

# APLIKASI JAVA DAN MYSQL PADA SISTEM INFORMASI MINIMARKET

David Robby Sanjaya, Yuniarto  
Program Studi Diploma III Teknik Elektro  
Fakultas Teknik Universitas Diponegoro

## ABSTRACT

*David Robby Sanjaya, Yuniarto, in paper java and mysql application at minimarket information system explain that an endless human needs combined by their daily activities, made a lot of people reluctant to do shopping at a traditional market. Minimarket or supermarket that finally became a way out for this problem, because it had a good management system for organized the items and the transactions. Regulation of items and money at minimarket or supermarket require a good software. There are a lot of programming language that can be used for create this software, one of that is Java programming language. Java in this information system be used for create an interface to user while Mysql be used to manage a databases. Java programming language, Mysql, Netbeans and iReport are tools that applied to this Minimarket information system.*

*Keywords: Minimarket Information System, iReport, Java, Mysql, Netbeans.*

## PENDAHULUAN

Perkembangan teknologi informasi mendorong manusia untuk mengatasi berbagai masalah yang timbul disekitarnya dengan tujuan untuk mempermudah pekerjaan dan efisiensi waktu. Supermarket, yang merupakan pasar modern, dimana sekarang ini banyak orang lebih memilih supermarket daripada pasar tradisional karena sistem pembelian di supermarket yang lebih praktis. Sistem yang diterapkan di supermarket tersebut juga tidak lepas dari perkembangan teknologi informasi yang ada. Di dalam supermarket terdapat banyak sekali barang, baik yang masih di gudang maupun yang sudah siap jual. Supaya arus barang bisa terpantau, maka perlu diperhitungkan antara barang yang masuk dan barang yang keluar, baik itu barang dari gudang maupun barang terjual. Dengan demikian akan terdata jumlah barang yang ada berikut dengan arus transaksi yang terjadi.

## TINJAUAN PUSTAKA

### Database dengan Mysql

*Database* adalah koleksi data item yang saling terkait terkelola sebagai satu unit. Istilah “*database*” juga digunakan di dunia perpustakaan untuk menjelaskan sekumpulan informasi yang biasanya tersimpan dan dapat diakses di perpustakaan. Untuk membuat definisi *database* yang kita maksud lebih jelas dan tepat, maka beberapa karakteristik dari *database* adalah :

- Pengelolaan melalui dengan suatu *Database Management System (DBMS)*
- Ada lapisan-lapisan abstraksi data
- Data fisik yang independen
- Data logis yang independen

### DBMS

*Database Management System (DBMS)* merupakan *software* yang disediakan oleh penyedia *database* untuk :

- Mengelola dan memelihara data
- Memindahkan data ke dan dari file data fisik yang dibutuhkan
- Mengelola akses data yang bersamaan oleh beberapa pengguna
- Mengawasi pembaharuan data dan mencegah konflik perubahan data
- Mengelola transaksi sehingga perubahan data terjadi secara lengkap atau tidak terjadi perubahan jika transaksi batal atau gagal.
- Mendukung bahasa *query*
- Pengawasan *back up database* dan pemulihan dari kesalahan
- Mekanisme keamanan

### Lapisan-lapisan abstraksi data (Layers of Data Abstraction)

*Database* memiliki kapabilitas unik unik untuk menyajikan informasi ke banyak pengguna sesuai dengan permintaan masing-masing. Pengguna di sini bisa saja manusia atau aplikasi yang masuk secara sah ke *database* untuk menyimpan atau mendapatkan data. Aplikasi tersebut adalah program yang dirancang untuk menyelesaikan tugas-tugas manajemen tertentu seperti sistem arus transaksi barang dan uang pada system informasi minimarket. *Database* memungkinkan setiap pengguna (manusia atau aplikasi) mendapatkan informasi hasil pengolahan data yang berbeda satu sama lain. Yang satu minta laporan pendapatan bulan tertentu sementara yang lain mencetak transaksi hari ini,

pengguna lain melakukan pencarian terhadap data tertentu. Masing-masing menggunakan data dari database yang sama. Setiap perubahan data yang dilakukan oleh salah satu, akan juga ditemukan oleh yang lain.

Lapisan-lapisan abstraksi data dapat digambarkan seperti pada Gambar 1. dimana ada 3 lapisan yaitu lapisan fisik, lapisan logika dan lapisan luar.

- Lapisan Fisik

Lapisan fisik berisi file data yang menyimpan semua data untuk *database*. Hampir semua DBMS yang modern mengizinkan *database* disimpan dalam *multiple* data files (banyak file untuk 1 *database*). Kecuali MS. Access menyimpan database dalam 1 file sehingga tidak mampu mengatasi ukuran data yang besar, tapi cocok untuk data personal. DBMS dan sistem operasi secara otomatis mengelola data files, yaitu membuka, menutup, membaca dan menulis data ke dan dari file. Pengguna database tidak perlu langsung berhubungan dengan lapisan fisik yaitu data files saat menggunakan database. Data-data yang tersimpan dalam file diwakili oleh lapisan berikutnya yaitu lapisan logika.

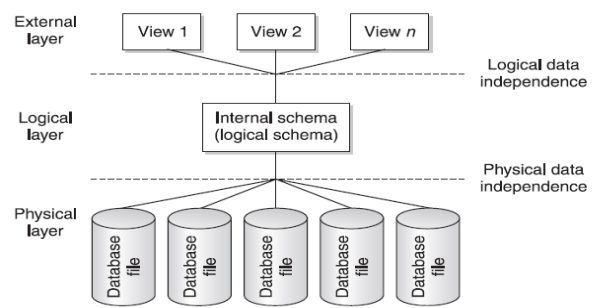
- Lapisan Logika

Lapisan ini merupakan lapisan abstraksi pertama dalam database. lapisan fisik bukan lapisan abstraksi karena datanya nyata, bukan abstrak. Lapisan logika terdiri dari struktur data abstrak yang sama dengan lapisan data. DBMS mengubah data di data file menjadi struktur/susunan yang sesuai. Lapisan ini sering disebut skema, istilah yang digunakan untuk koleksi semua data yang tersimpan dalam database.

Pada lapisan inilah struktur tabel database yang disusun oleh manusia dan kemudian data dalam table-tabel itulah yang kemudian diakses oleh manusia menggunakan bahasa *query* yang ada di lapisan luar.

- Lapisan Luar

Lapisan luar atau mmodel luar adalah layar kedua abstrak dalam database. Lapisan ini terdiri dari tampilan pengguna yang disebut subskema. Lapisan ini adalah tempat dimana pengguna dan aplikasi yang mengakses database berhubungan dan menghasilkan *query* ke database.



Gambar 1. Layers of Data Abstraction

### Data fisik yang independen (*Physical Data Independence*)

Kemampuan mengubah struktur file database secara fisik tanpa mengganggu pengguna dan proses yang berlaku disebut *physical data independence*. Seberapa besar perubahan yang dapat dilakukan tanpa berpengaruh pada lapisan logika disebut derajat *physical data independence*.

### Data logis yang independen (*Logical Data Independence*)

Kemampuan untuk membuat perubahan pada lapisan logika tanpa mengganggu penggunaan dan proses yang sedang berlaku disebut *logical data independence*. Perubahan pada lapisan logika juga berlaku pada lapisan fisik. Misalnya penambahan data di lapisan logika, berarti menambah data pada lapisan fisik. Perubahan format data pada lapisan logika akan berdampak pada lapisan fisik juga. Perubahan yang dimaksud termasuk penghapusan data.

### Model Database yang Umum

- Flat Files

Flat files adalah file sistem operasi yang record dalam filenya tidak berisi informasi tentang struktur file atau hubungan antar record yang dikomunikasikan ke aplikasi yang menggunakannya. Flat file bukan database karena tidak masuk dalam kriteria di atas yang sudah dibahas.

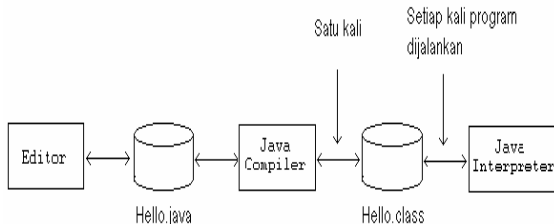
- Model Hirarki

Model ini menyusun *record-record* dalam suatu hirarki seperti struktur organisasi. Setiap file dari flat file menjadi tipe record atau node dalam hirarki, dan untuk memudahkan disebut *record*. *Record-record* terhubung melalui pointer yang berisi alamat record yang terkait. Pointer memberi tahu sistem komputer dimana record tersebut tersimpan secara fisik, seperti alamat yang mengacu ke suatu gedung tertentu di dunia nyata.

- **Model Relational**  
Model ini berusaha mengatasi masalah pada model sebelumnya yang tidak fleksibel. Model relational database mampu menghubungkan record-record yang dibutuhkan. Model ini dibangun sedemikian rupa sehingga permintaan atau query dapat bekerja dengan sekumpulan data, tidak dengan satu record seperti pada model hirarki. Model relational menyajikan data dalam tabel dua dimensi, seperti spreadsheet hanya saja tidak dipresentasikan dalam tabel seperti spreadsheet tetapi menggunakan kombinasi atau *joining relational* tabel untuk menghasilkan view atau hasil tampilan.
- **Model Object-Oriented**  
Model ini berusaha menjawab kesulitan RDBMS (*Relational Database Management System*) untuk mengatasi tipe data yang kompleks seperti file citra, file gambar dan file audio-video. Ini semua karena adanya Internet dan WWW yang memungkinkan pengiriman data yang kompleks itu. Object adalah pengelompokan logis dari data yang berelasi dan program logis yang mewakili hal nyata di dunia seperti barang, harga, type dalam sistem persediaan barang.

### Fase-fase Pemrograman Java

Gambar 2. menjelaskan aliran proses kompilasi dan eksekusi sebuah program Java.



Gambar 2. Fase dari sebuah Program Java

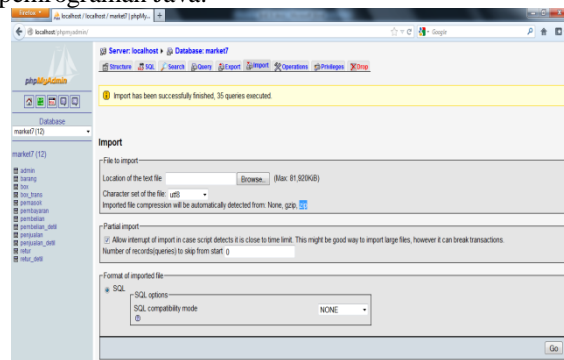
Langkah pertama dalam pembuatan sebuah program berbasis Java adalah menuliskan kode program pada *text editor*. Contoh *text editor* yang dapat digunakan antara lain : notepad, vi, emacs dan lain sebagainya. Kode program yang dibuat kemudian tersimpan dalam sebuah berkas berekstensi *.java*.

Setelah membuat dan menyimpan kode program, kompilasi file yang berisi kode program tersebut dengan menggunakan Java Compiler. Hasil dari kompilasi berupa berkas *bytecode* dengan ekstensi *.class*. Berkas yang mengandung *bytecode* tersebut kemudian akan dikonversikan oleh Java Interpreter menjadi bahasa mesin sesuai dengan jenis dan *platform* yang digunakan.

## HASIL DAN PEMBAHASAN

### Implementasi Sistem

Tahapan Implementasi Sistem mencakup pengkodean program, pengujian program, pemasangan program, serta pelatihan kepada pengguna. Program basis data yang telah dirancang dimulai dari diagram alir yang berguna untuk menjelaskan dan mempermudah dalam merancang serta merupakan implementasi dari program yang akan dibuat. Diagram alir data akan membuat logika berfikir pembuatan program tersusun dengan baik berdasarkan urutan yang telah direncanakan. Dari diagram alir data yang dibuat pada rancangan sistem, pada tahap implementasi sistem ini akan diterjemahkan ke dalam kata-kata bahasa pemrograman dengan menggunakan bahasa pemrograman Java.



Gambar 3. Implementasi Database

Gambar 3. menunjukkan implementasi database pada MySQL yang memiliki 12 tabel, yaitu:

- Tabel admin
- Tabel barang
- Tabel box
- Tabel box\_trans
- Tabel pemasok
- Tabel pembayaran
- Tabel pembelian
- Tabel pembelian\_detil
- Tabel penjualan
- Tabel penjualan\_detil
- Tabel retur
- Tabel retur\_detil

### Cara Kerja Program

Saat program dijalankan, maka tampilan awal adalah form login dimana hanya user terdaftar yang sudah memiliki username dan password yang bisa menjalankan program ini. Setelah melakukan pengisian pada form login, menu yang berikutnya muncul adalah menu utama. Di menu utama inilah semua kegiatan transaksi atau proses pengkasiran bisa dilakukan. Pada menu utama ini terdapat pilihan menu-menu yang lain seperti inventori, peralatom,

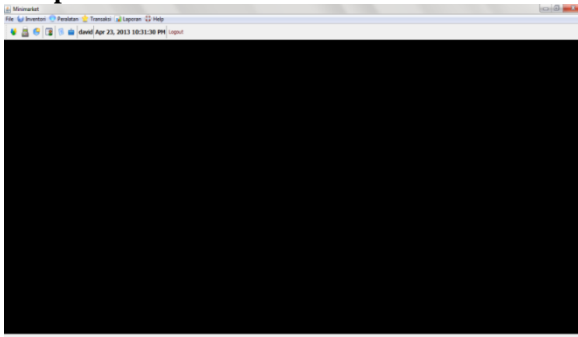
transaksi, laporan, dll. Akan tetapi tidak semua user bisa mengakses menu-menu tersebut, karena pada proses pembuatan user, terdapat pilihan *privileges*. Jadi hanya menu-menu sesuai *privileges* yang bisa diakses oleh user.

### Tampilan Form Login

Gambar 4. Tampilan Form Login

Gambar 4. menunjukkan tampilan awal dari program, yaitu form login. jadi, saat program dijalankan maka yang pertama muncul adalah form login. Pada form ini user harus memasukkan User ID dan password yang tepat untuk bisa masuk pada menu utama.

### Tampilan Menu Utama



Gambar 5. Pengujian Menu Utama

Gambar 5. di atas menunjukkan tampilan menu utama. Pada bagian atas menu utama terdapat beberapa menu bar, diantaranya: File, Inventori, Peralatan, Transaksi, Laporan dan Help. Dan tiap-tiap menu bar tersebut masih memiliki sub menu seperti box, admin, pembelian, penjualan, dll. Tampilan lain selain menu bar adalah adanya *shortcut* untuk memudahkan *user* mengakses menu yang sering digunakan, yaitu admin, box, barang, pemasok, pembelian dan penjualan. Diikuti dengan ID *user* yang sedang menggunakan program kemudian waktu dan tanggal dan yang terakhir adalah fasilitas *log out* yang berfungsi untuk keluar dari program dan langsung menampilkan form login.

### KESIMPULAN

Setelah melakukan perancangan, implementasi dan pengujian sistem informasi, maka penulis dapat memberi kesimpulan sebagai berikut :

- Sistem informasi minimarket ini merupakan pengaplikasian bahasa pemrograman Java dan MySQL.
- Peng-*input*-an barang yang terjadi bisa dilakukan dengan 2 cara, yaitu *scanning* dengan *barcode scanner* dan mencari secara manual kode barang yang akan dimasukkan.
- Struk penjualan masih berupa laporan penjualan yang harus di *print* secara manual dan belum otomatis.
- Pada sistem informasi minimarket ini masih belum tersedia pendataan secara detail mengenai pergantian kasir sehingga tidak bisa diketahui kasir yang bertanggung jawab untuk setiap transaksi yang terjadi.

### DAFTAR PUSTAKA

1. A Suhendar dan Hariman Gunadi. 2002. **Visual Modelling Menggunakan UML dan Rational Rose**. Bandung: Informatika Bandung.
2. IAN Somerville. 2002. **Rekayasa Perangkat Lunak**. Jakarta: PT Elex Media Komputindo.
3. JENI. **Java Fundamental All Bab**. <https://jeni.or.id/jeni1-oop> (17 Juni 2011).
4. Miftakhul Huda dan Bunafit Komputer. 2008. **Membuat Aplikasi Penjualan dengan Java dan MySQL**. Jakarta: PT Elex Media Komputindo.
5. Miftakhul Huda dan Bunafit Komputer. 2009. **Membuat Aplikasi Rental dengan Java dan MySQL**. Jakarta: PT Elex Media Komputindo.
6. Miftakhul Huda dan Bunafit Komputer. 2010. **Membuat Aplikasi Database dengan Java, MySQL dan Netbeans**. Jakarta: PT Elex Media Komputindo.
7. Miftakhul Huda dan Bunafit Komputer. 2010. **Trik Rahasia Pemrograman Database dengan Java**. Jakarta: PT Elex Media Komputindo.
8. Miftakhul Huda dan Bunafit Komputer. 2010. **Aplikasi Inventory Multi Store Plus Management Database dengan Java**. Jakarta: PT Elex Media Komputindo.
9. Oppel, Andy. 2004. **Databases Demystified: A self-teaching guide**.