

APLIKASI WEB SERVER BERBASIS BAHASA C SHARP

Rachmat Adi Purnama¹ Adi Tri Laksono Putra²

¹AMIK BSI TEGAL
e-mail : rachmat.rap@bsi.ac.id

²AMIK BSI Jakarta
e-mail : adi3man@gmail.com

Abstract— Seiring berkembangnya teknologi dalam pembuatan website, sebuah aplikasi yang dapat menjalankan suatu *Web Server*. Aplikasi itu harus mudah digunakan dan dapat digunakan di hampir semua peralatan. Tidak menggunakan memory yang cukup besar adalah suatu fitur yang harus dimiliki aplikasi *Web Server* agar *memory* yang tersedia tidak digunakan sia sia oleh server nya sendiri. Karena memori adalah salah satu hal penting yang berada di dalam sebuah server, jika *memory* penuh karena digunakan oleh sistem nya dengan sia sia, maka kebutuhan klien akan menjadi terhambat. Masalah yang biasanya muncul yaitu *Web Server* tidak dapat merespon permintaan klien karena sumber daya komputer server terlalu berat atau *Overload*. maka dibutuhkan aplikasi yang sangat ringan agar tidak membuat server berat. Aplikasi yang dibuat menggunakan bahasa c Sharp ini akan digunakan untuk menjalankan *Web Server* secara sederhana dan untuk mencoba sebuah web dapat berjalan secara bagus atau tidak, tanpa harus menginstall aplikasi web server yang biasanya berukuran besar dan agak sulit di install. Aplikasi ini berjalan di jaringan lokal seperti LAN dan WLAN

Keywords : *Web server, Aplikasi, c Sharp, LAN, WLAN*

Abstraction: As technology develops in website creation, an application that can run a Web Server. The application should be easy to use and can be used in almost any tool. Not using a large enough memory is a feature that must be owned by the Web Server application so that the available memory is not used in vain by its own server. Because memory is one of the important things that resides in a server, if the memory is full because it is used by his system in vain, then the client's needs will be hampered. The problem that usually arises is that the Web Server can not respond to client requests because the server computer resource is too heavy or Overload. So it takes a very lightweight application so as not to create a heavy server. Applications created using this Sharp c language will be used to run a Web Server simply and to try a web can run well or not, without having to install web server applications that are usually large and rather difficult to install. This app runs on local networks like LAN and WLAN

Keywords : *Web server, Aplikasi, c Sharp, LAN, WLAN*

BAB I. PENDAHULUAN

Internet sudah seperti kebutuhan pokok hampir setiap orang di dunia ini, khususnya mereka yang memiliki gadget dengan tamhanan layanan akses internet. Seperti tidak dipisahkan dari internet, Website seperti salah satu layanan yang harus ada pada internet. Kumpulan website dari seluruh website di dunia ini disebut *World Wide Web* atau yang biasanya disingkat menjadi *WWW*. Website di abad ke-21 ini sudah menjadi layanan utama untuk memperkenalkan produk, iklan, jasa, jual beli dan masih banyak lagi fungsi yang dapat dimanfaatkan oleh layanan ini.

Untuk menjalankan suatu website, pertama dibutuhkan koneksi internet yang bisa diakses oleh publik atau seluruh perangkat yang terkoneksi ke internet, kedua

yaitu server dari sebuah website, yang biasa disebut *web server*. Pemanfaatan *web server* menambah khasanah pengembangan aplikasi web seperti *elearning, egovernment* dan *ecommerce*, Yasin (2012). Dalam menjalankan web server dibutuhkan aplikasi untuk menampilkan dokumen dokumen tersebut. Dokumen dokumen website yang biasa digunakan berupa *HTML (Hyper Text Markup Language)*, *Bootstrap*, *CSS*, *Javascript*, *ASP.NET* dan masih banyak lagi.

Seiring berkembangnya teknologi dalam pembuatan website, sebuah aplikasi yang dapat menjalankan suatu *Web Server*. Aplikasi itu harus mudah digunakan dan dapat digunakan di hampir semua peralatan. Aplikasi tersebut tidak menggunakan *memory* yang cukup besar dalam suatu fitur yang harus dimiliki aplikasi *Web Server*,

sehingga *memory* yang tersedia tidak digunakan sia sia oleh server nya sendiri. Karena memori adalah salah satu hal penting yang berada di dalam sebuah server, jika *memory* penuh karena digunakan oleh sistemnya dengan sia sia, maka kebutuhan klien akan menjadi terhambat.

Masalah yang biasanya muncul yaitu *Web Server* tidak dapat merespon permintaan klien karena sumber daya komputer server terlalu berat atau *Overload*. Maka dibutuhkan aplikasi yang sangat ringan agar tidak membuat server berat.

Dalam pembuatan aplikasi web server ini, penulis menggunakan bahasa *C sharp*. Pengertian *C sharp* menurut Nakov (2013) adalah sebagai berikut : "*C# is a modern object-oriented, general-purpose programming language, created and developed by Microsoft together with the .NET platform. There is highly diverse software developed with C# and on the .NET platform: office applications, web applications, websites, desktop applications, mobile applications, games and many others.*"

Sebenarnya bahasa *C#* adalah gabungan yang kuat antara bahasa *C++* dan *Java*, tidak heran jika bahasa *C#* memiliki kemiripan diantara *C++* dan *Java*. Seperti yang dijelaskan oleh Miles (2016) tentang kemiripan bahasa pemrograman *C Sharp* "*C# bears a strong resemblance to the C++ and Java programming languages, having borrowed (or improved) features provided by these languages.*" Jika diterjemahkan dan disimpulkan berdasarkan pendapat Miles, bahasa *C Sharp* adalah bahasa gabungan yang kuat antara *C++* dan *Java*.

Bahasa *C#* tidak didistribusikan sebagai produk mandiri, *C#* bagian dari *Microsoft .NET Framework* platform (diucapkan "*Microsoft Dot Net Framework*") *.NET Framework* umumnya terdiri dari lingkungan untuk pengembangan dan pelaksanaan program, ditulis dalam bahasa *C #* atau yang lainnya Bahasa, kompatibel dengan *.NET* (seperti *VB.NET*, *Managed C ++*, *J #* atau *F #*). Yang terdiri dari :

1. Bahasa pemrograman *NET*. (*C #*, *VB.NET* dan lainnya);
2. Lingkungan untuk eksekusi kode yang dikelola, yang mana menjalankan program *C #* secara terkendali;
3. Satu set perpustakaan standar, seperti *ADO.NET*, yang memungkinkan akses ke *Database* (seperti *MS SQL Server* atau *MySQL*) dan *WCF* yang terhubung aplikasi melalui kerangka komunikasi str dan

protokol seperti soket *HTTP*, *REST*, *JSON*, *SOAP* dan *TCP*.

Pemrograman Berbasis Objek

Salah satu fungsi dari bahasa *C#* adalah mengolah data berorientasi objek, Menurut Nakov (2013) "*Object-oriented programming is the successor of procedural (structural) programming. Procedural programming describes programs as groups of reusable code units (procedures) which define input and output parameters. Procedural programs consist of procedures, which invoke each other*"

Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam teknik peranti lunak skala besar. Lebih jauh lagi, pendukung *OOP* mengklaim bahwa *OOP* lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya, dan pendekatan *OOP* lebih mudah dikembangkan dan dirawat. Di dalam pemrograman berorientasi objek dikenal istilah *class*, Menurut Nakov (2013) didalam bukunya mendefinisikan *class* sebagai berikut "*The class defines abstract characteristics of objects. It provides a structure for objects or a pattern which we use to describe the nature of something (some object). Classes are building blocks of OOP and are inseparably related to the objects.*"

Menurut Rosa dan salahudin(2013) metode berorientasi objek didasarkan pada penerapan prinsip-prinsip pengelolaan kompleksitas, meliputi rangkaian aktifitas analisis berorientasi objek, perancangan berorientasi objek, pemrograman berorientasi objek dan pengujian berorientasi objek.

Sedangkan Objek itu sendiri Menurut Miles (2016) "*Objects are an organisational mechanism which let you break your program down into sensible chunks, each of which is in charge of part of the overall system.*"

METHOD

Menurut Nakov (2013) "*A method is a basic part of a program. It can solve a certain problem, eventually take parameters and return a result*". Metode mewakili semua konversi data yang dilakukan program, untuk menyelesaikan sebuah tugas tertentu, metode terdiri dari logika program. Apalagi mereka tempat dimana "pekerjaan yang sebenarnya" selesai. Itulah sebabnya metode dapat diambil sebagai sebuah unit dasar untuk keseluruhan program. Di sisi lain, memberi kita Kesempatan, dengan

menggunakan blok sederhana, untuk membangun program yang lebih besar, yang dimana menyelesaikan masalah yang lebih kompleks dan canggih.

WEB SERVER

Menurut Yasin (2012) Web Service merupakan application logic yang dapat diakses dan dipublikasi menggunakan standard Internet (TCP/IP, HTTP, web). Web service dapat diimplementasikan pada lingkungan internal (intranet) untuk kebutuhan integritas antar sistem aplikasi (EAI =Enterprise Application Integration) ataupun pada lingkungan eksternal (internet) untuk mendukung aplikasi business-to-business (e-business)

Dan tidak hanya itu, tidak peduli di mana situs web host di dunia, halaman akan ditampilkan di layar komputer. *Web Server* selalu terhubung ke *internet*. Setiap *Web Server* yang terhubung ke internet akan dilengkapi dengan alamat unik yang disusun dengan serangkaian empat nomor antara 0 dan 255 yang dipisahkan oleh periode. Selain itu, *Web Server* memungkinkan penyedia hosting mengelola beberapa domain (pengguna) di server tunggal.

Cara kerja web server :

1. *Web server* merupakan mesin dimana tempat aplikasi atau *software* beroperasi dalam mendistribusikan web page ke user, tentu saja sesuai dengan permintaan user.
2. Hubungan antara *Web Server* dan *Browser Internet* merupakan gabungan atau jaringan Komputer yg ada di seluruh dunia. Setelah terhubung secara fisik, *Protocol TCP/IP (networking protocol)* yg memungkinkan semua komputer dapat berkomunikasi satu dengan yg lainnya. Pada saat *browser* meminta data *web page* ke *server* maka instruksi permintaan data oleh *browser* tersebut di kemas di dalam *TCP* yg merupakan *protocol transport* dan dikirim ke alamat yg dalam hal ini merupakan *protocol* berikutnya yaitu *Hyper Text Transfer Protocol (HTTP)*. *HTTP* ini merupakan *protocol* yg digunakan dalam *World Wide Web (WWW)* antar komputer yg terhubung dalam jaringan di dunia ini. Untuk mengenal *protocol* ini jelas sangat mudah sekali dimana setiap kali anda mengetik *http://...* anda

telah menggunakannya, dan membawa anda ke dunia internet. *Data* yg di *passing* dari *browser* ke *Web server* disebut sebagai *HTTP request* yg meminta *web page* dan kemudian *web server* akan mencari data *HTML* yg ada dan di kemas dalam *TCP protocol* dan di kirim kembali ke *browser*. *Data* yg dikirim dari *server* ke *browser* disebut sebagai *HTTP response*. Jika data yg diminta oleh *browser* tidak ditemukan oleh si *Web server* maka akan menimbulkan masalah yg sering anda lihat di web page yaitu *Error : 404 Page Not Found*

Hal ini memberikan cita rasa dari suatu proses yang tridimensional, artinya pengguna *internet* dapat membaca dari satu dokumen ke dokumen yang lain hanya dengan mengklik beberapa bagian dari halaman-halaman dokumen (*web*) itu. Proses yang dimulai dari permintaan *webclient (browser)*, diterima *web server*, diproses, dan dikembalikan hasil prosesnya oleh *web server* ke *web client* lagi dilakukan secara transparan. Setiap orang dapat dengan mudah mengetahui apa yang terjadi pada tiap proses. Secara garis besarnya *web server* hanya memproses semua masukan yang diperolehnya dari *web clientnya*.

II. METODE PENELITIAN

Metode untuk mendapatkan data dan teori pendukung untuk memperkuat data dan teori yang digunakan antara lain, yaitu :

a. Wawancara

Melakukan tanya jawab kepada beberapa staf IT dibidang perangkat lunak dan Jaringan untuk mendapatkan gambaran tentang pembahasan dan solusi dalam mengatasi permasalahan yang dihadapi.

b. Observasi

Penulis melakukan pengamatan langsung terhadap para pengguna layanan internet dan melihat sejauh mana kendala-kendala yang dialami oleh para pengguna jaringan tersebut.

c. Studi Pustaka

Untuk mendukung dan menambah bahan kajian, penulis mencari beberapa referensi baik melalui E-book maupun buku-buku referensi di perpustakaan.

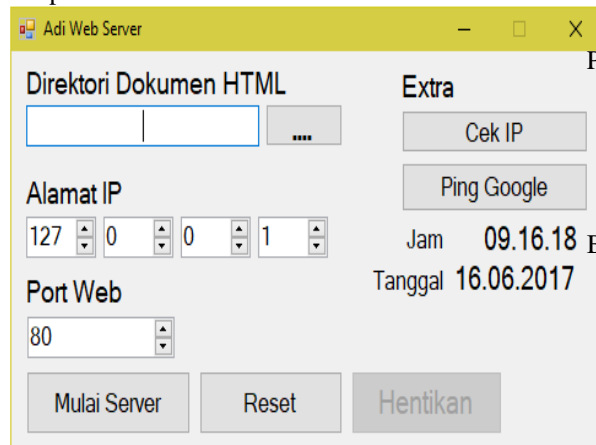
III. HASIL DAN PEMBAHASAN

Aplikasi web server ini bisa dimanfaatkan jika sedang membuat sebuah website. Sehingga perubahan website bisa langsung dilihat secara langsung atau *real time* jika website itu dijalankan di sebuah komputer atau laptop sebelum dijalankan di server langsung. Aplikasi ini sangat berguna untuk seorang pemula yang ingin membuat sebuah server dengan aplikasi yang mudah untuk dipakai.

Interface aplikasi ini pun tergolong terlihat mudah, pengguna hanya perlu memasukkan tempat dokumen HTML dengan menggunakan pemilihan *directory* tidak menggunakan *Command Line* seperti aplikasi serupa yang pada umumnya.

A Spesifikasi Rancangan Program

Rancangan Aplikasi Web Server hanya memiliki satu interface utama yang terdiri dari beberapa textbox dan tombol yang memiliki fungsinya sendiri. Terdapat juga tampilan Jam dan Tanggal untuk mengetahui waktu terkini. Program ini sangat ringan sehingga tidak membutuhkan media seperti gambar ataupun audio. Berikut ini adalah tampilan utama web server.



Sumber : Disain rancangan
 Gambar 2. Antarmuka Utama

Aplikasi Web Server hanya memiliki satu antarmuka utama, agar beban aplikasi tidak terlalu berat. Namun satu antarmuka utama itu sudah cukup untuk memaksimalkan kinerja Web Server ini. Adapun spesifikasi program aplikasi web server sebagai berikut :

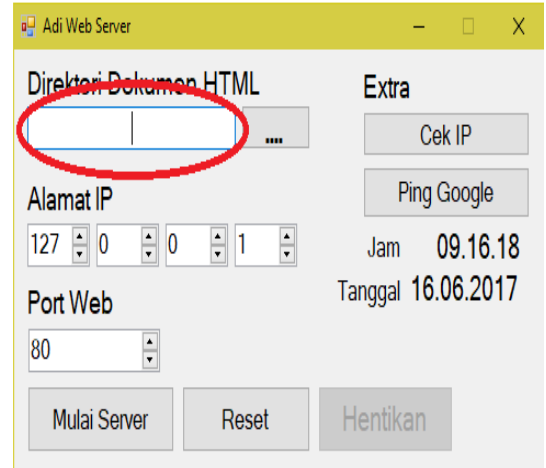
1. *Input Directory*

Nama Program : `textBox1`

Fungsi : Sebagai Input Direktori HTML Setiap Ingin Memulai dan Mengganti Halaman Web dengan yang baru.

Proses : 1. Buka folder yang digunakan sebagai halaman Web.
 2. Tuliskan lokasi direktori nya didalam box ini.

Bentuk :



Gambar .3 Input Direktori

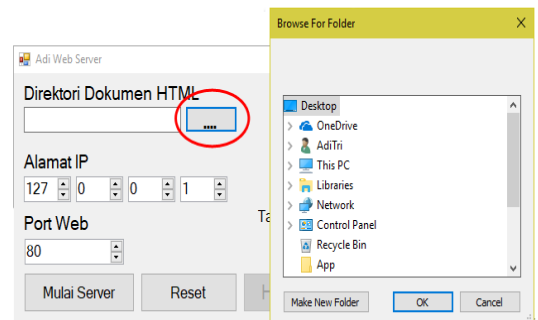
2. *Input Directory GUI*

Nama Program : `button1`

Fungsi : Sebagai Input Direktori HTML dengan GUI setiap ingin Memulai dan mengganti halaman Web dengan yang baru.

Proses : 1. Klik tombol button1.
 2. Setelahnya akan muncul GUI untuk memilih lokasi folder HTML yang akan digunakan sebagai halaman web.
 3. Jika sudah dipilih maka klik Ok

Bentuk :



Gambar .4. Input Direktori GUI

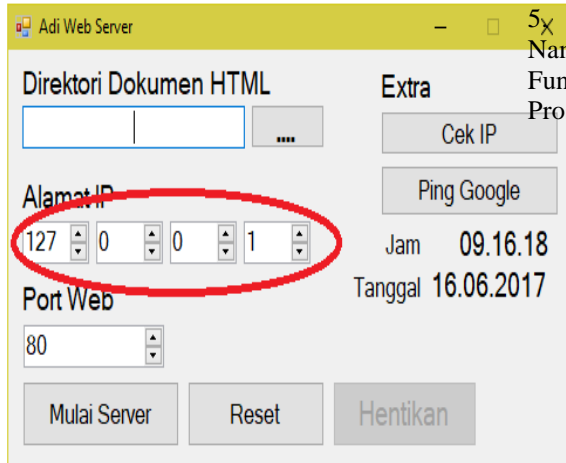
3. *Alamat IP*

Nama Program : `numericUpDown1-4`

Fungsi : Menentukan IP Address setiap ingin memulai server

Proses : 1. Masukkan alamat IP yang ingin digunakan untuk alamat untuk website yang ingin dijalankan

Bentuk :



Gambar .5. Input Alamat IP

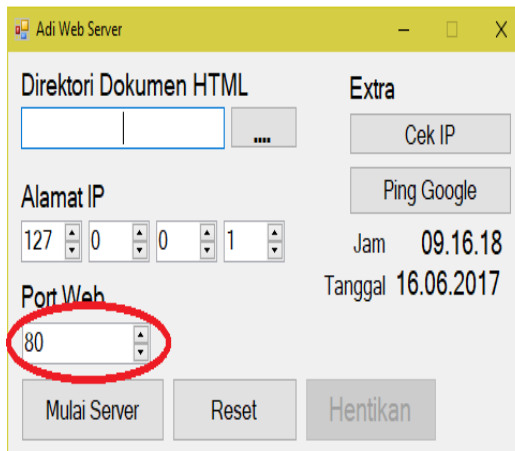
4. *Port Web*

Nama Program : numericUpDown5

Fungsi : Menentukan Port setiap ingin memulai server

- Proses :
1. Masukkan Port yang ingin digunakan untuk menjalankan website. Jika port default sudah digunakan oleh aplikasi lain, maka coba menggunakan port yang lain.
 2. Masukkan IP 127.0.0.1 jika hanya untuk jaringan lokal saja.
 3. Namun jika ingin menampilkan ke client yang terhubung ke jaringan yang sama, masukkan IP address yang sedang digunakan sesuai dengan jaringan.

Bentuk :



Gambar .6. Input Alamat Port Web

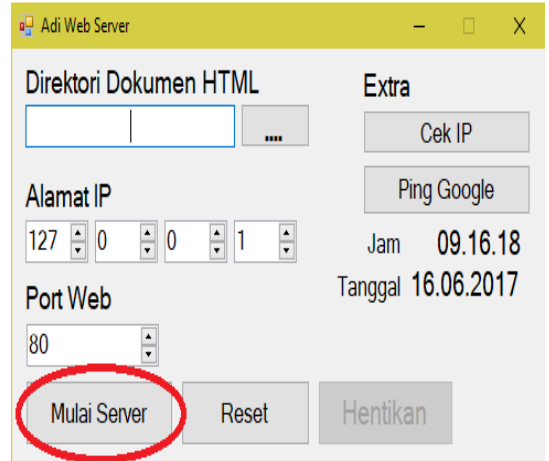
Mulai Server

Nama Program : btnMulai

Fungsi : Untuk menjalankan server

- Proses :
1. Pastikan textBox1 sudah terisi dengan lokasi dokumen HTML yang akan dijalankan sebagai halaman Web, karena jika tidak terisi server tidak akan bisa dijalankan.
 2. Pastikan juga alamat IP dan Port Web juga sudah terisi.
 3. Jika sudah server siap dijalankan.
 4. Buka web browser dan masukkan alamat IP untuk melihat halaman web yang dijalankan

Bentuk :



Gambar III.7 Mulai Server

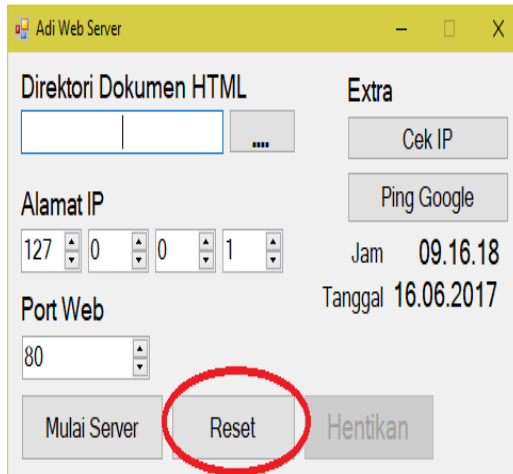
6. *Reset*

Nama Program : btnReset

Fungsi : Menghentikan server sekaligus menghapus semua input yang berada pada input HTML, alamat IP dan port web

- Proses :
1. Klik tombol reset jika ingin menghapus semua input yang sudah ada pada textBox1, IP address dan port web.
 2. Fungsi ini dapat dijalankan sebelum dan sesudah memulai server.
 3. Jika menggunakan tombol ini pada saat server sudah dijalankan, maka server akan terhenti dan semua input juga akan terhapus. Pengguna harus memasukkan kembali alamat direktori, IP address, dan port web dari awal lagi.

Bentuk :



Gambar.8.Reset

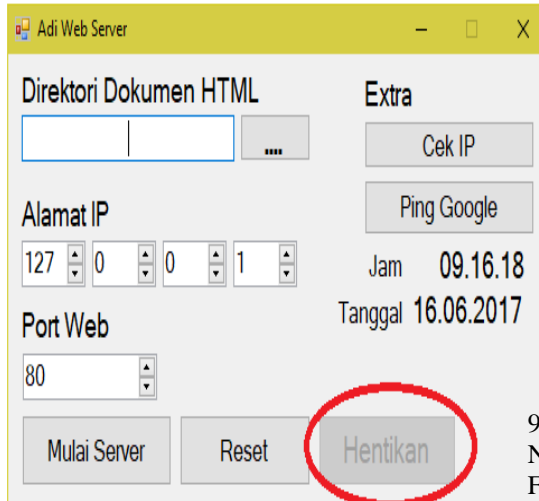
7. Hentikan

Nama Program : btnStop

Fungsi : Untuk menghentikan server yang sedang berjalan

- Proses:
1. Fungsi ini hanya untuk menghentikan server yang sedang berjalan saja
 2. Tidak dapat digunakan jika server tidak berjalan. Karena tidak akan ada gunanya jika tombol ini bisa digunakan jika server belum berjalan.

Bentuk :



Gambar .9. Hentikan

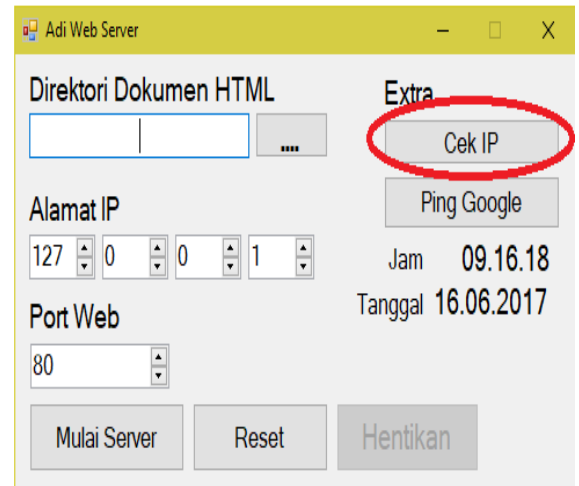
8. Cek IP

Nama Program : button5

Fungsi : Untuk mengecek IP yang sedang digunakan

- Proses :
1. Pastikan komputer/laptop sudah terhubung ke sebuah jaringan untuk mengecek IP Address.

Bentuk :



Gambar .10.Cek IP

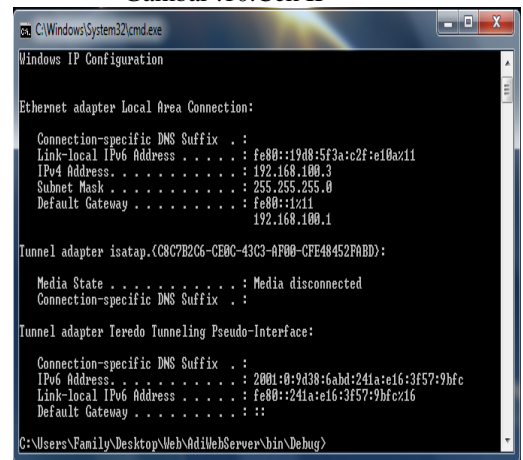
9. Ping Google

Nama Program : button8

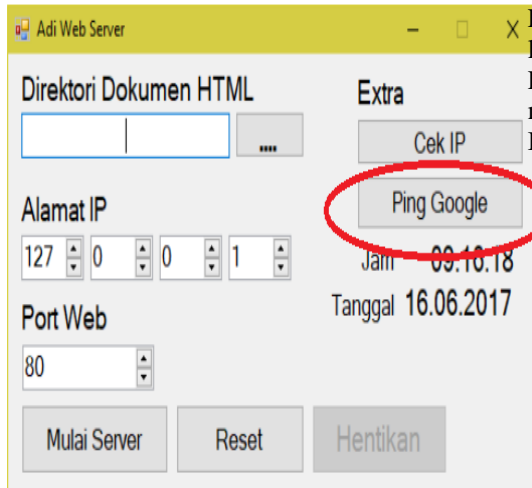
Fungsi : Melakukan ping ke alamat google.com

- Proses :
1. Klik untuk membuat ping ke google.com
 2. Proses ini digunakan untuk mengecek koneksi ke internet.
 3. Jika terhubung akan muncul pesan "reply from"
 4. Jika tidak terhubung ke internet maka akan muncul pesan "Request Timed Out"

Bentuk :

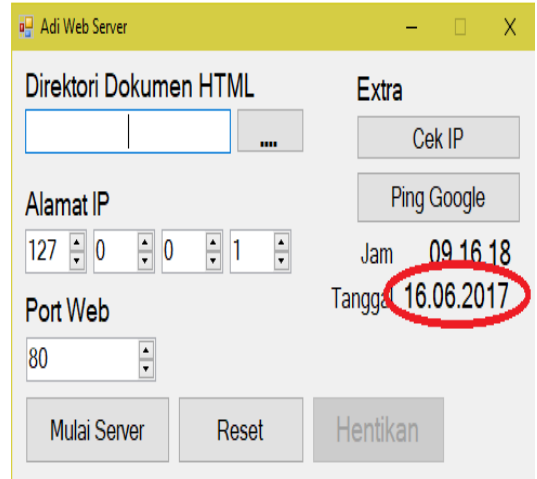


Gambar .11. Tampilan Cek IP

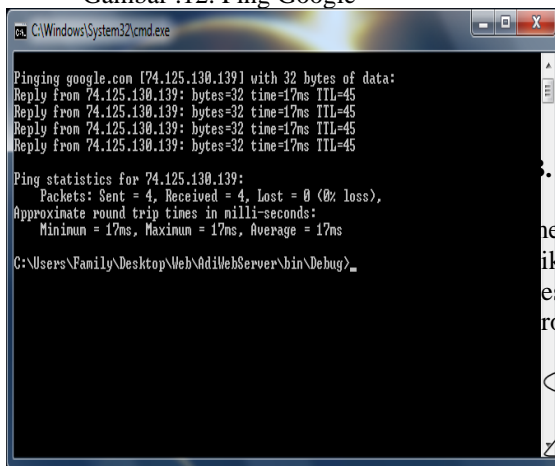


Gambar .12. Ping Google

Fungsi : Menampilkan tanggal terkini secara langsung
 Proses : Berjalan secara otomatis dan terus menerus
 Bentuk :

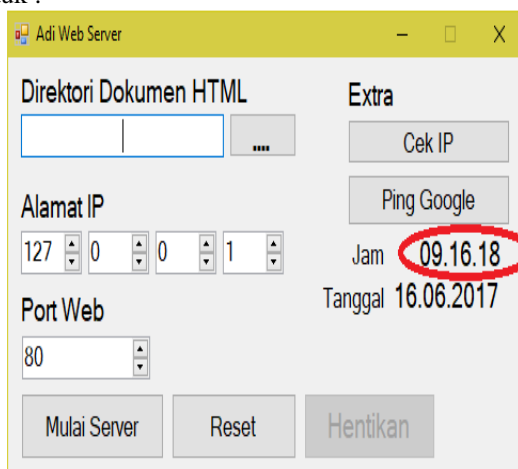


Gambar .15. Tampilan Tanggal



Gambar .13. Tampilan Ping Google

10. Jam
 Nama Program : timer1, label5
 Fungsi : Menampilkan jam terkini secara langsung
 Proses : Berjalan secara otomatis dan terus menerus
 Bentuk :

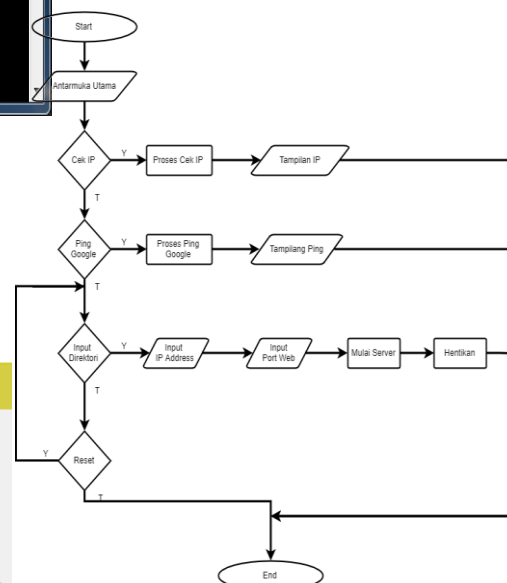


Gambar .14. Tampilan Jam

11. Tanggal
 Nama Program : timer2, label6

3. Flowchart

Flowchart merupakan bagan yang menunjukkan alur kerja atau apa yang sedang dikerjakan di dalam sistem secara keseluruhan dan menjelaskan urutan dari prosedur-prosedur yang ada di dalam sistem.



Gambar III.16
 Flowchart Sistem

LIST PROGRAM UTAMA

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.IO;
using System.Net;
using System.Threading;
using System.Net.NetworkInformation;
namespace AdiWebServer
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            Server server = new Server();
            private void button1_Click(object sender,
            EventArgs e)
            {
                using (var fbd = new
                FolderBrowserDialog())
                {
                    DialogResult result =
                    fbd.ShowDialog();

                    if (result == DialogResult.OK &&
                    !string.IsNullOrEmpty(fbd.SelectedPath))
                    {
                        Directory.GetDirectories(fbd.SelectedPath);
                        textBox1.Text =
                        fbd.SelectedPath;
                        /*MessageBox.Show("Files
                        found: " + files.Length.ToString(), "Message");*/
                    }
                }
                private void button4_Click(object sender,
                EventArgs e)
                {
                    textBox1.Text = "";
                    numericUpDown1.Value = 127;
                    numericUpDown2.Value = 0;
                    numericUpDown3.Value = 0;
                    numericUpDown4.Value = 1;
                    numericUpDown5.Value = 80;
                    server.stop();
                    btnMulai.Enabled = true;
                    btnStop.Enabled = false;
                }

                private void button3_Click(object sender,
                EventArgs e)
                {
                    if (textBox1.Text.Length <= 0)
                    {
                        MessageBox.Show("Harus diisi
                        !", "Info");
                    }
                    else
                    {
                        IPAddress ipAddress =
                        IPAddress.Parse(numericUpDown1.Value.ToStri
                        ng() + "." + numericUpDown2.Value.ToString() +
```

```
." + numericUpDown3.Value.ToString() + "." +
                    numericUpDown4.Value.ToString());
                    if (server.start(ipAddress,
                    Convert.ToInt32(numericUpDown5.Value), 100,
                    textBox1.Text))
                    {
                        btnMulai.Enabled = false;
                        btnStop.Enabled = true;
                        //btnTest.Enabled = true;
                    }
                    else
                    {
                        MessageBox.Show(this,
                        "Tidak dapat memulai Server! Pastikan Port " +
                        numericUpDown5.Value + " Tidak sedang
                        digunakan oleh aplikasi lain", "Error",
                        MessageBoxButtons.OK,
                        MessageBoxIcon.Error);
                    }
                }
                private void
                numericUpDown1_ValueChanged(object sender,
                EventArgs e)
                {
                    if (textBox1.Text == "")
                        numericUpDown1.Enabled =
                        false;
                    else
                        numericUpDown1.Enabled =
                        true;
                }
                private void button5_Click(object sender,
                EventArgs e)
                {
                    System.Diagnostics.Process.Start("CMD.exe",
                    "/K ipconfig");
                }
                private void label5_Click(object sender,
                EventArgs e)
                {
                }
                private void button8_Click(object sender,
                EventArgs e)
                {
                    System.Diagnostics.Process.Start("CMD.exe",
                    "/K ping google.com");
                }

                private void button2_Click(object sender,
                EventArgs e)
                {
                }
                private void button2_Click_1(object
                sender, EventArgs e)
                {
                    server.stop();
                    btnMulai.Enabled = true;
                    btnStop.Enabled = false;
                }
            }
        }
    }
}
```



```
private void
numericUpDown5_ValueChanged(object sender,
EventArgs e)
{
private void label5_Click_1(object
sender, EventArgs e)
{
private void timer1_Tick(object sender,
EventArgs e)
{
label5.Text =
DateTime.Now.ToString("HH:mm:ss");
}
private void timer2_Tick(object sender,
EventArgs e)
{
label6.Text =
DateTime.Now.ToString("dd:MM:yyyy");
}

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

IV KESIMPULAN

Berdasarkan pembahasan di atas, maka penulis dapat mengambil beberapa kesimpulan sebagai berikut :

- a. Web Server merupakan salah satu komponen utama yang penting dalam membangun sebuah website. Tanpa adanya web server, sebuah website tidak akan tersedia bagi pengguna internet.
- b. Aplikasi yang dijalankan di server harus memiliki beban ringan sehingga tidak mempengaruhi kinerja server itu sendiri.
- c. Aplikasi server juga harus *universal* atau dapat diakses dalam sistem operasi yang luas. Dan mudah didapatkan.
- d. Aplikasi server juga mudah dioperasikan dan mudah dimengerti.
- e. Aplikasi ini dapat digunakan sebagai server maupun sebagai sarana latihan membangun server.
- f. Aplikasi ini hanya memiliki satu interface utama karena faktor beban yang akan digunakan. Jika terlalu banyak bebannya, maka server tidak akan berjalan secara optimal.
- g. Setiap aplikasi Web Server juga memiliki kelebihan dan kekurangan masing masing.

REFERENSI

- Miles, Rob. 2016. C# Programming Yellow Book. Cheese Edition.
- Nakov , Svetlin. 2009 FUNDAMENTALS OF COMPUTER PROGRAMMING WITH C#. SoftUni. Bulgaria
- Rosa dan salahudin. 2013. Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek. Informatika.: Bandung
- Suryadi, Ucu. 2012. Pengertian Web Server diambil dari <http://ucu.suryadi.my.id/tkj/pengantar-webserver.html> (20-07-2017)
- Yasin, Verdi, 2012. Rekayasa Perangkat Lunak Berorientasi Objek Pemodelan, Arsitektur dan Perancangan. Mitra Wacana Media: Jakarta.