

PENERAPAN *GRID COMPUTING* UNTUK MENINGKOMPILASI PROGRAM BERBAHASA C/C++

Yuliani Indrianingsih, Hero Wintolo, Ika Kartika Sari
Teknik Informatika STTA Yogyakarta
Informatika@stta.ac.id

ABSTRACT

Grid computing is a group of computers that connected to computer network and working together to complete a certain task. The computers inside must be coheren so the processors can work evenly when given a task from server. Grid computing can be used in many areas and purpopes, and in this research it will be used to compile C/C++ program. Compilation is a process to transform a human readable source code into a computer readable code. The compilation process will take place in a personal computer (PC). By using grid system in compilation process, the waiting time of a source code will be reduced as the compilation task will be divided among the computers. For that reason a C/C++ compiler software that using grid computing will be made. The task will be distributed among the computers inside the grid. It can also help someone who want to compile C/C++ source code but he doesn't have the compiler on his PC.

Keywords : *grid computing, compile, C/C++*

1. PENDAHULUAN

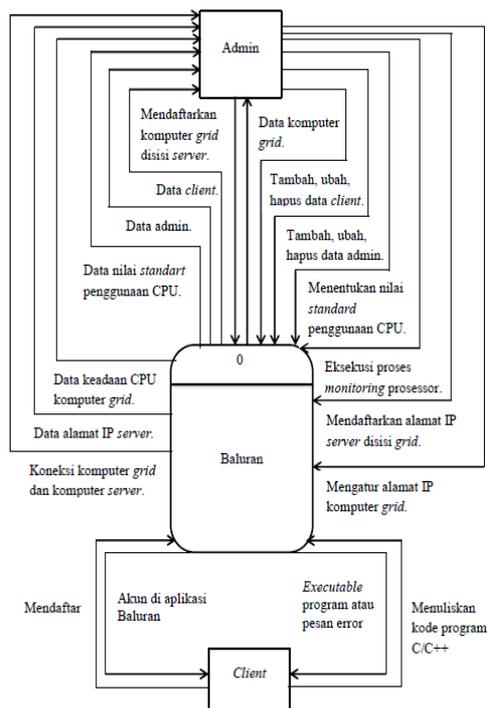
Grid computing merupakan suatu mekanisme pembagian tugas secara seimbang untuk membantu mempercepat waktu eksekusi suatu pekerjaan. *Grid computing* dapat diterapkan pada banyak bidang. Dalam

penelitian ini *grid computing* diterapkan dalam proses kompilasi kode program berbahasa C/C++. Proses kompilasi merupakan cara untuk mengubah bahasa pemrograman yang dipahami oleh manusia menjadi bahasa yang dipahami mesin. Proses kompilasi pada umumnya dilakukan oleh *compiler* yang sudah dipasang pada sebuah *Personal Computer* (PC). Dari sinilah muncul sebuah ide untuk merancang perangkat lunak berbasis *webyang* diberi nama Baluran, untuk membantu proses kompilasi kode program C/C++. Perangkat lunak yang dirancang akan diterapkan dalam *grid computing* agar beban mengkompilasi kode program C/C++ yang diterima dapat dibagi secara seimbang sehingga mempercepat waktu kompilasi. Perangkat lunak yang dirancang memanfaatkan *GNU's Not Unix (GNU) Compiler Collection* (GCC) sebagai *compiler*.

2. TINJAUAN PUSTAKA

Hero Wintolo, M.Kom (2010), dalam jurnal ini menerangkan bahwa kinerja processor komputer *client* dapat dipantau dari komputer *server* dengan bantuan perangkat lunak yang ter-*install* pada komputer *client*.

Hero Wintolo, M.Kom (2011), dalam jurnal ini menerangkan bahwa pengeksesian sebuah aplikasi pada komputer dapat dilakukan dengan meminjam processor komputer lain untuk mengeksesinya dengan syarat dua komputer tersebut terhubung.



Gambar 1 Diagram konteks Baluran

3. PERANCANGAN

Diagram Konteks Baluran

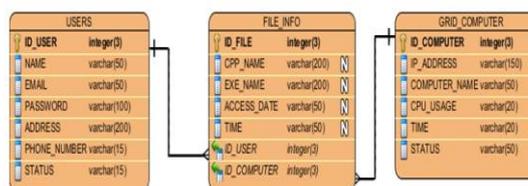
Secara keseluruhan Baluran mempunyai dua *entitas*, yaitu admin dan

client. Hubungan antara dua entitas tersebut terhadap Baluran seperti pada gambar 1. Berdasarkan diagram konteks di atas, tugas admin dalam Baluran adalah mendaftarkan komputer *grid*, memanipulasi data admin dan *client* jika diperlukan, menentukan nilai *idle standart* penggunaan CPU, mengeksesikan proses *montoring* CPU komputer *grid*, dan semua ini dilakukan pada komputer *server*. Sedangkan pada komputer *grid*, admin bertugas untuk mengatur alamat IP komputer *grid* dan mendaftarkan alamat IP komputer *server*. Beberapa informasi seperti data komputer *grid*, data *client*, data admin dan data keadaan CPU komputer *grid* akan didapatkan oleh admin dari pengoperasian Baluran pada sisi *server*. Data alamat IP *server*, koneksi antara komputer *server* dan komputer *grid* akan didapatkan oleh admin dari pengoperasian Baluran pada sisi *grid*.

Sedangkan *client* hanya perlu mendaftar dan menuliskan kode program C/C++ pada form yang disediakan pada Baluran. Dan Baluran akan memberikan *file* yang telah dikompilasi sebagai hasil dari penulisan kode program C/C++ tersebut, jika kode program yang dituliskan benar. Sedangkan jika penulisan kode program C/C++ tersebut salah maka Baluran akan memberikan pesan error.

Logical Record Structure(LRS)

LRS dari aplikasi Baluran seperti pada gambar 2. Database yang digunakan pada Baluran mempunyai tiga tabel, yaitu tabel USERS, tabel FILE_INFO dan tabel GRID COMPUTER. Field-field pada masing-masing tabel dapat dilihat pada ERD. Tiap tabel mempunyai relasi dengan tabel yang lainnya. Tabel USERS mempunyai relasi *one to many* dengan tabel FILE_INFO. Field ID_USER merupakan *foreign key* pada tabel FILE_INFO, yang merupakan *primary key* pada tabel USERS. Tabel GRID_COMPUTER mempunyai relasi *one to many* dengan tabel FILE_INFO. Dan Field ID_COMPUTER merupakan *foreign key* pada tabel FILE_INFO, yang merupakan *primary key* pada tabel GRID_COMPUTER.



Gambar 2. LRS dari aplikasi Baluran

Keterangan simbol :



many relationship

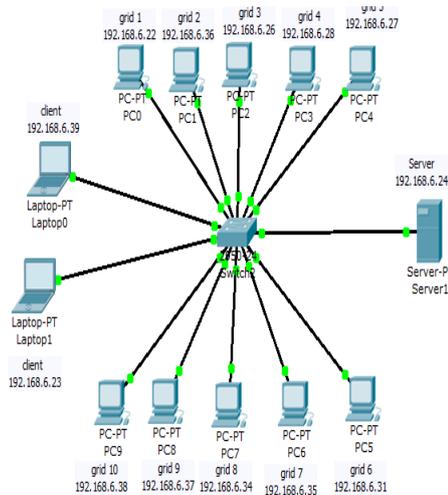
4. IMPLEMENTASI DAN ANALISA

Penjelasan Aplikasi

Aplikasi Baluran dirancang berbasis web, sehingga aplikasi ini bebas diakses oleh siapapun menggunakan platform apapun selama mempunyai kemampuan untuk membuka halaman web. Pada aplikasi Baluran terdapat dua user yaitu admin dan client. Bagian admin digunakan untuk *manage* dan manipulasi data komputer grid dan data user. Sedangkan bagian client digunakan untuk menuliskan kode program C/C++ yang akan dikompilasi.

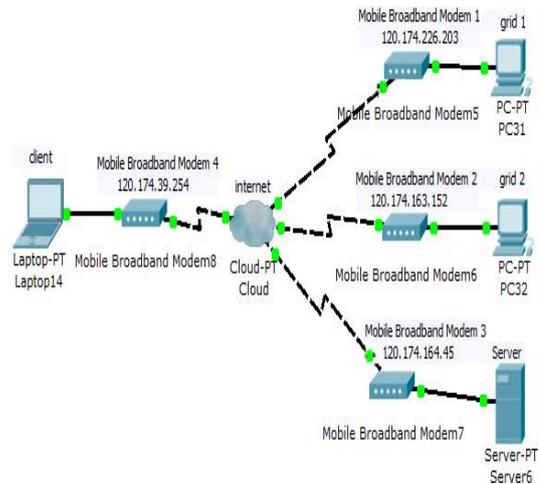
Uji Fungsi

Pengujian fungsi Baluran dilakukan pada Laboratorium Komputasi Sekolah Tinggi Teknologi Adisutjipto (STTA). Pengujian ini dilakukan 2 kali yaitu pada jaringan lokal dan pada jaringan internet. Pengujian pada jaringan lokal membutuhkan 13 komputer, dengan rincian 10 komputer grid, 1 komputer server dan 2 komputer client. Semua komputer-komputer tersebut terkoneksi dalam satu jaringan komputer. Skema jaringan pada pengujian jaringan lokal lengkap dengan alamat IP tiap computer seperti pada gambar 3.



Gambar 3 Skema jaringan pada pengujian jaringan lokal

Sedangkan pada pengujian fungsi Baluran pada jaringan internet dibutuhkan 4 komputer dan 4 modem. Dengan rincian 2 komputer dan 2 modem untuk komputer *grid*, 1 komputer dan 1 modem untuk komputer *server*, 1 komputer dan 1 modem untuk *client*. Modem yang dipakai dalam pengujian ini adalah modem ZTE MF 180 dengan kecepatan *up to* 3.6 Mbps, HUAWEI E303 dengan kecepatan *up to* 7.2 Mbps, HUAWEI E173 dengan kecepatan *up to* 7.2 Mbps, HUAWEI K3520 dengan kecepatan 3.6 Mbps. Keempat modem tersebut menggunakan *Subscriber Identity Module (SIM) card* IM3 dari *provider* Indosat dengan kecepatan *up to* 7.2 Mbps. Skema jaringan pengujian jaringan internet seperti pada gambar 4.



Gambar 4 Skema jaringan pada pengujian jaringan internet

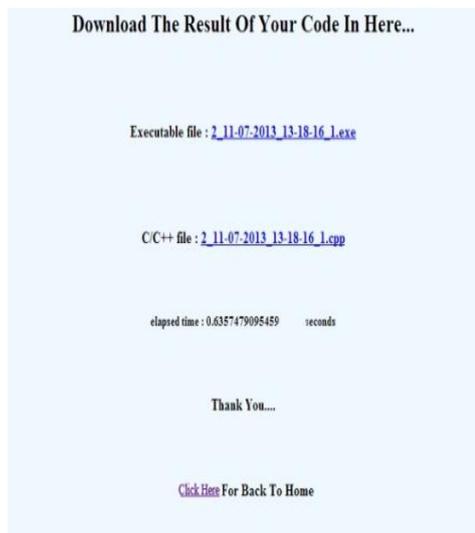
Pengujian Baluran akan dilakukan dengan mensimulasikan beberapa keadaan. Mengkompilasi kode program C/C++ untuk mendapatkan keadaan yang benar. Dan mengkompilasi kode program java untuk mendapatkan keadaan yang salah.

Analisa Uji Fungsi Pada Jaringan Lokal

Pelaksanaan pengujian pada jaringan lokal yang pertama dilakukan dengan mengkompilasi kode program C. Salah satu kode program C yang akan dikompilasi adalah sebagai berikut :

1. # include <stdio.h>
2. main(){ int a, b, c, max, min ;
3. printf("Masukan bil 1 :");scanf("%d",&a);
4. printf("Masukan bil 2 :");scanf("%d",&b);
5. printf("Masukan bil 3 :");scanf("%d",&c);
6. if (a>b) { max=a; min=b;}
7. else{ max=b; min=a;}
8. if (c>max) max=c;

9. `if (c<min) min=c;`
10. `printf("Maka bilangan terbesar adalah %d\n",max);`
11. `printf("Maka bilangan terkecil adalah %d",min);}`



Gambar 5 Hasil kompilasi kode program C

Pada hasil kompilasi, nama *file executable* dan *file* kode program dibuatkan oleh aplikasi baluran yang merupakan kombinasi dari beberapa data file tersebut. Sebagai contoh nama *file* seperti pada gambar 5 yaitu `2_11-07-2013_13-18-16_1.exe` dan `2_11-07-2013_13-18-16_1.cpp`. Nama tersebut merupakan kombinasi dari *id user client* yaitu 2, tanggal *file* dikompilasi yaitu 11-07-2013, waktu *file* dikompilasi yaitu pukul 13:18:16, dan *id komputer grid* yang mengkompilasi *file* tersebut yaitu 1. Hal ini dilakukan agar *file client* tidak bertukar dengan *file client* yang lain. Selain itu waktu tunggu proses kompilasi juga ditampilkan, seperti pada

gambar 5. Waktu tunggu ini dihitung mulai saat *client* meng-*click* tombol *Compile* pada halaman *home* sampai *link* untuk mengunduh *file executable* dan kode program muncul. Pada *file C* ini misalnya waktu tunggu proses kompilasi adalah 0.6357479095459 detik. Ukuran *file executable* yang dihasilkan dari kompilasi kode program C ini adalah 21 KB dan ukuran *file* yang berisi kode program yang dihasilkan adalah 1 KB.

Pelaksanaan pengujian pada jaringan lokal yang terakhir dilakukan dengan mengkompilasi kode program java. Salah satu kode program java yang akan dikompilasi sebagai berikut :

1. `package javaapplication8;`
2. `import javax.swing.JOptionPane;`
3. `public class JavaApplication8{`
4. `public static void`
5. `main(String[]args){`
6. `int a,b,bayar; double`
7. `harga,diskon,f; String`
8. `bil1,bil2,bil3,bil4;`
9. `bil3=JOptionPane.showInputDialog("Berapa besaran diskon [10%=0.1]");`
10. `diskon=Double.parseDouble(bil3);`
11. `bil1=JOptionPane.showInputDialog("Berapa jumlah RAM yang anda beli");`
12. `a=Integer.parseInt(bil1);`
13. `bil2=JOptionPane.showInputDialog("Berapa kapasitas RAM yang anda beli [1Gb atau 2Gb]");`
14. `b=Integer.parseInt(bil2);`
15. `if(a>=2){`
16. `if(b==1){`
17. `harga=(a*150000)-`
18. `(a*(150000*diskon));`

```

15. JOptionPane.showMessageDialog
    (null, "Harga yang harus dibayar
    "+harga);
16. bil4=JOptionPane.showInputDialog
    ("Jumlah uang anda ");
17. bayar=Integer.parseInt(bil4);
18. f=bayar-harga;
19. JOptionPane.showMessageDialog
    (null,"Uang kembaliannya " +f);}
20. else{
21. harga=a*250000;
22. JOptionPane.showMessageDialog
    (null, "Maaf anda tidak
    mendapatkan diskon dan harus
    membayar sebesar "+harga);
23. bil4=JOptionPane.showInputDialog
    ("Jumlah uang anda ");
24. bayar=Integer.parseInt(bil4);
25. f=bayar-harga;
26. JOptionPane.showMessageDialog
    (null, "Kembaliannya "+f);} }
27. else{
28. if(b==1){ harga=a*150000;
29. JOptionPane.showMessageDialog
    (null, "Maaf anda tidak
    mendapatkan diskon dan harus
    membayar sebesar "+harga);
30. bil4=JOptionPane.showInputDialog
    ("Jumlah uang anda ");
31. bayar=Integer.parseInt(bil4);
32. f=bayar-harga;
33. JOptionPane.showMessageDialog
    (null, "Kembaliannya "+f); }
34. else{
35. harga=a*250000;
36. JOptionPane.showMessageDialog
    (null, "Maaf anda tidak
    mendapatkan diskon dan harus
    membayar sebesar "+harga);
37.
    bil4=JOptionPane.showInputDialog
    ("Jumlah uang anda ");
38. bayar=Integer.parseInt(bil4);f=bay
    ar-harga;
39. JOptionPane.showMessageDialog
    (null, "Kembaliannya "+f); } } }

```



Gambar 6 Hasil kompilasi kode program java

Nama *file* kode program hasil kompilasi kode program java ini adalah 3_11-07-2013_13-44-32_1.cpp. Nama tersebut merupakan kombinasi dari *id user client* yaitu 3, tanggal *file* dikompilasi yaitu 11-07-2013, waktu *file* dikompilasi yaitu pukul 13:44:32, dan *id komputer grid* yang mengkompilasi *file* tersebut yaitu 1. Waktu tunggu proses kompilasi kode program java ini adalah 0.58773493766785 detik. Ukuran *file* yang berisi kode program yang dihasilkan adalah 3 KB. Dari hasil pengujian fungsi pada jaringan lokal maka dapat dilihat bahwa aplikasi Baluran dapat mengkompilasi program C/C++ dan dapat memilihkan komputer grid yang tepat untuk mengkompilasi kode program.

Analisa Uji Fungsi Pada Jaringan Internet

Pengujian pada jaringan internet dilakukan dengan mengkompilasi kode program C/C++ dan java yang sama dengan kode program yang dikompilasi saat pengujian pada jaringan lokal. Pengujian yang pertama dengan mengkompilasi kode program C dan hasil kompilasi seperti pada gambar 7.



Gambar 7 Hasil kompilasi kode program C pada pengujian internet

Nama *file executable* dan *file* kode program hasil kompilasi kode program C ini adalah 2_20-07-2013_17-35-47_1.exe dan 2_20-07-2013_17-35-47_1.cpp. Nama tersebut merupakan kombinasi dari *id user client* yaitu 2, tanggal *file* dikompilasi yaitu 20-07-2013, waktu *file* dikompilasi yaitu pukul 17:35:47, dan *id* komputer *grid* yang

mengkompilasi *file* tersebut yaitu 2. Waktu tunggu proses kompilasi kode program C ini adalah 10.053386926651 detik. Ukuran *file executable* yang dihasilkan dari kompilasi kode program C ini adalah 21 KB dan ukuran *file* yang berisi kode program yang dihasilkan adalah 1 KB.



Gambar 8 Hasil kompilasi kode program java pada pengujian internet

Kemudian dilanjutkan dengan mengkompilasi kode program java. Hasil kompilasi kode program seperti pada gambar 8. Nama *file executable* dan *file* kode program hasil kompilasi kode program C++ ini adalah 2_20-07-2013_18-07-25_1.exe dan 2_20-07-2013_18-07-25_1.cpp. Nama tersebut merupakan kombinasi dari *id user client* yaitu 2, tanggal *file* dikompilasi yaitu 20-07-2013, waktu *file* dikompilasi yaitu pukul 18:07:25, dan *id* komputer *grid* yang mengkompilasi *file* tersebut yaitu 1. Waktu tunggu proses kompilasi kode

program java ini adalah 9.2351660728455 detik. Ukuran *file* yang berisi kode program yang dihasilkan adalah 3 KB.

Dari hasil pengujian fungsi pada jaringan internet maka dapat dilihat bahwa waktu tunggu proses kompilasi kode program pada jaringan internet lebih lama daripada waktu tunggu proses kompilasi kode program pada jaringan lokal. Dari hasil pengujian fungsi pada jaringan internet dan pada jaringan lokal dapat dilihat pula bahwa waktu tunggu proses kompilasi dengan kode program yang benar lebih lama daripada proses kompilasi dengan kode program yang salah. Hal ini terjadi karena saat proses kompilasi dengan kode program salah, *compiler* tidak perlu membuat *file executable* lagi dan hanya perlu menampilkan pesan *error*.

5. KESIMPULAN DAN SARAN

Kesimpulan

Setelah melalui tahapan pengujian aplikasi, maka dapat disimpulkan :

1. Perangkat lunak hasil rancangan yang diberi nama Baluran, dapat diterapkan pada *grid computing*.
2. Kompilasi dapat dilakukan dengan memanfaatkan jaringan komputer lokal atau yang biasa disebut dengan LAN dan jaringan internet.

3. *Compiler* GCC dapat digunakan untuk mengkompilasi kode program C/C++ yang dikirimkan dari jarak jauh.
4. Dari hasil pengisian kuesioner oleh para responden, maka dapat disimpulkan bahwa Baluran merupakan aplikasi yang layak digunakan.

Saran

Setelah melalui tahapan pengujian aplikasi, maka saran yang dapat disampaikan untuk membantu pengembangan aplikasi ini sebagai berikut:

1. Jika Baluran akan diterapkan pada jaringan internet, disarankan memakai alamat IP public yang statis.
2. Aplikasi Baluran dapat dikembangkan lagi dengan penambahan mekanisme *load balancing* didalamnya.

DAFTAR PUSTAKA

Clement Salome, Pottle Brian, Singh, *Oracle Database : SQL Fundamentals I*, Oracle, Edition 1.0, 2010.

Hari Soetanto, *Teknik Kompilasi*, Fakultas Teknologi Informasi Universitas Budi Luhur, 2004.

Hero Wintolo, *Deteksi Kinerja Prosesor komputer client dengan cara remote untuk mendukung aplikasi pemrosesan parallel*, Sekolah

- Tinggi Teknologi Adisutjipto Yogyakarta, Volume 2, 2010.
- Hero Wintolo, *Design Of Parallel Processing applications With The Remote Execution*, Sekolah Tinggi Teknologi Adisutjipto Yogyakarta, Volume III, 2011.
- Jogiyanto, *Analisis & Desain, Sistem Informasi : Pendekatan Terstruktur Teori Dan Praktik Aplikasi Bisnis*, ANDI OFFSET, Edisi III, 2005.
- Raharjo Budi, *Belajar Otodidak Pemrograman Web Dengan PHP + Oracle*, Informatika, 2011.
- Wilikison Barry & Allen Michael, *Parallel Programming*, ANDI OFFSET, 2010.
- <http://www.databasedesign.co.uk/bookdatabasesafirstcourse/chap3/chap3.htm>, diakses pada tanggal 29 Juli 2013.
- http://data.bangtech.com/datamodel/visio_chen_erd.htm, diakses pada tanggal 29 Juli 2013.
- <http://albahrimaraxsa.wordpress.com/2012/05/01/erd-lrs-dan-relasi/>, diakses pada tanggal 29 Juli 2013.
- <http://www.w3schools.com/>, diakses pada tanggal 29 Juli 2013.
- <http://getbootstrap.com/>, diakses pada tanggal 29 Juli 2013.
- <http://php.net/manual/en/>, diakses pada tanggal 29 Juli 2013.
- <http://computer.howstuffworks.com/>, diakses pada tanggal 29 Juli 2013.
- <http://www.techonthenet.com/oracle/>, diakses pada tanggal 29 Juli 2013.