

**MAKALAH SEMINAR TUGAS AKHIR
PERANCANGAN WEB SERVER
MENGUNAKAN BAHASA PEMROGRAMAN PYTHON 2.3**

ABSTRAK

Dalam dunia komputer dan informasi saat ini mengalami perkembangan yang sangat pesat, dalam hal ini Internet mempunyai pengaruh yang sangat besar. Beberapa tahun yang lalu informasi umumnya hanya diperoleh melalui media massa baik yang bersifat cetak seperti surat kabar atau buku maupun media elektronik seperti televisi dan radio. Dengan Internet kita dapat mencari hampir semua informasi yang kita butuhkan dan tidak terbatas oleh waktu. Pengguna Internet melalui web browser dapat meminta suatu informasi pada Internet, oleh karena itu web server bertanggung jawab menyediakan informasi yang disimpan dalam suatu media, untuk melayani setiap informasi yang dibutuhkan oleh web browser. Web server atau lebih tepatnya world wide web server merupakan server Internet yang mampu melayani koneksi transfer data dalam protokol HTTP(Hyper Text Transfer Protocol). Web server dirancang untuk dapat melayani beragam jenis data, baik text, hypertext, gambar maupun suara. Tetapi pada umumnya web server melayani data dalam bentuk file HTML(Hypertext Markup Language)

Tugas akhir ini membuat perancangan web server menggunakan Python 2.3 dengan ukuran yang kecil dan mampu melayani permintaan informasi web browser baik gambar ataupun file HTML.

Perancangan system pada tugas akhir ini menggunakan flowchart untuk menjelaskan setiap aliran proses yang dibuat dalam pemrograman Python 2.3 dimana bahasa pemrograman python menyediakan modul-modul built-in yang mendukung pembuatan Tugas Akhir ini.

I. PENDAHULUAN

1.1. Latar Belakang

Dunia komputer dan informasi saat ini mengalami perkembangan yang sangat pesat, dalam hal ini Internet mempunyai pengaruh yang sangat besar. Beberapa tahun yang lalu informasi umumnya hanya diperoleh melalui media massa baik yang bersifat cetak seperti surat kabar atau buku maupun media elektronik seperti televisi dan radio. Dengan Internet kita dapat mencari hampir semua informasi yang kita butuhkan dan tidak terbatas oleh waktu. Pengguna Internet melalui *web browser* dapat meminta suatu informasi pada Internet, oleh karena itu *web server* bertanggung jawab menyediakan informasi yang disimpan dalam suatu media. untuk melayani setiap informasi yang dibutuhkan oleh *web browser*. *Web browser* merupakan aplikasi untuk mencari dokumen-dokumen web, oleh karena itu untuk dapat menampilkan dokumen web pada komputer lain diperlukan software yang disebut *web server*.

Web server atau lebih tepatnya world wide web server merupakan server Internet yang mampu melayani koneksi transfer data dalam protokol HTTP (*Hyper Text Transfer Protocol*) dan menunggu koneksi dari port tertentu. Web server bertugas untuk menerima permintaan terhadap dokumen tertentu yang ditulis dalam format URL, kemudian mencari file yang sesuai dengan file pada sistem, membacanya kemudian mengirimkannya pada client yang memintanya. Web server dirancang untuk dapat melayani beragam jenis data, baik text, hypertext, gambar maupun suara, tapi pada umumnya web server melayani data dalam bentuk file HTML (*Hypertext Markup Language*).

Web server menggunakan bahasa pemrograman Python 2.3 ini diharapkan mampu melayani permintaan dokumen oleh web browser, baik statis maupun dinamis. Web browser ini juga mempunyai ukuran yang cukup kecil, sehingga mudah dibawa kemana-mana.

1.2. Tujuan

Tujuan dari pembuatan Tugas Akhir ini adalah:

“Merancang Web Server menggunakan python 2.3 yang mampu melayani permintaan informasi dari browser yang mendukung protokol HTTP dari web server yang sudah ada.”

1.3. Batasan Masalah

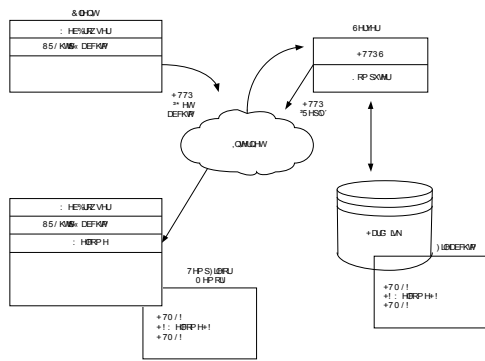
Pembatasan masalah pada penulisan tugas akhir ini sebagai berikut :

1. Program ini diterapkan untuk pembuatan web server yang sederhana.
2. Program ini menggunakan bahasa pemrograman Python.
3. Menggunakan Sistem Operasi Windows.
4. Menggunakan protokol HTTP 1.1.
5. Web browser yang digunakan adalah Internet Explorer, Opera dan Mozilla.

II. APLIKASI WEB SERVER MENGUNAKAN BAHASA PEMROGRAMAN PYTHON 2.3

2.1 Cara Kerja Web Server

Informasi web disimpan dalam dokumen yang disebut dengan halaman-halaman web (*web pages*). Web pages adalah file-file yang disimpan dalam komputer yang disebut dengan server-server web (*web servers*). Komputer-komputer membaca web page disebut sebagai web client. Web client menampilkan *page* dengan menggunakan program yang disebut dengan browser web (*web browser*). Browser web yang populer antara lain adalah Internet Explorer dan Netscape Navigator. Secara ringkas dapat dilihat pada Gambar 2.1:



Gambar 2.1 Skema WWW Bekerja

2.1.1 Web Browser

Browser web adalah *software* yang digunakan untuk menampilkan informasi dari server web. *Software* ini kini telah dikembangkan dengan menggunakan *user interface* grafis, sehingga pemakai dapat melakukan 'point dan click' untuk pindah antar dokumen.

Suatu browser mengambil sebuah *web page* dari server dengan sebuah *request*. Sebuah request adalah sebuah request HTTP standart yang berisi sebuah *page address*. Sebuah *page address* terlihat sebagai berikut : <http://www.yahoo.com>.

Seluruh *web page* berisi intruksi-intruksi bagaimana untuk ditampilkan. Browser menampilkan page dengan membaca intruksi-intruksi ini. Intruksi yang paling umum untuk menampilkan disebut dengan tag HTML. Tag HTML contohnya , yaitu `<p>` ini adalah paragraph

2.1.2. Web Server

Server web adalah komputer yang digunakan untuk menyimpan dokumen-dokumen web, komputer ini akan melayani permintaan dokumen web dari kliennya. Browser web seperti Explorer atau Navigator berkomunikasi melalui jaringan (termasuk jaringan Internet) dengan server Web, menggunakan HTTP. Browser akan mengirimkan *request* kepada server untuk meminta dokumen tertentu atau layanan lain yang disediakan oleh server. Server memberikan dokumen atau layanannya jika tersedia juga dengan menggunakan protocol HTTP.

Server adalah pemilik informasi yang menyediakan dirinya untuk memberikan servis atau layanan, sedangkan client adalah peminta layanan tersebut.^[14]

Sebuah web server adalah sebuah HTTP server^[3]. HTTP adalah protocol yang mendukung komunikasi antara web server dan web browser. HTTP mempunyai sebuah aturan sederhana yaitu ; client mengirim *request*, server mengembalikan jawaban. Pada HTTP server biasanya menggunakan port 80.

Selain web server masih banyak *utilitas* server yang lainnya, misalnya ftp server, mail server dan lain sebagainya. Kalau web server menangani permintaan untuk mengakses web, maka ftp server untuk menangani ftp (*file transfer protocol*), mail server menangani email, database server menangani database.

2.2 Protokol HTTP^[3]

Protocol HTTP adalah sebuah aturan atau protokol *request*(permintaan) dan *respon* (tanggapan).

Pesan HTTP berisi permintaan dari client ke server dan tanggapan dari server ke client. Format pesan permintaan adalah seperti Gambar 2.2 :

Request line	General header	Request header	Entity header	Message body
--------------	----------------	----------------	---------------	--------------

Gambar 2.2. Format request dari client

Sedangkan format respon dari server dapat dilihat seperti Gambar 2.3 :

Status line	General header	Response header	Entity header	Message body
-------------	----------------	-----------------	---------------	--------------

Gambar 2.3. Respon pesan dari server ke client

Komponen-komponen header ini memberikan informasi tentang server dan web browser.

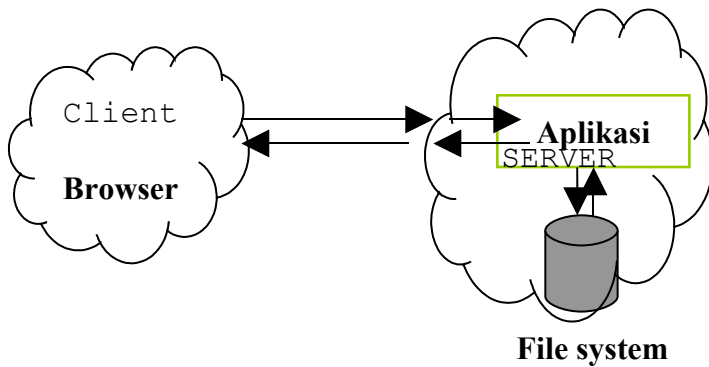
III. PERANCANGAN WEB SERVER MENGGUNAKAN BAHASA PEMROGRAMAN PYTHON 2.3

Perancangan web server ini menggunakan diagram alir untuk menjelaskan tahap demi tahap dalam pemrosesan pembuatan program. Web server pada dasarnya adalah HTTP server. Dan HTTP adalah sebuah protocol pada level aplikasi yang membutuhkan kecepatan dan ringan dalam mendistribusikan system informasi hypermedia. HTTP juga mendukung banyak perubahan baik pada request atau respon melalui sebuah koneksi tunggal TCP. Dalam perancangan ini menggunakan port 80.

Dalam komunikasi antara web server dengan web browser terdapat beberapa tahapan. Tahapan secara global komunikasi antara web browser/client dengan web server antara lain adalah :

1. Client melalui web browser meminta request pada web server berupa pesan/message
2. Server menanggapi request tersebut dengan mencari file yang ada dalam webroot
3. Server akan mengirimkan pesan berupa data yang diminta ataupun pesan error jika data yang diminta tidak ada.

Tahapan komunikasi antara web server dengan web browser diatas dapat digambarkan secara sederhana seperti Gambar 3.1.



Gambar 3.1 Diagram aplikasi web server

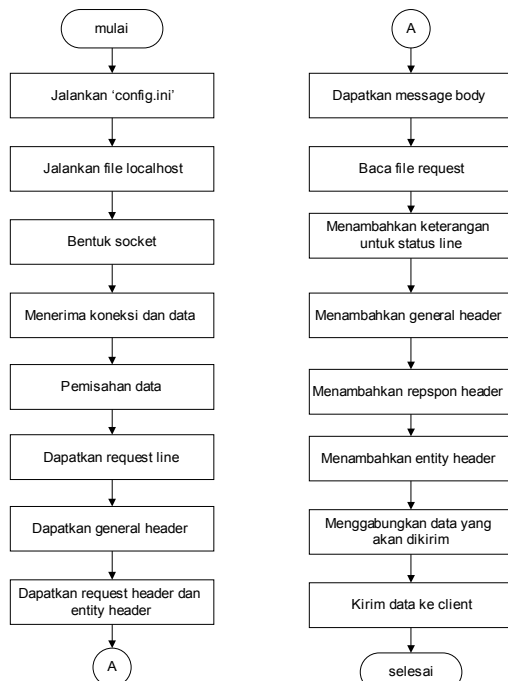
3.1. Diagram alir pada web server secara global

Aplikasi Server web dijalankan didalam konsol, tahapan-tahapan dalam aplikasi server ini dapat dilihat pada Gambar 3.2.

Pada saat program pertama kali dijalankan, akan menjalankan 'config.ini', setelah menampilkan pesan di server, program akan menjalankan file localhost sehingga akan muncul browser di komputer server. Kemudian dilanjutkan dengan pembentukan socket, disertai juga dengan koneksi dan penerimaan data permintaan dari client atau browser.

Pada data tersebut kemudian dipisah sesuai dengan format data request. Dimana data tersebut dipisah untuk mendapatkan Request-Line, General header, request header, entity header, dan message body.

Setelah data tersebut terpisah-pisah sesuai dengan formatnya, data request tersebut dibaca untuk dicari dalam weebroot. Setelah ketemu data yang dimaksud, data tersebut di sesuaikan dengan format response, yaitu dengan penambahan keterangan untuk status line, general header, respon header, dan entity header.



Gambar 3.2. Diagram alir web server secara global

Hasil penambahan tersebut diatas kemudian digabungkan semua untuk kemudian dikirimkan ke client atau web browser. Begitu data yang dikirim sudah terkirim atau di terima client maka server menutup socket dan akan kembali ke loop utama untuk menunggu permintaan koneksi dari client.

IV. IMPLEMENTASI DAN PENGUJIAN WEB SERVER DENGAN PYTHON 2.3

Implementasi tugas akhir ini adalah aplikasi web server dengan python. Pengujian diperlukan untuk mengetahui apakah aplikasi yang sudah dibuat sudah sesuai belum dengan apa yang diharapkan.

Perangkat keras yang digunakan dalam pembuatan dan pengujian program ini adalah :

1. AMD Athlon(tm) XP 2200+ 1.80 GHz sebagai server.
2. AMD Sempron(tm) 2200+ 1.49 GHz sebagai client.

Sedangkan untuk spesifikasi perangkat lunak yang digunakan adalah sebagai berikut:

1. Microsoft Windows XP pada sisi server dan client
2. Internet Explorer, Opera, dan Mozilla sebagai web browser pada sisi client
3. Python sebagai bahasa pemrogramannya

Adapun pengujian ini terdiri dari dua bagian, yaitu dari sisi server dan sisi client. Pada sisi client terdapat aplikasi yang dijalankan sedangkan pada sisi server akan menunjukkan aplikasi apa yang dijalankan.

4.1. Implementasi pada web server

Implementasi pada web server ini dibuat menggunakan bahasa pemrograman Python dengan tampilan menggunakan konsol. Seperti terlihat pada gambar 4.1. Pada bahasa pemrograman Python memiliki aturan pembacaan tersendiri. Untuk itu pada implementasi akan dijelaskan per blok sesuai alur program.

Pembuatan server diawali dengan menjalankan fungsi `startapp()` untuk menjalankan fungsi utama. Fungsi utama ini berfungsi untuk memberikan keterangan pada layer konsol serta menjalankan web browser pada server dengan fungsi `os.startfile('http://localhost')`. Setelah itu server akan membentuk socket dengan fungsi `socket.socket(socket.AF_INET, socket.SOCK_STREAM)`. Dilanjutkan dengan pengikatan port menggunakan fungsi `bind(('', Port))`. Kemudian dilanjutkan dengan pengaturan banyaknya client yang meminta koneksi dengan menggunakan fungsi `listen(5)`. Pemberian angka lima menyatakan jumlah maksimum koneksi yang akan dilayani.

Pemberian pesan error apabila terjadi kesalahan dalam fungsi socket, bind dan listen dilakukan oleh except, dan bila terjadi error akan keluar dari system dengan fungsi `sys.exit()`.

Apabila terjadi koneksi, koneksi tersebut akan dilayani dengan fungsi thread yang akan menjalankan program utama dan loop utama yang melayani permintaan dari client. Apabila pada saat yang

bersamaan terdapat koneksi baru maka akan ditangani oleh thread baru yang akan menjalankan loop utama yang sama dengan program sebelumnya. Begitu juga terdapat koneksi lebih dari dua dari client. Penanganan pembentukan socket sampai dengan penangan permintaan koneksi dari client dapat dilihat pada penggalan kode program dibawah ini :

```
def startapp():
    global
    serverSocket, running, Pekerja
    Pekerja=tPekerja()
    if verbose:
        print "\n", "*" * 80
        print "        ULO Web Server
version 1.0"
        print "        - Oleh Achmad
Rifky (L2F302451)"
        print "\n", "*" * 80
    running=1
    os.startfile('http://localhost')
    for Port in appPort:
        try:
            serverSocket=socket.soc
            ket(socket.AF_INET, sock
            et.SOCK_STREAM)
            serverSocket.bind((' ', Por
            t))
            serverSocket.listen(5)
        except:
            print 'Tidak bisa
menciptakan SOCKET ? Periksa jaringan
anda !'
```

```
            sys.exit()
    thread.start_new_thread(serverloop, (N
    one, serverSocket, Port))
```

Pada saat server berjalan fungsi startapp() ini juga menjalankan perintah untuk mematikan server dengan perintah pemencetan tombol q dilanjutkan tombol enter pada keyboard. Perintah keluar dengan fungsi exitweb(). Seperti penggalan program dibawah ini.

```
if verbose:
    while running:
        q=raw_input('->(q to quit)\n')
        if q=='q':exitweb()
```

Bila terdapat koneksi maka serverloop() akan dijalankan, pada verbose atau konsol akan dicetak keterangan “ :: Server dimulai pada port TCP 80.” kemudian akan masuk ke loop utama dengan menerima koneksi dari client dengan fungsi accept(). Setelah koneksi di terima maka data akan di ambil dengan fungsi recv(5096). Penerimaan lebar data sebanyak 5096 karakter. Setelah itu data akan diproses menggunakan fungsi process(data, port) .

Setelah data yang akan dikirim lengkap dengan header-header HTTP protocol maka data dikirim ke client atau ke web browser. Selanjutnya koneksi di tutup. Jika ada permintaan lagi dari client atau web browser maka akan kembali ke loop utama serverloop. Penggalan program untuk mengirim data response ke web browser dan menutup koneksi dapat dilihat pada penggalan program :

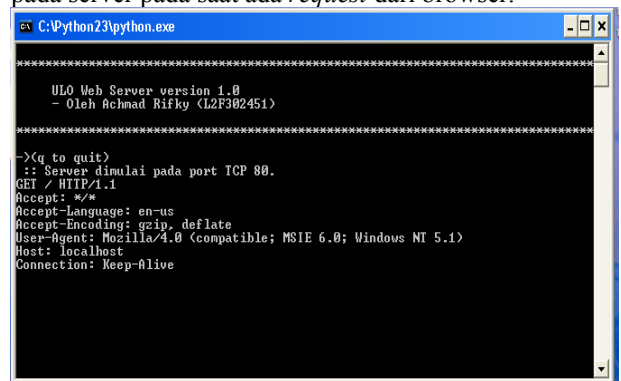
```
conn.send(res)
conn.close()
```

4.2. Pengujian

Pengujian dilakukan untuk mengetahui apakah program yang di buat sudah sesuai yang diharapkan atau tidak.

Pengujian terhadap web server menggunakan aplikasi web yang telah di buat untuk menguji tampilan server ketika mendapatkan request dari web browser dan response apa yang ditampilkan.

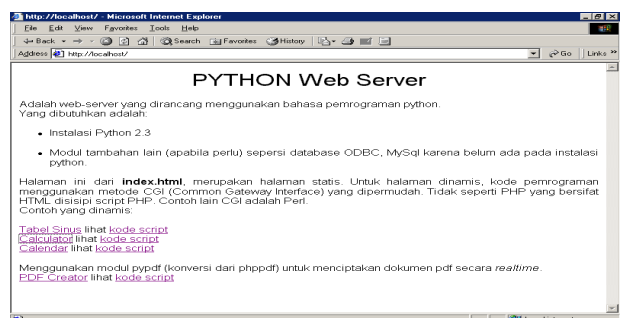
Gambar 4.1 menunjukkan tampilan utama pada server pada saat ada request dari browser.



Gambar 4.1 Tampilan utama pada server

Data yang muncul pada server mempunyai arti sebagai berikut :

- GET / HTTP/1.1 = merupakan request line
- accept : image/gif, image/x-xbitmap, image/jpeg, application/vnd.ms-excel, application /vnd.ms-powerpoint, application/vnd-msword,*/* menunjukkan request header.
- accept-Language: en-us merupakan request header.
- Accept-Encoding : gzip, deflate merupakan request header
- user-Agent : Mozilla/4.0 (compatible; msie 6.0; windows NT 5.1) merupakan request header
- host : 192.168.0.100 merupakan request header
- connection :keep-Alive :merupakan general header



Gambar 4.2. Tampilan utama pada browser client

Sedangkan response dari server ketika mendapatkan request gambar 4.2. adalah seperti terlihat pada gambar 4.3.

- [3] Fielding, R., *Hypertext Transfer Protocol/1.1*, Internet Society, 1995.
- [4] Hetland, Magnus Lie, *Beginning Python from Novice to Profesional*, Apress, Ney York, 2005.
- [5] Mitchell, Mark, Jeffrey Oldham, Alex Samuel. *Advanced Linux Programming*. New Riders Publishing, Indiana, 2001.
- [6] Newman, Frans, *Pemrograman Client/Server dengan ASP*, Elex Media Komputindo, Jakarta, 2001.
- [7] Noprianto, *Python dan Pemrograman Linux*, Andi, Yogyakarta, 2002.
- [8] Sidik, Betha, Ir, *Pemrograman WEB dengan PHP*, Informatika, Bandung, 2002.
- [9] Stevens, W. Richard, *Unix Network Programming, Network Applications : Sockets and XTI*, Prentice Hall, Inc, 1998.
- [10] Sugianto, David, Hapsoro Renaldy N, Muhammad F S, *Langkah Demi Langkah Membangun Website dengan PHP*, Datakom Lintas Buana, Jakarta, 2003.
- [11] Swaroop, *A Byte of Python*, www.python.g2swaroop.net. Desember 2006.
- [12] Tiponut, Sebastian V., *Python Network Programming*, Technical University Timisoara, 2001.
- [13] Utami, Ema, Suwanto Raharjo, *Logika Algoritman dan Implementasinya dalam Bahasa Python di GNU/Linux*, Andi, Yokyakarta, 2004.
- [14]--, *Pembuatan Program Sistem Informasi Akademik Berbasis ASP*, Salemba Infotek, Jakarta, 2003.

L2F 302451



Lahir pada tanggal 10 Desember 1978 di Kendal, Menempuh pendidikan di SD Negeri Pegandon lulus tahun 1991, melanjutkan ke SMP Negeri 1 Pegandon lulus tahun 1994. Melanjutkan ke SMU Negeri 1 Kendal lulus tahun 1997, kemudian menempuh pendidikan Diploma III di FNGT Undip lulus tahun 2001 dan sekarang sedang menyelesaikan studi

S1 pada Jurusan Teknik Elektro konsentrasi Informatika dan Komputer, Fakultas Teknik, Universitas Diponegoro Semarang.

Menyetujui dan mengesahkan

Pembimbing I

Agung Budi Prasetijo, ST, MIT
NIP. 132 137 932

Pembimbing II

Aghus Sofwan, ST, MT
NIP.132 163 757