



BAB II TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Pengertian Komputer

Menurut Sibero (2013:9), “Istilah komputer berasal dari kata “*Compute*” yang juga berasal dari bahasa latin “*Computare*” yang artinya menghitung”.

2.1.2. Pengertian Data

Hartono (2013:15) menyatakan, “Data adalah hasil pengukuran dan pencatatan terhadap fakta tentang sesuatu, keadaan, tindakan atau kejadian”.

Asropudin (2013:22) menyatakan, “Data adalah kumpulan dari angka-angka maupun karakter-karakter yang tidak memiliki arti. Data dapat diolah sehingga menghasilkan informasi”.

2.1.3. Pengertian Sistem

Hartono (2013:10) menyatakan, “Sistem mengandung dua macam konotasi, yakni suatu benda atau entitas (yaitu himpunan dari berbagai bagian atau komponen), dan sekaligus juga suatu proses atau metode atau cara untuk mencapai tujuan (yaitu saling berhubungan secara terorganisasi berdasarkan fungsi-fungsinya)”.

Sutabri (2012:6) menyatakan, ”Sistem adalah sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu”.

2.1.3.1. Siklus Hidup Pengembangan Sistem

Menurut Sutabri (2012:37), “Siklus hidup pengembangan sistem suatu bentuk yang digunakan untuk menggambarkan tahapan utama dan langkah-langkah pada tahapan tersebut dalam proses pengembangan sistem”.



Fase-fase dari siklus hidup pengembangan sistem informasi yaitu, sebagai berikut:

a. Fase Perencanaan

Perencanaan pengembangan sistem informasi bertujuan untuk mengidentifikasi dan memprioritaskan sistem informasi yang akan dikembangkan, sasaran-sasaran yang ingin dicapai, jangka waktu pelaksanaan, serta mempertimbangkan dana yang tersedia dan siapa yang akan melaksanakan.

b. Fase Pengembangan

Fase pengembangan sistem informasi disebut juga sebagai siklus hidup pengembangan sistem informasi yang garis besarnya terdiri dari enam langkah. Tahapan utama dalam proses pengembangan sistem informasi adalah sebagai berikut:

1. Investigasi Sistem

Manfaat dari fase penyelidikan ini adalah untuk menentukan masalah-masalah atau kebutuhan yang timbul. Hal ini memerlukan pengembangan sistem secara menyeluruh ataukah ada usaha lain yang dapat dilakukan untuk memecahkannya.

2. Analisis Sistem

Tahap analisis bertitik-tolak pada kegiatan-kegiatan dan tugas-tugas di mana sistem yang berjalan dipelajari lebih mendalam, konsepsi, dan usulan dibuat untuk menjadi landasan bagi sistem yang baru yang akan dibangun.

3. Desain Sistem

Pada tahap ini sebagian besar kegiatan yang berorientasi ke komputer dilaksanakan. Spesifikasi perangkat keras dan perangkat lunak (HW/SW) yang telah disusun pada tahap sebelumnya ditinjau kembali dan disempurnakan.



4. Implementasi Sistem

Tahap ini adalah prosedur yang dilakukan untuk menyelesaikan desain sistem yang ada dalam dokumen desain sistem yang disetujui dan menguji, menginstall dan memulai penggunaan sistem baru atau sistem yang diperbaiki.

5. Pemeliharaan Sistem

Disarankan adanya dua tahap review yang harus dilaksanakan. Pertama kali tidak terlalu lama setelah penerapan sistem, dimana tim proyek masih ada dan masing-masing anggota masih memiliki ingatan yang segar atas sistem yang mereka buat. *Review* berikutnya dapat dilakukan kira-kira setelah enam bulan berjalan. Tujuannya adalah untuk menyakinkan apakah sistem tersebut berjalan sesuai dengan tujuan semula dan apakah masih ada perbaikan atau penyempurnaan yang harus dilakukan.

Beberapa cara dapat ditempuh dalam penerapan tahapan pengembangan sistem informasi, yaitu sebagai berikut:

1. **Waterfall.** Setiap tahap harus diselesaikan terlebih dahulu secara penuh sebelum diteruskan ke tahap berikutnya untuk menghindari terjadinya pengulangan tahapan.
2. **Iterasi/Spiral.** Tahapan-tahapan tersebut dilaksanakan dengan memakai teknik iterasi atau dimana suatu proses dilaksanakan secara berulang-ulang sampai didapatkan hasil yang diinginkan.

2.1.4. Pengertian Informasi

Sutarman (2012:14) menyatakan, “Informasi adalah sekumpulan fakta (data) yang diorganisasikan dengan cara tertentu sehingga mereka mempunyai arti bagi si penerima”.

Sibero (2013:10) menyatakan, “Informasi adalah sekumpulan data yang memiliki maksud dan tujuan serta dapat memberikan keterangan akurat yang diperlukan dalam pengambilan keputusan”.



2.1.4.1. Nilai dan Kualitas Informasi

Menurut Sutabri (2012:10), nilai dari informasi ditentukan dari 2 (dua) hal, yaitu manfaat dan biaya untuk mendapatkannya. Suatu informasi dikatakan bernilai apabila manfaat yang diperoleh lebih berharga dibandingkan dengan biaya untuk mendapatkannya.

Nilai informasi ini didasarkan atas 10 (sepuluh) sifat, yaitu:

a. Mudah diperoleh

Sifat ini menunjukkan kemudahan dan kecepatan untuk memperoleh informasi.

b. Luas dan lengkap

Sifat ini menunjukkan kelengkapan isi informasi. Hal ini tidak hanya mengenai volumenya, akan tetapi juga mengenai keluaran informasinya.

c. Ketelitian

Sifat ini berhubungan dengan tingkat kebebasan dari kesalahan keluaran informasi.

d. Kecocokan

Sifat ini menunjukkan seberapa baik keluaran informasi dalam hubungannya dengan permintaan para pemakai. Isi informasi harus ada hubungannya dengan masalah yang sedang dihadapi sedangkan semua keluaran yang lainnya tidak berguna.

e. Ketepatan waktu

Sifat ini berhubungan dengan waktu yang dilalui, yang lebih pendek dari siklus untuk mendapatkan informasi.

f. Kejelasan

Sifat ini menunjukkan tingkat kejelasan informasi. Informasi hendaknya terbebas dari istilah-istilah yang tidak jelas.

g. Keluwesan

Sifat ini berhubungan dengan apakah informasi tersebut dapat digunakan untuk membuat lebih dari satu keputusan, tetapi juga apakah dapat digunakan untuk lebih dari seorang pengambil keputusan.



h. Dapat dibuktikan

Sifat ini menunjukkan sejauh mana informasi itu dapat diuji oleh beberapa pemakai hingga sampai didapatkan kesimpulan yang sama.

i. Tidak ada prasangka

Sifat ini berhubungan dengan ada tidaknya keinginan untuk mengubah informasi tersebut guna mendapatkan kesimpulan yang telah diarahkan sebelumnya.

j. Dapat diukur

Sifat ini menunjukkan hakikat informasi yang dihasilkan oleh sistem informasi formal.

Kualitas dari suatu informasi tergantung dari 2 (tiga) hal, yaitu informasi harus akurat (accurate), tepat waktu (timelines), dan relevan (relevance).

a. Akurat (accurate)

Informasi harus bebas dari kesalahan dan tidak bias atau menyesatkan. Akurat juga berarti bahwa informasi harus jelas mencerminkan maksudnya.

b. Tepat waktu (timelines)

Informasi yang disampaikan kepada penerima tidak boleh terlambat.

c. Relevan (relevance)

Informasi tersebut mempunyai manfaat untuk pemakainya.

2.1.5. Pengertian Sistem Informasi

Sibero (2013:10) menyatakan, “Sistem Informasi adalah hasil pengolahan dari kumpulan-kumpulan informasi yang saling terkait antara satu dan lainnya”.

Sutabri (2012:38) menyatakan, “Sistem Informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan laporan-laporan yang diperlukan oleh pihak luar tertentu”.



2.1.6. Pengertian Proyek

Sukamto dan Shalahuddin (2013:69) menyatakan, “Proyek adalah urutan kegiatan yang unik, kompleks, dan saling terkait, memiliki satu tujuan, dan tujuan harus diselesaikan dalam waktu tertentu, sesuai anggaran, dan memenuhi spesifikasi”.

2.1.7. Pengertian Aplikasi Rencana dan Realisasi Proyek Konstruksi Pada PT Feberco Palembang

Aplikasi Rencana dan Realisasi Proyek Konstruksi Pada PT Feberco Palembang adalah suatu aplikasi yang dibangun untuk proses perencanaan menghitung berapa besar kecilnya dana yang diperlukan untuk pelaksanaan kegiatan pekerjaan baik mulai dana operasional hingga dana material bertujuan untuk dapat menentukan budget dana yang harus disediakan dengan menggunakan aplikasi *Visual Basic .NET* dengan menggunakan *Database MySQL XAMPP*.

2.2. Teori Khusus

2.2.1. Pengertian Database

Priyadi (2014:2) menyatakan, “Basis data adalah sekumpulan fakta berupa representasi tabel yang saling berhubungan dan disimpan dalam media penyimpanan secara digital”.

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

Hidayatullah (2014:137) menyatakan, “Basis Data dapat didefinisikan sebagai himpunan kelompok data yang saling berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah”.

Heryanto (2012:1) menyatakan, “Pengertian database (basis data) yang paling sederhana adalah kumpulan dari tabel. Satu tabel mempresentasikan suatu entitas tertentu. Suatu entitas terdiri atas beberapa atribut.



2.2.2. Pengertian Kamus Data

Sukamto dan Shalahuddin (2013:43) menyatakan, “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)”. Kamus data memiliki beberapa symbol untuk menjelaskan informasi tambahan sebagai berikut:

Tabel 2.1 *Simbol-Simbol Kamus Data*

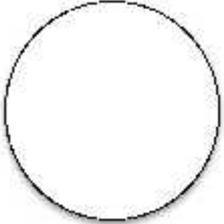
No	Simbol	Uraian
1.	=	Disusun atau terdiri dari
2.	+	Dan
3.	[]	Baik...atau...
4.	{ }n	n kali diulang/bernilai banyak
5.	()	Data opsional
6.	*...*	Batas komentar

2.2.3. Pengertian Data Flow Diagram (DFD)

Sukamto dan Shalahuddin (2013:69), *Data Flow Diagram* (DFD) representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*)”.

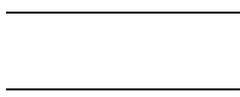
Notasi-notasi pada DFD adalah sebagai berikut:

Tabel 2.2 *Notasi-Notasi Data Flow Diagram*

No	Simbol	Keterangan
1.		Proses atau fungsi atau prosedur; pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang seharusnya menjadi fungsi atau prosedur di dalam kode program



Lanjutan Tabel 2.2 Notasi-Notasi Data Flow Diagram

No	Simbol	Keterangan
2.		<p><i>File</i> atau basis data atau penyimpanan (<i>storage</i>); pada pemodelan perangkat lunak 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 basis data (<i>Entity Relationship Diagram</i> (ERD), <i>Conceptual Data Model</i> (CDM) <i>Physical Data Model</i> (PDM)).</p>
3.		<p>Entitas luar (<i>external entity</i>) atau masukan (<i>input</i>) atau keluaran (<i>output</i>) atau orang yang memakain/berinteraksi dengan perangkat lunak yang dimodelkan atau sistem lain yang terkait dengan aliran data dari sistem yang dimodelkan.</p>
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>

2.2.4. Pengertian ERD

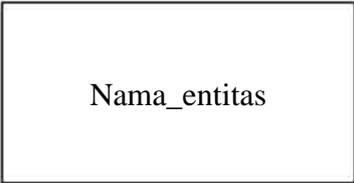
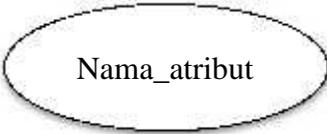
Sukamto dan Shalahuddin (2013:53) menyatakan, “ERD (*Entity Relationship Diagram*) adalah bentuk paling awal dalam melakukan perancangan basis data relasional”.

Hubungan pada ERD yang paling umum pada umumnya pada binary relationship terdiri atas:

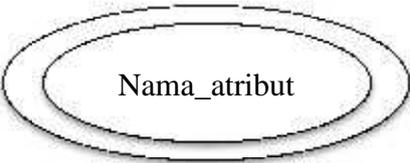
1. One to one (1:1)
2. One to many (1:*)
3. Many to many (*:*)



Tabel 2.3 Simbol-Simbol Entity Relationship Diagram

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 	Field atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses record 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)

Lanjutan **Tabel 2.3** Simbol-Simbol Entity Relationship Diagram

No	Simbol	Keterangan
4.	Atribut multivalui/ <i>multivalue</i> 	Field atau kolom data yang butuh disimpan dalam suatu entitas yang dapat dimiliki nilai lebih dari satu
5.	Relasi 	Relasi yang menghubungkan antarentitas; biasanya diawali dengan kata kerja.
6.	Asosiasi/ <i>association</i> 	Penghubungan antara relasi dan entitas dimana di kedua ujungnya memiliki multiplicity kemungkinan jumlah pemakaian.

2.2.5. Pengertian Blokchart

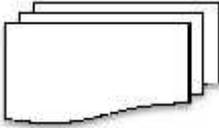
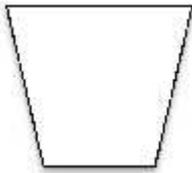
Kristanto (2008:68), "Blokchart berfungsi untuk memodelkan masukan keluaran, proses, maupun transaksi dengan menggunakan simbol-simbol tertentu".

Adapun simbol-simbol yang sering digunakan dalam *Blokschart* dapat dilihat pada tabel:

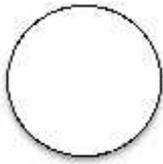
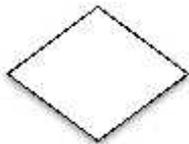
Tabel 2.4 Simbol-Simbol dalam Blockchart

No	Simbol	Keterangan
1.		Document, digunakan dalam menandakan suatu dokumen, bisa dalam bentuk surat, formulir, buku/benda/berkas atau cetakan.

Lanjutan **Tabel 2.4** *Simbol-Simbol dalam Blockchart*

No	Simbol	Keterangan
2.		Multi Document, digunakan dalam menandakan Multi dokumen, bisa dalam bentuk surat, formulir, buku/benda/berkas atau cetakan.
3.		Manual Operational, digunakan dalam bentuk proses yang dilakukan secara manual.
4.		Process, suatu proses yang dilakukan dengan menggunakan komputer.
5.		Menandakan dokumen yang diarsipkan (arsip manual).
6.		Data penyimpanan (Data Storage).
7.		Predifined Process, digunakan dalam proses apa saja yang tidak terdefinisi termasuk aktifitas fisik.
8.		Off-Page Connector, digunakan dalam termisi yang mewakili simbol-simbol tertentu untuk digunakan pada aliran halaman lain.

Lanjutan **Tabel 2.4** Simbol-Simbol dalam Blockchart

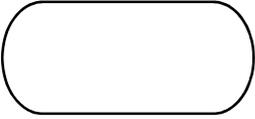
No	Simbol	Keterangan
9.		Connector, digunakan dalam terminasi yang mewakili simbol-simbol tertentu yang digunakan pada aliran lain pada halaman sama.
10.		Terminasi yang menandakan awal dan akhir dari suatu aliran .
11.		Decision, digunakan dalam pengambilan keputusan.
12.		Layar peraga, digunakan sebagai monitor.
13.		Manual input, digunakan dalam pemasukan data secara manual.

2.2.6. Pengertian Flowchart

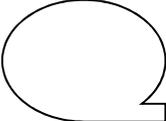
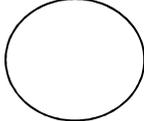
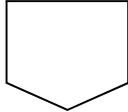
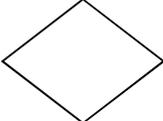
Indrajani (2015:36), *Flow chart* adalah penggambaran secara grafik dari langkah-langkah dan urutan prosedur suatu program.

Indrajani (2015:38), menjelaskan simbol-simbol dalam *Flowchart* adalah sebagai berikut:

Tabel 2.5 Simbol-simbol dalam *Flowchart*

No.	Simbol	Arti
1.		Simbol Start atau End yang mendefinisikan awal atau akhir dari sebuah <i>flowchart</i> .

Lanjutan Tabel 2.5 Simbol-simbol dalam *Flowchart*

No.	Simbol	Arti
2.		Simbol pemrosesan yang terjadi pada sebuah alur kerja.
3.		Simbol yang menyatakan bagian dari program (sub program).
4.		Simbol masukan atau keluaran dari atau ke sebuah pita magnetic.
5.		Simbol Input/Output yang mendefinisikan masukan dan keluaran proses.
6.		Simbol konektor untuk menyambung proses pada lembar kerja yang sama.
7.		Simbol konektor untuk menyambung proses pada lembar kerja yang berbeda.
8.		Simbol masukan atau keluaran dari atau ke sebuah dokumen.
9.		Simbol untuk memutuskan proses lanjutan dari kondisi tertentu.
10.		Simbol database atau basis data.
11.		Simbol yang menyatakan piranti keluaran, seperti layar monitor, <i>printer</i> , dll.
12.		Simbol yang mendefinisikan proses yang dilakukan secara manual.
13.		Simbol untuk menghubungkan antar proses atau antar symbol



2.2.7. Pengertian *Event List*

Kristanto (2008:64), "Daftar kejadian digambarkan dalam bentuk kalimat sederhana dan berfungsi untuk memodelkan kejadian yang terjadi dalam lingkungan sehari-hari dan membutuhkan tanggap atau respon dari sistem".

Adapun cara mendeskripsikan daftar kejadian adalah sebagai berikut:

1. Pelaku adalah entity luar, jadi bukan sistem.
2. Menguji setiap entity luar dan mencoba mengevaluasi setiap entity luar yang terjadi pada sistem.
3. Hati-hati dengan kejadian yang spesifik, yang tidak sengaja menyatu dalam paket yang sama.
4. Harus diingat bahwa kejadian yang dimodelkan bukan hanya interaksi normal antara sistem dengan entity luar, karena itu harus dievaluasi kebutuhan sistem untuk menanggapi kejadian yang gagal.
5. Setiap aliran keluaran sebaliknya merupakan respon dari kejadian.
6. Setiap kejadian yang tidak berorientasi pada waktu dalam daftar kejadian sebaiknya mempunyai masukan sehingga sistem dapat mendeteksi kejadian yang berlangsung.
7. Setiap kejadian sebaiknya menghasilkan keluaran langsung sebagai respon atau disimpan dalam berkas untuk bahan masukan.

2.3. Teori Program

2.3.1. Pengertian *MySQL*

Kadir (2013:15) menjelaskan, *MySQL* adalah nama *database server*. *Database server* adalah *server* yang berfungsi untuk menangani *database*. *Database* adalah suatu pengorganisasian data dengan tujuan memudahkan penyimpanan dan pengaksesan data. Dengan menggunakan *MySQL*, kita bisa menyimpan data dan kemudian data bisa diakses dengan cara yang mudah dan cepat.

Nugroho (2013:26) menyatakan, "*MySQL* itu adalah software atau program database server, dia itu software atau programnya".

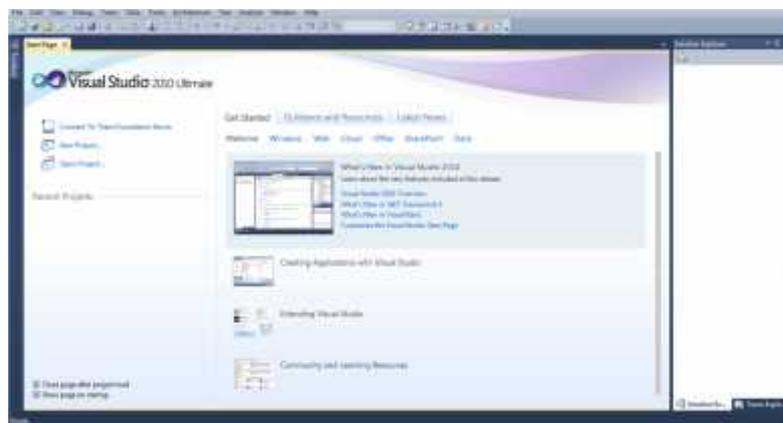


Winarno dkk (2014:102) menyatakan, “MySQL merupakan tipe relasional yang artinya MySQL menyimpan datanya dalam bentuk tabel-tabel yang saling berhubungan.

2.3.2. Pengenalan Visual Basic .NET

2.3.2.1. IDE (*Integrated Development Environment*) Visual Basic 2010

Pada waktu Visual Basic 2010 dijalankan, akan tampil sebuah Start Page dari seperti terlihat pada gambar 2.1.



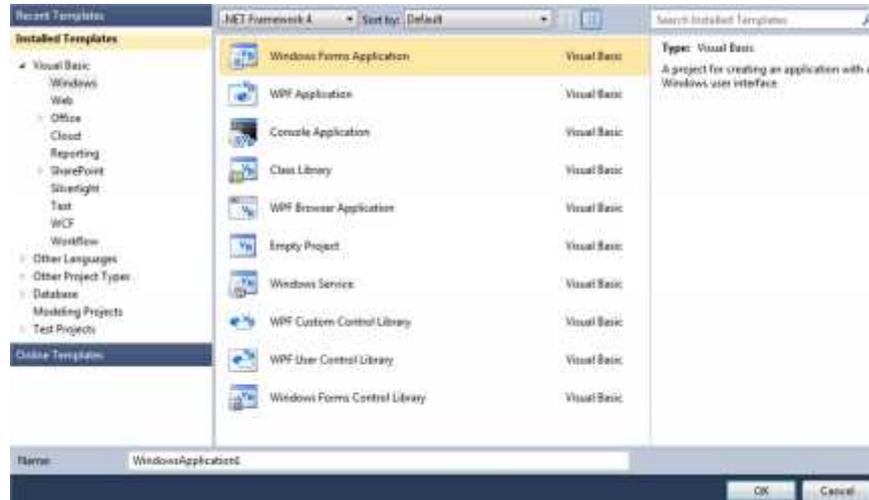
Gambar 2.1 Start Page dari Visual Studio 2010

Untuk membuka proyek yang ada digunakan tombol **Open Project** atau langsung mengklik pada daftar proyek yang ditampilkan sedangkan untuk membuat sebuah proyek baru, klik tombol **New Project**. Setelah itu akan muncul kotak dialog **New Project**. Pada kotak pilih **Other Languages > Visual Basic > Windows > Windows Form Application**. Untuk memberi nama proyek dapat dilakukan pada bagian Name, tentukan posisi penyimpanan *file-file* proyek dan tentukan nam *solution*-nya dan tekan **OK** (Gambar 2.1). selanjutnya muncul **Visual Basic 2010 IDE** tempat untuk membangun aplikasi Visual Basic .NET

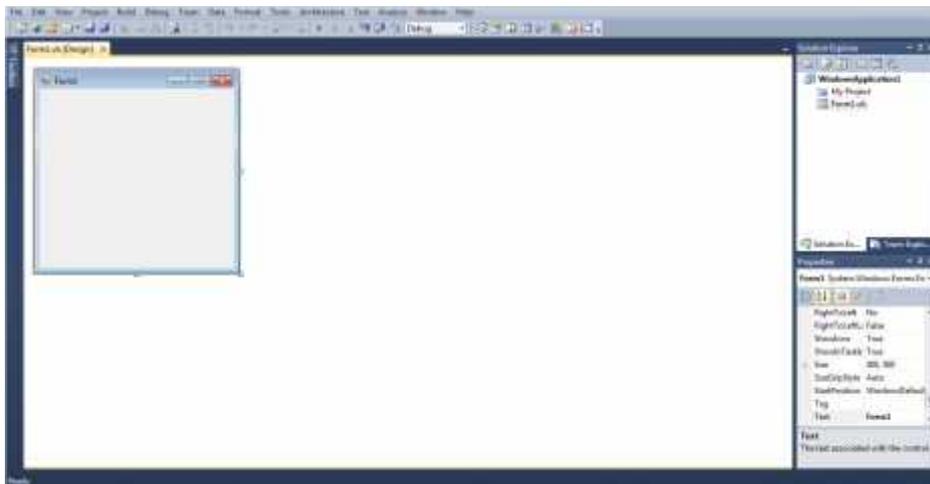
Pada **IDE Visual Studio 2010** untuk *Windows Application* secara *default* telah terdapat sebuah *form*. Form tersebut bernama form1. Pada form inilah tempat meletakkan kontrol-kontrol atau komponen-komponen untuk membuat sebuah aplikasi *Windows Form* dan control-kontrol dari aplikasi inilah yang biasanya disebut dengan GUI (*Graphical User Interface*). Jadi user akan berinteraksi dengan sebuah program aplikasi melalui GUI. Pada IDE Visual



Studio 2010 terdapat menu *bar*, *toolbar*, *toolbox*, *solution explorer*, dan *properties window*.



Gambar 2.2 Kotak Dialog New Project



Gambar 2.3 IDE Visual Studio 2010

a. Menu Bar

Menu bar adalah bagian dari IDE yang terdiri atas perintah-perintah untuk mengatur IDE, mengedit kode, dan mengeksekusi program. Di dalam menu bar, perintah-perintah di kelompokkan ke dalam beberapa bagian sesuai jenis perintah tersebut. Menu bar pada Visual Studio 2010 terlihat seperti pada gambar 2.4.



Gambar 2.4 *Menu Bar*

Pada IDE Visual Studio 2010, terdapat 10 menu utama. Menu-menu tersebut antara lain sebagai berikut:

1. Menu File berisi perintah-perintah untuk membuat proyek, membuka proyek, menutup proyek, mencetak data dari proyek dan lain-lain.
2. Menu Edit berisi perintah-perintah untuk undo, cut, paste dan lain-lain.
3. Menu View berisi perintah-perintah untuk menampilkan window-window dari IDE dan toolbar.
4. Menu Project berisi perintah-perintah untuk mengatur proyek dan file-file.
5. Menu Build berisi perintah-perintah untuk meng-compile program.
6. Menu Debug berisi perintah-perintah untuk men-debug dan menjalankan program.
7. Menu Data berisi perintah-perintah untuk berhubungan dengan basis data.
8. Menu Tools berisi perintah-perintah untuk mengakses komponen IDE tambahan dan mengubah IDE.
9. Menu Architecture berisi tools-tools pemodelan data menggunakan ER-Diagram, UML, dsb.
10. Menu Test mengelola pengujian yang akan dilakukan pada proyek.
11. Menu Analyze, menganalisis kode untuk mengetahui performnsinya.
12. Menu Window berisi perintah-perintah untuk mengatur dan menampilkan Windows.
13. Menu Help berisi perintah-perintah untuk mengakses fasilitas bantuan.

b. Toolbar

Toolbar fungsinya sama seperti menu. Bedanya pada toolbar pilihan-pilihan berbentuk icon. Untuk memilih suatu proses yang akan dilakukan, kita tinggal menekan icon yang sesuai dengan proses yang kita inginkan. Bagian toolbar terlihat seperti pada Gambar 2.5.



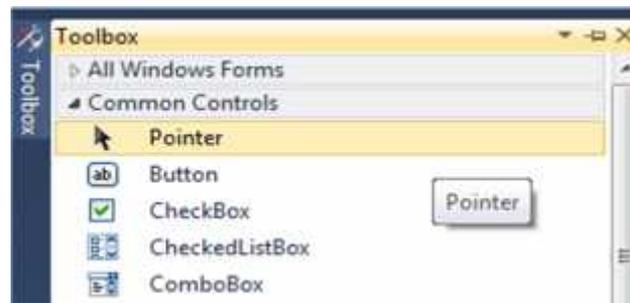
Gambar 2.5 *Toolbar*

Icon-icon pada toolbar adalah pilihan-pilihan pada menu yang sering digunakan untuk membuat program aplikasi. Toolbar memudahkan kita untuk memilih proses yang sering dilakukan tanpa harus memilihnya pada menu. Pada toolbar terdapat beberapa tombol untuk window dari IDE Visual Studio .NET, antara lain Solution Explorer, Properties, dan Toolbox.

c. Toolbox

Toolbox adalah tempat di mana control-control dan komponen-komponen diletakkan. Kontrol dan komponen disimpan pada Toolbox dengan berbagai kategori:

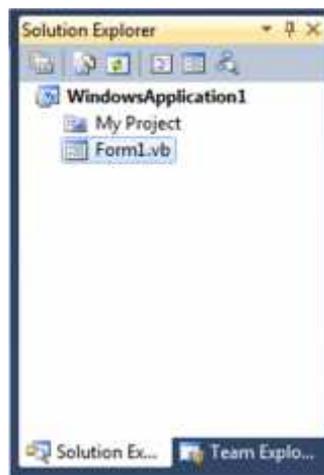
1. Common Controls, berisi kontrol-kontrol umum yang sering digunakan seperti button, label, textbox, dsb.
2. Containers, berisi kontrol penyimpan kontrol lainnya seperti panel, group box, tabcontrol, dsb.
3. Menus & Toolbars, berisi kontrol dan komponen menu, context menu dan toolbar.
4. Data, berisi kontrol dan komponen pengolahan data.
5. Components, berisi komponen-komponen seperti timer, imagelist, dsb.
6. Printing, berisi komponen untuk pencetakan dokumen.
7. Dialogs, berisi komponen untuk berinteraksi dengan pengguna dalam hal membuka file, menyimpan file, membuka folder dll.
8. WPF Interoperability, berisi komponen untuk Windows Presentation Foundation.
9. Reporting, berisi kontrol untuk membuat laporan pada Visual Studio 2010.
10. Visual Basic PowerPacks, berisi beberapa control tambahan untuk menggambar (contoh: oval, garis, persegi) dan fungsi tambahan lainnya.
11. General, jika ada control atau komponen yang mau ditambahkan dan belum memiliki kategori yang jelas, maka bisa dimasukkan ke kategori ini.



Gambar 2.6 *Toolbox*

d. Solution Explorer

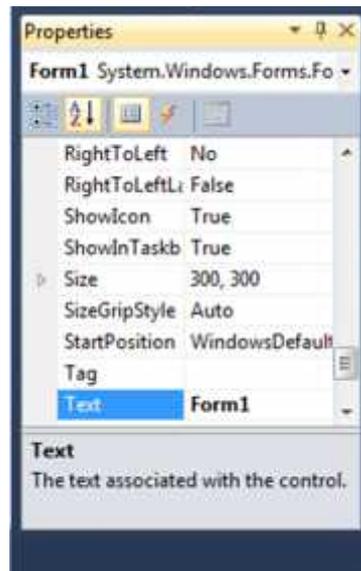
Solution explorer memberikan tampilan daftar file-file dari proyek yang sedang dibuat. Pada jendela solution explorer terdapat beberapa tombol dan tree yang berisi daftar dari file-file yang digunakan dalam proyek. Jika anda tidak dapat menemukan Toolbox di IDE anda, pilih menu View > Solution Explorer atau tekan Ctrl + Alt + L.



Gambar 2.7 *Solution Explorer*

e. Properties Window

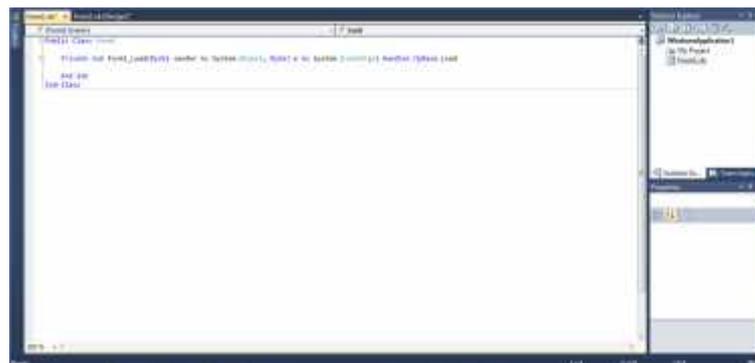
Properties window adalah tempat menyimpan property dari setiap objek kontrol dan komponen. Properties window juga dipakai untuk mengatur property dari objek kontrol dan komponen yang dipakai. Dengan properties window, kita dapat mengubah property yang nantinya akan dipakai sebagai default dari objek kontrol dan komponen pada waktu pertama kali program dieksekusi. Jika anda tidak dapat menemukan Properties Window di IDE anda, pilih menu View > Properties Window atau tekan F4.



Gambar 2.8 Properties Window

f. Code Editor

Code editor adalah tempat dimana kita meletakkan atau menuliskan kode program dari program aplikasi kita. Pada code editor terdapat bagian objek dan event dari control. Pada bagian objek terdapat semua objek yang digunakan pada proyek. Untuk melihat code editor, tekan tombol View Code yang terletak pada bagian Solution Explorer sebelah kiri atas (lihat gambar solution explorer). Bagian code editor terlihat seperti gambar 2.9.



Gambar 2.9 Code Editor

g. Output Window

Output window menunjukkan langkah-langkah dalam mengkompilasi aplikasi. Bagian output window terlihat seperti gambar 2.10.



Gambar 2.10 *Output Window*

2.3.2.2. Form

Form merupakan tempat di mana control-control diletakkan. *Form* juga berfungsi sebagai tempat pembuatan tampilan atau antarmuka (*user interface*) dari sebuah aplikasi Windows. Dengan menggunakan form, kita dapat lebih mudah mengatur tampilan dari aplikasi yang akan dibuat sehingga kita dapat lebih banyak berkonsentrasi pada jalannya program aplikasi tanpa membuang banyak waktu untuk mendesain tampilan program. Secara *default*, pada waktu kita membuat proyek Windows Application, maka telah terdapat sebuah form bernama Form1 pada proyek tersebut.

2.3.2.3. Label

Label adalah sebuah control untuk menampilkan suatu tulisan tertentu. Biasanya digunakan untuk menampilkan tulisan pada *form* sebagai keterangan untuk control-control lain.

Beberapa property label:

Tabel 2.6 *Properti Label*

Properti	Keterangan
Name	Nama dari label
Autosize	Jika bernilai true, ukuran label menyesuaikan dengan teks
BackColor	Warna label
BorderStyle	Bentuk garis tepi label
Cursor	Bentuk kursor waktu berada pada label
Font	Jenis teks dari label
ForeColor	Warna teks pada label
Image	Gambar pada label
ImageAlign	Posisi gambar pada label

Lanjutan **Tabel 2.6** *Properti Label*

Properti	Keterangan
Location	Posisi dari label pada form
RightToLeft	Posisi teks pada label
Size	Ukuran label
Text	Teks atau tulisan pada label
TextAlign	Posisi dari teks pada label
Visible	Jika bernilai true, label kelihatan

2.3.2.4. TextBox

TextBox merupakan control yang dipakai sebagai tempat mengisi maupun menampilkan data. Tulisan yang tampil pada textbox dapat diubah oleh pemakai pada saat program berjalan atau diubah oleh program.

Berikut ini adalah property dari textbox:

Tabel 2.7 *Properti TextBox*

Properti	Keterangan
Name	Nama dari textbox
AutoSize	Jika bernilai true, ukuran textbox menyesuaikan dengan teks
BackColor	Warna textbox
BorderStyle	Bentuk garis tepi textbox
CharacterCasing	Jenis teks dari textbox (huruf besar/kecil/normal)
Enabled	Jika bernilai true, textbox bisa diklik
Font	Jenis teks dari textbox
ForeColor	Warna teks pada textbox
Location	Posisi dari textbox pada form
Image	Gambar pada textbox
MaxLength	Maksimum Karakter pada textbox
Multiline	Jika bernilai true, teks pada textbox bisa lebih dari satu baris
PasswordChar	Karakter yang akan ditampilkan sebagai password
ReadOnly	Jika bernilai true, teks tidak dapat diubah
RightToLeft	Posisi teks pada textbox
Size	Ukuran textbox
Text	Teks atau tulisan pada textbox
TextAlign	Posisi dari teks pada textbox
Visible	Jika bernilai true, textbox kelihatan



2.3.2.5. Button

Button adalah control yang berfungsi sebagai tombol pada *form*. Gunanya untuk menjalankan, menghentikan ataupun menginterupsi suatu proses.

Berikut ini beberapa property dari button:

Tabel 2.8 *Properti Button*

Properti	Keterangan
Name	Nama dari button
BackColor	Warna dari button
BackgroundImage	Gambar latar belakang pada button
Cursor	Bentuk kursor waktu berada pada button
Enabled	Jika bernilai true, button bisa diklik
FlatStyle	Bentuk button
Font	Jenis teks dari button
ForeColor	Warna teks; pacia button
Image	Gambar pada button
ImageAlign	Posisi gambar pada button
Location	Posisi dari button pada form
RightToLeft	Posisi teks pada button
Size	Ukuran button
Text	Teks atau tulisan pada button
TextAlign	Posisi teks pada button
Visible	Jika bernilai true, button kelihatan

2.3.2.6. GroupBox

GroupBox memiliki fungsi untuk mengelompokkan kontrol-kontrol yang ada pada sebuah *form*. Selain itu, group box dapat membantu mengatur tampilan *form* dan kontrol-kontrol pada *form*. Saat group box dipindahkan, maka control yang berada di dalam group box tersebut akan ikut pindah.

Berikut ini beberapa property dari group box:

Tabel 2.9 *Properti GroupBox*

Properti	Keterangan
Name	Nama dari <i>group box</i>
BackColor	Warna dari <i>group box</i>

Lanjutan **Tabel 2.9** *Properti GroupBox*

Properti	Keterangan
BackgroundImage	Gambar latar belakang pada group box
Cursor	Bentuk kursor waktu berada pada group box
Enabled	Jika bernilai true, group box bisa diklik
FlatStyle	Bentuk group box
Font	Jenis teks dari group box
ForeColor	Warna teks pada group box
GridSize	Ukuran tinggi dan lebar dari group box
Location	Posisi group box pada form
RightToLeft	Posisi teks pada group box
Size	Ukuran tinggi dan lebar group box dalam pixel

2.3.2.7. CheckBox

CheckBox dapat berfungsi untuk menampilkan daftar pilihan. Pada checkbox kita bisa memilih lebih dari satu pilihan.

Berikut ini property-property dari checkbox:

Tabel 2.10 *Properti Checkbox*

Properti	Keterangan
Name	Nama checkbox
BackColor	Warna checkbox
BackgroundImage	Gambar latar belakang pada checkbox
CheckAlign	Posisi teks dari checkbox
Checked	Jika bernilai true, checkbox dicek
Enabled	Jika bernilai true, checkbox bisa diklik
FlatStyle	Bentuk tampilan dari checkbox
Font	Jenis teks dari checkbox
ForeColor	Warna teks pada checkbox
Image	Gambar pada checkbox
ImageAlign	Posisi gambar pada checkbox
Location	Posisi checkbox pada <i>form</i>
RightToLeft	Posisi teks pada checkbox
Size	Ukuran checkbox

Lanjutan **Tabel 2.10** *Properti Checkbox*

Properti	Keterangan
Text	Teks atau tulisan pada checkbox
TextAlign	Posisi dari teks pada checkbox
Visible	Jika bernilai true, checkbox kelihatan

2.3.2.8. RadioButton

RadioButton mempunyai fungsi hamper sama dengan checkbox, yaitu menampilkan daftar pilihan. Hanya saja pada radiobutton kita hanya bisa memilih satu pilihan pada satu kelompok radiobutton. Contoh sederhana sebagai penggunaannya adalah pada *survey*.

Berikut ini beberapa property dari radiobutton:

Tabel 2.11 *Properti Radiobutton*

Properti	Keterangan
Name	Nama radiobutton
BackColor	Warna radiobutton
BackgroundImage	Gambar latar belakang padaradiobutton
CheckAlign	Posisi teks dari radiobutton
Checked	Jika bernilai true, radiobutton dicek
Cursor	Bentuk kursor waktu berada pada radiobutton
Enabled	Jika bernilai true, radiobutton bisa diklik
FlatStyle	Bentuk tampilan radiobutton
Font	Jenis teks pada radiobutton
ForeColor	Warna teks pada radiobutton
Image	gambar pada radiobutton
ImageAlign	Posisi gambar pada <i>radiobutton</i>
Location	Posisi radiobutton pada <i>form</i>
RightToLeft	Posisi teks pada radiobutton
Size	Ukuran radiobutton
Text	Teks atau tulisan pada radiobutton
TextAlign	Posisi dari teks pada radiobutton
Visible	Jika bernilai true, radiobutton kelihatan



2.3.2.9. ListBox

ListBox memiliki fungsi untuk menampilkan beberapa daftar pilihan. Kita tinggal memilih pilihan yang ada pada listbox.

2.3.2.10. ComboBox

ComboBox memiliki fungsi untuk menampilkan beberapa daftar pilihan dan kita hanya bisa memilih salah satu pilihan saja.

2.3.2.11. Timer

Komponen *timer* mewakili pengatur waktu yang dipicu oleh *event* dengan interval tertentu. Pada masing-masing interval, objek timer akan menimbulkan event tick (detak). Anda dapat menulis kode yang mengatur event tick secara regular. Interval timer dapat diatur dengan cara memasukkan nilai pada property interval. Nilainya berupa angka yang mewakili interval dalam milidetik.

Berikut ini beberapa property dari *timer*:

Tabel 2.14 *Properti Timer*

Properti	Keterangan
Name	Nama timer
Enabled	Jika bernilai true, timer akan mulai berjalan
Interval	Menunjukkan seberapa lama <i>delay</i> antardetakan dari <i>timer</i> (dalam satuan mili detik)

2.3.2.12. DateTimePicker

Control *DateTimePicker* akan menampilkan combobox yang berisi tanggal yang bisa dipilih. Ketika pemakai menekan panah, control akan menampilkan karakter drop-down seperti control *MonthCalender* dimana pemakai dapat memilih tanggal. Control *DateTimePicker* juga dapat digunakan untuk menampilkan waktu dengan cara mengatur property *Format* dengan *Time*.

2.3.2.13. TabControl

Tabcontrol adalah control yang menampilkan halaman dengan tabulasi. Dengan *Tabcontrol* kita bisa menghemat *file* karena beberapa tampilan tidak harus



disimpan dalam beda *file* vb, tapi tetap ditampilkan dalam satu *file* vb hanya pada tab yang berbeda.

2.3.2.14. PictureBox

PictureBox adalah kotak yang digunakan untuk menampilkan gambar.

2.3.2.15. MessageBox

MessageBox adalah kotak untuk menampilkan pesan kepada pemakai program aplikasi. *Messagebox* dapat digunakan sebagai alat untuk menyampaikan permintaan persetujuan, peringatan, dan lain-lain.

2.3.3. Penjelasan Visual Basic .NET

2.3.3.1. Sejarah Singkat Visual Basic .NET

Menurut Hidayatullah (2014:3), Pada zaman dahulu ada sebuah bahasa pemrograman yang diberi nama *Basic (Beginner's All-purpose Symbolic Instruction Code)*. Sesuai dengan namanya, *Basic* ditujukan sebagai bahasa yang paling sederhana bagi mereka yang tidak terlalu familiar dengan dunia pemrograman.

Pada tahun 1991 Microsoft mengeluarkan *Visual Basic*, pengembangan dari *Basic* yang berubah dari sisi pembuatan antarmukanya. *Visual Basic* sampai sekarang masih menjadi salah satu bahasa pemrograman terpopuler di dunia.

Pada akhir tahun 1999, Teknologi *.NET* diumumkan. Microsoft memosisikan teknologi tersebut sebagai *platform* untuk membangun *XML Web service* memungkinkan aplikasi tipe apapun dapat berjalan pada sistem komputer dengan tipe manapun dan dapat mengambil data yang tersimpan pada server dengan tipe apapun melalui internet.

Berikut ini perkembangan *Visual Basic .NET*:

- a. *Visual Basic .NET* 2002 (VB 7.0)
- b. *Visual Basic .NET* 2003 (VB 7.1)
- c. *Visual Basic* 2005 (VB 8.0)



- d. *Visual Basic* 2008 (VB 9.0)
- e. *Visual Basic* 2010 (VB 10.0)
- f. *Visual Basic* 2012 (VB 11.0)
- g. *Visual Basic* 2013

2.3.3.2. Pengertian Visual Basic .NET

Hidayatullah (2014:5) menyatakan, “Visual Basic .NET adalah Visual Basic yang direkayasa kembali untuk digunakan pada *platform* .NET sehingga aplikasi yang dibuat menggunakan Visual Basic .NET dapat berjalan pada system komputer apa pun, dan dapat mengambil data dari *server* dengan tipe apapun asalkan terinstal .NET Framework”.

2.3.3.3. Kelebihan Visual Basic .NET

Hidayatullah (2014:7), ada beberapa kelebihan VB .NET antara lain:

1. Sederhana dan mudah dipahami
Seperti pada VB, bahasa yang digunakan pada VB .NET sangat sederhana sehingga lebih mudah dipahami bagi mereka yang masih awam terhadap dunia pemrograman.
2. Mendukung GUI
VB .NET bisa membuat *software* dengan antarmuka gratis yang lebih *user friendly*.
3. Menyederhanakan deployment
VB .NET mengatasi masalah *deployment* dari aplikasi berbasis Windows yaitu DLL Hell dan registrasi COM (*Component Object Model*). Selain itu tersedia wizard yang memudahkan dalam pembuatan file setup.
4. Menyederhanakan pengembangan perangkat lunak
Ketika terjadi kesalahan penulisan kode dari sisi sintaks (bahasa), maka VB .NET langsung menuliskan kesalahannya pada bagian *Message Windows* sehingga *Programmer* dapat memperbaiki kode dengan lebih cepat.



5. Mendukung penuh OOP

Memiliki fitur bahasa pemrograman berorientasi objek seperti *inheritance* (pewarisan), *encapsulation* (pembungkusan), dan *polymorphism* (banyak bentuk).

6. Mempermudah pengembangan aplikasi berbasis web

Disediakan desainer form Web. Selain itu disediakan layanan Web XML sehingga memungkinkan suatu aplikasi “berkomunikasi” dengan aplikasi lainnya dari berbagai *platform* menggunakan protocol internet terbuka.

7. Migrasi ke VB .NET dapat dilakukan dengan mudah.

Jika anda sudah mengembangkan aplikasi di VB, Maka konversi ke VB .NET dapat anda jelaskan dengan mudah.

8. Banyak digunakan oleh *programmer-programmer* di seluruh dunia. Salah satu keuntungannya adalah jika kita memiliki masalah/pertanyaan, maka kita bisa tanyakan kepada *programmer-programmer* lain di seluruh dunia melalui forum-forum di Internet.

2.3.4. PHPMyadmin

Nugroho (2012:71) menyatakan, ”PhpAMyadmin adalah tools MySQL Client berlisensi *Freeware*, anda dapat mengunduhnya secara gratis pada situs resmi nya di www.phpmyadmin.net. phpMyAdmin harus dijalankan di sisi server web (misalnya: *Apache web server*) dan pada komputer harus tersedia PHP, karna berbasis web.

2.3.5. Pengertian XAMPP

Nugroho (2012:1) menyatakan, “XAMPP adalah paket program web lengkap yang dapat anda pakai untuk belajar pemrograman web, khususnya PHP dan MySQL.