



Kementerian Pendidikan Dan Kebudayaan
Republik Indonesia
2013



TEKNIK PEMOGRAMAN

Semester

2

Untuk SMK / MAK Kelas X

Penulis : **WELDAN KUSUF**
Editor Materi : **DJOKO SUGIONO**
Editor Bahasa :
Ilustrasi Sampul :
Desain & Ilustrasi Buku : **PPPPTK BOE MALANG**
Hak Cipta © 2013, Kementerian Pendidikan & Kebudayaan

MILIK NEGARA
TIDAK DIPERDAGANGKAN

Semua hak cipta dilindungi undang-undang.

Dilarang memperbanyak (merekproduksi), mendistribusikan, atau memindahkan sebagian atau seluruh isi buku teks dalam bentuk apapun atau dengan cara apapun, termasuk fotokopi, rekaman, atau melalui metode (media) elektronik atau mekanis lainnya, tanpa izin tertulis dari penerbit, kecuali dalam kasus lain, seperti diwujudkan dalam kutipan singkat atau tinjauan penulisan ilmiah dan penggunaan non-komersial tertentu lainnya diizinkan oleh perundangan hak cipta. Penggunaan untuk komersial harus mendapat izin tertulis dari Penerbit.

Hak publikasi dan penerbitan dari seluruh isi buku teks dipegang oleh Kementerian Pendidikan & Kebudayaan.

Untuk permohonan izin dapat ditujukan kepada Direktorat Pembinaan Sekolah Menengah Kejuruan, melalui alamat berikut ini:

Pusat Pengembangan & Pemberdayaan Pendidik & Tenaga Kependidikan Bidang Otomotif & Elektronika:

Jl. Teluk Mandar, Arjosari Tromol Pos 5, Malang 65102, Telp. (0341) 491239, (0341) 495849, Fax. (0341) 491342, Surel: vedcmalang@vedcmalang.or.id, Laman: www.vedcmalang.com

DISKLAIMER (*DISCLAIMER*)

Penerbit tidak menjamin kebenaran dan keakuratan isi/informasi yang tertulis di dalam buku tek ini. Kebenaran dan keakuratan isi/informasi merupakan tanggung jawab dan wewenang dari penulis.

Penerbit tidak bertanggung jawab dan tidak melayani terhadap semua komentar apapun yang ada didalam buku teks ini. Setiap komentar yang tercantum untuk tujuan perbaikan isi adalah tanggung jawab dari masing-masing penulis.

Setiap kutipan yang ada di dalam buku teks akan dicantumkan sumbernya dan penerbit tidak bertanggung jawab terhadap isi dari kutipan tersebut. Kebenaran keakuratan isi kutipan tetap menjadi tanggung jawab dan hak diberikan pada penulis dan pemilik asli. Penulis bertanggung jawab penuh terhadap setiap perawatan (perbaikan) dalam menyusun informasi dan bahan dalam buku teks ini.

Penerbit tidak bertanggung jawab atas kerugian, kerusakan atau ketidaknyamanan yang disebabkan sebagai akibat dari ketidakjelasan, ketidaktepatan atau kesalahan didalam menyusun makna kalimat didalam buku teks ini.

Kewenangan Penerbit hanya sebatas memindahkan atau menerbitkan mempublikasi, mencetak, memegang dan memproses data sesuai dengan undang-undang yang berkaitan dengan perlindungan data.

Katalog Dalam Terbitan (KDT)

Teknik Elektronika, Edisi Pertama 2013

Kementerian Pendidikan & Kebudayaan

Direktorat Jenderal Peningkatan Mutu Pendidik & Tenaga Kependidikan, th. 2013: Jakarta

KATA PENGANTAR

Puji syukur kami panjatkan kepada Tuhan yang Maha Esa atas tersusunnya buku teks ini, dengan harapan dapat digunakan sebagai buku teks untuk siswa Sekolah Menengah Kejuruan (SMK) Bidang Studi Keahlian dan Rekayasa, Teknik Elektronika.

Penerapan kurikulum 2013 mengacu pada paradigma belajar kurikulum abad 21 menyebabkan terjadinya perubahan, yakni dari pengajaran (*teaching*) menjadi BELAJAR (*learning*), dari pembelajaran yang berpusat kepada guru (*teachers-centered*) menjadi pembelajaran yang berpusat kepada peserta didik (*student-centered*), dari pembelajaran pasif (*pasive learning*) ke cara belajar peserta didik aktif (*active learning-CBSA*) atau *Student Active Learning-SAL*.

Buku teks "Teknik Pemrograman" ini disusun berdasarkan tuntutan paradigma pengajaran dan pembelajaran kurikulum 2013 diselaraskan berdasarkan pendekatan model pembelajaran yang sesuai dengan kebutuhan belajar kurikulum abad 21, yaitu pendekatan model pembelajaran berbasis peningkatan keterampilan proses sains.

Penyajian buku teks untuk Mata Pelajaran "Teknik Pemrograman " ini disusun dengan tujuan agar supaya peserta didik dapat melakukan proses pencarian pengetahuan berkenaan dengan materi pelajaran melalui berbagai aktivitas proses sains sebagaimana dilakukan oleh para ilmuwan dalam melakukan eksperimen ilmiah (penerapan *scientific*), dengan demikian peserta didik diarahkan untuk menemukan sendiri berbagai fakta, membangun konsep, dan nilai-nilai baru secara mandiri.

Kementerian Pendidikan dan Kebudayaan, Direktorat Pembinaan Sekolah Menengah Kejuruan, dan Direktorat Jenderal Peningkatan Mutu Pendidik dan Tenaga Kependidikan menyampaikan terima kasih, sekaligus saran kritik demi kesempurnaan buku teks ini dan penghargaan kepada semua pihak yang telah berperan serta dalam membantu terselesaikannya buku teks siswa untuk Mata Pelajaran Teknik Pemrograman kelas X/Semester 1 Sekolah Menengah Kejuruan (SMK).

Jakarta, 12 Desember 2013

Menteri Pendidikan dan Kebudayaan

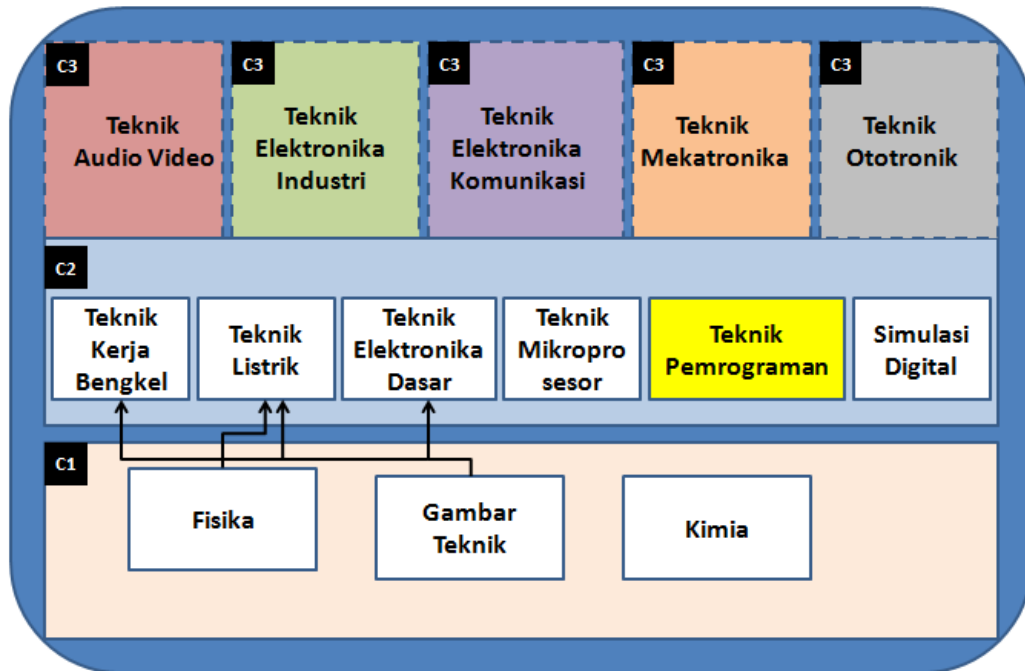
Prof. Dr. Mohammad Nuh, DEA

DAFTAR ISI

DISKLAIMER (<i>DISCLAIMER</i>)	iii
KATA PENGANTAR.....	iv
DAFTAR ISI.....	v
PETA KEDUDUKAN MODUL.....	vii
PENDAHULUAN	1
A. DESKRIPSI MATERI PEMBELAJARAN	1
B. PRASYARAT	2
C. PETUNJUK PENGGUNAAN	2
D. TUJUAN AKHIR	2
E. KOMPTENSI	3
E. CEK KEMAMPUAN AWAL	4
KEGIATAN BELAJAR 3	5
Mendesain program visual grafis GUI (<i>Graphical User Interface</i>) sebagai tampilan interface dan komunikasi.....	5
KEGIATAN 1 : Mengelola Graphical User Interface(GUI) Visual Basic 2010	6
1.1 Membuat Obyek Grafik.....	6
1.2 Menggambar Persegi Panjang.....	12
1.3 Menggambar Elips	16
1.4 Menggambar Lingkaran.....	18
1.5 Menggambar Teks	19
1.6 Menggambar Poligon atau Segi Banyak	23
1.7 Menggambar Segitiga.....	24
1.8 Menggambar Trapesium	25
1.9 Menggambar Pie	26
1.10 Menggambar dan Memberi Warna Pada Bentuk.....	27
KEGIATAN 2 : Aplikasi Grafis dengan menggunakan GUI Visual Basic 2010.....	33
2.2 Membuat Grafik Gelombang Sinusoida	44
2.3 Membuat Chart Grafik	55

KEGIATAN 3 : Membuat GUI Animasi Obyek Bergerak Pada VB 2010	64
3.1 Menggerakkan suatu obyek.....	64
3.2 Membuat Animasi Menggunakan Timer	67
3.3 Animasi Bergerak Lampu Lalu Lintas.....	68
3.4 Visualisasi Progress Bar	78
3.5 Vertikal Progress Bar	81
KEGIATAN 4 : Format data, waktu dan tanggal pada Visual Basic 2010.....	88
4.1 Menampilkan waktu dan tanggal Menggunakan Format Standar	88
4.2 Format Tanggal dan Waktu menggunakan Format dari Pengguna	89
4.3 Konsep Manipulasi Data String Pada Visual Basic 2010.....	91
4.4 Fungsi mid, right, left, trim, Ucase, Lcase dan Chr	92
KEGIATAN BELAJAR 4	96
Menerapkan Program Aplikasi Dengan Bahasa Pemrograman <i>Visual Basic</i> Untuk Keperluan Input/Output Pada Port USB/Serial Pada Komputer.....	96
KEGIATAN 1 : Akses Data String Komunikasi Serial Pada Visual Basic 2010.....	98
1.1 Mengakses Port Serial Pada Visual Basic 2010	100
1.2 Program Aplikasi Serial Tx-Rx PC to PC.....	105
KEGIATAN 2 : Komunikasi Serial Komputer ke Mikrokontroler	117
2.1 Minimum Sistem Arduino	118
2.2 Aplikasi Komunikasi Serial Mikrokontroler-Komputer dengan Visual Basic 2010.....	120
KEGIATAN 3 : Aplikasi Visual Kontrol Input – Output VB 2010.....	132
3.1 Komunikasi Komputer-Mikrokontroler Fungsi Kontrol LED.....	132
3.1.1 Rancangan Elektronik.....	133
3.1.2 Software Program Mikrokontroler	134
3.1.2 Desain Program Visual Basic.....	136
3.2 Visualisasi Grafik ADC (Analog to Digital Converter)	146
Daftar Pustaka.....	149

PETA KEDUDUKAN MODUL



PENDAHULUAN

A. DESKRIPSI MATERI PEMBELAJARAN

Microsoft Visual Basic (sering disingkat sebagai VB saja) merupakan sebuah bahasa pemrograman yang menawarkan *Integrated Development Environment* (IDE) visual untuk membuat program perangkat lunak berbasis sistem operasi Microsoft Windows dengan menggunakan model pemrograman (COM).

Visual Basic merupakan turunan bahasa pemrograman BASIC dan menawarkan pengembangan perangkat lunak komputer berbasis grafik dengan cepat. Meskipun awalnya program visual basic lebih banyak digunakan untuk program aplikasi bisnis, akan tetapi perkembangannya visual basic banyak juga digambarkan untuk mendukung aplikasi-aplikasi dunia teknik.

Pada buku siswa ini dibahas tentang bahasa pemrograman Visual Basic 2010, dimana pembahasan buku mencakup dari pengenalan visual basic, dasar pemrograman hingga aplikasi input-output yang berhubungan langsung dengan port serial mikrokontroler. Dengan mempelajari teknik pemrograman visual basic diharapkan siswa dapat memahami dasar teknik pemrograman dan mengaplikasikan Visual Basic sebagai fungsi aplikasi kontrol maupun sebagai antarmuka antara manusia dengan mesin atau biasa disebut HMI (*Human Machine Interface*).

B. PRASYARAT

Materi Teknik Pemrograman 2 memberikan bekal awal dalam mengembangkan kemampuan siswa untuk membuat program aplikasi yang berhubungan dengan kompetensi teknik pemrograman pada jurusan teknik elektronika. Materi ini disampaikan pada kelas X semester 2.

C. PETUNJUK PENGGUNAAN

Buku ini disusun dengan memberikan penjelasan tentang konsep dasar pemrograman Visual Basic 2010 dengan beberapa contoh aplikasi sederhana yang berkaitan dengan dunia teknik pada umumnya dan elektronika pada khususnya. Untuk memungkinkan siswa belajar sendiri secara tuntas, maka perlu diketahui bahwa isi buku ini pada setiap kegiatan belajar umumnya terdiri atas, uraian materi, contoh-contoh aplikasi, tugas dan tes formatif serta lembar kerja, sehingga diharapkan siswa dapat belajar mandiri (*individual learning*) dan *mastery learning* (belajar tuntas) dapat tercapai.

D. TUJUAN AKHIR

Tujuan akhir yang hendak dicapai adalah agar siswa mampu:

- Mengenal pemrograman grafis dan animasi sederhana dengan menggunakan visual basic.
- Mampu membuat program aplikasi sederhana dengan bahasa *Visual Basic* untuk keperluan input/output pada komputer/laptop
- Memahami saluran output (port output) pada Port serial dan USB sebagai saluran perantara kontrol
- Memahami saluran input (port input) pada komputer/laptop menggunakan Port Serial dan USB sebagai pengolah data sinyal input.
- Mampu membuat program aplikasi visualisasi sederhana input-output menggunakan visual basic terhubung mikrokontroler.

E. KOMPETENSI

KOMPETENSI INTI (KI-3)	KOMPETENSI INTI (KI-4)
<p>Kompetensi Dasar (KD):</p> <ol style="list-style-type: none"> Memahami bahasa pemrograman Visual Basic 	<p>Kompetensi Dasar (KD):</p> <ol style="list-style-type: none"> Menerapkan program aplikasi dengan bahasa pemrograman <i>Visual Basic</i> untuk keperluan input/output pada port USB/serial pada Komputer
<p>Indikator:</p> <ul style="list-style-type: none"> Mengenal Lingkungan Pengembangan Terintegrasi (<i>Integrated Development Enviroment-IDE</i>) bahasa <i>Visual Basic</i>. Menjelaskan bagaimana memulai menggunakan Lingkungan Pengembangan Terintegrasi (<i>Integrated Development Enviroment-IDE</i>) bahasa <i>Visual Basic</i>. Memahami konsep dasar bahasa <i>Visual Basic</i>. Memahami program aplikasi sederhana dengan bahasa <i>Visual Basic</i> 	<p>Indikator:</p> <ol style="list-style-type: none"> Memahami saluran input (port input) pada computer/laptop dari Port Serial dan USB Memahami saluran output (port output) pada Port serial dan USB. Membuat program aplikasi sederhana dengan bahasa <i>Visual Basic</i> untuk keperluan input/output pada komputer/laptop.

E. CEK KEMAMPUAN AWAL

1. Apa yang anda ketahui tentang GUI (*Graphical User Interface*)!
2. Sebutkan obyek grafis apa saja yang dapat digambar oleh visual basic dan sebutkan *source code* perintah-nya!
3. Apa yang anda ketahui tentang animasi sederhana pada visual basic?
4. Sebutkan dua cara sederhana untuk menggambar grafik pada visual basic!
5. Apa yang anda ketahui tentang manipulasi data string pada visual basic? Sebutkan kegunaan memanipulasi data tersebut!
6. Apa yang anda ketahui tentang akses komunikasi serial?
7. Sebutkan perbedaan komunikasi serial dengan komunikasi via USB?
8. Apa yang anda ketahui tentang komunikasi serial PC ke PC, PC ke mikrokontroler dan mikrokontroler ke mikrokontroler?
9. Mengapa masih dibutuhkan mikrokontroler untuk melakukan fungsi kontrol input/output ketika kita menggunakan visual basic? Jelaskan!
10. Apa yang dimaksud dengan istilah berikut :
 - Tx - Baudrate
 - Rx - Parity

KEGIATAN BELAJAR 3

Mendesain program visual grafis GUI (*Graphical User Interface*) sebagai tampilan interface dan komunikasi

A. Tujuan Pembelajaran

1. Mengetahui pemrograman berbasis grafis GUI.
2. Menjelaskan bagaimana pemrograman menggunakan GUI.
3. Membuat program aplikasi berbasis GUI.
4. Memahami konsep manipulasi data untuk kepentingan dasar komunikasi .

B. Uraian Materi

1. Mengelola grafis GUI dengan memahami konsep menggambar garis, segitiga, lingkaran, polygon, grafik dan chart .
2. Membuat GUI animasi obyek bergerak.
3. Format data, waktu dan tanggal.
4. Konsep manipulasi data untuk kepentingan dasar komunikasi .

C. Alokasi Waktu

12 jam pelajaran

D. Metode Pembelajaran

Teori dan Praktek

E. Media pembelajaran

- PC/Notebook
- Windows 7
- Visual Basic 2010

Mendesain program visual grafis GUI (*Graphical User Interface*) sebagai tampilan interface dan komunikasi

KEGIATAN 1 : Mengelola Graphical User Interface(GUI) Visual Basic 2010

Meskipun penggunaan grafis pada Visual Basic versi sebelumnya terlihat lebih mudah karena memiliki banyak *tool* gambar, pada Visual Basic 2010 jauh lebih fleksibel dalam pembuatan grafis. Misalnya dalam Visual Basic 6 peralatan gambar termasuk di dalam toolbox dimana programmer hanya membutuhkan menyeret kontrol bentuk ke dalam form untuk membuat lingkaran, kotak, elips, persegi panjang dan lainnya. Namun, suatu penyederhanaan juga memiliki kekurangan, Anda tidak memiliki banyak pilihan dalam membuat gambar.

Sejak Visual Basic berevolusi menjadi bahasa OOP sepenuhnya dalam kerangka VB.net, kontrol bentuk tidak lagi tersedia. Sekarang programmer perlu menuliskan kode untuk membuat banyak bentuk dan gambar. Meskipun terlihat lebih rumit tetapi programmer dapat menulis kode dengan baik untuk menciptakan semua jenis grafis. Anda bahkan dapat membuat kontrol Anda sendiri.

Visual Basic 2010 menawarkan berbagai kemampuan yang memungkinkan programmer untuk menulis kode yang dapat menggambarkan berbagai bentuk dan font. Dalam pelajaran ini Anda akan belajar bagaimana menuliskan kode untuk menggambar garis dan bentuk pada antarmuka Visual Basic.

1.1 Membuat Obyek Grafik

Sebelum Anda mulai menggambar pada form, Anda perlu membuat obyek grafis pada Visual Basic 2010. Obyek grafik dibuat menggunakan sebuah metode `CreateGraphics()`. Anda dapat membuat obyek grafis yang digambar ke form atau kontrol. Sebagai contoh, jika Anda ingin menggambar pada form, Anda dapat menggunakan pernyataan ;

```
Dim Grafik As Graphics = me.CreateGraphics
```

Selalu gunakan Dim untuk mendefinisikan obyek. Gunakan kode “**me**” jika Anda ingin menggambar pada Form. Jika Anda ingin menggambar pada picture box Anda dapat menggunakan perintah berikut :

```
Dim Grafik As Graphics = PictureBox1.CreateGraphics
```

Anda juga dapat menggambar pada text box dengan perintah :

```
Dim Grafik As Graphics = TextBox1.CreateGraphics
```

Obyek grafis yang dibuat, tidak tampak tergambar pada layar sampai Anda memanggil *method* dari obyek grafis tersebut. Sebelumnya Anda perlu membuat obyek Pena sebagai peralatan gambar.

a. Membuat Pena

Pena dapat dibuat menggunakan kode :

```
Pena = New Pen (Brushes.DarkMagenta, 10)
```

Dimana Pena adalah sebuah variabel yang akan digunakan sebagai variabel pengatur dari detail garis atau gambar yang akan kita buat. Pernyataan pertama dari obyek pen mendefinisikan warna dari garis yang digambar dan pernyataan kedua mendefinisikan tebal dari garis tersebut. Anda juga dapat menggambar pena dengan pernyataan berikut :

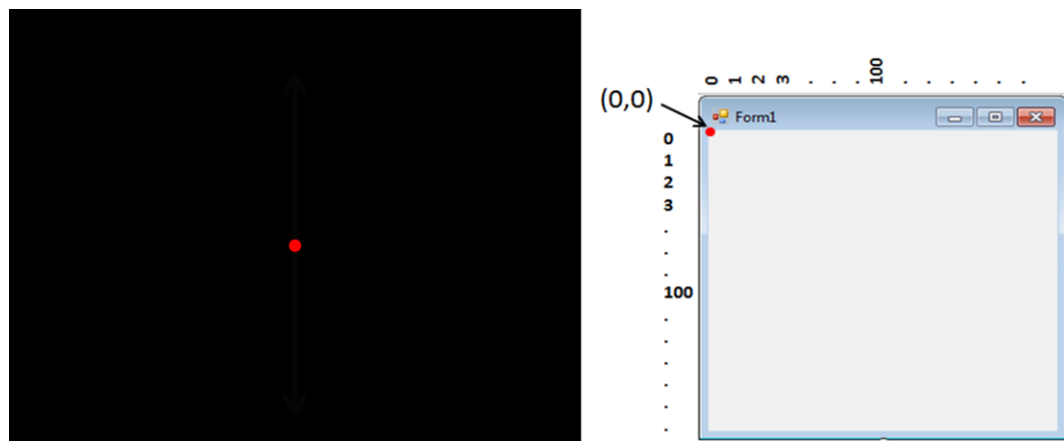
```
Dim Pena As Pen
Pena = New Pen (Drawing.Color.Blue, 5)
```

Dimana pernyataan pertama mendefinisikan warna (disini adalah biru, Anda dapat merubah warna sesuai dengan keinginan) dan pernyataan kedua adalah tebal dari garis yang digambar.

Setelah membuat Grafis dan obyek pena, sekarang Anda telah siap untuk menggambar grafis pada layar.

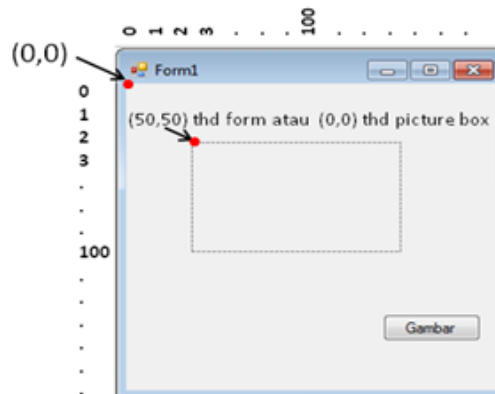
b. Pixel koordinat form aplikasi

Ketika kita belajar membuat desain grafis pada visual basic, dasar dari semua proses desain adalah mengetahui koordinat kerja dari form yang akan kita buat. Sama halnya ketika kita telah belajar di bangku sekolah dasar mengenai diagram kartesius (x,y), maka pada form design visual basic juga memiliki koordinat (0,0). Perbedaan yang mendasar adalah pada diagram kartesius, diagram terdiri atas empat kuadran dan titik (0,0) terletak tepat pada titik tengah kuadran. Sedangkan pada form design visual basic terdiri atas satu kuadran dan titik (0,0) terletak di ujung pojok kiri atas. Untuk lebih jelasnya perhatikan titik warna merah yang terletak pada gambar di bawah ini.



Gambar 1.1 Koordinat pixel Visual Basic

Pada gambar di atas, gambar sebelah kiri menunjukkan titik nol dari diagram kartesius sedangkan gambar sebelah kanan menunjukkan titik nol dari form design visual basic 2010. Pada form design visual basic nilai x dan y berdasarkan titik pixel dimana obyek tersebut akan di gambarkan. Sebagai contoh tambahkan komponen PictureBox pada form design dan atur *properties location* dari komponen tersebut menjadi (50,50). Maka akan muncul tampilan seperti gambar dibawah ini.



Gambar 1.2 Koordinat titik awal

Pada gambar di atas titik merah pada ujung kiri atas *PictureBox* mempunyai dua arti. Yang pertama lokasi titik tersebut terletak pada koordinat (50,50) terhadap form aplikasi, yang kedua titik merah tersebut memiliki titik koordinat (0,0) terhadap *PictureBox* aplikasi.

Dengan kondisi tersebut kita memiliki dua kemungkinan untuk menggambar, yang pertama kita dapat menggambar (contoh: menggambar suatu garis lurus) dengan menggunakan referensi terhadap titik koordinat koordinat form dan yang kedua referensi terhadap titik koordinat *PictureBox*. Untuk lebih jelasnya perhatikan contoh pada sub c menggambar baris di bawah ini.

c. Menggambar garis

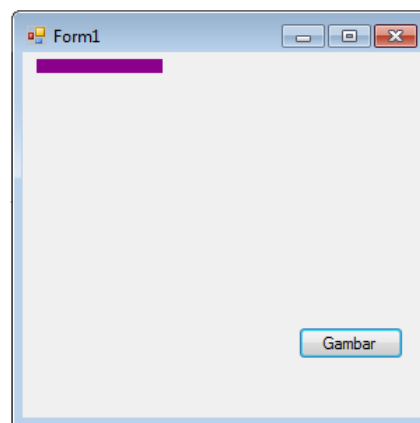
Pada sesi ini, Anda akan belajar untuk menggambar garis lurus pada form. Pertama bukalah Visual Basic 2010. Pada halaman awal seretlah sebuah tombol ke dalam form. Double klik pada tombol tersebut dan tulislah pernyataan berikut :

```
Private Sub Button1_Click (ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
Dim Grafik As Graphics = me.CreateGraphics
Dim Pena As Pen
Pena = New Pen (Brushes.DarkMagenta, 10)
Grafik.DrawLine (Pen, 10, 10, 100, 10)
End sub
```

Pernyataan kedua dari program diatas adalah membuat obyek grafis, pernyataan pada baris kedua dan ketiga membuat obyek Pena. Pernyataan baris kelima menggambar garis pada form menggunakan *method DrawLine*. Pernyataan pertama dari baris kelima adalah menggunakan obyek Pena yang Anda buat sebelumnya, pernyataan kolom kedua dan ketiga mendefinisikan koordinat untuk titik awal, pernyataan keempat dan terakhir merupakan akhir koordinat dari garis yang dibuat. Sintaks secara umum dari *Drawline* adalah sebagai berikut :

```
Object.DrawLine(Pen, x1, y1, x2, y2)
```

Tampilan dari garis yang dibuat akan tampak seperti gambar berikut:

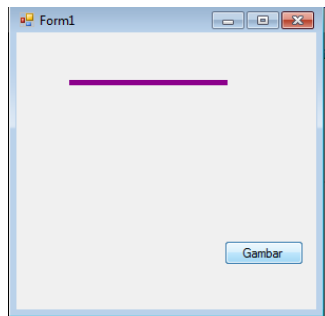


Gambar 1.3 Menggambar garis lurus pada form

Pada contoh di atas tampak bahwa program menggambar garis lurus dari koordinat (10,10) menuju ke koordinat (100,10) pada *form design*. Sekarang kita akan lihat perbedaan saat menggambar garis lurus pada koordinat (50,50) menuju koordinat (200,50) pada *form design* dengan menggunakan *picturebox*.

Script menggambar pada form design (program 1) :

```
Dim Grafik As Graphics = me.CreateGraphics
Dim Pena As Pen
Pena = New Pen (Brushes.DarkMagenta, 10)
Grafik.DrawLine (Pen, 50, 50, 200, 50)
End sub
```



Untuk menggambar pada *PictureBox*, tambahkan terlebih dahulu komponen *PictureBox* kemudian atur *properties location* menjadi 50;50. Kemudian tuliskan script program berikut ini (program 2)

```
Dim Grafik As Graphics = PictureBox1.CreateGraphics
Dim Pena As Pen
Pena = New Pen(Brushes.DarkMagenta, 10)
Grafik.DrawLine(Pena, 0, 0, 150, 0)
```

Perhatikan penggalan program diatas, hasil eksekusi program antara menggunakan *form design* dan *PictureBox* akan menghasilkan tampilan yang sama. Akan tetapi ada perbedaan dari penulisan *script* koordinat program.

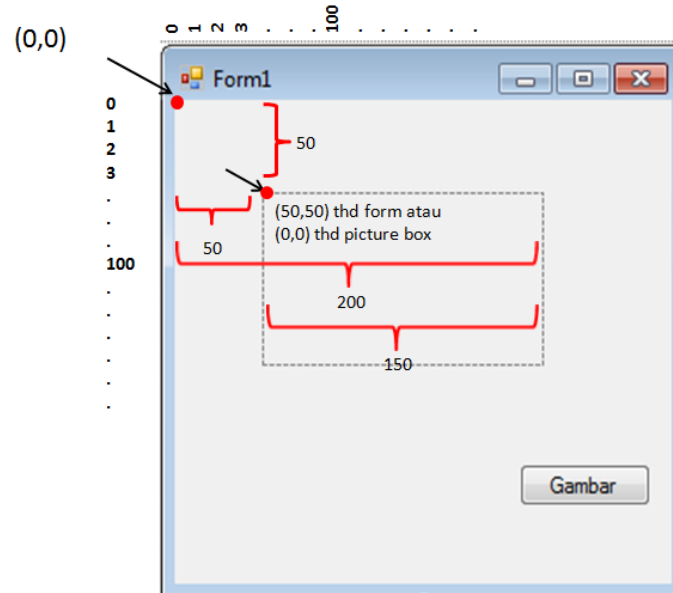
Pada program pertama tertulis kode program berikut

```
Grafik.DrawLine (Pen, 50, 50, 200, 50)
```

Sedangkan pada program kedua tertulis koordinat yang berbeda

```
Grafik.DrawLine(Pena, 0, 0, 150, 0)
```

Alasan mengapa koordinat program 2 di tuliskan koordinat (0,0) menuju koordinat (150,0) adalah karena kita telah mengatur posisi *PictureBox* sebelumnya dengan mengubah *properties location* (50;50). Sehingga koordinat titik awal dari *PictureBox* (0,0) identik dengan koordinat *form design* (50,50). Sedangkan titik akhir koordinat *PictureBox* (150,0) identik dengan koordinat akhir form design (200,50). Untuk lebih jelasnya perhatikan ilustrasi gambar berikut



Gambar 1.4 Ilustrasi koordinat dan ukuran PictureBox

1.2 Menggambar Persegi Panjang

Untuk menggambar sebuah persegi panjang pada layar Visual Basic 2010 dapat dilakukan dengan dua cara :

- a. Pertama untuk menggambar sebuah persegi panjang gunakanlah metode *DrawRectangle* dengan menentukan koordinat sudut kiri, lebar dan tinggi. Anda juga perlu membuat Grafik dan obyek Pena untuk membuat gambar yang sebenarnya. Sintaks untuk menggambar persegi panjang dengan metode *DrawRectangle* adalah sebagai berikut :

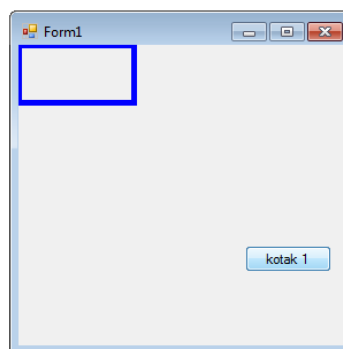
```
Persegipanjang.DrawRectangle(Pena, X, Y, width, height)
```

Dimana Persegipanjang adalah nama variabel dari obyek Graphics dan Pena adalah nama variabel dari obyek Pen yang Anda buat. Anda dapat menggunakan berbagai nama variabel. X, Y adalah

koordinat sudut kiri dari persegi panjang sedangkan width dan height adalah ukuran panjang dan lebar dari persegi panjang.

Contoh kode persegi panjang :

```
Dim Pena As Pen
Pena = New Pen(Drawing.Color.Blue, 5)
Dim Grafik As Graphics = me.CreateGraphics
Grafik.DrawRectangle(Pena, 0, 0, 100, 50)
```



Gambar 1.5 Menggambar persegi panjang

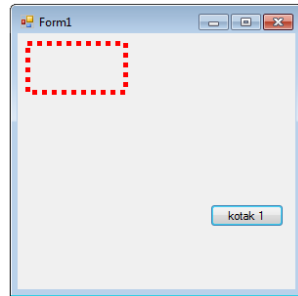
Eksekusi program potongan program tersebut akan menghasilkan gambar seperti tampak pada gambar di atas. Bentuk persegi panjang pada gambar tersebut berupa garis, akan tetapi garis tersebut dapat juga diubah menjadi titik-titik, strip maupun dot tergantung dengan jenis Pen yang kita gunakan. Sebagai contoh, sintaks untuk menggambar dengan bentuk dot adalah :

```
Pena.DashStyle = Drawing.Drawing2D.DashStyle.Dot
```

Dimana pernyataan terakhir Dot menentukan garis tertentu dengan nilai DashStyle, garis yang membentuk titik-titik. Kode berikut ini menggambarkan sebuah persegi panjang dengan garis putus-putus berwarna merah.

```
Dim Pena As Pen
Pena = NewPen(Drawing.Color.Red, 5)
Dim Grafik As Graphics = me.CreateGraphics
```

```
Pena.DashStyle = Drawing.Drawing2D.DashStyle.Dot
Grafik.DrawRectangle(Pena, 10, 10, 100, 50)
End Sub
```



Beberapa nilai garis DashStyle dari obyek Pen tampak pada daftar dibawah ini :

Nilai DashStyle	Macam Garis
Dot	Garis terdiri dari titik-titik
Dash	Garis terdiri dari strip
DashDot	Garis terdiri dari strip dan titik-titik
DashDotDot	Garis terdiri dari strip dan double titik-titik
Solid	Garis tegas
Custom	

- b. Cara kedua untuk menggambar persegi panjang adalah dengan cara manual menggambar dari garis lurus dari titik satu ke titik lain. Sebagai contoh kita akan menggambar suatu persegi panjang seperti contoh di atas yang terdiri dari empat titik yaitu titik A(10,10), titik B(100,10), titik C (100,50) dan titik D(10,50).

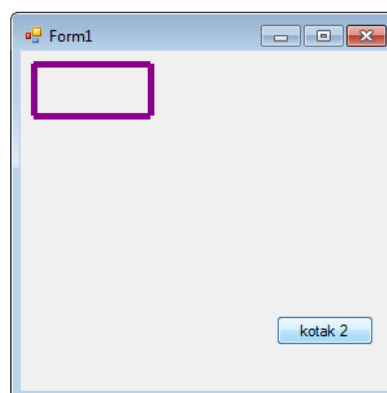


Gambar 1.6 Titik koordinat persegi panjang

Untuk menggambar persegi panjang di atas maka di gunakan perintah gambar garis lurus `Drawline`. Perintah tersebut kita eksekusi empat kali untuk menggambarkan ke empat sisi persegi panjang yaitu sisi A-B, sisi B-C, sisi C-D dan sisi D-A. Berikut potongan program untuk membuat persegi panjang dengan perintah `Drawline`.

```
Pena = New Pen(Brushes.DarkMagenta, 5)
Grafik.DrawLine(Pena, 10, 10, 100, 10)
Grafik.DrawLine(Pena, 100, 10, 100, 50)
Grafik.DrawLine(Pena, 100, 50, 10, 50)
Grafik.DrawLine(Pena, 10, 50, 10, 10)
```

Baris ke dua script program di atas menggambar sisi A-B, baris ketiga menggambar sisi B-C, baris keempat menggambar sisi C-D, baris kelima menggambar sisi D-A.



Gambar 1.7 Hasil gambar menggunakan `DrawLine`

1.3 Menggambar Elips

Untuk menggambar bidang lingkaran atau elips, terlebih dahulu perlu dipahami prinsip dari menggambar sebuah elips di Visual Basic 2010. Hampir semua struktur dasar dari menggambar pada visual basic 2010 adalah persegi panjang dengan menggunakan garis `line`, pengecualian untuk elips. Elips adalah suatu bentuk oval yang dibatasi oleh persegi panjang seperti gambar dibawah ini :



Gambar 1.8 Persinggungan elips terhadap persegi panjang

Oleh karena itu Anda perlu membuat sebuah persegi panjang sebelum Anda membuat sebuah elips. Persegi panjang memberikan batas untuk elips. Pada latihan selanjutnya Anda dapat menggambar elips dengan metode ***DrawEllipse*** tanpa membuat persegi panjang terlebih dahulu.

Metode pertama Anda membuat sebuah persegi panjang dengan variabel `Persegipanjang` dan obyek pena dengan nama variabel `Pena`, kemudian Anda dapat menggambar sebuah elips menggunakan pernyataan berikut

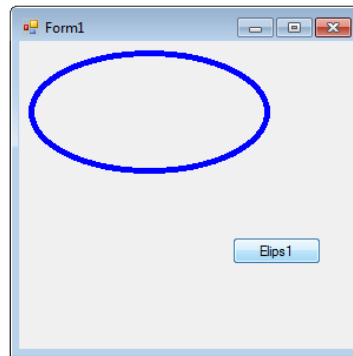
```
Grafik.DrawEllipse(Pena, Persegipanjang)
```

Sama halnya dengan perintah sebelumnya seperti menggambar garis lurus maupun persegi panjang, terlebih dahulu kita siapkan objek form yang akan kita gambar dan kita siapkan juga variabel pena sebagai variabel pengatur alat gambar yang akan kita gunakan

```
Dim Grafik As Graphics = Me.CreateGraphics
Dim Pena As Pen
```


Kemudian tambahkan komponen button dan ubah property text menjadi “Elips1”. Jika sudah maka double klik komponen button tersebut dan tuliskan kode program berikut :

```
Pena = New Pen(Drawing.Color.Blue, 5)
Dim Persegipanjang As New Rectangle
Persegipanjang.X = 10
Persegipanjang.Y = 10
Persegipanjang.Width = 200
Persegipanjang.Height = 100
Grafik.DrawEllipse(Pena, Persegipanjang)
```



Gambar 1.9 Menggambar bentuk elips

Gambar di atas menunjukkan hasil eksekusi aplikasi program yang ditulis sebelumnya. Baris pertama dari program di atas bertujuan untuk menyiapkan pengaturan alat gambar yang akan digunakan, sedangkan baris kedua mendeklarasikan variabel sebagai tempat penyimpanan pengaturan komponen `Rectangle` yang akan digunakan. Pengaturan `Rectangle` meliputi titik koordinat x awal, titik koordinat y awal, panjang dan lebar dari persegi panjang.

Baris ketiga menunjukkan koordinat x awal = 10, baris keempat menunjukkan koordinat y awal = 10, baris kelima menunjukkan panjang persegi panjang = 200 dan baris keenam menampilkan lebar dari persegi panjang = 100. Baris terakhir merupakan perintah menggambar elips berdasar frame rectangle yang telah dibuat.

Metode kedua untuk membuat bentuk elips adalah dengan menggunakan metode *DrawEllipse* tanpa menggambar persegi panjang terlebih dahulu. Tentu saja Anda harus membuat obyek grafis dan pena sebelumnya. Sintak dari program ini adalah :

```
Gambar.DrawEllipse(Pena, X, Y, Width, Height)
```

Dimana (X,Y) adalah koordinat dari sudut kiri batas persegi panjang, width adalah lebar dari elips dan height adalah tinggi dari elips.

Kode program untuk menggambar elips adalah :

```
Dim Pena As Pen
Pena = New Pen(Drawing.Color.Blue, 5)
Dim Grafik As Graphics = me.CreateGraphics
Grafik.DrawEllipse(Pena, 10, 10, 200, 100)
```

1.4 Menggambar Lingkaran

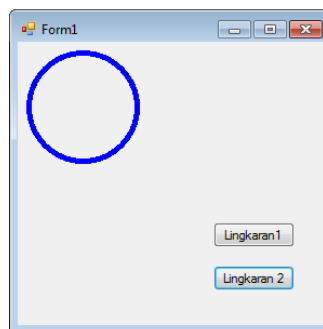
Setelah anda belajar membuat bentuk elips, maka untuk menggambar lingkaran sangat mudah. Anda dapat menggunakan metode yang sama seperti prosedur sebelumnya tetapi untuk tinggi dan lebar menggunakan nilai yang sama.

Contoh pertama menggambar lingkaran dengan menggunakan batas persegi panjang.

```
Dim Pena As Pen
Pena = New Pen(Drawing.Color.Blue, 5)
Dim Grafik As Graphics = me.CreateGraphics
Dim Persegipanjang As New Rectangle
Persegipanjang.X = 10
Persegipanjang.Y = 10
Persegipanjang.Width = 100
Persegipanjang.Height = 100
Grafik.DrawEllipse(Pena, Persegipanjang)
```

Contoh kedua menggambar lingkaran tanpa menggambar persegi panjang terlebih dahulu :

```
Dim Pena As Pen
Pena = New Pen(Drawing.Color.Blue, 5)
Dim Grafik As Graphics = me.CreateGraphics
Grafik.DrawEllipse(Pena, 10, 10, 100, 100)
```



Gambar 1.10 Menggambar lingkaran

1.5 Menggambar Teks

Kita telah mempelajari bagaimana menggambar persegi panjang, elips dan lingkaran di Visual Basic 2010, sekarang anda akan mempelajari bagaimana menggambar teks di layar. Selain menggunakan perintah Print Anda dapat menggambar teks pada layar.

Untuk menggambar teks dapat menggunakan metode *DrawString*, sintaknya adalah :

```
Grafik.DrawString(Teks, Font, Brush, X, Y)
```

Dimana Gambar adalah obyek Grafis, Teks adalah tulisan atau teks yang ingin Anda tampilkan di layar, Font adalah obyek font yang Anda buat, Brush adalah *brush* yang Anda buat, sedangkan X dan Y adalah koordinat awal (sudut kiri) dari Teks.

Anda dapat membuat obyek Font di Visual Basic 2010 dengan menggunakan pernyataan :

```
Font = NewSystem.Drawing.Font("Verdana".20)
```

Dimana pernyataan pertama dari font tersebut adalah tipe font dan pernyataan kedua adalah ukuran dari font. Anda dapat menambahkan pernyataan ketiga seperti jenis font, tulisan tebal, miring, garis bawah seperti contoh :

```
Font = New System.Drawing.Font("Verdana", 20,
    FontStyle.Bold)
Font = New System.Drawing.Font("Verdana", 20,
    FontStyle.Underline)
Font = New System.Drawing.Font("Verdana", 20,
    FontStyle.Italic)
Font = New System.Drawing.Font("Verdana", 20,
    FontStyle.Regular)
```

Untuk membuat Obyek Brush Anda dapat menggunakan pernyataan :

```
myBrush = New Drawing.SolidBrush(Color.BrushColor)
```

Tujuh warna paling sering digunakan dalam aplikasi adalah, AliceBlue, AquaMarine, Beige, DarkMagenta, DarkOliveGreen, SkyBlue dan banyak lagi. Anda tidak perlu mengingat nama dari warna-warna tersebut, karena IntelliSense (program deteksi otomatis) akan menunjukkan anda warna-warna yang ingin Anda gunakan setelah Anda mengetik titik setelah kata warna (color).

Dibawah ini adalah contoh program untuk menggambar teks :

```
Dim Grafik As Graphics = me.CreateGraphics
Dim myFont As Font
Dim myBrush As Brush
myBrush = New Drawing .SolidBrush(Color.DarkOrchid)
```

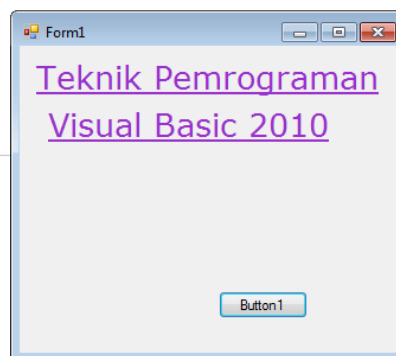
Teknik Pemrograman

```

myFont = New System.Drawing.Font("Verdana", 20,
FontStyle.Underline)
Grafik.DrawString("Teknik Pemrograman", myFont,
myBrush, 10, 10)
Grafik.DrawString("Visual Basic 2010", myFont,
myBrush, 20, 10)

```

Jalankan program diatas dan akan terlihat tampilan hasil eksekusi seperti gambar berikut :



Gambar 1.11 Hasil menggambar teks

Anda juga dapat memodifikasi prosedur diatas jika tidak ingin membuat obyek Font dan Brush. Anda dapat menggunakan font dari obyek yang telah tersedia seperti Form dan System Colors. Gantilah baris terakhir dari program contoh diatas dengan program berikut (Anda perlu menghapus baris yang membuat obyek Brush dan Font)

```

Grafik.DrawString ("VisualStudio2010",me.Font,
SystemDrawing.Brush.DarkOrchid, 10, 10)

```

Anda juga dapat menambahkan sebuah InputBox dimana pengguna akan memasukkan pesan yang kemudian menampilkan pesan tersebut di layar.

Contoh :

```

Dim Grafik As Graphics = Me.CreateGraphics
Dim myFont As Font
Dim myBrush As Brush

```

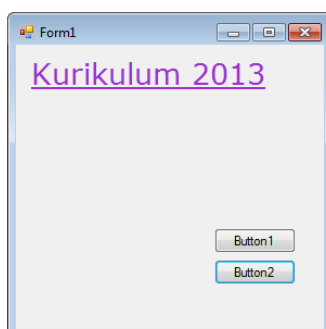
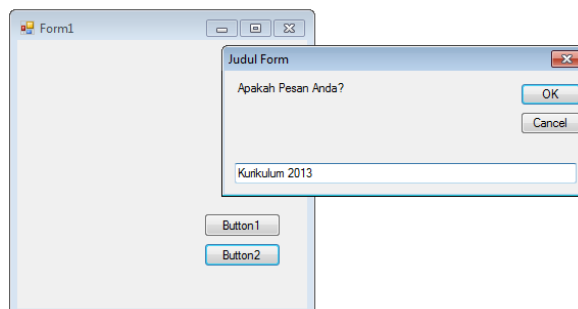
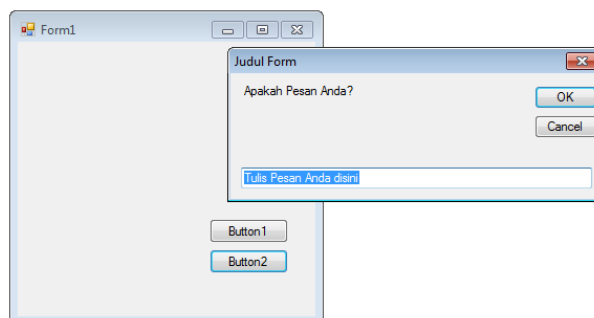
Dim userMsg As String

```
userMsg = InputBox("Apakah Pesan Anda?", "Judul Form",
    "Tulis Pesan Anda disini", 100, 200)
```

```
myBrush = New Drawing.SolidBrush(Color.DarkOrchid)
```

```
myFont = New System.Drawing.Font("Verdana", 20,
    FontStyle.Underline)
```

```
Grafik.DrawString(userMsg, myFont, myBrush, 10, 10)
```



1.6 Menggambar Poligon atau Segi Banyak

Poligon adalah sebuah bidang tertutup yang dibatasi oleh tiga atau lebih garis lurus. Untuk menggambar poligon pada layar, kita perlu menentukan koordinat dari semua titik (dikenal sebagai simpul) yang bergabung membentuk poligon atau segi banyak. Dengan menggunakan sintak dari `polygon` kita dapat membangun suatu segi enam, segi lima atau segi banyak hanya dengan menentukan koordinat ujung simpul bangun tersebut dengan perintah ini.

Sintak untuk menentukan titik dari poligon dengan simpul A1, A2, A3, A4 :

```
Dim A1 As New Point(X1, Y1)
Dim A2 As New Point(X2, Y2)
Dim A3 As New Point(X3, Y3)
Dim A4 As New Point(X4, Y4)
-
-
Dim An As New Point(Xn, Yn)
```

Setelah mendeklarasikan poin, kita perlu menentukan sebuah struktur poin yang mengelompokkan semua titik secara bersamaan menggunakan sintak :

```
Dim Titik As Point() = (A1, A2, A3, ...An)
```

Selanjutnya kelompok koordinat atau point tersebut akan digambarkan sesuai dengan titik koordinat yang telah dimasukan dengan menggunakan perintah *DrawPolygon*. Berikut sintak *DrawPolygon* untuk menggambar grafis segi banyak :

```
Dim Grafik As Graphis = me.CreateGraphis
Grafik.DrawPolygon(Pena, Titik)
```

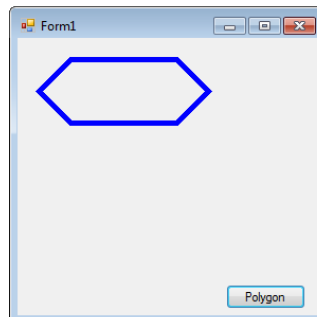
Selanjutnya tidak lupa untuk menggunakan Pena sebagai obyek Pen yang dibuat untuk menggambar :

```
Pena = NewPen(Drawing.ColorBlue, 5)
```

Berikut contoh tampilan hasil eksekusi dari penggunaan *DrawPolygon* untuk menggambar segi enam beserta script program aplikasi yang akan menggambar segi enam tersebut.

```
Dim A1 As New Point(50, 20)
Dim A2 As New Point(150, 20)
Dim A3 As New Point(180, 50)
Dim A4 As New Point(150, 80)
Dim A5 As New Point(50, 80)
Dim A6 As New Point(20, 50)

Dim Titik As Point() = {A1, A2, A3, A4, A5, A6}
Pena = New Pen(Drawing.Color.Blue, 5)
Grafik.DrawPolygon(Pena, Titik)
End Sub
```



Gambar 1.12 Menggambar polygon

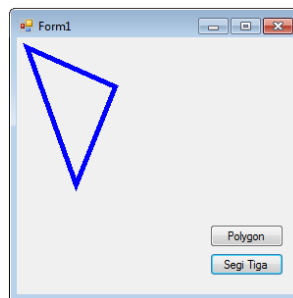
1.7 Menggambar Segitiga

Perintah *DrawPolygon* dapat di modifikasi untuk menggambar suatu segitiga. Untuk mendapatkan gambar segitiga, maka titik point poligon yang kita perlukan hanyalah tiga simpul, berikut ini adalah contoh kode programnya :


```
Dim Pena As Pen
Dim A As New Point(10, 10)
Dim B As New Point(100, 50)
Dim C As New Point(60, 150)
Dim Titik As Point() = {A, B, C}
```

```
Pena = New Pen(Drawing.Color.Blue, 5)
Grafik.DrawPolygon(Pena, Titik)
```

Jalankan program tersebut, dan Anda akan melihat tampilan seperti gambar dibawah ini :



Gambar 1.13 Hasil gambar segitiga

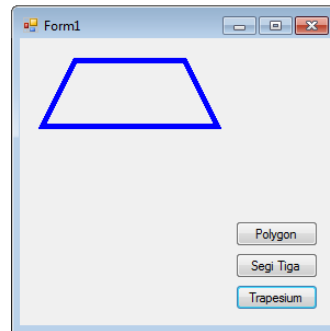
1.8 Menggambar Trapesium

Trapesium merupakan sebuah poligon yang terdiri dari empat sisi, maka Anda perlu untuk mendefinisikan empat simpul dengan kode program :

```
Dim Pena As Pen
Dim A1 As New Point(50, 20)
Dim A2 As New Point(150, 20)
Dim A3 As New Point(180, 80)
Dim A4 As New Point(20, 80)

Dim Titik As Point() = {A1, A2, A3, A4}

Pena = New Pen(Drawing.Color.Blue, 5)
Grafik.DrawPolygon(Pena, Titik)
```



Gambar 1.13 Hasil gambar trapesium

1.9 Menggambar Pie

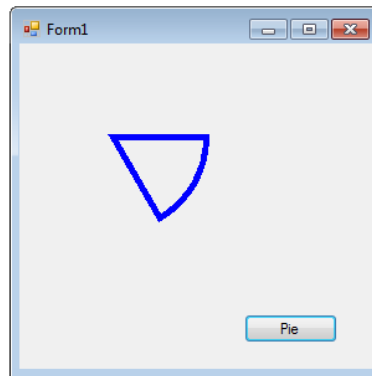
Untuk menggambar Phi, Anda dapat menggunakan *method* DrawPie dari obyek grafik. Seperti biasanya, Anda perlu menggambar obyek Graphics dan Pen terlebih dahulu. Sintak untuk menggambar Pie adalah :

```
Grafik.DrawPie (Pena, X, Y, width, height, StartAngle, SweepAngle)
```

Dimana X dan Y adalah koordinat dari persegi panjang, StartAngle dan SweepAngle adalah ukuran dalam satuan derajat, SweepAngle dapat berupa nilai negatif maupun positif. Jika nilai positif maka akan searah dengan jarum jam sementara jika nilai negatif berarti berlawanan dengan arah jarum jam.

Dibawah ini adalah contoh kode menggambar Phi yang dimulai dari 0 derajat dan searah jarum jam 60 derajat :

```
Dim Pena As Pen
Pena = New Pen(Drawing.Color.Blue, 5)
Dim Grafik As Graphics = me.CreateGraphics
Grafik.DrawPie(Pena, 0, 0, 150, 150, 0, 60)
```



Gambar 1.20 Hasil gambar pie chart

1.10 Menggambar dan Memberi Warna Pada Bentuk

Pada pembelajaran sebelumnya, siswa telah belajar bagaimana menggambar berbagai macam bentuk seperti persegi panjang, lingkaran, elips, poligon dan pie. Kali ini siswa akan belajar bagaimana memberi warna pada bentuk-bentuk tersebut. Ada tiga metode yang dapat digunakan untuk mengisi sebuah bentuk yaitu, `FillRectangle`, `FillEllipse`, `FillPolygon` dan `FillPie`. Untuk memberikan warna pada bentuk, kita perlu membuat obyek `Brush` dengan menggunakan sintak :

```
myBrush = New SolidBrush(Color. myColor)
```

Dimana `myColor` berupa warna seperti `red`, `blue`, `yellow`, dan sebagainya. kita tidak perlu khawatir tentang nama-nama dari warna tersebut karena Visual Basic akan menampilkan secara otomatis warna-warna tersebut setelah penulisan tanda “titik” pada kata kunci `Color`.

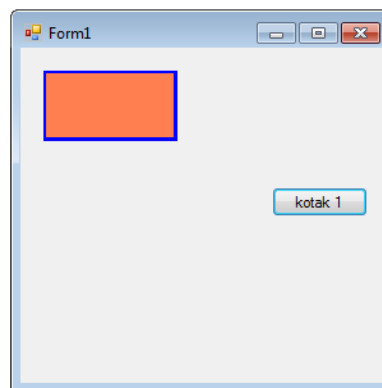
a. Menggambar dan Mengisi Bentuk Persegi Panjang

Sintak untuk mengisi persegi panjang dengan warna adalah :

```
Grafik.FillRectangle (myBrush, x, y, width, height)
```

Keseluruhan kode program diatas sebagai berikut :

```
Dim Pena As Pen
Dim myBrush As Brush
Dim Grafik As Graphics = me.CreateGraphics
Pena = New Pen(Drawing.Color.Blue, 5)
myBrush = New SolidBrush(Color.Coral)
Grafik.DrawRectangle(Pena, 20, 20, 150, 150)
Grafik.FillRectangle(myBrush, 20, 20, 150, 150)
```



Gambar 1.21 Mengisi warna pada persegi panjang

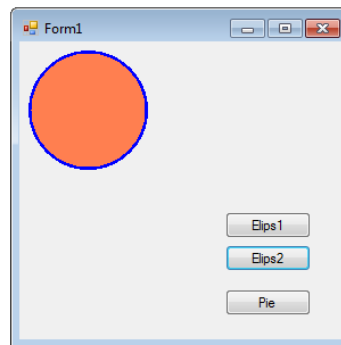
b. Menggambar dan Mengisi Bentuk Elips

Sintak untuk mengisi bentuk elipas dengan warna adalah :

```
Grafik.FillEllipse(myBrush, x, y, width, height)
```

Keseluruhan kode program tersebut adalah :

```
Dim Pena As Pen
Dim myBrush As Brush
Dim Grafik As Graphics = me.CreateGraphics
Pena = New Pen(Drawing.Color.Blue, 5)
myBrush = New SolidBrush(Color.Coral)
Grafik.DrawEllipse(Pena, 10, 10, 100, 100)
Grafik.FillEllipse(myBrush, 10, 10, 100, 100)
```



Gambar 1.22 Mengisi warna pada lingkaran

c. Menggambar dan Mengisi Bentuk Poligon

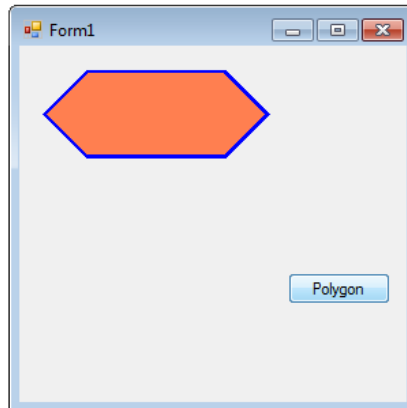
Sintak untuk mengisi bentuk poligon dengan warna adalah :

```
Grafik.FillPolygon(myBrush, Titik)
```

Keseluruhan kode program tersebut adalah :

```
Dim Pena As Pen
Dim myBrush As Brush
Dim A As New Point(10, 10)
Dim B As New Point(100, 50)
Dim C As New Point(120, 150)
Dim D As New Point(60, 200)
Dim Titik As Point() = (A, B, C, D)
Pena = New Pen(Drawing.Color.Blue, 5)
myBrush =New SolidBrush(Color.Coral)
Dim Grafik As Graphics = me.CreateGraphics
Grafik.DrawPolygon(Pena, Titik)
Grafik.FillPolygon(myBrush, Titik)
```

Jalankan program tersebut dan hasilnya akan tampak seperti gambar berikut :



Gambar 1.23 Mengisi warna pada polygon

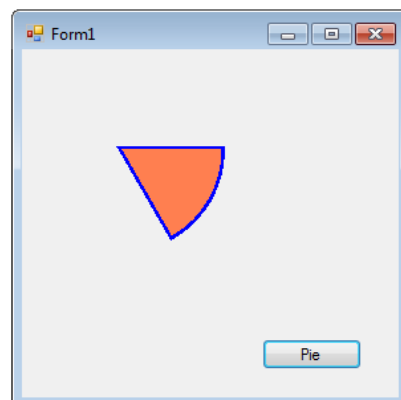
d. Menggambar dan Mengisi Pie

Sintak untuk mengisi bentuk phi dengan warna adalah :

```
Grafik.FillPie(myBrush, X, Y, width, height,
StartAngle, SweepAngle)
```

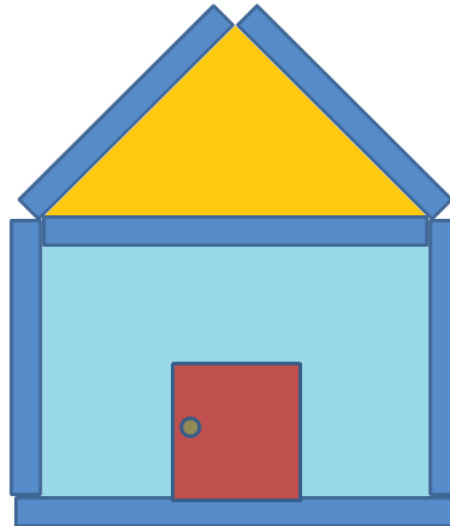
Keseluruhan kode program diatas adalah :

```
Dim Pena As Pen
Dim myBrush As Brush
Pena = New Pen(Drawing.Color.Blue, 5)
myBrush = new SolidBrush(Color.Coral)
Dim Grafik As Graphics = me.CreateGraphics
Grafik.DrawPie(Pena, 30, 40, 150, 150, 0, 60)
Grafik.FillPie(myBrush, 30, 40, 150, 150, 0, 60)
```



TUGAS

Buatlah project baru bernama “Gambar_Rumah”, desainlah suatu aplikasi yang akan menggambarkan suatu rumah dengan konsep tampilan akhir seperti berikut :



Gambar di atas harus dibangun dengan menggunakan script terprogram. Gunakanlah metode drawline, drawrectangle, drawlips dan fillshape untuk membangun tampilan rumah di atas!

TES FORMATIF

1. Jelaskan perbedaan menggambar suatu persegi panjang dengan menggunakan metode drawline dengan menggunakan metode drawrectangle!
2. Apakah syarat untuk membuat bangun segi tiga dengan menggunakan metode poligon segi banyak?

Jawab :

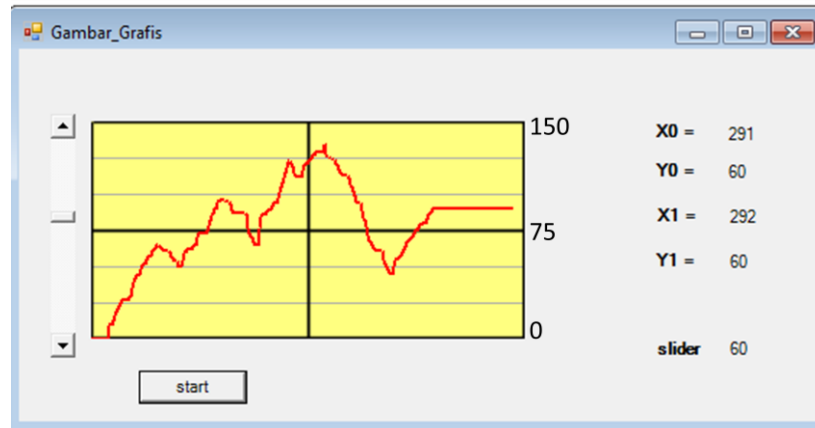
KEGIATAN 2 : Aplikasi Grafis dengan menggunakan GUI Visual Basic 2010

2.1 Menggambar Grafik Dengan Input Slider

Dasar membuat gambar grafis telah kita pelajari, sekarang saatnya kita membuat dasar aplikasi yang menerapkan fungsi gambar grafis dalam dunia teknik. Salah satu program aplikasi yang paling sering dan penting dalam dunia teknik adalah menggambarkan grafik. Gambar grafik tersebut digunakan untuk menampilkan sekumpulan data yang diperoleh oleh suatu pengukuran suatu sensor misalnya.

Untuk menyajikan sekumpulan data menjadi suatu grafik dapat menggunakan dua buah cara dalam visual basic 2010. Yang pertama dan akan kita bahas terlebih dahulu adalah menggunakan fungsi `Drawline`, yaitu sama dengan prinsip menggambar garis dari satu titik ke titik yang lain dengan merubah titik koordinat awal dan koordinat tujuan. Perbedaannya adalah proses perubahan dari satu titik ke titik yang lain dilakukan secara otomatis dengan menggunakan timer. Cara kedua adalah dengan menggunakan fungsi `chart` yang telah disediakan oleh toolbox visual basic 2010.

Baik kita mulai dengan program menggambar grafik yang pertama dengan menggunakan `DrawLine`. Program ini akan menggambarkan sekumpulan data yang di wakili oleh slider `VScrollBar1` dan nilai dari slider tersebut akan di tampilkan ke suatu grafik yang kita gambar secara manual. Berikut hasil dari tampilan grafik tersebut nantinya :



Gambar 2.1 Menggambar grafik dengan menggunakan DrawLine

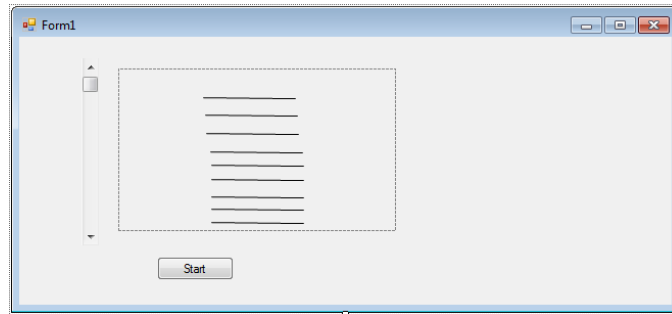
Gambar di atas menunjukkan tampilan grafik sejumlah data yang mewakili nilai dari **VscrollBar1** yang berkisar antara 0 sampai dengan 150. Nilai tersebut digambar pada sebuah komponen **Panel1** dimana nilai selalu *ter-update* secara *real-time* berdasar *interval* waktu dari **Timer**. Tombol “**Start**” berfungsi untuk memulai program dengan cara mengaktifkan *enable* dari properti **Timer**. Pada sebelah kanan terdapat sekumpulan angka dan parameter, yang fungsi ke semuanya itu untuk mengecek titik koordinat titik per titik (X0, Y0, X1 dan Y1). Disamping itu juga terdapat parameter “**slider**” yang berfungsi untuk mengecek nilai dari **VscrollBar1**.

Untuk membuat program aplikasi diatas, ikutilah langkah-langkah berikut :

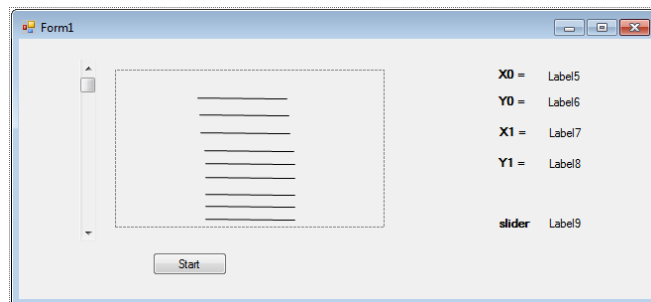
1. Buatlah **Project** baru dengan mengarahkan kursor pada menu File, klik New Project dan buatlah Form baru dengan nama Grafik_Slider.
2. Sesuaikan ukuran form hingga tampak kurang lebih seperti gambar di atas.
3. Klik komponen **VScrollBar1** pada Toolbox dan letakan di form sebagai fungsi perubah nilai data yang hendak digambarkan. Ubah properties dari komponen tersebut menjadi seperti berikut : Maximum = 150, Minimum = 0, Location = 20;45, Size = 20;170.
4. Tambahkan komponen **Button1** sebagai tombol untuk memulai program dan gantilah properties Text komponen tersebut menjadi Start.

Teknik Pemrograman

- Tambahkan komponen **LineShape1, LineShape2, LineShape3, LineShape4, LineShape5, LineShape6, LineShape7, dan LineShape8** serta letakkan semua komponen LineShape tersebut di dalam Panel1 seperti tampak pada gambar berikut.



- Tambahkan komponen **Label1, Label2, Label3, Label4, Label5, Label6, Label7, Label8, Label9, Label10**. Letakkan sebelah kanan dari form, dan ganti properties text hingga tampak seperti gambar di bawah ini.



- Arahkan kursor ke toolbox dan pilihlah komponen Timer1 dengan mendouble klik dua kali. Ubahlah interval properties Timer1 menjadi 100.

Untuk lebih detail dari properties komponen yang digunakan, perhatikan tabel berikut :

Obyek	Properti	Pengaturan
Panel1	Name	Panel1
	BackColor	255; 255; 128
	Location	50;50

VScrollBar1	Name LargeChange Maximum Minimum Location Size	VScrollBar1 1 150 0 20;45 20:170
LineShape1	Name BorderWidth X0 Y0 X1 Y1	LineShape1 2 0 75 300 75
LineShape2	Name BorderWidth X0 X1 Y0 Y1	LineShape2 2 0 300 0 0
LineShape3	Name BorderWidth X0 X1 Y0 Y1	LineShape3 2 0 300 149 149
LineShape4	Name BorderWidth X0 X1 Y0 Y1	LineShape4 2 299 299 0 150
LineShape5	Name BorderWidth X0	LineShape5 2 0

	X1	0
	Y0	0
	Y1	150
LineShape6	Name	LineShape6
	BorderColor	AppWorkspace
	X0	0
	X1	300
	Y0	25
	Y1	25
LineShape7	Name	LineShape7
	BorderColor	AppWorkspace
	X0	0
	X1	300
	Y0	50
	Y1	50
LineShape8	Name	LineShape8
	BorderColor	AppWorkspace
	X0	0
	X1	300
	Y0	100
	Y1	100
LineShape9	Name	LineShape9
	BorderColor	AppWorkspace
	X0	0
	X1	300
	Y0	125
	Y1	125
Button1	Name	Button1
	Text	"Start"
Label1	Name	Label1
	Text	"X0"
Label2	Name	Label2
	Text	"Y0"

Label3	Name Text	Label3 "X1"
Label4	Name Text	Label4 "Y1"
Label5	Name	Label5
Label6	Name	Label6
Label7	Name	Label7
Label8	Name	Label8
Label9	Name	Label9
Label10	Name Text	Label10 "Slider"
Timer1	Enabled Interval	False 100
Form1	Text	"Gambar_Grafis_Slider"

Berikut script program secara keseluruhan :

```

Public Class slider
  Dim l, x0, y0, x1, y1, x2, y2, gain As Integer
  Dim myPen As Pen

  Private Sub Form1_Load(ByVal sender As System.Object,
  ByVal e As System.EventArgs) Handles MyBase.Load
    x0 = 0
    y0 = 150
    x1 = x0
    y1 = y0
  End Sub

  Private Sub Timer1_Tick(ByVal sender As System.Object,
  ByVal e As System.EventArgs) Handles Timer1.Tick

  Dim Grafik As Graphics = Panel1.CreateGraphics
  myPen = New Pen(Drawing.Color.Red, 2)

```

```
x1 = x1 + 1
y1 = VScrollBar1.Value
Grafik.DrawLine(myPen, x0, y0, x1, y1)

Label5.Text = x0
Label6.Text = y0
Label7.Text = x1
Label8.Text = y1
Label9.Text = VScrollBar1.Value

    If x1 >= 300 Then
        Me.Refresh()
        x1 = 0
    End If

    y0 = y1
    x0 = x1
End Sub

Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click
    Timer1.Enabled = True
End Sub

End Class
```

Pembahasan program

Inisialisasi program dimulai dengan menentukan titik awal dimana garis penggambaran dimulai, yaitu di titik koordinat (0,150).

```
x0 = 0
```

```
y0 = 150
```

Seperti kita pelajari sebelumnya, untuk menggambar garis dengan menggunakan metode **DrawLine**, maka kita membutuhkan koordinat awal dan koordinat tujuan. Oleh karena itu dalam pembuatan aplikasi menggambar grafik titik koordinat tujuan x1 digantikan oleh fungsi increment (tambah satu; $i = i+1$), sedangkan koordinat Y1 di tentukan nilai dari VScrollBar1. Akan tetapi untuk inisialisasi awal maka kita dapat menuliskan nilai koordinat (X1,Y1) sama dengan koordinat awal (X0,Y0) artinya hanya menggambar satu titik di titik start awal.

```
x1 = x0
```

```
y1 = y0
```

Selanjutnya kita perlu mengaktifkan timer1 berdasar fungsi tombol tekan Button1. Langkah yang diperlukan adalah double klik komponen Button1 pada jendela design, sehingga kursor kerja akan otomatis berada di antara sub program button1.click dan end sub. Kemudian dituliskan script program yang akan mengaktifkan Timer1 dengan mengeset **True** enable dari fungsi komponen Timer1.

```
Timer1.Enabled = True
```

Program utama dari penggambaran fungsi grafik terletak pada sub program *Timer*. Dimulai dari menyiapkan *Panel1* sebagai tempat untuk menggambar fungsi *Drawline*, kemudian mengatur *pen* yang akan digunakan baik itu jenis garis, warna maupun ketebalannya.

```
Dim Grafik As Graphics = Panel1.CreateGraphics
myPen = New Pen(Drawing.Color.Red, 2)
```


Proses berikutnya adalah membuat fungsi *increment* yang akan digunakan sebagai nilai titik koordinat tujuan X1 dan nilai dari *VScrollBar1* sebagai titik koordinat tujuan Y1.

```
x1 = x1 + 1
```

```
y1 = VScrollBar1.Value
```

Langkah selanjutnya adalah fungsi menggambar garis dari titik awal (X0,Y0) menuju titik tujuan (X1,Y1) dengan menggunakan fungsi *DrawLine*

```
Grafik.DrawLine(myPen, x0, y0, x1, y1)
```

Secara umum dengan menuliskan perintah *DrawLine*, maka penggambaran grafik telah dilakukan. Akan tetapi untuk memudahkan siswa sebagai programmer, maka akan lebih baik jika kita membuat suatu fungsi pengecekan setiap komponen variabel dengan menampilkan pada kelompok label seperti berikut :

```
Label5.Text = x0           → Mengecek variabel X0
Label6.Text = y0           → Mengecek variabel Y0
Label7.Text = x1           → Mengecek variabel X1
Label8.Text = y1           → Mengecek variabel Y1
Label9.Text = VScrollBar1.Value → Mengecek nilai slider
```

Untuk membuat tampilan grafik pada panel lebih menarik, maka kita perlu mengatur dimana jika tampilan data grafik telah mencapai ujung kanan dari panel1, maka tampilan grafik panel1 perlu di refresh dan penggambaran selanjutnya dimulai dari sebelah kanan dari panel lagi. Untuk itu kita harus membuat fungsi syarat IF...THEN, dimana jika posisi titik gambar sudah berada di ujung panel1 (X1=300) maka harus di refresh dan menggambar dimulai dari titik koordinat awal lagi (X0=X1=0).

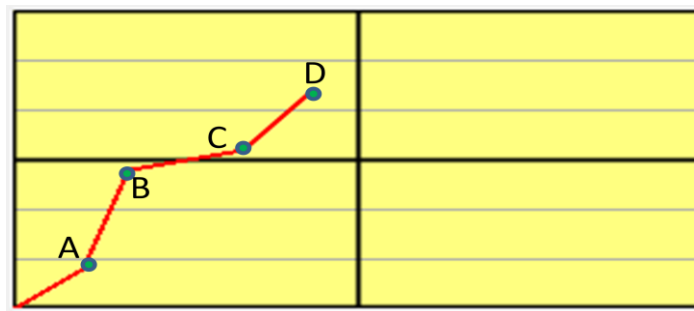
```
If x1 >= 300 Then           → Jika posisi X1 berada di ujung panel
    Me.Refresh()           → Fungsi merefresh Panel1
    x1 = 0                 → Mengembalikan ke titik awal panel
End If
```

Perintah terakhir dan paling penting adalah membuat titik koordinat tujuan menjadi titik koordinat awal. Hal ini penting karena jika kita tidak memperbarui titik koordinat awal, maka fungsi penggambaran grafik akan tidak sesuai dengan yang kita harapkan.

$$y_0 = y_1$$

$$x_0 = x_1$$

Untuk lebih jelasnya perhatikan ilustrasi gambar berikut :



Gambar 2.2 Ilustrasi koordinat grafik titik pertitik

TITIK	KOORDINAT AWAL	KOORDINAT TUJUAN	DrawLine
A	(0,150)	(30,130)	Grafik.DrawLine(myPen, 0, 150, 30, 130)
B	(30,130)	(50,80)	Grafik.DrawLine(myPen, 30, 130, 50, 80)
C	(50,80)	(100,70)	Grafik.DrawLine(myPen, 50, 80, 100, 70)
D	(100,70)	(130,40)	Grafik.DrawLine(myPen, 100, 70, 130, 40)

Dari ilustrasi di atas tampak bahwa koordinat tujuan pada titik A, akan menjadi titik awal pada titik B. Koordinat tujuan pada titik B, akan menjadi titik awal pada titik C, sedangkan koordinat tujuan pada titik C, akan menjadi titik awal pada titik D.

Demikianlah seluruh penjelasan perintah program aplikasi menggambar grafik dimana akan dilakukan terus menerus secara berulang-ulang selama Timer1 masih aktif.

Pada program diatas, pada awal membuat desain tampilan kita melakukan pengaturan setiap komponen dan propertiesnya satu persatu dengan menyesuaikan daftar rincian tabel properties yang telah disebutkan di atas. Cara lain yang dapat kita lakukan adalah dengan mengatur secara otomatis

atau terprogram. Artinya mengatur setiap properties komponen dengan menuliskan script program pada sub program Form Load. Berikut contoh potongan program sub Form Load untuk mengatur properties komponen program aplikasi di atas.

```
Panel1.BackColor = Color.Turquoise
```

```
LineShape1.X1 = 0
```

```
LineShape1.Y1 = 75
```

```
LineShape1.X2 = 300
```

```
LineShape1.Y2 = 75
```

```
LineShape1.BorderWidth = 2
```

```
LineShape2.X1 = 0
```

```
LineShape2.X2 = 300
```

```
LineShape2.Y1 = 0
```

```
LineShape2.Y2 = 0
```

```
LineShape2.BorderWidth = 2
```

```
LineShape3.X1 = 0
```

```
LineShape3.X2 = 300
```

```
LineShape3.Y1 = 149
```

```
LineShape3.Y2 = 149
```

```
LineShape3.BorderWidth = 2
```

```
LineShape4.X1 = 299
```

```
LineShape4.X2 = 299
```

```
LineShape4.Y1 = 0
```

```
LineShape4.Y2 = 150
```

```
LineShape4.BorderWidth = 2
```

```
LineShape5.X1 = 0
```

```
LineShape5.X2 = 0
```

```
LineShape5.Y1 = 0
```

```
LineShape5.Y2 = 150
```

```
LineShape5.BorderWidth = 2
```

```

LineShape6.X1 = 0
LineShape6.X2 = 300
LineShape6.Y1 = 25
LineShape6.Y2 = 25
LineShape6.BorderColor = Color.Yellow
  
```

```

LineShape7.X1 = 0
LineShape7.X2 = 300
LineShape7.Y1 = 50
LineShape7.Y2 = 50
LineShape7.BorderColor = Color.Yellow
  
```

```

LineShape8.X1 = 0
LineShape8.X2 = 300
LineShape8.Y1 = 100
LineShape8.Y2 = 100
LineShape8.BorderColor = Color.Yellow
  
```

```

LineShape9.X1 = 0
LineShape9.X2 = 300
LineShape9.Y1 = 125
LineShape9.Y2 = 125
LineShape9.BorderColor = Color.Yellow
  
```

2.2 Membuat Grafik Gelombang Sinusoida

Fungsi grafik selanjutnya yang akan kita pelajari adalah fungsi grafik gelombang sinusoida. Gelombang sinusoida merupakan gelombang yang dibentuk dari fungsi matematika dan akan membentuk gelombang osilasi secara berulang-ulang. Gelombang ini sering kita jumpai pada aplikasi matematika, fisika, pemrosesan sinyal, teknik elektro dan bidang-bidang lainnya. Bentuk persamaan dasar dari fungsi sinusoida adalah :

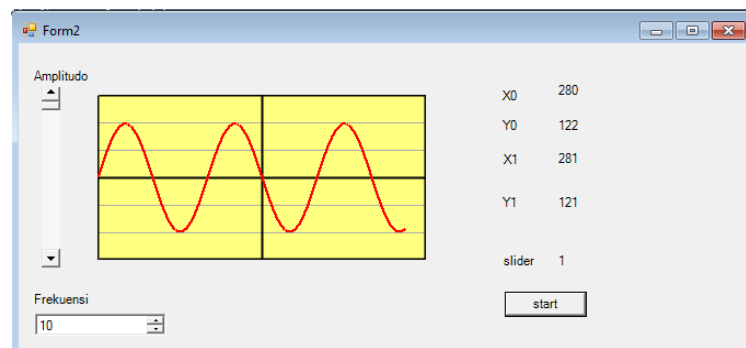
$$Y = A \sin(\omega t)$$

$$Y = A \sin(2\pi f t)$$

Dimana :

- A , amplitudo, adalah puncak deviasi fungsi dari posisi pusatnya
- ω , frekuensi sudut, yang menentukan berapa banyak osilasi terjadi dalam interval waktu tertentu (rad/s)
- t , waktu, nilai waktu dari detik ke detik dimana osilasi terjadi (direpresentasikan pada fungsi x atau horisontal pada grafik maupun diagram kartesian).
- f , frekuensi, yang menentukan berapa banyak osilasi terjadi dalam interval waktu tertentu (Hz)

Metode yang digunakan sama halnya dengan metode grafik sebelumnya, yaitu dengan menggunakan perintah `DrawLine`. Hal yang yang perlu diperhatikan adalah bahwa kita ingin menggambar fungsi sinus tersebut ke dalam suatu media datar pada sebuah panel. Seperti yang kita ketahui bersama bahwa fungsi sinus adalah fungsi dalam satuan derajat. Bagaimana menggambar sudut derajat tersebut hingga menjadi suatu gelombang sinus, maka diperlukan konversi satuan. Konversi satuan yang diperlukan adalah merubah dari satuan derajat menjadi satuan radian. Berikut hasil tampilan akhir hasil program aplikasi menggambar fungsi sinus dengan menggunakan Visual Basic 2010.

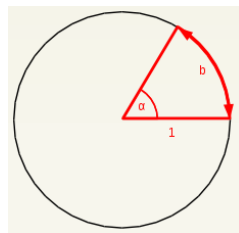


Gambar 2.3 Grafik gelombang sinusoida

Sebelum membuat program aplikasi tersebut alangkah baiknya jika mengetahui secara singkat apa rumus fungsi gelombang sinus dan bagaimana merubah fungsi dalam satuan derajat menjadi radian terlebih dahulu.

Kita semua tahu bahwa degree adalah derajat (bahasa indonesia dari degree), dimana satu putaran penuh didefinisikan sebagai 360 derajat / 360 degree / 360°.

Definisi dari satu radian atau 1 rad adalah besar sudut yang dibentuk oleh dua buah jari-jari lingkaran dengan panjang jari-jari tersebut 1 satuan meter dan membentuk busur sepanjang 1 meter juga. Perhatikan gambar berikut :



Jari-jari lingkaran di atas adalah 1 meter dan busur b yang dibentuk juga 1 meter, maka dengan kondisi seperti itu sudut yang kecil yang dibentuk oleh kedua jari-jari lingkaran itu (α) didefinisikan sebagai sudut dengan besar 1 radian.

Mengubah derajat ke radian

Ingat kembali rumus mencari keliling lingkaran, misalkan keliling 1 lingkaran kita simbolkan dengan K , maka: $K = 2\pi r$

Maka panjang busur 1 lingkaran (keliling lingkaran) dengan panjang jari-jari 1 satuan tersebut adalah $2\pi r = 2 \times \pi \times 1 = 2\pi$

Misalkan 1 radian = x° . Kita tahu bahwa 1 lingkaran mempunyai sudut 360° dengan panjang busurnya (keliling lingkaran) 2π satuan, dan kita tahu bahwa 1 radian (x°) mempunyai panjang busur 1 satuan. Kita akan menggunakan perbandingan untuk mencari nilai x .

$$\frac{\text{Derajat 1 lingkaran}}{\text{Derajat 1 radian}} = \frac{\text{panjang busur 1 lingkaran}}{\text{panjang busur 1 radian}}$$

$$\frac{360^0}{x} = \frac{2\pi}{1}$$

$$x = \frac{360^0}{2\pi}$$

$$x = \frac{180^0}{\pi}$$

Jadi di dapat bahwa besar sudut 1 radian = $\frac{180^0}{\pi}$ derajat

$$1\text{rad} = \frac{180^0}{\pi}$$

$$180^0 = \pi \text{ rad}$$

$$360^0 = 2\pi \text{ rad}$$

Contoh :

$$30^0 \rightarrow 30^0 \times \frac{\pi}{180^0} = \frac{1}{6} \times \pi = 0,524$$

Sehingga 30 derajat = 0,524 radian

Langkah-langkah pembuatan program

Untuk membuat program aplikasi diatas, ikutilah langkah-langkah berikut :

1. Tambahkan *Windows Form* baru dengan mengarahkan kursor pada menu *Project*, klik *Add Windows Form* dan beri nama *Grafik_Sinus*.
2. Sesuaikan ukuran form hingga tampak kurang lebih seperti gambar di atas.
3. Bukalah form design "Grafik_slider" yang telah anda buat, arahkan kursor pada jendela form kemudian tekan tombol **Ctrl+A** dan salin dengan menekan tombol **Ctrl+C**.
4. Kembalilah ke form design "Grafik_Sinus" dan tempelkan salinan seluruh komponen dengan menekan **Ctrl+V**. (Anda juga bisa melakukan secara manual, membuat ulang design dengan menambahkan komponen per-komponen beserta pengaturan propertiesnya seperti halnya yang telah anda lakukan pada program grafik_slider yang telah dibuat sebelumnya).
5. Atur properties *location* komponen **VScrollBar1** menjadi = 20;45.
6. Atur properties *location* komponen **Panel1** menjadi = 50;50.
7. Tambahkan komponen **NumericUpDown1**, letakan dibawah Panel1 dan ubah *properties value* menjadi 10.
8. Sesuaikan posisi lokasi komponen lain hingga kurang lebih sesuai dengan tampilan program seperti gambar di atas.

Untuk lebih detail dari properties komponen yang digunakan, perhatikan tabel berikut :

Obyek	Properti	Pengaturan
Panel1	Name	Panel1
	BackColor	255; 255; 128
	Location	50;50
VScrollBar1	Name	VScrollBar1
	LargeChange	1
	Maximum	50
	Minimum	0
	Location	20;45

	Size	20:170
LineShape1	Name BorderWidth X0 Y0 X1 Y1	LineShape1 2 0 75 300 75
LineShape2	Name BorderWidth X0 X1 Y0 Y1	LineShape2 2 0 300 0 0
LineShape3	Name BorderWidth X0 X1 Y0 Y1	LineShape3 2 0 300 149 149
LineShape4	Name BorderWidth X0 X1 Y0 Y1	LineShape4 2 299 299 0 150
LineShape5	Name BorderWidth X0 X1 Y0 Y1	LineShape5 2 0 0 0 150
LineShape6	Name BorderColor	LineShape6 AppWorkspace

	X0 X1 Y0 Y1	0 300 25 25
LineShape7	Name BorderColor X0 X1 Y0 Y1	LineShape7 AppWorkspace 0 300 50 50
LineShape8	Name BorderColor X0 X1 Y0 Y1	LineShape8 AppWorkspace 0 300 100 100
LineShape9	Name BorderColor X0 X1 Y0 Y1	LineShape9 AppWorkspace 0 300 125 125
Button1	Name Text	Button1 "Start"
Label1	Name Text	Label1 "X0"
Label2	Name Text	Label2 "Y0"
Label3	Name Text	Label3 "X1"
Label4	Name Text	Label4 "Y1"
Label5	Name	Label5

Label6	Name	Label6
Label7	Name	Label7
Label8	Name	Label8
Label9	Name	Label9
Label10	Name Text	Label10 "Slider"
Label11	Name Text	Label11 "Amplitudo"
Label12	Name Text	Label12 "Frekuensi"
NumericUpDown1	Name Value	Label12 10
Timer1	Enabled Interval	False 100
Form1	Text	"Gambar_Grafis_Sinus"

Berikut script program secara keseluruhan :

```
Public Class cosinus
    Dim i, l, x0, y0, x1, y1, x2, y2, A, freq, gain As Integer
    Dim sudut_radian, garis_null As Double
    Dim myPen As Pen

    Private Sub cosinus_Load(ByVal sender As System.Object,
        ByVal e As System.EventArgs) Handles MyBase.Load
        x0 = 0
        y0 = 75
        garis_null = y0

        x1 = x0
        y1 = y0
        i = 0
    End Sub
```

```
Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click
Timer1.Enabled = True
End Sub
```

```
Private Sub Timer1_Tick(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Timer1.Tick
```

```
Dim Grafik As Graphics = Panel1.CreateGraphics
freq = NumericUpDown1.Value
myPen = New Pen(Drawing.Color.Red, 2)
x1 = x1 + 1
A = 50 - VScrollBar1.Value
sudut_radian = Math.Sin(x1 * ((360 * freq) * (3.14 /
180)))
y1 = garis_null + (sudut_radian * A)
Grafik.DrawLine(myPen, x0, y0, x1, y1)
Label5.Text = x0
Label6.Text = y0
Label7.Text = x1
Label8.Text = y1
Label9.Text = VScrollBar1.Value
```

```
If x1 >= 300 Then
Panel1.Refresh()
    x1 = 0
End If
```

```
y0 = y1
x0 = x1
End Sub
End Class
```

Pembahasan program

Program aplikasi Grafik_slider dan Grafik_sinus hampir sama, karena memang Grafik_sinus hanyalah pengembangan dari program aplikasi Grafik_slider. Oleh karena itu pembahasan program kali ini tidak akan panjang, hanya mencakup perbedaan dari script yang ada dari kedua program tersebut saja.

Perbedaan antara program Grafik_slider dengan program Grafik_sinus adalah sumber penentuan titik koordinat Y1 sebagai penentu titik tujuan penggambaran. Jika pada program Grafik_slider sumber titik Y1 di ambil dari perubahan nilai Y1, maka pada program Grafik_sinus sumber titik Y1 berasal dari fungsi matematika bentuk dasar gelombang sinusoida

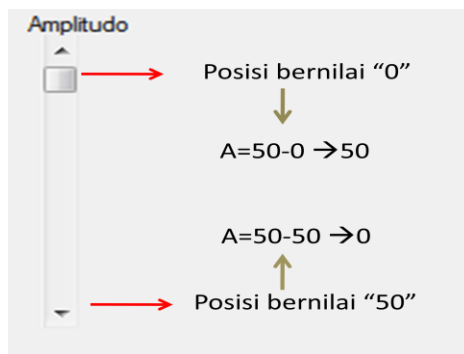
$$Y = A \sin(2\pi ft)$$

Akan tetapi karena kita hendak menggambar pada bidang datar maka kita perlu merubah satuan dari derajat ke radian, sehingga persamaan berubah menjadi

$$Y = A \sin(2\pi ft) \times \left(\frac{\pi}{180^\circ}\right)$$

A merupakan amplitudo, nilai variabel A dipengaruhi oleh nilai VScrollBar1. Karena nilai VScrollBar1 bernilai 0 pada posisi slider di atas dan bernilai 50 saat slider dibawah, maka kita perlu membalik nilai tersebut dengan mengetikan script berikut :

$$A = 50 - VScrollBar1.Value$$



Selanjutnya kita melakukan fungsi operasi matematika $\sin(2\pi ft) \times \left(\frac{\pi}{180^\circ}\right)$, untuk kita menuliskan program seperti berikut :

```
sudut_radian = Math.Sin(x1 * ((360 * freq) * (3.14 /
180)))
```

sudut_radian merupakan variabel fungsi yang digunakan untuk menampung hasil operasi matematika. Sedangkan operasi matematika dalam visual basic 2010 selalu diawali perintah "Math.". Kecepatan sudut ω identik dengan $2\pi f$, sedangkan 2π itu sendiri sama dengan sudut 360 derajat.

Sedangkan proses mengubah derajat menjadi radian dengan melakukan perkalian $\left(\frac{\pi}{180^\circ}\right)$ dapat di wakili script dalam program `...*(3.14 / 180)`.

Karena awal penggambaran di mulai dari titik tengah panel, maka kita tentukan dulu titik awal dimana kita akan memulai melakukan fungsi penggambaran. Untuk itu kita menggunakan variabel garis_null yang telah di set sebagai titik tengah yakni koordinat (0,150). Setelah itu baru kita menambahkan nilai garis_null dengan sudut_radian.

```
y1 = garis_null + (sudut_radian * A)
```

Variabel sudut_radian menampung hasil operasi matematika gelombang sinus, proses ini akan menghasilkan penentuan perubahan titik koordinat Y dari gelombang sinus tiap satuan waktu. Faktor lain yang berpengaruh adalah variabel frekuensi, variabel ini akan mempengaruhi jumlah gelombang yang dihasilkan tiap periode waktu tertentu. Dengan demikian masih ada satu variabel lagi yang kurang untuk memenuhi persamaan dasar $Y = A \sin(2\pi ft)$, yaitu variabel A atau amplitudo.

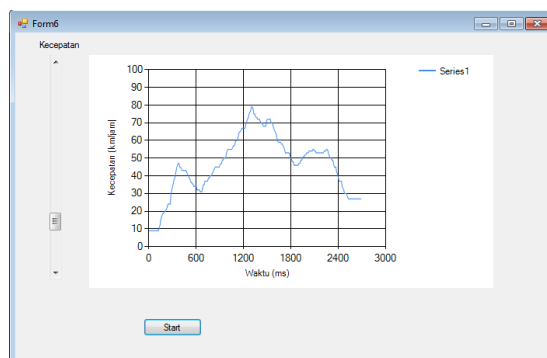
Oleh karena itu pada penghitungan titik koordinat Y1, variabel sudut_radian di kali dengan variabel A terlebih dahulu seperti tampak pada script di atas. Jalankan program dan ubah-ubahlah nilai dari

frekuensi dan amplitudo sehingga tampak perubahan gelombang sinusoida yang dihasilkan.

2.3 Membuat Chart Grafik

Metode penyajian sekumpulan data menjadi suatu grafik dalam visual basic 2010 selain dengan menggunakan metode **DrawLine**, kita dapat juga menggunakan metode **Chart**. Metode ini relatif lebih mudah karena kita memanfaatkan komponen chart pada toolbox yang telah disediakan oleh Visual basic. Penggunaan komponen ini memudahkan programmer untuk merancang aplikasi dalam bentuk grafik, akan tetapi juga memiliki keterbatasan dalam tampilan visual. Tampilan visual yang dihasilkan aplikasi kurang halus terutama ketika proses *Autoscale* yang dilakukan oleh program secara otomatis.

Gambar di bawah ini menunjukkan tampilan hasil aplikasi dari program grafik menggunakan komponen **Chart**.

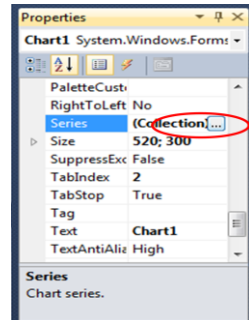


Gambar 2.3 Ilustrasi chart grafik

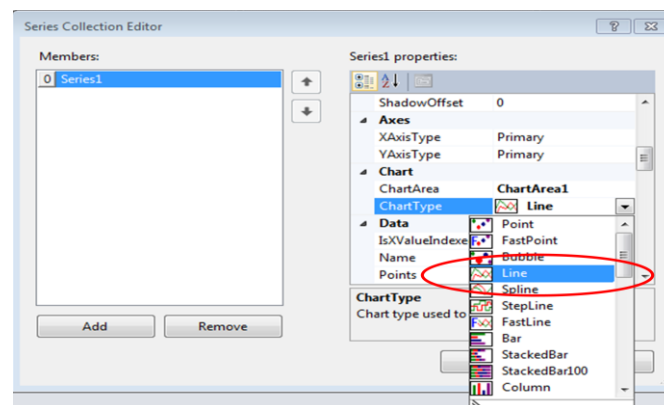
Untuk membuat program aplikasi grafik seperti contoh di atas, ikuti langkah-langkah membuat program berikut :

1. Tambahkan *Windows Form* baru dengan mengarahkan kursor pada menu *Project*, klik *Add Windows Form* dan beri nama *Grafik_Chart*.
2. Sesuaikan ukuran form hingga tampak kurang lebih seperti gambar di atas.
3. Arahkan kursor pada **toolbox**, pilih menu **Data** dan klik-lah komponen **Chart**.

- Atur properties komponen Chart untuk mengubah tipe jenis chart menjadi grafik dengan cara arahkan kursor ke properties, pilih menu **Series**, kliklah kotak kecil yang berada di sebelah kanan dari properti series tersebut.

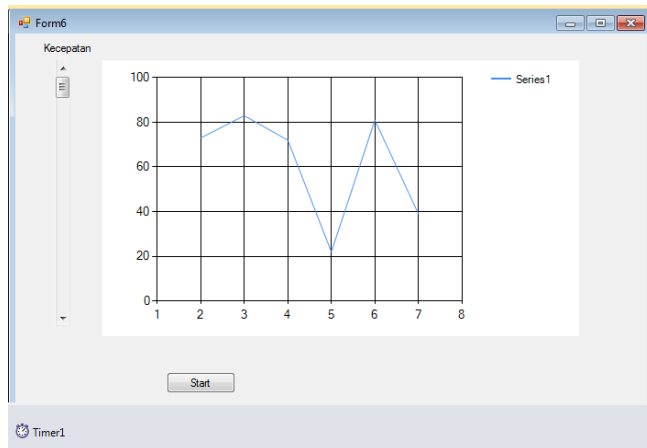


- Setelah anda klik icon tersebut maka akan muncul jendela baru editor dari properties **Series**. Arahkan kursor pada menu **ChartType** dan pilihlah menu **Line**.



- Tambahkan komponen **Timer1**.
- Tambahkan komponen tombol **Button1** dan ubah properti Text-nya menjadi **"Start"**.
- Tambahkan komponen **VScrollBar1**, letakan disamping Chart1 dan sesuaikan ukuran tampilan menjadi kurang lebih seperti gambar berikut.

Teknik Pemrograman



Gambar 2.4 Form design program chart grafik

Untuk lebih detail dari properties komponen yang digunakan, perhatikan tabel berikut :

Obyek	Properti	Pengaturan
Chart1	Name	Chart1
	ChartType	Line
VScrollBar1	Name	VScrollBar1
	LargeChange	1
	Maximum	100
	Minimum	0
Button1	Name	Button1
	Text	"Start"
Label1	Name	Label1
	Text	"Kecepatan"
Timer1	Enabled	False
	Interval	100
Form1	Text	"Gambar_Grafis_Chart"

Berikut script program secara keseluruhan :

```
Public Class Form6
    Dim gambar_plot As New DataTable
    Dim x,y As Integer

    Private Sub Form6_Load(ByVal sender As System.Object,
        ByVal e As System.EventArgs) Handles MyBase.Load

        gambar_plot.Columns.Add("Waktu", GetType(Integer))
        gambar_plot.Columns.Add("Kecepatan", GetType(Integer))

        With Chart1.ChartAreas(0)
            .AxisX.Minimum = 0
            .AxisX.Maximum = 3000
            .AxisY.Minimum = 0
            .AxisY.Maximum = 100
            .AxisY.Interval = 10
            .AxisX.Title = "Waktu (ms)"
            .AxisY.Title = "Kecepatan (km/jam)"
        End With

        x = 0
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object,
        ByVal e As System.EventArgs) Handles Button1.Click

        Timer1.Enabled = True
    End Sub
```

```
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal  
    e As System.EventArgs) Handles Timer1.Tick  
  
    y = 100 - VScrollBar1.Value  
    gambar_plot.Rows.Add(x, y)  
  
    With Chart1.Series(0)  
        .Points.DataBind(gambar_plot.DefaultView, "Waktu",  
            "kecepatan", Nothing)  
        .BorderWidth = 1  
    End With  
  
    x = x + 10  
    If x >= 3000 Then  
        Chart1.Refresh()  
        x = 0  
        y = 0  
    End If  
End Sub  
End Class
```

Pembahasan Program

Program dimulai dengan mendeklarasikan semua variabel yang akan kita gunakan dalam penulisan program.

```
Dim gambar_plot As New DataTable
Dim x,y As Integer
```

Variabel x dan y adalah variabel penampung koordinat-x dan koordinat-y pada komponen chart. Sedangkan variabel gambar_plot merupakan variabel penampung data koordinat dalam bentuk bentuk database tabel. Data tabel disini diperlukan karena fungsi penggambaran dengan menggunakan komponen Chart membutuhkan fungsi database sebagai sumber penggambaran dari titik ke titik-nya.

Pada form load atau default dari tampilan program awal, di tampilkan pengaturan ukuran dari chart. Chart memiliki nilai x maksimum sebesar 3000 dan y maksimum sebesar 100.

```
With Chart1.ChartAreas(0)
    .AxisX.Minimum = 0
    .AxisX.Maximum = 3000
    .AxisY.Minimum = 0
    .AxisY.Maximum = 100
    .AxisY.Interval = 10
    .AxisX.Title = "Waktu (ms) "
    .AxisY.Title = "Kecepatan (km/jam) "
End With
```

Disamping itu juga disiapkan tabel terdiri dari dua baris kolom sebagai sumber penyimpanan database koordinat x dan y. Kedua kolom tersebut diberi nama kolom "Waktu" dan kolom "Kecepatan".

```
gambar_plot.Columns.Add("Waktu", GetType(Integer))
gambar_plot.Columns.Add("Kecepatan", GetType(Integer))
```

Tombol Button1 berfungsi untuk menjalankan timer1, dimana semua program penulisan grafik dari titik ke titik berada di sub program timer1. Sehingga jika timer1 aktif maka semua isi program di dalamnya akan dieksekusi dan program aplikasi bisa dijalankan seperti yang kita rancang dari awal.

```
Timer1.Enabled = True
```

Sub program timer1 merupakan program utama yang akan menjalankan penggambaran grafik dari titik ke titik. Titik koordinat-x didapatkan dari fungsi incremen 10 dari timer ($x=x+10$), sedangkan fungsi y didapat dari nilai perubahan VScrollBar1. Akan tetapi sifat default dari VScrollBar selalu bersifat terbalik dimana posisi slider di atas bernilai 0 dan posisi slider di bawah bernilai maksimum (dalam contoh :100), maka kita perlu membalik dengan mengurangi dari nilai maksimum-nya ($y = 100-y$).

```
y = 100 - VScrollBar1.Value
gambar_plot.Rows.Add(x, y)
```

Pada baris berikutnya adalah kode fungsi program memasukan nilai koordinat-x dan koordinat-y ke dalam database kolom "kecepatan" dan kolom "waktu".

Langkah selanjutnya adalah membaca tabel data base yang telah dibuat dan menggambarkan ke dalam grafik chart berikut properti ketebalan dari plot grafik.

```
With Chart1.Series(0)
    .Points.DataBind(gambar_plot.DefaultView, "Waktu",
        "kecepatan", Nothing)
    .BorderWidth = 1
End With
```

TUGAS

Buatlah suatu program aplikasi yang grafik gelombang sinus cosinus, dimana satu tampilan jendela grafik terdapat dua buah gelombang yaitu gelombang sinus berwarna biru dan gelombang cosinus berwarna merah.

Kedua gelombang tersebut merepresentasikan persamaan sebagai berikut :

Gelombang sinus → $Y = A \sin(2\pi ft)$

Gelombang cosinus → $Y = A \cos(2\pi ft)$

Dimana :

- A , amplitudo, adalah puncak deviasi fungsi dari posisi pusatnya
- π , konstanta phi bernilai 3,14.
- t , waktu, nilai waktu dari detik ke detik dimana osilasi terjadi (direpresentasikan pada fungsi x atau horisontal pada grafik maupun diagram kartesian).
- f , frekuensi, yang menentukan berapa banyak osilasi terjadi dalam interval waktu tertentu (Hz)

TES FORMATIF

1. Bagaimana cara merubah sudut derajat menjadi radian dan jelaskan mengapa saat kita menggambar fungsi gelombang sinus pada aplikasi program, kita harus merubahnya terlebih dahulu menjadi radian!
2. Sebutkan kelebihan dan kekurangan menggambar grafik dengan metode Drawline dan dengan menggunakan komponen Chart.
3. Mengapa dalam menggambar grafik pada program visual basic, ditekankan kita harus mengetahui koordinat awal, koordinat tujuan dan titik awal penggambaran?

Jawab :

KEGIATAN 3 : Membuat GUI Animasi Obyek Bergerak Pada VB 2010

3.1 Menggerakkan suatu obyek

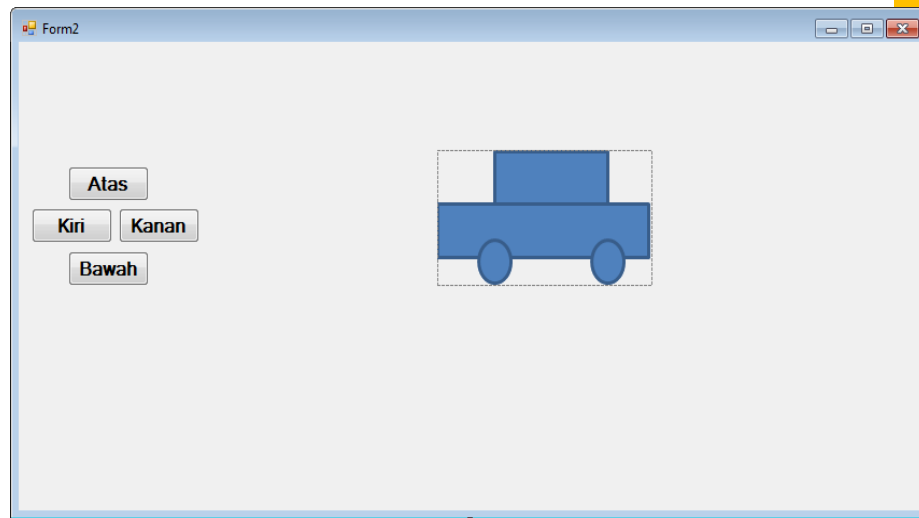
Meskipun Visual Basic 2010 secara umum adalah sebuah bahasa pemrograman yang dibuat untuk bisnis dan aplikasi industri dan bukan untuk membuat bentuk animasi, tetapi Visual Basic 2010 dapat digunakan untuk membuat dasar dari obyek animasi.

Pada bagian ini Anda akan belajar bagaimana menggerakkan suatu obyek dengan menekan tombol perintah. Anda perlu membuat properti Top dan Left dari sebuah obyek untuk membuat animasi. Properti Top mendefinisikan jarak obyek dari batas atas layar, sedangkan properti Left mendefinisikan jarak obyek dari batas kiri layar. Dengan menambahkan jarak obyek kita dapat membuat efek animasi atau menggerakkan obyek.

Mulailah dengan project baru yang bernama Obyek_Bergerak atau nama apapun yang Anda inginkan. Kemudian masukkan PictureBox dan pada propertinya pilihlah gambar yang anda inginkan yang dipilih dari file anda.

Contoh : buat gambar mobil sederhana dengan menggunakan aplikasi *windows paint*, simpan gambar tersebut di hardisk komputer kemudian pada visual basic 2010, khususnya pada komponen *PictureBox*, *imports* gambar tersebut hingga muncul pada jendela aplikasi form.

Selanjutnya, masukkan empat buah tombol perintah atau command button beri nama komponen-komponen tersebut **BtnAtas**, **BtnBawah**, **BtnKiri** dan **BtnKanan** serta ganti properties **Text** menjadi **Atas**, **Bawah**, **Kiri** dan **Kanan**, sehingga tampilan jendela form aplikasi akan tampak seperti gambar berikut :



Gambar 3.1 Tampilan aplikasi program animasi bergerak

Kemudian klik pada tombol-tombol tersebut dengan menuliskan kode program berikut pada code editor :

```
Private Sub BtnAtas_Click(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles Button1.Click
    PictureBox1.Top = PictureBox1.Top - 10
End Sub
```

```
Private Sub BtnKanan_Click(ByVal sendr As System.Object,
    ByVal e As System.EventArgs) Handles Button2.Click
    PictureBox1.Left = PictureBox1.Left + 10
End Sub
```

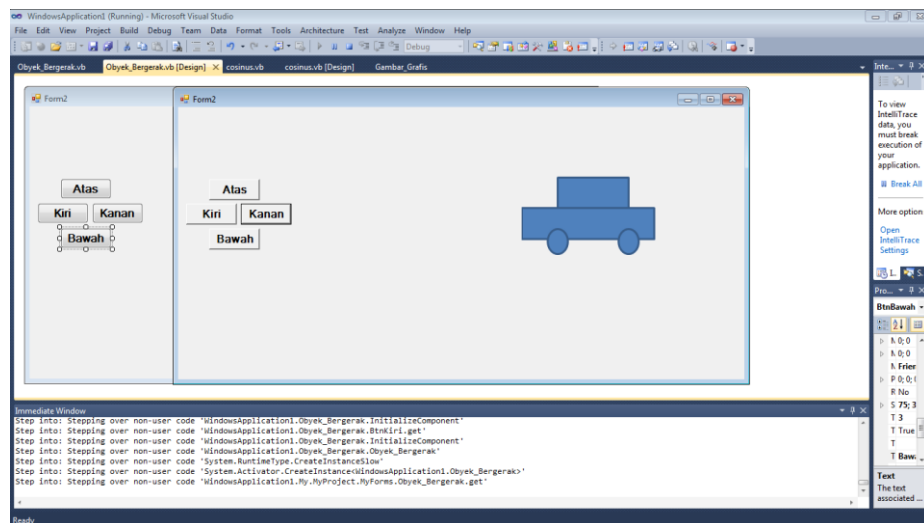
```
Private Sub Btnkiri_Click(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles Button4.Click
    PictureBox1.Left = PictureBox1.Left - 10
End Sub
```

```
Private Sub BtnBawah_Click(ByVal sendr As System.Object,
    ByVal e As System.EventArgs) Handles Button3.Click
    PictureBox1.Top = PictureBox1.Top + 10
End Sub
```

Pembahasan Program :

Setiap kali pengguna mengklik tombol **Bawah**, maka jarak dari PictureBox akan bertambah 10 piksel dari posisi batas atas, menciptakan gerakan kebawah. Di sisi lain, setiap kali pengguna mengklik tombol **Atas**, maka jarak PictureBox berkurang sebesar 10 piksel dari posisi batas atas, sehingga menciptakan gerakan keatas. Selain itu, setiap kali pengguna mengklik pada tombol **Kiri**, jarak dari PictureBox berkurang sebesar 10 piksel dari batas kiri, sehingga menciptakan gerakan ke kiri. Terakhir, setiap kali pengguna mengklik tombol **Kanan**, maka jarak PictureBox akan bertambah sebesar 10 piksel dari batas kiri, sehingga menciptakan gerakan ke kanan.

Tampilan dari program tersebut adalah :

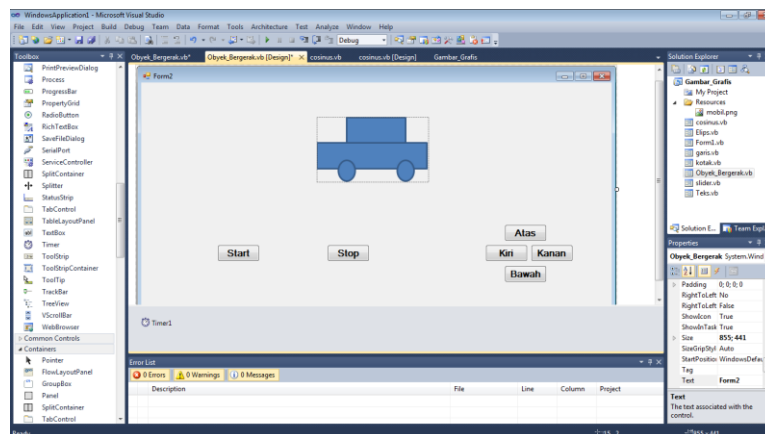


Gambar 3.2 Form design aplikasi program animasi bergerak

3.2 Membuat Animasi Menggunakan Timer

Animasi selanjutnya yang hendak kita pelajari adalah membuat gerakan animasi dengan menggunakan timer. Fungsi timer disini adalah untuk menggerakkan obyek secara otomatis tanpa perlu mengklik tombol perintah secara manual. Anda dapat membuat gerakan dari kiri ke kanan atau dari atas ke bawah dengan menuliskan kode program.

Pada aplikasi Obyek_Bergerak yang telah dibuat sebelumnya, tambahkan dua buah komponen Button. Ubah properties text button1 = "Start" dan text button2 = "Stop". Kemudian masukkan sebuah kontrol Timer ke dalam form, dan atur properti intervalnya menjadi 100, yang berarti sama dengan 0,1 detik.



Gambar 3.3 Form design aplikasi program animasi dengan timer

Disini kita menggunakan properti Left dari PictureBox untuk membuat gerakan. PictureBox.Left berarti jarak PictureBox dari batas kiri Form. Sekarang klik kontrol Timer dan ketiklah kode berikut :

```
Private Sub Timer1_Tick (ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer1.Tick
    If pictureBox1.Left < Me.Width Then
        pictureBox1.Left = pictureBox1.Left + 10
    Else
```

```

PictureBox1.Left = 0
End If
End Sub

```

Pada kode diatas, Me.Width merepresentasikan lebar Form. Jika jarak PictureBox dari kiri kurang dari lebar Form, maka nilai 10 akan ditambahkan pada jarak PictureBox dari batas kiri setiap kali Timer menghitung atau setiap 0,1 detik. Ketika jarak PictureBox1 dari batas kiri adalah sama dengan lebar dengan Form, maka jarak dari batas kiri akan diatur sebesar 0, yang menggerakkan obyek PictureBox1 kembali ke batas kiri kemudian menggerakkan ke kanan lagi, sehingga menciptakan gerakan dari kiri ke kanan secara berulang-ulang. Kita perlu untuk menambahkan sebuah tombol **“Stop”** untuk menghentikannya. Kode programnya adalah :

```

Timer1.Enabled = False

```

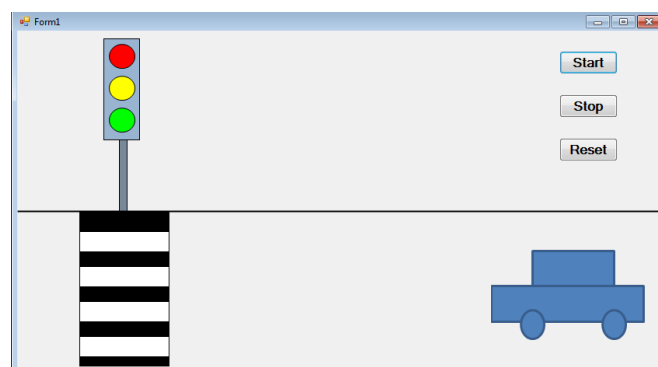
Untuk menggerakkan obyek PictureBox1, kita masukkan sebuah tombol **“Start”** dengan kode :

```

Timer1.Enabled = True

```

3.3 Animasi Bergerak Lampu Lalu Lintas



Gambar 3.4 Tampilan aplikasi animasi bergerak rambu lalu lintas

Tampilan program aplikasi di atas merupakan implementasi sederhana program animasi bergerak yang dipadukan dengan program timer untuk

visualisasi rambu lalu lintas seperti yang telah di bahas pada buku Teknik Pemrograman 1 pada semester sebelumnya.

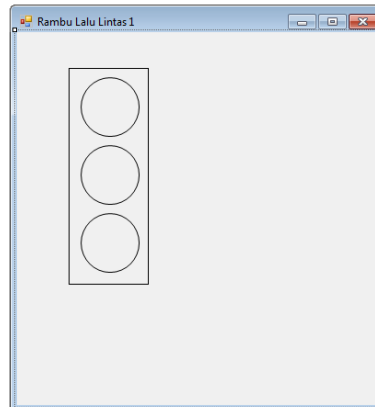
Penambahan komponen yang diperlukan untuk program di atas adalah dengan menambahkan zebra cross yang di buat dari **RectangleShape** berwarna hitam dan putih, **PictureBox** yang meng-import gambar mobil, tombol **Button** untuk tombol “Start”, “Stop” dan “Reset” serta komponen **OvalShape** sebagai tanda sinyal lampu lalu lintas berwarna merah, kuning dan hijau.

Jalannya program aplikasi dimulai dengan menekan tombol “Start” sehingga mengaktifkan timer1 dan timer2. Fungsi timer1 untuk menjalankan visualisasi rambu lalu-lintas merah, kuning dan hijau, sedangkan fungsi timer2 untuk menggerakkan animasi mobil sebagai obyek bergerak.

Obyek mobil akan bergerak melewati jalan secara terus menerus selama warna lampu rambu lalu lintas hijau. Jika rambu lalu lintas berganti kuning dan hijau maka obyek mobil akan berhenti tepat sebelum posisi zebra cross berada.

Berikut langkah-langkah pembuatan program aplikasi animasi bergerak rambu lalu lintas ;

1. Buatlah project baru dengan mengarahkan kursor pada menu File, klik New Project dan buatlah Form baru dengan nama Animasi_Lampu_Lalu_Lintas.
2. Sesuaikan ukuran form hingga tampak kurang lebih seperti gambar di atas.
3. Klik komponen **RectangleShape1** pada Toolbox dan letakan di form sebagai bingkai dari lampu-lampu lalu lintas merah, kuning dan hijau.
4. Tambahkan komponen **OvalShape1**, **OvalShape2** dan **OvalShape3** pada form dan letakan di dalam **RectangleShape1**. Atur letak ketiga OvalShape tersebut berderet secara vertikal seperti gambar berikut.



Gambar 3.5 Komponen Program Aplikasi Rambu Lalu Lintas 1

5. Tambahkan komponen **Button1**, **Button2** dan **Button3** dan letakkan di dalam *GroupBox1* dan atur properties text komponen *Button1* = "Start", *Button2*="Stop" dan *Button3*="Reset".
6. Tambahkan komponen *RectangleShape* sebagai tiang dari rambu lalu lintas.
7. Berikan garis sepadan jalan dengan menarik komponen *LineShape1* dan ubah *Borderwidth* = 2.
8. Buatlah tampilan zebra cross dengan menambahkan komponen *RectangleShape* berwarna hitam, atur properti *BackColor* = *ActiveCaptionText*, *BackStyle* = *Opaque*. Tambahkan empat *RectangleShape* berwarna putih dengan mengubah properti *BackColor* = *White*, *BackStyle* = *Opaque*.
9. Tambahkan *PictureBox* pada form, klik kanan dan pilih *Choose Image*. Pilihlah gambar mobil yang telah anda buat dan anda simpan dikomputer.
10. Tambahkan dua buah komponen timer dengan mengklik dua kali icon komponen timer pada toolbox.

11. Secara umum semua komponen yang kita butuhkan sudah tersedia pada form design, sedangkan untuk detail setting komponen properties dapat di lihat dari tabel berikut:

Obyek	Properti	Pengaturan
RectangleShape1	Name BackStyle BackColour	RectangleShape1 Opaque ActiveCaption
RectangleShape2	Name BackStyle BackColour	RectangleShape2 Opaque LightSlightGray
RectangleShape3	Name BackStyle BackColour Location Size	RectangleShape3 Opaque ActiveCaptiontext 80;232 115;200
RectangleShape4	Name BackStyle BackColour Location Size	RectangleShape4 Opaque White 115;26
RectangleShape5	Name BackStyle BackColour Size	RectangleShape5 Opaque White 115;26
RectangleShape6	Name BackStyle BackColour Size	RectangleShape6 Opaque White 115;26
RectangleShape7	Name BackStyle BackColour Size	RectangleShape7 Opaque White 115;26

Teknik Pemrograman

OvalShape1	Name BackColor BackStyle	OvalShape1 Lime Opaque
OvalShape2	Name BackColor BackStyle	OvalShape2 Yellow Opaque
OvalShape3	Name BackColor BackStyle	OvalShape3 Red Opaque
Button1	Name Text	Button1 "Start"
Button2	Name Text	Button2 "Stop"
Button3	Name Text	Button3 "Reset"
LineShape1	Name BorderWidth	LineShape1 2
Timer1	Enabled Interval	False 100
Timer2	Enabled Interval	False 100
Form1	Text	"Animasi Rambu Lalu Lintas"

Berikut script program secara keseluruhan :

```

Public Class Animasi_Lampu_Lalu_Lintas
    Dim counter As Integer
    Dim flag As Integer
    Dim t As Double

    Private Sub Animasi_Lampu_Lalu_Lintas_Load(ByVal sender
    As System.Object, ByVal e As System.EventArgs) Handles
    MyBase.Load

        Timer1.Enabled = False
        Timer2.Enabled = False

        OvalShapel.BackColor = Color.DarkGray
        OvalShape2.BackColor = Color.DarkGray
        OvalShape3.BackColor = Color.DarkGray

        Timer1.Interval = 1000
        t = 0
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles Button1.Click

        OvalShapel.BackColor = Color.Lime
        Timer1.Enabled = True
        Timer2.Enabled = True
        Timer1.Start()
    End Sub

```

```
Private Sub Button2_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles Button2.Click
```

```
Timer1.Enabled = False
```

```
Timer2.Enabled = False
```

```
End Sub
```

```
Private Sub Timer1_Tick(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles Timer1.Tick
```

```
If t >= 0 And t <= 5 Then
```

```
    OvalShape1.BackColor = Color.Lime
```

```
    OvalShape2.BackColor = Color.DarkGray
```

```
    OvalShape3.BackColor = Color.DarkGray
```

```
    flag = 1
```

```
ElseIf t >= 6 And t <= 8 Then
```

```
    OvalShape1.BackColor = Color.DarkGray
```

```
    OvalShape2.BackColor = Color.Yellow
```

```
    OvalShape3.BackColor = Color.DarkGray
```

```
    flag = 0
```

```
ElseIf t >= 9 And t <= 12 Then
```

```
    OvalShape1.BackColor = Color.DarkGray
```

```
    OvalShape2.BackColor = Color.DarkGray
```

```
    OvalShape3.BackColor = Color.Red
```

```
    flag = 0
```

```
End If
```

```
t = t + 1
```

```
If t = 12 Then t = 0
```

```
End Sub
```

```

Private Sub Timer2_Tick(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Timer2.Tick
If flag = 1 Then
    PictureBox1.Left = PictureBox1.Left - 50
    If PictureBox1.Left < 0 Then
        PictureBox1.Left = 650
    End If
ElseIf flag = 0 Then
    If (flag = 0 And PictureBox1.Left <> 200) Then
        PictureBox1.Left = PictureBox1.Left - 50
        If PictureBox1.Left < 0 Then
            PictureBox1.Left = 650
        End If
    End If
End If
End Sub

```

```

Private Sub Button3_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button3.Click

PictureBox1.Left = 650
Timer1.Enabled = False
Timer2.Enabled = False

End Sub
End Class

```

Pembahasan program

Program aplikasi Animasi_Lampu_Lalu_Lintas dimulai dengan menginisialisasi kondisi lampu menjadi non-aktif semua ketika program di debug pertama kali.

```
OvalShape1.BackColor = Color.DarkGray
OvalShape2.BackColor = Color.DarkGray
OvalShape3.BackColor = Color.DarkGray
```

Kemudian kita mengatur interval timer menjadi 1000 atau sama dengan 1 detik. Disamping itu diperlukan variabel bebas “t” yang akan digunakan sebagai fungsi penghitung waktu penyalan lampu lalu lintas. Sebagai contoh jika pada program tertulis $t \geq 0$ and $t \leq 5$, artinya nilai t mulai dari 0 s/d 5 atau bisa dikatakan $t = 6$ satuan. Karena interval telah di atur 1000, maka waktu penyalan lampu tersebut menjadi $6 \times 1000 = 6000$ ms atau 6 detik.

Langkah selanjutnya adalah ketika tombol “Start” di tekan, maka lampu warna hijau akan dinyalakan kemudian semua timer di aktifkan serta memanggil sub program bagian timer1 seperti pada potongan script program dibawah ini.

```
OvalShape1.BackColor = Color.Lime
Timer1.Enabled = True
Timer2.Enabled = True
Timer1.Start()
```

Sub program timer1 bertujuan untuk mengatur penyalan lampu lalu lintas. Prinsip pengaturannya adalah dengan melakukan increment variabel t ($t=t+1$) serta melakukan pembagian kriteria. Jika variabel t berada pada range $0 \leq t \leq 5$ maka lampu warna hijau menyala.

```
If t >= 0 And t <= 5 Then
OvalShape1.BackColor = Color.Lime           → lampu hijau
OvalShape2.BackColor = Color.DarkGray      → lampu kuning
OvalShape3.BackColor = Color.DarkGray      → lampu merah
```

```
flag = 1
```

Fungsi variabel “flag” adalah sebagai tanda apakah obyek animasi harus bergerak atau berhenti. Variabel “flag = 1” artinya obyek animasi harus bergerak, sedangkan variabel “flag = 0” berarti obyek animasi harus berhenti.

Selanjutnya range variabel t pada daerah $6 \geq t \geq 8$ maka lampu lalu lintas warna kuning akan aktif dan variabel flag bernilai 0. Range terakhir nilai t pada daerah $9 \geq t \geq 12$, dimana lampu warna merah akan aktif menyala.

Setelah program lampu merah di eksekusi, maka proses selanjutnya adalah melakukan fungsi perulangan semua program yang telah kita lakukan tadi. Untuk itu kita harus me-reset kembali variabel t menjadi 0, sehingga proses perulangan mulai pengaktifan warna hijau dapat dilakukan kembali. Untuk me-reset variabel t maka script berikut perlu di tuliskan.

```
If t = 12 Then t = 0
```

Pembahasan selanjutnya adalah memahami proses menggerakkan obyek PictureBox1 yang di atur oleh sub program timer2. Pada sub program timer2 jika flag bernilai “1”, maka obyek akan bergerak ke kiri. Proses ini dengan memberikan nilai “-50” pada properties “left” pada komponen PictureBox1 seperti pada potongan script dibawah ini.

```
PictureBox1.Left = PictureBox1.Left - 50
```

Kondisi selanjutnya adalah dengan memberikan syarat jika PictureBox1.Left= 0 (berada di ujung kiri dari form aplikasi) maka obyek gambar mobil di reset ke posisi awal yakni koordinat 650.

```
If PictureBox1.Left < 0 Then
    PictureBox1.Left = 650
End If
```

Tombol “stop” digunakan untuk menghentikan program secara sementara, artinya ketika tombol ditekan maka gerakan animasi obyek bergerak akan

berhenti dan rambu lalu lintas juga akan berhenti sejenak. Untuk menghasilkan skenario program seperti itu maka kita perlu me-nonaktifkan semua timer dengan membuat properties enable dari komponen timer menjadi false.

```
Timer1.Enabled = False
Timer2.Enabled = False
```

Tombol reset bertujuan untuk mengembalikan posisi obyek gerak kembali ke titik awal dan membuat program kembali ke posisi awal secara keseluruhan. Untuk itu lokasi koordinat properti "left" kompoenen PictureBox1 di ubah ke 650 dan semua timer di non-aktifkan.

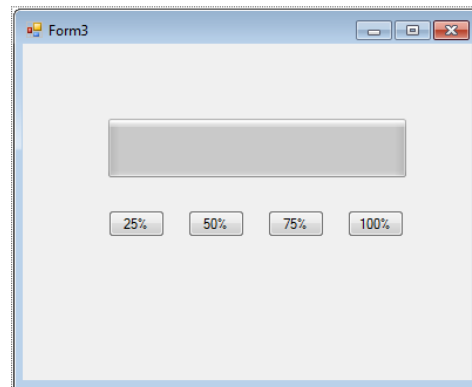
```
PictureBox1.Left = 650
```

3.4 Visualisasi Progress Bar

ProgressBar merupakan sebuah kotak dimana bagian tengahnya bisa diberi nilai yang menunjukkan presentase terhadap keseluruhan suatu nilai. Kondisi nilai ProgressBar akan berisi maksimal (100) ketika ProgressBar berisi penuh dan akan bernilai minimal (0) ketika ProgressBar kosong.

Visualisasi ProgressBar umumnya digunakan untuk menunjukkan loading atau proses tunggu dari suatu even, akan tetapi didalam dunia teknik khususnya elektro dan elektronika, ProgressBar dapat di modifikasi menjadi suatu tampilan HMI (Human Machine Interface). Sebagai contoh kita dapat memanfaatkan ProgressBar sebagai tampilan untuk visualisasi kontrol level ketinggian tangki air, visualisasi kontrol distribusi air, visualisasi jalannya arus dari suatu rangkaian dan lain-lain.

Untuk dasar membuat aplikasi ProgressBar, perhatikan contoh program berikut dan ikuti langkah-langkah untuk membangun program aplikasi tersebut.



Gambar 3.6 Form design aplikasi program Progress bar

Berikut langkah-langkah membuat program aplikasi ProgressBar :

1. Buat Form baru dan masukan ProgressBar ke dalam Form.
2. Masukan empat tombol Button dibawah ProgressBar tersebut.
3. Ganti properti teks dari masing-masing tombol dengan nilai prosentasi. Button1 = "25%", Button2 = "50%", Button3 = "75%", Button4 = "100%".
4. Klik dua kali tombol "25%" dan tuliskan script berikut diantara private sub button1 dan end sub-nya.

```
ProgressBar1.Value = 25
```

5. Klik dua kali tombol "50%" dan tuliskan script berikut diantara private sub button2 dan end sub-nya.

```
ProgressBar1.Value = 50
```

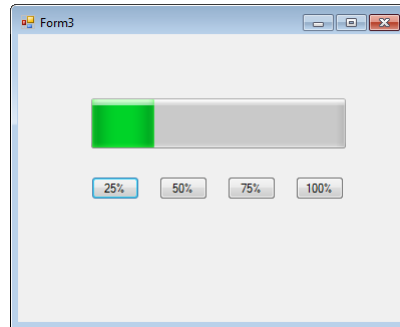
6. Klik dua kali tombol "75%" dan tuliskan script berikut diantara private sub button3 dan end sub-nya.

```
ProgressBar1.Value = 75
```

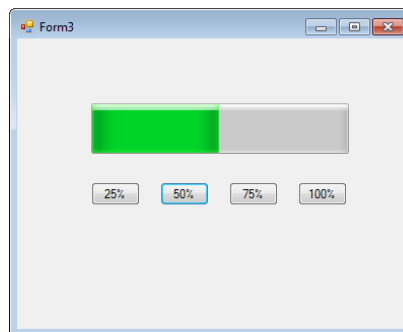
7. Klik dua kali tombol "100%" dan tuliskan script berikut diantara private sub button4 dan end sub-nya.

```
ProgressBar1.Value = 100  
MsgBox("Proses Sempurna")
```

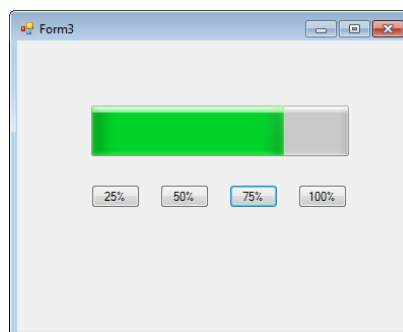
8. Jalankan program aplikasi dan klik button1 sehingga nilai progressbar menjadi 25, yang artinya seperempat dari nilai total.



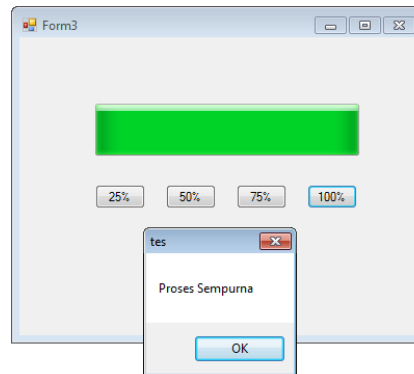
9. Ketika Button2 diklik, nilai progressbar menjadi 50 yang artinya separuh dari nilai total.



10. Ketika Button3 diklik, nilai progressbar menjadi 75 yang artinya tiga per empat dari nilai keseluruhan.



11. Ketika Button4 diklik, nilai progressbar menjadi 100 yang artinya keseluruhan dari nilai total.



Gambar 3.7 hasil aplikasi program progress bar

3.5 Vertikal Progress Bar

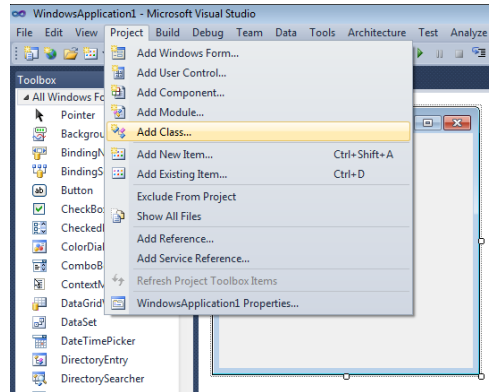
Dalam aplikasi dunia teknik sering kali dibutuhkan proses memonitoring suatu keadaan secara real time dan ditampilkan visualisasi-nya pada monitor komputer (*Human Machine Interface*). Salah satu aplikasi yang paling sering dibutuhkan dalam melakukan data akusisi secara real time adalah memonitoring level ketinggian volume suatu tangki dari proses produksi di suatu industri.

Pada visual basic 2010 cara paling mudah untuk menghasilkan proses ini adalah dengan menggunakan progress bar. Sayangnya kondisi default dari komponen progress bar pada visual basic 2010 berjalan secara horizontal saja. Jika kita ingin mendapatkan visualisasi dari progres bar bergerak ke atas atau vertikal, kita dapat melakukan hanya dengan memutar komponen tersebut.

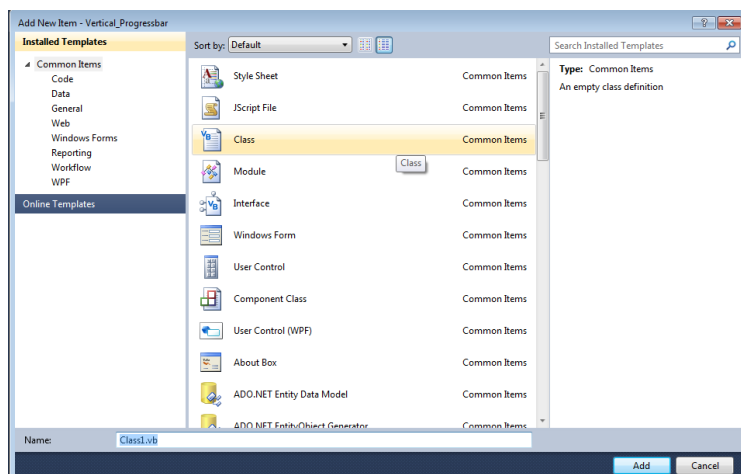
Untuk menghasilkan vertical progress bar, kita perlu menambahkan library komponen baru pada tool box dengan cara menambah modul class pada visual basic. Berikut langkah-langkah untuk membuat komponen baru vertical progress bar pada visual basic 2010 ;

1. Bukalah project baru dan pilihlah windows application form, kemudian ubah nama project menjadi Vertical_ProgressBar, ubah nama form menjadi VprogressBar dan ganti properti teks dari form menjadi "Vertical Progress Bar".

- Buatlah modul class baru dengan mengarahkan kursor pada menu **Project** dan pilih menu **Add Class** seperti gambar berikut.



- Kemudian akan muncul jendela baru yang akan memberikan menu pilihan template. Pilihlah **Class** seperti gambar di bawah ini

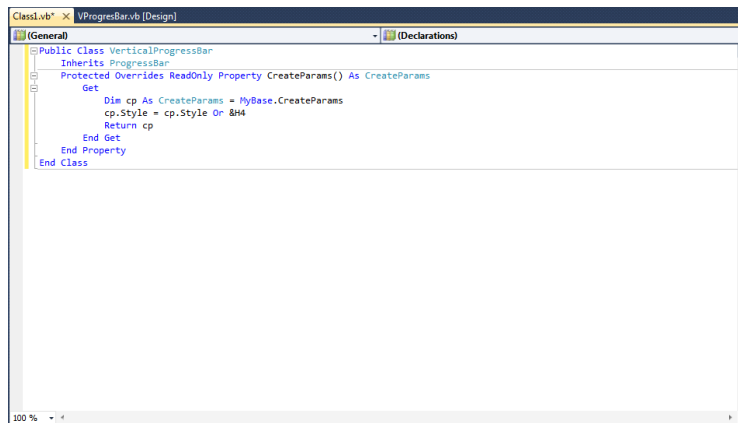


- Setelah men-double klik template Class pada langkah tiga, maka akan muncul form baru. Hapuslah semua default kode program pada form tersebut dan gantilah dengan kode script berikut :

```
Public Class VerticalProgressBar
    Inherits ProgressBar
    Protected Overrides ReadOnly Property CreateParams()
        As CreateParams

    Get
        Dim cp As CreateParams = MyBase.CreateParams
        cp.Style = cp.Style Or &H4
        Return cp
    End Get
End Property
```

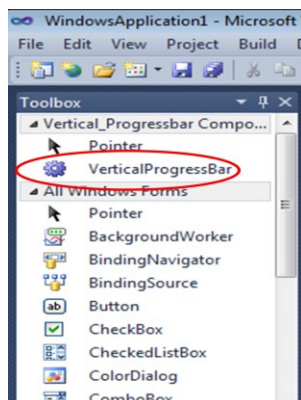
End Class



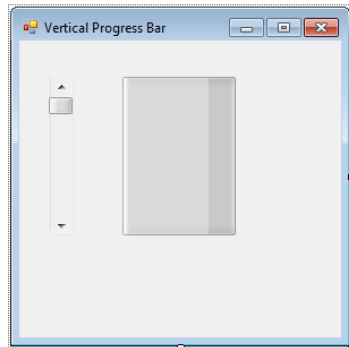
```

Public Class VerticalProgressBar
    Inherits ProgressBar
    Protected Overrides ReadOnly Property CreateParams() As CreateParams
        Get
            Dim cp As CreateParams = MyBase.CreateParams
            cp.Style = cp.Style Or &H4
            Return cp
        End Get
    End Property
End Class
    
```

5. Debug program modul class tersebut kemudian non-aktifkan kembali program tersebut.
6. Bukalah form design dari aplikasi dan perhatikan pada jendela toolbox paling kiri atas akan terdapat komponen baru bernama VerticalProgressbar.



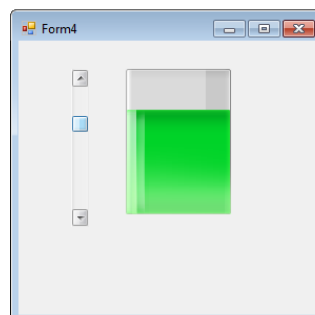
7. Pilihlah komponen VerticalProgressbar tersebut dan letakkan pada form design serta atur ukuran komponen tersebut kurang lebih seperti gambar di bawah ini.



8. Tambahkan komponen VScrollbar1 dan letakkan pada form design. Atur komponen properties LargeChange VScrollbar1 menjadi 1.
9. Double klik komponen VScrollbar1 dan ketikkan script program berikut

```
VerticalProgressBar1.Value = 100 - VScrollbar1.Value
```

10. Jalankan program dan gerakan posisi slider VScrollbar naik turun, amati pergerakan VerticalProgressbar akan bergerak mengikuti perubahan dari Nilai VScrollbar.



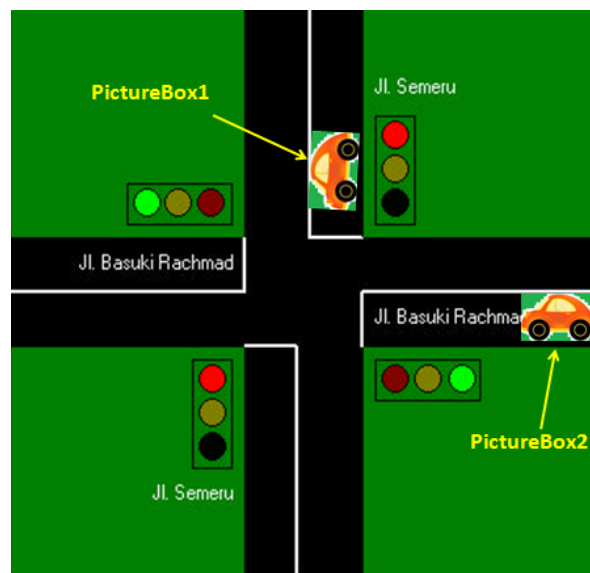
Gambar 3.7 Tampilan aplikasi program vertikal progres bar

TUGAS

Buatlah suatu program aplikasi animasi rambu lalu lintas pada persimpangan jalan Basuki rachmad dengan jalan Semeru. Aturlah nyala aktivasi lampu rambu lalu lintas sesuai dengan tabel kebenaran berikut.

Waktu	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Jl. Basuki rachmad	Green	Green	Green	Yellow	Yellow	Yellow	Red	Red	Red	Red	Red	Red	Red	Yellow	Yellow	Yellow	Green	Green	Green	Green
Jl. Semeru	Red	Red	Red	Yellow	Yellow	Yellow	Green	Green	Green	Green	Green	Green	Green	Yellow	Yellow	Yellow	Red	Red	Red	Red

Tambahkan dua buah PictureBox yang menampilkan gambar mobil seperti gambar dibawah ini. Aturlah gerakan mobil tersebut dari kanan ke kiri dan dari atas ke bawah sesuai dengan rambu lalu lintas yang sedang aktif pada masing-masing jalan.



TES FORMATIF

1. Jelaskan langkah-langkah membuat library komponen baru (sebagai contoh : vertical progress bar) pada toolbox!
2. Jelaskan mengapa ketika properties “left” komponen PictureBox kita beri operasi “-“ arah gerakan komponen tersebut bergerak ke kiri dan ketika kita beri operasi “+” maka arah gerakan menjadi ke kanan!
3. Apa yang anda ketahui tentang HMI (*Human Machine Interface*)?, apakah program visualisasi dengan menggunakan Visual Basic 2010 bisa di katakan bagian dari HMI? Jelaskan!

Jawab :

KEGIATAN 4 : Format data, waktu dan tanggal pada Visual Basic 2010

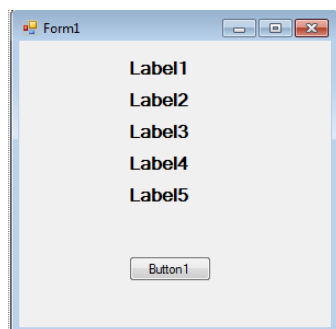
4.1 Menampilkan waktu dan tanggal Menggunakan Format Standar

Di Visual Basic 2010 tanggal dan waktu dapat dibuat menggunakan format standar dan format yang dibuat oleh pengguna. Format standar dari tanggal dan waktu adalah :

Format	Penjelasan
Format (Now, "General Date")	Format tanggal dan waktu
Format (Now, "Long Date")	Menampilkan tanggal dalam format yang panjang
Format (Now, "Short Date")	Menampilkan tanggal dalam format yang pendek
Format (Now, "Long Time")	Menampilkan waktu dalam format yang panjang.
Format (Now, "Short Time")	Menampilkan waktu dalam format yang pendek

Selain "General Date", Anda juga dapat menggunakan format yang disingkat "G" yaitu Format (Now, "G"). Dan untuk "Long Time" Anda dapat menggunakan format yang disingkat "T". Untuk "Short Time" Anda dapat menggunakan format yang disingkat "t".

Contoh :



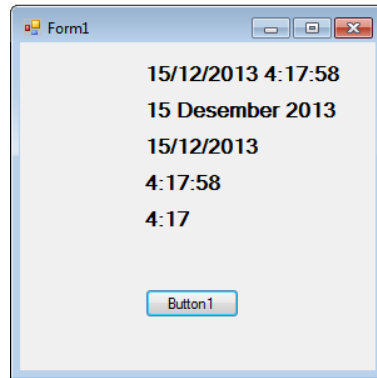
```
Private Sub Button1_Click (ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
```

```
Label1.Text = Format(Now, "General Date")
```



```

Label2.Text = Format(Now, "Long Date")
Label3.Text = Format(Now, "Short Date")
Label4.Text = Format(Now, "Long Time")
Label5.Text = Format(Now, "Short Time")
End Sub
    
```



Gambar 4.1 tampilan aplikasi format waktu

4.2 Format Tanggal dan Waktu menggunakan Format dari Pengguna

Selain menggunakan format standar, Anda juga dapat menggunakan format yang ditentukan oleh pengguna. Sintak umum dari format tanggal dan waktu tersebut adalah :

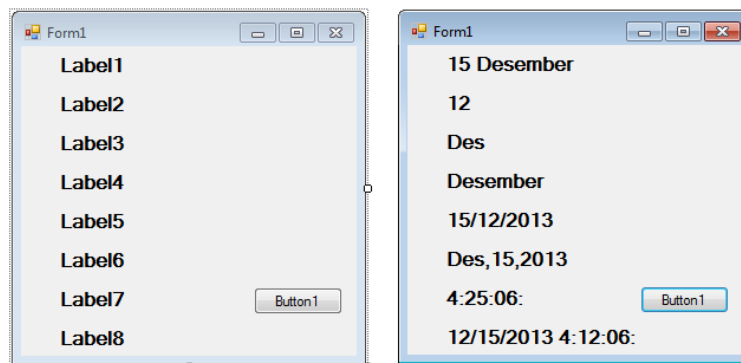
Format	Penjelasan
Format (Now, "M")	Menampilkan bulan dan tanggal
Format (Now, "MM")	Menampilkan bulan dengan digit ganda.
Format (Now, "MMM")	Menampilkan nama singkatan dari bulan.
Format (Now, "MMMM")	Menampilkan nama dari bulan.
Format (Now, "dd/MM/yyyy")	Menampilkan tanggal dalam hari / bulan / tahun.
Format (Now, "MMM,D,yyyy")	Menampilkan tanggal dalam Format Bulan, Hari, Tahun

Format (Now, "h:mm:ss:tt")	Menampilkan waktu dalam jam : menit : detik : dan menampilkan am/pm.
Format (Now, "MM/dd/yyyy h:mm:ss")	Menampilkan tanggal dan waktu dalam format jam : menit : detik.

Contoh :

```
Private Sub Button1_Click (ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click, Button2.Click, Button3.Click
```

```
Label1.Text = Format(Now, "M")
Label2.Text = Format(Now, "MM")
Label3.Text = Format(Now, "MMM")
Label4.Text = Format(Now, "MMMM")
Label5.Text = Format(Now, "dd/MM/yyyy")
Label6.Text = Format(Now, "MMM, d, yyyy")
Label7.Text = Format(Now, "h:mm:ss:tt")
Label8.Text = Format(Now, "MM/dd/yyyy H:MM:ss:tt")
End Sub
```



Gambar 4.2 Form tampilan aplikasi format waktu

4.3 Konsep Manipulasi Data String Pada Visual Basic 2010

Proses manipulasi data dalam Visual Basic cukup penting untuk diketahui, karena hal ini akan memudahkan kita dalam pemrosesan data yang berupa karakter (string). Terlebih lagi ketika kita melakukan pemrograman dengan data dengan jumlah karakter yang panjang khususnya ketika kita akan mempelajari komunikasi serial multi input-multi output pada bab-bab selanjutnya.

Pada sub bab ini siswa akan mempelajari dan memahami bagaimana cara melakukan manipulasi string dengan menggunakan Visual Basic.

Pada umumnya proses manipulasi string pada visual basic menggunakan dua macam operator, yaitu dengan tanda “+” dan “&”. Perhatikan contoh berikut :

```
Dim Teks1, Teks2, Teks3 as String
Teks1 = "Teknik"
Teks2 = "Pemrograman"
Teks3 = Teks1 + Teks2
Label1.Text = Teks3
```

Potongan program diatas , kode `Teks1 + Teks2` dapat digantikan dengan kode `Teks1 & Teks2`. Hasil dari eksekusi program tersebut akan menghasilkan tulisan string teks pada label1 **“Teknik Pemrograman”**.

Untuk melakukan manipulasi variabel dengan tipe data yang berbeda, misalkan data bertipe String akan digabungkan dengan data bertipe Integer, maka kode programnya adalah sebagai berikut :

```
Dim Teks1, Teks3 as String
Dim Teks2 as Integer
Teks1 = "Teknik Pemrograman"
Teks2 = 2
```

```
Teks3 = Teks1 + Teks2
```

```
Label1.Text = Teks3
```

Yang perlu diperhatikan adalah, ketika menuliskan kode berupa huruf, maka harus diawali dan diakhiri dengan tanda kutip (“”).

4.4 Fungsi mid, right, left, trim, Ucase, Lcase dan Chr

Fungsi manipulasi data string selanjutnya melakukan fungsi pemotongan terhadap suatu data bertipe string. Proses ini sangat penting dan bermanfaat terutama bagi komunikasi serial baik itu dari komputer ke komputer, mikrokontroler ke mikrokontroler maupun komputer ke mikrokontroler.

Pemanfaatan umum fungsi aplikasi ini adalah bagaimana kita membaca sejumlah sensor (multi sensor) dan memilah-milah data mana yang merupakan protokol, data mana milik sensor 1, data mana milik sensor2, sensor 3 dan seterusnya.

Perhatikan potongan program berikut ini ;

```
Dim Teks1 As String
```

```
Teks1 = "Teknik Pemrograman"
```

```
Label1.Text = Left(Teks1,3)
```

Kode program `Left(Teks1,3)` diatas memiliki arti bahwa program melakukan pemotongan data string yang tersimpan di variabel **Teks1** dari ujung kiri sebanyak tiga karakter. Sehingga eksekusi program di atas akan menghasilkan tampilan label1 menjadi **"Tek"** (tiga karakter = tiga huruf).

Tambahkan potongan kode berikut pada form tempat program di atas berada :

```
Label2.Text = Right(Teks1,4)
```

Kode program `Right(Teks1,4)` diatas memiliki arti bahwa program melakukan pemotongan data string yang tersimpan di variabel **Teks1** dari ujung paling kanan sebanyak empat karakter. Sehingga eksekusi program

di atas akan menghasilkan tampilan label1 menjadi “**aman**” (empat karakter = empat huruf terakhir).

Tambahkan potongan kode berikut pada form tempat program di atas berada :

```
Teks2 = " Teknik Pemrograman " → terdapat spasi di awal
```

```
Label2.Text = Trim(Teks2)
```

Kode program `Label2.Text = Trim(Teks2)` diatas akan menghilangkan spasi kosong di awal dan akhir suatu data string yang tersimpan di variabel **Teks2**. Eksekusi program di atas akan menghasilkan tampilan label2 menjadi “**Teknik Pemrograman**” (tanpa spasi di awal dan di akhir kalimat).

Tambahkan potongan kode berikut pada form tempat program di atas berada :

```
Label1.Text = Ucase(Teks1)
```

Kode program diatas akan mengubah data string yang tersimpan di variabel **Teks2** menjadi huruf besar semua. Hasil program di atas akan menghasilkan tampilan label1 menjadi “**TEKNIK PEMROGRAMAN**”.

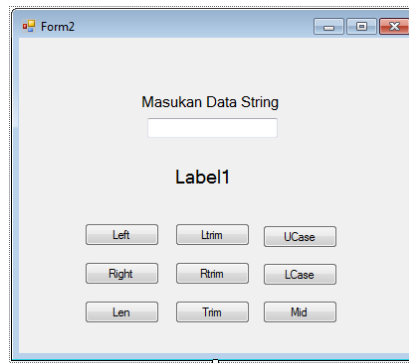
```
Label1.Text = Mid(Teks1,3,6)
```

Fungsi program `mid(Teks1,3,6)` diatas memiliki arti bahwa program melakukan pemotongan data string yang tersimpan di variabel **Teks1**. Dimulai dari karakter ke tiga ujung kiri sebanyak enam karakter. Sehingga eksekusi program di atas akan menghasilkan tampilan label1 menjadi “**knik P**” (enam karakter = lima huruf dan satu karakter spasi).

TUGAS

Pelajari fungsi pengolahan data string pada VB berikut :

- Left
- Right
- Trim
- Ltrim
- Rtrim
- Ucase
- Lcase
- Mid
- Len



Buatlah suatu program aplikasi dimana data string yang hendak diproses merupakan suatu isian dari suatu TextBox, sehingga tulisan kalimat yang hendak dimanipulasi oleh program dapat di ubah-ubah berdasar input dari pengguna.

TES FORMATIF

1. Jelaskan manfaat dan aplikasi dari pengolahan data string!
2. Jelaskan mengapa format tanggal dan waktu pada visual basic dapat menampilkan keterangan waktu secara otomatis dan dari mana sumber asal pengaturan waktu tersebut di dapatkan!
3. Jelaskan bagaimana cara menyatukan dua buah variabel berbeda tipe menjadi satu!

Jawab :

KEGIATAN BELAJAR 4

Menerapkan Program Aplikasi Dengan Bahasa Pemrograman *Visual Basic* Untuk Keperluan Input/Output Pada Port USB/Serial Pada Komputer

A. Tujuan Pembelajaran

1. Memahami saluran input (port input) pada komputer/laptop dari Port Serial dan USB.
2. Memahami saluran output (port output) pada Port serial dan USB.
3. Membuat program aplikasi sederhana dengan bahasa *Visual Basic* untuk keperluan input/output pada komputer/laptop.

B. Uraian Materi

1. Pengertian Serial Port / dan USB.
2. Pemrograman aplikasi pada Input/Output Komputer dengan bahasa *Visual Basic*.
3. Penerapan program aplikasi dengan bahasa *Visual Basic* untuk eksekusi Input/Output melalui Serial Port/USB yang divisualisasikan dengan menggunakan bantuan alat deretan LED (Light Emitting Diode).

C. Alokasi Waktu

24 jam pelajaran

D. Metode Pembelajaran

Teori dan Praktek

E. Media pembelajaran

- PC/Notebook
- Windows 7
- Visual Basic 2010

Teknik Pemrograman

KEGIATAN 1 : Akses Data String Komunikasi Serial Pada Visual Basic 2010

Personal Computer (PC) memiliki arsitektur yang terbuka sehingga memungkinkan penggunaan komputer menjadi sangat luas untuk berbagai keperluan dan pengolahan data baik data internal maupun data eksternal.

Pada masa lalu peralatan input output tersebut berupa input output card yang dipasang pada slot-slot ISA bus yang terdapat pada motherboard komputer. Pada masa sekarang produk komputer tidak lagi dilengkapi dengan slot-slot ekspansi ISA bus melainkan dengan slot-slot PCI yang jarak antar pin-pin sangat dekat sehingga tidak memungkinkan untuk membuat input output card sendiri dengan menyolder komponen dalam bentuk Surface Mount device yang kecil menggunakan tangan dan solder konvensional, melainkan harus menggunakan mesin solder SMD khusus dan canggih.

Seiring dengan perkembangan teknologi, kecenderungan pemakaian komputer jenis Laptop semakin meningkat dibandingkan dengan pemakaian personal computer (PC) dan peralatan input output card tidak mungkin lagi dipasang pada laptop. Oleh sebab itu memenuhi kebutuhan peralatan input output yang dapat dipergunakan pada komputer dan laptop maka peralatan input output harus berada diluar sistim komputer (external) dan model penyambungan datanya dapat menyesuaikan dengan komunikasi port yang tersedia pada kebanyakan komputer misalnya RS232 atau USB.

Kelebihan dari sistim peralatan input output external ini adalah kepraktisan pemakaiannya tidak perlu membuka cover computer untuk memasang card melainkan cukup dengan menghubungkan peralatan input output ini ke port RS232 atau USB computer.



Gambar 1.1 Aplikasi Visual Basic terhubung I/O

Pada gambar di atas merupakan contoh peralatan analog-digital input output yang menggunakan program aplikasi visual basic via komunikasi USB-Serial RS-232. Spesifikasi peralatan tersebut dapat di lihat dari keterangan berikut :

Input Digital : PORTG, 8 saluran

Output Digital : PORTA, 8 saluran

Input Analog

Jumlah Channel : 8 Channel s

Resolusi ADC : 8 bits

Tegangan input : DC 0 s.d. + 5,10 Volt DC

Output Analog

Jumlah Channel : 2 Channels

Resolusi ADC : 8 bits

Tegangan output : -10 s.d. + 10 Volt DC

Arus maksimal : 2 A

Output Power Supply

Tegangan Output : - 12 Volt DC Fixed, 2 A

+ 12 Volt DC Fixed, 2 A

+ 5 Volt DC Fixed, 2 A

GND

Tegangan Line : 220 Volt AC 50 Hz.

Komunikasi Data : RS232

Processor : MC68HC11F1

Parameter Komunikasi

Transmission rate : 4800 baud

Character coding : 8 bit ASCII

Parity : None

Stop Bits : 1

1.1 Mengakses Port Serial Pada Visual Basic 2010

Pada umumnya suatu komputer (PC) menyediakan dua macam saluran serial yaitu Standard Communication Port RS232 (COM) dan Universal Serial Bus (USB). Akan tetapi perkembangan dewasa ini laptop keluaran terbaru sudah jarang dilengkapi dengan port RS-232, karena trend yang ada di masyarakat menginginkan ukuran laptop yang tipis dan ringan.

Meskipun laptop keluaran terbaru tidak dilengkapi dengan port RS-232, akan tetapi kita dapat menggunakan alat bantu yang disebut konverter USB to RS-232. Dengan alat ini kita dapat mengakses port RS-232 via USB yang ada pada laptop kita.

Perkembangan teknologi juga memungkinkan laptop untuk langsung berinteraksi dengan mikrokontroler cukup dengan menggunakan RJ-45 pada mikrokontroler terhubung langsung dengan USB pada laptop. Hal ini dapat dilakukan karena ada rangkaian konverter TTL to USB pada minimum sistem mikrokontroler.

COM Port RS232 pada komputer biasanya berupa socket DB9 male seperti tampak pada Gambar di bawah ini :

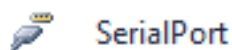


Gambar 1.2 COM Port RS232

Susunan pin DB-9 pada RS-232 adalah sebagai berikut

Nomor Pin	Signal
1	DCD Data Carrier Detect
2	RxD Received Data
3	TxD Transmitted Data
4	DTR Data Terminal Ready
5	GND Signal Ground
6	DSR Data Set Ready
7	RTS Request To Send
8	CTS Clear To Send
9	RI Ring Indicator

Untuk mengakses Port serial, Visual Basic 2010 menyediakan komponen kontrol yaitu **SerialPort** yang terdapat pada toolbox dengan icon seperti gambar di bawah ini.



Komponen **SerialPort** menyediakan fasilitas komunikasi serial yang meliputi pengiriman dan penerimaan data melalui port serial dengan berbagai properti sebagai berikut :

PortName

Dipergunakan untuk memilih jalur COM, misalnya COM1, COM2, COM3 dan seterusnya. Portname harus sudah disetting terlebih dahulu sebelum membuka port komunikasi pada program.

Contoh pemakaian :

```
SerialPort1.PortName = 1
```

DTREnable

Menentukan apakah dimungkinkan jalur Data Terminal Ready (DTR) selama komunikasi. Sinyal DTR biasanya dikirim oleh komputer ke modem untuk menunjukkan bahwa komputer telah siap untuk menerima datangnya transmisi.

Contoh pemakaian :

```
SerialPort1.DTREnable = True
```

EOFEnable

Menentukan apakah isyarat End Of File dimungkinkan

Contoh pemakaian :

```
SerialPort1.EOFEnable = False
```

Handshaking

Handshaking mengacu pada protocol komunikasi internal, yang mana data ditransfer dari port perangkat keras ke buffer penerima. Manakala suatu karakter data tiba di port serial, alat komunikasi harus memindahkannya ke dalam buffer penerima sedemikian rupa sehingga program dapat membaca data yang datang tersebut.

Jika tidak ada buffer penerima dan program diharapkan untuk membaca setiap karakter yang datang secara langsung dari perangkat keras, mungkin data akan hilang disebabkan karakter datang dengan cepat.

Protokol handshaking menjamin data tidak hilang selama buffer bekerja. Jika data datang pada port dengan cepat, maka alat komunikasi akan memindahkan data ke dalam buffer penerima.

Data dalam bilangan integer.

Pilihan untuk protocol handshaking adalah :

- 0 = comNone (Default) No handshaking.
- 1 = comXOn/Xoff XON/XOFF handshaking.
- 2 = comRTS RTS/CTS (Request To Send / Clear To Send) handshaking
- 3 = comRTSXOnXOff (Both Request To Send and XON/XOFF handshaking.)

Contoh pemakaian :

```
SerialPort1.Handshaking = False
```

InBufferSize

Menentukan besarnya kapasitas memory buffer penerima.

Contoh pemakaian :

```
SerialPort1.InBufferSize = 1024
```

InputLen

Membaca jumlah karakter yang masuk dalam buffer penerima

Contoh pemakaian :

```
Dim D1 As Integer
```

```
D1 = SerialPort1.InputLen
```

InputMode

Menentukan mode masukan

0 = cominputModeText

1 = cominputModeBinary

Contoh pemakaian :

```
SerialPort1.InputMode = 0
```

NullDiscard

Menentukan pengosongan buffer masukan

Contoh pemakaian :

```
SerialPort1.NullDiscard = False
```

OutBufferSize

Menentukan besarnya kapasitas memory buffer pengirim

Contoh pemakaian :

```
SerialPort1.OutBufferSize = 512
```

ParityReplace

Membaca apakah parity berubah

RThreshold

Menetapkan dan mngembalikan banyaknya karakter ke penerima sebelum kontrol MSCOMM menetapkan property CommEvent ke comEvReceive dan menghasilkan even OnComm.

Setting berupa bilangan integer yang menetapkan banyaknya karakter yang akan diterima sebelum menghasilkan event OnComm.

Contoh pemakaian :

```
SerialPort1.RThreshold = 0
```

RTSEnable

Menentukan apakah memungkinkan jalur Request To Send (RTS). Biasanya sinyal RTS yang minta ijin untuk mengirim data dikirim ke komputer oleh modem.

Contoh pemakaian :

```
SerialPort1.RTSEnable = False
```

Settings

Dipergunakan untuk mengatur parameter baud rate, parity, data bit dan stop bit dengan format string sebagai berikut :

“B,P,D,S” yang mana :

B = baud rate

P = parity

D = jumlah data bit

S = jumlah stop bit

Contoh pemakaian :

```
SerialPort1.Setting = "9600,N,8,1"
```


SThreshold

Setting berupa bilangan integer yang menetapkan banyaknya karakter yang akan dikirim sebelum menghasilkan event OnComm.

Contoh pemakaian :

```
SerialPort1.SThreshold = 0
```

Input

Dipergunakan untuk membaca masukan yang diterima

Contoh pemakaian :

```
Dim D1 As String
D1 = SerialPort1.Input
```

Output

Dipergunakan untuk mengirim data keluaran

Contoh pemakaian :

```
SerialPort1.Output = "Hallo"
```

1.2 Program Aplikasi Serial Tx-Rx PC to PC

Berikut ini adalah contoh program yang berfungsi untuk mengirim dan menerima data melalui port serial RS232. Data string teks akan kita tuliskan pada RichTextBox1 **"Input"** kemudian ketika tombol **"Write"** maka teks string yang ada di RichTextBox1 akan dikirimkan secara serial melalui Serialport1.

Untuk mencoba program ini, sambungkanlah kabel RS232 dengan stecker DB9. Kemudian hubung singkatlah pin nomor 2 (TxD) dan pin nomor 3 (RxD).

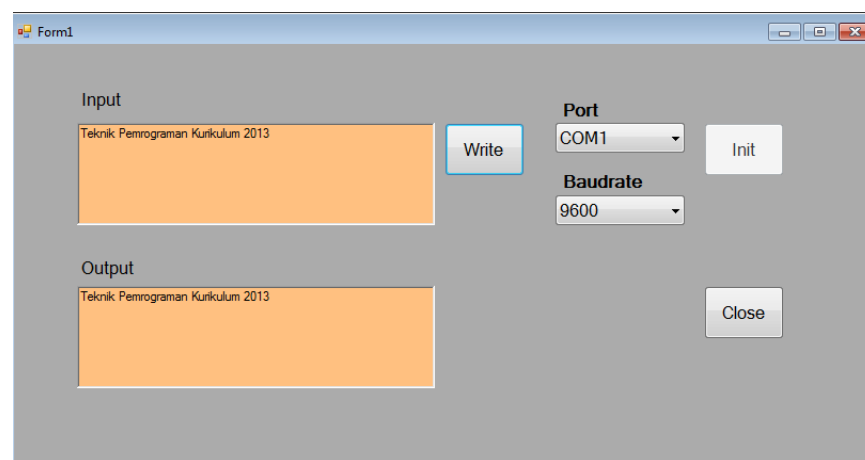


Jika Tombol **"Write"** ditekan maka text pada RichTextBox1 akan keluar dari pin Txt dan melalui kabel diterima pin RxD kemudian hasilnya ditampilkan

pada RichTextBox2. Jika tidak ada gangguan maka text yang diterima harus sama seperti teks yang dikirim.

Selanjutnya, coba lepas kabel hubung singkat tersebut dan coba lakukan pengiriman teks yang lain. Hasilnya richTextBox2 tidak akan muncul text baru.

Hasil jalannya program kirim terima teks RX-TX ini tampak seperti pada Gambar dibawah ini.

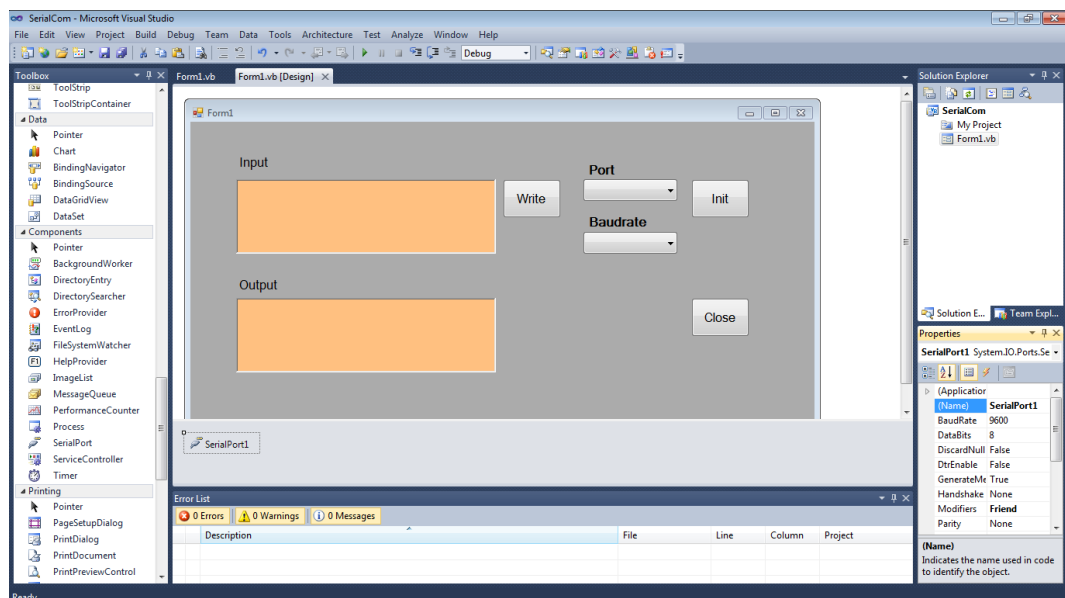


Gambar 1.3 tampilan aplikasi komunikasi serial

Berikut langkah-langkah pembuatan program komunikasi serial RS-232 dengan menggunakan Visual Basic :

1. Buatlah project baru dengan mengarahkan kursor pada menu File, klik New Project dan buatlah Form baru dengan nama **Serial_PC_to_PC**.
2. Sesuaikan ukuran form hingga tampak kurang lebih seperti gambar di atas.
3. Tambahkan komponen **RichTextBox1** dan **RichTextBox2** pada form sebagai tempat pengiriman dan penerimaan data string .
4. Tambahkan komponen **Button1**, **Button2** dan **Button3** dan ubah properti text ketiganya menjadi sebagai berikut : Button1 = "Init", Button2 = "Write" dan Button3 = "End".

5. Tambahkan komponen **ComboBox1** dan **ComboBox2** sebagai tempat memilih port dan menentukan kecepatan pengiriman data atau baudrates yang digunakan.
6. Berikan komponen **Label1**, **Label2**, **Label3** dan **Label4** sebagai tanda informasi setiap komponen guna memudahkan pengguna dalam menggunakan program. Aturlah properti text komponen label tersebut seperti berikut : Label1 = "Input", Label2 = "Output", Label3 = "Port" dan Label4 = "Baudrates".
7. Tambahkan komponen **SerialPort1** yang terletak pada toolbox bagian *component*, klik dua kali icon tersebut dan otomatis komponen SerialPort1 akan berada di bawah dari jendela form design seperti gambar berikut.



Gambar 1.4 Form design aplikasi komunikasi serial

8. Secara umum semua komponen yang kita butuhkan sudah tersedia pada form design, sedangkan untuk detail setting komponen properties dapat di lihat dari tabel berikut:

Obyek	Properti	Pengaturan
RichTextBox1	Name BackColor BorderStyle	RichTextBox1 255; 192; 128 Fixed3D
RichTextBox2	Name BackColor BorderStyle	RichTextBox2 255; 192; 128 Fixed3D
Button1	Name Text	Button1 "Init"
Button2	Name Text	Button2 "Write"
Button3	Name Text	Button3 "End"
ComboBox1	Name FormatingEnable	ComboBox1 True
ComboBox2	Name Item	ComboBox2 9600 38400 57600 115200
Label1	Name Text	Label1 Input
Label2	Name Text	Label2 Output
Label3	Name Text FontStyle, Size	Label3 Port Bold, 12
Label4	Name Text FontStyle, Size	Label4 Baudrates Bold, 12
Form1	Text	"Serial_PC_to_PC"

Berikut script program secara keseluruhan :

```
Imports System
Imports System.Threading
Imports System.IO.Ports
Imports System.ComponentModel

Public Class Form1
    \=====
    Dim myPort As Array
    Delegate Sub SetTextCallback(ByVal [text] As String)
    'Ditambahkan untuk mencegah kesalahan threading selama
    'penerimaan data
    \=====
    Private Sub Form1_Load(sender As System.Object, e As
    System.EventArgs) Handles MyBase.Load

    myPort = IO.Ports.SerialPort.GetPortNames()
    ComboBox1.Items.AddRange(myPort)
    Button2.Enabled = False
    End Sub

    \=====
    Private Sub Button1_Click(sender As System.Object, e As
    System.EventArgs) Handles Button1.Click

    SerialPort1.PortName = ComboBox1.Text
    SerialPort1.BaudRate = ComboBox2.Text
    SerialPort1.Open()
    Button1.Enabled = False
    Button2.Enabled = True
    Button3.Enabled = True

    End Sub
```

```

Private Sub Button2_Click(sender As System.Object, e As
System.EventArgs) Handles Button2.Click

SerialPort1.Write(RichTextBox1.Text & vbCrLf)
End Sub

\=====
Private Sub Button4_Click(sender As System.Object, e As
System.EventArgs) Handles Button4.Click

SerialPort1.Close()
Button1.Enabled = True
Button2.Enabled = False
Button3.Enabled = False
End Sub

\=====
Private Sub SerialPort1_DataReceived(sender As
System.Object, e As
System.IO.Ports.SerialDataReceivedEventArgs) Handles
SerialPort1.DataReceived

ReceivedText(SerialPort1.ReadExisting())
End Sub

\=====
Private Sub ReceivedText(ByVal [text] As String)

If Me.RichTextBox2.InvokeRequired Then
    Dim x As New SetTextCallback(AddressOf ReceivedText)
    Me.Invoke(x, New Object() {(text)})
Else
    Me.RichTextBox2.Text &= [text]
End If
End Sub
End Class
  
```

Catatan Khusus Program

Pada potongan program di atas terdapat suatu even atau kejadian dimana program akan di eksekusi ketika suatu even atau kejadian terjadi. Even atau kejadian yang dimaksud adalah proses ketika komputer menerima data serial pada port serial RX.

```

\=====
Private Sub SerialPort1_DataReceived(sender As
System.Object, e As
System.IO.Ports.SerialDataReceivedEventArgs) Handles
SerialPort1.DataReceived

```

ReceivedText (SerialPort1.ReadExisting())

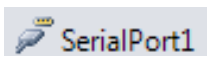
```
End Sub
```

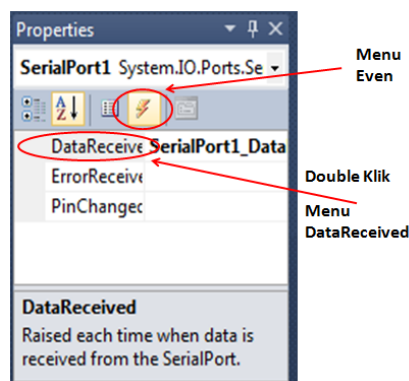
```

\=====

```

Script yang perlu di tulis hanyalah bagian yang ber-font tebal, sedangkan even kejadian akan muncul secara otomatis. Untuk mendapatkan even tersebut secara otomatis berikut langkah-langkahnya :

1. Bukalah form design program aplikasi dan arahkan kursor ke komponen SerialPort1 yang terletak dibawah jendela form design  .
2. Perhatikan jendela properties komponen SerialPort1, pilihlah menu even dan double klik kolom DataReceived.



3. Setelah double klik maka akan muncul jendela form editor dan anda dapat menuliskan program di dalamnya.

Pembahasan Program

Program Serial_PC_to_PC bertujuan untuk melakukan pengiriman data dari satu komputer ke komputer lain dengan menggunakan jalur komunikasi RS-232. Akan tetapi kita dapat memodifikasi dimana data yang dikirimkan dilewatkan kabel TX dari RS232 kemudian dimana seharusnya data tersebut di tangkap oleh kabel RX dari RS232 komputer lain, kita modifikasi jalur TX dari kabel RS232 komputer kita, dihubungkan secara langsung (*Jumper*) ke kabel RX RS232 komputer kita sendiri.

Aplikasi ini sangat berguna sebagai dasar mempelajari perintah pengiriman serial via RS232 dengan menggunakan Visual Basic. Disamping itu kita dapat melakukan monitoring program yang kita buat secara mandiri tanpa membutuhkan komputer lain. Artinya program ini penting sebagai aplikasi untuk mengetes apakah dasar aplikasi komunikasi serial yang kita bangun sudah berjalan dengan baik atau tidak? Jika sudah baik maka barulah kita dapat mengembangkan aplikasi kita dengan menambahkan program-program yang lain, contohnya menambahkan gambar, timer, grafik, data base, animasi dan lain-lain.

Program ini diawali dengan memanggil fungsi modul untuk mengaktifkan komponen dan komunikasi serial yang telah disediakan oleh Visual Basic 2010

```
Imports System
Imports System.Threading
Imports System.IO.Ports
Imports System.ComponentModel
```

Selanjutnya dengan mendeklarasikan variabel **myPort** sebagai array. Tujuannya adalah sebagai penampung port yang disediakan oleh komputer dan terdeteksi oleh program. Dinyatakan sebagai array karena variabel ini memungkinkan berisi lebih dari satu nilai (contoh : COM1, COM2, COM3 dst). Kemudian juga ditambahkan fungsi untuk mencegah kesalahan threading selama penerimaan data dengan menambahkan kode berikut


```
Delegate Sub SetTextCallback(ByVal [text] As String)
```

Langkah program berikutnya adalah *Form Load* kondisi awal program aplikasi atau default program ketika pertama kali aplikasi dijalankan. Pertama mendeteksi secara otomatis port serial yang telah aktif dan disediakan oleh komputer, kemudian hasil deteksi port serial tersebut di simpan pada variabel myPort.

```
myPort = IO.Ports.SerialPort.GetPortNames()
```

Setelah itu memasukan isi dari variabel myPort tersebut kedalam komponen ComboBox1, sehingga kita dapat memilihnya.

```
ComboBox1.Items.AddRange(myPort)
```

Disamping itu kita juga perlu menonaktifkan tombol "Write" sehingga pada tampilan tombol tersebut tidak dapat di akses.

```
Button2.Enabled = False
```

Prosedur jalannya program aplikasi, setelah kita memilih Port komunikasi maka kita juga menentukan kecepatan baudrate yang kita gunakan, maka kita menekan tombol **Init**. Ketika Tombol Init ditekan, maka program akan mengeksekusi perintah mengatur setting komunikasi serial baik itu port mana yang digunakan maupun berapa kecepatan baudrate yang dipakai serta selanjutnya mengaktifkan port serial tersebut.

```
SerialPort1.PortName = ComboBox1.Text
SerialPort1.BaudRate = ComboBox2.Text
SerialPort1.Open()      → pengaktifan port serial
Button2.Enabled = True → tombol "Write" aktif
Button3.Enabled = True
```

Selanjutnya tombol "**Write**" aktif dan dapat di akses. Sebelum menekan tombol "Write", kita harus menuliskan terlebih dahulu data string yang hendak kita kirimkan pada komponen RichTextBox1 "Input". Sebagai contoh ketikkan "Teknik Pemrograman Kurikulum 2013".

Ketika tombol “Write” ditekan, maka semua data yang terketik pada RichTextBox1 “Input” akan dikirimkan ke port serial dan di tambahkan karakter enter di akhir pengiriman. Kode program “& vbCr” merupakan kode untuk menambahkan karakter enter. Sedangkan kode program “SerialPort1.Write ” adalah perintah untuk mengirimkan data secara serial.

```
SerialPort1.Write(RichTextBox1.Text & vbCr)
```

Langkah berikutnya adalah bagaimana kita membuat program menangkap data serial yang dikirimkan oleh TX serial dari komputer dan dibaca oleh RX serial komputer kita dan ditampilkan ke RichTextbox2 “Output” pada program aplikasi. Untuk itu kita perlu membuat sub-program SerialPort1_DataReceived untuk membaca data yang diterima oleh RX-serial.

```
ReceivedText (SerialPort1.ReadExisting())
```

Potongan script program di atas bermakna, jika pada SerialPort1 terdapat data atau SerialPort1 menerima data kiriman dari luar, maka program akan memanggil sub program bagian ReceivedText. Berikut potongan program yang ada pada sub-program received text.

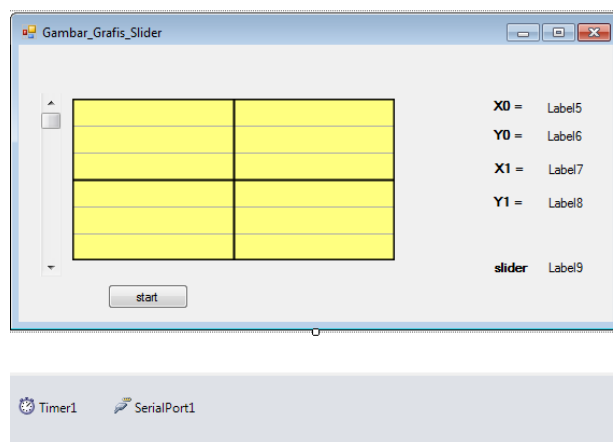
```
If Me.RichTextBox2.InvokeRequired Then
    Dim x As New SetTextCallback(AddressOf ReceivedText)
    Me.Invoke(x, New Object() {(text)})
Else
    Me.RichTextBox2.Text &= [text]
End If
```

Pada sub-program received text di atas, semua data yang diterima oleh port serial akan disimpan pada variabel [text]. Untuk menuliskan data serial tersebut kedalam RichTextBox, maka RichTextBox harus di isi data [text] seperti tertulis pada bagian akhir program di atas.

TUGAS

Buatlah project baru bernama "Serial_Grafik_TxRx", desainlah suatu aplikasi yang akan menggambarkan suatu Grafik, dimana nilai grafik tersebut didapat dari pengiriman nilai slider secara serial melalui kabel Tx-RS232 kemudian akan di terima oleh Rx-RS232 komputer dan selanjutnya baru digambarkan pada suatu grafik.

Berikut tampilan form design yang akan dibuat :



Rancanglah desain di atas dan gunakanlah metode drawline untuk menggambar grafik serta gunakan SerialPort1 sebagai fungsi komunikasi serial!

TES FORMATIF

4. Jelaskan secara singkat proses menangkap data serial pada port RX-serial Visual Basic 2010!
5. Sebutkan alasan mengapa dilakukan *jumper* pada pin nomer 2 dan nomer 3 pada konektor DB9 RS-232!
6. Gambar dan terangkan susunan pin DB-9 pada RS-232!

Jawab :

KEGIATAN 2 : Komunikasi Serial Komputer ke Mikrokontroller

Seperti halnya dijelaskan sebelumnya, bahwa perkembangan laptop saat ini mengarah ke desain yang ringan dan tipis. Bagi seorang engineer dan teknisi hal ini berarti harus mengorbankan sisi periperal input output. Sebagai contoh port paralel dan port RS-232 pada komputer dan laptop sudah tidak tersedia lagi.

Oleh sebab itu, untuk melakukan fungsi kontrol input-output *control engineering*, kita membutuhkan alat bantu mikrokontroller yang akan mengeksekusi program melakukan tugas-tugas kontrol input-output. Mikrokontroller tersebut dapat melakukan tugas secara mandiri maupun terkoneksi dengan komputer sebagai *master-master*, *master-slave* maupun *slave-master*.

Sebelum kita mempelajari bagaimana melakukan fungsi kontrol input-output antara mikrokontroller dengan komputer, maka terlebih dahulu kita mempelajari sekilas tentang mikrokontroller dan bagaimana cara mengkomunikasikan antara mikrokontroller dengan komputer.

Pada buku ini minimum system mikrokontroller yang digunakan adalah Arduino UNO R3 dengan menggunakan tipe mikrokontroller AVR 328. Pemilihan minimum sistem ini dikarena beberapa kelebihan yang ditawarkan. Beberapa kelebihan itu antara lain software yang disediakan open source artinya gratis dan selalu update terhadap perubahan operating system yang dikembangkan oleh windows. Kelebihan kedua banyak pengembang yang menggunakan produk ini sehingga memudahkan siswa untuk belajar dari contoh-contoh program aplikasi yang tersebar di dunia maya/internet. Selanjutnya kelebihan yang lain adalah komunikasi serial yang terdapat pada minimum system ini sudah menggunakan USB, sehingga kita tidak perlu lagi untuk membeli konverter USB to Serial.

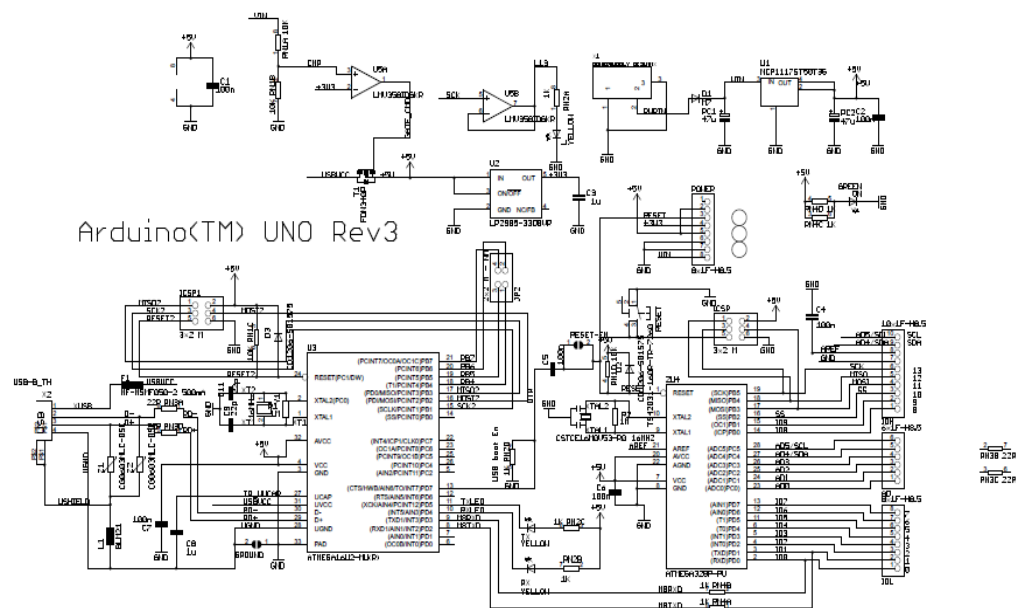
2.1 Minimum Sistem Arduino

Arduino merupakan sistem mikrokontroler yang relatif mudah dan cepat dipelajari dalam membuat aplikasi elektronika maupun robotika. Didalam Arduino terdapat sebuah komponen utama yaitu sebuah chip mikrokontroler dengan jenis AVR dari perusahaan Atmel.

Mikrokontroler itu sendiri adalah chip atau IC (integrated circuit) yang bertugas membaca dan memproses input dari sebuah rangkaian elektronik sehingga menghasilkan output sesuai dengan yang diinginkan. Mikrokontroler ini bisa diprogram menggunakan komputer.



Gambar 2.1 Minimum sistem Arduino Uno R3



Gambar 2.2 Skematik rangkaian minimum sistem arduino

Berikut adalah beberapa kelebihan dari Arduino :

- **Hardware dan Softwarena Open Source**

Dari sini kita bisa membuat tiruan board yang kompatibel dengan board Arduino tanpa harus membeli board asli buatan Itali dan kita juga tidak akan dianggap membajak selama kita tidak menggunakan *trade mark* "Arduino".

- **Fasilitas chip yang cukup lengkap**

Arduino menggunakan chip AVR ATmega 168/328 yang memiliki fasilitas PWM, komunikasi serial, ADC, timer, interupt, SPI dan I2C. Dengan fasilitas chip yang demikian, Arduino bisa digabungkan dengan modul atau alat lain walaupun protokol yang digunakan berbeda-beda.

- **Proses Upload tidak memerlukan chip programmer**

Chip pada Arduino sudah dilengkapi dengan bootloader yang akan menangani proses upload dari komputer. Dengan begitu kita tidak memerlukan chip programmer kecuali untuk menanamkan bootloader pada chip yang masih blank.

- **Ukuran board kecil**

Ukuran board Arduino yang kecil ini mudah di bawah kemana-mana atau dimasukkan ke dalam saku atau tas yang kecil.

- **Koneksi menggunakan Port USB**

Ini akan memudahkan kita jika menghubungkan Arduino ke PC atau laptop yang tidak memiliki port serial/paralel.

- **Bahasa pemrograman yang mudah**

Bahasa pemrograman Arduino adalah bahasa C yang sudah dipermudah menggunakan fungsi-fungsi yang sederhana sehingga dapat dipelajari dengan mudah.

- **Library gratis**

Library-library ini dapat di download gratis di website Arduino.

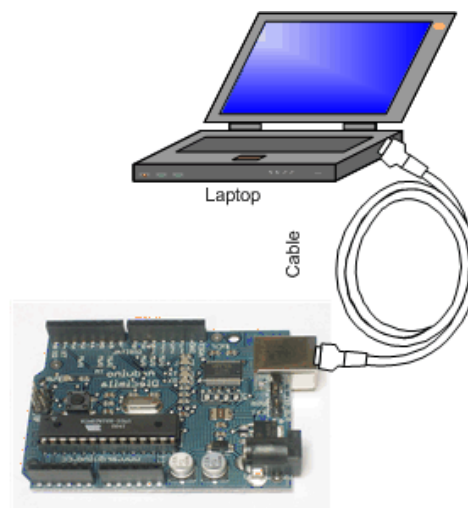
- **Pengembangan aplikasi lebih mudah**

Pengembangan aplikasi ini menjadi lebih mudah karena didukung oleh bahasa yang mudah dipelajari serta adanya library dasar yang lengkap.

2.2 Aplikasi Komunikasi Serial Mikrokontroler-Komputer dengan Visual Basic 2010

Aplikasi pertama yang akan kita buat adalah untuk mengetes komunikasi antara mikrokontroler dengan komputer melalui jalur serial. Mikrokontroler akan mengirimkan suatu data string kemudian akan dikirimkan melalui jalur TX-serial kemudian laptop akan menerima data tersebut melalui jalur RX-serial pada USB port.

Untuk itu kita membutuhkan dua kali pemrograman, yang pertama memprogram mikrokontroler untuk mengirimkan data yang kita inginkan (contoh : "Komunikasi Serial Mikro ke Komputer OK"). Untuk memprogram mikrokontroler, kita menggunakan software Arduino 1.0.5 dengan menggunakan bahasa pemrograman C (dapat di unduh pada halaman internet <http://www.arduino.cc/en/Main/Software> secara bebas).

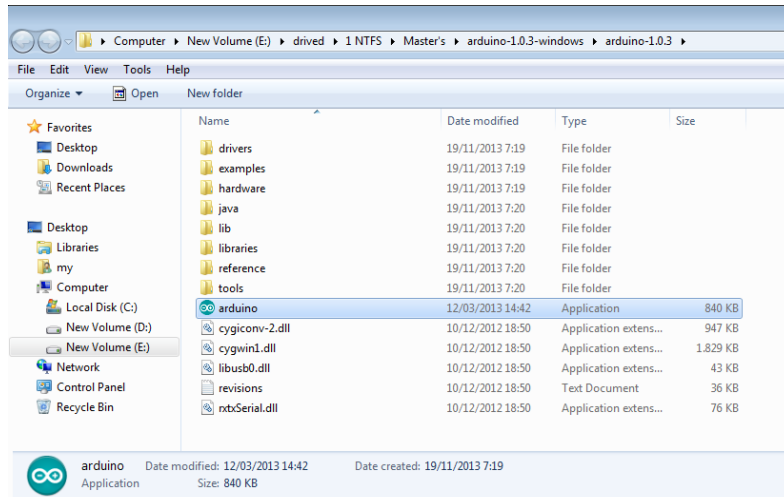


Gambar 2.3 Interkoneksi Arduino ke Laptop

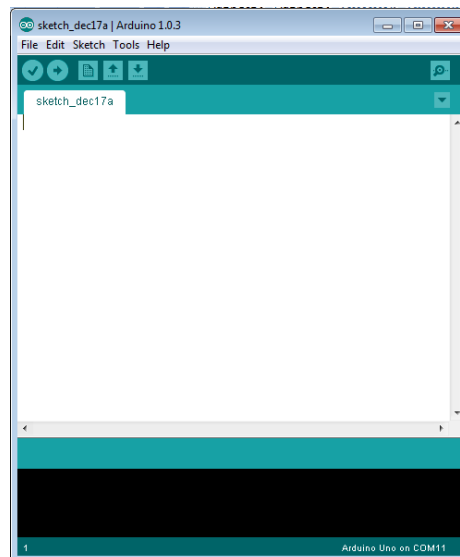
Langkah-langkah memprogram Arduino :

1. Bukalah file hasil download dari software Arduino 1.0.5 dan klik dua kali

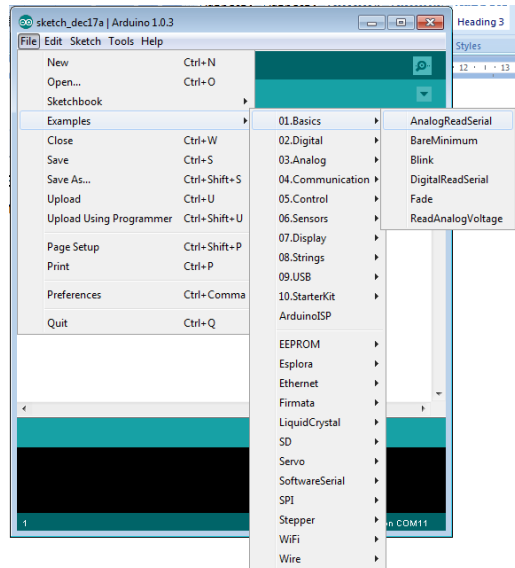
icon arduino  .



2. Setelah mengklik dua kali akan muncul jendela baru tempat kita menuliskan program mikrokontroller, jendela kerja ini disebut *sketch*.



3. Arahkan kursor pada menu **File**, pilih menu **Example**, kemudian arahkan ke menu **01.Basic** dan pilih menu **AnalogReadSerial**.



4. Setelah itu akan muncul jendela baru yang berisi sketch AnalogReadSerial. Terdapat banyak tulisan teks, tulisan awal merupakan suatu komentar atau keterangan dan tidak akan di eksekusi. Untuk menuliskan komentar multi baris maka komentar diawali dengan karakter “/*” dan diakhiri “*/”, sedangkan komentar hanya satu baris cukup diawali karakter “//”.

5. Program dimulai dengan program inisialisasi dan dituliskan pada fungsi void setup seperti dibawah ini. Inisialisasi yang diperlukan adalah menginisialisasi komunikasi serial dengan kecepatan baudrates 9600.

```
void setup() {
    // initialize serial communication at 9600 bits per
    second:
    Serial.begin(9600);
}
```

6. Selanjutnya program utama akan di letakan di fungsi void loop().

7. Perhatikan potongan program dibawah ini, pada program utama di deklarasikan sebuah variabel bernama **sensorValue** sebagai integer. Kemudian mikrokontroler diperintahkan membaca nilai analog dari ADC (*analog to Digital Converter*) channel 0 (A0). Hasil pembacaan tersebut akan

disimpan pada variabel `sensorValue` dan akan dikirimkan ke port serial.

```
// the loop routine runs over and over again forever:
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  // print out the value you read:
  Serial.println(sensorValue);
  delay(1);        // delay in between reads for stability
}
```

8. Karena data yang ingin kita kirimkan ke laptop bukanlah data dari ADC, melainkan suatu data teks string = "Teknik Pemrograman Kurikulum 2013". Untuk itu non aktifkan baris program pembacaan ADC dan pengiriman serial dengan memberi awalan karakter `"/"` serta tambahkan potongan program berikut :

```
Serial.println("Teknik Pemrograman Kurikulum 2013");
```

```
void loop() {
  // read the input on analog pin 0:
  //int sensorValue = analogRead(A0);
  // print out the value you read:
  //Serial.println(sensorValue);
  Serial.println("Teknik Pemrograman Kurikulum 2013");
  delay(100);        // delay in between reads for stability
}
```

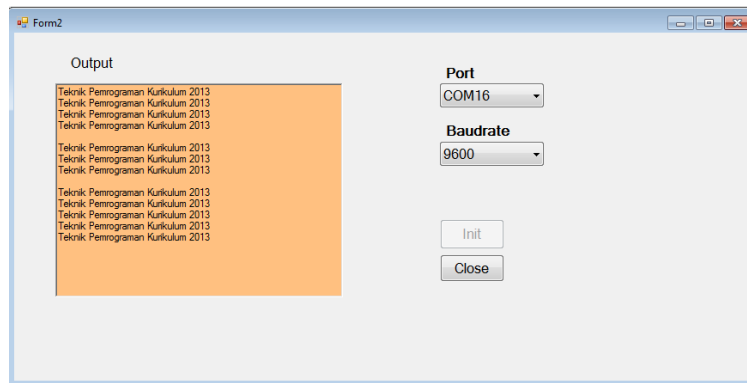
9. Program utama akan dieksekusi berulang-ulang, tambahkan nilai delay (100) untuk menunda proses perulangan program sehingga proses pengiriman tidak berlangsung terlalu cepat.
10. Langkah terakhir adalah meng-upload program tersebut ke dalam mikrokontroler. Untuk itu arahkan kursor ke ujung kiri atas dan klik icon

upload 



Langkah persiapan untuk menyiapkan mikrokontroler sudah selesai, sekarang kita akan membuat program visual basic yang akan menangkap

data serial yang telah dikirimkan oleh mikrokontroller tersebut. Dibawah ini merupakan tampilan program aplikasi Visual Basic 2010 yang dikembangkan dari program komunikasi serial_PC_to_PC yang telah kita buat pada kegiatan 1. Perbedaan dari program ini adalah kita tidak memerlukan tombol “Write” dan RichTextBox “input”.



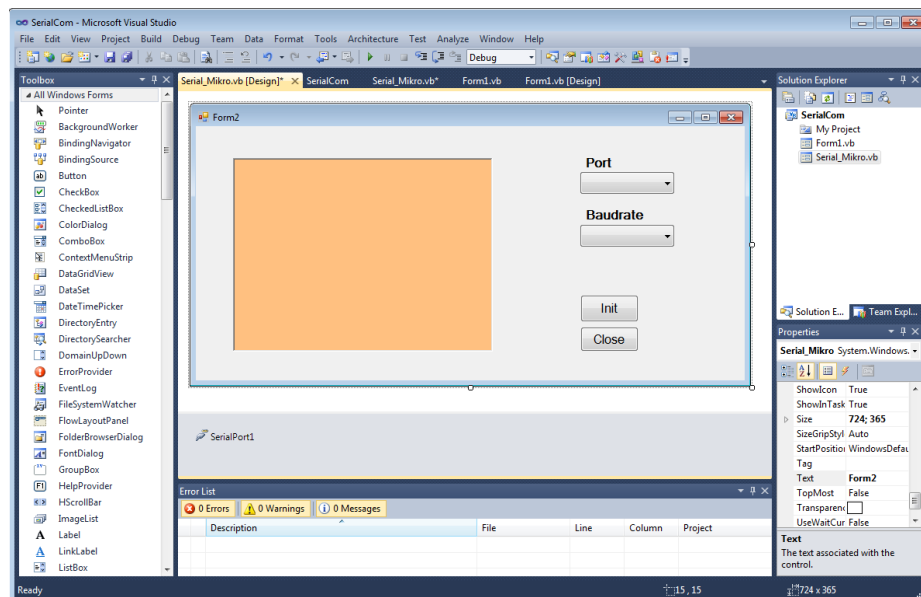
Gambar 2.3 tampilan hasil aplikasi komunikasi serial

Untuk membuat program aplikasi di atas ikutilah langkah kerja berikut :

1. Buatlah Form baru dengan mengarahkan kursor pada menu Project, pilihlah menu Add Window Form dan ubahlah nama Form baru tersebut dengan nama **Serial_Mikro**.
2. Sesuaikan ukuran form hingga tampak kurang lebih seperti gambar di atas.
3. Tambahkan komponen **RichTextBox1** pada form sebagai tempat penerimaan data string teks dari mikrokontroller.
4. Tambahkan komponen **Button1** dan **Button2** dan ubah properti text ketiganya menjadi sebagai berikut : Button1 = “Init” dan Button3 = “Close”.
5. Tambahkan komponen **ComboBox1** dan **ComboBox2** sebagai tempat memilih port dan menentukan kecepatan pengiriman data atau baudrates yang digunakan.

Teknik Pemrograman

6. Berikan komponen **Label1** dan **Label2** sebagai tanda informasi setiap komponen guna memudahkan pengguna dalam menggunakan program. Aturlah properti text komponen label tersebut seperti berikut : Label1 = "Port" dan Label2 = "Baudrates".
7. Tambahkan komponen **SerialPort1** yang terletak pada toolbox bagian *component*, klik dua kali icon tersebut dan otomatis komponen SerialPort1 akan berada di bawah dari jendela form design seperti gambar berikut.



Gambar 2.4 Form design program aplikasi Serial_Mikro

8. Secara umum semua komponen yang kita butuhkan sudah tersedia pada form design, sedangkan untuk detail setting komponen properties dapat di lihat dari tabel berikut:

Obyek	Properti	Pengaturan
RichTextBox1	Name BackColor BorderStyle	RichTextBox1 255; 192; 128 Fixed3D
Button1	Name Text	Button1 "Init"

Button2	Name Text	Button2 "Write"
ComboBox1	Name FormatingEnable	ComboBox1 True
ComboBox2	Name Item	ComboBox2 9600 38400 57600 115200
Label1	Name Text FontStyle, Size	Label1 Port Bold, 12
Label2	Name Text FontStyle, Size	Label2 Baudrates Bold, 12
Form1	Text	"Serial_Mikro"

Berikut script program secara keseluruhan :

```
Imports System
Imports System.Threading
Imports System.IO.Ports
Imports System.ComponentModel
\=====
Public Class Serial_Mikro
Dim myPort As Array
Delegate Sub SetTextCallback(ByVal [text] As String)
\=====
Private Sub Form2_Load(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MyBase.Load

myPort = IO.Ports.SerialPort.GetPortNames()
ComboBox1.Items.AddRange(myPort)
```

```
End Sub
```

```
Private Sub Button1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
```

```
SerialPort1.PortName = ComboBox1.Text
SerialPort1.BaudRate = ComboBox2.Text
SerialPort1.Open()
Button1.Enabled = False
```

```
End Sub
```

```
\=====
Private Sub Button4_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button4.Click
```

```
SerialPort1.Close()
Button1.Enabled = True
Button4.Enabled = False
```

```
End Sub
```

```
\=====
Private Sub SerialPort1_DataReceived(ByVal sender As
System.Object, ByVal e As
System.IO.Ports.SerialDataReceivedEventArgs) Handles
SerialPort1.DataReceived
```

```
ReceivedText (SerialPort1.ReadExisting())
```

```
End Sub
```

```
\=====
```

```
Private Sub ReceivedText(ByVal [text] As String) 'input
from ReadExisting

If Me.RichTextBox2.InvokeRequired Then
    Dim x As New SetTextCallback(AddressOf ReceivedText)
    Me.Invoke(x, New Object() {(text)})
Else
    Me.RichTextBox2.Text &= [text] 'append text
End If

End Sub

End Class
```


Teknik Pemrograman

TUGAS

Bagilah siswa dikelas menjadi beberapa kelompok yang terdiri atas lima orang dan diskusikan beberapa materi di bawah ini!

- Perbedaan pemrograman pada visual basic dengan pemrograman di mikrokontroller.
- Aplikasi komunikasi serial dalam kehidupan sehari-hari.
- Pemrograman bersifat *Case Sensitive*
- Fungsi mengatur setting port serial dalam komunikasi serial.
- Batasan-batasan komunikasi menggunakan RS-232

Diskusikan materi di atas dan carilah referensi sumber dengan membaca buku pendamping maupun sumber lain (contoh : Internet)!

TES FORMATIF

1. Sebutkan kelebihan dari minimum sistem arduino sehingga banyak digunakan dalam pembelajaran mikrokontroller oleh siswa di seluruh dunia!
2. Setting apa saja yang diperlukan dalam melakukan komunikasi secara serial?
3. Gambar dan terangkan susunan pin DB-9 pada RS-232!
4. Apa fungsi dari kode “`delay ()`” pada program mikrokontroller komunikasi serial dan bagaimana cara untuk merubah parameter yang ada di dalamnya?

Jawab :

KEGIATAN 3 : Aplikasi Visual Kontrol Input – Output VB 2010
3.1 Komunikasi Komputer-Mikrokontroler Fungsi Kontrol LED

Pembelajaran selanjutnya bertujuan membuat suatu program aplikasi Visual Basic yang dapat mengakses port output dari mikrokontroler untuk menyalakan dan mematikan lampu LED. Sumber perintah penyalakan dan pemadaman LED tersebut bukan berasal dari saklar secara fisik, akan tetapi berasal dari tombol yang berada di program Visual Basic.

Untuk merealisasikan tugas tersebut ada tiga komponen kerja yang perlu dikerjakan yaitu :

- a. Merangkai rangkaian elektronik.
- b. Membuat software program mikrokontroler.
- c. Membuat software program Visual Basic.

Alat dan Bahan :

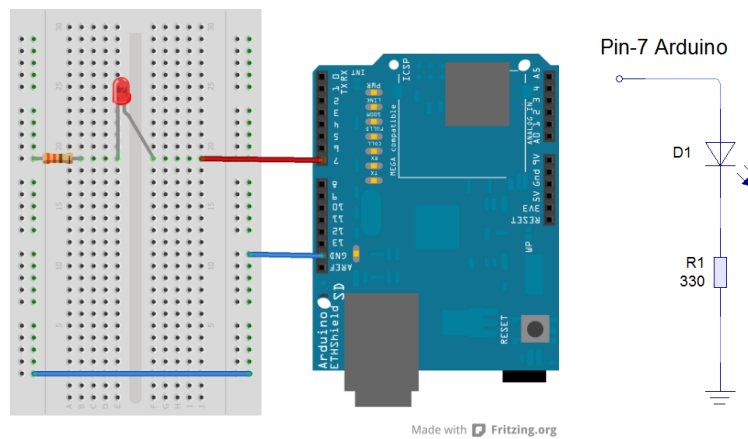
- Arduino Uno R3 1 unit
- LED 3mm 1 buah
- Resistor 330Ω 1 buah
- Kabel USB printer 1 buah
- Bread Board 1 buah
- Kabel

Media pembelajaran

- PC/Notebook
- Windows 7
- Visual Basic 2010
- Arduino 1.0.5

3.1.1 Rancangan Elektronik

Bacalah rangkaian skematik yang ada pada gambar dibawah ini kemudian bangun rangkaian sederhana tersebut pada sebuah bread board. Hubungkan rangkaian pada bread board tersebut dengan minimum sistem arduino seperti gambar di bawah ini.



Gambar 3.1 Rancangan hardware kontrol LED

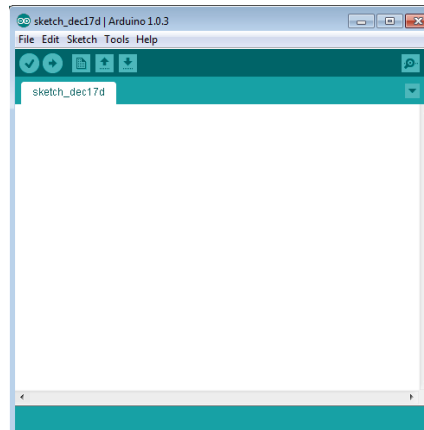
Pada gambar di atas lampu LED terhubung dengan pin 7 arduino (kabel warna merah) dan terpasang seri dengan resistor 330Ω. Kemudian resistor tersebut terhubung dengan pin GND dari arduino (kabel warna biru). Lampu LED akan menyala ketika pin 7 berlogika tinggi “1”(High).

Ketika pin-7 berlogika tinggi maka pada pin tersebut akan terukur tegangan sebesar 5 volt. Arus selalu mengalir dari potensial tinggi ke potensial rendah, artinya arus akan mengalir dari pin 7 melewati lampu LED kemudian melalui resistor 330Ω dan akhirnya menuju ground (GND) sehingga lampu LED akan menyala.

3.1.2 Software Program Mikrokontroller

Langkah-langkah memprogram Arduino :

1. Bukalah software Arduino 1.0.5 kemudian arahkan kursor pada menu **File**, pilih menu **New** sehingga muncul jendela sketch baru.



2. Tuliskan kode program berikut ke dalam sketch (ingat bahwa bahasa C menggunakan *case sensitive*, artinya perbedaan huruf besar kecil akan mempengaruhi berjalannya program).

```
char a;
int led=7;
void setup()
{
  pinMode(led,OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  if(Serial.available() !=0) {
    a=Serial.read();

    if(a=='1') {
      digitalWrite(led,HIGH);}

    if(a=='0') {
      digitalWrite(led,LOW);}
  }}
}
```

3. Program dimulai dengan mendeklarasikan variabel **"a"** bertipe char, dan variabel **"led"** yang berisi angka 7 sebagai integer.

```
char a;
int led=7;
```

4. Selanjutnya program inisialisasi dituliskan pada fungsi `void setup` seperti dibawah ini. Terdapat dua buah inisialisasi dalam sub program ini yaitu inisialisasi kecepatan baudrate serial dan inisialisasi variabel `led = pin-7` sebagai output.

```
void setup() {
    Serial.begin(9600);
    pinMode(led, OUTPUT);
}
```

5. Selanjutnya program utama akan di letakan di fungsi `void loop()`.
6. Pada program utama memberikan syarat jika terjadi koneksi serial maka perintah dibawah akan di eksekusi, jika tidak terjadi koneksi maka program tidak hanya akan menunggu koneksi dan tidak terjadi proses apapun.

```
if(Serial.available() !=0) {
```

7. Kode program berikutnya menunjukkan jika persyaratan di atas terpenuhi (terjadi koneksi port serial), maka perintah selanjutnya adalah membaca nilai yang masuk pada port serial (dari komputer ke mikrokontroller) dan menyimpannya pada variabel **"a"**.

```
a=Serial.read();
```

8. Langkah selanjutnya adalah menyeleksi data yang masuk ke dalam port serial mikrokontroller, jika data yang dikirimkan bernilai "1" maka LED akan menyala dan jika data yang dikirimkan bernilai "0" maka LED akan padam.

```
if(a=='1') {
    digitalWrite(led, HIGH); }

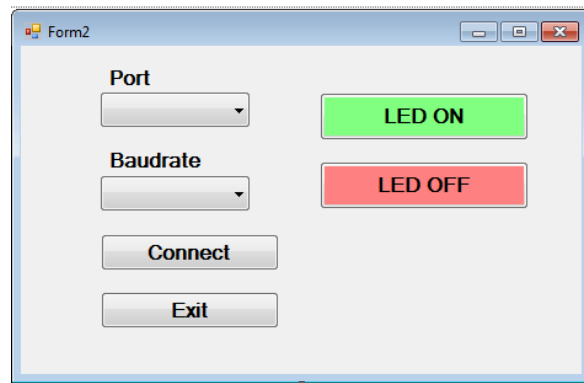
if(a=='0') {
    digitalWrite(led, LOW); }
```

9. Langkah terakhir adalah meng-upload program tersebut ke dalam mikrokontroller. Untuk itu arahkan kursor ke ujung kiri atas dan klik icon



3.1.2 Desain Program Visual Basic

Tahapan akhir kerja untuk menyelesaikan permasalahan fungsi kontrol LED dengan menggunakan menggunakan komputer adalah dengan membuat rancang bangun desain aplikasi visual basic. Dibawah ini merupakan tampilan akhir program aplikasi yang akan kita bangun.

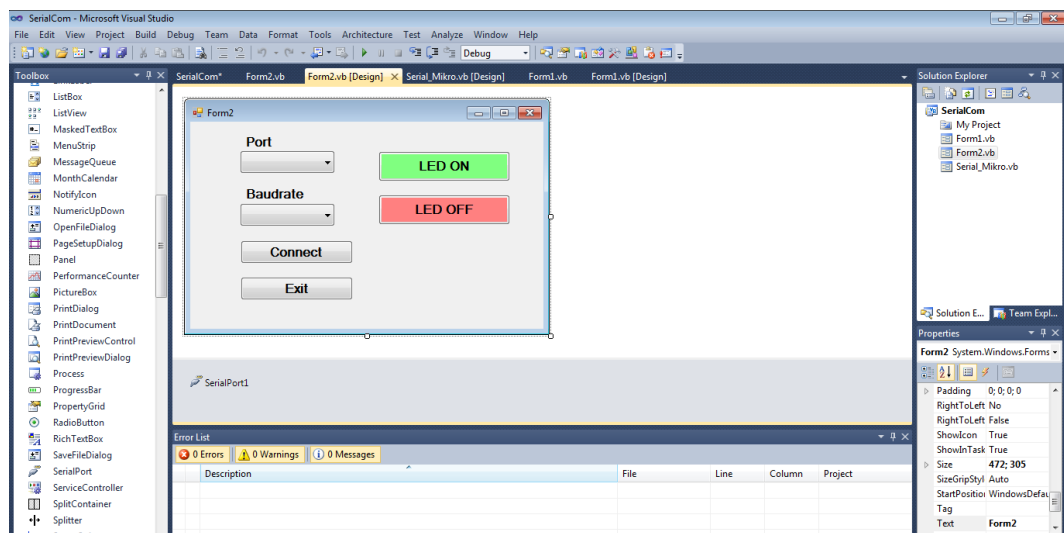


Gambar 3.2 Rancangan software kontrol LED

Untuk membuat program aplikasi di atas ikutilah langkah kerja berikut :

1. Buatlah Form baru dengan mengarahkan kursor pada menu Project, pilihlah menu Add Window Form dan ubahlah nama Form baru tersebut dengan nama **Serial_Mikro_LED**.
2. Sesuaikan ukuran form hingga tampak kurang lebih seperti gambar di atas.

3. Tambahkan komponen **Button1**, **Button2**, **Button3**, **Button4** dan ubah properti text komponen tersebut menjadi sebagai berikut : Button1 = “**LED ON**”, Button2 = “**LED OFF**”, Button3 = “**Connect**” dan Button4 = “**Exit**”.
4. Tambahkan komponen **ComboBox1** dan **ComboBox2** sebagai tempat memilih port dan menentukan kecepatan pengiriman data atau baudrates yang digunakan.
5. Berikan komponen **Label1** dan **Label2** sebagai tanda informasi setiap komponen guna memudahkan pengguna dalam menggunakan program. Aturlah properti text komponen label tersebut seperti berikut : Label1 = “**Port**” dan Label2 = “**Baudrates**”.
6. Tambahkan komponen **SerialPort1** yang terletak pada toolbox bagian *component*, klik dua kali icon tersebut dan otomatis komponen SerialPort1 akan berada di bawah dari jendela form design seperti gambar berikut.



7. Seluruh komponen yang kita butuhkan sudah tersedia pada form design, sedangkan untuk detail setting komponen properties dapat di lihat dari tabel berikut:

Obyek	Properti	Pengaturan
Button1	Name Text BackColor	Button1 "LED ON" 128; 255; 128
Button2	Name Text BackColor	Button2 "LED OFF" 255; 128; 128
Button3	Name Text	Button3 "Connect"
Button4	Name Text	Button4 "Exit"
ComboBox1	Name FormatingEnable	ComboBox1 True
ComboBox2	Name Item	ComboBox2 9600 38400 57600 115200
Label1	Name Text FontStyle, Size	Label1 Port Bold, 12
Label2	Name Text FontStyle, Size	Label2 Baudrates Bold, 12
Form1	Text	"Serial_Mikro"

Berikut script program secara keseluruhan :

```
Imports System
Imports System.Threading
Imports System.IO.Ports
Imports System.ComponentModel

Public Class Form2
  Dim myPort As Array
  Delegate Sub SetTextCallback(ByVal [text] As String)

  Private Sub Form2_Load(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles MyBase.Load

    myPort = IO.Ports.SerialPort.GetPortNames()
    ComboBox1.Items.AddRange(myPort)
    Button1.Enabled = False
    Button2.Enabled = False
  End Sub

  Private Sub Button3_Click(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    Button3.Click

    SerialPort1.PortName = ComboBox1.Text
    SerialPort1.BaudRate = ComboBox2.Text
    SerialPort1.Open()
    Button1.Enabled = True
    Button2.Enabled = True
    Button3.Enabled = False
  End Sub

  Private Sub Button4_Click(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    Button4.Click
```

```
SerialPort1.Close()  
End  
End Sub  
  
Private Sub Button1_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Button1.Click  
  
SerialPort1.Write("1")  
Button1.Enabled = False  
Button2.Enabled = True  
End Sub  
  
Private Sub Button2_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Button2.Click  
  
SerialPort1.Write("0")  
Button1.Enabled = True  
Button2.Enabled = False  
End Sub  
  
End Class
```

Pembahasan Program

Program **Serial_Mikro_LED** bertujuan untuk melakukan pengiriman data dari satu komputer ke Mikrokontroler dengan menggunakan jalur komunikasi RS-232 via USB. Data yang dikirimkan tersebut berisi perintah komputer pada mikrokontroler untuk menyalakan dan mematikan lampu LED.

Program ini diawali dengan memanggil fungsi modul untuk mengaktifkan komponen dan komunikasi serial yang telah disediakan oleh Visual Basic 2010

```
Imports System
Imports System.Threading
Imports System.IO.Ports
Imports System.ComponentModel
```

Selanjutnya dengan mendeklarasikan variabel **myPort** sebagai array. Tujuannya adalah sebagai penampung port yang disediakan oleh komputer dan terdeteksi oleh program. Dinyatakan sebagai array karena variabel ini memungkinkan berisi lebih dari satu nilai (contoh : COM1, COM2, COM3 dst). Kemudian juga ditambahkan fungsi untuk mencegah kesalahan threading selama penerimaan data dengan menambahkan kode berikut

```
Delegate Sub SetTextCallback(ByVal [text] As String)
```

Langkah program berikutnya adalah *Form Load* kondisi awal program aplikasi atau default program ketika pertama kali aplikasi dijalankan. Pertama mendeteksi secara otomatis port serial yang telah aktif dan disediakan oleh komputer, kemudian hasil deteksi port serial tersebut disimpan pada variabel myPort.

```
myPort = IO.Ports.SerialPort.GetPortNames()
```

Setelah itu memasukan isi dari variabel myPort tersebut kedalam komponen ComboBox1, sehingga kita dapat memilihnya.

```
ComboBox1.Items.AddRange(myPort)
```

Prosedur jalannya program aplikasi, setelah kita memilih Port komunikasi maka kita juga diminta untuk menentukan kecepatan baudrate yang akan kita gunakan. Setelah itu tekan tombol **Connect** untuk mengatur setting komunikasi serial baik itu port mana yang digunakan maupun berapa kecepatan baudrate yang dipakai serta selanjutnya mengaktifkan port serial tersebut.

```
SerialPort1.PortName = ComboBox1.Text
SerialPort1.BaudRate = ComboBox2.Text
SerialPort1.Open()      → pengaktifan port serial
```

Langkah selanjutnya adalah menhidupkan lampu LED yang terhubung dengan mikrokontroler. Jika tombol **“LED ON”** ditekan maka komputer akan mengirimkan karakter **“1”** yang akan di olah oleh mikrokontroler untuk menhidupkan LED.

```
SerialPort1.Write("1")
```

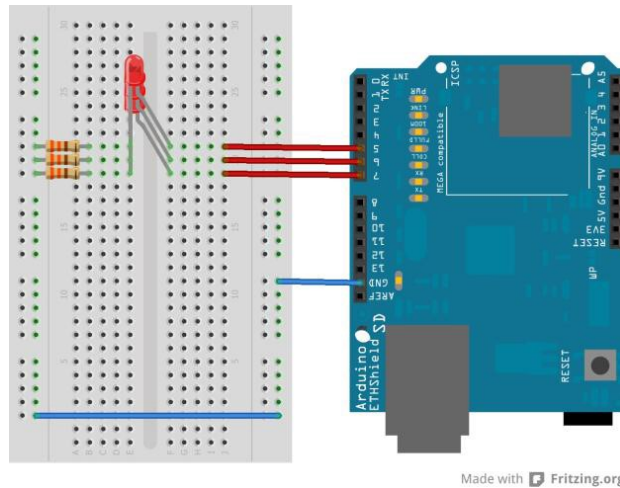
Ketika tombol **“LED OFF”** ditekan, ditekan maka komputer akan mengirimkan karakter **“0”** yang akan di olah oleh mikrokontroler untuk mematikan LED.

```
SerialPort1.Write("0")
```

Demikianlah proses alur program berjalan, program tersebut akan di eksekusi berulang-ulang hingga tombol **“Exit”** di tekan.

TUGAS

Buatlah suatu aplikasi program pada visual basic untuk menyalakan dan mematikan tiga buah lampu LED yang terhubung dengan mikrokontroller seperti ilustrasi gambar berikut :



Alat dan Bahan :

- Arduino Uno R3 1 unit
- LED 3mm 3 buah
- Resistor 330Ω 3 buah
- Kabel USB printer 1 buah
- Bread Board 1 buah
- Kabel

Software :

- Visual Basic 2010
- Arduino 1.0.5

No	Komponen	Arduino
1	LED 1	Pin-7
2	LED 2	Pin-6
3	LED 3	Pin-5
4	Ground	Pin-Gnd

TES FORMATIF

1. Gambarkan skematik rangkaian untuk mengaktifkan lampu LED dengan menggunakan port I/O dari mikrokontroler!
2. Apa fungsi resistor 330 Ω pada rangkaian tersebut?
3. Rancanglah rangkaian skematik pengaktifan LED, dimana LED akan aktif jika mikrokontroler memberikan output pada pin sebesar 0 volt (LOW)!
4. Mengapa komponen SerialPort1 pada contoh program **Serial_Mikro_LED** mengirimkan karakter "1" dan "0"? Apakah bisa karakter tersebut kita ganti dengan karakter lain seperti "A", "B", "Lampu hidup", "Lampu mati" dll? Jelaskan!

Jawab :

3.2 Visualisasi Grafik ADC (Analog to Digital Converter)



Script Program Mikrokontroler :

```
int tegangan;

void setup() {
    Serial.begin(9600);
}

void loop() {
    int sensorValue = analogRead(A0);
    tegangan=sensorValue*0.1466;
    Serial.println(tegangan);
    delay(100);
}
```

Script Program Visual Basic :

```
Imports System
Imports System.Threading
Imports System.IO.Ports
Imports System.ComponentModel

Public Class Serial_ADC
    Dim l, x0, y0, x1, y1, x2, y2, gain As Integer
    Dim myPen As Pen
    Dim myPort As Array
    Delegate Sub SetTextCallback(ByVal [text] As String)

    Private Sub Serial_ADC_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

        myPort = IO.Ports.SerialPort.GetPortNames()
        ComboBox1.Items.AddRange(myPort)

        x0 = 0
        y0 = 150

        x1 = x0
        y1 = y0

    End Sub

    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer1.Tick
        Dim Grafik As Graphics = Panell1.CreateGraphics
        myPen = New Pen(Drawing.Color.Red, 2)
        x1 = x1 + 1
        'y1 = VScrollBar1.Value
        'Grafik.DrawLine(myPen, x0, y0, x1, y1)

        Label5.Text = x0
        Label6.Text = y0
        Label7.Text = x1
        Label8.Text = y1

        If x1 >= 300 Then
            Me.Refresh()
            x1 = 0
        End If

        y0 = y1
        x0 = x1

    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        'Timer1.Enabled = True
    End Sub

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
        SerialPort1.PortName = ComboBox1.Text
        SerialPort1.BaudRate = ComboBox2.Text
    End Sub
End Class
```

```

SerialPort1.Open()
Button1.Enabled = False
Button3.Enabled = False

End Sub

Private Sub Button4_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button4.Click
    SerialPort1.Close()
End
End Sub

Private Sub SerialPort1_DataReceived(ByVal sender As
System.Object, ByVal e As
System.IO.Ports.SerialDataReceivedEventArgs) Handles
SerialPort1.DataReceived
    ReceivedText(SerialPort1.ReadExisting())
End Sub

Private Sub ReceivedText(ByVal [text] As String) 'input from
ReadExisting
    If Me.RichTextBox2.InvokeRequired Then
        Dim x As New SetTextCallback(AddressOf ReceivedText)
        Me.Invoke(x, New Object() {(text)})
    Else
        Me.RichTextBox2.Text &= [text] 'append text

        Dim Grafik As Graphics = Panel1.CreateGraphics
        myPen = New Pen(Drawing.Color.Red, 2)
        x1 = x1 + 10
        y1 = 150 - Val([text])
        Grafik.DrawLine(myPen, x0, y0, x1, y1)

        Label5.Text = x0
        Label6.Text = y0
        Label7.Text = x1
        Label8.Text = y1

        If x1 >= 300 Then
            Me.Refresh()
            x1 = 0
        End If

        y0 = y1
        x0 = x1

    End If
End Sub

End Class

```

Daftar Pustaka

Halvorson, Michael. “ *Visual Basic 2010 Step By Step* ”. Microsoft Press. Redmond, Washington, 2010.

Bradley, J.C. Millspaugh, A.C. “*Programming In Visual Basic 2010*”. McGraw-Hill press. New York, America, 2011.

<http://www.vbtutor.net/index.php/visual-basic-2010-tutorial/>

<http://www.arduino.cc/en/Main/HomePage>

Putranto, A. “*Teknik Otomasi Industri Untuk Sekolah Menengah Kejuruan*”. Direktorat Pembinaan Sekolah Menengah dan Kejuruan. Jakarta, 2008.

