

MODUL PRAKTIKUM

NETWORK PROGRAMMING II



DISUSUN OLEH:

Sardiarinto

TEKNOLOGI KOMPUTER
UNIVERSITAS BINA SARANA INFORMATIKA
2018

KATA PENGANTAR

Segala puji syukur kepada Tuhan Yang Maha Esa karena berkat rahmatNya penulisan modul Mata Kuliah Network Programming II dapat terselesaikan dengan baik. Modul ini disusun untuk memenuhi kebutuhan mahasiswa dalam mata Network Programming II yang disajikan dalam bentuk praktikum dan diharapkan dapat membekali mahasiswa dalam memahami bahasa pemrograman Java lanjutan.

Dengan konsep database dan networking, diharapkan mahasiswa dapat mengimplementasikannya dalam bentuk final project sebagai prototype hasil karya dan syarat kelulusan mata kuliah Network Programming II. Teknik penyajiannya dilakukan secara terpadu dan sistematis.

Seperti layaknya sebuah modul, maka pembahasan dimulai dengan menjelaskan target pembelajaran yang hendak dicapai. Dengan demikian pengguna modul ini secara mandiri dapat mengukur tingkat ketuntasan yang dicapainya.

Penulis menyadari sepenuhnya bahwa modul ini tentu memiliki banyak kekurangan. Untuk itu penulis dengan lapang dada menerima masukan dan kritik yang konstruktif dari berbagai pihak demi kesempurnaannya di masa yang akan datang. Semoga modul ini dapat bermanfaat bagi para penggunanya.

Penulis

Sardiarinto

DAFTAR ISI

KATA PENGANTAR	2
DAFTAR ISI	3
PERTEMUAN 1 GUI	4
PERTEMUAN II SYSTEM NET FRAMEWORK.....	14
PERTEMUAN III REGISTRY	26
PERTEMUAN IV WINDOWS MANAGEMENT INSTRUMENTATION	36
PERTEMUAN V MANIPULASI JARINGAN	50
PERTEMUAN VI PENGENALAN SOCKET	57
PERTEMUAN VII TUGAS & EVALUASI.....	67
PERTEMUAN VIII PRESENTASI.....	68
PERTEMUAN IX MULTIPLEXING SOCKET	69
PERTEMUAN X SOCKET PROGRAMMING.....	78
PERTEMUAN XI FTP (FILE TRANSFER PROTOCOL)	89
PERTEMUAN XII-XVI TUGAS KELOMPOK & PRESSENTASI.....	94



Lembar Kerja Praktik Network Programming II

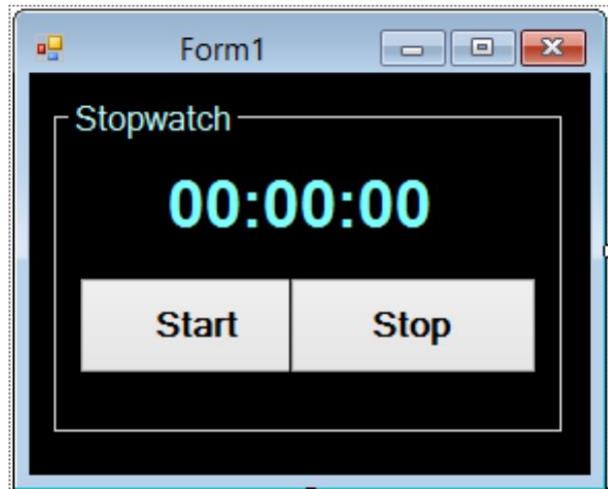
PERTEMUAN 1 GUI

1.1. Refresh Ulang Mengenai Form dan Pembuatan Stopwatch

Pada pembelajaran sebelumnya telah diperkenalkan bagaimana cara membuat aplikasi dengan GUI pada Microsoft Visual Studio 2010. Dalam C# sebuah form terbentuk dalam sebuah class, sebenarnya memiliki dua buah class. Contoh dimisalkan terdapat sebuah form dengan nama “form1.cs” maka form1.cs memiliki dua class yaitu “form1.cs” dan “form1.cs[design]”. ketika kita menggunakan tampilan desain (GUI) maka class yang digunakan adalah class utama yaitu “form1.cs[design]”, sedangkan semua hal yang berhubungan dengan kode dalam desain class tersebut terletak pada “form1.cs”.

Mari kita ambil contoh, Buatlah sebuah form dan berikan dua buah button beserta satu label di tengah.

a) Aplikasi Stopwatch



Adapun Komponennya adalah:

Objek	Properties
Groupbox 1	Text = Stopwatch
Label 1	Text = 00:00:00
Button 1	Text = Start
Button 2	Text = Stop
Timer 1	Enable = True

Kemudian dengan dua buah tombol tersebut kita akan menjalankan timer yang akan ditunjukkan pada label di atas.

Jika **Start** di tekan, maka Output Timer akan berjalan dengan di tunjukan pada label, dan jika **Stop** ditekan Output Timer akan berhenti. Berikut adalah kode programnya:

Sebenarnya Microsoft Visual Studio 2010 sudah menyediakan kelas Stopwatch dalam frameworknya, untuk memanggil fasilitas Stopwatch yang sudah disediakan tambahkan code berikut pada baris program.

```
└─ using System;
   └─ using System.Collections.Generic;
   └─ using System.ComponentModel;
   └─ using System.Data;
   └─ using System.Drawing;
   └─ using System.Linq;
   └─ using System.Text;
   └─ using System.Windows.Forms;
   └─ //tambahkan
   └─ using System.Diagnostics;
```

Kemudian buat sebuah variable yang bertipe data Stopwatch untuk menyimpan nilai dari stopwatch. Tipe data Stopwatch ini sebenarnya hampir sama dengan tipe data Time, namun tipe data Stopwatch bersifat continue dan dapat menampung nilai berupa waktu ter pendek (milisecond). Buatlah sebuah variable dengan code program sebagai berikut:

```
//tambahkan
using System.Diagnostics;

namespace Stopwatchproj
{
    public partial class Form1 : Form
    {
        private Stopwatch stopw = null;
        public Form1()
        {
            InitializeComponent();
        }
    }
}
```

Pada code program diatas menunjukkan bahwa variable stopw bertipe Stopwatch dan bersifat private yaitu hanya bisa digunakan dikelas tersebut. Pembuatan variable diluar main program (public Form1()) disebut dengan variable global, sebab variable tersebut dideklarasikan diluar main program dan dapat digunakan untuk object apa saja.

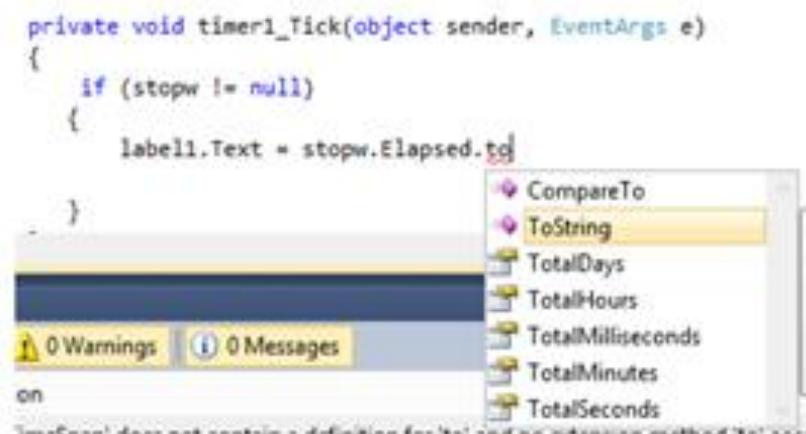
Pemberian nilai stopw = null untuk menandakan bahwa stopw bukanlah object dari kelas Stopwatch, tetapi hanya sebuah variable.

Setelah merancang tampilan program (*design*) double klik pada object Timer1,

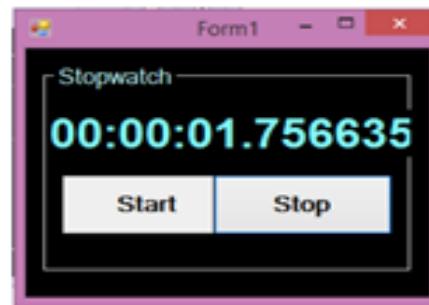
```
private void timer1_Tick(object sender, EventArgs e)
{
    if (stopw != null)
    {
        label1.Text = stopw.Elapsed.ToString(@"hh\:mm\:ss");
    }
}
```

kemudian tambahkan code program berikut ini:

Code program “Elapsed” digunakan untuk menghitung berapa banyak waktu yang dikeluarkan oleh stopwatch. Waktu yang telah dihitung bisa ditampilkan hanya jumlah milisecondnya saja “ElapsedMilliseconds”, atau bisa juga mengeluarkan jumlah menit, jumlah detik,jumlah jam, dan jumlah hari atau dalam bentuk string seperti tampilan berikut:



Untuk keluaran berupa string maka pilih “ToString”, kemudian karena ToString merupakan sebuah method/fungsi maka perlu menambahkan kurung buka dan kurung tutup “ToString()”. Format stopwatch yang dirubah kedalam string secara default adalah Jam : Menit : Detik : Milidetik seperti tampilan di samping:



Namun, tampilan tersebut dapat diatur menjadi Jam saja, menit saja, detik saja, atau jam:menit:detik sesuai dengan kebutuhan aplikasi. Cara mengatur tampilan stopwatch yaitu dengan menambahkan (@”hh:\mm:\dd”); pada method ToString()

Selanjutnya untuk dapat memulai penggunaan stopwatch beberapa code program dapat ditambahkan pada button_1 seperti berikut:

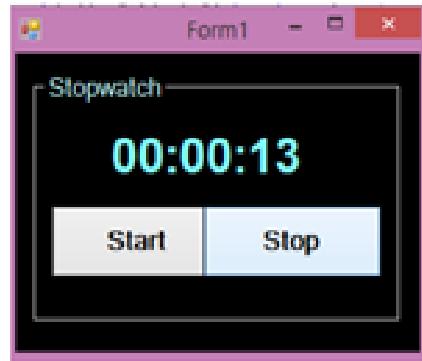
```
private void button1_Click_1(object sender, EventArgs e)
{
    stopw = new Stopwatch();
    stopw.Start();
    button1.Enabled = false;
}
```

Pertama-tama variable stopw yang bertipe data Stopwatch akan dijadikan object yang mewakili kelas Stopwatch. Pembuatan object yang diambil dari variable ini bertujuan agar object stopw dapat menjalankan semua sifat – sifat yang ada pada kelas Stopwatch seperti Start, Stop, dan Reset.

Selanjutnya untuk dapat menghentikan waktu yang telah dihitung secara konstant oleh stopwatch beberapa code program dapat ditambahkan pada button_2 sebagai berikut:

```
private void button2_Click_1(object sender, EventArgs e)
{
    stopw.Stop();
    button1.Enabled = true;
}
```

Aplikasi stopwatch telah selesai dibuat, untuk menjalankan program ini dapat dilakukan dengan menekan (*click*) tombol berwarna hijau (Start Debugging) atau dengan menekan F5



1.2. Manipulasi objek dan penggunaan percabangan pada Aplikasi Billing

Dari form yang sebelumnya pada pertemuan I, mari coba kita tambahkan satu buah textbox yang akan digunakan menjadi sebuah input Masukkan Nama untuk Log In pada Aplikasi Billing.

a) Aplikasi Billing

Buatlah sebuah Form dan berikan sebuah Label, sebuah TextBox dan satu buah Button. Seperti gambar dibawah ini:



Adapun Komponen sebagai berikut:

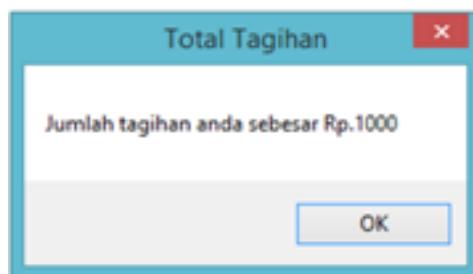
Object	Properties
Label 1	Text = Nama
Label 2	Text = 00:00:00 Visible = false
textbox 1	Text = “ ”
Button 1	Text = Log in
Timer 1	Enable = true

Kondisi

- Pertama kali program dijalankan label 2 tidak terlihat
- Isi nama pada textbox, jika nama tidak diisi maka akan keluar messagebox seperti berikut



- Setelah isi nama kemudian klik tombol log in, maka tombol akan berubah text menjadi Stop, dan label 2 akan muncul dan menghitung dari detik pertama
- Setiap 30 menit akan ditambahkan biaya sebesar 1000 rupiah
- Ketika di klik stop, maka akan muncul message box sebagai berikut.



Masukkan Coding pada program dengan cara Tekan **F7**.

Full Coding:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Diagnostics;

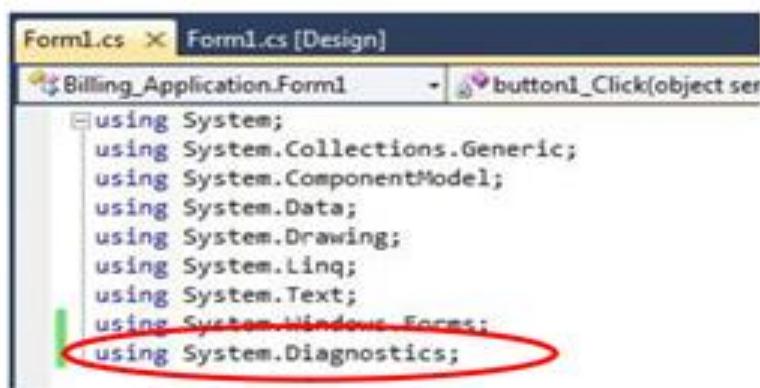
namespace Billing_Application
{
    public partial class Form1 : Form
    {
        private Stopwatch wkt = null;
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            if (textBox1.Text != "")
            {
                wkt = new Stopwatch();
                if (button1.Text == "Log In")
                {
                    label2.Visible = true;
                    wkt.Start();
                    button1.Text = "Stop";
                }
                else if (button1.Text == "Stop")
                {
                    wkt.Stop();
                    if (wkt.Elapsed.TotalMinutes <= 30.00)
                    {
                        MessageBox.Show("Jumlah tagihan anda sebesar Rp.1000", "Total Tagihan");
                    }
                    else if (wkt.Elapsed.TotalHours <= 1.00)
                    {
                        MessageBox.Show("Jumlah Tagihan anda sebesar Rp.2000", "Total Tagihan");
                    }
                    wkt.Reset();
                    label2.Visible = false;
                    button1.Text = "Log In";
                    textBox1.Text = null;
                }
            }
        }
    }
}
```

```
        }
        else if (textBox1.Text == "")
        {
            MessageBox.Show("Nama Harus diisi! ", "Important Message");
        }
    }
private void timer1_Tick(object sender, EventArgs e)
{
    if (wkt != null)
    {
        label2.Text = wkt.Elapsed.ToString(@"hh\:mm\:ss");
    }
}
}
```

Penjelasan Coding Billing Application

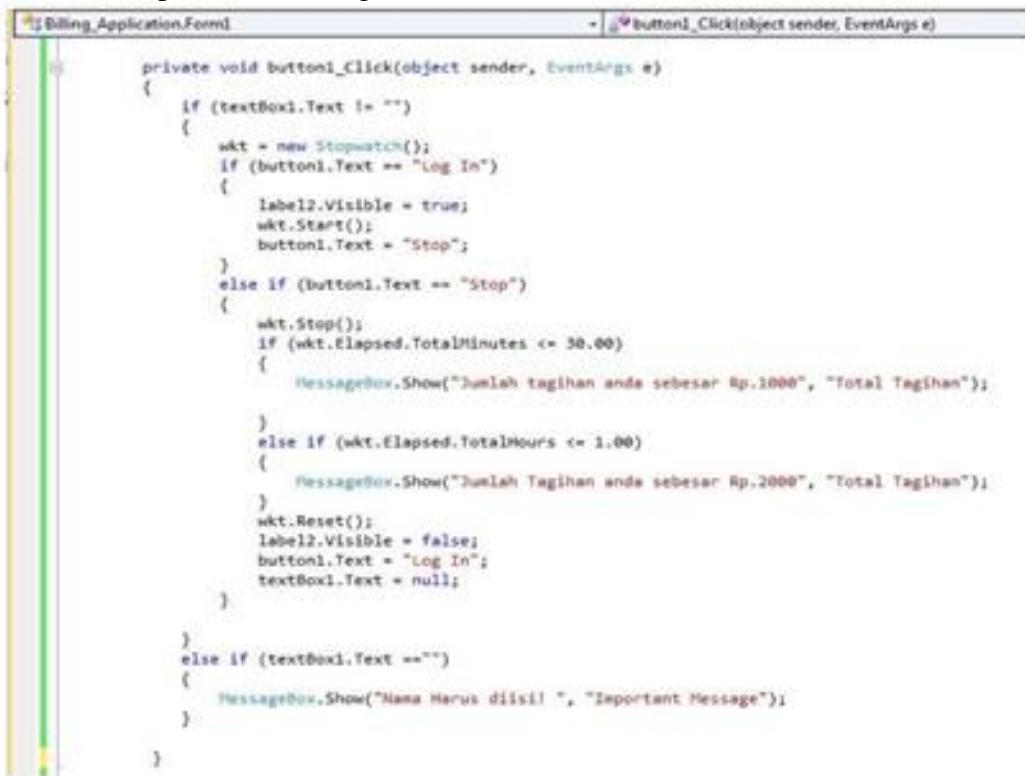
Penambahan Library



Deklarasi Variable waktu = Null

```
namespace Billing_Application
{
    public partial class Form1 : Form
    {
        private Stopwatch wkt = null;
        public Form1()
        {
            InitializeComponent();
        }
    }
}
```

Double klik pada Button Log In.



A screenshot of the Microsoft Visual Studio IDE. The title bar shows "Billing_Application.Form1" and "button1_Click(object sender, EventArgs e)". The main window displays the C# code for the button1_Click event handler. The code initializes a Stopwatch object named wkt, checks if the text box contains a name, and then toggles between "Log In" and "Stop" states. It also handles the total bill calculation based on time spent. If no name is provided, it shows an important message.

```
private void button1_Click(object sender, EventArgs e)
{
    if (textBox1.Text != "")
    {
        wkt = new Stopwatch();
        if (button1.Text == "Log In")
        {
            label2.Visible = true;
            wkt.Start();
            button1.Text = "Stop";
        }
        else if (button1.Text == "Stop")
        {
            wkt.Stop();
            if (wkt.Elapsed.TotalMinutes <= 30.00)
            {
                MessageBox.Show("Jumlah tagihan anda sebesar Rp.1000", "Total Tagihan");
            }
            else if (wkt.Elapsed.TotalHours <= 1.00)
            {
                MessageBox.Show("Jumlah Tagihan anda sebesar Rp.2000", "Total Tagihan");
            }
            wkt.Reset();
            label2.Visible = false;
            button1.Text = "Log In";
            textBox1.Text = null;
        }
    }
    else if (textBox1.Text == "")
    {
        MessageBox.Show("Nama Harus diisi!", "Important Message");
    }
}
```

Double klik pada Timer lalu isikan Coding Berikut:

```
private void timer1_Tick(object sender, EventArgs e)
{
    if (wkt != null)
    {
        label2.Text = wkt.Elapsed.ToString(@"hh\:mm\:ss");
    }
}
```

LATIHAN:

Buatlah Form Aplikasi Billing dengan penambahan Paket – paket seperti dibawah ini:

Ketentuan Paket:

Warnet Ceria		
Nama Paket	Waktu	Harga
Regular	1 jam	Rp. 2,000
Hemat	5 Jam	Rp, 9000
Game 1	2 Jam	Rp, 5000
Game 2	4 Jam	Rp, 9000

Keterangan: Boleh lihat dari pembahasan cuman di tambahkan paket – paket diatas.



Lembar Kerja Praktik Network Programming II

PERTEMUAN II SYSTEM NET FRAMEWORK

Framework .NET merupakan suatu komponen Windows yang terintegrasi dan dibuat agar dapat menjalankan berbagai macam aplikasi .NET, termasuk pengembangan aplikasi Web Service yang mempergunakan XML (extensible markup language). Beberapa bahasa pemrograman yang sudah cukup dikenal mengadopsi teknologi .NET ini antara lain, Visual Basic dan C++. Kemudian C# (C Sharp).

Fasilitas Framework .NET

1. Menyediakan tools dan teknologi untuk membangun aplikasi terdistribusi.
2. Semua bahasa pemrograman yang tergabung dalam teknologi .NET dapat saling berkomunikasi

Tujuan Framework .NET

- a) Untuk menyediakan environment kerja yang konsisten bagi bahasa pemrograman yang berorientasi objek (object-oriented programming - OOP).
- b) Untuk menyediakan lingkungan kerja di dalam mengeksekusi kode yang dapat meminimaliasi proses penyebaran software (software deployment) dan menghindari konflik penggunaan versi software yang di buat.
- c) Untuk menyediakan environment kerja yang aman dalam hal pengeksekusian kode.
- d) Untuk menyediakan lingkungan kerja yang dapat mengurangi masalah pada persoalan performa dari kode atau dari lingkungan interpreter-nya.
- e) Membuat para developer lebih mudah mengembangkan berbagai macam jenis aplikasi yang lebih bervariasi, seperti aplikasi berbasis windows dan aplikasi berbasis web.
- f) Membangun semua komunikasi yang ada di dalam standar industri untuk memastikan bahwa semua kode aplikasi yang berbasis Framework .NET dapat berintegrasi dengan berbagai macam kode aplikasi lain.
- g) Untuk memenuhi tujuan tersebut maka diciptakan berbagai macam bahasa pemrograman berbasis .NET.
- h) Dapat Digunakan VB, C#, J#, C++, Perl, ASD dan lain-lain.

Kelebihan Utama Framework .NET

- a. Mudah
Garbage Collector yang bertanggung jawab untuk mencari dan membuang objek yang sudah tidak terpakai secara otomatis.
- b. Efisien

Efisien dalam hal waktu pembuatan aplikasi atau juga efisien dalam hal lain, seperti biaya (cost).

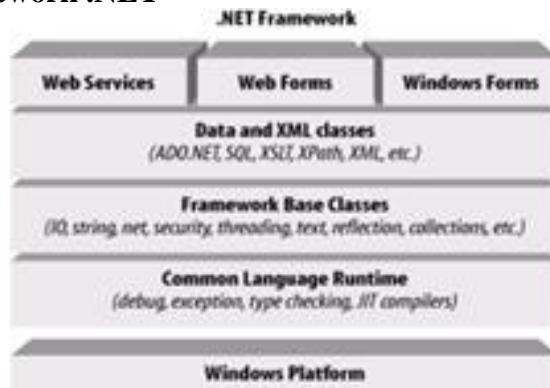
c. Konsisten

Dengan adanya BCL, maka kita bisa menggunakan objek atau Class yang dibuat untuk aplikasi berbasis windows pada aplikasi berbasis web.

d. Produktivitas

Produktivitas para developer, akan berdampak pada meningkatnya produktivitas suatu perusahaan.

Arsitektur Framework .NET



Komponen Utama Framework .NET

1. Common Language Runtime (CLR)

Konsep paling penting dalam .NET framework adalah keberaaan dan fungsionalitas .NET Common Language Runtime (CLR). CLR adalah sebuah layer atau lapisan framework yang berada pada sistem operasi dan menangani eksekusi dari aplikasi .NET. Program yang nantinya akan dibuat tidak terkomunikasi langsung dengan Sistem Operasi, tetapi melalui CLR terlebih dahulu.

2. NET Framework Class Library (FCL) atau Base Class Library (BCL).

FCL merupakan library standar besar yang dikemas dalam satu bahasa pemrograman. Bagian terbaik dari library ini adalah peggunaan teknik pemrograman object oriented yang membuat teknik pengaksesan dan penggunaannya menjadi sangat sederhana. Anda dapat menggunakan FCL pada program yang akan dibuat hanya seperti menggunakan kelas-kelas lainnya. Anda bahkan dapat menerapkan teknik inheritance dan polimorfisme pada kelas FCL ini

Tabel Info Framework

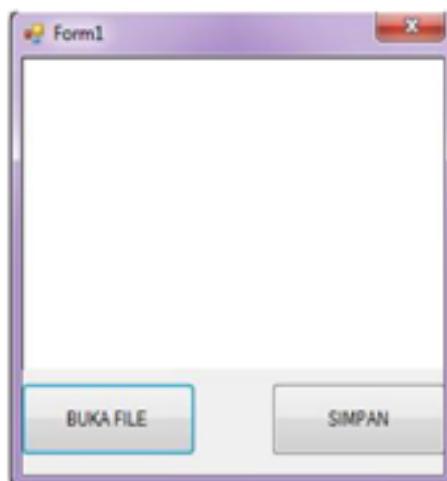
Microsoft.Win32	Class yang digunakan untuk menangani event yang berkaitan dengan OS dan class yang terdapat pada Registry
System	Dasar class .NET yang biasa digunakan untuk tipe data dan konversi nya
System.Collections	Digunakan untuk perintah <i>list, queues, bit arrays, dan string collection</i>

System.Data	Digunakan untuk penggunaan struktur database ADO.NET
System.Data.OleDb	Digunakan untuk penggunaan struktur database OLE DB .NET
System.Drawing	Penyedia akses pada fungsi dasar grafis
System.IO	Digunakan pada sistem <i>data streams</i> dan <i>files</i> (Input Output streams)
System.Management	Digunakan untuk mengakses infrastruktur <i>Windows Management Instrumentation</i> (WMI)
System.Net	Digunakan untuk mengakses fungsi jaringan Windows
System.Net.Sockets	Digunakan untuk mengakses antarmuka <i>Windows Sockets</i> (Winsock)
System.Runtime.Remoting	Digunakan untuk mengakses <i>Windows distributed computing platform</i>
System.Security	Digunakan untuk mengakses sistem keamanan pada CLR
System.Text	Class penyedia encoding ASCII, unicode, UTF-7, UTF-8
System.Threading	Digunakan untuk membuat program yang bersifat multi-threading
System.Timers	Memungkinkan pemanggilan sebuah <i>event</i> pada interval waktu tertentu

System.Web	Memungkinkan penggunaan fungsi server dan web
System.Web.Mail	Digunakan untuk mengirim pesan email
System.Windows.Forms	Class yang dapat digunakan untuk pembuatan aplikasi dasar windows dengan menggunakan <i>standard Windows graphical interface</i>
System.XML	Digunakan untuk menyediakan layanan pemrosesan tipe file XML

2.1. Aplikasi Simple notepad

Aplikasi ini di buat untuk membuka file yang berextension .txt



Adapun komponen nya adalah:

NO	Component Name	Text	Name
1	RichText Box	-	rtinput
2	Button 1	Buka File	bbukafile
3	Button 2	Simpan	bsimpan

Coding Project:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using Microsoft.Win32; //library tambahan
using System.IO;//dalam program windows 32

namespace SimpleNotepad
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            this.StartPosition = FormStartPosition.CenterScreen; // posisi form berada di tengah,
            this.FormBorderStyle = FormBorderStyle.FixedSingle; // form tidak bisa dibesar / dikecilkan,
            this.MaximizeBox = false; //dan menghilangkan tombol maximize dan minimize.
            this.MinimizeBox = false;
        }

        void bersih() //fungsi untuk membersihkan richtext input
        {
            rtinput.Text = "";
        }

        void bukafile() //fungsi untuk membuka file ".txt"
        {
            bersih();
            OpenFileDialog buka = new OpenFileDialog();
            buka.DefaultExt = ".txt";
            buka.Filter = " Text Documents | *.txt";
            buka.FileName = "";

            if (buka.ShowDialog() != DialogResult.Cancel)
            {
                string fileTerpilih = buka.FileName;
                if (fileTerpilih != "")
                {
                    rtinput.LoadFile(fileTerpilih, RichTextBoxStreamType.PlainText);
                }
            }
        }

        void simpanfile() //fungsi untuk menyimpan file
        {
            SaveFileDialog simpan = new SaveFileDialog();
            simpan.Filter = " Text Documents | *.txt";
            simpan.RestoreDirectory = true;
            if (simpan.ShowDialog() != DialogResult.Cancel)
            {
                StreamWriter filesimpan = new StreamWriter(File.Create(simpan.FileName));
            }
        }
    }
}
```

```

        filesimpan.Write(rtinput.Text);
        filesimpan.Dispose();
    }
}

private void bbukafile_Click(object sender, EventArgs e)
{
    if (rtinput.Text != "")
    {
        var pesan = MessageBox.Show("File belum tersimpan, yakin ingin membuka file  
baru???", "konfirmasi", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);
        if (pesan == DialogResult.Yes)
        {
            bukafile();
        }
        Else
        {
            bukafile();
        }
    }
}

private void bsimpan_Click(object sender, EventArgs e)
{
    {
        simpanfile();
    }
}

private void Form1_Load(object sender, EventArgs e)
{
    bersih();
}

}

```

Penjelasan Kode program:

1. Klik menu **File**, pilih **New Project**, pilih **WindowsFormsApplication**, kemudian klik **Ok**.
2. Kemudian masuk ke lembar listing program, ketik :

```

using Microsoft.Win32;
using System.IO;

```

//untuk library tambahan dalam program windows 32.

3. Listing program di bawah ini menjelaskan posisi form berada di tengah, form tidak bisa dibesar/dikecilkan, dan menghilangkan tombol maximize dan minimize.

```

public Form1()
{
    InitializeComponent();
    this.StartPosition = FormStartPosition.CenterScreen;
    this.FormBorderStyle = FormBorderStyle.FixedSingle;
    this.MaximizeBox = false;
    this.MinimizeBox = false;
}

```

4. Listing Program di bawah ini menjelaskan buat fungsi bersih

```
void bersih()
{
    rtinput.Text = "";
}
```

5. Listing program di bawah ini menjelaskan ketika form pertama kali dijalankan tampilan richText Box diform akan bersih.

```
private void Form1_Load(object sender, EventArgs e)
{
    bersih();
}
```

6. Listing program di bawah ini menjelaskan buat fungsi open file/ buka file.

```
void bukafile()      //fungsi untuk membuka file ".txt"
{
    bersih();
    OpenFileDialog buka = new OpenFileDialog();
    buka.DefaultExt = ".txt";
    buka.Filter = " Text Documents | *.txt";
    buka.FileName = "";

    if (buka.ShowDialog() != DialogResult.Cancel)
    {
        string fileTerpilih = buka.FileName;
        if (fileTerpilih != "")
        {
            rtinput.LoadFile(fileTerpilih, RichTextBoxStreamType.PlainText);
        }
    }
}
```

7. Listing program di bawah ini menjelaskan digunakan buat tombol buka file yang berfungsi untuk membuka file yang berada di sistem komputer/notebook.

```
private void bbukafile_Click(object sender, EventArgs e)
{
    if (rtinput.Text != "")
    {
        var pesan = MessageBox.Show("File belum tersimpan, yakin ingin membuka file baru???", "konfirmasi", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);
        if (pesan == DialogResult.Yes)
        {
            bukafile();
        }
    }
    Else
    {
        bukafile();
    }
}
```

8. Listing program di bawah ini menjelaskan fungsi simpan.

```
private void bsimpan_Click(object sender, EventArgs e)
{
{
    simpanfile();
}
}
```

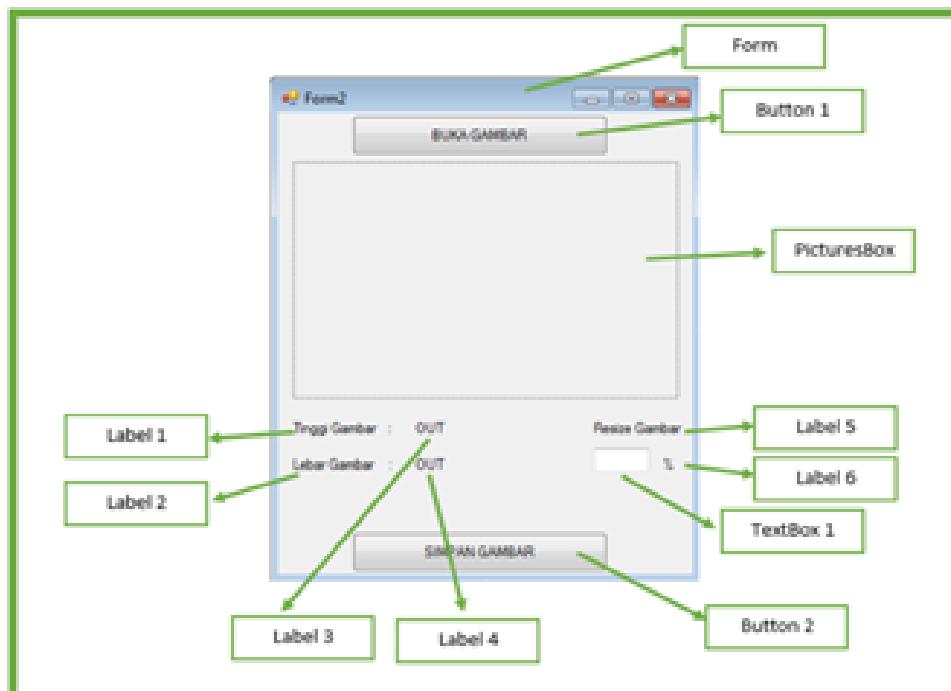
9. Listing program di bawah ini menjelaskan fungsi tombol simpan yang berfungsi untuk menyimpan file yang dibuat di form

```
void simpanfile()      //fungsi untuk menyimpan file
{
    SaveFileDialog simpan = new SaveFileDialog();
    simpan.Filter = " Text Documents | *.txt";
    simpan.RestoreDirectory = true;
    if (simpan.ShowDialog() != DialogResult.Cancel)
    {
        StreamWriter filesimpan = new StreamWriter(File.Create(simpan.FileName));
        filesimpan.Write(rtinput.Text);
        filesimpan.Dispose();
    }
}
```

2.2. Membuat Aplikasi Image Resizer

Aplikasi image resizer adalah aplikasi untuk mengubah ukuran gambar yang mempunyai format JPEG. Dibawah ini langkah-langkahnya.

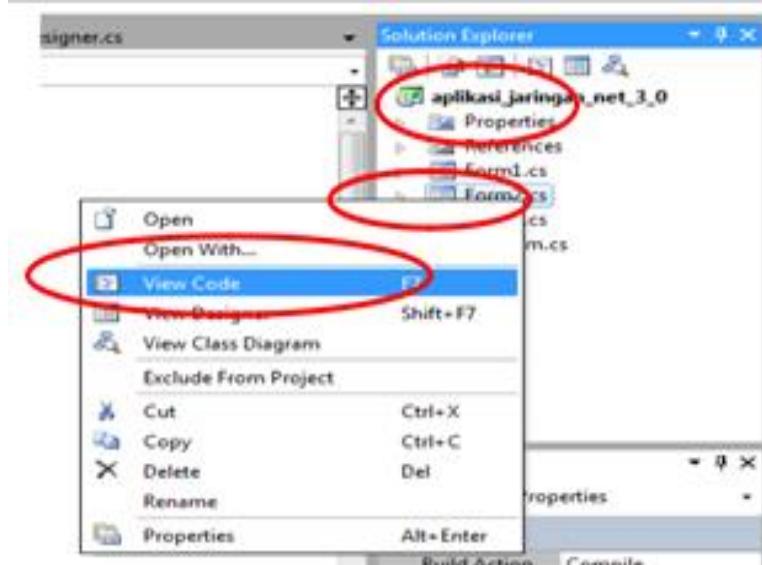
Buatlah form design seperti dibawah ini:



Atur properties setiap object diatas dengan ketentuan dibawah ini:

No.	Object	Name	Text
1	Form1	Form1	Image Resizer
2	Label1	Label1	Tinggi Gambar
3	Label2	Label2	Lebar Gambar
4	Label3	ltinggi	Out
5	Label4	llebar	Out
6	Label5	Label5	Resize Gambar
7	Label6	Label6	%
8	PictureBox1	Picture	-
9	TextBox1	Tsize	-
10	Button1	Bbuka	Buka Gambar
11	Button2	bsimpan	Simpan Gambar

Lihat Solution Explorer pada sebelah kanan program anda, klik kanan pada Form -> Pilih View Code.



Tambahkan code dibawah ini pada project code Form anda.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace aplikasi_jaringan_net_3_0
{
    public partial class Form2 : Form
    {
        private Image gambar;//di definisikan global
```

```

public Form2()
{
    InitializeComponent();
    this.StartPosition = FormStartPosition.CenterScreen;
    this.FormBorderStyle = FormBorderStyle.FixedSingle;
    this.MaximizeBox = false;
    this.MinimizeBox = false;
    tsize.MaxLength = 3;
    tsize.Enabled = false;
}

void ubahsize()
{
    if (tsize.Text != "")
    {
        int persen = Convert.ToInt32(tsize.Text);
        int tinggi = (persen * Convert.ToInt32(ltinggi.Text)) /
        100; int lebar = (persen * Convert.ToInt32(llebar.Text)) /
        100; ltinggi.Text = Convert.ToString(tinggi); llebar.Text =
        Convert.ToString(lebar);
    }
}
void simpangambar()
{
    int tinggi = Convert.ToInt32(ltinggi.Text);
    int lebar = Convert.ToInt32(llebar.Text);
    Bitmap ukuranbaru = new Bitmap(lebar, tinggi,
    System.Drawing.Imaging.PixelFormat.Format24bppRgb);
    Graphics gbr = Graphics.FromImage(ukuranbaru);
    gbr.CompositingQuality =
    System.Drawing.Drawing2D.CompositingQuality.HighSpeed;
    gbr.SmoothingMode = System.Drawing.Drawing2D.SmoothingMode.HighSpeed;
    gbr.InterpolationMode =
    System.Drawing.Drawing2D.InterpolationMode.HighQualityBicubic;
    gbr.PixelOffsetMode =
    System.Drawing.Drawing2D.PixelOffsetMode.HighSpeed;
    Rectangle rect = new Rectangle(0, 0, lebar,
    tinggi); gbr.DrawImage(gambar, rect);
    SaveFileDialog simpan = new SaveFileDialog();
    //listing program untuk penyimpanan
    simpan.Filter = "Jpeg Format|*.Jpg";
    simpan.RestoreDirectory = true;

    if (simpan.ShowDialog() != DialogResult.Cancel)
    {
        ukuranbaru.Save(simpan.FileName);
        ukuranbaru.Dispose();
        MessageBox.Show("Gambar Berhasil Disimpan", "Info");
    }
}

void bukagambar()
{
    OpenFileDialog bukagambar = new OpenFileDialog();
    if (bukagambar.ShowDialog() == DialogResult.OK)
    {
        this.gambar = Image.FromFile(bukagambar.FileName);
        picture.SizeMode = PictureBoxSizeMode.StretchImage;
        //menyesuaikan gambar sesuai dengan picturebox
        picture.ImageLocation = bukagambar.FileName;
        //mencari lokasi gambar
    }
}

```

```

        ltinggi.Text = gambar.Height.ToString();
        llebar.Text = gambar.Width.ToString();
        tsize.Enabled = true;
        tsize.Clear();
    }

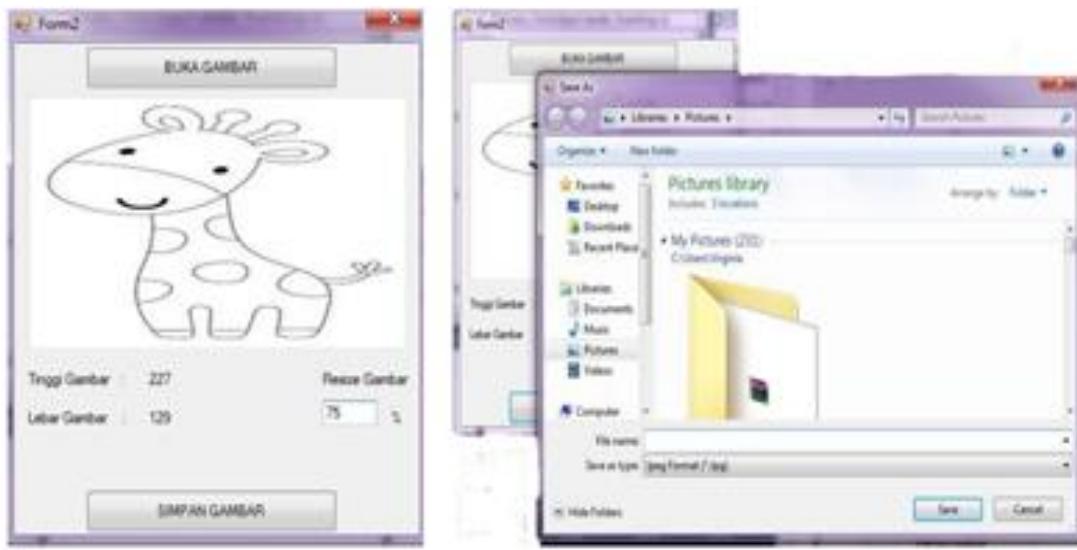
private void bbuka_Click(object sender, EventArgs e)
{
    bukagambar();
}

private void tsize_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        ubahsize();
    }
}

private void bsimpan_Click(object sender, EventArgs e)
{
    simpangambar();
}
}
}

```

Hasil Running Image Resizer





Lembar Kerja Praktik Network Programming II

PERTEMUAN III REGISTRY

1. Pengertian Registry

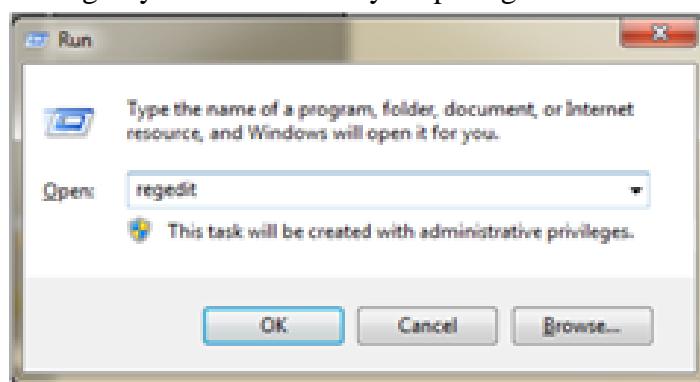
Registry adalah sebuah pusat database yang menyimpan pengaturan konfigurasi pada sebuah sistem operasi dalam hal ini yaitu Microsoft Windows. Registry yang biasa disebut Windows Registry. Registry pada OS Windows ini biasanya terdapat informasi dan pengaturan untuk hardware, software dan system operasi itu sendiri, software non-system. Pengaturan-pengaturan tersebut diatur dengan Key.

Biasanya pada Registry Windows terdiri dari komponen-komponen diantaranya:

- 1) Key
- 2) Subkey
- 3) Value

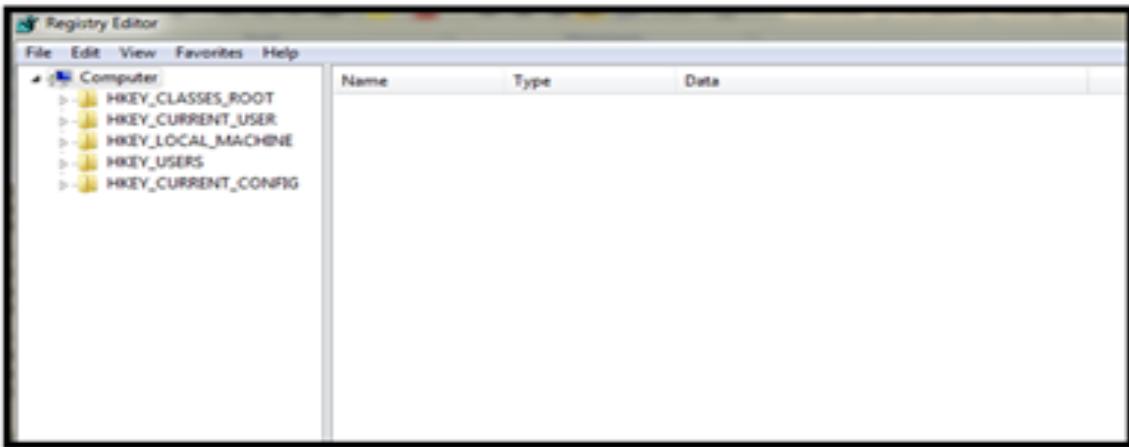
A. Cara Mengakses Registry Editor (Regedit)

Ada dua cara mudah untuk membuka Regedit yaitu menggunakan fitur "Run" dan buka melalui start menu windows. Setelah buka salah satu pilihan tersebut ketikkan saja "Regedit" tanpa tanda kutip lalu tekan enter. maka sudah bisa langsung mengakses Registry Editor. Contohnya seperti gambar berikut ini :



1. Class Registry

Class registry adalah kumpulan dari key-key yang terdapat di bawahnya dimana key-key tersebut ditempatkan pada class yang sesuai dengan fungsi key tersebut. Di dalam registry terdapat lima buah member class. Berikut adalah jenis member class dan fungsi dari class tersebut.

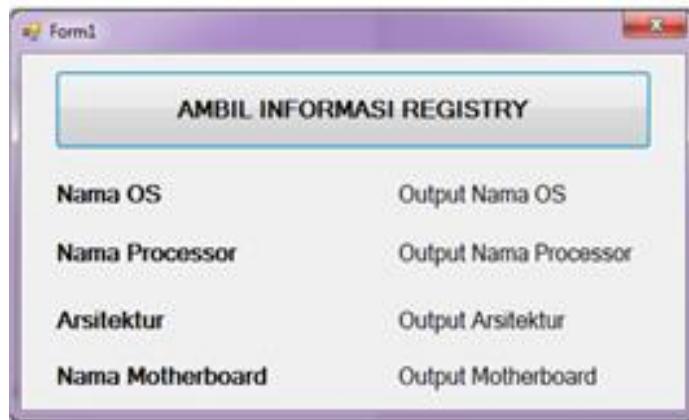


MEMBERS	FUNGSI		KEY
ClassesRoot	Menyimpan informasi tentang dan class serta propertinya.	type	HKEY_CLASSES_ROOT
CurrentUser	Menyimpan informasi tentang dan class serta propertinya.	type	HKEY_CURRENT_USER
LocalMachine	Menyimpan informasi konfigurasi dari mesin lokal.		HKEY_LOCAL_MACHINE
Users	Menyimpan informasi mengenai konfigurasi default pemakai.		HKEY_USERS
CurrentConfig	Menyimpan informasi hardware yang tidak spesifik pemakai.		HKEY_CURRENT_CONFIG

2. Fungsi Registry

Adalah Untuk Mengatur Bagaimana Hardware Dan Software di dalam laptop/PC Agar Dapat difungsikan Sebagaimana Mestinya. Pada registry windows kita hanya bisa melihat code-code atau menggunakan bahasa komputer. Dan Untuk manfaatnya sudah jelas yaitu untuk memudahkan sebuah Sistem Operasi bekerja dengan baik dan serta agar kita bisa mengontrol Operating System dengan mudah.

3.1. Latihan Aplikasi Informasi Registry dengan C#



Buatlah sebuah project Windows Form Application dengan nama “informasi.sln”. Tambahkan komponen-komponen berikut ini pada form:

Komponen	Text	Name
Button	AMBIL INFORMASI REGISTRY	breged
Label1	Nama OS	label1
Label2	NamaProcesor	label2
Label3	Arsitektur	label3
Label4	Nama Motherboard	label4
Label5	Output Nama OS	Ios
Label6	Output Nama Prosesor	Iprocessor
Label7	Output Arsitektur	Iarsitektur
Label8	Output Motherboard	Imotherboard

Ketikan listing berikut:

```
smi.cs
Ipertemuan3\Form1.cs
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using Microsoft.Win32;

namespace pertemuan3
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            this.StartPosition = FormStartPosition.CenterScreen;
            this.FormBorderStyle = FormBorderStyle.FixedSingle;
            this.MaximizeBox = false;
            this.MinimizeBox = false;
        }

        private void breged_Click(object sender, EventArgs e)
        {
            RegistryKey buka = Registry.LocalMachine;

            RegistryKey namaOS = buka.OpenSubKey("SOFTWARE\Microsoft\Windows NT\CurrentVersion");
            RegistryKey procesor = buka.OpenSubKey("HARDWARE\DESCRIPTION\System\CentralProcessor\0");
            RegistryKey arsitektur = buka.OpenSubKey("HARDWARE\DESCRIPTION\System\CentralProcessor\0");
            RegistryKey motherboard = buka.OpenSubKey("HARDWARE\DESCRIPTION\System\BIOS");

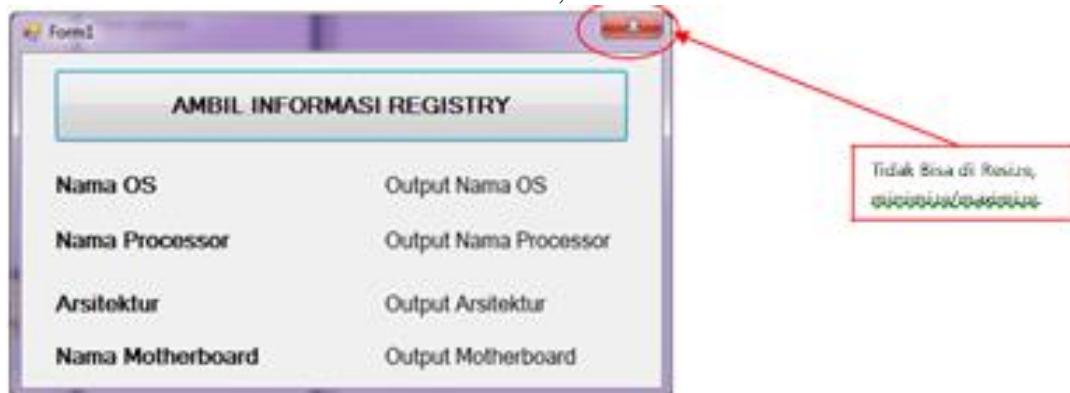
            Ios.Text = Convert.ToString(namaOS.GetValue("ProductName"));
            Iprocessor.Text = Convert.ToString(procesor.GetValue("ProcessorNameString"));
            Iarsitektur.Text = Convert.ToString(arsitektur.GetValue("Identifier"));
            Imotherboard.Text = Convert.ToString(motherboard.GetValue("BaseboardManufacturer"));
        }
    }
}
```

Penjelasan listing:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using Microsoft.Win32;
```

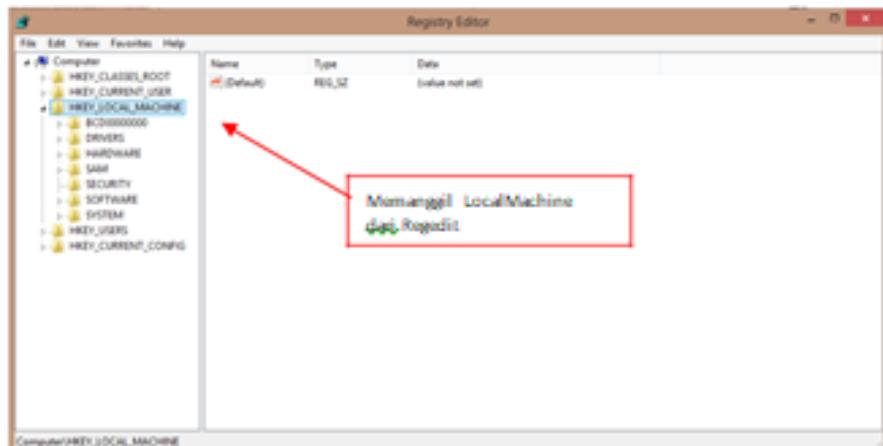
```
//namespace yang di gunakan dalam program  
  
public Form1()  
{  
    InitializeComponent();  
    this.StartPosition = FormStartPosition.CenterScreen;  
    this.FormBorderStyle = FormBorderStyle.FixedSingle;  
    this.MaximizeBox = false;  
    this.MinimizeBox = false;  
}
```

//membuat ukuran form tidak bisa di resize, minimize/maximize



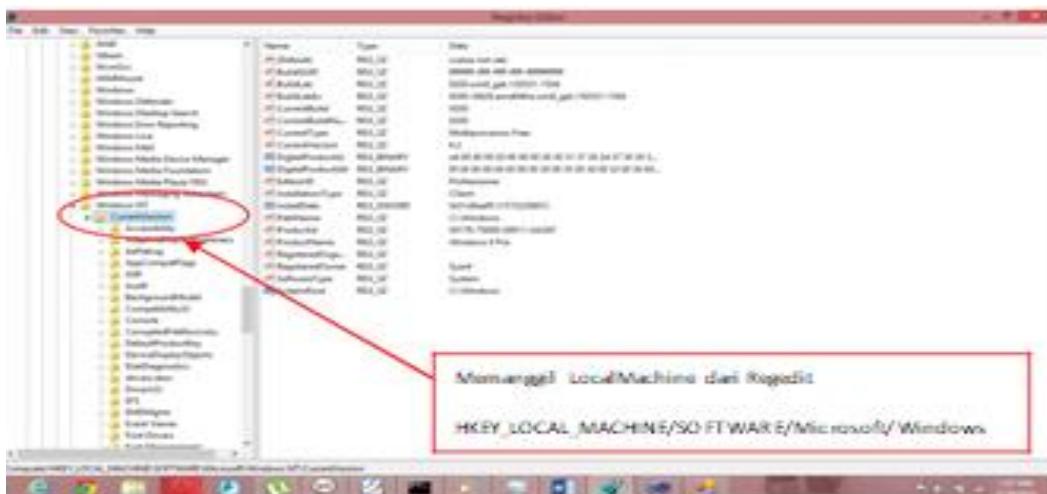
```
RegistryKey buka = Registry.LocalMachine;
```

//varibel “buka” memanggil LocalMachine dari Regedit:



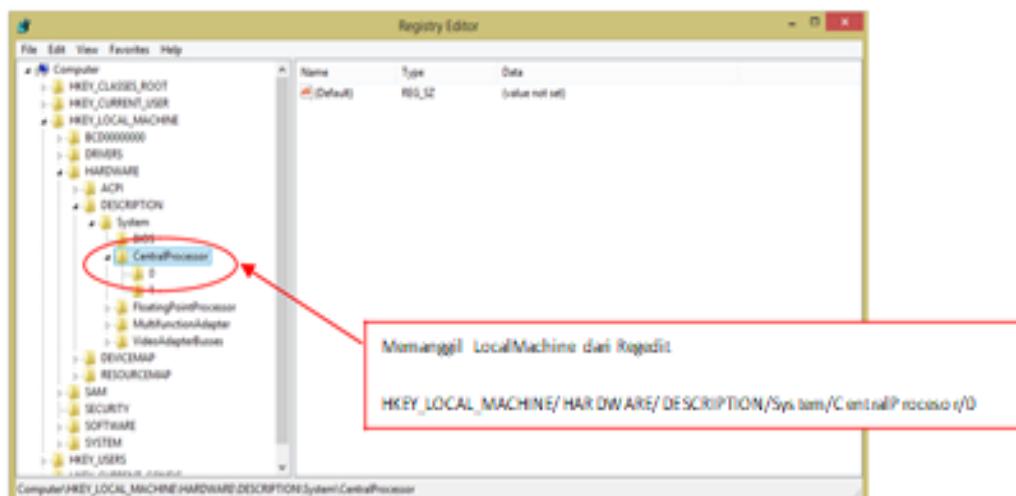
```
RegistryKey namaOS = buka.OpenSubKey("SOFTWARE\\Microsoft\\Windows  
NT\\CurrentVersion");
```

//membuat variable “namaOS” berisikan komponen informasi yang ada pada registry



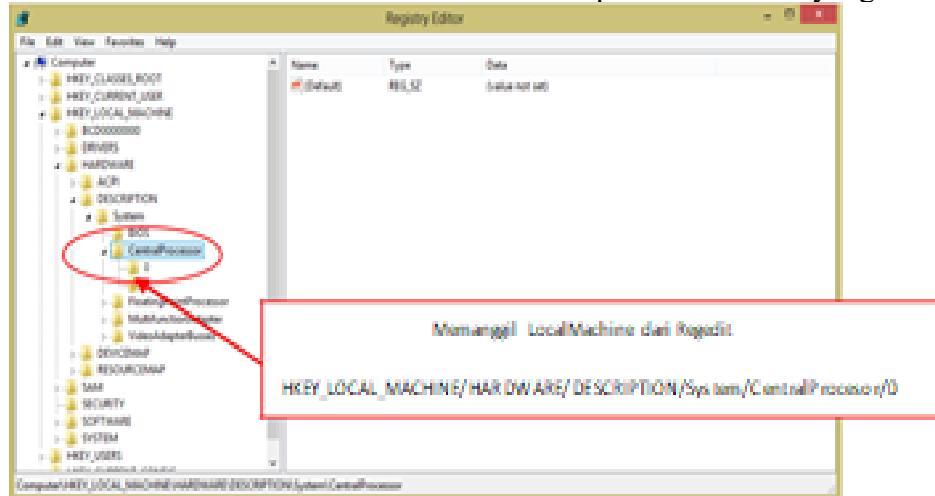
```
RegistryKey procesor =  
buka.OpenSubKey("HARDWARE\\DESCRIPTION\\System\\CentralProcessor\\0");
```

//membuat variable “prosesor” berisikan komponen informasi yang ada pada registry



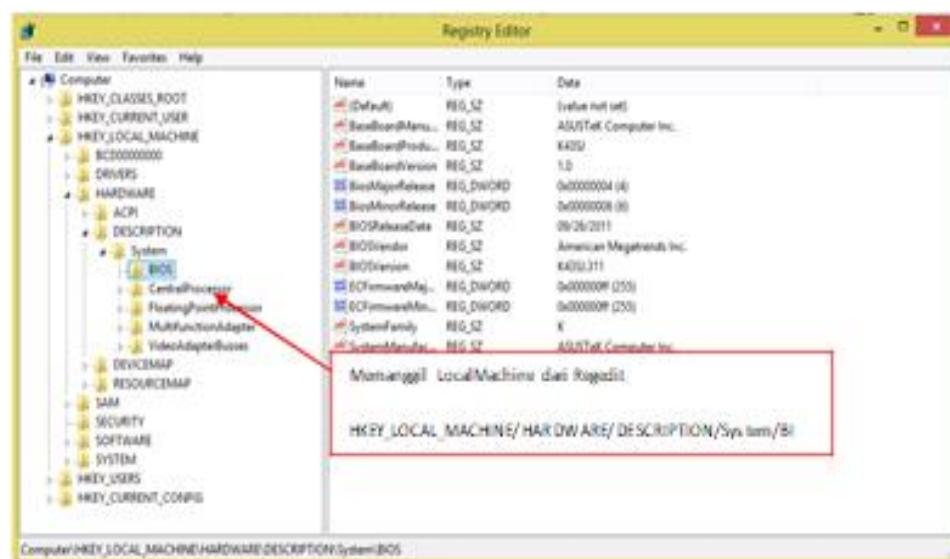
```
RegistryKey arsitektur =  
buka.OpenSubKey("HARDWARE\\DESCRIPTION\\System\\CentralProcessor\\0");
```

//membuat variable “arsitektur”berisikan komponen informasi yang ada pada registry



```
RegistryKey motherboard =  
buka.OpenSubKey("HARDWARE\\DESCRIPTION\\System\\BIOS");
```

//membuat variable “namaOS”, “prosesor”, “arsitektur”, “motherboard” berisikan komponen informasi yang ada pada registry



```

l_n_os.Text = Convert.ToString(namaOS.GetValue("ProductName"));
l_np.Text=Convert.ToString(prosesor.GetValue("ProcessorNameString"));
l_a_os.Text = Convert.ToString(arsitektur.GetValue("Identifier"));
l_mobo.Text = Convert.ToString(motherboard.GetValue("BaseBoardManufacturer"));

```

//listing di atas mengkonversi value dari variable Registry Key dan memasukkan ke label pada form



3.2. Latihan aplikasi informasi alamat jaringan

Aplikasi ini menampilkan informasi alamat jaringan sebuah Laptop/PC yang informasinya diambil dari “regedit



Ketikan listing berikut ini:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using Microsoft.Win32;

namespace jaringan_dengan_form
{
    public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();
            this.StartPosition = FormStartPosition.CenterScreen;
            this.FormBorderStyle = FormBorderStyle.FixedSingle;

```

```

        this.MaximizeBox = false;
        this.MinimizeBox = false;
    }

    private void button1_Click(object sender, EventArgs e)
    {
        RegistryKey buka = Registry.LocalMachine;
        RegistryKey kartuJaringan =
            buka.OpenSubKey("SOFTWARE\Microsoft\Windows
NT\CurrentVersion\NetworkCards");
        RegistryKey konfJaringan =
            buka.OpenSubKey("SYSTEM\CurrentControlSet\Services");

        if (kartuJaringan == null)
        {
            MessageBox.Show("Maaf Registry Key Tidak Valid", "Peringatan",
                MessageBoxButtons.OK, MessageBoxIcon.Warning);
        }

        StringBuilder tambahText = new StringBuilder();

        foreach (string namaKey in kartuJaringan.GetSubKeyNames())
        {
            RegistryKey namaServiceKartu =
            buka.OpenSubKey("SOFTWARE\Microsoft\Windows
NT\CurrentVersion\NetworkCards" + "\\ " + namaKey);
            {
                MessageBox.Show("Maaf Registry Key Tidak Valid",
                    "Peringatan", MessageBoxButtons.OK, MessageBoxIcon.Warning);
            }
            tambahText.AppendLine("Kartu Network: " +
                (string)namaServiceKartu.GetValue("Description"));

            RegistryKey keyJaringan =
            buka.OpenSubKey("SYSTEM\CurrentControlSet\\Services\" +
                ((string)namaServiceKartu.GetValue("ServiceName")) +
                "\Parameters\Tcpip");

            if (keyJaringan == null)
            {
                MessageBox.Show("Belum ada konfigurasi ip", "Peringatan",
                    MessageBoxButtons.OK, MessageBoxIcon.Warning);
            }
            Else
            {
                string[] alamatip = (string[])keyJaringan.GetValue("IPaddress");
                string[] subnetmask = (string[])keyJaringan.GetValue("SubnetMask");
                string[] defaultgateway =
                (string[])keyJaringan.GetValue("DefaultGateway");
                string[] defaultdns = (string[])keyJaringan.GetValue("DefaultDns");
            }
        }
    }
}

```

```
if (alamatip != null || subnetmask != null || defaultdns != null||defaultgateway != null )
{
    foreach (string ipaddress in alamatip)
    {
        tambahText.AppendLine("IP Address = " + ipaddress);
    }

    foreach (string subnet in subnetmask)
    {
        tambahText.AppendLine("Subnet Mask = " + subnet);
    }

    foreach (string gateway in defaultgateway)
    {
        tambahText.AppendLine("Default Gateway = " + gateway);
    }
    tambahText.AppendLine("Default Dns = " + defaultdns);
}
}

MessageBox.Show(tambahText.ToString());
}
```



Lembar Kerja Praktik Network Programming II

PERTEMUAN IV WINDOWS MANAGEMENT INSTRUMENTATION

4.1. Windows Management Instrumentation (WMI)

WMI adalah core teknologi management pada windows yang memungkinkan kita untuk melakukan tugas-tugas yang bersifat management windows dengan menulis script secara programmatika, WMI menyediakan pendekatan yang konsisten untuk melakukan tugas-tugas yang bersifat management Windows yang kita ketemu sehari-hari dengan programming atau scripting language. Dengan WMI, kita tidak hanya bisa melakukan tugas-tugas management windows pada komputer lokal, tetapi juga bisa mengelola komputer yang remote (tentunya selama kita punya hak akses terhadap komputer tersebut).

Adapun beberapa contoh tugas-tugas yang dapat dilakukan dengan WMI:

1. Melakukan query (permintaan data) ke sistem operasi kita untuk mendapatkan sejumlah informasi seperti:
 - a. Pemakaian harddisk.
 - b. Membuka File
 - c. Mendapatkan kondisi baterai yang di pakai
 - d. Mendapatkan sejumlah aplikasi yang telah terinstall di komputer kita,
2. Menciptakan atau menjalankan sebuah proses atau aplikasi secara scripting / programmatik. Misalnya kita dapat membuka sebuah calculator windows (calc.exe) secara programmatik.
3. Membuat schedule atau jadwal untuk menjalankan sebuah proses pada waktu yang telah ditentukan.
4. Mengelola (bahkan, me-restart atau mematikan) komputer lokal ataupun remote.

Oleh karena fleksibilitas WMI yang tinggi, kita bisa mengkombinasikan berbagai kemampuannya (seperti yang disebutkan di atas), sehingga dapat menjadi sebuah aplikasi yang sangat membantu. Salah satu contohnya adalah kita ingin membuat sebuah aplikasi yang memonitor pemakaian memory pada komputer kita, bila telah mencapai 3 GB, maka akan terjadi aksi tertentu (misalnya mengirimkan email kepada administrator, mencatat proses-proses yang sedang aktif ke log file, dan lain sebagainya). WMI memungkinkan kita untuk melakukan tugas-tugas tersebut secara custom dan flexible.

Berikut ini adalah aplikasi pendeksi Jaringan, Aplikasi ini memanfaatkan Fungsi WMI yang ada pada Visual Studio 2010 (C#).

a) Membuat Aplikasi Deteksi Device Adapter dan Status Jaringan (IP)

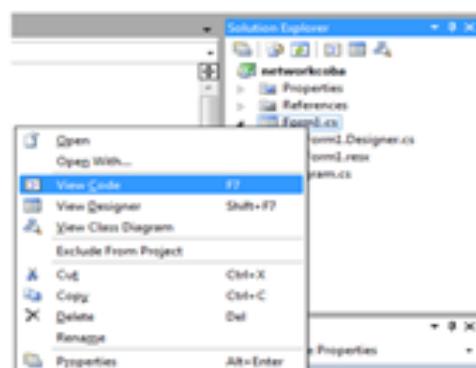
1. Buatlah form design seperti dibawah ini



2. Atur properties setiap object diatas dengan ketentuan dibawah ini:

No	Object	Name	Text
1	Form1	Form1	Aplikasi Deteksi Device Manager
2	Label1	Label1	Choice Adapter Device:
3	ComboBox1	comboBox1	-
4	Button	bget	Get
5	ListBox1	listBox1	-

3. Lihat Solution Explorer pada sebelah kanan program anda, klik kanan pada Form
-> Pilih View Code.



4. Tambahkan code dibawah ini pada project code Form anda.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Net.NetworkInformation;
using System.Collections;

namespace networkcoba
{
    public partial class Form1 : Form
    {
        private NetworkInterface[] nicArr;
        public Aplikasi_Pendeteksi_Jaringan()
        {
            InitializeComponent();
            InitializeNetworkInterface();
        }

        public void InitializeNetworkInterface()
        {
            nicArr = NetworkInterface.GetAllNetworkInterfaces();

            for (int i = 0; i < nicArr.Length; i++)
            {
                comboBox1.Items.Add(nicArr[i].Name)
            }
            comboBox1.SelectedIndex = 0;
        }

        public void networkproc()
        {
            NetworkInterface nic = nicArr[comboBox1.SelectedIndex];
            //Secara otomatis network interface yang ada akan terdaftar pada ComboBox1
            IPGlobalProperties global_propertise =
                IPGlobalProperties.GetIPGlobalProperties();
            ArrayList info = new ArrayList();

            info.Add("Interface Information for: "
+global_propertise.HostName+global_propertise.DomainName);
            info.Add("NetBIOS node type : "+global_propertise.NodeType);
            info.Add("=====");
            info.Add("Name: " + nic.Name); info.Add("Description : "+nic.Description);
            info.Add("Network Interface Type : "+nic.NetworkInterfaceType);
```

```

info.Add("Physical Address : "+nic.GetPhysicalAddress().ToString());

                // Info MAC Address yang di pakai
info.Add("Adapter ID: "+nic.Id.ToString());
info.Add("Receive Only: "+nic.IsReceiveOnly.ToString());
info.Add("Status : " +nic.OperationalStatus.ToString());
info.Add("Speed : "+nic.Speed.ToString());
PIInterfaceProperties properties = nic.GetIPProperties();
info.Add("Properties: "); info.Add(" |DNS Addresses : ");

foreach(IPAddressInformation uniCast in properties.UnicastAddresses)
info.Add(" -> : "+uniCast.Address.ToString());
info.Add(" | AnyCast Addresses: ");
foreach (IPAddressInformation anycast in properties.AnycastAddresses)
info.Add(" -> : " + anycast.Address.ToString());
info.Add(" |Support multi-cast : " + nic.SupportsMulticast.ToString ());
info.Add(" |Multicast Addresses: ");
foreach(IPAddressInformation multicast in properties.MulticastAddresses)
info.Add(" -> : " + multicast.Address.ToString());
info.Add(" |Gateway Addresses : ");
foreach (GatewayIPAddressInformation gateway in
properties.GatewayAddresses)
info.Add(" -> : " + gateway.Address.ToString());

if (nic.Supports(NetworkInterfaceComponent.IPv4) == true)
{
    IPv4InterfaceProperties ipv4props =
    properties.GetIPv4Properties(); info.Add("+IPV4 Properties : ");
if (ipv4props != null)
{
    info.Add(" |Interface Index : " + ipv4props.Index.ToString());
    info.Add(" |Automatic Private Addressing Active : " +
    ipv4props.IsAutomaticPrivateAddressingActive.ToString()); info.Add(" |
Automatic Private Addressing Enabled : " +
    ipv4props.IsAutomaticPrivateAddressingEnabled.ToString());
    info.Add(" |DHCP Enabled : " + ipv4props.IsDhcpEnabled.ToString());
    info.Add(" |Forwadding Enabled: " +
    ipv4props.IsForwardingEnabled.ToString());
    info.Add(" |MTU Size : " + ipv4props.Mtu.ToString());
    info.Add(" \\|Uses Wins : " + ipv4props.UsesWins.ToString());
}
Else
{
    info.Add(" |Device has no Ipv4 properties : ");
}
}
Else
{
    info.Add(" |+IPv4 is not implemented : ");
}
if (nic.Supports(NetworkInterfaceComponent.IPv6) == true)
{
    IPv6InterfaceProperties ipv6props =
    properties.GetIPv6Properties(); info.Add(" +IPV6 Properties : ");
if (ipv6props != null)

```

```
        {
            info.Add(" +IPV6 Properties : ");
            info.Add(" |Interface Index : " + ipv6props.Index.ToString());
            info.Add(" \\MTU Size : " + ipv6props.Mtu.ToString());
        }
    Else
    {
        info.Add(" |Device has no IPV6 properties");
    }
}
Else
{
    info.Add(" +IPV6 is not Implemented");
}
foreach (string a in info)
{
    listBox1.Items.Add(a);
}
}

private void bget_Click(object sender, EventArgs e)
{
    listBox1.Items.Clear();
    networkproc();
}
}
```

b) Aplikasi Bandwidth

Bandwidth monitoring adalah suatu program yang digunakan untuk memantau lalu lintas data dalam jaringan atau bandwidth . pada pembahasan kali ini adalah pembuatan program tersebut menggunakan microsoft visual studio c#. untuk tampilan interfacenyanya adalah :



Langkah awal buatlah satu project baru atau gunakan project yang sudah ada dan tambah form baru dalam project tersebut. Kemudian desainlah form tersebut sesuai dengan tampilan diatas menggunakan tool box. Adapun komponennya adalah :

NO	Component Name	Text
1	Label 1	Interface
2	Label 2	Speed
3	Label 3	Sort Bytes
4	Label 4	Received Bytes
5	Label 5	Download
6	Label 6	Upload
7	Label 7	0
8	Label 8	0
9	Label 9	0
10	Label 10	0
11	Label 11	0
12	Label 12	Bandwidth Monitoring
13	comboBox1	-
14	Button 1	Start
15	Button 2	Stop

Untuk membuat source programnya kita perlu men-include class bernama **System.Net.NetworkInformation**. berikut programmnya :

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Net.NetworkInformation;

namespace pertemuan3
{
    public partial class Aplikasi_Bandwidth : Form
    {
        public const double waktu_update=1000;
        private NetworkInterface[] network_interface;
        public Timer timer;
        public Aplikasi_Bandwidth()
        {
            InitializeComponent();
            InitializeNetworkInterface();
            Timer_intial();
        }

        private void Aplikasi_Bandwidth_Load(object sender, EventArgs e)
```

```

        {
            timer.Stop();
        }
    public void InitializeNetworkInterface()
    {
        network_interface = NetworkInterface.GetAllNetworkInterfaces();

        for (int i = 0; i < network_interface.Length; i++)
        {
            comboBox1.Items.Add(network_interface[i].Name);
        }
        comboBox1.SelectedIndex = 0;
    }

    long lngBytesSend;
    long lngBytesReceived;

public void proc()
{
    NetworkInterface NIC = network_interface[comboBox1.SelectedIndex];
    IPv4InterfaceStatistics status = NIC.GetIPv4Statistics();
    int sent_Speed = (int)(status.BytesSent - lngBytesSend) / 1024;
    int received_speed = (int)(status.BytesReceived - lngBytesReceived) / 1024;

    label8.Text = (NIC.Speed / 1000000) + "Mbps";
    label10.Text = status.BytesReceived.ToString();
    label9.Text = status.BytesSent.ToString();
    label12.Text = (sent_Speed).ToString() + "KB/s";
    label11.Text = (received_speed).ToString() + "KB/s";
    lngBytesSend = status.BytesSent; lngBytesReceived
    = status.BytesReceived;
}
public void clear()
{
    label8.Text = "0";
    label9.Text = "0";
    label10.Text = "0";
    label11.Text = "0";
    label12.Text = "0";
}

public void Timer_intial()
{
    timer = new Timer();
    timer.Interval = (int)waktu_update;
    timer.Tick += new EventHandler(Update_waktu);
}

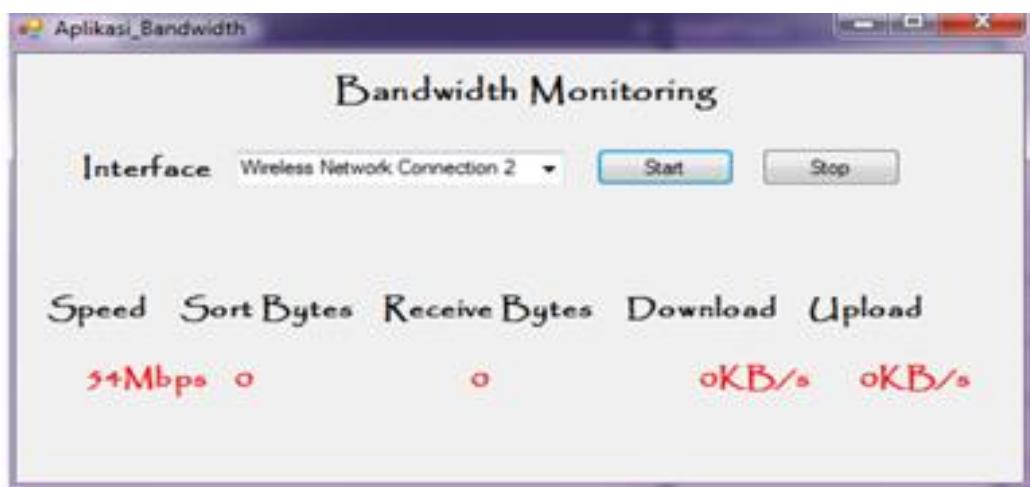
void Update_waktu(object sender, EventArgs e)
{ proc(); }

```

```
private void button1_Click(object sender, EventArgs e)
{
    timer.Start();
}

private void button2_Click(object sender, EventArgs e)
{
    timer.Stop();
    clear();
}
}
```

Hasil Output



Penjelasan Kode :

Pada awal penggunaan name space kita menggunakan fungsi “**using System.Net.NetworkInformation;** ”. banyak fungsi jaringan yang menggunakan library ini salah satunya dengan informasi bandwidth. Kemudian bagian ini kita menggunakan timer sebagai status netwok kita dengan intervalnya =1000

```
public partial class Aplikasi_Bandwidth : Form
{
    public const double waktu_update=1000;
    private NetworkInterface[] network_interface;
    public Timer timer;
```

- jangan lupa untuk memasukan method tersebut pada bagian :

```
public Aplikasi_Bandwidth()
{
    InitializeComponent();
    InitializeNetworkInterface();
    Timer_intial();
}
```

Dan pada bagian ini kita mendeskripsikan NIC pada komputer kita dengan perintah :” **NetworkInterface.GetAllNetworkInterfaces();** ”. Dan informasi tersebut akan ditampilkan pada label 7 sampai dengan 11.

```
public void InitializeNetworkInterface()
{
    network_interface = NetworkInterface.GetAllNetworkInterfaces();

    for (int i = 0; i < network_interface.Length; i++)
    {
        comboBox1.Items.Add(network_interface[i].Name);
    }
    comboBox1.SelectedIndex = 0;
}
long lngBytesSend;
long lngBytesReceived;
```

```

public void proc()
{
    NetworkInterface NIC = network_interface[comboBox1.SelectedIndex];
    IPv4InterfaceStatistics status = NIC.GetIPv4Statistics();

    int sent_Speed = (int)(status.BytesSent - lngBytesSend) / 1024;
    int received_speed = (int)(status.BytesReceived - lngBytesReceived) / 1024;

    label8.Text = (NIC.Speed / 1000000) + "Mbps";
    label10.Text = status.BytesReceived.ToString();
    label9.Text = status.BytesSent.ToString();
    label12.Text = (sent_Speed).ToString() + "KB/s";
    label11.Text = (received_speed).ToString() + "KB/s";
    lngBytesSend = status.BytesSent; lngBytesReceived
    = status.BytesReceived;
}

```

- Kemudian kita set untuk label 7 sampai 11 sebelum button start dimulai dengan nilainya 0 :

```

public void clear()
{
    label8.Text = "0";
    label9.Text = "0";
    label10.Text = "0";
    label11.Text = "0";
    label12.Text = "0";
}

```

- Kemudian untuk menjalankan tombol start dan stop masukan perintah berikut

```

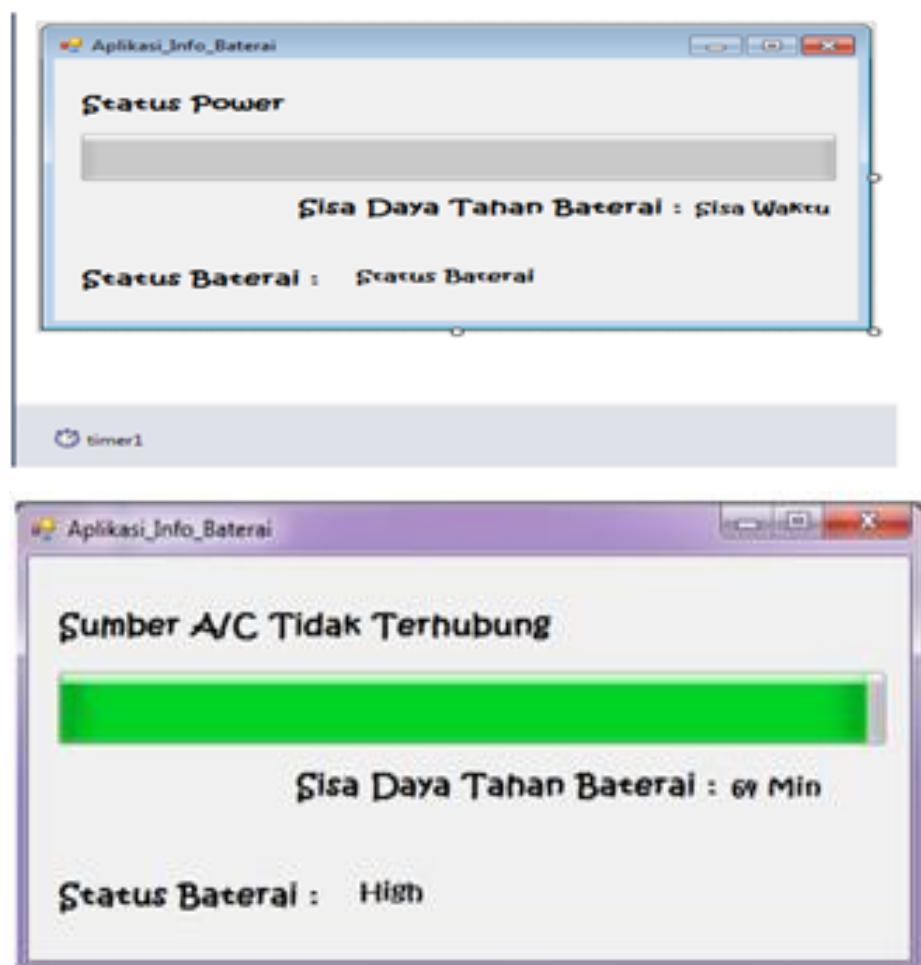
Private void button1_Click_1(object sender, EventArgs e)
{
    timer.Start();
}

Private void button2_Click_1(object sender, EventArgs e)
{
    timer.Stop();
    clear();
}

```

c) **Aplikasi Info Baterai**

Aplikasi ini berfungsi untuk memberikan informasi kondisi battery.



Adapun komponennya adalah:

No	Component Name	Text	Name
1	Label1	Status Power	lstatuspower
2	Label2	Sisa Daya Tahan Baterai :	label2
3	Label3	Sisa Waktu	lsisa
4	Label4	Status Baterai :	label4
5	Label5	Status Baterai	lstatusbaterai
6	Progress Bar	-	progressBar1
7	Timer1	-	timer1

Listing Program:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace aplikasi_jaringan_net_3_0
{
    public partial class Aplikasi_Info_Baterai : Form
    {
        public Aplikasi_Info_Baterai()
        {
            InitializeComponent();
            timer1.Enabled = true;
        }

        private void refresBaterai()
        {
            PowerStatus daya_baterai = SystemInformation.PowerStatus;
            switch(daya_baterai.PowerLineStatus) {

                case PowerLineStatus.Online:
                    Istatuspower.Text="Terhubung ke Sumber
A/C"; break;
                case PowerLineStatus.Offline:
                    Istatuspower.Text = "Sumber A/C Tidak Terhubung";
                    break;
                case PowerLineStatus.Unknown:
                    Istatuspower.Text = "Status Sumber A/C Tidak di
Ketahui"; break;
            }
            int persenBaterai = (int)(daya_baterai.BatteryLifePercent *
100); if (persenBaterai <= 100)
            {
                progressBar1.Value = persenBaterai;
            }
            Else
            {
                progressBar1.Value = 0;
            }
            int sisa_waktu = daya_baterai.BatteryLifeRemaining;
            if (sisa_waktu >= 0)
            {
                Isisa.Text = string.Format("{0} Min", sisa_waktu / 60);
            }
            Else
            {
                Isisa.Text = string.Empty;
            }
            Istatusbaterai.Text = daya_baterai.BatteryChargeStatus.ToString();
        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            refresBaterai(); }
```

```
    }
```

Penjelasan Kode program:

1. Klik menu **File**, pilih **New Project**, pilih **WindowsFormsApplication**, kemudian klik **Ok**.
2. Kemudian masuk ke lembar listing program `timer1.Enabled = true;`, untuk merefresh waktu ke sistem komputer/notebook.

```
public Aplikasi_Info_Baterai()
{
    InitializeComponent();
    timer1.Enabled = true;
}
```

- 3 Listing program di bawah ini menjelaskan sebuah fungsi yang berguna untuk merefresh baterai yang ada di sistem komputer/Notebook .

```
private void refresBaterai()
{
    PowerStatus daya_baterai = SystemInformation.PowerStatus;
    switch(daya_baterai.PowerLineStatus)
        //memeriksa status sumber AC
    {
        case PowerLineStatus.Online:
            Istatuspower.Text="Terhubung ke Sumber A/C";
            break;
        case PowerLineStatus.Offline:
            Istatuspower.Text = "Sumber A/C Tidak Terhubung";
            break;
        case PowerLineStatus.Unknown:
            Istatuspower.Text = "Status Sumber A/C Tidak di Ketahui";
            break;
    }
    int persenBaterai = (int)(daya_baterai.BatteryLifePercent * 100);
```

```

//mendaftarkan persenBaterai sebagai pengakses jumlah daya baterai
//untuk ditampilkan pada progress bar
if (persenBaterai <= 100)
{
    progressBar1.Value = persenBaterai;
}
Else
{
    progressBar1.Value = 0;
}

int sisa_waktu = daya_baterai.BatteryLifeRemaining;

//mendaftarkan sisa_waktu sebagai pengakses sisa waktu pakai baterai
if (sisa_waktu >= 0)
{
    Isisa.Text = string.Format("{0} Min", sisa_waktu / 60);
}
Else
{
    Isisa.Text = string.Empty;
}
Istatusbaterai.Text = daya_baterai.BatteryChargeStatus.ToString();
}

```

5. Listing program di bawah ini menjelaskan eksekusi waktu yang di hitung perinterval nya.

```

private void timer1_Tick(object sender, EventArgs e)
{
    refresBaterai();
}

```



Lembar Kerja Praktik Network Programming II

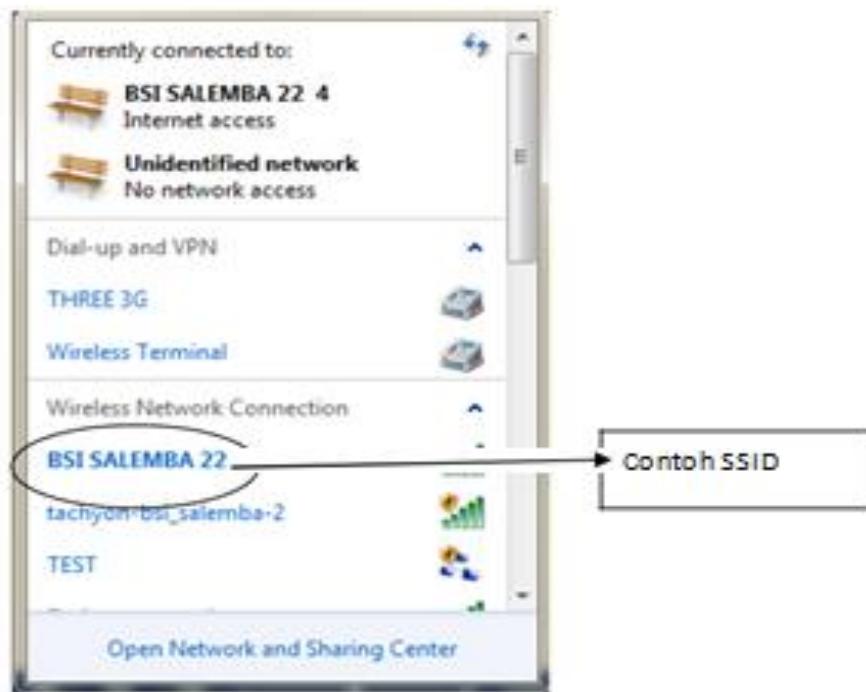
PERTEMUAN V MANIPULASI JARINGAN

Pada pertemuan ini kita membahas tentang manipulasi jaringan, pada program C# banyak sekali program jaringan yang dapat kita buat dengan memanfaatkan, socket, registry, framework, WMI. Salah satunya adalah membuat jaringan WLAN. Pada Operating System sendiri disediakan tools untuk membuat sebuah Jaringan WLAN, namun kita dapat mempersingkat langkah-langkah tersebut dengan membuat sebuah program manipulasi jaringan .

a) HOTSPOT ENGINE

Hotspot adalah area dimana seorang client dapat terhubung dengan jaringan internet secara wireless (nirkabel/tanpa kabel) dari PC, note book atau gadget seperti Handphone dalam jangkauan radius kurang lebih beberapa ratus meteran atau tergantung dari kekuatan frekuensi/signal.Pada pertemuan ini Buatlah desain seperti dibawah ini dengan dua text box, tiga label dan satu button.

Pada pembuatan WLAN hal yang wajib di buat adalah SSID, SSID atau Service set identifier adalah tempat mengisikan nama dari access point yang akan disetting. Setelah membuat SSID kita membuat Password, hal ini di perlukan untuk keamanan dalam komunikasi data.



Langkah- Langkah Pembuatan Program Hotspot Engine

Buatlah Form di bawah ini



Daftar Komponen:

NO	Component Name	Text
1	Label 1	Hotspot Engine
2	Label 2	SSID
3	Label 3	PASSWORD
4	TextBox 1	
5	TextBox 2	
6	Button	START

Listing Program :

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Diagnostics;
using System.Security.Principal;
using System.Net;

namespace pertemuan3
{
    public partial class Hospot_Engine : Form
    {
        public Hospot_Engine()
        {
            InitializeComponent();
        }
    }
}
```

```

        if (!IsAdmin())
        {
            RestartElevated();
        }

        bsi_hotspot(null, null, false);
        Application.Exit();
    }

private void bsi_hotspot(string ssid, string key, bool status)
{
    ProcessStartInfo processStartInfo = new ProcessStartInfo("cmd.exe");
    processStartInfo.RedirectStandardInput = true;
    processStartInfo.RedirectStandardOutput = true;
    processStartInfo.CreateNoWindow = true;
    processStartInfo.UseShellExecute = false;
    Process process = Process.Start(processStartInfo);

    if (process != null)
    {
        if (status)
        {
            process.StandardInput.WriteLine("netsh wlan set hostednetwork mode=allow ssid=" +
                ssid + " key=" + key);

//pemanggilan untuk membuat ssid
            process.StandardInput.WriteLine("netsh wlan start hostednetwork");
            process.StandardInput.Close();
        }

        Else
        {
            process.StandardInput.WriteLine("netsh wlan stop
hostednetwork"); process.StandardInput.Close();
        }
    }
}

public static bool IsAdmin()
{
    WindowsIdentity id = WindowsIdentity.GetCurrent();
    WindowsPrincipal p = new WindowsPrincipal(id); return
    p.IsInRole(WindowsBuiltInRole.Administrator);
}

public void RestartElevated()
{
    ProcessStartInfo startinfo = new ProcessStartInfo();
    startinfo.UseShellExecute = true;
    startinfo.CreateNoWindow = true;
    startinfo.WorkingDirectory = Environment.CurrentDirectory;
    startinfo.FileName = System.Windows.Forms.Application.ExecutablePath;
    startinfo.Verb = "runas";

    try

```

```

        {
            Process p = Process.Start(startinfo);
        }
    Catch
    {
        System.Windows.Forms.Application.Exit();

    }
private void button1_Click(object sender, EventArgs e)
{
    string ssid = textBox1.Text, key = textBox2.Text;
    if (!connect)
    {
        if (textBox1.Text == null || textBox1.Text == "")
        {
            MessageBox.Show("Anda Belum Mengisi SSID!", "Informasi",
MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
        Else
        {
            if (textBox2.Text == null || textBox2.Text == "")
            {
                MessageBox.Show("Anda Belum Mengisi Password!", "Informasi",
MessageBoxButtons.OK, MessageBoxIcon.Information);
            }
        }
        Else
        {
            if (key.Length >= 6)
            {
                bsi_hotspot(ssid, key, true);
                textBox1.Enabled = false;
                textBox2.Enabled = false;
                button1.Text = "Berhenti";
                connect = true;
            }
            Else
            {
                MessageBox.Show("Password harus 6 karakter atau lebih !",
"Information", MessageBoxButtons.OK, MessageBoxIcon.Information);
            }
        }
    }
    else
    {
        bsi_hotspot(null, null, false);
        textBox1.Enabled = true;
        textBox2.Enabled = true;
        button1.Text = "Start";
        X = false;
    }
}

public bool connect { get; set; }
public bool X { get; set; }
}

```

```
}
```

Perhatian :

Untuk Program ini tdk dapat di jalankan pada OS Windows XP

Pastikan Driver WLAN terinstal di Netbook or PC jika tidak akan menyebabkan Sistem Windows Crash

Penjelasan Kode :

Buat lah Project / file baru. Kemudian masuk ke bagian code program. Pada program diatas kita membutuhkan namespase :

```
using System.Diagnostics;
using System.Security.Principal;
using System.Net;
```

kemudian double klik pada button Start dan tuliskan perintah dibawah ini :

```
string ssid = textBox1.Text, key = textBox2.Text;
if (!connect)
{
    if (textBox1.Text == null || textBox1.Text == "")
    {
        MessageBox.Show("Anda Belum Mengisi SSID !", "Informasi",
                        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    else
    {
        if (textBox2.Text == null || textBox2.Text == "")
        {
            MessageBox.Show("Anda Belum Mengisi Password !",
                            "Informasi", MessageBoxButtons.OK,
                            MessageBoxIcon.Information);
        }
        else
        {
            if (key.Length >= 6)
            {
                bsi_hotspot(ssid, key, true);
                textBox1.Enabled = false;
                textBox2.Enabled = false;
                button1.Text = "Berhenti";
                connect = true;
            }
            else
            {
                MessageBox.Show("Password harus 6 karakter
                                atau lebih !", "Information",
                                MessageBoxButtons.OK,
                                MessageBoxIcon.Information);
            }
        }
    }
}
```

```

    Else
    {
        bsi_hotspot(null, null, false);
        textBox1.Enabled = true;
        textBox2.Enabled = true;
        button1.Text = "Start";
        x = false;
    }
}

```

Pada text box 1 dan 2 masih kosong maka akan muncul pesan dalam bentuk message box. Password harus lebih besar dari 6 karakter. Setelah itu buatlah sebuah method baru dengan nama bsi_hotspot. Method baru inilah yang akan menjalankan fungsi utamanya sebagai hotspot :

```

private void bsi_hotspot(string ssid, string key,bool status)
{
    ProcessStartInfo processStartInfo = new
    ProcessStartInfo("cmd.exe");
    processStartInfo.RedirectStandardInput = true;
    processStartInfo.RedirectStandardOutput = true;
    processStartInfo.CreateNoWindow = true;
    processStartInfo.UseShellExecute = false;
    Process process = Process.Start(processStartInfo);

    if (process != null)
    {
        if (status)
        {
            process.StandardInput.WriteLine("netsh wlan set
                hostednetwork mode=allow ssid=" + ssid + " key=" + key);
            // pemanggilan untuk membuat SSID
            process.StandardInput.WriteLine("netsh wlan start hosted
                network");
            process.StandardInput.Close();
        }
        else
        {
            process.StandardInput.WriteLine("netsh wlan stop
                hostednetwork");
            process.StandardInput.Close();
        }
    }
}

```

Selanjutnya buatlah dua method baru yang mendeskripsikan bahwa komputer anda adalah seorang administrator dan sebuah method untuk merestart fungsi dari hotspot :

```

public static bool IsAdmin()
{
    WindowsIdentity id = WindowsIdentity.GetCurrent();
    WindowsPrincipal p = new WindowsPrincipal(id); return
    p.IsInRole(WindowsBuiltInRole.Administrator);
}

```

```
public void RestartElevated()
{
    ProcessStartInfo startInfo = new ProcessStartInfo();
    startInfo.UseShellExecute = true;
    startInfo.CreateNoWindow = true;
    startInfo.WorkingDirectory = Environment.CurrentDirectory;
    startInfo.FileName = System.Windows.Forms.Application.ExecutablePath;
    startInfo.Verb = "runas";
    try
    {
        Process p = Process.Start(startInfo);
    }
    catch
    {

    }

    System.Windows.Forms.Application.Exit();
}
```

Terakhir jangan lupa mendeskripsikannya pada constructornya :

```
InitializeComponent();
if (!IsAdmin())
{
    RestartElevated();
}
bsi_hotspot(null,null,false);
Application.Exit()
```



Lembar Kerja Praktik Network Programming II

PERTEMUAN VI PENGENALAN SOCKET

6.1. Pengertian Socket

Socket adalah salah satu layer application yang menghubungkan aplikasi dengan Network Card dari Komputer. Melalui Socket ini, kita dapat melakukan komunikasi di beberapa komputer atau device lainnya. Implementasi dari socket akan berbeda berdasarkan Operating System yang digunakan tetapi secara sistem transfer datanya akan sama.

6.2. Pengertian Socket Pada Framework.Net

Framework.Net telah menyediakan library untuk mengimplementasikan layanan internet dalam aplikasi dengan mudah. Untuk mengakses library socket, kita dapat memanfaatkan namespace System.Net dan System.Net.Sockets.

6.3. Pengertian Sistem Socket pada Jaringan Komputer

Pada umumnya, socket dalam jaringan komputer dikenal berdasarkan orientasi koneksinya. Ini terbagi menjadi

- Connection Oriented. Antar Socket saling mengikat dan berhubungan sebelum socket akan mengirimkan data.
- Connectionsless Oriented. Antar socket tidak terikat selamanya hanya terikat ketika melakukan pengiriman dan penerimaan data selanjutnya putus

Tabel Port

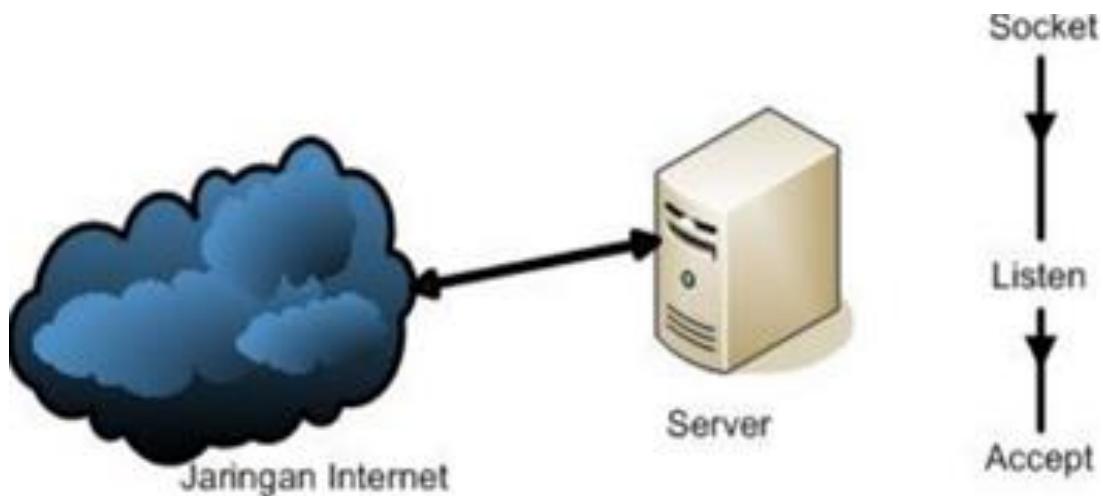
Port Number	Protocol
20	FTP
21	FTP
25	SMTP (Email Outgoing)
53	DNS
80	HTTP (Web)
110	POP3 (email incoming)
143	IMAP (email incoming)

6.4. Pemrograman Jaringan TCP/IP

Pada Sistem Jaringan TCP/IP menerapkan sistem connection-oriented artinya server dan client melakukan koneksi terlebih dahulu sebelum melakukan transfer data, sedangkan framework.NET, kita dapat memanfaatkan library socket yang terdapat pada namespace System.Net.Sockets.

6.5. Implementasi Socket Pada Sistem Client – Server

Dalam pemrograman jaringan komputer, kita memahami bagaimana sistem kerja dari socket itu sendiri, oleh karena itu hal yang paling mudah untuk memahami prinsip kerja dari socket ini yaitu melalui sistem client-server.



1. Socket TCP/IP

Pada framework .NET, penggunaan socket untuk jaringan TCP/IP dapat menggunakan kelas `TCPListener`.

```
TcpListener server = new TcpListener(IPLocal, port);
```

untuk menjalankan socketnya cukup memanggil method `Start()`,

```
server.Start();
```

2. Listen

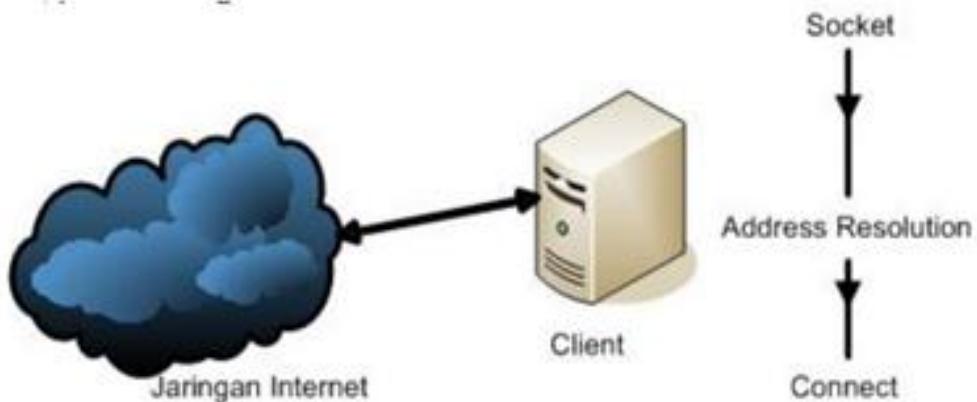
Listen merupakan proses menunggu suatu koneksi yang akan terhubung dengan socket pada port yang telah dibuat oleh socket. Untuk melakukan proses listen, cukup dengan method `Start()`.

3. Accept

Jika ada socket client yang terhubung dengan server maka server dapat menerima koneksi ini dengan menggunakan method `AcceptTcpClient()` yang mengembalikan socket client yaitu `TcpClient`. Berikut implementasinya:

```
TcpClient client=server.AcceptTcpClient();
```

Selanjutnya kita akan membahas bagaimana proses socket pada client. Untuk sistem clientnya perhatikan gambar berikut



4. Socket Client

Untuk membuat socket client,cukup dengan menggunakan kelas TcpClient, Berikut cara membuat socketnya:

```
TcpClient client=new client TcpClient(server,port);
```

Dimna :

Server adalah Ip Address server dengan string format .dot(.) contoh "192.168.0.5"

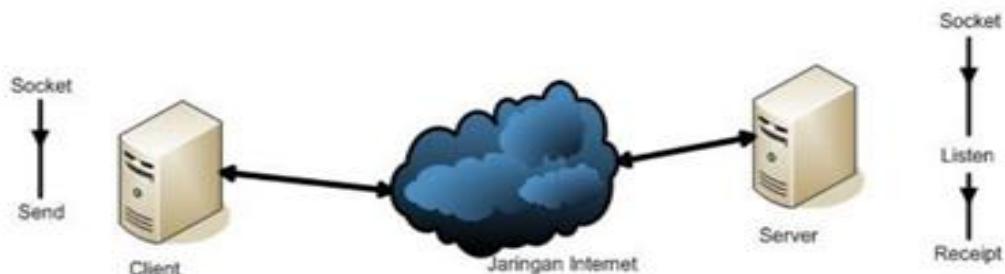
Port server yang digunakan oleh server untuk proses listen.

5. Address Resolution

Ini merupakan proses validasi Ip Address dari server. Dalam hal ini, kita perlu melalukannya karena kelas TcpClient akan melakukan sndiri pada saat instasiasi kelas TcpClient

Pemrograman Jaringan UDP/IP

Pada sistem jaringan UDP menerapkan sistem connectionless-oriented artinya pada saat melakukan hubungan antara server dan client tidak memerlukan koneksi terlebih dahulu untuk melakukan proses transfer data.dalam mendalami pemrograman jaringan UDP/IP, framework.NET telah menyediakan kelas yang berguna yaitu UDPClient, dengan menggunakan kelas ini kita dapat memoptimalkan pemrograman jaringan berbasis UDP/IP



Kita akan bahas satu persatu bagian dari gambar tersebut:

1. Socket

untuk socket UDP/IP, cukup menggunakan kelas UdpClient.

untuk Server:

```
Int32 port = 9077;  
UdpClient udpserver=new UdpClient(port);
```

untuk Client:

```
UdpClient udpclient=new UdpClient();
```

2. Listen

Proses ini hanya ada pada server, Untuk melakukan ini,Cukup kita memanggil method Receive(). Method ini akan melakukan proses Listen pada port yang telah diisi pada kelas UdpClient.

3. Receipt

Pada saat kita memanggil method Receive() maka method ini akan mengembalikan array byte dari data yang telah dikirim client.Berikut ini cara pemakaianya.

```
IPEndPoint RemoteIPEndPoint =null;  
Byte[] receiveBytes=udpserver.Receive(ref RemoteIPEndPoint);  
String returnData  
System.Text.Encoding.ASCII.GetString(receiveBytes);
```

4. Send

Kelas Udp Client juga menyediakan method untuk mengirim data yaitu send().Berikut ini cara mengirim data ke server:

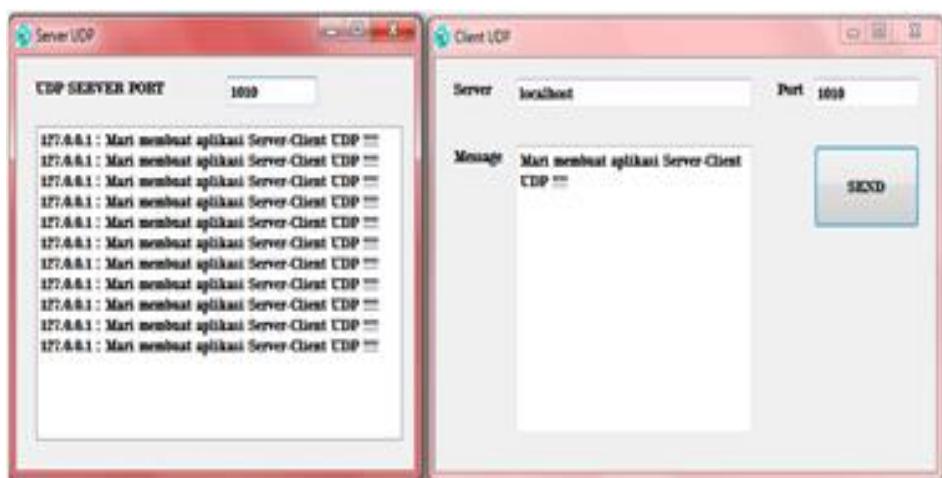
```
Int32 port =9077;  
String Ipserver="192.168.0.5";  
String msg="HEllo C# BSI";  
Byte[] databytes=System.Text.Encoding.ASCII.GetBytes(msg);  
IPEndPoint endPoint = new  
IPEndPoint(IPAddress.Parse(IPServer),port);udpclient.Send(databyt  
es,data  
bytes.Length,  
endPoint);
```

Registry adalah sebuah pusat database yang menyimpan pengaturan konfigurasi pada sebuah sistem operasi dalam hal ini yaitu Microsoft Windows. Registry yang biasa disebut Windows Registry. Registry pada OS Windows ini biasanya terdapat informasi dan pengaturan untuk hardware, software dan system operasi itu sendiri, software non-system. Pengaturan-pengaturan tersebut diatur dengan Key.

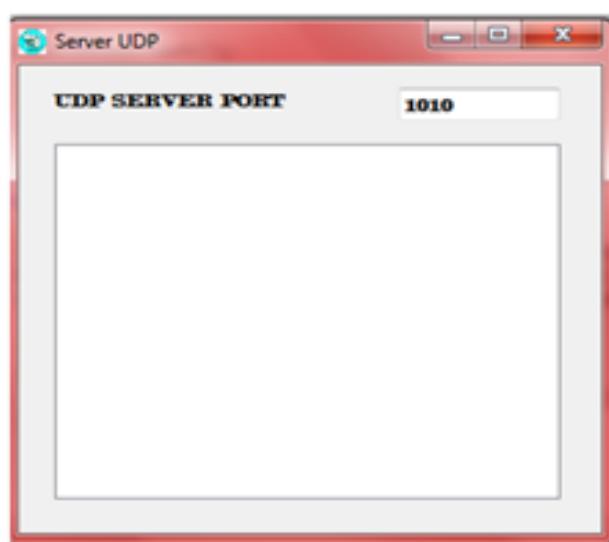
Biasanya pada Registry Windows terdiri dari komponen-komponen diantaranya:

- 4) Key
- 5) Subkey

a) APLIKASI CLIENT – SERVER UDP SEDERHANA



Aplikasi untuk Server UDP



Adapun komponen nya adalah:

NO	Component Name	Text	Name
1	ListBox	-	listBox1
2	Label	UDP SERVER PORT	label1
3	TextBox	1010	textBox1

Listing Program:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Net;
using System.Net.Sockets;
using System.Threading;
using System.IO;

namespace Tugas_1_ServerUDP
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void serverThread() //membuat metode
        {
            UdpClient clUdp = new UdpClient(int.Parse(textBox1.Text));
            //membuat sebuah objek udpclient baru
            while(true)
            {
                IPEndPoint remoteIp = new IPEndPoint(IPAddress.Any, 0);
                //membuat sebuah objek ipendpoint yg akan mengirim
                //paket byte[] receiveByte = clUdp.Receive(ref remoteIp);
                string returnData = Encoding.ASCII.GetString(receiveByte);
                //untuk memberikan delegasi kepada listbox
                listBox1.Invoke(new MethodInvoker(delegate()
                {
                    listBox1.Items.Add(remoteIp.Address.ToString() +
                        " : " + returnData.ToString());
                }));
            }
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            Thread svrThread = new Thread(new ThreadStart(serverThread));
            svrThread.Start();
        }
    }
}
```

Penjelasan Kode program untuk Server UDP:

1. Klik menu **File**, pilih **New Project**, pilih **WindowsFormsApplication**, kemudian klik **Ok**.
2. Kemudian masuk ke lembar listing program, ketik :

```
using System.Net;
using System.Net.Sockets;
using System.Threading;
using System.IO;
```

//untuk library tambahan penggunaan soket UDP pada server dan client.

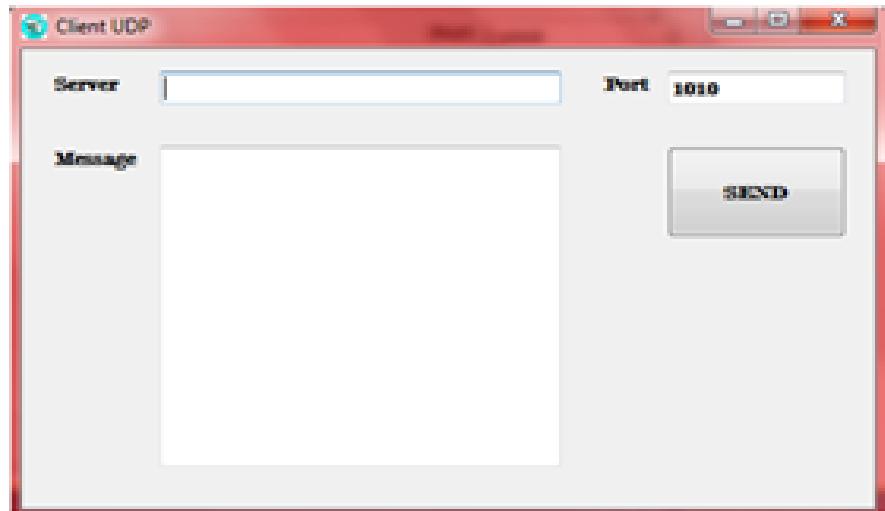
3. Pada Listing dibawah ini digunakan untuk membuat metode , agar server terhubung ke client.

```
private void serverThread() //membuat metode
{
    UdpClient c1Udp = new
    UdpClient(int.Parse(textBox1.Text));
    //membuat sebuah objek udpclient baru
    while(true)
    {
        IPEndPoint remoteIp = new IPEndPoint(IPAddress.Any, 0);
        //membuat sebuah objek ipendpoint yg akan mengirim
        paket byte[] receiveByte = c1Udp.Receive(ref remoteIp);
        string returnData = Encoding.ASCII.GetString(receiveByte);
        //untuk memberikan delegasi kepada listbox
        listBox1.Invoke(new MethodInvoker (delegate()
        {
            listBox1.Items.Add(remoteIp.Address.ToString() + " : "
            + returnData.ToString());
        }));
    }
}
```

4. Pada Listing dibawah ini digunakan pada saat form di run metode yang digunakan diatas akan berfungsi.

```
private void Form1_Load(object sender, EventArgs e)
{
    Thread svrThread = new Thread(new
    ThreadStart(serverThread)); svrThread.Start();
}
```

Aplikasi Untuk Client UDP



Adapun komponennya adalah:

NO	Component Name	Text	Name
1	TextBox	1010	textPort
2	TextBox	-	textAlmt
3	TextBox	-	textPesan
4	Label	Server	label1
5	Label	Port	label2
6	Label	Message	Label3
7	Button	SEND	button1

Listing Program:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Net;
using System.Net.Sockets;
using System.Threading;
using System.IO;

namespace Tugas_1_ClientUDP
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            UdpClient ucl = new UdpClient();

            try
            {
                ucl.Connect(textAlmt.Text, int.Parse(textPort.Text));
                //menghubungkan ke server
                Byte[] datasend = Encoding.ASCII.GetBytes(textPesan.Text);
                //merubah string menjadi byte
                ucl.Send(datasend, datasend.Length);
                //mengirim data ke server
                ucl.Close();
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }
    }
}
```

Penjelasan Kode program untuk Client UDP:

1. Klik menu **File**, pilih **New Project**, pilih **WindowsFormsApplication**, kemudian klik **Ok**.
2. Kemudian masuk ke lembar listing program, ketik :

```
using System.Net;
using System.Net.Sockets;
using System.Threading;
using System.IO;
```

untuk library tambahan penggunaan soket UDP pada server dan client .

3. Pada Listing dibawah ini digunakan sebuah button yang berfungsi menghubungkan client ke server, mengirimkan data ke server.

```
private void button1_Click(object sender, EventArgs e)
{
    UdpClient ucl = new UdpClient();

    try
    {
        ucl.Connect(textAlmt.Text, int.Parse(textPort.Text));
        //menghubungkan ke server
        Byte[] datasend = Encoding.ASCII.GetBytes(textPesan.Text);
        //merubah string menjadi byte
        ucl.Send(datasend, datasend.Length);
        //mengirim data ke server
        ucl.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

**Tambahkan Cara penggunaan nya , Run kedua form sever dan client secara bersamaan.*



Lembar Kerja Praktik Network Programming II

PERTEMUAN VII TUGAS & EVALUASI



Lembar Kerja Praktik Network Programming II

PERTEMUAN VIII PRESENTASI



Lembar Kerja Praktik Network Programming II

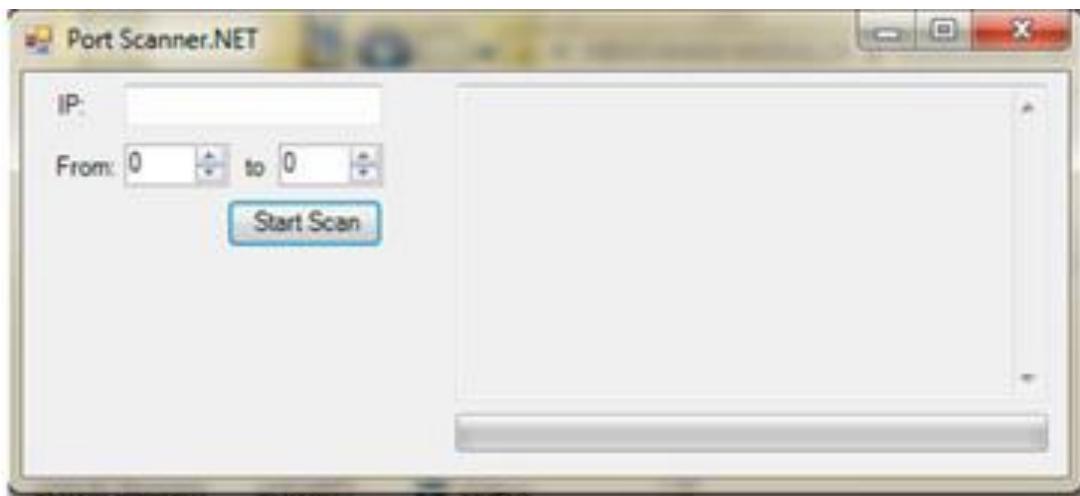
PERTEMUAN IX MULTIPLEXING SOCKET

7.1. Pengertian Socket

Socket adalah salah satu layer application yang menghubungkan aplikasi dengan Network Card dari Komputer. Melalui Socket ini, kita dapat melakukan komunikasi di beberapa komputer atau device lainnya. Implementasi dari socket akan berbeda berdasarkan Operating System yang digunakan tetapi secara sistem transfer datanya akan sama. Jika sebuah aplikasi mencoba untuk membaca dari soket buffer yang kosong atau menulis ke soket yang buffer penuh, soket biasanya akan memblokir; jeda aplikasi sampai itu dapat menyelesaikan operasi. Ini adalah masalah jika aplikasi bekerja dengan lebih dari satu soket pada suatu waktu. Sebagai contoh, perhatikan sebuah program dengan terbuka dua soket, s1 dan s2, yang mencetak setiap masukan yang diterima pada soket ke layar. Tidak ada data ini tersedia pada soket salah. Jika program membaca S1, eksekusi blok sampai data tersedia pada soket itu. Sementara itu, data yang bisa tiba di s2, tetapi program tidak pernah menerimanya. Solusi yang lebih baik adalah untuk memblokir pada soket keduanya secara bersamaan

a) PORT SCANNER

Port Scanning adalah aktivitas yang dilakukan untuk memeriksa status port TCP dan UDP pada sebuah mesin. Banyak aplikasi yang menawarkan fasilitas untuk melakukan pemeriksaan port pada sebuah mesin, seperti netcat, unicornscan.



Berikut adalah komponen yang di perlukan.

NO	Component Name	Name
1	TextBox	txtIP
2	ComboBox 1	numStart
3	ComboBox 2	numEnd
4	Button 1	btnScan
5	TextBox	txtLog
6	ProgressBar	prgScanning

Ketikan listing di bawah ini.

Code Program :

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Net.Sockets;
using System.Windows.Forms;

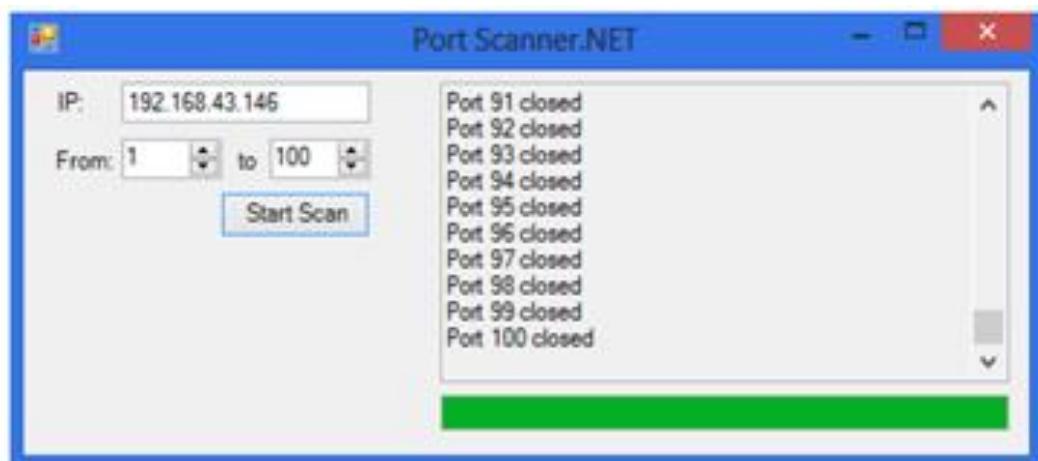
namespace PortScanner
{
    public partial class frmMain : Form
    {
        Protected int StartPort;
        Protected int EndPort;

        public frmMain()
        {
            InitializeComponent();
        }

        private void btnScan_Click(object sender, EventArgs e)
        {
            // Store values from the NumericUpDown to variables
            StartPort = Convert.ToInt32(numStart.Value); EndPort
            = Convert.ToInt32(numEnd.Value);
            // Reset the progress bar
            prgScanning.Value = 0;
            // Set the max value of the progress bar
            prgScanning.Maximum = EndPort - StartPort + 1;
            // Let the user know the application is busy
            Cursor.Current = Cursors.WaitCursor;
            // Loop through the ports between start port and end port
            for (int CurrPort = StartPort; CurrPort <= EndPort;
            CurrPort++)
            {
                TcpClient TcpScan = newTcpClient();
```

```
        try
        {
            TcpScan.Connect(txtIP.Text, CurrPort);
            // Try to connect
            // If there's no exception, we can say the
            // port is open
            txtLog.AppendText("Port " + CurrPort + " "
                + "open\r\n");
        }
        catch
        {
            // An exception occurred, thus the port is
            // probably closed
            txtLog.AppendText("Port " + CurrPort + " "
                + "closed\r\n");
        }
        // Increase the progress on the progress bar
        prgScanning.PerformStep();
    }

    Cursor.Current = Cursors.Arrow;
    // Set the cursor back to normal
}
}
}
```



Penjelasan Kode :

Buatlah Project / file baru. Kemudian masuk ke bagian code program. Pada program diatas kita membutuhkan namespase “**using System.Net.Sockets;**” kemudian double klik pada button scan dan tuliskan perintah dibawah ini :

```
StartPort = Convert.ToInt32(numStart.Value);
EndPort = Convert.ToInt32(numEnd.Value);
prgScanning.Value = 0;
prgScanning.Maximum = EndPort - StartPort + 1;
Cursor.Current = Cursors.WaitCursor;

for (int CurrPort = StartPort; CurrPort <= EndPort; CurrPort++)
{
    TcpClient TcpScan = new TcpClient();
    Try
    {
        TcpScan.Connect(txtIP.Text, CurrPort);
        txtLog.AppendText("Port " + CurrPort + " open\r\n");
    }
    catch
    {
        txtLog.AppendText("Port " + CurrPort + " closed\r\n");
    }
    prgScanning.PerformStep();
}

Cursor.Current = Cursors.Arrow;
```

Catatan:

- Pada Aplikasi ini menscan port jaringan mana saja yang terbuka dan yang tertutup.
- Total maksimum jumlah port untuk setiap protokol transport yang digunakan adalah 65536 buah.

b) DNS (Domain Name System)

Protokol jaringan tidak hanya digunakan untuk memindahkan data dari satu titik ke titik lain. Beberapa protokol memiliki tujuan tertentu yang membantu menjaga Internet traffic mengalir dan membuat menggunakan jaringan lebih mudah. Protokol utilitas ini mungkin tidak diperlukan untuk setiap aplikasi jaringan , namun, karena ini adalah teknologi berkembang, banyak pengembang tidak mengtahui bagaimana menerapkan fitur tersebut. Dengan memanfaatkan teknologi ini, dimungkinkan untuk menambahkan fitur unik untuk produk , yang mungkin menyatakan bahwa keunggulan kompetitif. Bab ini menjelaskan DNS dengan pembahasan protokol utilitas yang menarik yang dikembangkan oleh Microsoft, bernama WMI.

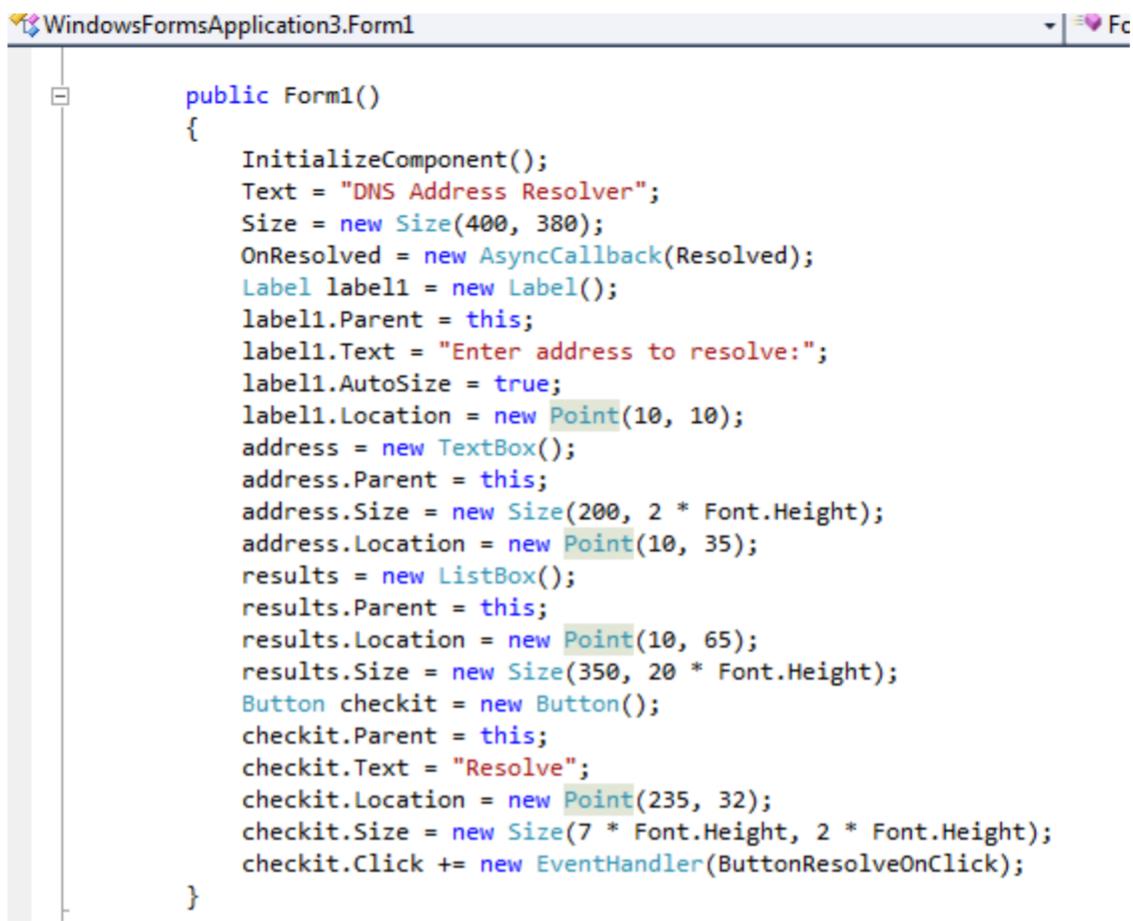
DNS beroperasi pada UDP port 53 dan dijelaskan dalam RFC 1010, RFC 1304, RFC 1035, dan RFC 1183. DNS adalah untuk mengubah nama domain menjadi alamat IP karena orang merasa sulit untuk mengingat string angka lebih dari sembilan digit. DNS dikembangkan untuk menyediakan sistem yang mengkonversi mudah dikenali nama domain menjadi alamat IP. Tidak ada komputer pusat menyimpan daftar nama domain terhadap alamat IP. Sebaliknya, jaringan di seluruh

dunia menyimpan informasi ini di server DNS. setiap Situs web biasanya akan terdaftar pada dua server DNS, mesin ini dikatakan otoritatif dalam domain. DNS server secara rutin mengecek setiap permintaan lainnya untuk informasi terbaru. Anda dapat menggunakan DNS.GetHostByName untuk mengkonversi nama domain (string) ke alamat IP (IPHostEntry). Kebalikan dari tindakan ini, mengubah sebuah IP alamat untuk nama domain, dapat dicapai dengan menggunakan DNS.GetHostByAddress. Ada lebih dari DNS mengubah alamat IP untuk nama domain dan sebaliknya, Bahkan, sebagian besar DNS penanganan di belakang layar, dan paling pemrograman jaringan tingkat tinggi jarangmengetahui IP alamat server atau klien dengan yang berkomunikasi.

1. DNS ADDRESS RESOLVER

Berikut ini adalah Program untuk aplikasi penangkap DNS dimna kita memasukan sebuah nama website lalu kita mencari DNS (IP DNS) dari website tersebut, perlu diperhatikan aplikasi ini berjalan mencari DNS berdasarkan Domain yang di daftarkan ke penyedia layanan Domain. sehingga memerlukan jaringan yang aktif terkoneksi dengan **Internet**.

- a. Buatlah Project Baru dengan nama DNS_RESOLVER
- b. Tidak perlu mendesain form dikarenakan kita akan mendesain form lewat constraktornya(desain langsung dengan list program).



The screenshot shows the Microsoft Visual Studio IDE with the title bar "WindowsFormsApplication3.Form1". The code editor displays the constructor of Form1. The code initializes components like a label, a text box, and a list box, and sets up an event handler for a button click.

```
public Form1()
{
    InitializeComponent();
    Text = "DNS Address Resolver";
    Size = new Size(400, 380);
    OnResolved = new AsyncCallback(Resolved);
    Label label1 = new Label();
    label1.Parent = this;
    label1.Text = "Enter address to resolve:";
    label1.AutoSize = true;
    label1.Location = new Point(10, 10);
    address = new TextBox();
    address.Parent = this;
    address.Size = new Size(200, 2 * Font.Height);
    address.Location = new Point(10, 35);
    results = new ListBox();
    results.Parent = this;
    results.Location = new Point(10, 65);
    results.Size = new Size(350, 20 * Font.Height);
    Button checkit = new Button();
    checkit.Parent = this;
    checkit.Text = "Resolve";
    checkit.Location = new Point(235, 32);
    checkit.Size = new Size(7 * Font.Height, 2 * Font.Height);
    checkit.Click += new EventHandler(ButtonResolveOnClick);
}
```

Code program Lengkap :

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Net;

namespace WindowsFormsApplication3
{
    public partial class Form1 : Form
    {
        TextBox address;
        ListBox results;
        private AsyncCallback OnResolved;

        public Form1()
        {
            InitializeComponent();
            Text = "DNS Address Resolver";
            Size = new Size(400, 380);
            OnResolved = new AsyncCallback(Resolved);
            Label label1 = new Label();
            label1.Parent = this;
```

```

        label1.Text = "Enter address to resolve:";
        label1.AutoSize = true;
        label1.Location = new Point(10, 10);
        address = new TextBox();
        address.Parent = this;
        address.Size = new Size(200, 2 * Font.Height); address.Location = new Point(10, 35); results = new ListBox();
        results.Parent = this;
        results.Location = new Point(10, 65);
        results.Size = new Size(350, 20 * Font.Height);
        Button checkit = new Button();
        checkit.Parent = this;
        checkit.Text = "Resolve";
        checkit.Location = new Point(235, 32);
        checkit.Size = new Size(7 * Font.Height, 2 * Font.Height);
        checkit.Click += new EventHandler(ButtonResolveOnClick);
    }

    void ButtonResolveOnClick(object obj, EventArgs ea)
    {
        results.Items.Clear();
        string addr = address.Text;
        Object state = newObject();
        Dns.BeginResolve(addr, OnResolved, state);
    }

    private void Resolved(IAsyncResult ar)
    {
        string buffer;
        IPHostEntry iphe = Dns.EndResolve(ar);
        buffer = "Host name: " + iphe.HostName;
        results.Items.Add(buffer);

        foreach(string alias in ipheAliases)
        {
            buffer = "Alias: " + alias;
            results.Items.Add(buffer);
        }

        foreach(IPAddress addrs in iphe.AddressList)
        {
            buffer = "Address: " + addrs.ToString();
            results.Items.Add(buffer);
        }
    }

    private void Form1_Load(object sender, EventArgs e)
    {
    }
}

```

Penjelasan listing program:

1. Buatlah project / file baru dengan cara klik menu **File**, pilih **New Project**, pilih **WindowsFormsApplication**, kemudian klik **Ok**.
2. Kemudian masuk ke file source program, tambahkan namespace dengan **using System.Net;** untuk library tambahan. Untuk penjelasan **using System.Net;** lihat di bab sebelumnya .

- Buatlah variable untuk mendeskripsikan textbox dan listview dengan code dibawah ini.

```
TextBox address;
listBox results;
private AsyncCallback OnResolved;
```

- Pada method ini yang akan ditampilkan berdasarkan listing diatas adalah sebuah text box,label,button dan listbox. Text box berfungsi untuk menginput DNS nya untuk mencari alamat IP nya.
- Listbox berfungsi untuk melihat hasil IP address nya.

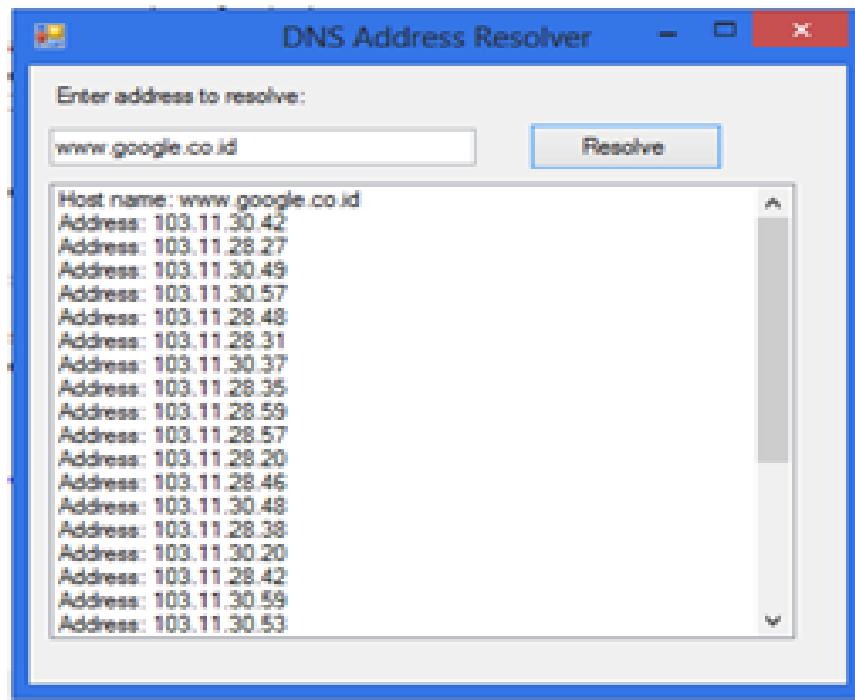
```
public Form1()
{
    InitializeComponent();
    Text = "DNS Address Resolver";
    Size = newSize(400, 380);
    OnResolved = new AsyncCallback(Resolved);
    Label label1 = newLabel();
    label1.Parent = this;
    label1.Text = "Enter address to resolve:";
    label1.AutoSize = true;
    label1.Location = newPoint(10, 10);
    address = newTextBox();
    address.Parent = this;
    address.Size = newSize(200, 2 * Font.Height);
    address.Location =
    newPoint(10, 35); results = newListBox();
    results.Parent = this;
    results.Location = newPoint(10, 65);
    results.Size = newSize(350, 20 * Font.Height);
    Button checkit = newButton();
    checkit.Parent = this;
    checkit.Text = "Resolve";
    checkit.Location = newPoint(235, 32);
    checkit.Size = newSize(7 * Font.Height, 2 * Font.Height);
    checkit.Click += newEventHandler(ButtonResolveOnClick);
}
```

- Pada bagian berikut ini adalah perintah ketika button di klik dan menemukan IP address dari sebuah DNS yang ingin dicari.
- pada method ini digunakan untuk mencari alamat IP dari DNS yang diketik dengan perintah `IPHostEntry iphe = Dns.EndResolve(ar);`

```
private void Resolved(IAsyncResult ar)
{
    string buffer;
    IPHostEntry iphe = Dns.EndResolve(ar);
    buffer = "Host name: " + iphe.HostName;
    results.Items.Add(buffer);
    foreach(string alias in ipheAliases)
    {
        buffer = "Alias: " + alias;
        results.Items.Add(buffer);
    }

    foreach(IPAddress addrs in iphe.AddressList
```

```
{  
    buffer = "Address: " + addrs.ToString();  
    results.Items.Add(buffer);  
}  
}
```





Lembar Kerja Praktik Network Programming II

PERTEMUAN X SOCKET PROGRAMMING

Pada saat suatu aplikasi berkomunikasi, awalnya aplikasi membuat *socket* baru, maka pada aplikasi tersebut akan diberikan nomer yang digunakan sebagai referensi *socket*. Jika ada suatu sistem yang menggunakan nomer referensi *socket* tersebut, maka akan terjalin suatu jaringan komunikasi antar komputer sebaik transfer data local.Untuk berkomunikasi dengan server, client harus tahu nomor IP server begitu juga nomor port yang dituju, nomor port menunjukkan service yang dijalankan. Contoh port 23 untuk Telnet Server, port 25 (SMTP) untuk Mail Server dan port 80 (HTTP) untuk Web Server. Dalam hal ini aplikasi di client sudah mengetahui port yang akan dituju. Contoh program aplikasi di client yang meminta service di server ada ftp, telnet, ssh. Untuk melihat service bisa dilihat pada file /etc/services.Program yang berjalan di server, akan berjalan sepanjang waktu (disebut sebagai daemon) sampai mesin/service dimatikan, menunggu request dari client sesuai service yang diminta

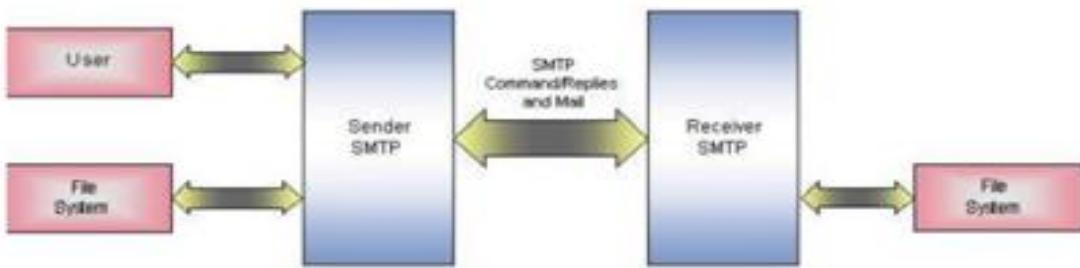
1. UDP (User Data Protocol)

User Datagram Protocol (UDP) adalah salah satu protokol lapisan transpor TCP/IP yang mendukung komunikasi yang tidak andal (unreliable), tanpa koneksi (connectionless) antara host-host dalam jaringan yang menggunakan TCP/IP. UDP memiliki karakteristik-karakteristik berikut:

- a. Connectionless (tanpa koneksi): Pesan-pesan UDP akan dikirimkan tanpa harus dilakukan proses negosiasi koneksi antara dua host yang hendak berukur informasi.
- b. Unreliable (tidak andal): Pesan-pesan UDP akan dikirimkan sebagai datagram tanpa adanya nomor urut atau pesan acknowledgment. Protokol lapisan aplikasi yang berjalan di atas UDP harus melakukan pemulihan terhadap pesan-pesan yang hilang selama transmisi. Umumnya, protokol lapisan aplikasi yang berjalan di atas UDP mengimplementasikan layanan keandalan mereka masing-masing, atau mengirim pesan secara periodik atau dengan menggunakan waktu yang telah didefinisikan.
- c. UDP menyediakan mekanisme untuk mengirim pesan-pesan ke sebuah protokol lapisan aplikasi atau proses tertentu di dalam sebuah host dalam jaringan yang menggunakan TCP/IP. Header UDP berisi field Source Process Identification dan Destination Process Identification.
- d. UDP menyediakan penghitungan checksum berukuran 16-bit terhadap keseluruhan pesan UDP

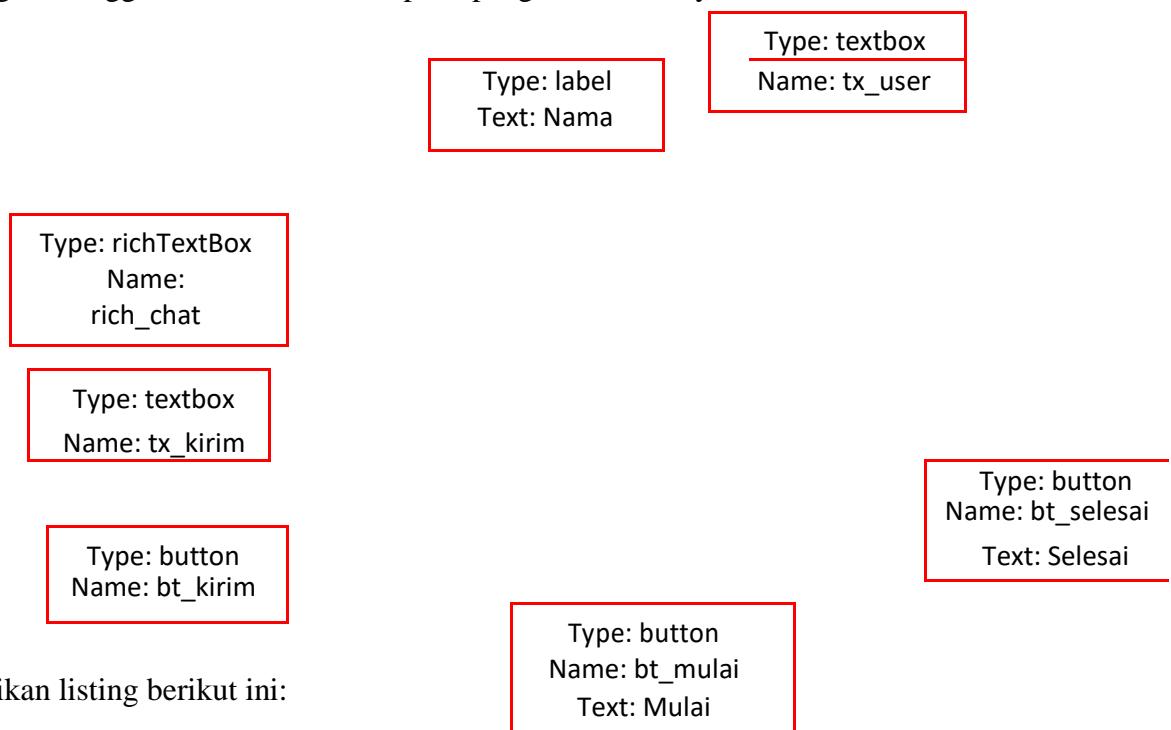
2. SMTP

SMTP digunakan untuk mengirim, tapi tidak dapat menerima, email. Setiap mail server di dunia harus sesuai dengan standar SMTP untuk mengirim email tanpa tujuan. Panduan definitif untuk SMTP dipegang oleh Internet Engineering Task Force (IETF).SMTP beroperasi pada port 25.



a) Aplikasi Chating

Aplikasi chatting sudah mulai banyak dibuat semenjak teknologi internet mulai ada. Banyak program chatting yang di buat baik yang online maupun yang ofline (Intranet), selain itu penggunaan socket pun berbeda-beda ada yang menggunakan socket UDP, TCP, FTP. Kita akan mencoba membuat Aplikasi Chating dengan menggunakan Socket UDP pada pengiriman datanya,



Ketikan listing berikut ini:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Net;
using System.Net.Sockets;
using System.Threading;

namespace chating
{
    public partial class Form1 : Form
    {
        Delegate void AddMessage(string message);
        string nama_user;
    }
}

```

```

        int port = 11000;
        string broadcast;
        IPAddress alamat_broadcast = IPAddress.Broadcast;
        UdpClient penerima_client = new UdpClient(11000);
        UdpClient pengirim_client;
        Thread jalur_penerima;

    public Form1()
    {
        InitializeComponent();
        this.Load += new EventHandler(Form1_Load);
        bt_kirim.Click += new
            EventHandler(bt_kirim_Click); this.ActiveControl =
        tx_user; rich_chat.ReadOnly = true;
        tx_kirim.Enabled = false;
        bt_kirim.Enabled = false;
        bt_selesai.Enabled = false;
    }

    private void Form1_Load(object sender, EventArgs e)
    {
        ThreadStart mulai = new
        ThreadStart(penerima); jalur_penerima = new
        Thread(mulai); jalur_penerima.IsBackground =
        true; jalur_penerima.Start();
    }

    private void aksi()
    {
        tx_kirim.Text = tx_kirim.Text.TrimEnd();
        if (!string.IsNullOrEmpty(tx_kirim.Text))
        {
            if (nama_user != "")
            {
                string u_kirim = "<" + nama_user + "> : " +
                tx_kirim.Text;
                byte[] data =
                Encoding.ASCII.GetBytes(u_kirim);
                pengirim_client.Send(data, data.Length);
                tx_kirim.Text = "";
            }
            else
            {
                string u_kirim = "<" +
                Environment.MachineName + "> : " +
                tx_kirim.Text;
                byte[] data =
                Encoding.ASCII.GetBytes(u_kirim);
                pengirim_client.Send(data, data.Length);
                tx_kirim.Text = "";
            }
        }
    }

    private void bt_kirim_Click(object sender, EventArgs e)
    {
        aksi();
    }

    private void penerima()
    {
        IPEndPoint end_point = new IPEndPoint(IPAddress.Any, port);

```

```

        AddMessage pesan_wakil = pesan_diterima;
        while (true)
        {
            byte[] data = penerima_client.Receive(ref end_point);
            string message = Encoding.ASCII.GetString(data);
            Invoke(pesan_wakil, message);
            System.Console.Beep(1500, 300);
        }
    }

    private void pesan_diterima(string pesan)
    {
        rich_chat.Text += pesan + "\n";
    }

    private void tx_kirim_KeyDown(object sender, KeyEventArgs e)
    {
        if (e.KeyCode == Keys.Enter)
        {
            aksi();
        }
    }

    private void bt_mulai_Click(object sender, EventArgs e)
    {
        nama_user = Convert.ToString(tx_user.Text);
        broadcast=Convert.ToString(alamat_broadcast);
        pengirim_client = new UdpClient(broadcast, port);
        pengirim_client.EnableBroadcast = true;
        tx_kirim.Enabled = true;
        tx_user.Enabled = false;
        bt_mulai.Enabled = false;
        bt_kirim.Enabled = true;
        bt_selesai.Enabled = true;
    }

    private void bt_selesai_Click(object sender, EventArgs e)
    {
        tx_user.Enabled = true;
        tx_kirim.Enabled = false;
        bt_kirim.Enabled = false;
        bt_mulai.Enabled = true;
        bt_selesai.Enabled = false;
    }
}

```

Penjelasan listing:

```

using System.Net;
using System.Net.Sockets;
using System.Threading;

```

//Digunakan untuk memanggil library yang akan dipakai, fungsi yang diperlukan seperti penggunaan thread, akses port dan socket

```

delegate void AddMessage(string message);

```

//fungsi “AddMessage” yang bertipe delegate void. Delegate void digunakan karena pengaksesan tool (richTextBox) dengan 2 event yang berbeda.

```
int port = 11000;
string broadcast;
IPAddress alamat_broadcast = IPAddress.Broadcast;
```

//mendefinisikan variable “port” yang digunakan sebagai port yang digunakan. //variable ”broadcast” bertipe data string yang nantinya akan digunakan untuk menampung alamat broadcast.

//variable “alamat_broadcast” digunakan untuk menampung IP broadcast.

```
UdpClient penerima_client = new UdpClient(11000);
UdpClient pengirim_client;
```

//mendeklarasikan variable “penerima_client” dan “pengirim_client” yang menggunakan tipe UdpClient. Deklarasi UdpClient menjelaskan bahwa protocol yang digunakan pada saat penransferan data adalah protocol UDP.

```
Thread jalur_penerima;
```

//mendeklarasikan thread yang digunakan.

```
public Form1()
{
    InitializeComponent();
    this.Load += new EventHandler(Form1_Load);
    bt_kirim.Click += new EventHandler(bt_kirim_Click);
    this.ActiveControl = tx_user; rich_chat.ReadOnly =
    true; tx_kirim.Enabled = false;

    bt_kirim.Enabled = false;
    bt_selesai.Enabled = false;
}
```

//fungsi Form1 berisikan kondisi awal komponen tools yang ada pada form pada saat aplikasi dijalankan.

```
private void Form1_Load(object sender, EventArgs e)
{
    ThreadStart mulai = new
    ThreadStart(penerima); jalur_penerima = new
    Thread(mulai); jalur_penerima.IsBackground =
    true; jalur_penerima.Start();
}
```

//inisialisasi saat aplikasi menerima data dari data client/user yang lain.

```
private void aksi()
{
    tx_kirim.Text = tx_kirim.Text.TrimEnd();
    if (!string.IsNullOrEmpty(tx_kirim.Text))
    {
        if (nama_user != "")
        {
```

```

        string u_kirim = "<" + nama_user + "> : " + tx_kirim.Text;
        byte[] data = Encoding.ASCII.GetBytes(u_kirim);
        pengirim_client.Send(data, data.Length); tx_kirim.Text = "";

    }
else
{
    string u_kirim = "<" + Environment.MachineName + "> : " +
tx_kirim.Text;
    byte[] data = Encoding.ASCII.GetBytes(u_kirim);
    pengirim_client.Send(data, data.Length);
    tx_kirim.Text = "";
}
}

```

//fungsi aksi() nantinya akan digunakan pada method dari button kirim dan pada saat tombol enter ditekan ketika mengirim message.

```

private void bt_kirim_Click(object sender, EventArgs e)
{
    aksi();
}

```

//pemanggilan fungsi aksi() pada button kirim ketika di klik.

```

private void penerima()
{
    IPEndPoint end_point = new IPEndPoint(IPAddress.Any,
port); AddMessage pesan_wakil = pesan_diterima;
while (true)
{
    byte[] data = penerima_client.Receive(ref end_point);
    string message = Encoding.ASCII.GetString(data);
    Invoke(pesan_wakil, message);
    System.Console.Beep(1500, 300);
}
}

```

//fungsi diatas merupakan fungsi yang digunakan untuk menagani setiap data yang masuk melalui port yang dugunakan.

```

private void pesan_diterima(string pesan)
{
    rich_chat.Text += pesan + "\n";
}

```

//fungsi diatas merupakan fungsi yang dipanggil setelah penerima selesai mengubah data yang masuk ke data bertipe string.

```

private void tx_kirim_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        aksi();
    }
}

```

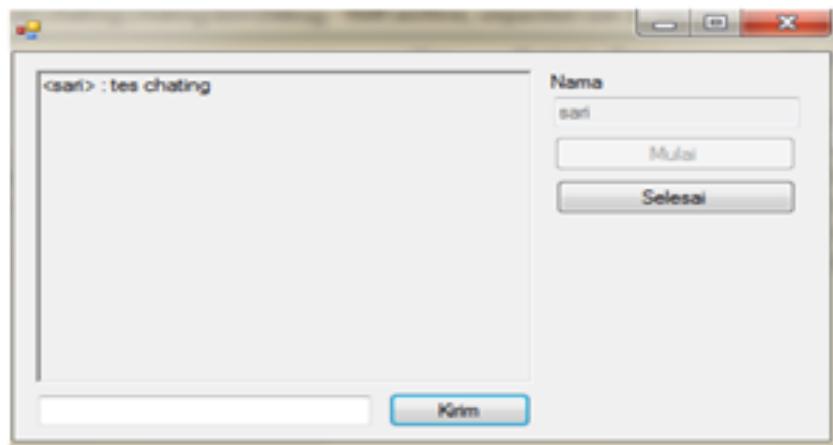
//fungsi yang mendeklarasikan ketikan tombol enter ditekan

```
private void bt_mulai_Click(object sender, EventArgs e)
{
    nama_user = Convert.ToString(tx_user.Text);
    broadcast=Convert.ToString(alamat_broadcast);
    pengirim_client = new UdpClient(broadcast, port);
    pengirim_client.EnableBroadcast = true;
    tx_kirim.Enabled = true;
    tx_user.Enabled = false;
    bt_mulai.Enabled = false;
    bt_kirim.Enabled = true;
    bt_selesai.Enabled = true;
}
```

//fungsi yang mendeklarasikan event klik pada button mulai

```
private void bt_selesai_Click(object sender, EventArgs e)
{
    tx_user.Enabled = true;
    tx_kirim.Enabled = false;
    bt_kirim.Enabled = false;
    bt_mulai.Enabled = true;
    bt_selesai.Enabled = false;
}
```

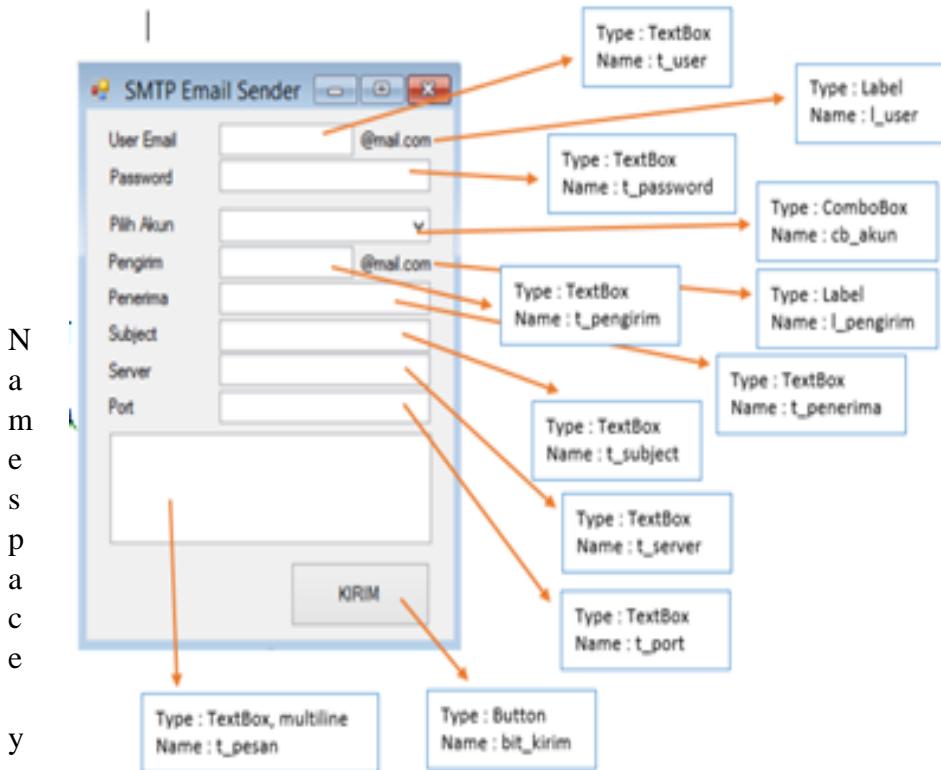
//fungsi yang mendeklarasikan event klik pada button selesai



b) Aplikasi SMTP email

Aplikasi Email Server ini menggunakan socket/port SMPT, Socket SMPT hanya mampu mengirim Email untuk menerima email, kita memerlukan socket POP3 untuk menerima email.

Desain lah form seperti dibawah ini



g akan digunakan adalah

```
using System.Net;
using System.Net.Mail;
using System.Text.RegularExpressions;
```

kemudian dalam class utama for tambahkan listing berikut

```
bool ssl_status = false;
```

list di atas digunakan untuk mendaftarkan “ssl_status” sebagai boolean dan memiliki default “false”

```
public Form1()
{
    InitializeComponent();

    this.StartPosition = FormStartPosition.CenterScreen;
    this.FormBorderStyle = FormBorderStyle.FixedSingle;
    this.MaximizeBox = false;
    this.MinimizeBox = false;
```

```

    t_password.PasswordChar = '*';
    cb_akun.Items.AddRange(new object[] {"GMAIL","YAHOO"});

    t_server.Enabled = false;
    t_port.Enabled = false;
}

```

Pada class public form1(), berisi listing yang akan dieksekusi pertama kali pada saat program dijalankan. Seperti formcenter screen, border yang tidak dapat di resize, menghilangkan tombol maximize dan minimize pada for, mendefinisikan textbox password menggunakan cahar “*”, dan menambahkan Item “gmail&yahoo” pada combobox akun, dan menonaktifkan textbox server dan Port.

```

private void cb_akun_SelectedIndexChanged(object sender, EventArgs e)
{
    if (cb_akun.SelectedIndex==0)
    {
        t_server.Text = "smtp.gmail.com";
        t_port.Text = "587";
        ssl_status = true;
        l_user.Text = "@gmail.com";
        l_pengirim.Text = "@gmail.com";
    }
    else if (cb_akun.SelectedIndex==1)
    {
        t_server.Text = "smtp.mail.yahoo.com";
        t_port.Text = "587";
        ssl_status = true;
        l_user.Text = "@yahoo.com";
        l_pengirim.Text = "@yahoo.com";
    }
}

```

Pada class cb_akun merupakan baris program yang berisi pilihan dari combobox akun, yaitu jika index yang terpilih adalah 0 (gmail) maka server dan konfigurasi yang digunakan adalah dari server gmail, sedangkan jika index Noya 1 (yahoo), maka server yang digunakan adalah milik dari yahoo.

```

private void bt_kirim_Click(object sender, EventArgs e)
{
    MailMessage pesan = new MailMessage(t_pengirim.Text+l_pengirim,
    t_penerima.Text,t_subject.Text, t_pesan.Text);
    SmtpClient client_email = new SmtpClient(t_server.Text,
    Convert.ToInt32(t_port.Text));

    client_email.Credentials = new NetworkCredential
    (t_user.Text+l_user.Text, t_password.Text);

    client_email.EnableSsl = ssl_status;
    client_email.Send(pesan);

    MessageBox.Show("Email Telah Dikirim", "Info");
}

```

Pada tombol “kirim” berisi perintah yang akan di eksekusi , pada baris “mailmessage” digunakan sebagai format pengiriman pesan yang berisi secara berurut (alamat pengirim, alamat penerima, subjek email, dan isi email). Sedangkan pada baris “smtpclient” merupakan baris program untuk memanggil Service smtp dari server penyedia layanan email yang berisi (alamat smtp server, Port yang digunakan). Pada credential digunakan untuk mengakses akun email yang digunakan dan pasword.

Full Coding

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Net;
using System.Net.Mail;
using System.Text.RegularExpressions;

namespace SMTPEmailSender
{
    public partial class Form1 : Form
    {
        bool ssl_status = false;
        public Form1()
        {
            InitializeComponent();
            this.StartPosition = FormStartPosition.CenterScreen;
            this.FormBorderStyle = FormBorderStyle.FixedSingle;
            this.MaximizeBox = false;
            this.MinimizeBox = false;

            t_password.PasswordChar = '*';
            cb_akun.Items.AddRange(new object[] { "GMAIL", "YAHOO" });
            t_server.Enabled = false;
            t_port.Enabled = false;
        }

        private void cb_akun_SelectedIndexChanged(object sender, EventArgs e)
        {
            if (cb_akun.SelectedIndex == 0)
            {
                t_server.Text = "smtp.gmail.com";
                t_port.Text = "587"; ssl_status = true;
                i_user.Text = "@gmail.com";
                i_pengirim.Text = "@gmail.com";
            }

            else if (cb_akun.SelectedIndex == 1)
            {
                t_server.Text = "smtp.yahoo.com";
                t_port.Text = "587"; ssl_status = true;
                i_user.Text = "@yahoo.com";
            }
        }
}
```

```
        i_pengirim.Text = "@yahoo.com";
    }

}

private void bit_kirim_Click(object sender, EventArgs e)
{
    MailMessage pesan = new MailMessage(t_pengirim.Text
+ i_pengirim.Text,
t_penerima.Text, t_subject.Text, t_pesanan.Text);

SmtpClient client_email=new SmtpClient (t_server.Text,
Convert.ToInt32(t_port.Text));
client_email.Credentials = new NetworkCredential(t_user.Text
+ i_user.Text, t_password.Text);
client_email.EnableSsl = ssl_status;
client_email.Send(pesan);
MessageBox.Show("Email Telah Dikirim","Info");
}
}
}
```

Tugas

1. Buatlah Aplikasi Chatting Berbasis Client Server dengan menggunakan Socket TCP/ IP (Bentuk pengiriman Tidak Broadcast).
2. Buatlah Penjelasan dari Listing yang dibuat.

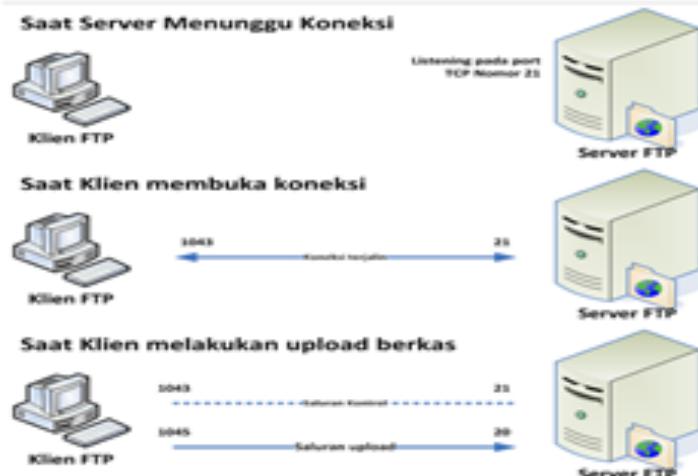


Lembar Kerja Praktik Network Programming II

PERTEMUAN XI FTP (FILE TRANSFER PROTOCOL)

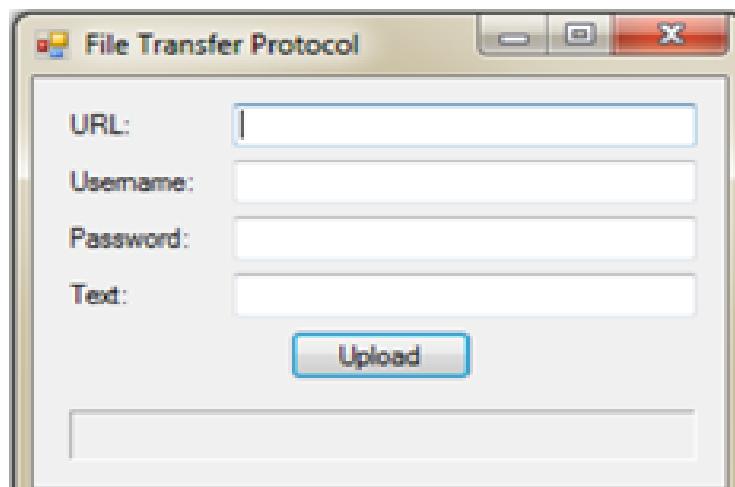
1. Pengertian FTP

FTP adalah suatu protocol yang berfungsi untuk pertukaran file dalam suatu jaringan komputer yang mendukung protocol TCP/IP. Dua hal pokok pada FTP yaitu FTP Server dan FTP Client, jadi fungsi FTP adalah melakukan pertukaran file dalam jaringan, FTP juga merupakan salah satu protocol internet yang berjalan di dalam level aplikasi yang standar untuk proses transfer file antar mesin komputer dalam sebuah framework.



1. Aplikasi FTP

Aplikasi ini untuk mentrasfer text pada sebuah sebuah website, aplikasi ini berbasis online .



Daftar Komponen

NO	Komponen	name	text
1	Label 1	Label 1	URL :
2	Label 2	Label 2	Username :
3	Label 3	Label 3	Password :
4	Label 4	Label 4	Text
5	Text box 1	-	txtUri
6	Text box 2	-	txtUsername
7	Text box 3	-	txtPassword
8	Text box 4	-	txtText
9	Button	Upload	btnUpload
10	Label 5		lblStatus

Kode Program :

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

using System.Net;
using System.IO;

namespace howto_ftp_upload_text
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        // Upload the text.
        private void btnUpload_Click(object sender, EventArgs e)
        {
            try
            {
                this.Cursor = Cursors.WaitCursor;
                lblStatus.Text = "Working...";
                Application.DoEvents();

                FtpUploadString(
                    DateTime.Now.ToString() + ":" + txtText.Text,
                    txtUri.Text, txtUsername.Text, txtPassword.Text);

                lblStatus.Text = "Done";
            }
            catch (Exception ex)
            {
                lblStatus.Text = "Error";
                MessageBox.Show(ex.Message);
            }
            finally
        }
    }
}
```

```
        {
            this.Cursor = Cursors.Default;
        }
    }

    // Use FTP to upload a string into a file.
    private void FtpUploadString(string text, string to_uri, string user_name, string password)
    {
        // Get the object used to communicate with the server. FtpWebRequest
        request = (FtpWebRequest)WebRequest.Create(to_uri); request.Method =
        WebRequestMethods.Ftp.UploadFile;

        // Get network credentials.
        request.Credentials = new NetworkCredential(user_name, password);

        // Write the text's bytes into the request
        stream. request.ContentLength = text.Length;
        using (Stream request_stream = request.GetRequestStream())
        {
            byte[] bytes = Encoding.UTF8.GetBytes(text);
            request_stream.Write(bytes, 0, text.Length);
            request_stream.Close();
        }
    }
}
```

Buat project/file baru kemudian desainlah seperti tampilan diatas. Setelah itu masuklah ke bagian code dan tambahkan namespace berikut ini :

```
using System.Net;  
using System.IO;
```

Namespace system.net berfungsi untuk mengambil semua informasi yang berhubungan dengan jaringan(lihat pembahasan sebelumnya) sedangkan namespace IO berfungsi sebagai input dan output file.kemudian double klik pada button upload dan ketikan perintah berikut :

```
try
{
    this.Cursor = Cursors.WaitCursor;
    lblStatus.Text = "Working...";
    Application.DoEvents();

    FtpUploadString(
        DateTime.Now.ToString() + ": " + txtText.Text,
        txtUri.Text, txtUsername.Text, txtPassword.Text);

    lblStatus.Text = "Done";
}
catch (Exception ex)
{
    lblStatus.Text = "Error";
    MessageBox.Show(ex.Message);
}
finally
{
    this.Cursor = Cursors.Default;
```

```
}
```

Kemudian buat method baru dengan nama FtpUploadString.

```
private void FtpUploadString(string text, string to_uri, string user_name,
string password)
{
    FtpWebRequest request = (FtpWebRequest)WebRequest.Create(to_uri);
    request.Method = WebRequestMethods.Ftp.UploadFile;

    request.Credentials = new NetworkCredential(user_name, password);

    request.ContentLength = text.Length;
    using (Stream request_stream = request.GetRequestStream())
    {
        byte[] bytes = Encoding.UTF8.GetBytes(text);
        request_stream.Write(bytes, 0, text.Length);
        request_stream.Close();
    }
}
```

FtpWebRequest digunakan untuk request kealamat ftp yang dituju. Sedang **request.Credentials = new NetworkCredential(user_name, password);** untuk login atau verifikasi penggunanya

FULL CODING

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Net;
using System.IO;

namespace SMTPEmailSender
{
    public partial class FTP : Form
    {
        public FTP()
        {
            InitializeComponent();
        }

        private void btnupload_Click(object sender, EventArgs e)
        {
            try
            {
                this.Cursor = Cursors.WaitCursor;
                lblstatus.Text = "Working...";
                Application.DoEvents();

                FtpUploadString(DateTime.Now.ToString() + ":" +
txtpassword.Text, txturl.Text, txtusername.Text, txtpassword.Text);
            }
        }
    }
}
```

```
        lblstatus.Text = "DONE";
    }

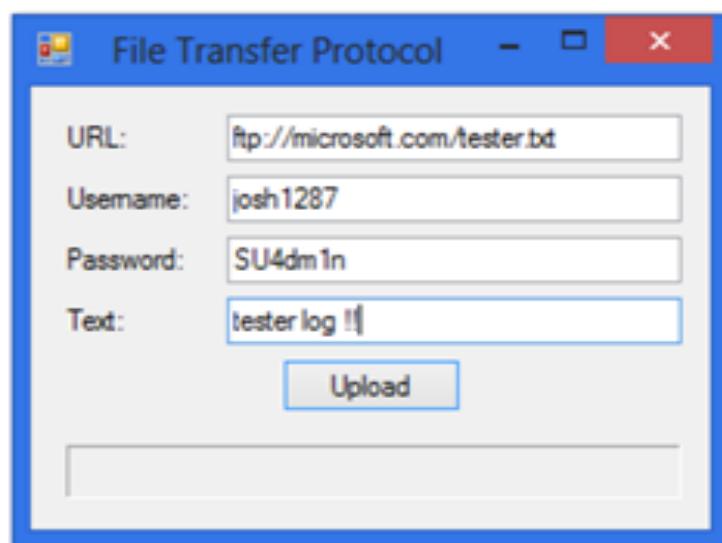
    catch (Exception ex)
    {
        lblstatus.Text = "ERROR";
        MessageBox.Show(ex.Message);
    }

    finally
    {
        this.Cursor = Cursors.Default;
    }
}

private void FtpUploadString(string text, string to_url, string user_name, string password)
{
    FtpWebRequest request = (FtpWebRequest)WebRequest.Create(to_url);
    request.Method = WebRequestMethods.Ftp.UploadFile;

    request.Credentials = new NetworkCredential(user_name, password);

    request.ContentLength = text.Length;
    using (Stream request_stream = request.GetRequestStream())
    {
        byte[] bytes = Encoding.UTF8.GetBytes(text);
        request_stream.Write(bytes, 0, text.Length);
        request_stream.Close();
    }
}
}
```





Lembar Kerja Praktik Network Programming II

PERTEMUAN XII-XVI TUGAS KELOMPOK & PRESSENTASI