



## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Teori Umum

##### 2.1.1. Pengertian Komputer

Sutanta (2011:1), “Komputer adalah alat elektronik yang mampu melaksanakan beberapa tugas, yaitu menerima *input*, memproses *input* sesuai dengan programnya, menyimpan perintah dan hasil pengolahan, serta menyediakan *output* dalam bentuk informasi.”

Wahyudi (2012:3) menjelaskan bahwa, “Komputer adalah Peralatan (*device*) yang menerima data (*input*) dan menyimpan (*storage*) kemudian di proses (*process*) untuk menghasilkan data dalam bentuk lain (*output*).”

Asropudin (2013:19), “Komputer adalah alat bantu pemrosesan data secara elektronik dan cara pemrosesan datanya berdasarkan urutan instruksi atau program yang tersimpan dalam memori masing-masing komputer.”

Kesimpulannya, komputer adalah alat elektronik yang digunakan untuk memproses data dan menghasilkan informasi yang berguna.

##### 2.1.2. Pengertian Aplikasi

Sugiar (2014:83) menjelaskan bahwa, “Aplikasi adalah program yang dibuat untuk melaksanakan tugas tertentu yang dibutuhkan oleh pengguna komputer (*user*).”

Asropuddin (2013:6) menjelaskan bahwa, “Aplikasi adalah *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya *Microsoft Word* dan *Microsoft Excel*.”

Kesimpulannya, aplikasi merupakan bagian dari *software* yang dibuat untuk melakukan tugas tertentu.



### **2.1.3. Pengertian Pengolahan Data**

Kristanto (2011:8) menjelaskan bahwa, “Pengolahan data adalah waktu yang digunakan untuk menggambarkan perubahan bentuk data menjadi informasi yang memiliki kegunaan.”

### **2.1.4. Pengertian Koperasi**

Subagyo (2014:189) menjelaskan bahwa, “Koperasi adalah badan usaha yang beranggotakan orang seorang atau badan hukum koperasi dengan melandaskan kegiatannya berdasarkan prinsip kopersai sekaligus sebagai gerakan ekonomi rakyat yang berdasarkan atas asas kekeluargaan.”

### **2.1.5. Pengertian Simpanan**

Subagyo (2014:189) menjelaskan bahwa, “Simpanan adalah dana yang dipercayakan oleh anggota, calon anggota, koperasi lain, dan atau anggotanya kepada KSP (Koperasi Simpan Pinjam) dalam bentuk tabungan.”

### **2.1.6. Pengertian Pinjaman**

Subagyo (2014:34) menjelaskan bahwa, “Pinjaman adalah penyediaan uang atau tagihan yang dapat dipersamakan dengan itu, berdasarkan persetujuan atau kesepakatan pinjam meminjam antara koperasi dengan pihak peminjam untuk melunasi utangnya setelah jangka waktu tertentu disertai dengan pembayaran sejumlah imbalan.”

### **2.1.7. Pengertian Kopersi Simpan Pinjam**

Subagyo (2014:189) menjelaskan bahwa, “Koperasi Simpan Pinjam adalah koperasi yang kegiatannya hanya usaha simpan pinjam.”

### **2.1.8. Pengertian Aplikasi Pengolahan Data Simpan Pinjam pada Koperasi**

#### **Sepakat Jaya Banyuasin**

Aplikasi Pengolahan Data Simpan Pinjam pada Koperasi Sepakat Jaya Banyuasin adalah aplikasi yang berguna untuk mengolah data-data penyimpanan

---



dan peminjaman, menginput dan membuat laporan data-data penyimpanan dan peminjaman yang dapat memudahkan dan membantu Koperasi Sepakat Jaya Banyuasin untuk menjadi lebih terkomputerisasi dan tidak terjadi kesalahan data, kehilangan data, dan ketidak akuratan data.

## 2.2. Teori Khusus

### 2.2.1. Tahap Pengembangan Perangkat Lunak

Sukamto dan Shalahuddin (2014:25) menjelaskan bahwa dalam pengembangan perangkat lunak atau sering disebut *Software Development Life Cycle* (SDLC). SDLC memiliki beberapa model dalam penerapan dan tahapan prosesnya, salah satunya adalah model *Waterfall*.

Model *Waterfall* menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung. Berikut penjelasan dari model *Waterfall*:

#### 1. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan *user*.

#### 2. Desain

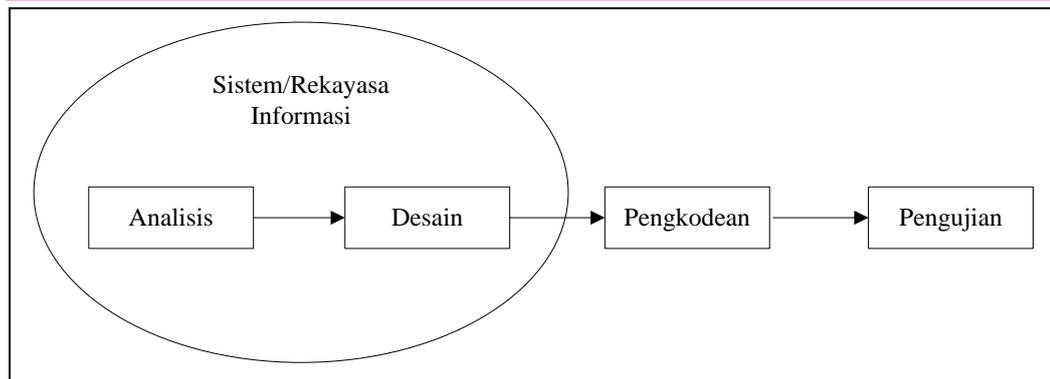
Desain perangkat lunak adalah proses multi langkah yang focus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan proses pengodean.

#### 3. Pembuatan Kode Program

Desain harus ditranslasikan ke dalam program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

#### 4. Pengujian

Pengujian berfokus kepada perangkat lunak secara dari segi logic dan memastikan bahwa semua bagian sudah diuji.



(Sumber: Sukamto dan Salahuddin, 2013:29)

**Gambar 2.1.** Gambar Ilustrasi Model Waterfall

### 2.2.2. Pengertian *Data Flow Diagram* (DFD)

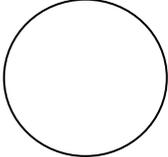
Indrajani (2015:27) menjelaskan bahwa, “*Data Flow Diagram* adalah sebuah alat yang menggambarkan aliran data sampai sebuah system selesai, dan kerja atau proses dilakukan dalam system tersebut.”

Saputra (2014:8) menjelaskan bahwa, “*Data Flow Diagram* (DFD) atau yang disingkat DFD merupakan suatu diagram yang menggambarkan aliran data dalam suatu entitas ke sistem atau sistem ke entitas. DFD juga diartikan sebagai teknik grafis yang menggambarkan alir data dan transformasi yang digunakan sebagai perjalanan data dari input atau masukan menuju keluaran atau output”.

Sutabri (2012:116) menjelaskan bahwa, “*Data Flow Diagram* adalah suatu network yang menggambarkan suatu sistem automat/komputerisasi, manualisasi, atau gabungan dari keduanya, yang penggambarannya disusun dalam bentuk kumpulan komponen sistem yang saling berhubungan sesuai dengan aturan mainnya.”

Sukamto dan Salahuddin (2014:71), menjelaskan notasi pada DFD adalah sebagai berikut:

Tabel 2.1. Simbol-simbol *Data Flow Diagram*

No	Simbol	Keterangan
1.		<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:</p> <p>nama yang diberikan pada sebuah proses biasanya berupa kata kerja.</p>
2.		<p><i>File</i> atau basis data 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>). catatan :</p> <p>nama yang diberikan pada sebuah penyimpanan biasanya kata benda.</p>
3.		<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 sistem lain yang terkait dengan aliran data dari sistem yang dimodelkan</p> <p>catatan :</p> <p>nama yang digunakan pada masukan (<i>input</i>) atau keluaran (<i>output</i>) berupa kata benda.</p>



No	Simbol	Keterangan
4.		<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>)</p> <p>catatan :</p> <p>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>

(Sumber: Sukamto dan Salahuddin, 2014:71)

Sukamto dan Salahuddin (2014:72) menjelaskan tentang tahapan-tahapan perancangan dengan menggunakan *Data Flow Diagram* yaitu:

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 kedetailan modul tersebut. Apabila modul tersebut sudah cukup detail dan rinci maka modul tersebut sudah tidak perlu 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

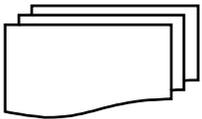
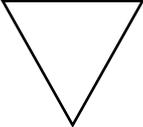
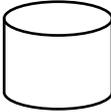
DFD Level 3,4,5, dan seterusnya merupakan *breakdown* dari modul pada DFD Level di-atasnya. *Breakdown* pada level 3,4,5, dan seterusnya aturannya sama persis dengan DFD Level 1 atau 2.

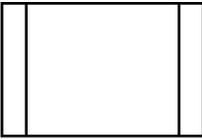
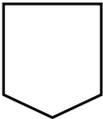
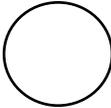
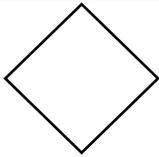
#### 2.2.3. Pengertian *Blockchart*

Kristanto (2011:68) menjelaskan bahwa, “*Block chart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu. Pembuatan *block chart* harus memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi.”

Kristanto (2011:68), menjelaskan simbol-simbol yang sering digunakan dalam *block chart* dapat dilihat pada tabel berikut ini:

**Tabel 2.2.** Simbol-simbol *Blockchart*

No.	Simbol	Keterangan
1.		Menandakan dokumen, bisa dalam bentuk surat, formulir, buku/benda/berkas/cetakan.
2.		Multi dokumen
3.		Proses manual
4.		Proses yang dilakukan oleh komputer
5.		Menandakan dokumen yang diarsipkan (arsip manual)
6.		Data penyimpanan (data storage)

No.	Simbol	Keterangan
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.		Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang sama
10.		Terminasi yang menandakan awal dan akhir dari suatu aliran
11.		Pengambilan keputusan (decision)
12.		Layar peraga (monitor)
13.		Pemasukan data secara manual

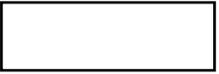
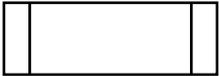
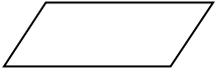
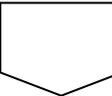
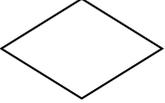
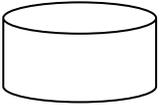
(Sumber: Kristanto, 2011:68)

#### 2.2.4. Pengertian *Flowchart*

Supardi (2013:51) menjelaskan bahwa, “*Flowchart* adalah merupakan diagram alur yang digunakan sistem analis dalam membuat atau menggambarkan logika program.”

Supardi (2013:51) menjelaskan simbo-simbol yang digunakan *flowchart* adalah sebagai berikut:

Tabel 2.3. Simbol-simbol dalam *Flowchart*

No	Simbol	Keterangan
1.		Terminal menyatakan awal atau akhir dari suatu algoritma.
2.		Menyatakan proses.
3.		Proses yang terdefinisi atau sub program.
4.		Persiapan yang digunakan untuk memberi nilai awal suatu besaran.
5.		Menyatakan masukan dan keluaran ( <i>input/output</i> ).
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.



No	Simbol	Keterangan
11.		Menyatakan <i>input/output</i> menggunakan disket.
12.		Menyatakan operasi yang dilakukan secara manual.
13.		Menyatakan <i>input/output</i> dari kartu plong.
14.		Menyatakan arah aliran pekerjaan (proses).
15.		<i>Multidocument</i> (banyak dokumen).
16.		<i>Delay</i> (penundaan atau kelambatan).

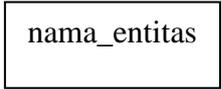
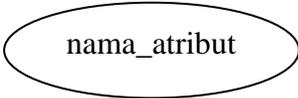
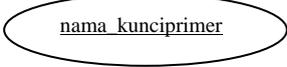
(Sumber: Supardi, 2013:53)

### 2.2.5. Pengertian ERD (*Entity Relational Diagram*)

Sukanto dan Salahuddin (2014:289), menjelaskan bahwa, “*Entity Relationship Diagram* (ERD) adalah pemodelan awal basis data yang paling banyak digunakan yang dikembangkan berdasarkan teori himpunan dalam bidang matematika untuk pemodelan basis data relasional.”

Sukanto dan Salahuddin (2014:50), menjelaskan simbol-simbol yang digunakan dalam ERD, yaitu:

Tabel 2.4. Simbol-simbol *Entity Relationship Diagram (ERD)*

No	Simbol	Keterangan
1.	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 komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.
2.	Atribut 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas
3.	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)
4.	Relasi 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja
5.	Asosiasi / <i>association</i> 	Penghubung antara relasi dan entitas di mana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian. Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan <i>one to many</i> menghubungkan entitas A dan entitas B.

(Sumber: Sukamto dan Salahuddin, 2014:50)



### 2.2.6. Pengertian Kamus Data

Sukamto dan Salahuddin (2013:73), “Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan).”

Asropudin (2015:30), “Kamus data adalah katalog fakta tentang data dan kebutuhan informasi suatu sistem informasi.”

Sukamto dan Salahuddin (2013:74), menjelaskan simbol-simbol yang di gunakan dalam kamus data, yaitu :

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

No	Simbol	Arti
1	=	disusun atau terdiri atas
2	+	Dan
3	[]	baik ...atau...
4	{ } <sup>n</sup>	n kali diulang/ bernilai banyak
5	( )	data operasional
6	*...*	batas komentar

(Sumber: Sukamto dan Salahuddin, 2013:74)

## 2.3. Teori Program

### 2.3.1. Pengertian *Microsoft Visual Basic 2010*

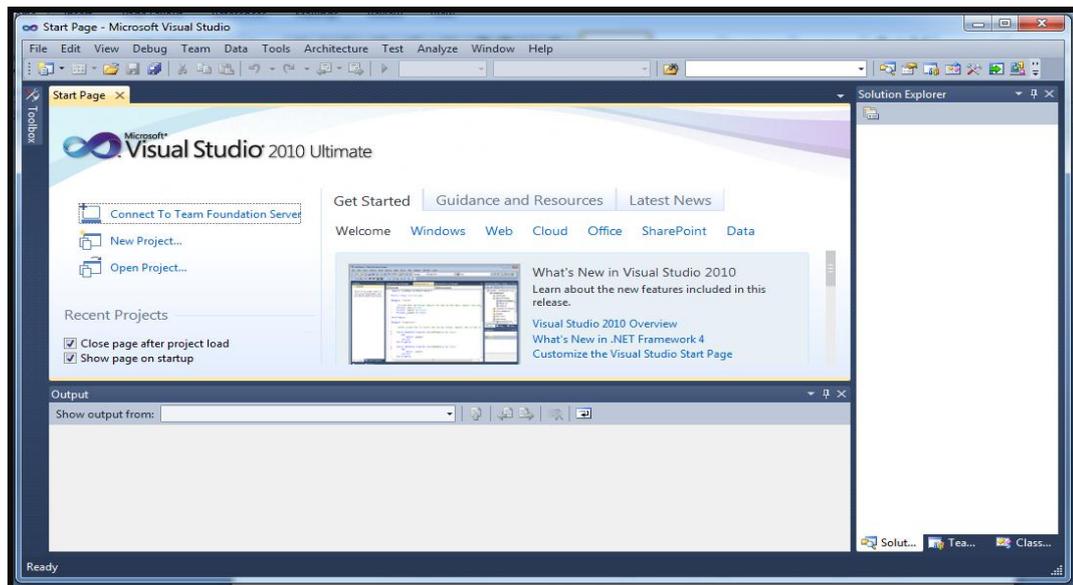
Komputer (2010:03), “*Visual Basic* merupakan salah satu aplikasi pemrograman visual yang paling banyak digunakan dan cukup populer karena kemudahan penggunaannya.”

Komputer (2010:03), “*Visual Basic 2010* merupakan varian terbaru dari aplikasi pemrograman aplikasi visual basic yang memiliki perubahan signifikan pada berbagai fasilitas pendukungnya dibandingkan versi sebelumnya.”



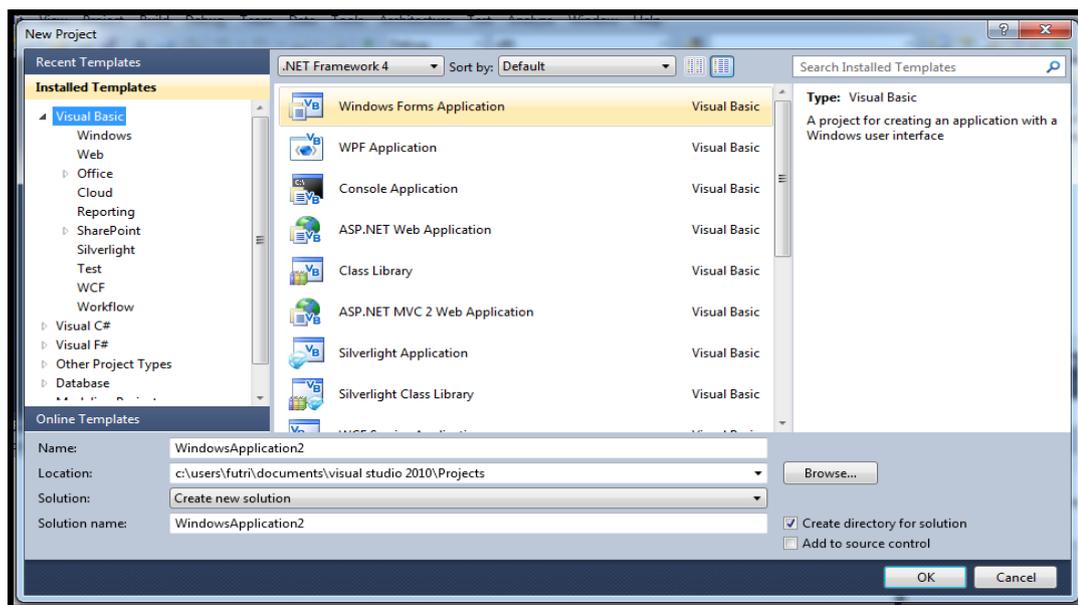
### 2.3.2. Mengenal Intergrated Development Environment (IDE) Visual Basic 2010

Pada waktu Visual Basic dijalankan, maka akan tampil sebuah start page seperti terlihat pada gambar 2.2.



**Gambar 2.2** Tampilan Awal *Visual Basic* 2010

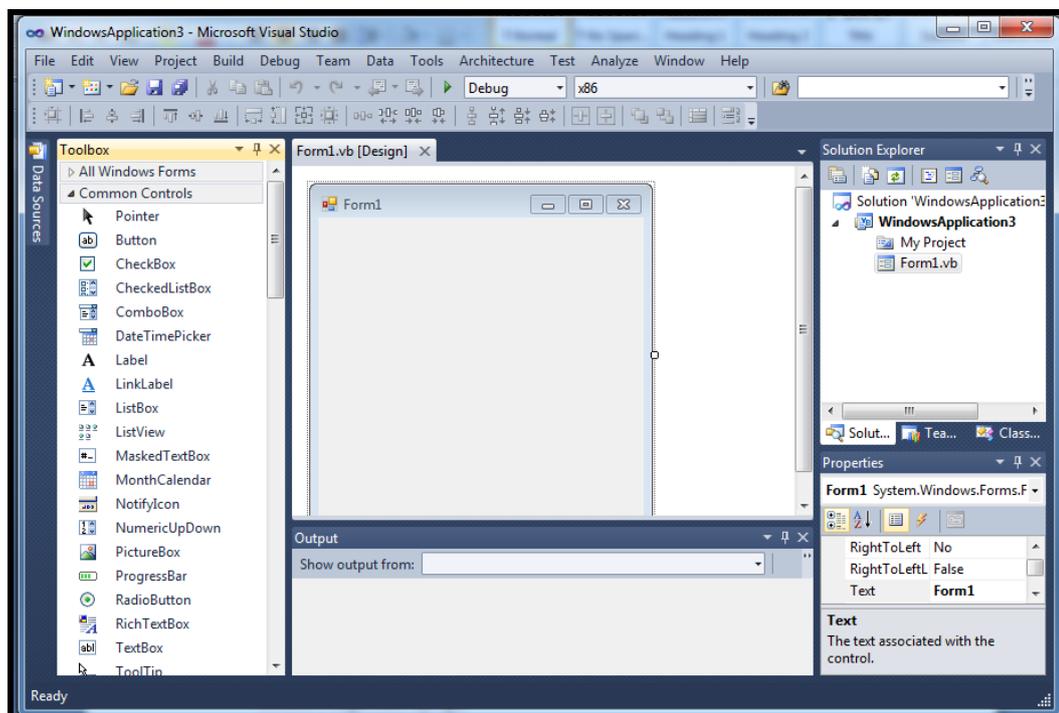
Untuk Membuat proyek baru, klik tombol *New Project* > Pilih *Windows Form Application* > Ok, Lalu beri nama proyek baru tersebut di bagian Name. Jika ingin membuka proyek yang sudah ada, maka gunakan tombol *Open Project*.



**Gambar 2.3** Kotak Dialog New Project.



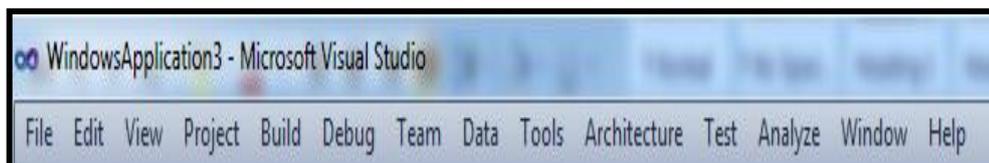
Selanjutnya muncul *Visual Basic* 2010 IDE yaitu tempat untuk membangun aplikasi *Visual Basic.NET*. Pada IDE *Visual Basic* 2010 untuk *windows application default* telah terdapat sebuah *form*. *Form* tersebut bernama *Form1* yaitu tempat untuk meletakkan kontrol-kontrol atau komponen-komponen untuk membuat sebuah *aplikasi Windows Form* dan aplikasi inilah yang biasanya disebut dengan GUI. Pada IDE *Visual Studio* 2010 terdapat menu *bar*, *toolbar*, *solution explorer*, dan *properties window* seperti terlihat pada gambar 2.4.



Gambar 2.4 IDE *Visual Basic* 2010.

Keterangan :

1. Menu *Bar* adalah bagian dari IDE yang terdiri dari atas perintah-perintah untuk mengatur IDE, mengedit kode, dan mengeksekusi program.

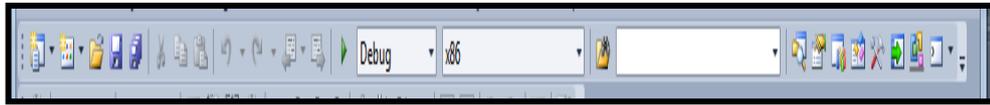


Gambar 2.5 Menu *Bar*



## 2. *Toolbar*

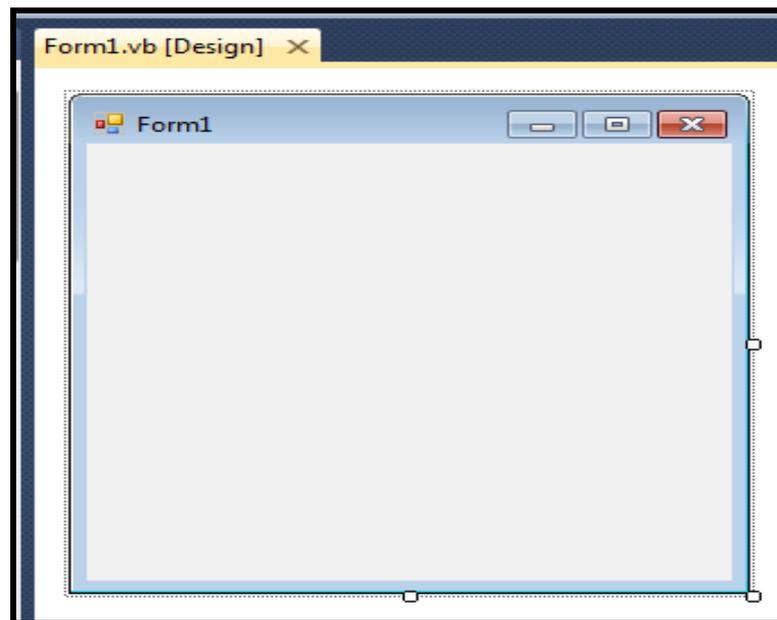
*Toolbar* fungsinya sama seperti menu. Bedanya pada *toolbar* pilihan-pilihannya berbentuk *icon*.



**Gambar 2.6 *Toolbar***

## 3. *Form Designer*

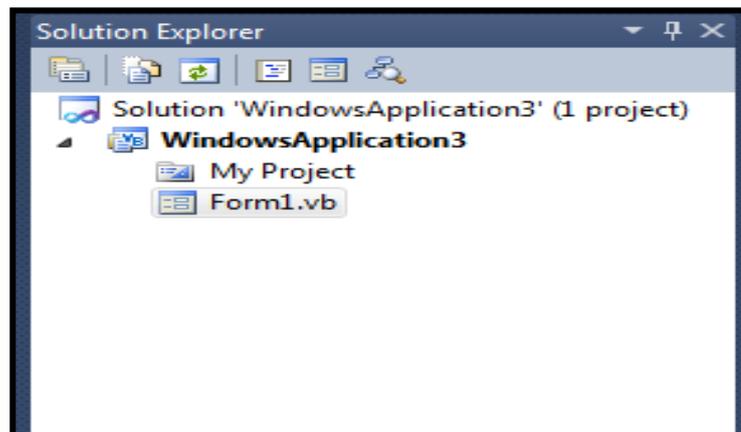
*Form* merupakan fitur dari Visual Basic 2010 yang digunakan untuk membuat desain antarmuka atau interface dari aplikasi yang anda kembangkan.



**Gambar 2.7 *Form***

## 4. *Solution Explorer*

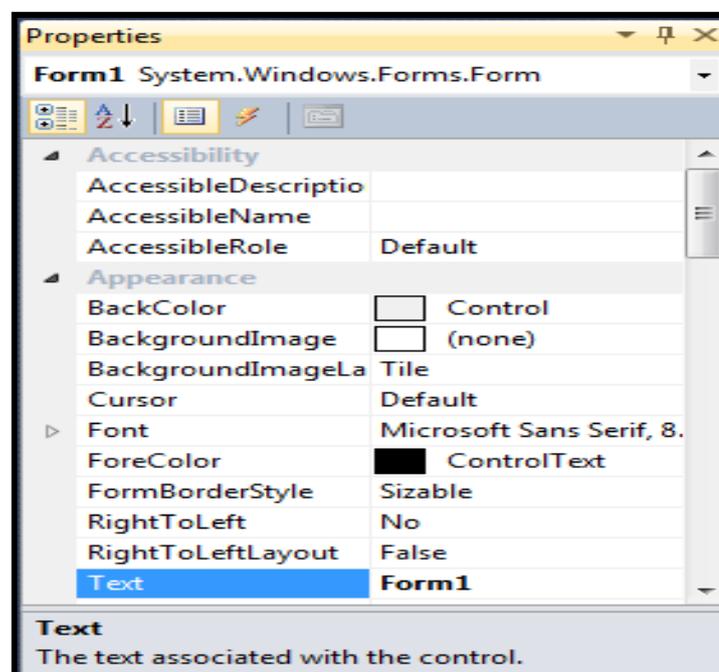
*Solution Explorer* merupakan fitur yang terletak di sebelah kanan atas di bawah menu aplikasi yang digunakan untuk menampilkan daftar desain form dengan struktur tree ddari project yang sedang dibuka.



**Gambar 2.8** *Solution Explorer*

#### 5. *Properties Window*

*Properties Window* adalah tempat untuk melakukan pengaturan property dari objek-objek yang digunakan dalam desain form yang sedang anda buat.

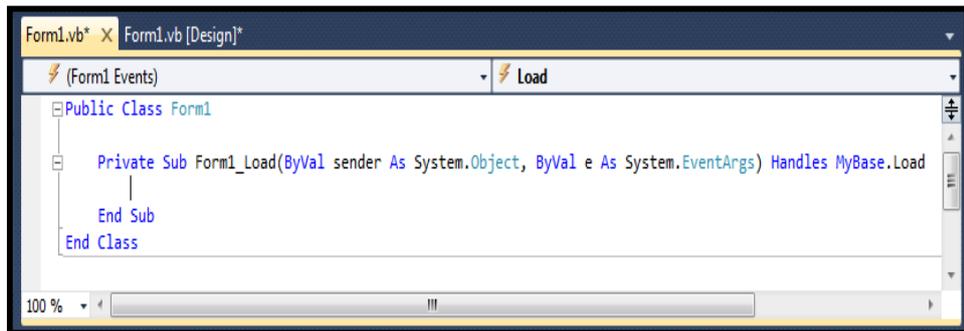


**Gambar 2.9** *Properties Window*



## 6. Code Editor

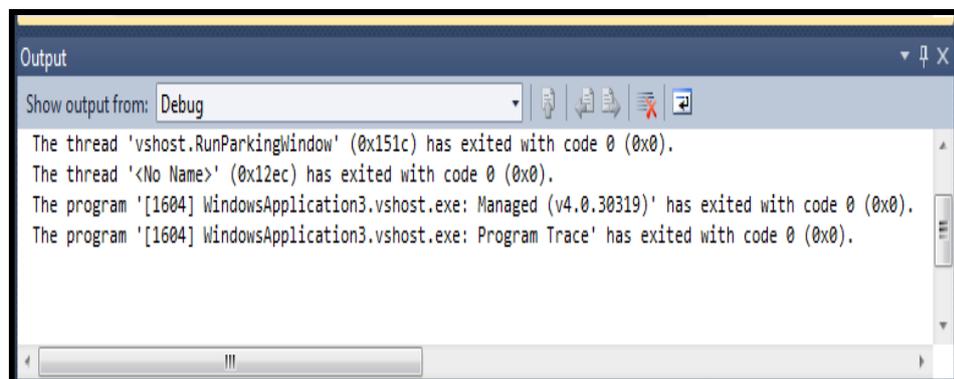
*Code Editor* merupakan fitur Visual Basic 2010 yang digunakan untuk menambahkan kode program dari aplikasi atau project yang sedang anda kerjakan.



**Gambar 2.10 Code Editor**

## 7. Output Window

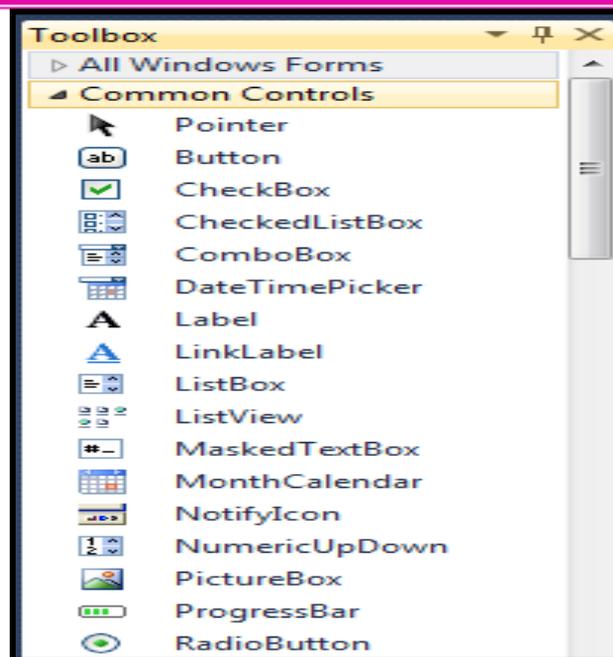
*Output Window* menunjukkan langkah-langkah dalam mengompilasi aplikasi.



**Gambar 2.11 Output Window**

## 8. Toolbox

*Toolbox* digunakan untuk menampilkan daftar dari komponen baik visual maupun nonvisual yang bisa ditambahkan ke dalam desain form yang dibuat.



Gambar 2.12 Toolbox

### 2.3.3. Pengertian Basis Data (*Database*)

Sutanta (2011:35), menjelaskan “*Database* (basis data) merupakan sekumpulan dari bermacam-macam tipe record yang memiliki hubungan antar-*record* dan rincian data terhadap obyek tertentu.”

Sutabri (2012:47), “Basis data (*database*) merupakan kumpulan data yang saling berkaitan dan berhubungan satu sama lain, tersimpan di perangkat keras komputer dan menggunakan perangkat lunak untuk memanipulasinya.”

Sukanto dan Salahuddin (2014:43), “Basis Data (*database*) adalah system terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan.”

### 2.3.4. Sekilas Tentang *MySQL*

#### 2.3.4.1. Pengertian *MySQL*

Komputer (2014:73) menjelaskan bahwa, “*MySQL* merupakan sistem manajemen database yang bersifat *relational*. Artinya, data yang dikelola dalam database akan diletakkan pada beberapa tabel yang terpisah sehingga manipulasi



data akan menjadi jauh lebih cepat. *MySQL* dapat digunakan untuk mengelola database mulai dari yang kecil sampai dengan yang sangat besar.”

#### 2.3.4.2. Pengertian *PhpMyAdmin*

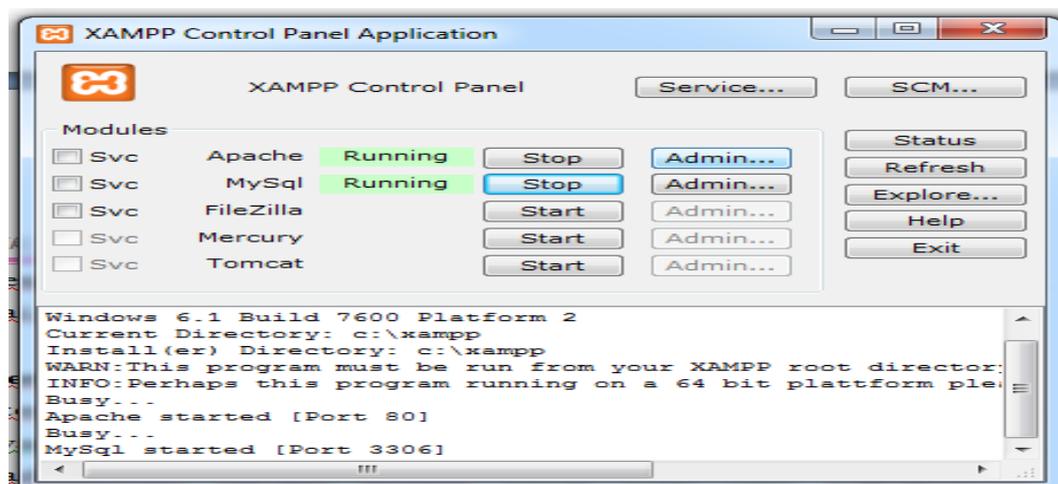
Komputer (2014:46) menjelaskan bahwa, “*PhpMyAdmin* adalah perangkat lunak bebas yang ditulis dalam bahasa pemrograman *PHP* yang digunakan untuk menangani administrasi *MySQL* melalui *world wide web*. *PhpMyAdmin* mendukung berbagai operasi *MySQL*, diantaranya mengelola basis data, tabel-tabel, *fields*, relasi, indeks, *users*, *permissions*, dan lain-lain.”

Sibero (2013:52) menjelaskan bahwa, “*PhpMyAdmin* adalah aplikasi *web* yang dibuat oleh *phpmyadmin.net*. *PhpMyAdmin* digunakan untuk administrasi database *MySQL*.”

Dari kedua pengertian di atas penulis dapat menyimpulkan bahwa *phpMyAdmin* adalah sebuah perangkat lunak yang digunakan sebagai administrasi *MySQL* yang mengelola basis data, tabel-tabel, *fields*, relasi, *indeks*, *users*, *permissions*, dan lain-lain.

#### 2.3.4.3. Pengertian *XAMPP*

Komputer (2014:72) menjelaskan bahwa, “*XAMPP* merupakan singkatan dari *X* (empat sistem operasi apa pun), *Apache*, *MySQL*, *PHP*, dan *Perl*. *XAMPP* adalah *tool* yang menyediakan paket perangkat lunak dalam satu buah paket.”



Gambar 2.13 *XAMPP*