RANCANG BANGUN APLIKASI *CLIENT-SERVER* BERBASIS PROTOKOL TCP/IP DENGAN PENDEKATAN PADA LAYER TRANSPORT

Oleh:

HERMANSYAH [1], IRZAMAN [2], BUDHI IRAWAN [3]

[1)] JURUSAN TEKNIK INFORMATIKA (10199089)

[2)] DOSEN INSTITUT PERTANIAN BOGOR

[3)] DOSEN TEKNIK INFORMATIKA

FAKULTAS TEKNIK DAN ILMU KOMPUTER

UNIVERSITAS KOMPUTER INDONESIA

BANDUNG

1. PENDAHULUAN

Penggabungan antara teknologi komputer dan komunikasi berpengaruh sekali terhadap bentuk organisasi sistem komputer. Konsep "pusat komputer" dalam sebuah ruangan yang berisi sebuah komputer besar, tempat dimana semua pemakai mengolah pekerjaannya, merupakan konsep yang sudah ketinggalan zaman. [K. Wahana , 2001].

Model komputer tunggal yang malayani tugas-tugas komputasi suatu organisasi telah diganti oleh sekumpulan komputer yang berjumlah banyak dan terpisah tetapi masih saling berhubungan dalam melaksanakan tugasnya. Sistem ini disebut sebagai Jaringan komputer (*Computer network*). [Salemba Infotek, 2003].

Pada dasarnya komunikasi data merupakan proses pengiriman informasi diantara dua titik menggunakan kode biner melewati saluran transmisi dan peralatan *switching*, bisa antara komputer dan komputer, komputer dengan terminal, atau komputer dengan peralatan, atau peratalan dengan peralatan. Awal tahun 1980 menjadi tonggak revolusi komunikasi data yang menjadikan jaringan komputer global sebagai media komunikasi yang kemudian berkembang pesat. Komputer di rumah dapat dihubungkan dengan internet untuk berkomunikasi dan mencari informasi ke seluruh dunia tanpa perlu pergi ke kantor pos atau perpustakaan, *electronic-teller-machine* dapat membantu agar para nasabah bank leluasa bertransaksi dimanapun tempatnya tanpa repot pergi ke kantor, bank, dan banyak lagi contoh lainnya. [B. Irawan, 2000].

Komputer dan peralatan pendukungnya pada dewasa ini telah menjadi sarana komunikasi yang cukup efisien dengan kemampuan pengolah data yang paling luas penggunaannya, maka tidak ada bentuk informasi yang tidak dapat ditangani oleh

komputer beserta perlengkapannya. Oleh karena itu sarana telekomunikasi dewasa ini akan mengarah kepada komunikasi data dengan komputer sebagai peran sentral, suatu saat komunikasi data dapat menggantikan sarana komunikasi konvensional yang ada saat ini. [B. Irawan, 2000].

Socket dapat dianggap sebagai suatu generalisasi mekanisme akses *file* UNIX yang menyediakan ujung akses untuk komunikasi. Sebagaimana halnya dengan akses *file*, program aplikasi meminta sistem operasi membentuk sebuah *Socket* jika diperlukan. Sistem mengembalikan suatu nilai intejer yang akan digunakan oleh program aplikasi untuk mengacu pada *Socket* yang baru saja dibentuk, yang disebut juga dengan *Socket descriptor.* Perbedaan antara *file descriptor* dengan *Socket descriptor* adalah dari *sistem call* yang dipanggil dan *target binding*. Sistem operasi menggandengkan (*bind*) *file descriptor* dengan suatu *file* atau *device* tertentu ketika program aplikasi memanggil *open*, sedangkan *Socket* dapat dibentuk tanpa menggandengkannya ke alamat tujuan tertentu. [D. E. Comer,2000].

Program aplikasi dapat memberikan alamat tujuan saat ia menggunakan *Socket*. Masih mengadopsi paradigma tradisional *open-read-write-close, Socket* dapat digunakan bersama dengan operasi tradisional *read* dan *write*. Agar sistem dapat mengakomodasi operasi *read* dan *write* baik untuk *file* maupun *Socket*, sistem akan mengalokasikan *descriptor* secara ekslusif untuk *file* yang tidak akan sama dengan *Socket*.

2. BATASAN MASALAH

Identifikasi masalah yang diambil dalam penulisan skripsi ini mengenai perancangan aplikasi *client-sever* untuk pengiriman data antara dua komputer (*Client-Server*) dengan berbasis pada protokol TCP/IP.

3. HASIL dan PEMBAHASAN

Flow chart aplikasi *client-server.*

1. Koneksi *Client*

   Server host memulai *Koneksi* dengan memasukkan IP *address* tujuan

2. Koneksi TCP

   Client membangun koneksi terhadap *server* melaui protokol TCP/UDP jika gagal koneksi tidak terjadi dan koneksi diakhiri.

3. Respon Keadaan

   Ketika koneksi TCP/UDP telah dibuka dengan *client*, *server* memberikan respons sambutan. Ini dapat diartikan suatu keadaan respons positif (“Koneksi”). contoh ‘S: +OK *server connect* ’ atau respons negative (“Putus”). sekarang berada dalam sesi *authorization state*. Ada tiga mekanisme yang ada pada sesi ini, yaitu dengar/mendengar, Koneksi, dan Koneksi

putus. koneksi, didalamnya terdapat pengisian IP *addres* komputer yang dituju apabila ada hubungan *request* dan *reply* akan berjalan baik putus atau koneksi putus adalah hubungan yang gagal akan ada pesan *error,*jika gagal menuju *state akhir*, dan *Listen* adalah status tunggu apakah hunbungan baik atau akan terjadi *error*.

4. Permintaan

   Ketika *client* telah berhasil mengidentifikasi dirinya terhadap *server*. Maka diantara *client-server* telah terjadi hubungan. Maka secara otomatis protocol TCP/UDP akan melakukan pencatatan terhadap kejadian yang berlangsung antara *client-server. Client* sekarang dapat melakukan permintaan.

5. Ambil Permintaan

   Setelah client berada pada *transaction state*, *client* kini dapat mengambil permintaan di*server* dan menampilkannya pada menu program aplikasi.

6. Load

   Load disini merupakan tempat mengambil, *file* gambar, dan *text* yang diperlukan dalam aplikasi *client-server* ini.

7. Pengiriman permintaan

   Ketika *client* memberikan permintaan, memasuki state ini. *Server* akan memberikan balasan pada proses ini TCP/UDP akan melakukan pencatatan dan memberikan informasi kejadian kepada user.

8. End

   Akhir dari koneksi atau salah satu sisi baik *server* atau client memutus koneksi TCP.

Ilustrasi kerja aplikasi *Client-Server*

1. *Server* memberi layanan permintaan
2. *Client* membangun koneksi dengan protokol TCP/UDP
3. User Authentikasi
   a. Login Koneksi dengan Komputer tujuan.
4. Tampilkan pengiriman text, gambar *clear*, permitaan hapus.

Komponen-komponen jaringan yang digunakan yaitu:

1. Kartu jaringan (*Ethernet Card*)
2. Kabel UTP
3. Konektor UTP tipe RG-45

Spesifikasi komputer yang penulis gunakan pada perancangan LAN adalah:

1. Komputer 1
   a. Processor : Intel Pentium III 800 MHz
   b. Bus Type : PCI
   c. NIC : Giga fast
   d. Komputer name : Gun

e. Workgroup : MSHOME

f. IP Address : 192.168.10.1

g. Subnet mask : 255.255.255.0

2. Komputer 2

a. Processor : Intel Pentium IV 2,4 GHz

b. Bus Type : PCI

c. NIC : Realtek RTL8139(A)-based PCI Fast Ethernet Adapter

d. komputer name : Hermansyah

e. Workgroup : MSHOME

f. IP Address : 192.168.10.3

g. Subnet mask : 255.255.255.0

Perancangan *software* aplikasi *client-server* pada jaringan TCP/IP diperlukan *software* yang harus di instalasi dan dikonfigurasi agar dapat menampilkan program aplikasi dan hasil inforemasi TCP/IP aplikasi *client-server* pada tipe jaringan *peer to peer* . *Software* yang diperlukan adalah:

1. Borland Delphi .

Instalasi Program *Client-Server.*

1. Instalasi program aplikasi *Clien-Server* dilakukan pada komputer *client* dan *server.*
2. Konfigurasi IP *address* sebagai berikut:

a. Subnet mask : 255.255.255.0

b. Work Group : MSHOME

c. User Name 1 : Athalan (Hermansyah)

d. User Name 2 : Coba (Guns)

Program aplikasi *Client-Server* ini dapat mengirimkan berbagai macam File, seperti : File image (gambar), *File* data, MP3, dll dengan ukuran *file* yang di kirim cukup besar pada saat pengiriman tidak membutuhkan waktu yang lama.

4. KESIMPULAN

1. Aplikasi *Client-Server* merupakan aplikasi yang berbasis jaringan. Untuk mendapatkan informasi TCP/IP mengenai permintaan dan balasan dalam aplikasi ini, tidak harus menggunakan *software sniffer* atau sejenisnya.
2. Program aplikasi *Client-server* ini di harapkan mampu bekeja lebih baik dari jaringan komputer tipe *peer to peer* biasa.
3. Untuk mendapatkan informasi TCP/IP dari suatu program aplikasi *client-server* harus selalu terkoneksi dengan protokol TCP/UDP agar protokol TCP/UDP dapat memberikan informasi kepada user.
4. Program aplikasi *client-server* dapat melayani permintaan apa saja dari *client* yang bersesuaian dengan layanan yang tersedia.
5. Informasi dengan protokol TCP memberikan informasi yang lengkap tidak seperti pada informasi protokol UDP.

6. Tidak semua jenis *file* gambar dapat ditampilkan pada saat terjadi pengiriman dari suatu proses permintaan dari *client*.
7. Tidak dapat melakukan pengiriman satu *file drive* atau lebih.
8. Program aplikasi *client-server* dalam melayani permintaan cukup cepat.

DAFTAR PUSTAKA

1. A. Kurniawan. *Pemograman jaringan internet dengan visual c++*. PT. ElexMedia Komputindo, 2000.
2. K.Jamsha, p.h.D. dan K. Cope. *Internet programming*, Jamsha Press, 1995.
3. K. Wahana. *Penanganan Jaringan Komputer*. Andi Offset, 2001.
4. D. Maryanto, *Pedoman Instalasi dan Penggunaan FreeBSD*. FreeBSD Maniac Press, 2000.
5. Salemba infotek, *Konsep Perancangan jaringan Komputer*, 2003.
6. A.Pranata. *Pemograman Borland Delphi 6.0*, edisi 4, 2003.
7. A. Edmunson.*Winsock*. *Winsock Faq*, 1999.
8. B. A. Forouzan,*TCP/IP Protocol Suite*. McGraw-Hill Inc, 2000.
9. A.S. Tanenbaum. *Jaringan Komputer Edisi Bahasa Indonesia*. Prehallindo, 1997.