

Design Of Parallel Processing Applications With The Remote Execution

Hero Wintolo

Department of Informatics

Sekolah Tinggi Teknologi Adisucipto Yogyakarta

Abstract

Each execution of the application, both of which became part of the operating system or the user's komputer, requiring the process to be done by the processor. Application execution can be done by borrowing another komputer processors connected to the komputer where the application is loaded by using the transmission medium in a komputer network remotely.

Remote process execution of an application requires a software designed to utilize a given socket port number. In this study the number of socket port 300 for sending messages and port number 3000 to send the application, each of which is on the server side and client side. The server side to provide command and file delivery applications, while more passive client side, receive and file the application and directly execute the file the application.

Software designed to perform remote process execution are then tested in a komputer network peer to peer, local area network (LAN) and Internet with the results showing the application server can send the messages and files to the client application through the transmission medium. Delivery of messages and files from the server to the client komputer can be done on condition that use such software has the Internet Protocol address (IP).

Keywords : Design, parallel processing, remote

1. Pendahuluan

Eksekusi sebuah aplikasi pada *Personal Komputer* (PC) sangat tergantung pada fungsi aplikasi tersebut. Jika fungsi melekat pada sistem operasi dari PC maka eksekusi menjadi sesuatu yang rutin dilakukan dan menjadi tanggung jawab dari sistem operasi tersebut. Tetapi, jika aplikasi itu menjadi bagian dari seseorang untuk menyelesaikan pekerjaan yang membutuhkan bantuan komputer, maka eksekusi menjadi tanggungjawab bagi pemilik aplikasi tersebut. Setiap eksekusi dari aplikasi, baik yang menjadi bagian dari sistem operasi atau milik pengguna komputer, membutuhkan proses yang akan dikerjakan oleh prosesor.

Pada aplikasi yang berbasis PC, eksekusi aplikasi akan menjadi tanggung jawab dari prosesor PC tersebut. Eksekusi aplikasi dapat dilakukan dengan meminjam prosesor komputer lainnya yang terhubung dengan komputer tempat aplikasi tersebut dipasangkan dengan menggunakan media transmisi dalam sebuah jaringan komputer. Peminjaman prosesor untuk eksekusi aplikasi secara *remote* dilakukan pada jaringan komputer yang digunakan untuk pemrosesan paralel.

Pemrosesan paralel membutuhkan perangkat lunak yang khusus, selain perangkat keras komputer dan jaringan komputer. Pada kasus tertentu, pemrosesan paralel menggunakan komputer mainframe atau superkomputer. Tetapi dengan harga komputer mainframe atau superkomputer yang sangat mahal, pemrosesan paralel dapat dilakukan dalam jaringan komputer *Local Area Network* (LAN) dengan menggunakan perangkat lunak yang dirancang secara khusus

untuk menyelesaikan masalah yang khusus pula. Permasalahan khusus membutuhkan perangkat lunak khusus, sehingga *remote* eksekusi aplikasi dapat dilakukan pada LAN.

2. Tinjauan Pustaka

Berdasarkan penelitian sebelumnya yang saya lakukan dengan judul “Deteksi Kinerja Prosesor Komputer Klien Dengan Cara Remote Untuk Mendukung Aplikasi Pemrosesan Paralel”, proses remote dari komputer dengan aplikasi server ke komputer yang beraplikasi klien dapat dilakukan dengan membuat socket yang dapat menghubungkan kedua komputer tersebut. Sehingga proses pengiriman data yang ukurannya lebih besar pasti dapat dilakukan dan media transmisi yang digunakan dapat mendukung terjadinya pengiriman dan eksekusi terhadap data tersebut.

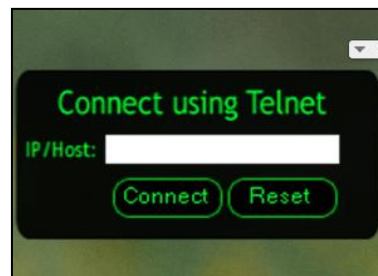
3. Landasan Teori

3.1 Remote Eksekusi

Remote yang berasal dari dua kata *remote control* dikenal dalam bidang ilmu dan teknologi elektronika yang diartikan sebagai sebuah alat yang digunakan untuk mengendalikan peralatan yang lainnya dengan jarak tertentu. Penggunaan peralatan yang digunakan sebagai pengendali ini dimaksudkan agar pergerakan manusia dalam pengendalian peralatan dapat dilakukan dengan jarak yang mempertimbangkan rasa aman dan nyaman bagi manusia. Dalam perkembangan teknologi elektronika yaitu dalam bidang komputer, penggunaan istilah *remote* ini juga diberlakukan pada keadaan dimana sebuah komputer dapat mengendalikan komputer melalui sebuah jaringan komputer yang menghubungkan kedua buah komputer tersebut. Peralatan yang digunakan untuk melakukan proses *remote* dalam bentuk perangkat lunak, yang prinsip kerjanya melakukan koneksi terhadap komputer yang akan dikenai proses *remote* dengan menggunakan *port* dan *socket* yang telah diatur dan disepakati oleh kedua komputer tersebut yang dapat dipahami oleh perangkat lunaknya, salah satunya adalah *telnet*.



a



b

Gambar 1(a) Peralatan elektronika *remote control* (b)Perangkat lunak telnet untuk melakukan proses *remote*

Perangkat lunak yang merupakan peralatan yang secara fisik tidak dapat disentuh ini menjadi bagian terpenting dari sebuah komputer, bahkan tanpa perangkat lunak sebuah komputer yang memiliki keunggulan dalam perangkat kerasnya menjadi tidak berguna. Salah satu perangkat lunak yang wajib dimiliki oleh sebuah komputer adalah sistem operasi, yang jenis bermacam-macam, mulai yang berbayar hingga yang *open source*. Dengan adanya sistem operasi, semua program atau aplikasi dapat dijalankan, hal ini disebabkan karena kinerja dari prosesor menjalankan jutaan instruksi, instruksi diberikan oleh program atau aplikasi dalam

bentuk proses yang diproses oleh prosesor, dan proses inilah yang dikenal sebagai eksekusi dalam sebuah *Personal Komputer* (PC).

Sistem operasi merupakan suatu perangkat lunak sistem yang bertanggung jawab untuk mengendalikan langsung dan mengelola perangkat keras dan pengoperasian dasar sistem, termasuk juga menjalankan bermacam aplikasi di atasnya misalnya pengolah kata atau perambah website. Secara umum, Sistem Operasi adalah *software* pada lapisan pertama yang ditaruh pada memori komputer pada saat komputer dinyalakan. Sedangkan *software-software* lainnya dijalankan setelah Sistem Operasi berjalan, dan Sistem Operasi akan melakukan layanan inti umum untuk *software-software* itu.

Layanan inti umum tersebut seperti akses ke disk, manajemen memori, skeduling task, dan antar-muka user. Sehingga masing-masing *software* tidak perlu lagi melakukan tugas-tugas inti umum tersebut, karena dapat dilayani dan dilakukan oleh Sistem Operasi.

Bagian kode yang melakukan tugas-tugas inti dan umum tersebut dinamakan dengan “kernel” suatu Sistem Operasi.

1. Sistem Operasi secara umum terdiri dari beberapa bagian:
Mekanisme *Boot*, yaitu meletakkan kernel ke dalam *memory*
2. *Kernel*, yaitu inti dari sebuah Sistem Operasi
3. *Command Interpreter* atau *shell*, yang bertugas membaca input dari pengguna
4. Pustaka-pustaka, yaitu yang menyediakan kumpulan fungsi dasar dan standar yang dapat dipanggil oleh aplikasi lain
5. *Driver* untuk berinteraksi dengan *hardware* eksternal, sekaligus untuk mengontrol mereka.

Eksekusi yang terjadi dalam sebuah PC dapat dilakukan pada PC yang lainnya, dengan syarat PC tersebut dihubungkan ke dalam sebuah jaringan komputer, jika hal ini telah dilakukan maka *remote* eksekusi dapat dilakukan dari sebuah PC ke PC lainnya yang terhubung dalam jaringan komputer yang berskala kecil, misal *Local Area Network* (LAN), hingga jaringan komputer berskala besar, misal internet.

3.2 Socket Programming

Pemrograman komputer merupakan cabang dari keilmuan bidang komputer *science*, dengan menggunakan program yang dibuat oleh seseorang yang dikenal dengan nama programmer, komputer yang merupakan susunan perangkat keras yang terorganisir secara rapi, dapat digunakan sesuai dengan kebutuhan manusia dalam menyelesaikan permasalahan. Karena program yang dibuat oleh seorang programmer akan menjadi aplikasi yang user friendly dan dapat dikembangkan sesuai dengan spesifikasi perangkat keras komputer.

Pemrograman socket termasuk dalam pemrograman komputer tingkat rendah. Pemrograman tingkat rendah disini diartikan sebagai pemrograman bahasa komputer yang mudah dipahami oleh komputer, tetapi sulit dipahami oleh manusia. Hal ini mengakibatkan bahasa komputer yang mendukung pemrograman tingkat rendah tidak membutuhkan compiler yang baik dibandingkan dengan bahasa pemrograman tingkat tinggi.

Program yang dituliskan dengan bahasa pemrograman untuk membuat socket yang berguna untuk komunikasi antara satu komputer dengan komputer yang lainnya yang terhubung dalam jaringan komputer. Socket-socket ini akan sangat berguna dalam pengiriman dan penerimaan pesan yang menjadi tulang punggung pada aplikasi untuk pemrosesan paralel.

4. Metode Penelitian

4.1 Metode Studi *Literature*

Studi *literature* dilakukan untuk mendukung data teori yang dapat digunakan dalam proses perancangan perangkat lunak. Sumber utama dari metode ini adalah buku perpustakaan dan jurnal bidang keilmuan Teknik Informatika atau Ilmu Komputer. Seluruh sumber pustaka didapat dijumpai di perpustakaan STTA Yogyakarta dalam bentuk *hard book* ataupun *soft book*. Buku-buku yang digunakan untuk menyusun landasan teori merupakan buku yang kompetensi dalam keilmuan pemrosesan paralel. Sedangkan buku yang digunakan untuk merancang aplikasi menggunakan buku yang bersifat aplikatif dalam bidang ilmu komputer.

4.2 Metode Pengembangan Perangkat Lunak (*Water Fall*)

Pengembangan perangkat lunak yang digunakan untuk *remote* eksekusi dalam penelitian ini menggunakan metode *water fall*. Hasil identifikasi permasalahan dianalisa dan dibuat desain awalnya. Setelah selesai maka selanjutnya adalah proses implementasi yaitu pengujian, jika dalam pengujian masih terdapat permasalahan maka akan dilakukan proses ulang identifikasi dan seterusnya sehingga perangkat lunak hasil rancangan sudah dapat dibuktikan keandalannya.

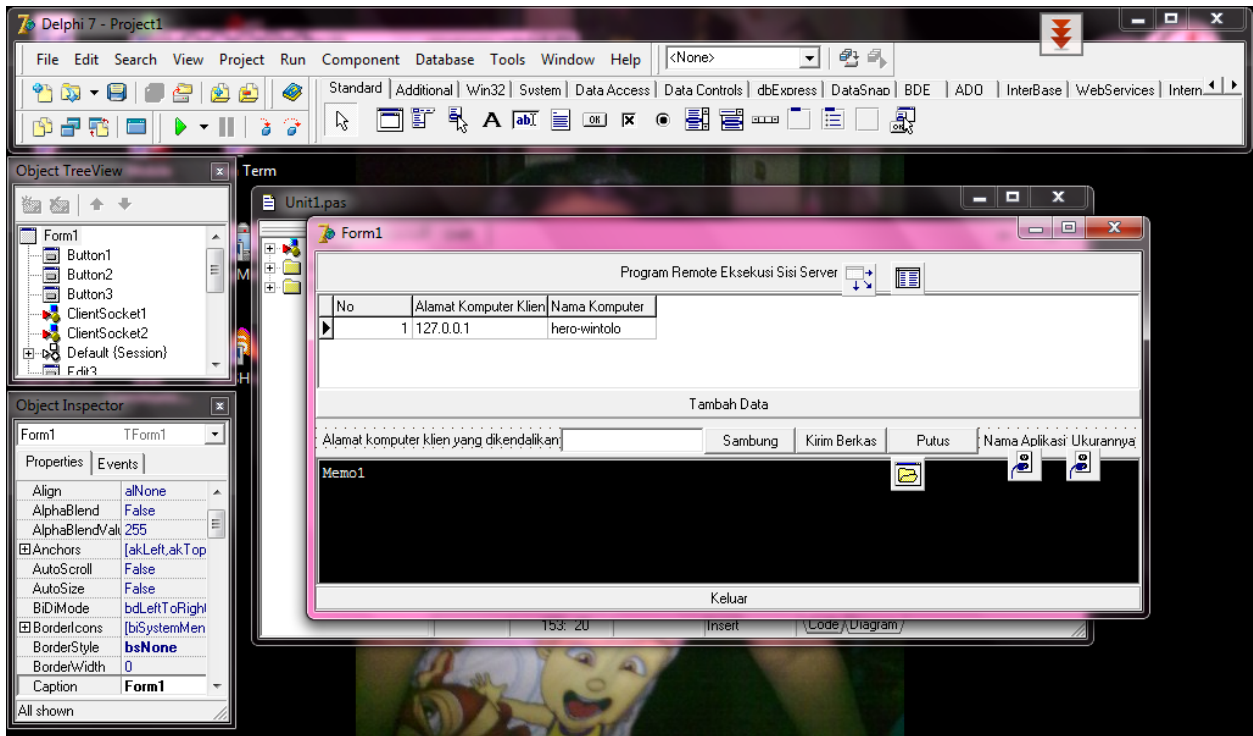
5. Hasil dan Pembahasan

Perancangan perangkat lunak dilakukan dengan menginisialisasi kebutuhan perangkat keras dan perangkat lunak serta sumber daya jaringan komputer yang tersedia, sehingga hasil dari perancangan dapat langsung diujicobakan, jika masih ada permasalahan akan segera diperbaiki.

Penelitian ini membutuhkan peralatan jaringan komputer yang akan digunakan dalam proses uji coba hasil perancangan perangkat yang menghubungkan antar komputer dalam sebuah jaringan komputer, peralatan tersebut adalah *Switch*.

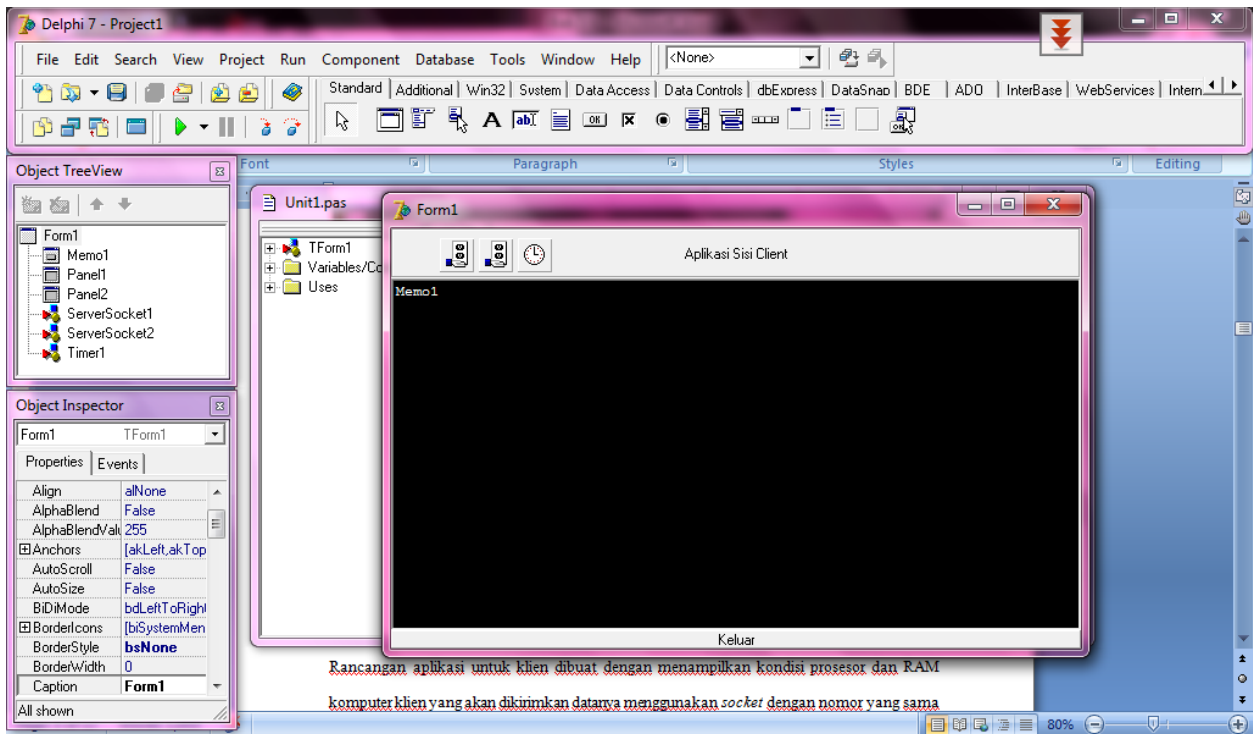
5.1 Perancangan Aplikasi

Aplikasi ini dirancang dalam dua bentuk, bentuk pertama untuk *server* dan bentuk kedua untuk klien. Tampilan aplikasi untuk *server* dapat dilihat pada gambar 2 dan tampilan aplikasi untuk klien dapat dilihat pada gambar 3.



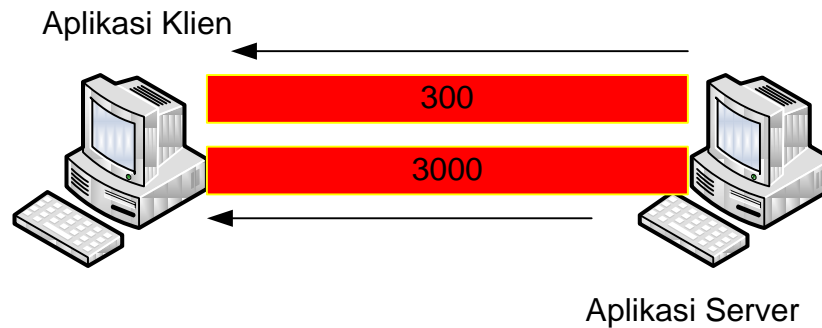
Gambar 2. Rancangan Aplikasi Server

Pada tampilan aplikasi untuk *server* , dibutuhkan komponen panel 3 buah, dbgrid, table, datasource, 3 tombol, edit text, 3 label dan memo line. Selain itu ada komponen yang paling dibutuhkan untuk membuat aplikasi ini, yaitu *client socket* sebanyak 2 buah.



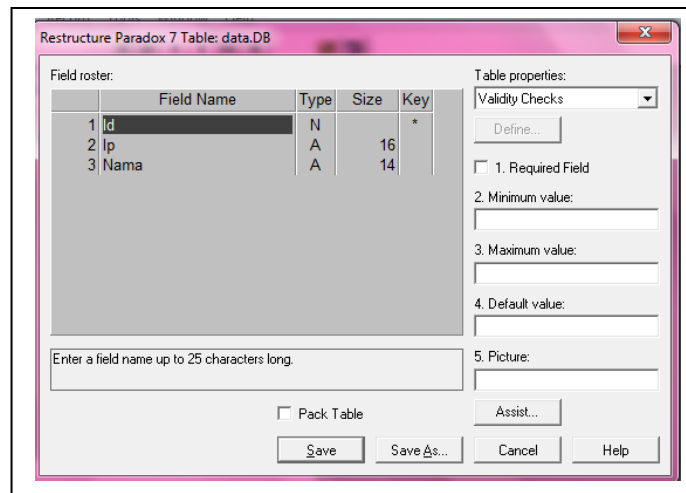
Gambar 3. Rancangan Aplikasi Klien

Rancangan aplikasi untuk klien dibuat dengan menggunakan komponen panel 2 buah, memo line, timer dan 2 buah *server socket*. Penggunaan komponen *server* pada aplikasi klien ini dimaksudkan untuk penggunaan aplikasi bagi kepentingan *remote* eksekusi, sehingga ketika *client socket* mengirimkan sesuatu dari aplikasi *server* dapat diterima oleh *socket* yang terpasang pada aplikasi klien.



Gambar 4. Ilustrasi hubungan antara aplikasi klien dan *server*

Pada aplikasi klien dipasang 2 buah *socket* dengan nomor yang sama dengan *socket* yang dipasang di *server*, agar *socket-socket* ini dapat terhubung dan mengirimkan data dari klien ke *server*. Pada gambar 4, *socket* nomor 300 pada klien terhubung dengan *socket* nomor 300 pada *server*, yang digunakan untuk mengirimkan pesan yang akan ditampilkan pada memo line sisi klien, sedangkan *socket* nomor 3000 pada klien terhubung dengan *socket* nomor 3000 pada *server*, digunakan untuk mengirimkan berkas yang berupa aplikasi atau program yang akan dieksekusi pada komputer klien.



Gambar 5. Struktur table data

5.2 Rancangan Tabel

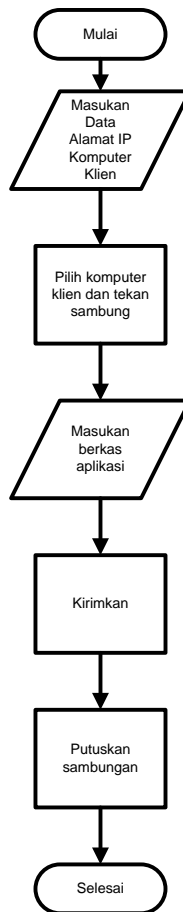
Rancangan aplikasi dalam penelitian ini membutuhkan sebuah table yang digunakan untuk menyimpan data alamat dan nama komputer klien yang digunakan untuk melakukan proses eksekusi berkas aplikasi yang dikirim dari komputer *server*, rancangan tabel tersebut disimpan dengan nama data dan dapat dilihat pada gambar 5.

The image shows a Windows application window titled "Form2". It contains a table with three rows and two columns. The first column contains labels: "No.", "Alamat Komputer", and "Nama Komputer". The second column contains text input fields labeled "Edit1", "Edit2", and "Edit3". Below the table is a blue button labeled "Simpan".

Gambar 6. Rancangan *interface* pengisian table data

Tabel dengan nama data tersebut yang berfungsi untuk menampung data komputer klien yang akan dikenai proses *remote* eksekusi diakses menggunakan aplikasi dengan tampilan seperti pada gambar 6. Aplikasi ini menjadi bagian dari aplikasi sisi *server* dan diharapkan dapat berfungsi seperti yang dimaksudkan diatas.

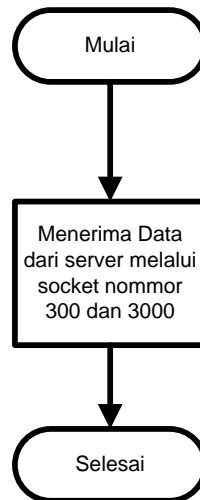
5.3 Flowchart System



Gambar 7. *Flowchart system Server*

Flowchart system atau sistem diagram alir yang digunakan dalam rancangan aplikasi ini ada 2 jenis, yang pertama untuk sisi server dan yang kedua untuk sisi kliennya. Baik *flowchart system server* atau klien berfungsi untuk menjelaskan jalannya aplikasi yang dirancang sehingga jika terjadi permasalahan dalam hal *syntax error* ataupun *symantic error* hal itu mudah untuk dilacak dan diperbaiki.

Aplikasi yang dirancang nantinya akan dijalankan dengan langkah pertama adalah memasukan nama komputer klien, kemudian melakukan proses pemanggilan komputer yang sudah tersimpan datanya dan setelah itu memilih berkas yang akan dikirim, setelah berkas dikirim maka selanjutnya koneksi antar socket harus diputuskan demi menjaga keamanan.



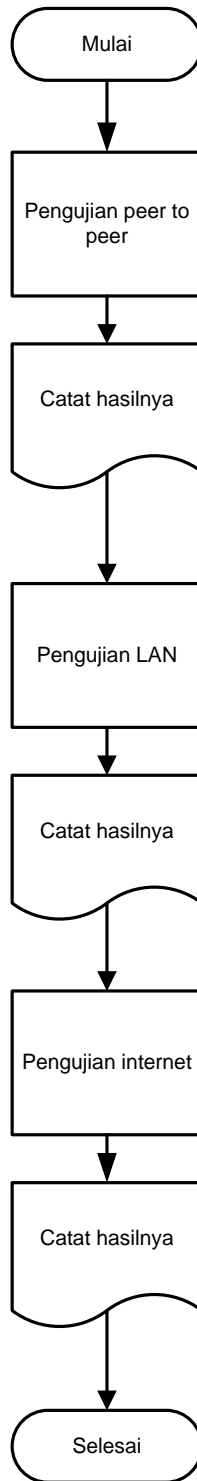
Gambar 8. *Flowchart system* Klien

Flowchart system klien lebih sederhana karena posisinya sebagai aplikasi yang pasif dan hanya menerima pesan dan berkas aplikasi dari *server* melalui socket dengan nomor *port* 300 dan 3000.

5.4 Algoritma dan Code Program

Pada aplikasi *server*, algoritmanya adalah mulai, kemudian memasukkan ip komputer klien untuk disambungkan ke aplikasi *server* tersebut. Setelah itu berkas dalam bentuk aplikasi dipilih dan kemudian dikirimkan ke komputer klien untuk dieksekusi atau dijalankan disana.

Sedangkan pada aplikasi komputer klien, algoritmanya adalah mulai menjalankan program dan menunggu hingga berkas dalam bentuk aplikasi dari komputer *server* dikirimkan, setelah sampai berkas dalam bentuk aplikasi itu langsung dieksekusi.

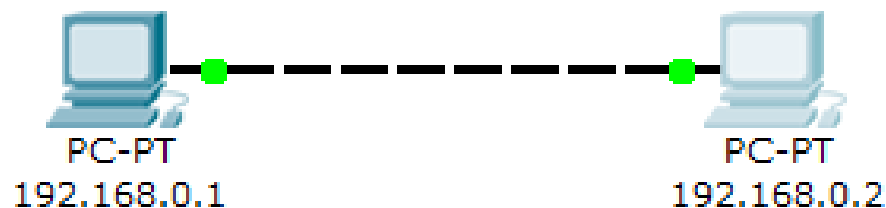


Gambar 10. *Flowchart* pengujian aplikasi

5.5 Cara Uji Coba

Pengujian aplikasi hasil rancangan ini dilakukan dalam sebuah jaringan komputer *peer to peer*, LAN, dan internet untuk dua buah aplikasi yang dipasang pada dua buah komputer yang bertindak sebagai server dan klien. Selain untuk menguji kinerja aplikasi hasil rancangan, pengujian dilakukan untuk mengetahui kecepatan eksekusi pada tiga jaringan komputer yang berbeda agar hasil uji coba ini dapat digunakan oleh orang lain yang melakukan penelitian yang memiliki tema yang sama.

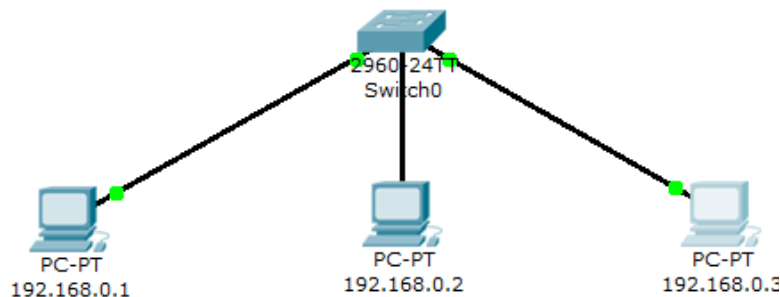
5.6 Uji Coba



Gambar 11. Skema pengujian *peer to peer*

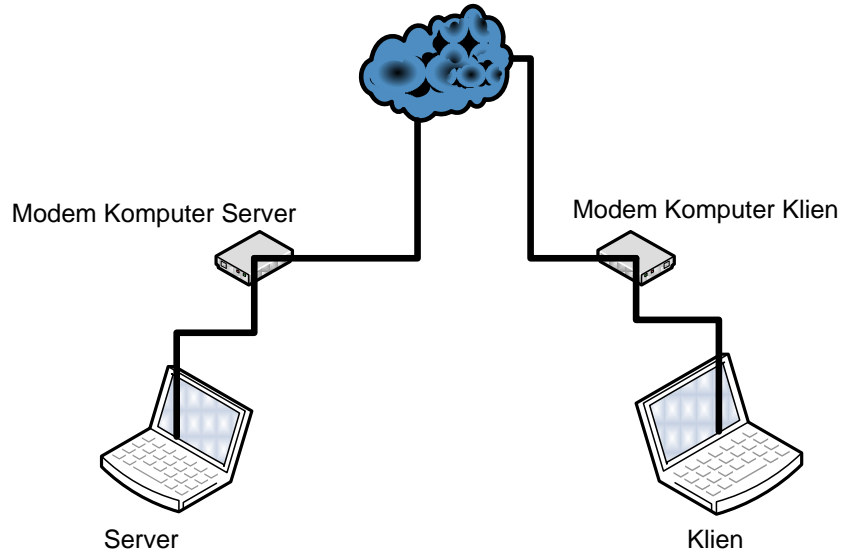
Pengujian yang pertama dilakukan pada dua buah komputer yang terhubung dengan media kabel secara *peer to peer* seperti tampak pada gambar 11. Pada skema pengujian ini dua buah komputer tersebut memiliki alamat jaringan yang sama yaitu 192.168.0.0/24 dengan media transmisi kabel *Unshielded Twisted Pair* (UTP) berjenis *cross over*. Pada komputer dengan alamat 192.168.0.1 dipasangkan aplikasi sisi server dan pada komputer dengan alamat 192.168.0.2 dipasangkan aplikasi sisi klien.

Pengujian berikutnya dilakukan dalam LAN, seperti tampak pada gambar 12, yang terdiri dari 3 komputer dengan alamat jaringan yang sama, 192.168.0.0/24, dengan aplikasi server dipasang pada komputer dengan alamat 192.168.0.1 dan aplikasi klien dipasang pada 2 komputer dengan alamat 192.168.0.2 dan 192.168.0.3. Skema pengujian ini menggunakan peralatan jaringan *switch* dan kabel UTP *straight trough*. Kecepatan pemindahan data pada jaringan ini maksimum 100Mb/detik, sehingga dari pengujian ini diharapkan akan mendapatkan data kecepatan eksekusi saat data melewati LAN.



Gambar 12. Skema pengujian dalam LAN

Pengujian yang ketiga atau yang terakhir adalah menguji aplikasi ini pada jaringan internet. Skemanya adalah 2 buah komputer yang telah terpasang aplikasi untuk sisi server dan sisi klien dihubungkan dengan modem, seperti tampak pada gambar 13. Setelah terhubung dengan modem dan provider penyedia jasa layanan internet aplikasi ini diujikan, hingga didapatkan kecepatan data saat terjadi proses *remote* eksekusi.



Gambar 13. Skema pengujian melalui jaringan internet

Ketiga skema pengujian ini dilakukan untuk menguji aplikasi *remote* eksekusi ketika mengirim berkas aplikasi yang dibuat menggunakan bahasa pemrograman c/c++ dan akan dieksekusi pada komputer yang bertindak sebagai klien. Ukuran berkas, kecepatan pemindahan data dan kecepatan eksekusi menjadi bagian dari data yang akan didapatkan saat pengujian untuk dianalisa.

Program Remote Eksekusi Sisi Server

No	Alamat Komputer Klien	Nama Komputer
▶ 1	127.0.0.1	hero-wintolo

Tambah Data

Alamat komputer klien yang dikendalikan

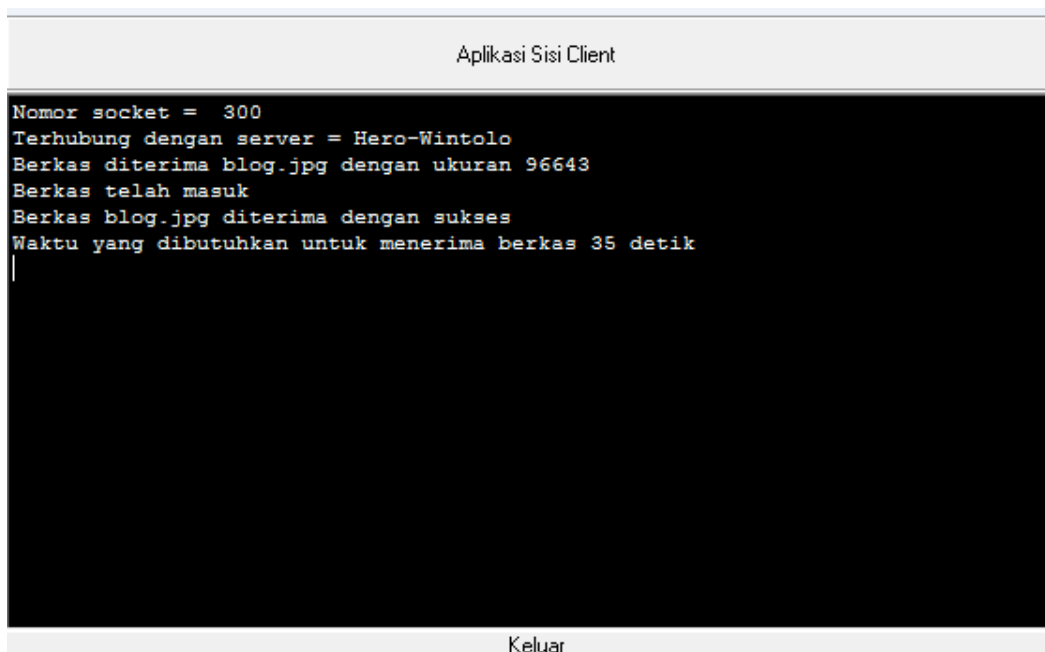
```
Tersambung dengan Komputer Klien : Hero-Wintolo
Berkas dikirim ke komputer 127.0.0.1.....
```

Keluar

Gambar 14. Aplikasi sisi server

Saat pengujian, pada aplikasi sisi server seperti tampak pada gambar 14, klik dua kali pada alamat komputer klien dan kemudian klik tombol **sambung**. Setelah tersambung maka pada program akan muncul pesan “tersambung dengan komputer klien :” . Kemudian klik tombol **kirim berkas** untuk memilih berkas yang akan dikirimkan dan dieksekusi pada komputer klien. Setelah hal ini dilakukan akan muncul pesan “berkas dikirim ke komputer

Jika ingin memutuskan koneksi komputer klien maka tekan tombol **putus**. Sedangkan pada aplikasi sisi klien tidak ada proses penekanan tombol untuk melakukan eksekusi, artinya aplikasi sisi client ini bersifat pasif, yaitu hanya menerima berkas dari server kemudian berkas tersebut dieksekusi secara otomatis saat tiba. Hal ini terjadi karena penggunaan dua buah socket yang memiliki nomor berbeda yaitu 300 dan 3000, satu socket untuk mengirimkan pesan sedangkan socket yang lainnya untuk mengirimkan berkas, seperti tampak pada gambar 15.



Gambar 15. Aplikasi sisi klien

5.7 Analisa Hasil Uji Coba

Pengujian terhadap aplikasi baik dari sisi klien atau server dilakukan pada 3 jenis jaringan yang berbeda untuk mengirimkan sebuah berkas yang memiliki ukuran yang sama pada nomor uji setiap jaringan yang berbeda, hal ini dimaksudkan untuk menguji kecepatan jaringan dalam memindahkan data yang digunakan dalam pengujian menjadi lebih obyektif. Kecepatan pemindahan data ini dibutuhkan datanya untuk dilakukan analisa, apakah ada pengaruh kecepatan jaringan dalam memindahkan data dari komputer server ke komputer klien terhadap ukuran berkas atau data yang dipindahkan. Sedangkan kecepatan eksekusi yang datanya juga diambil terhadap berkas atau data yang sama ketika dikirimkan dari komputer server ke komputer klien digunakan untuk mengukur respon prosesor dari komputer klien ketika diberi *job* secara remote hingga *job* tersebut dieksekusi.

Tabel 1 Hasil uji coba

No	Ukuran Berkas (byte)	Kecepatan Pemindahan Data	Kecepatan Eksekusi
<i>Jaringan peer to peer</i>			
1	110774	6:29 detik	2 detik
2	96182	6:33 detik	2 detik
3	77247	6:07 detik	2 detik
4	75892	6:01 detik	2 detik
5	105771	6:12 detik	2 detik
<i>Jaringan LAN</i>			
6	110774	5:12 detik	2 detik
7	96182	6:02 detik	2 detik
8	77247	5:21 detik	2 detik
9	75892	5:13 detik	2 detik
10	105771	5:34 detik	2 detik
<i>Jaringan internet</i>			
11	110774	8:11 detik	2 detik
12	96182	5:37 detik	2 detik
13	77247	6:42 detik	2 detik
14	75892	7:43 detik	2 detik
15	105771	8:05 detik	2 detik

Dari hasil pengujian yang terlihat pada tabel 1, kecepatan pemindahan data relative stabil baik pada saat jaringan *peer to peer*, LAN dan internet. Proses yang terjadi sebenarnya adalah proses *up load* dan *down load* antara komputer *server* dengan komputer klien yang menggunakan media transmisi yang berbeda, artinya cepat atau lambatnya berkas yang dikirim dari komputer *server* ke komputer klien sangat tergantung dengan kondisi kecepatan dan bandwidth dari jaringan komputer yang digunakan.

Ketika berkas yang dikirimkan oleh komputer *server* sampai pada komputer klien, maka berkas dalam bentuk *file* aplikasi tersebut secara otomatis langsung dieksekusi oleh prosesor komputer klien. Dari table 1, terlihat bahwa kecepatan eksekusi oleh prosesor komputer klien sangat stabil, hal ini dikondisikan saat menerima dan mengeksekusi berkas tersebut, komputer klien dalam keadaan *idle*.

Ketiga data kelompok pengujian dalam jaringan komputer yang berbeda ini hanya digunakan untuk membuktikan bahwa aplikasi yang dirancang, mampu mengirimkan berkas dan kemudian melakukan proses eksekusi terhadap berkas tersebut, sehingga remote eksekusi terjadi antara komputer yang menggunakan aplikasi *server* terhadap komputer yang menggunakan aplikasi klien. Baik *server* dan klien, semua komputer yang menggunakan aplikasi harus memiliki alamat IP, sehingga penggunaannya dapat lebih luas terhadap kebutuhan pemrosesan paralel. Meski demikian ada hal terpenting yang harus diperhatikan bahwa berkas dalam aplikasi yang hendak dikirimkan dan dieksekusi secara remote harus memiliki tujuan khusus seperti halnya pemrosesan paralel tersebut dibutuhkan.

6. Kesimpulan dan Saran

6.1 Kesimpulan

Dari hasil ujicoba yang dilakukan pada bab empat dari penelitian ini, maka dapat disimpulkan :

1. Proses *remote* eksekusi terhadap berkas aplikasi yang dikirimkan dari komputer *server* ke komputer klien telah berhasil dikirimkan dan kemudian dieksekusi secara otomatis menggunakan aplikasi yang dirancang.
2. Dukungan aplikasi yang dirancang terhadap pemrosesan *parallel* dapat terjadi jika berkas aplikasi yang dikirimkan dan dieksekusi tersebut merupakan bagian dari penyelesaian permasalahan yang dilakukan dengan pemrosesan *parallel*.

6.2 Saran

Penelitian ini dapat dilanjutkan untuk pengembangan pemrosesan *parallel* yang dilakukan secara *remote* terhadap beberapa komputer yang saling terhubung dalam jaringan komputer dengan mempertimbangkan aspek *load balancing* dengan metode *sorting*.

7. Daftar Pustaka

- Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar, 2003, Introduction to Parallel Computing, Second Edition, Addison Wesley
- Hero Wintolo, 2010, Deteksi Kinerja Prosesor Komputer Klien Dengan Cara Remote Untuk Mendukung Aplikasi Pemrosesan Paralel, Jurnal Ilmiah Angkasa, STTA
- Hesham El-Rewini, Mostafa Abd-El-Barr, 2005, Advanced Computer Architecture And Parallel Processing, A John Wiley & Sons, Inc Publication
- Ibaroudene, Djaffer, 2008, Parallel Processing, EG6370G: Chapter 1, Motivation and History, St Mary's University, San Antonio, TX. Spring
- Quinn, Michael J. , 2004, Parallel Programming in C with MPI and Open MP, Boston: McGraw Hill