

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Pengantar

Tinjauan pustaka menguraikan teori, temuan, dan penelitian lain yang diperoleh untuk dijadikan dasar dalam melakukan kegiatan penelitian. Dalam tinjauan pustaka ini akan dibahas mengenai beberapa hal, yaitu:

1. Manajemen persediaan
2. Sistem informasi *database*
3. Metode Perancangan Sistem *Database*
4. *Software* pendukung yang digunakan dalam penelitian ini

#### 2.2 Penelitian yang Relevan

Suryani dan Eka (2006) melakukan perancangan *database telecenter* berbasis *web* di BPDE Jatim. Dengan adanya perancangan *database* ini BPDE mampu menyediakan dan menyebarkan informasi untuk pemerintah dan masyarakat dengan menggunakan sistem informasi dan telematika dalam rangka terciptanya budaya informasi. Perancangan ini dibuat dengan menggunakan *database MySQL*, serta PHP karena dapat mengerjakan semua yang dapat dikerjakan oleh program CGI, seperti mendapatkan data dari *form*, menghasilkan isi halaman *web* yang dinamik dan menerima *cookies*.

Widhyaestoeti (2011) melakukan perancangan *database* untuk pengelolaan nilai siswa di sebuah sekolah tingkat menengah. Untuk meraih prestasi belajar yang baik dapat dilihat dari nilai yang dihasilkan seorang siswa dari hasil proses belajar yang telah dijalani sehingga data nilai tersebut perlu dikelola dengan baik. Dengan adanya perancangan ini, data-data administrasi nilai siswa menjadi lebih terintegrasi dan terorganisir dengan baik karena memiliki pangkalan data.

Robby dkk. (2009) melakukan perancangan sebuah basis data pada bagian *Academic Management and Content Preparation* yang terintegrasi untuk setiap jenjang yang ada (S1, S2, dan BI). Metodologi perancangan basis data yang digunakan adalah perancangan *Semantic Object Modelling (SOM)* yang diawali dengan *form*, *report*, dan *query* kemudian pembuatan *Semantic Object Model* lalu transformasi ke bentuk *Relational Model*. Basis data yang terintegrasi ini akan dapat mendukung aplikasi *ERP*

(*Enterprise Resources Planning*) Education terutama dalam hal kemudahan penggunaan karena cukup dengan satu aplikasi dapat digunakan oleh semua jenjang yang ada.

### 2.3 Manajemen Persediaan

Setiap perusahaan, baik jasa ataupun perusahaan manufaktur, selalu memerlukan persediaan. Tanpa adanya persediaan, para pengusaha akan dihadapkan resiko bahwa perusahaannya pada suatu waktu tidak dapat memenuhi keinginan para pelangganya. Persediaan ini diadakan apabila keuntungan yang diharapkan dari persediaan tersebut terjamin kelancarannya. Persediaan adalah suatu hal yang sangat penting karena akan berpengaruh terhadap sistem produksi yang berlangsung.

Persediaan adalah sumber daya menganggur (*idle resource*) yang menunggu proses lebih lanjut tersebut adalah berupa kegiatan produksi pada sistem manufaktur, kegiatan pemasaran pada sistem distribusi ataupun kegiatan konsumsi pangan pada sistem rumah tangga. (Hakim, 1999:95). Selain itu persediaan merupakan salah satu unsur aktif dalam operasi perusahaan yang secara kontinu diperoleh, diubah kemudian dijual kembali.

Menurut (Rangkuti, 2004:5), setiap jenis persediaan memiliki karakteristik tersendiri dan cara pengelolaan yang berbeda. Persediaan dapat dibedakan :

1. Persediaan bahan mentah (*raw material*) yaitu persediaan barang-barang berwujud, seperti besi, kayu, serta komponen-komponen lain yang digunakan dalam proses produksi.
2. Persediaan komponen – komponen rakitan (*purchased parts components*), yaitu persediaan barang-barang yang terdiri dari dari komponen-komponen yang berasal dari perusahaan lain dimana secara langsung dapat dirakit menjadi suatu produk
3. Persediaan bahan pembantu atau penolong (*supplies*), yaitu persediaan barang-barang yang diperlukan dalam proses produksi, tetapi tidak merupakan bagian atau komponen barang jadi.
4. Persediaan barang dalam proses (*work in proses*), yaitu persediaan barang-barang yang merupakan keluaran dari tiap-tiap bagian dalam proses produksi atau yang telah diolah menjadi suatu bentuk, tetapi masih perlu diproses lebih lanjut menjadi barang jadi.

5. Persediaan barang jadi (*finished goods*), yaitu persediaan barang-barang yang telah selesai diproses atau diolah dalam pabrik dan siap untuk dipakai atau dijual kepada pelanggan atau konsumen.

## 2.4 Metode Pengendalian Persediaan

Model pengendalian persediaan dapat didefinisikan dalam 3 metode yaitu metode tradisional, metode perencanaan kebutuhan material (MRP) dan metode kanban. Dalam metode pengendalian secara tradisional menggunakan matematika dan statistika sebagai alat utama dalam memecahkan masalah. Dalam metode pengendalian tradisional berusaha mencari jawaban optimal dalam menentukan :

1. jumlah pemesanan ekonomis
2. jumlah pesanan kembali (*Reorder Point*)
3. jumlah cadangan pengamanan (*safety stock*) yang diperlukan

Metode ini sering disebut juga persediaan secara statistik biasanya digunakan untuk mengendalikan barang yang permintaannya bersifat bebas (*dependent*) dan dikelola saling tidak bergantung. Yang dimaksud permintaan bebas adalah permintaan yang hanya dipengaruhi mekanisme pasar sehingga bebas dari fungsi operasi produksi.

Ditinjau dari sejarah perkembangannya metode ini secara formal diperkenalkan oleh Wilson pada tahun 1929 dengan mencoba mencari jawaban 2 pertanyaan dasar, yaitu :

1. Berapa jumlah barang yang harus dipesan untuk setiap kali pemesanan?
2. Kapan saat pesan yang harus dilakukan?

Pengembangan formula Wilson kemudian dikembangkan pada keadaan yang lebih realistik, terutama untuk fenomena yang bersifat probabilistik. Hal ini kemudian memunculkan dua metode yaitu :

1. **Metode P**, yang menganut aturan bahwa saat pemesanan bersifat regular mengikuti suatu periode yang tetap, sedangkan kuantitas pemesanan akan berulang-ulang
2. **Metode Q**, yang menganut aturan bahwa jumlah ukuran pemesanan (kuantitas pemesanan) selalu tetap untuk setiap kali pesan, sehingga saat pemesanan dilakukan akan bervariasi.
3. Diantara kedua metode P dan Q terdapat metode gabungan P dan Q (Hakim, 1999:100)

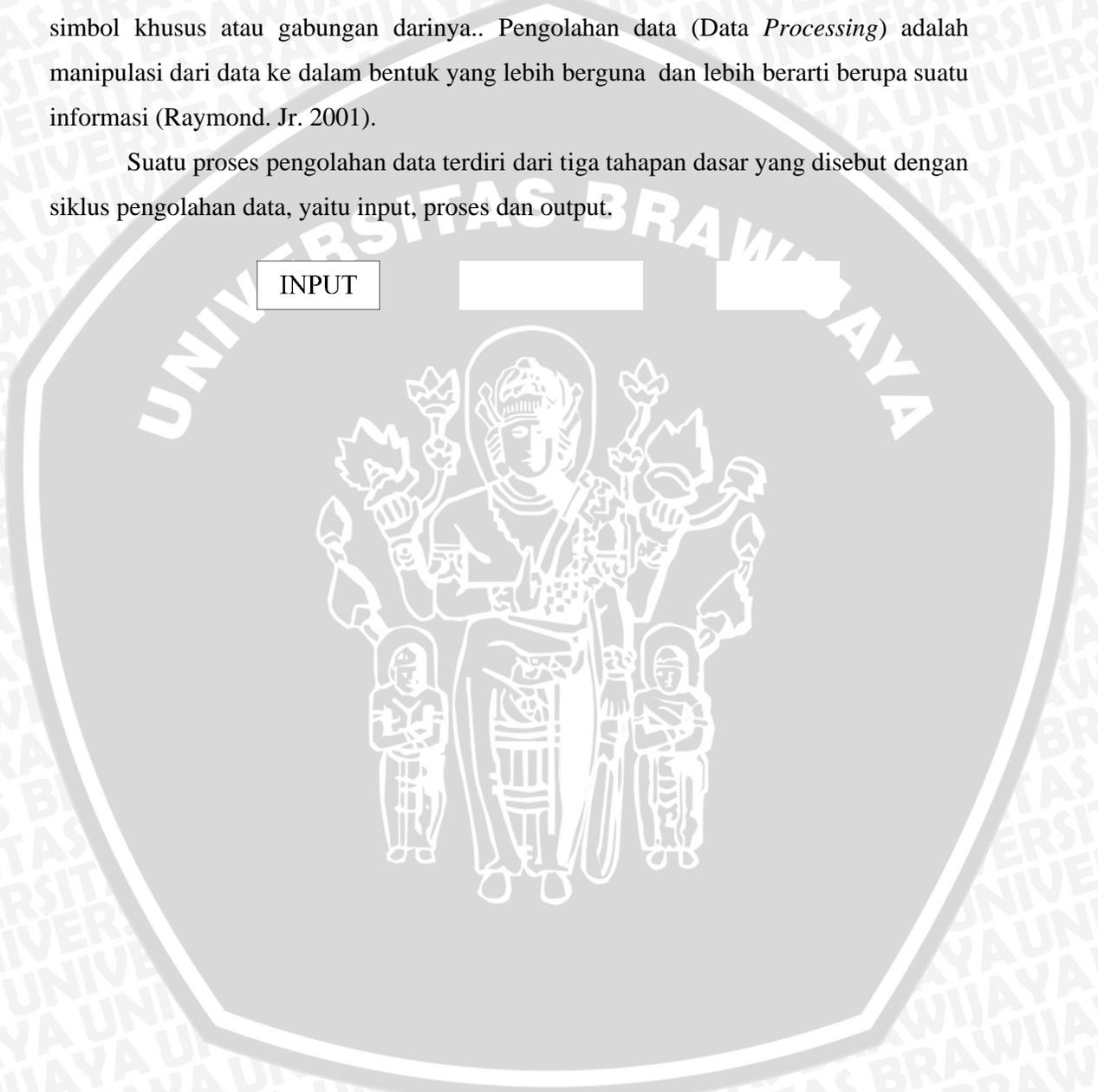
## 2.5 Database

### 2.5.1 Pengertian Data

Menurut kamus bahasa, data diterjemahkan sebagai istilah yang berasal dari kata “*datum*” yang berarti fakta atau bahan-bahan keterangan kumpulan kejadian yang diangkat dari suatu kenyataan. Data dapat berupa angka-angka, huruf, atau simbol-simbol khusus atau gabungan darinya.. Pengolahan data (*Data Processing*) adalah manipulasi dari data ke dalam bentuk yang lebih berguna dan lebih berarti berupa suatu informasi (Raymond. Jr. 2001).

Suatu proses pengolahan data terdiri dari tiga tahapan dasar yang disebut dengan siklus pengolahan data, yaitu input, proses dan output.

INPUT



2. *Byte*, adalah bagian terkecil yang dapat dialamatkan dalam memori. *Byte* merupakan sekumpulan dari *bit* yang secara konvensional terdiri atas kombinasi delapan bit. Satu *byte* digunakan untuk mengkodekan satu buah karakter dalam memori.

Contoh: kode Ascii untuk J ialah 10101010

3. *Field*, adalah unit terkecil yang disebut data. *Field* merupakan sekumpulan *byte* yang mempunyai makna

Contoh: Joni merupakan *field* nama

4. *Record*, adalah kumpulan item yang secara *logic* saling berhubungan. Setiap *record* dapat dikenali oleh sesuatu yang mengenalinya, yaitu *field* kunci

Nama	kursus	Grade
Joni	IS101	A

Gambar 2.2. Contoh *Record*

5. *File*, adalah kumpulan *record* yang sejenis dan secara *logic* berhubungan. Pembuatan dan pemeliharaan *file* adalah faktor yang sangat penting dalam sistem informasi manajemen yang memakai komputer.

Nama	kursus	Grade
Joni	IS101	A
Karen	IS101	B
Emily	IS101	B+

Gambar 2.3. Contoh *File*

6. *Database*, adalah merupakan kumpulan *file-file* yang berhubungan secara logis dan digunakan secara rutin pada operasi sistem informasi manajemen. Semua database umumnya berisi elemen-elemen data yang disusun ke dalam *record* dan *file-file* dengan berbagai cara memenuhi kebutuhan informasi bagi pemakainya.

File  
Kursus



program aplikasi. Data yang disimpan dalam *database* tergantung dari program aplikasi yang menggunakannya dan tipe media penyimpanannya”.

Kemudian, pengertian sistem *database* sendiri adalah sistem yang terdiri atas kumpulan *file* (tabel) yang saling berhubungan (dalam sebuah *database* di sebuah sistem komputer) dan sekumpulan program (DBMS) yang memungkinkan beberapa pemakai dan/atau program lain untuk mengakses dan memanipulasi *file-file* (tabel-tabel) tersebut (Fathansyah, 1999:10).

### 2.6.2 Tujuan Sistem Database

Tujuan sistem *database* antara lain:

1. Isolasi data

Menempatkan setiap data pada tempatnya masing-masing.

2. Multi User

Saat perusahaan mengambil pendekatan berorientasi masalah, pertama masalah didefinisikan. Kemudian keputusan yang diperlukan untuk memecahkan masalah didefinisikan dan untuk tiap keputusan definisikan informasi yang diperlukan. Selanjutnya, pemrosesan yang diperlukan untuk menghasilkan informasi ditentukan dan akhirnya data yang diperlukan pemrosesan ditetapkan.

Tujuan dari manajemen *database* adalah sebagai berikut:

1. Menyediakan tempat penyimpanan massal untuk data yang *relevan*.
2. Memudahkan pemakai dalam mengakses data.
3. Memungkinkan respon yang segera atas permintaan data dari pemakai.
4. Melakukan modifikasi terakhir dengan segera pada *database*.
5. Memungkinkan secara serentak dan bersamaan beberapa pemakaian yang berarti juga meningkatkan kebebasan data sehingga berguna untuk beberapa program.
6. Meminimalisasi duplikasi dan redundansi dalam penyimpanan data.
7. Melindungi data dari gangguan kerusakan atau pemakaian oleh orang yang tidak terotorisasi.

### 2.6.3 Metode Perancangan Sistem Database

Tahapan perancangan (*design*) memiliki tujuan untuk membuat sistem baru yang dapat menyelesaikan masalah-masalah yang dihadapi perusahaan yang diperoleh

dengan pemilihan sistem yang terbaik. Adapun tahap-tahap perancangan *database* adalah sebagai berikut:

1. *Conceptual design*

*Conceptual design* adalah model lengkap yang merangkap keseluruhan struktur organisasi data, yang tidak mungkin terlepas dari sistem manajemen *database* atau pertimbangan implementasi lainnya. Konsep data yang digunakan adalah model data relasional. Konsep data yang termasuk di dalamnya adalah:

- a. *Entity identification*

Pada tahap ini dilakukan identifikasi *entity* yang terlibat. Setiap *entity* tersebut merupakan calon dari tabel yang akan dibuat.

- b. *Functional decomposition*

Suatu proses untuk menguraikan *entity* yang terlibat tersebut menjadi lebih *detail*. Selanjutnya hasil dari *functional decomposition* disebut sebagai atribut.

- c. *Entity relationship diagram*

Pada tahap ini dilakukan penentuan hubungan atau relasi yang terjadi antar *entity*.

2. Perancangan *Database Logis (Logical Database Design)*

Konsep model data yang telah terbentuk pada tahap *conceptual database design* dibawa ke bentuk yang logis, yaitu:

- a. Membuat model relasional

*Entity-entity* yang telah teridentifikasi menjadi sebuah tabel dengan *field* berasal dari atribut pada setiap *entity*.

- b. Normalisasi

Semua struktur tabel yang telah terbentuk dinormalisasikan dengan menggunakan tahapan dan aturan yang telah dijelaskan pada sub bab sebelumnya.

3. Perancangan *Database fisik (Physical Database Design)*

Model data yang telah terbentuk pada tahap *logical design* dibawa ke bentuk suatu penyimpanan data yang nyata dengan menggunakan sebuah *software*. Oleh karena itu pada tahap ini diawali dengan pemilihan *software* yang akan dipakai. Setelah itu baru dirancanglah struktur tabel sesuai dengan kebutuhan kita untuk melakukan penyimpanan data.

#### 4. Penerapan (*Implementations*)

Pada tahap ini dilakukan pengembangan *database* yaitu merancang hirarki *menu*, *form* dan *report* beserta program (*procedure* atau *function*) yang menghubungkan ketiganya. Tahap ini diakhiri dengan memasukkan data ke dalam sistem *database*.

#### 5. Pengujian Program

Pada langkah ini dilakukan pengujian terhadap program aplikasi yang telah dibuat. Dalam pengujian program ini dapat kita tinjau dari dua segi, yaitu verifikasi dan validasi.

- a. Verifikasi : menguji apakah program berjalan sesuai yang telah direncanakan.
- b. Validasi : menguji apakah fungsi program telah memenuhi sebagai suatu sistem informasi *inventory raw material*

### 2.6.4 Entity Relational Diagram (E-R Diagram)

*Entity Relational Diagram* merupakan salah satu pemodelan data konseptual yang paling sering digunakan dalam proses pengembangan *basis data* bertipe relasional. Model E-R adalah rincian yang merupakan representasi logika dari data pada suatu organisasi atau area bisnis tertentu.

Model E-R terdiri dari beberapa komponen dasar yaitu sebagai berikut:

#### 1. Entitas

Entitas adalah sesuatu atau objek di dunia nyata yang dapat dibedakan dari sesuatu atau objek yang lainnya. Sebagai contoh, setiap mahasiswa dalam suatu universitas adalah suatu entitas. Setiap fakultas dalam suatu universitas adalah juga suatu entitas. Selain menggambarkan objek nyata, entitas juga dapat menggambarkan suatu konsep kegiatan yang terjadi. Sebagai contoh, transaksi konsumen

#### 2. Hubungan antar relasi (*Relationship*)

Hubungan antar relasi adalah hubungan antara suatu himpunan entitas dengan himpunan entitas yang lainnya. Misalnya, entitas mahasiswa memiliki hubungan tertentu dengan entitas matakuliah (mahasiswa mengambil matakuliah). Pada penggambaran model E-R, relasi adalah perekat yang menghubungkan suatu entitas dengan entitas yang lainnya.

### 3. Kardinalitas/Derajat Relasi

Kardinalitas relasi menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain. Sebagai contoh: entitas-entitas pada himpunan entitas mahasiswa dapat berelasi dengan satu entitas, banyak entitas atau tidak satupun entitas dari himpunan entitas kuliah. Kardinalitas relasi yang terjadi di antara dua himpunan entitas dapat berupa:

- Satu ke Satu (*One to One*)

Setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas B, begitupun sebaliknya.

- Satu ke Banyak (*One to Many*)

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, sebaliknya, dimana setiap entitas pada himpunan entitas B berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas A

- 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 dengan entitas B.

- Banyak ke Banyak (*Many to Many*)

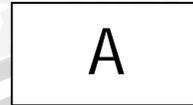
Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, dan demikian sebaliknya, dimana setiap entitas pada himpunan entitas B dapat berhubungan dengan banyak entitas pada himpunan entitas A.

### 4. Notasi-Notasi Simbolik

Berikut ini adalah notasi-notasi simbolik yang digunakan dalam pembuatan *Entity Relationship Diagram* :

a. Persegi Panjang

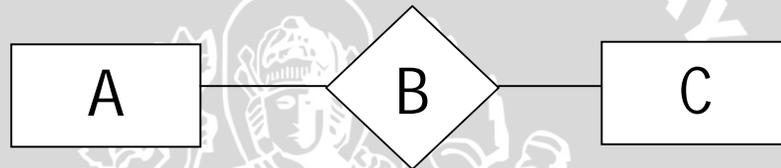
Dalam E-R Diagram, persegi panjang digunakan untuk menyatakan himpunan entitas. Contoh penggambaran himpunan entitas dapat dilihat pada gambar 2.5 berikut



Gambar 2.5 Simbol Himpunan Entitas A

b. Belah Ketupat

Dalam E-R Diagram, belah ketupat digunakan untuk menyatakan himpunan relasi. Contoh penggambaran himpunan relasi dapat dilihat pada gambar 2.6 berikut



Gambar 2.6 Simbol Himpunan Relasi B

c. Garis

Dalam E-R Diagram, garis berfungsi sebagai penghubung antara Himpunan Relasi dengan Himpunan

## 2.7 Software Pendukung

Pembuatan *software* ini menggunakan *Microsoft Access 2010* dan *Visual Basic for Application*. Kedua alat tersebut saling terintegrasi satu sama lain, sehingga dapat memudahkan membuat program aplikasi yang kompleks dan handal.

### 2.7.1 Microsoft Access 2010

*Microsoft Access* merupakan salah satu contoh produk RDBMS yang sangat populer di lingkungan *Windows*. *Microsoft Access* adalah program aplikasi pengolah *database* yang merupakan bagian dari *Microsoft Office*. Sama seperti versi-versi sebelumnya, *Microsoft Access 2010* merupakan bagian dari *Microsoft Office 2010*, dimana di dalamnya terdapat beberapa *software* lain seperti *Microsoft Word*, *Microsoft Excel*, *Microsoft PowerPoint*, dan lain sebagainya.

Program-program pengolah *database* telah tersedia bagi pengguna PC sejak lama. Akan tetapi sebagian besar program-program tersebut hanya mendukung penyimpanan data yang sederhana, dan sangat sulit dipelajari dan digunakan apabila ingin membuat suatu aplikasi yang rumit dan kompleks. *Microsoft Access* merupakan *software* yang berfungsi untuk pengembangan aplikasi *database*, khususnya aplikasi *database* berskala kecil sampai menengah

### 2.7.1.1 Objek *Microsoft Access*

Ada tujuh objek yang dimiliki *Microsoft Access*, antara lain:

1. *Tabel*, merupakan struktur dasar tempat data disimpan di dalam *database* sekaligus obyek pertama yang harus dibuat. *Tabel* terdiri dari baris (*record*) dan kolom (*field*). *Tabel* yang dibuat dapat berjumlah satu atau lebih disesuaikan dengan kebutuhan.
2. *Form*, merupakan suatu piranti yang memungkinkan untuk melihat dan mengedit data di dalam *tabel*. *Form* dapat dibuat lebih menarik dan mudah digunakan daripada *tabel*.
3. *Query*, bagian ini digunakan untuk mengatur data mana saja dari suatu *tabel* yang perlu ditampilkan. Dalam *query* ini juga dapat diatur kriteria atau syarat penampilan suatu data serta bagaimana data tersebut diurutkan.
4. *Report*, bagian ini digunakan untuk membuat laporan dari suatu data yang telah diolah menjadi informasi. Dari *report* dapat langsung dibuat *print out* informasi yang telah diolah.
5. *Pages*, bagian ini berfungsi untuk menciptakan halaman Web berupa data *Access Page*.
6. *Macros*, bagian ini berfungsi untuk mengotomasi perintah-perintah yang dikehendaki dalam mengolah data.
7. *Modules*, bagian ini digunakan untuk pembuatan dan pengolahan modul di dalam *Visual Basic for Application*.

*Microsoft Access* memungkinkan pengguna untuk membuat aplikasi *database* sederhana hingga kompleks dengan menawarkan *tools* yang sangat mudah dipelajari dan digunakan. *Microsoft Access 2010* adalah pengembangan dari versi *Microsoft Access* sebelumnya yang dirancang untuk *Windows* versi 32-bit, sehingga menghasilkan aplikasi yang sangat handal. *Microsoft Access* sebelumnya telah memiliki banyak versi,

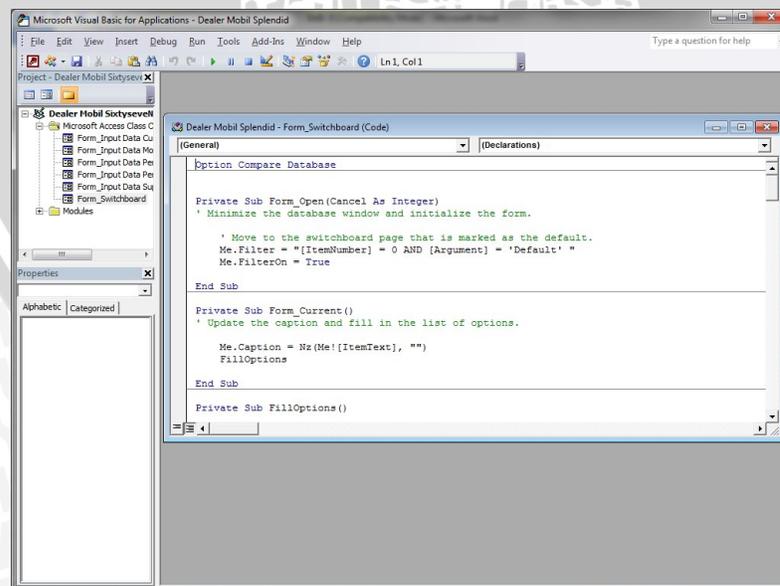
misalnya *Microsoft Access 97*, *Microsoft Access 2000*, *Microsoft Access XP* (*Microsoft Access 2002*), *Microsoft Access 2003*, hingga *Microsoft Access 2007*

*Microsoft Access 2010* dapat menggunakan data yang disimpan di dalam format *Microsoft Access*, *Microsoft Jet Database Engine*, *Microsoft SQL Server*, *Oracle Database*, atau semua kontainer basis data yang mendukung standar ODBC. Para pengguna/programmer yang mahir dapat menggunakannya untuk mengembangkan perangkat lunak aplikasi yang kompleks, sementara para programmer yang kurang mahir dapat menggunakannya untuk mengembangkan perangkat lunak aplikasi yang sederhana. *Access* juga mendukung teknik-teknik pemrograman berorientasi objek, tetapi tidak dapat digolongkan ke dalam perangkat bantu pemrograman berorientasi objek.

### 2.7.1.2 Visual Basic for Application (VBA)

VBA merupakan bahasa pemrograman yang berada di belakang aplikasi *Microsoft Access 2010*. VBA mirip dengan bahasa pemrograman *Microsoft Visual Basic*. Dengan kata lain, VBA adalah *Visual Basic* untuk *Microsoft Office* yang salah satunya adalah *Microsoft Access*. Dengan menggunakan VBA, aplikasi yang dibuat dapat menjadi lebih handal.

*Editor* VBA hanya terdapat pada objek *form*, *report*, dan *module*. Untuk membuka editor VBA, dapat melalui menu *database tools* lalu memilih VBA. Berikut ini adalah contoh tampilan dari editor VBA.



Gambar 2.7 Editor VBA