktifitas admin dan pelanggan dalam proses transaksi disajikan pada gambar 7.

#### **Gambar 7 DFD *Level 1* Transaksi**

Proses 2.0: pelanggan melakukan pemesanan yang diolah oleh admin.

Proses 2.1: admin memasukan pesanan ke dalam transaksi yang menghasilkan kuitansi pembelian atau penjualan rangkap dua (2).

2. DFD *Level 1* *Stock*, yang memuat aktifitas admin untuk *update* dan cek *stock* motor disajikan pada gambar 8.

#### **Gambar 8 DFD *Level 1* *Stock***

Proses 3.0: admin melakukan cek *stock* motor.

Proses 3.1: admin melakukan *update stock* motor dan mencetak laporan *stock* motor.

3. DFD *Level 1* Laporan, yang memuat aktifitas admin dan pemilik dalam proses laporan disajikan pada gambar 9.

#### **Gambar 9 DFD *Level 1* Laporan**

Proses 4.0: admin merekap penjualan yang menghasilkan laporan penjualan rangkap dua (2) yang dilaporkan kepada pemilik.

Proses 4.1: admin merekap pembelian yang menghasilkan laporan pembelian rangkap dua (2) yang dilaporkan kepada pemilik.

Proses 4.2: admin merekap biaya perawatan yang dikeluarkan yang menghasilkan laporan operasional rangkap dua (2) yang dilaporkan kepada pemilik.

Proses 4.3: admin merekap semua transaksi yang menghasilkan laporan penjualan rangkap dua (2) yang dilaporkan kepada pemilik.

## **HASIL DAN PEMBAHASAN**

### **Pengujian Aplikasi Sistem Informasi Jual Beli**

Setelah persiapan teknis dilakukan, maka pengujian layanan berbasis SMS dapat dilakukan yaitu dengan mengetikkan <http://localhost/ruhma-motor/Admin/Login.php> pada *browser*

#### **Halaman Utama**

Setelah mengetikkan <http://localhost/ruhma-motor/Admin/Login.php> pada *web browser* maka akan tampil halaman utama seperti pada gambar 10.



### **Gambar 10. Tampilan Halaman Utama**

#### **Halaman Merk**

Halaman merk bisa diakses setelah administrator *login* sebagai admin terlebih dahulu. Halaman merk digunakan untuk mengolah data merk meliputi tambah, ubah, hapus dan menampilkan data merk. Halaman merk disajikan pada gambar 11.

### **Gambar 11 Tampilan Halaman Merk**

Keterangan dari masing-masing fungsi kontrol pada halaman merk sebagai berikut:

#### **1. Menambah data**

Administrator dapat menambah data merk dengan mengisi setiap *field* untuk keperluan pendataan merk. Proses penambahan data ini disajikan pada gambar 12.

### **Gambar 12. Tampilan Tambah Data Merk**

#### **2. Melihat data merk**

Administrator dapat melihat data merk secara otomatis setelah *input* data merk yang ditampilkan pada gambar 12 di atas.

#### **3. Mengubah data merk**

Administrator dapat mengubah data setiap merk dengan mengklik tulisan Ubah pada baris data merk yang akan diubah datanya. Proses perubahan data sama seperti proses penambahan data.

#### **4. Menghapus data merk**

Administrator dapat menghapus data setiap merk yang telah tersimpan dengan mengklik tulisan Hapus pada baris data merk yang sebelumnya memilih data mana yang akan dihapus. Apabila administrator menghapus data merk, maka data merk tersebut akan dihapus dari basisdata.

#### **Halaman Tipe Merk**

Halaman tipe merk digunakan untuk menampilkan data tipe motor berdasarkan masing-masing merk, administrator dapat menambah, mengubah, dan menghapus data tipe merk. Halaman tipe merk disajikan pada gambar 13.

### **Gambar 13. Tampilan Halaman Tipe Merk**

Keterangan dari masing-masing fungsi kontrol pada halaman tipe merk sebagai berikut:

#### **1. Menambah data**

Administrator dapat menambah data tipe merk dengan mengisi setiap *field* untuk keperluan pendataan tipe merk. Proses penambahan data ini disajikan pada gambar 14.

### **Gambar 14 Tampilan Tambah Data Tipe Merk**

#### **2. Melihat data tipe merk**

Administrator dapat melihat data tipe merk secara otomatis setelah *input* data

tipe merek yang ditampilkan pada gambar 14 di atas.

3. Mengubah data tipe merek

Administrator dapat mengubah data setiap tipe merek dengan mengklik tulisan Ubah pada baris data tipe merek yang akan diubah datanya. Proses perubahan data sama seperti proses menambah data.

4. Menghapus data tipe merek

Administrator dapat menghapus data setiap tipe merek yang telah tersimpan dengan mengklik tulisan Hapus pada baris data tipe merek yang sebelumnya memilih data mana yang akan dihapus. Apabila administrator menghapus data tipe merek, maka data tipe merek tersebut akan dihapus dari basisdata.

#### **Halaman Jenis Biaya**

Halaman ini menampilkan daftar jenis biaya operasional selama motor belum terjual. Satu motor bisa memiliki beberapa jenis biaya operasional yang dapat membantu admin untuk mengetahui total biaya yang dikeluarkan selama perawatan. Halaman jenis biaya disajikan pada gambar 15.

#### **Gambar 15 Tampilan Halaman Jenis Biaya**

Keterangan dari masing-masing fungsi kontrol pada halaman jenis sebagai berikut:

1. Menambah data

Administrator dapat menambah data jenis biaya dengan mengisi setiap *field* untuk keperluan pendataan jenis biaya

operasional. Proses penambahan data ini disajikan pada gambar 16.

#### **Gambar 16. Tampilan Tambah Data Jenis Biaya**

2. Melihat data jenis biaya

Administrator dapat melihat data jenis biaya secara otomatis setelah *input* data jenis biaya yang ditampilkan pada gambar 16.

3. Mengubah data jenis biaya

Administrator dapat mengubah data jenis biaya dengan mengklik tulisan Ubah pada baris data jenis biaya yang akan diubah datanya. Proses perubahan data sama seperti proses menambah data.

4. Menghapus data jenis biaya

Administrator dapat menghapus data setiap jenis biaya yang telah tersimpan dengan mengklik tulisan Hapus pada baris data jenis biaya yang sebelumnya memilih data mana yang akan dihapus. Apabila administrator menghapus data jenis biaya, maka data jenis biaya tersebut akan dihapus dari basisdata.

#### **Halaman Pelanggan**

Halaman pelanggan digunakan untuk mengolah data pelanggan. Administrator dapat melakukan *input* dan mengubah data pelanggan. Untuk mengubah data mahasiswa yaitu dengan mengganti data pelanggan yang telah tersimpan dan data lama akan secara otomatis diganti oleh data pelanggan baru. Halaman pelanggan disajikan pada gambar 17.

### **Gambar 17. Tampilan Halaman Pelanggan**

Keterangan dari masing-masing fungsi kontrol pada halaman jenis sebagai berikut:

1. Menambah data  
Administrator dapat menambah data pelanggan dengan mengisi setiap *field* untuk keperluan pendataan pelanggan. Proses penambahan data ini disajikan pada gambar 18.

### **Gambar 18. Tampilan Tambah Data Pelanggan**

2. Melihat data pelanggan  
Administrator dapat melihat data pelanggan secara otomatis setelah *input* data pelanggan yang ditampilkan pada gambar 18.
3. Mengubah data pelanggan  
Administrator dapat mengubah data pelanggan dengan mengklik tulisan Ubah pada baris data pelanggan yang akan diubah datanya. Proses perubahan data sama seperti proses penambahan data.
4. Menghapus data pelanggan  
Administrator dapat menghapus data setiap pelanggan yang telah tersimpan dengan mengklik tulisan Hapus pada baris data pelanggan yang sebelumnya memilih data mana yang akan dihapus. Apabila administrator menghapus data

pelanggan, maka data pelanggan tersebut akan dihapus dari basis data.

### **Halaman Motor**

Halaman motor digunakan untuk mengolah data motor. Administrator dapat melakukan *input* dan mengubah data motor. Untuk mengubah data motor yaitu dengan mengganti data motor yang telah tersimpan dan data lama akan secara otomatis diganti oleh data baru. Halaman motor disajikan pada gambar 19.

### **Gambar 19. Tampilan Halaman Motor**

Penambahan data motor dapat dilakukan oleh administrator. Dalam proses penambahan data motor terdapat *field* harga beli dan harga jual. Motor yang baru dibeli berarti cukup *field* harga beli yang dijual, untuk *field* harga jual dapat dikosongi terlebih dahulu. Halaman *input* motor disajikan pada gambar 20.

### **Gambar 20. Tampilan Halaman Tambah Data Motor**

Administrator dapat mengubah data motor dengan mengklik tulisan Ubah pada baris data motor yang akan diubah datanya. Proses perubahan data sama seperti proses penambahan data. Proses perubahan data biasa dilakukan jika terdapat motor yang terjual.

Pada halaman motor terdapat pilihan Biaya Operasional yang digunakan untuk *input* data pengeluaran selama motor dibeli sampai terjual. Setiap kali ada pembiayaan untuk motor maka administrator harus menambahkan data biaya perawatan dengan cara klik Tambah pada

baris data biaya operasional, kemudian pilih nama biaya dan mengisi nilai biaya yang dikeluarkan. Tampilan biaya perawatan disajikan pada gambar 21.

#### **Gambar 21. Tampilan Halaman Biaya Perawatan**

Data pada biaya perawatan dapat dihapus dengan cara memilih data mana yang akan dihapus kemudian klik Hapus pada baris data biaya operasional.

Pada halaman motor administrator dapat melakukan cetak kuitansi pembelian dan penjualan. Masing-masing kuitansi rangkap dua (2), satu untuk arsip perusahaan dan yang satu untuk pelanggan. Tampilan kuitansi pembelian disajikan pada gambar 22

#### **Gambar 22. Tampilan Halaman Cetak Kuitansi Pembelian**

Tampilan kuitansi penjualan disajikan pada gambar 23.

#### **Gambar 23. Tampilan Halaman Cetak Kuitansi Penjualan**

##### **Halaman Report**

Halaman *report* digunakan untuk mengetahui jenis laporan yang dihasilkan oleh aplikasi. Pada halaman *report* ini terdapat lima (5) jenis *report* yaitu:

1. *Report stock* motor, untuk mengetahui motor apa saja yang belum terjual.

2. *Report* penjualan, untuk mengetahui motor apa saja yang sudah terjual dan harga jual.
3. *Report* pembelian, untuk mengetahui motor apa saja yang sudah dibeli dan harga beli.
4. *Report* operasional, untuk mengetahui biaya perawatan yang dikeluarkan untuk masing-masing motor.
5. *Report* rugi laba, untuk mengetahui keuntungan atau kerugian dalam transaksi jual beli yang sudah dilakukan.

Tampilan halaman *report* disajikan pada gambar 24.

#### **Gambar 24. Tampilan Halaman Report**

Tampilan halaman laporan *stock* motor disajikan pada gambar 25.

#### **Gambar 25 Tampilan Laporan Stock Motor**

Tampilan halaman laporan penjualan motor di bulan Mei 2011 disajikan pada gambar 26.

#### **Gambar 26. Tampilan Laporan Penjualan**

Tampilan halaman laporan pembelian motor di bulan Mei 2011 disajikan pada gambar 27.

#### **Gambar 27 Tampilan Laporan Pembelian**

Tampilan halaman laporan operasional motor di bulan Mei 2011 disajikan pada gambar 28.

### **Gambar 28 Tampilan Laporan Operasional**

Tampilan halaman laporan rugi laba motor di bulan Mei 2011 disajikan pada gambar 29

### **Gambar 29 Tampilan Laporan Rugi Laba**

## **KESIMPULAN DAN SARAN**

### **Kesimpulan**

1. Sistem informasi jual beli berbasis *web* dapat membantu pemilik usaha untuk tetap memantau pengembangan perusahaan meskipun tidak berada di tempat usaha.
2. Data motor dan pelanggan dapat selalu terkontrol dan disimpan dengan rapi di dalam basisdata yang membantu admin untuk membuat laporan.
3. MotorID merupakan kunci primer yang bersifat unik. Laporan rugi laba yang dicetak mencakup nilai transaksi jual beli motor berdasarkan MotorID.
4. Nomor kuitansi pembelian dan penjualan memiliki kode yang berbeda. Jika nomor kuitansi dihapus dan menambah transaksi baru maka nomor tersebut tidak akan digunakan kembali.

### **Saran**

1. Untuk pengembangan lebih lanjut sebaiknya dikembangkan untuk sistem jual beli berbasis *e-commerce* atau penjualan motor secara *online* dengan menambah halaman *guest* yang lebih menarik dengan berbagai informasi yang membangun perusahaan.
2. Untuk kedepannya sebaiknya aplikasi ini bisa diterapkan juga pada sistem penggajian dan presensi karyawan.
3. Pengembangan jual beli yang sudah berjalan dapat dilakukan secara kredit dengan kerja sama dari salah satu koperasi dan menambahkan halaman kredit yang membantu admin dan pemilik.
4. Aplikasi ini sebaiknya menyediakan *link* alamat *website* rekan kerja atau dealer motor yang selama ini sudah menjalin kerjasama.

## **DAFTAR PUSTAKA**

- Deptan. 2011. *Materi PHP (online)*. [http://www.deptan.go.id/daerah\\_new/data/Materi%20PHP.doc](http://www.deptan.go.id/daerah_new/data/Materi%20PHP.doc). Diakses tanggal 28 Maret 2011 pk1 19:39.
- Eepisits. 2008. *DFD (online)*. [http://lecturer.eepisits.edu/~arna/Praktikum\\_RPL/DFD.pdf](http://lecturer.eepisits.edu/~arna/Praktikum_RPL/DFD.pdf). Diakses tanggal 2 Juni 2011 pk1 12:59.
- Fahd, dkk. 2007. *Skripsi Analisa dan Perancangan Sistem Pemasaran Mobil Bekas pada Showroom Arjuna Motor dengan Menggunakan Website*. Universitas Bina Nusantara <http://thesis.binus.ac.id> (download tanggal 3 April 2012)

Jogiyanto. 2005. *Metodologi Penelitian Sistem Informasi*. Yogyakarta: CV. Andi OFFSET.

Kadir, Abdul. 2004. *Dasar Aplikasi Database MySQL-Delphi*. Yogyakarta: CV. Andi OFFSET.

Universitas Petra. 2006. *Bab III Dasar-dasar teori* (online).  
[http://pcucamel.petra.ac.id/free/Bab%20III%20%20Dasardasar%20teori1\(mdifed\).doc](http://pcucamel.petra.ac.id/free/Bab%20III%20%20Dasardasar%20teori1(mdifed).doc). Diakses tanggal 28 Maret 2011 pk1 19:04.

Universitas Brawijaya. 2011. *DFD* (online).  
<http://power.lecture.ub.ac.id/files/2011/04/DFD.pdf>. Diakses tanggal 2 Juni 2011 pk1 13:04.

Unikom. 2011. *Penjualan* (online).  
<http://elib.unikom.ac.id/download.php?id=1886>. Diakses tanggal 25 Maret 2011 pk1. 04:22.