

BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Pengertian Komputer

Siagian (2011:92), “Komputer adalah alat mesin elektronika yang menerima dan mengolah data sedemikian rupa sehingga menghasilkan informasi”.

Sujatmiko (2012:156), “Komputer adalah mesin yang dapat mengolah data digital dengan mengikuti serangkaian perintah atau program”.

Dari kedua pengertian tersebut, dapat disimpulkan bahwa pengertian komputer adalah alat mesin elektronika yang dapat mengelola data mengikuti perintah atau program sehingga menghasilkan suatu informasi.

2.1.2. Pengertian Perangkat Keras (*Hardware*)

Kurniawan (2014:2) menyatakan, “*Hardware* adalah perangkat komputer yang terdiri atas susunan komponen-komponen elektronik berbentuk fisik (berupa benda). Jenis-jenis *hardware* pada *Personal Computer* (PC), antara lain *Motherboard*, *Hardisk*, Memori, dan lain-lain.”

Susilowati dan Purnama (2011:11) menjelaskan, “Perangkat keras (*hardware*) adalah instalasi perangkat atau sistem komputer secara umum yang merupakan rangkaian elektronik yang disusun sedemikian rupa sehingga dapat mengolah data dan menghasilkan informasi”.

Dari kedua pengertian tersebut, dapat disimpulkan bahwa pengertian perangkat keras adalah perangkat komputer yang terdiri atas susunan komponen-komponen elektronik berbentuk fisik yang disusun sedemikian rupa sehingga dapat mengolah data dan menghasilkan informasi.

2.1.3. Pengertian Perangkat Lunak (*Software*)

Susilowati dan Purnama (2011:11) menjelaskan, “Perangkat lunak (*software*) adalah program yang berisi perintah-perintah atau instruksi-instruksi untuk melaksanakan pengolahan data”.



Sujatmiko (2012:256) menyatakan, “*Software* adalah kumpulan beberapa perintah yang dieksekusi oleh mesin komputer dalam menjalankan pekerjaannya”.

Dari kedua pengertian tersebut, dapat disimpulkan bahwa pengertian perangkat lunak adalah program yang berisi perintah-perintah yang dieksekusi mesin komputer untuk melaksanakan pengolahan data.

2.1.4. Pengertian Data

Muslih dan Purnama (2013:50) menjelaskan, Data merupakan deskripsi tentang benda, kejadian, aktivitas, dan transaksi yang tidak mempunyai makna atau tidak berpengaruh secara langsung kepada pemakai. Data juga dapat diartikan suatu bahan mentah yang kelak dapat diolah lebih lanjut untuk menjadi sesuatu yang lebih bermakna.

Prihartanto (2011:55) menjelaskan, “Data adalah representasi dari suatu fakta, yang dimodifikasi dalam bentuk gambar, kata, dan angka. Manfaat data adalah sebagai satuan representasi yang dapat diingat, direkam, dan dapat diolah menjadi informasi”.

Dari kedua pengertian tersebut, dapat disimpulkan bahwa pengertian data adalah suatu bahan mentah yang dapat diolah lebih lanjut sehingga menjadi sebuah informasi.

2.1.5. Metode Pengembangan Sistem

Sukanto dan Shalahuddin (2013:28) menjelaskan tentang metode pengembangan sistem yaitu *waterfall*. Metode air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut mulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*).

a. Analisis kebutuhan perangkat lunak

Tahap analisis dilakukan secara intensif untuk menspesifikasikan kebutuhan sistem agar dapat dipahami sistem seperti apa yang dibutuhkan oleh *user*.



b. Desain

Tahap desain adalah proses multi langkah yang fokus pada desain pembuatan program sistem termasuk struktur data, arsitektur sistem, representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan sistem dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya.

c. Pembuatan kode program

Pada tahap pengkodean, desain harus ditranslasikan ke dalam program sistem. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain

d. Pengujian

Tahap pengujian fokus pada sistem dari segi logika dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

e. Pendukung (*support*) atau Pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah sistem mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau sistem harus beradaptasi dengan lingkungan baru. Tahap pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan sistem yang sudah ada, tapi tidak untuk sistem baru.

2.2. Teori Judul

2.2.1. Pengertian Sistem

Nasikin (2011:21) menjelaskan, "Sistem adalah sebuah tatanan (keterpaduan) yang terdiri atas sejumlah komponen fungsional (dengan satu fungsi atau tugas khusus) yang saling berhubungan dan secara bersama-sama bertujuan untuk memenuhi suatu proses pekerjaan tertentu."



2.2.2. Pengertian Sistem Otomasi Perpustakaan

Waluyo, dkk (2011:204) menjelaskan bahwa, Sistem otomasi perpustakaan merupakan suatu manajemen sistem yang dapat mempermudah akses baik pengelola maupun pengguna perpustakaan. Sistem otomasi perpustakaan yang baik adalah sistem yang terintegrasi, mulai dari sistem pengadaan bahan pustaka, pengolahan bahan pustaka, sistem pencarian kembali bahan pustaka, sistem sirkulasi (peminjaman, pengembalian dan perpanjangan peminjaman), keanggotaan (*membership*), pengaturan hak akses keanggotaan, pengaturan denda keterlambatan pengembalian, sistem booking dan sistem reporting aktifitas perpustakaan dengan berbagai parameter pilihan.

2.2.3. Pengertian Perpustakaan Digital

Waluyo, dkk (2011:205) menjelaskan bahwa, “Perpustakaan Digital (*digital library*) adalah perpustakaan yang harus memenuhi atau menyediakan semua jasa yang esensial dari jasa perpustakaan tradisional dan juga harus mengeksploitasi kelebihan dan manfaat penyimpanan penelusuran dan komunikasi digital”.

2.2.4. Pengertian Kartu

Kamus Besar Bahasa Indonesia menyatakan, Kartu adalah kertas tebal, berbentuk persegi panjang (untuk berbagai keperluan, hampir sama dengan karcis).

2.2.5. Pengertian Perpustakaan

Waluyo, dkk (2011:204) menjelaskan bahwa, Perpustakaan adalah institusi atau lembaga yang menyediakan koleksi bahan perpustakaan tertulis, tercetak dan terekam sebagai pusat sumber informasi yang diatur menurut sistem dan aturan yang baku dan didayagunakan untuk keperluan pendidikan, penelitian dan rekreasi intelektual bagi masyarakat.



2.2.6. Pengertian Manajemen Informatika

Manajemen Informatika adalah salah satu jurusan pada Politeknik Negeri Sriwijaya Palembang, yang pendiriannya ditetapkan melalui surat Direktur jenderal Pendidikan Tinggi (Dirjen Dikti) nomor 2800/D/T/2001.

2.2.7. Pengertian Algoritma

Lidya dan Munir (2016:5) menjelaskan bahwa, Algoritma adalah urutan langkah-langkah untuk menyelesaikan suatu persoalan. Terdapat beberapa definisi lain tentang algoritma, tetapi prinsipnya senada dengan definisi yang diungkapkan di atas, yang dikutip dari berbagai literatur, antara lain:

1. Algoritma adalah deretan langkah-langkah komputasi yang mentransformasikan data masukan menjadi luaran.
2. Algoritma adalah deretan instruksi yang jelas untuk memecahkan persoalan, yaitu untuk memperoleh luaran yang diinginkan dari suatu masukan dalam jumlah waktu yang terbatas.

2.2.8 Pengertian Algoritma *Scan Line*

Ardiantoro dan Murinto (2014:896) menjelaskan bahwa, Algoritma *Scan Line* adalah suatu dari algoritma *Visible Surface Determination* yang digunakan untuk memecahkan masalah penggunaan memori yang besar dengan satu baris *scan* untuk memproses semua permukaan objek, biasanya *Scan Line* akan mensweeping layar dari atas ke bawah.

2.2.9. Konsep Dasar *Visible-Surface Determination*

Ardiantoro dan Murinto (2016:896) menjelaskan bahwa, *Visible-Surface Determination* atau yang sering disebut dengan *Hidden Surface Removal* adalah suatu algoritma yang digunakan untuk menghilangkan penampilan bagian yang tertutup oleh objek yang didepannya. Apabila ada dua bidang yang berpotongan, apabila ditampilkan biasa tanpa menggunakan algoritma *Visible Surface Determination* maka bagian yang berpotongan itu akan tidak kelihatan, oleh karena



bidang yang satu ditutupi oleh bagian yang lain tanpa memotong. Oleh karena itu untuk menampilkan bidang perpotongan, diperlukan *Algoritma Hidden Surface Removal*.

2.2.10. Pengertian Judul Secara Keseluruhan

Sistem Otomasi Kartu Digital Pada Perpustakaan Jurusan Manajemen Informatika di Politeknik Negeri Sriwijaya Menggunakan Algoritma *Scan Line*, merupakan suatu aplikasi yang dibangun untuk proses pengolahan data perpustakaan secara otomatis dengan penerapan algoritma *scan line* pada alat *scanning barcode* untuk mengidentifikasi data yang telah terregistrasi pada *database* dan diaplikasikan pada *barcode* kartu digital.

2.3. Teori Khusus

2.3.1. Pengertian DFD (*Data Flow Diagram*)

Sukanto dan Shalahuddin (2014:70) menjelaskan, “*Data Flow Diagram (DFD)* atau dalam bahasa Indonesia menjadi Diagram Alir Data (DAD) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*).”

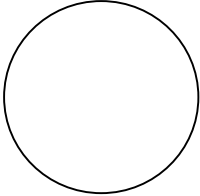
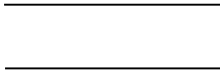
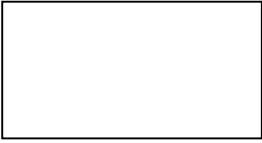
Ladjmudin (2013:64) mendefinisikan bahwa, “*DFD* adalah model dari sistem untuk menggambarkan pembagian sistem ke modul yang lebih kecil.”

Kristanto (2008:61) menjelaskan “*Data Flow Diagram* merupakan suatu model logika data atau proses yang dibuat untuk menggambarkan darimana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut.”




Sukamto dan Shalahuddin (2014:71) menjelaskan, “notasi-notasi pada DFD (Edward Yourdon dan Tom DeMarco) adalah sebagai berikut:

Tabel 2.1. Notasi-notasi pada DFD (Edward Yourdon dan Tom DeMarco)

Notasi	Keterangan
	<p>Proses atau fungsi atau prosedur; pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya menjadi fungsi atau prosedur di dalam kode program.</p> <p>Catatan: Nama yang diberikan pada sebuah proses biasanya berupa kata kerja.</p>
	<p><i>File</i> atau basisdata atau penyimpanan (<i>storage</i>); pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya dibuat menjadi tabel-tabel basis data yang dibutuhkan, tabel-tabel ini juga harus sesuai dengan perancangan tabel-tabel pada basis data (<i>Entity Relationship Diagram (ERD)</i>, <i>Conceptual Data Model (CDM)</i>, <i>Physical Data Model (PDM)</i>)`</p>
	<p>Entitas luar (<i>external entity</i>) atau masukan (<i>input</i>) atau keluaran (<i>output</i>) atau orang yang memakai/berinteraksi dengan perangkat lunak yang dimodelkan atau system lain yang terkait dengan aliran data dari system yang dimodelkan.</p> <p>Catatan: Nama yang digunakan pada masukan (<i>input</i>) atau keluaran (<i>output</i>) biasanya berupa kata benda</p>



Lanjutan Tabel 2.1. Notasi-notasi pada DFD (Edward Yourdon dan Tom DeMarco)

Notasi	Keterangan
	<p>Aliran data; merupakan data yang dikirim antar proses, dari penyimpanan ke proses, atau dari proses ke masukan (<i>input</i>) atau keluaran (<i>output</i>) Catatan: Nama yang digunakan pada aliran data biasanya berupa kata benda, dapat diawali dengan kata data misalnya “data siswa” atau tanpa kata data misalnya “siswa”</p>

Berikut ini adalah tahapan-tahapan perancangan dengan menggunakan DFD:

1. Membuat DFD Level 0 atau sering disebut juga *Context Diagram*

DFD Level 0 menggambarkan sistem yang akan dibuat sebagai suatu entitas tunggal yang berinteraksi dengan orang maupun sistem lain. DFD Level 0 digunakan untuk menggambarkan interaksi antara sistem yang akan dikembangkan dengan entitas luar.

2. Membuat DFD Level 1

DFD Level 1 digunakan untuk menggambarkan modul-modul yang ada dalam sistem yang akan dikembangkan. DFD Level 1 merupakan hasil *breakdown* DFD Level 0 yang sebelumnya sudah dibuat.

3. Membuat DFD Level 2

Modul-modul pada DFD Level 1 dapat di-*breakdown* menjadi DFD Level 2. Modul mana saja yang harus di-*breakdown* lebih detail tergantung pada tingkat kedetailan modul tersebut. Apabila modul tersebut sudah cukup detail dan rinci maka modul tersebut sudah tidak perlu untuk di-*breakdown* lagi. Untuk sebuah sistem, jumlah DFD Level 2 sama dengan jumlah modul pada DFD Level 1 yang di- *breakdown*.



4. Membuat DFD Level 3 dan seterusnya

DFD Level 3, 4, 5 dan seterusnya merupakan *breakdown* dari modul pada DFD Level di-atasnya. *Breakdown* pada level 3, 4, 5, dan seterusnya aturannya sama persis dengan DFD Level 1 atau Level 2.


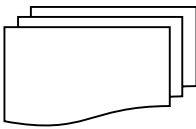
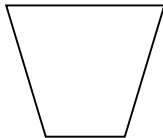
Pada satu diagram DFD sebaiknya jumlah modul tidak boleh lebih dari 20 buah. Jika lebih dari 20 buah modul, diagram akan terlihat rumit dan susah untuk dibaca sehingga menyebabkan system yang dikembangkan juga menjadi rumit.

2.3.2. Pengertian *Blockchart*


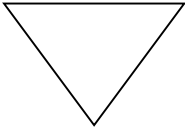
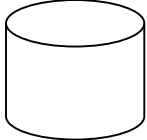

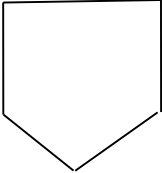
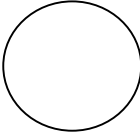
Kristanto (2008:68) menjelaskan, “*Blockchart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu. Pembuatan *Blockchart* harus memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi.”

Kristanto (2008:68) menjelaskan, “Simbol-simbol yang sering digunakan dalam *blockchart* dapat dilihat pada tabel berikut ini:

Tabel 2.2. Simbol-simbol *Block Chart*


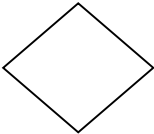


No	Simbol	Keterangan
1.		Menandakan dokumen, bisa dalam bentuk surat, formulir, buku/bendel/berkas atau cetakan
2.		Multi dokumen
3.		Proses Manual

Lanjutan Tabel 2.2. Simbol-simbol *Block Chart*

No	Simbol	Keterangan
4.		Proses dilakukan oleh komputer.
5.		Menandakan dokumen yang diarsipkan (arsip manual)
6.		Data penyimpanan (<i>Storage</i>)
7.		Proses apa saja yang tidak terdefinisi termasuk aktivitas fisik.
8.		Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang lain.
9.		Terminal yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang sama.



Lanjutan Tabel 2.2. Simbol-simbol dalam *Block Chart*

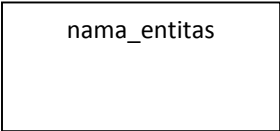
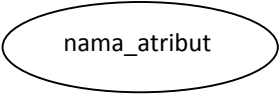
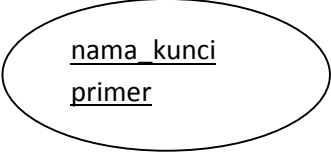
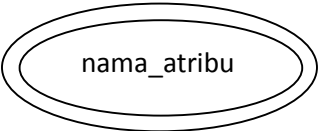
No	Simbol	Keterangan
10.		Terminal yang menandakan awal dan akhir dari suatu aliran.
11.		Pengambilan keputusan (<i>Decision</i>).
12.		Layar peraga (<i>monitor</i>).
13.		Pemasukkan data secara manual.

2.3.3. Pengertian ERD (*Entity Relationship Diagram*)

Sukanto dan Shalahuddin (2014:50) menjelaskan, tentang pemodelan awal basis data yang paling banyak digunakan adalah: menggunakan *Entity Relationship Diagram* (ERD). ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional. Sehingga jika penyimpanan basis data menggunakan OODBMS maka perancangan basis data tidak perlu menggunakan ERD. ERD memiliki beberapa aliran notasi seperti notasi Chen (dikembangkan oleh Peter Chen), Barker (dikembangkan oleh Richard Barker, Ian Palmer, Harry Ellis), notasi Crow's Foot, dan beberapa notasi lain. Namun yang banyak digunakan adalah notasi dari Chen.

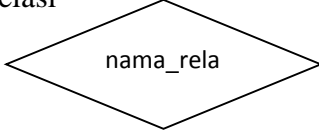
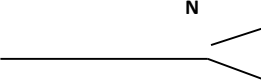
Shalahuddin (2014:50) menjelaskan, “Berikut adalah simbol-simbol yang digunakan pada ERD dengan notasi Chen:

Tabel 2.3. Simbol-simbol ERD dengan notasi Chen

Simbol	Deskripsi
Entitas / <i>entity</i> 	Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi computer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.
Atribut 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas
Atribut kunci primer 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan; biasanya berupa id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama)
Atribut multivalai / multivalue 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu



Lanjutan Tabel 2.3. Simbol-simbol ERD dengan notasi Chen

Simbol	Deskripsi
	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja
	<p>Penghubung antara relasi dan entitas di mana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian.</p> <p>Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas</p>



ERD biasanya memiliki hubungan *binary* (satu relasi menghubungkan dua buah entitas). Beberapa metode perancangan ERD menoleransi hubungan relasi *ternary* (satu relasi menghubungkan tiga buah relasi) atau *N-ary* (satu relasi menghubungkan banyak entitas), tapi banyak metode perancangan ERD yang tidak mengizinkan hubungan *ternary* atau *N-ary*.

2.3.4. Pengertian *Flowchart*



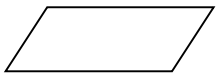

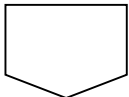

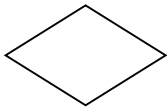

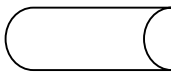
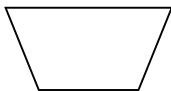

eWolf Community (2012:6), “*Flowchart* adalah simbol-simbol pekerjaan yang menunjukkan bagan alir proses yang saling terhubung. Jadi, setiap simbol *flowchart* melambangkan pekerjaan dan instruksinya.”

Siallagan (2009:6), “menjelaskan simbol-simbol dalam *Flowchart* adalah sebagai berikut:

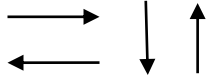

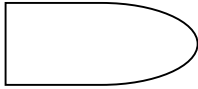
Tabel 2.4. Simbol-simbol dalam *Flow Chart*

No.	Simbol	Keterangan
1.		Simbol <i>Start</i> atau <i>End</i> yang mendefinisikan awal atau akhir dari sebuah <i>flowchart</i> .
2.		Simbol pemerosesan yang terjadi pada sebuah alur kerja.

Lanjutan **Tabel 2.4.** Simbol-simbol dalam *Flow Chart*

No.	Simbol	Keterangan
3.		Simbol yang menyatakan bagian dari program (sub program).
4.		Persiapan yang digunakan untuk memberi nilai awal suatu besaran.
5.		Simbol <i>Input/Output</i> yang mendefinisikan masukan dan keluaran proses.
6.		Menyatakan penyambung ke simbol lain dalam satu halaman.
7.		Menyatakan penyambung ke halaman lainnya.
8.		Menyatakan pencetakan (dokumen) pada kertas.
9.		Menyatakan <i>desicion</i> (keputusan) yang digunakan untuk penyeleksian kondisi di dalam program.
10.		Menyatakan media penyimpanan drum magnetik.
11.		Menyatakan <i>input/output</i> menggunakan disket.
12.		Menyatakan operasi yang dilakukan secara manual.
13.		Menyatakan <i>input/output</i> dari kartu plong.

Lanjutan **Tabel 2.4.** Simbol-simbol dalam *Flow Chart*

No.	Simbol	Keterangan
14.		Menyatakan arah aliran pekerjaan (proses).
15.		<i>Multidocument</i> (banyak dokumen).
16.		<i>Delay</i> (penundaan atau kelambatan).

2.3.5. Pengertian Kamus Data

Sukanto dan Shalahuddin (2014:73) menjelaskan, “Kamus data (*data dictionary*) dipergunakan untuk memperjelas aliran data yang digambarkan pada DFD. Kamus data adalah kumpulan daftar elemen data yang mengalir pada system perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan). Kamus data dalam implementasi program dapat menjadi parameter masukan atau keluaran dari sebuah fungsi atau prosedur.

Kamus data biasanya berisi:

- a. Nama – nama dari data
- b. Digunakan – pada proses-proses yang terkait data
- c. Deskripsi – merupakan deskripsi data
- d. Informasi tambahan – seperti tipe data, nilai data, batas nilai data, dan komponen yang membentuk data

Kamus data memiliki beberapa simbol untuk menjelaskan informasi tambahan sebagai berikut:

**Tabel 2.5. Simbol-simbol dalam Kamus Data**

Simbol	Keterangan
=	disusun atau terdiri dari
+	Dan
[]	baik ...atau...
{ } ⁿ	n kali diulang/ bernilai banyak
()	data opsional
...	batas komentar

Kamus data pada DFD nanti harus dapat dipetakan dengan hasil perancangan basis data yang dilakukan sebelumnya. Jika ada kamus data yang tidak dapat dipetakan pada tabel hasil perancangan basis data berarti hasil perancangan basis data dengan perancangan dengan DFD masih belum sesuai, sehingga harus ada yang diperbaiki baik perancangan basis datanya, perancangan DFD-nya, atau keduanya.

2.4. Teori Program

2.4.1. Basis Data (*Database*)

Sujatmiko (2012:76) mengatakan bahwa “Basis data adalah kumpulan fakta yang saling berhubungan disimpan secara bersama sedemikian rupa dan tanpa pengulangan (*redundansi*) yang tidak perlu, untuk memenuhi berbagai kebutuhan”.

Rosa dan Shalahuddin (2013:43), mengatakan “Basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat”.



2.4.2. Pengertian Pemrograman Visual Basic 6.0

Sujatmiko (2012:307) mengatakan bahwa “*Visual Basic* adalah bahasa pemrograman yang digunakan untuk membuat program interaktif di dalam sebuah *web page*, yang dikembangkan oleh perusahaan *Microsoft Corp*”.

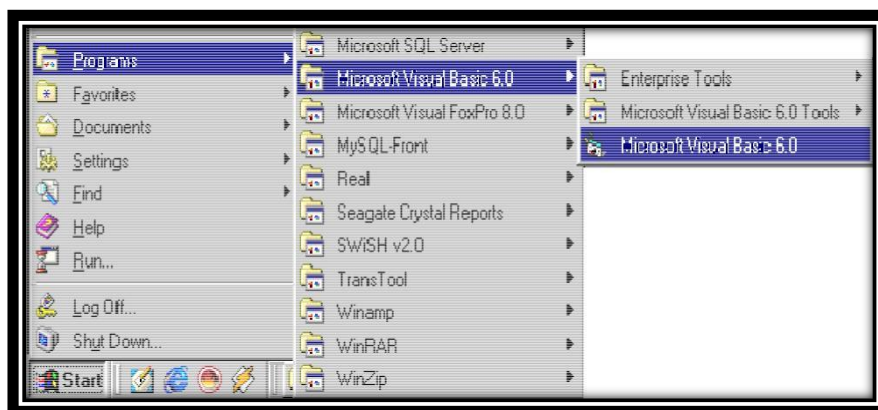
Fauzi (2012:1) mengatakan bahwa, *Visual Basic 6.0* adalah salah satu produk bahasa pemrograman yang dikeluarkan Microsoft, salah satu perusahaan *software* terkemuka di dunia. *Visual basic 6.0* merupakan bahasa pemrograman yang mudah digunakan untuk pengembangan aplikasi, baik itu aplikasi kecil maupun aplikasi besar.

2.4.3. Mengenal Elemen IDE (*Interactive Development Environment*)

Sujatmiko (2012:137) mengatakan bahwa “*Interactive Development Environment* adalah perangkat kontrol yang digunakan sebagai antar muka yang biasa lazim digunakan untuk *hard drive*, *CDROM drive*, dan lain-lain”.

Fauzi (2012:2) menguraikan tahapan untuk menjalankan atau mengaktifkan *Visual Basic 6.0*, ikuti langkah-langkah berikut ini:

1. Pada tampilan utama *Window*, klik **Start**.
2. Pilih **Programs > Microsoft Visual Basic 6.0 > Microsoft Visual Basic 6.0**.

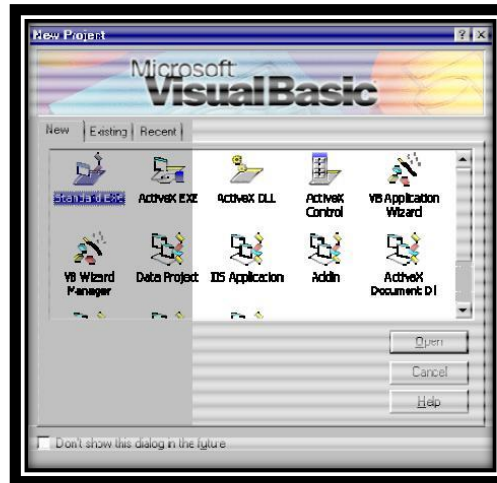


Gambar 2.1. Cara Membuka *Visual Basic*

3. Pada kotak dialog *New Project* pastikan *Standart.EXE* dalam keadaan terpilih, lalu klik tombol *open*, atau dapat juga dengan menekan **Existing**



untuk membuka *project* yang sudah ada dan **Recent** untuk membuka *project* yang sudah terbuka sebelumnya.



Gambar 2.2. Kotak Dialog New Project

2.4.4. Mengenal Toolbox Visual Basic 6.0

Fauzi (2012:5) menjelaskan bahwa, *Toolbox* adalah tempat di mana kontrol-kontrol diletakkan. Kontrol-kontrol yang terdapat pada *toolbox* dipakai dalam pembuatan program aplikasi. Saat pertama kali dijalankan program *Visual Basic 6.0* akan menempatkan *toolbox* di sebelah kiri *layer* dan berisi 21 kontrol standar.






















Gambar 2.3. Toolbox



Berikut ini penjelasan dan fungsi dari masing-masing kontrol yang ada dalam *Toolbox Visual Basic 6.0*, yaitu :



Table 2.6. Penjelasan dan Fungsi Kontrol

Bentuk	Nama Kontrol	Fungsi
	<i>Pointer</i>	Kontrol yang berfungsi untuk memindahkan atau mengubah ukuran <i>control</i> yang ada pada <i>form</i> .
	<i>PictureBox</i>	Untuk menampilkan file gambar (<i>Bitmaps, Icon, Gif, Jpeg, dsb</i>).
	<i>Label</i>	Untuk menampilkan teks, tetapi pemakai tidak bisa berinteraksi dengannya.
	<i>TextBox</i>	Untuk menempatkan teks pada <i>form</i> dan pemakai dapat mengedit teks tersebut.
	<i>Frame</i>	Untuk mengelompokkan beberapa <i>control (Group)</i> pada suatu <i>form</i> .
	<i>Command Button</i>	Untuk membuat tombol pelaksana suatu perintah atau tindakan ketika digunakan.
	<i>CheckBox</i>	Untuk membuat kotak <i>check</i> yang dapat memilih satu atau banyak pilihan.
	<i>Option Button</i>	Untuk memilih dan mengaktifkan satu pilihan dari banyak pilihan yang ada.
	<i>ComboBox</i>	Sebagai tempat mengetikkan pilihan atau memilih suatu pilihan lewat <i>DropDown – List</i> .
	<i>ListBox</i>	Untuk menampilkan daftar pilihan yang dapat digulung secara <i>horizontal</i> maupun <i>vertical</i> .
	<i>HScrollBar</i>	Untuk menggulung suatu area kerja dengan jangka lebar pada posisi <i>horizontal</i> .
	<i>VScrollBar</i>	Untuk menggulung suatu area kerja dengan jangka lebar pada posisi <i>vertical</i> .
	<i>Timer</i>	Untuk mengoperasikan waktu kejadian pada rutin program dalam <i>interval</i> yang ditentukan.
	<i>DriveListBox</i>	Untuk menampilkan daftar <i>drive computer</i> yang aktif dan dapat dipilih sebuah <i>drive</i> .
	<i>DirListBox</i>	Untuk menampilkan daftar <i>direktori</i> dan <i>path</i> pada <i>drive</i> kerja terpilih.
	<i>FileListBox</i>	Untuk menampilkan daftar <i>file</i> pada <i>direktori</i> dan <i>drive</i> yang aktif.
	<i>Shape</i>	Untuk membentuk obyek dua dimensi seperti <i>square, oval, ellips</i> dan lain-lain.
	<i>Line</i>	Untuk menggambar garis lurus dengan banyak variasi dengan ketebalan yang bisa diatur.
	<i>Image</i>	Untuk menampilkan gambar <i>icon bitmap</i> , atau <i>metafile</i> pada <i>form</i> .

Lanjutan **Table 2.6.** Penjelasan dan Fungsi Kontrol

	<i>Data Control</i>	Sebagai sarana akses data dalam suatu <i>database</i> .
	<i>OLE</i>	Untuk menghasilkan proses <i>link</i> dan <i>Embed</i> obyek antar aplikasi.

2.4.5. Pengertian Microsoft Access 2013

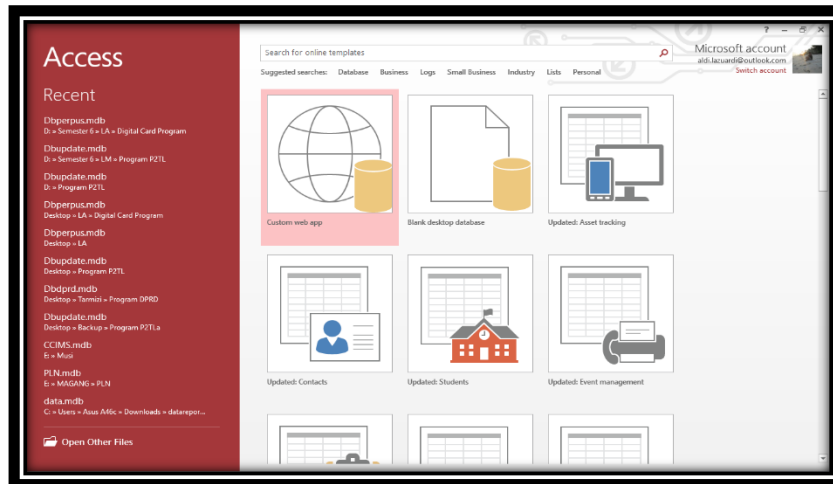
MADCOMS (2015:1) mengatakan, “*MS-Access 2013* merupakan program pengolahan *database* yang dapat mengolah berbagai jenis data serta membuat hasil akhir berupa laporan dengan tampilan yang lebih menarik”.

Terdapat beberapa cara untuk menjalankan program *MS-Access 2013* dimana hal ini dipengaruhi oleh sistem operasi yang digunakan. Berikut ini adalah langkah-langkah untuk menjalankan program *MS-Access 2013*:

1. Gunakan salah satu cara berikut berdasarkan sistem operasi yang digunakan komputer anda.
 - a. Sistem operasi *windows 7* dijalankan dengan cara: klik tombol ***Start*** di sudut kiri area ***Desktop***, kemudian pilih ***All Programs*** dan pilih folder ***Miscrosoft Office***.
 - b. Sistem operasi *windows 8* dijalankan dengan cara: arahkan *pointer* ke ***Start Screen*** di sudut kiri bawah area ***Desktop Windows 8***. Geser *scroll* ke kanan hingga tampil program yang telah *diinstall* pada komputer.
2. Pilih ***Access 2013*** sehingga akan ditampilkan animasi awal program *MS-Access 2013*.



3. Setelah animasi selesai maka akan tampil jendela program *MS-Access 2013*, di mana terdapat pilihan *template* yang telah disediakan.




Gambar 2.4. Jendela utama *Microsoft Access 2013*

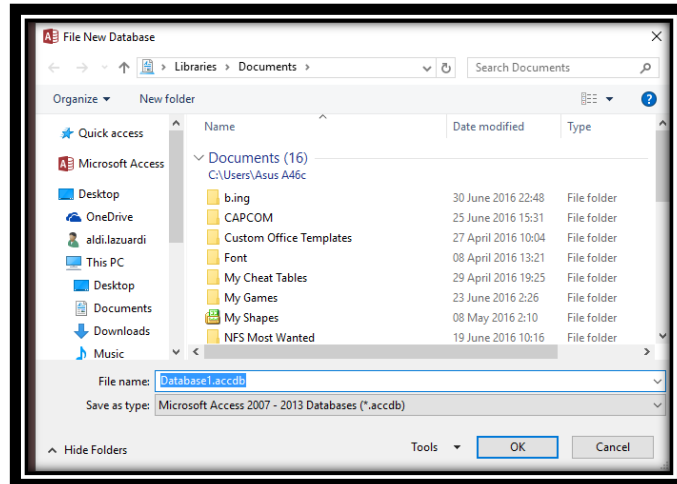
4. Pilih *Blank desktop database* untuk membuka *database baru MS-Access 2013* yang masih kosong.
5. Ketik nama *file database* yang akan dibuat pada bagian *File Name*, secara default *MS-Access 2013* akan memberi nama *file database* dengan *Database1*.



Gambar 2.5. Kotak Dialog *Blank Desktop Database*



- Selanjutnya klik tombol **Browse**  yang ada di sebelah kanan **File Name** untuk menentukan lokasi penyimpanan *file* sehingga akan tampil dialog **File New Database**.



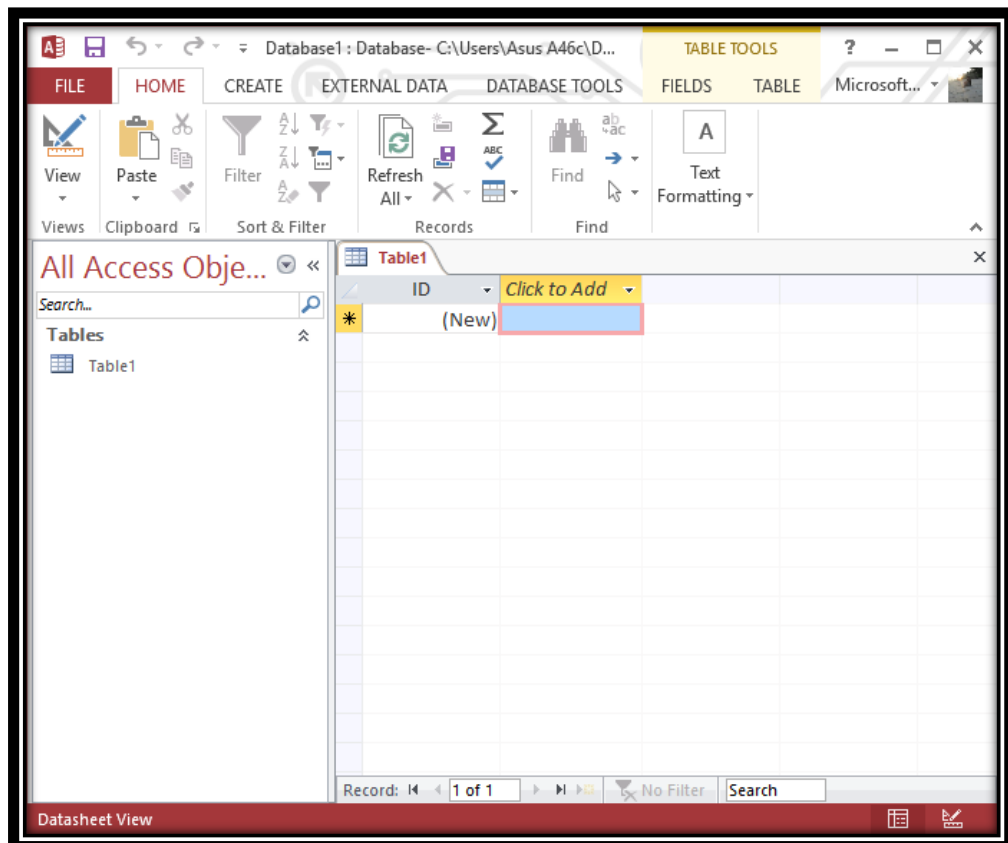
Gambar 2.6. Tampilan Kotak Dialog *File New Database*

- Tentukan lokasi *drive* dan *folder* tempat penyimpanan *file*. Kemudian klik tombol **OK**.
- Klik tombol **Create** sehingga tampil jendela **Database (MS-Access 2013)**. Dalam jendela *database* yang baru dibuat, tampil objek tabel yang masih kosong.

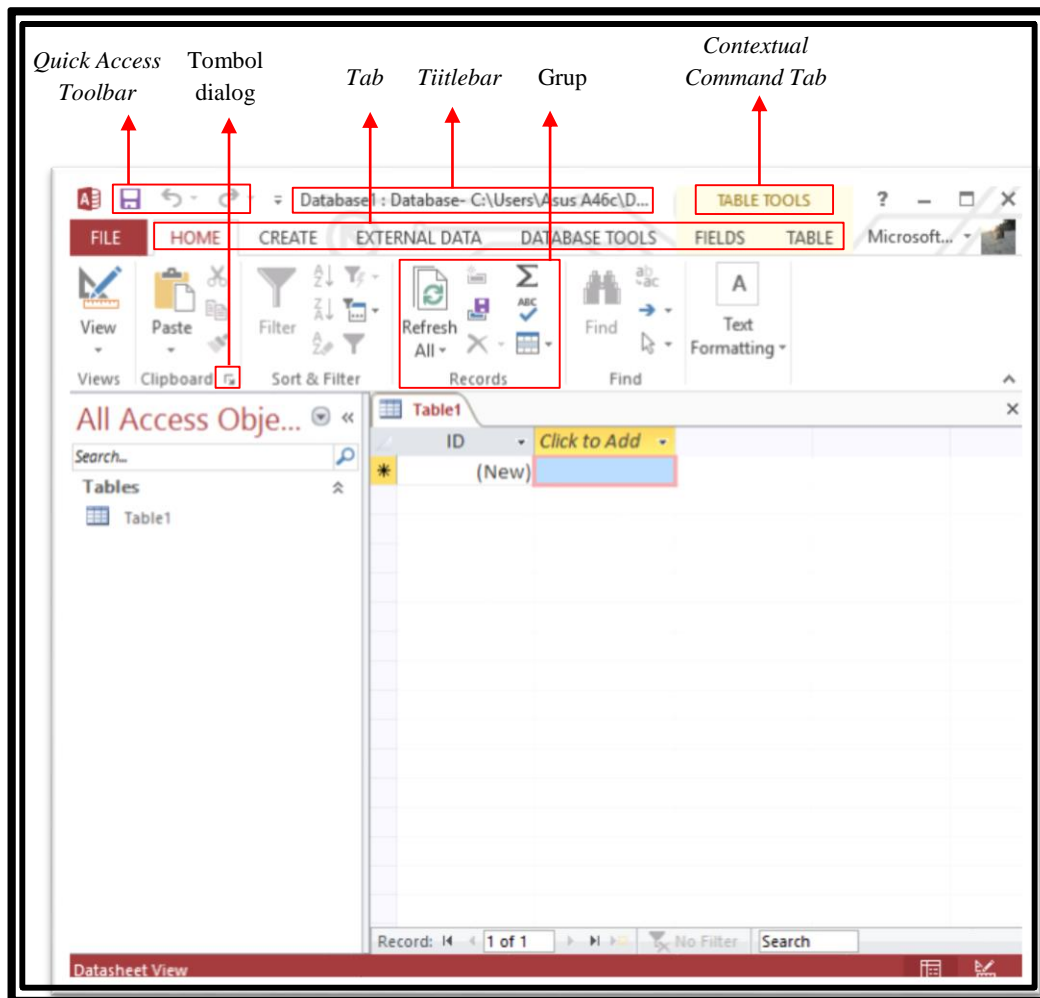


2.4.6. Komponen Microsoft Access 2013

Di dalam *Database MS-Access 2013* (atau dalam istilah lain adalah jendela kerja), masih terdapat *tab-tab* yang berisi tombol-tombol perintah di dalamnya yang terangkai pada *Ribbon*.



Gambar 2.7. Tampilan Jendela Kerja Database1







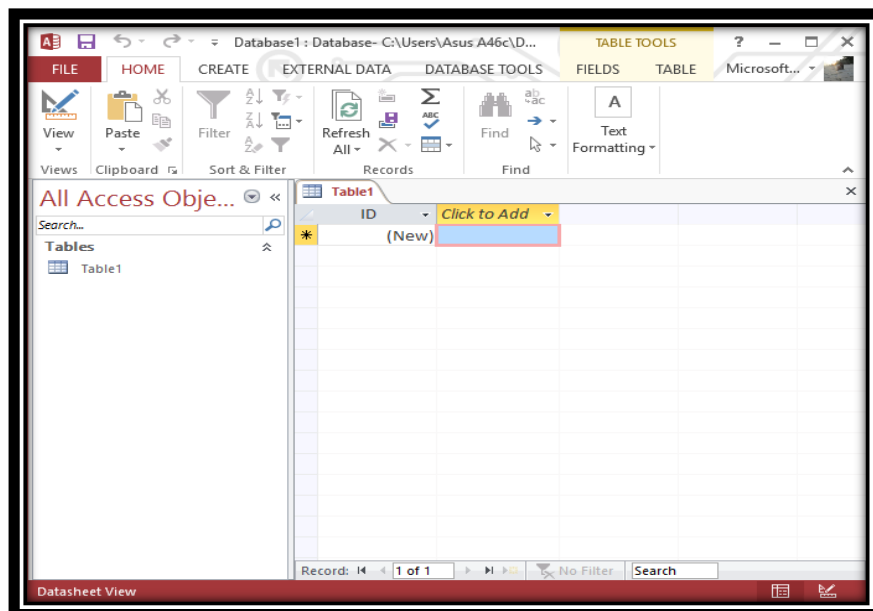
Gambar 2.8. Komponen-komponen dalam *MS-Access 2013*

Beirkut ini beberapa komponen-komponen dalam *MS-Access 2013*:

1. **Quick Access Toolbar**, merupakan *toolbar* standar yang berisi tombol-tombol perintah yang sering digunakan, seperti: tombol *Save*, tombol *Undo*, tombol *Redo*.
2. **Grup**, merupakan isi dari *tab* yang berupa kumpulan tombol perintah yang terletak di bawah *tab*.
3. **Tittlebar**, merupakan judul dari jendela program atau nama *file* yang sedang terbuka atau aktif.



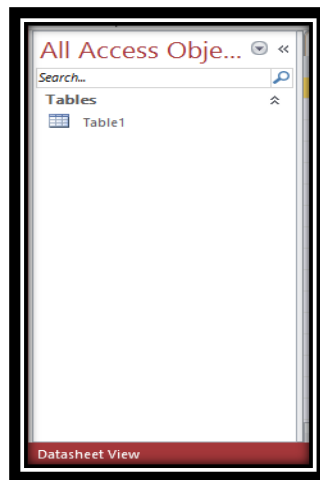
4. **Minimize** , merupakan tombol yang berfungsi untuk meminimalkan ukuran jendela program *MS-Access 2013* sehingga tampilannya berubah menjadi sebuah ikon yang diletakkan pada bagian *taskbar*.
5. **Maximize** , tombol yang berfungsi untuk memaksimalkan ukuran jendela program *MS-Access 2013* sehingga tampilannya menjadi satu layar monitor penuh.
6. **Restore Down** , tombol yang berfungsi untuk mengembalikan ukuran jendela program *MS-Access 2013* menjadi ukuran standar.
7. **Close** , tombol yang berfungsi untuk menutup jendela program atau keluar dari program *MS-Access 2013*.
8. **Status Bar**, merupakan batang yang berada di bawah jendela program *MS-Access 2013*, yang digunakan untuk menampilkan informasi *status* termasuk tombol-tombol untuk mengubah tampilan. Misalnya: *Status* aktif *Caps Lock*, *Num Lock*, *Scroll Lock* dan sebagainya.
9. **Ribbon**, area yang berada di atas jendela program, dimana anda dapat memilih tombol perintah.



Gambar 2.9. Tampilan *Ribbon MS-Access 2013*

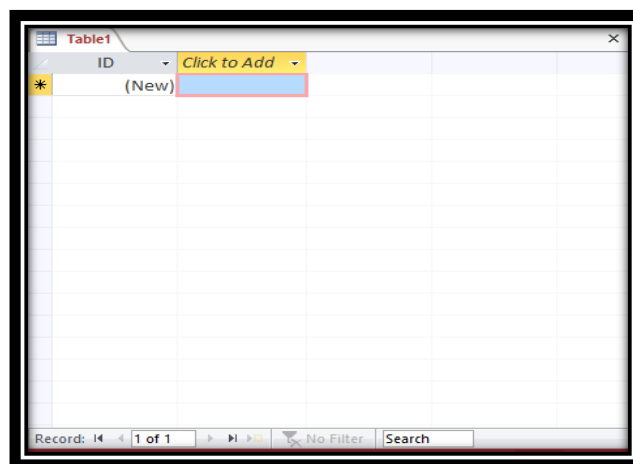


10. **Tabs**, berisi kumpulan dari grup yang berbentuk tabulasi secara otomatis menyesuaikan dokumen aktif.
11. **Contextual Command Tab**, terletak di sebelah kanan grup dan berfungsi untuk membuka kotak dialog.
12. **Navigation Pane**, merupakan bagian pada sisi kiri sebelah jendela *database* yang menampilkan objek-objek *database*.



Gambar 2.10. Tampilan *Navigation Pane*

13. **Tabbed Documents**, tampilan objek-objek dalam *MS-Access 2013* seperti: Tabel, *Query*, *Form Report* dan lainnya dalam bentuk tabulasi atau kartu jika kondisinya terbuka.



Gambar 2.11. Tampilan *Tabbed Documents*