

BAB IV PERANCANGAN DAN IMPLEMENTASI

4.1 Konfigurasi Perangkat Keras

Sistem komputasi paralel menggunakan 17 komputer: 1 *front-end*, 6 *node* komputasi primer dan 10 *node* komputasi sekunder. Jaringan area lokal sistem komputasi paralel menggunakan *switch* FastEthernet dengan perkabelan Cat 5e.

Masing-masing satu komputer dari *node* komputasi primer dan sekunder dilengkapi dengan energimeter Wanf D02A untuk pengukuran daya. Gambar 4.1 memperlihatkan bentuk fisik dari energimeter Wanf D02A.



Gambar 4.1 Energimeter Wanf D02A

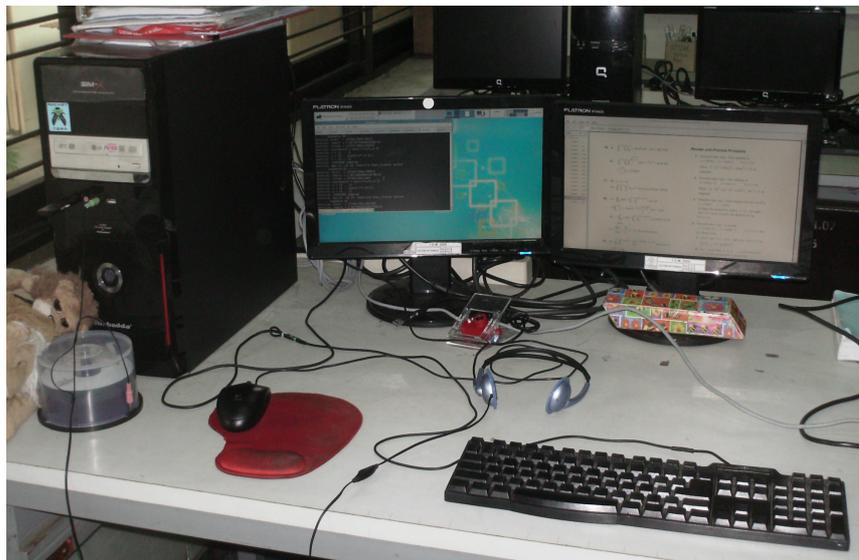
4.1.1 Konfigurasi Perangkat Keras *Front-end*

Komputer *front-end* adalah komputer tunggal yang digunakan sebagai antarmuka pengguna dan sistem komputasi paralel. Spesifikasi perangkat keras komputer *front-end* diuraikan dalam Tabel 4.1.

Tabel 4.1 Perangkat Keras Komputer *Front-end*

Prosesor	Intel Pentium E5400 (2M Cache, 2.70 GHz, 800 MHz FSB)
Board	BIOSTAR G31-M7 TE
Memori	2 x 1GB DDR2 6400
Storage	Seagate ST3500414CS Seagate ST3250318AS
NIC	Realtek Semiconductor Co., Ltd. RTL8101E/RTL8102E PCI Express Fast Ethernet controller (rev 02)

Gambar 4.1.1 memperlihatkan unit komputer *front-end* yang digunakan sebagai antarmuka sistem komputasi paralel: kompilasi dan distribusi program dan penampilan hasil komputasi.

**Gambar 4.1.1** Unit Komputer *Front-end*

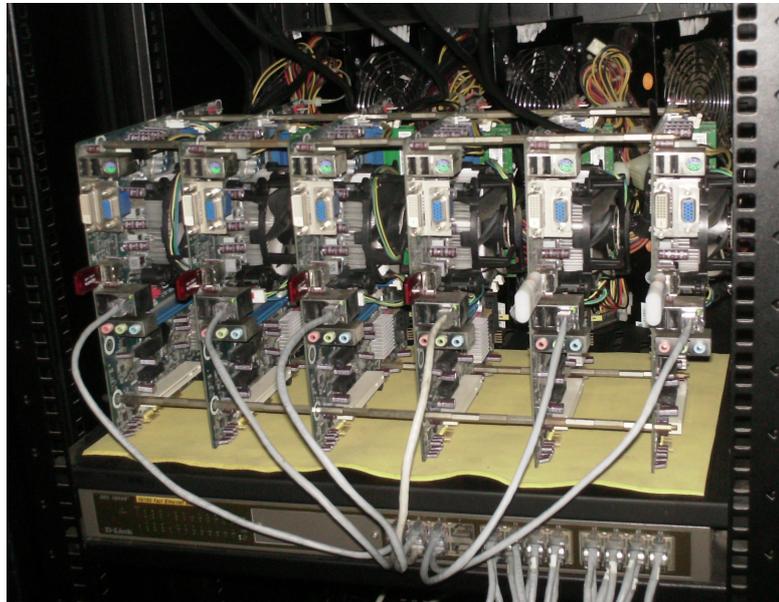
4.1.2 Konfigurasi Perangkat Keras Node Komputasi Primer

Komputer yang digunakan sebagai *node* komputasi primer berjumlah 6 unit dan digunakan untuk semua tahap pengujian, 5 unit menggunakan prosesor Intel Core i3-550 dan 1 unit menggunakan prosesor Intel Core i3-540. Media *boot* yang digunakan adalah flashdrive dengan ukuran nominal 4 GB. Spesifikasi perangkat keras komputer *node* komputasi primer diuraikan dalam Tabel 4.2.

Tabel 4.2 Perangkat Keras Komputer *Node* Komputasi Primer

Prosesor	Intel Core i3 550 (4M Cache, 3.20 GHz) Intel Core i3 540 (4M Cache, 3.06 GHz)
Board	Intel Desktop Board DH55PJ
Memori	2GB DDR3 10600
Storage	ADATA DashDrive UV100 4GB Toshiba TransMemory 4GB
NIC	Intel Corporation 82578DC Gigabit Network Connection (rev 06)

Gambar 4.1.2 memperlihatkan 6 unit *node* komputasi primer yang telah terpasang beserta *switch* DES 1024R.

**Gambar 4.1.2** Enam Unit *Node* Komputasi Primer

4.1.3 Konfigurasi Perangkat Keras *Node* Komputasi Sekunder

Komputer yang digunakan sebagai *node* komputasi sekunder adalah 10 unit dan digunakan hanya dalam pengujian pengaruh cacah *node*. Spesifikasi *node* komputasi sekunder adalah paket komputer pribadi *built-up* Compaq CQ3138L yang dijelaskan dalam Tabel 4.3.

Tabel 4.3 Perangkat Keras Komputer *Node* Komputasi Sekunder

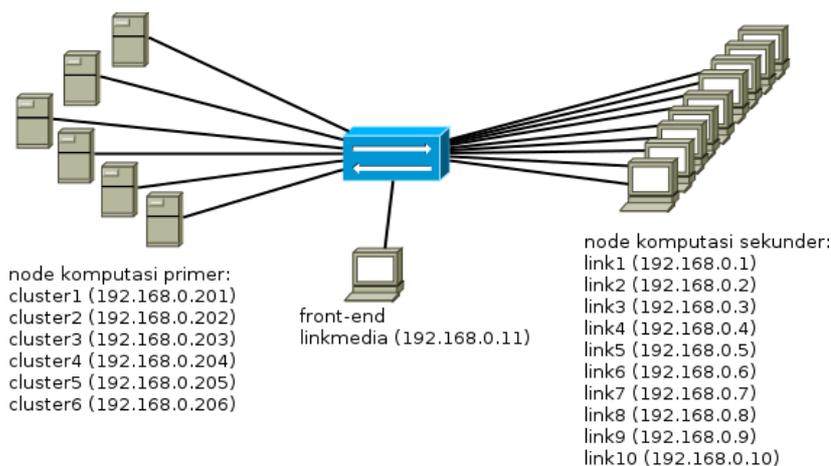
Prosesor	Intel E7500 (3M Cache, 2.93 GHz, 1066 MHz FSB)
Board	HP H-I41-uATX
Memori	2 x 1GB DDR3 10600
Storage	Seagate ST3320418AS
NIC	Realtek Semiconductor Co., Ltd. RTL8101E/RTL8102E PCI Express Fast Ethernet controller (rev 02)

Gambar 4.1.3 memperlihatkan unit *node* komputasi sekunder yang digunakan pada percobaan pengaruh cacah *node*.

**Gambar 4.1.3** Unit *Node* Komputasi Sekunder

4.1.4 Konfigurasi Jaringan Lokal

Jaringan sistem komputasi paralel menggunakan jaringan lokal dengan topologi bintang. NIC dari tiap komputer terhubung langsung dengan *switch* DES 1024R yang mengakomodasi kecepatan Fast Ethernet (10/100 Mbps) melalui kabel UTP dengan kategori 5e. Konfigurasi pengalamatan jaringan lokal menggunakan IPv4 dengan lingkup jaringan 192.168.0.0/24. Gambar 4.2 memperlihatkan topologi jaringan, asosiasi nama *host* dan alamat IP.



Gambar 4.2 Jaringan Lokal Sistem Komputasi Paralel

4.2 Konfigurasi Perangkat Lunak

Konfigurasi perangkat lunak sistem komputasi paralel meliputi konfigurasi jaringan, *environment* sistem operasi, *filesystem* terdistribusi, kompilasi dan instalasi paket program dependensi, implementasi program komputasi paralel, dan otomatisasi pengujian.

4.2.1 Konfigurasi *Frontend* (E5400)

Komputer *front-end* bertanggung jawab atas distribusi berkas *executable* program melalui *filesystem* terdistribusi (NFS), kompilasi (`mpicc`) dan inisialisasi eksekusi (`mpirun`) program komputasi paralel, operasi masukan parameter program komputasi paralel melalui argumen `stdin`, dan menampilkan keluaran program melalui `stdout`.

4.2.1.1 Konfigurasi Alamat IP dan *Hosts Front-end*

Front-end menggunakan sistem operasi Salix64 14.0 dengan kernel Linux 3.2.29 64-bit. Paket-paket dependensi: `gcc`, `nfsutils`, `openssh` merupakan telah terpasang sejak instalasi *default*.

Front-end menggunakan nama *host* `linkmedia` dengan alamat IP 192.168.0.11. Sistem operasi Salix64 menggunakan berkas `/etc/hosts` untuk mengasosiasikan nama *host* dan alamat IP. Semua komputer dalam sistem komputasi paralel menggunakan berkas `/etc/hosts` yang berisi semua entri semua *hosts* dalam sistem komputasi paralel agar dapat mengenali *hosts* lainnya. Berkas `/etc/hosts` selengkapnya dapat dilihat di Lampiran 1.

Pengaturan alamat IP *front-end* dilakukan dengan menambahkan entri:

```
IPADDR[0]="192.168.0.11"
NETMASK[0]="255.255.255.0"
```

ke berkas `/etc/rc.d/rc.inet1.conf`.

Pengaturan nama *host* dilakukan dengan penambahan berkas `/etc/HOSTNAME` dengan entri:

```
linkmedia.link.elektro.ub.ac.id
```

4.2.1.2 Konfigurasi User mpi

Sebuah *user* identik diperlukan untuk menjalankan Open MPI di beberapa *user* sekaligus, hal ini mengharuskan adanya instans *user* dengan UID (*user id*) dan GID (*group id*) yang identik di semua *hosts* sistem komputasi paralel. Untuk menambahkan *user* dengan nama `mpi` dan sandi lewat `mpi`, berkas `/etc/passwd` ditambahkan entri:

```
mpi:$5$CUgEJj9IK3$/txgiLkORWmB2V0VQ
/Sh40sxEMIR0qnYByJcX0tJoK1:2000:2000:~/home/mpi:/bin/bash
```

dan berkas `/etc/group` ditambahkan entri:

```
mpi:x:2000:
```

User `mpi` perlu memiliki direktori *home* agar berbagai berkas dapat disimpan.

Konfigurasi yang dilakukan:

```
# mkdir /home/mpi
# chmod 777 /home/mpi
# chown mpi:mpi /home/mpi
```

4.2.1.3 Konfigurasi Server NFS

Distribusi kunci publik SSH dan program komputasi paralel menggunakan NFS (Network File System) sebagai media. Direktori yang di-*export* melalui NFS akan dapat di-*mount* di *node*.

Penambahan direktori *home user* `mpi` (`/home/mpi`) dengan NFS di *front-end* yang bertindak sebagai server dilakukan dengan pengubahan berkas `/etc/exports` dengan entri:

```
/home/mpi 192.168.0.0/255.255.255.0(rw)
```

Pengaktifan *service* NFS dilakukan dengan cara memberikan moderasi *executable* terhadap berkas `/etc/rc.d/rc.rpc` dan `/etc/rc.d/rc.nfsd`:

```
# chmod +x /etc/rc.d/rc.rpc
# chmod +x /etc/rc.d/rc.nfsd
```

4.2.1.2 Konfigurasi SSH

Eksekusi Open MPI (`mpirun`) membutuhkan login *remote* dengan SSH di *node* tanpa *prompt* sandi lewat. Hal ini dilakukan dengan penambahan entri di berkas `/home/mipi/.ssh/authorized_keys` dengan kunci publik *user mpi* di *front-end*.

```
$ cp ~/.ssh/id_ecdsa.pub ~/.ssh/authorized_keys
```

4.2.1.4 Kompilasi dan Instalasi OpenMPI *Front-end*

Program Open MPI harus dikompilasi dan dipasang pada *path* yang sesuai di semua komputer sistem komputasi paralel. Langkah konfigurasi dan instalasi adalah sebagai berikut:

```
# wget http://www.open-mpi.org/software/ompi/v1.7/downloads/openmpi-1.7.tar.bz2
# tar -jxvf openmpi-1.7.tar.bz2 -C /tmp
# cd /tmp/openmpi-1.7
# ./configure && make && make install
# ldconfig
```

Proses instalasi telah berhasil jika `mpirun` dan `mpicc` dikenali dalam *path* sistem dan pustaka yang berkaitan berhasil dimuat.

Hasil kompilasi kompilasi dapat diarsip untuk instalasi di *node* komputasi primer karena memiliki versi kernel, gcc, dan glibc yang sama.

```
# tar -jcvf /tmp/openmpi-1.7-salix.tar.bz2 /tmp/openmpi-1.7
```

4.2.2 Konfigurasi *Node* Komputasi Primer (6 Unit i3 550/540)

Node komputasi primer adalah komputer yang digunakan di semua tahap pengujian sistem komputasi paralel, menggunakan alamat IP 192.168.0.201-192.168.0.206, mendapatkan berkas *executable* dan kunci publik SSH melalui direktori NFS. Sistem operasi yang digunakan *node* komputasi primer adalah Salix64 14.0 dengan gcc 4.7.1, glibc 2.15 dan kernel Linux 3.2.29. Paket `nfsutils` dan `openssh` telah terpasang sejak instalasi *default*.

Node komputasi primer dikonfigurasi *headless* (tanpa layar dan perangkat I/O) sehingga penyalaan dilakukan dengan paket *wake on lan*.

4.2.2.1 Konfigurasi Alamat IP dan *Host Node* Komputasi Primer

Pengaturan alamat IP *node* dan nama *host* sama seperti pengaturan alamat IP dan nama *host front-end*. Tabel 4.4 menjelaskan asosiasi nama *host* dan alamat IP untuk keenam *node* komputasi primer.

Tabel 4.4 Asosiasi Nama *Host* dan Alamat IP *Node* Komputasi Primer

Nama <i>Host</i>	Alamat IP
cluster1	192.168.0.201
cluster2	192.168.0.202
cluster3	192.168.0.203
cluster4	192.168.0.204
cluster5	192.168.0.205
cluster6	192.168.0.206

4.2.2.2 Konfigurasi User *mpi*

Konfigurasi *user mpi* di *node* komputasi primer sama dengan konfigurasi *user mpi* di *front-end*.

4.2.2.3 Konfigurasi Klien NFS

Node komputasi primer melakukan mounting direktori NFS dari *front-end* secara otomatis tiap *boot* dengan menambahkan entri ke berkas */etc/fstab*:

```
192.168.0.11:/home/mpi /home/mpi nfs sloppy,rw 0 0
```

4.2.2.4 Konfigurasi SSH

Node komputasi primer mendapatkan kunci publik SSH secara otomatis setelah direktori *home user mpi* di-*mount* dan konfigurasi SSH di *front-end* telah dilakukan.

4.2.2.5 Instalasi OpenMPI *Node* Komputasi Primer

Node komputasi primer dapat melakukan instalasi Open MPI 1.7 dari arsip hasil kompilasi *front-end* karena memiliki versi kernel, gcc, dan glibc yang sama.

```
# tar -jxvf openmpi-1.7.tar.bz2 -C /tmp
# cd /tmp/openmpi-1.7
# make install
# ldconfig
```

Pogram mpirun dan mpicc akan dikenali dalam *path* sistem dan pustaka yang berkaitan akan dimuat jika program dijalankan.

4.2.2.6 Konfigurasi *Wake on Lan*

Node komputasi primer dikonfigurasi sehingga *headless* sehingga tidak terdapat tombol untuk menyalakan unit komputer, sebagai gantinya *node* dinyalakan dengan paket *wake on lan* dengan alamat MAC (*media access control*) yang bersesuaian. Tabel 4.5 menjelaskan asosiasi nama *host* dan alamat MAC untuk keenam *node* komputasi primer.

Tabel 4.5 Asosiasi Nama *Host* dan Alamat MAC *Node* Komputasi Primer

Nama Host	Alamat MAC
cluster1	70:71:bc:ac:e3:b6
cluster2	70:71:bc:a3:9e:85
cluster3	70:71:bc:a3:9e:82
cluster4	70:71:bc:a3:9e:6f
cluster5	70:71:bc:a3:9e:7e
cluster6	70:71:bc:ac:e4:d5

Langkah penyalaan *node* komputasi primer dilakukan dengan menjalankan skrip *shell* *wakeonlan.sh* sesuai dengan argumen alamat MAC *node* komputasi primer dari *front-end*. Kode skrip *shell* *wakeonlan.sh* terlampir di Lampiran 2.

4.2.2.7 Konfigurasi *Boot*

Node komputasi primer melakukan *boot* dari media *flashdrive*, hal ini menyebabkan kernel perlu menggunakan *initrd* (initial RAM disk). Konfigurasi *boot* dilakukan dengan melakukan instalasi paket kernel *generic* dan modul kernel, pembuatan *initrd* dan pengaturan *bootloader* lilo yang merupakan *bootloader* sistem operasi Salix64.

```
# installpkg kernel-generic-3.2.29-x86_64-1.txz
# cd /usr/share/mkinitrd/
# . mkinitrd_command_generator.sh -a "-w 10" > /boot/new.sh
# . /boot/new.sh
```

Initrd yang dihasilkan akan membuat kernel menunggu selama 10 detik sebelum melakukan *boot* ke partisi *root* karena perangkat *flashdrive* memerlukan waktu yang lebih lama untuk dikenali oleh kernel.

Konfigurasi *bootloader* diperlukan untuk mengatur parameter kernel agar *boot* dilakukan ke perangkat *flashdrive* yang telah dienumerasi menurut UUID oleh *udev*.

Berkas `/etc/lilo.conf` ditambahkan dengan entri:

```
image = /boot/vmlinuz-generic
initrd = /boot/initrd.gz
label = salix
append = "root=/dev/disk/by-uuid/1bb9bf67-3acf-4ae1-be72-
9c10653b1f95"
read-only
```

Langkah terakhir konfigurasi *bootloader* adalah instansiasi *bootloader* pada MBR dengan `lilo`.

```
# lilo -v
```

4.2.2.8 Duplikasi Media *Boot Node* Komputasi Primer

Duplikasi media *boot node* komputasi primer dilakukan dengan utilitas `dd`:

```
# dd if=/dev/sda of=/dev/sdb bs=4M
```

dengan `/dev/sda` adalah enumerasi *flashdrive* berisi sistem operasi yang telah dikonfigurasi dan `/dev/sdb` adalah enumerasi *flashdrive* yang belum dikonfigurasi.

Media *boot* yang hasil salinan hanya perlu mengulangi prosedur konfigurasi alamat IP dan *host* sesuai alamat IP dan nama *host* masing-masing.

4.2.3 Konfigurasi *Node* Komputasi Sekunder (10 Unit E7500)

Node komputasi sekunder adalah komputer tambahan yang digunakan pada tahap pengujian pengaruh cacah *node* komputasi, menggunakan alamat IP 192-168.0.1-192.168.0.10, mendapatkan berkas *executable* dan kunci publik SSH melalui direktori NFS.

Sistem operasi yang digunakan *node* komputasi sekunder adalah WattOS R6 yang merupakan *remaster* dari Ubuntu 12.04LTS, menggunakan kernel Linux 3.2.0, glibc 2.15, dan gcc 4.6.3

4.2.3.1 Konfigurasi Alamat IP dan *Hosts Node* Komputasi Sekunder

Konfigurasi nama *host node* komputasi sekunder dilakukan dengan mengubah berkas `/etc/HOSTNAME` sesuai nama *host* yang bersesuaian. Pengaturan alamat IP dilakukan melalui berkas `/etc/NetworkManager/system-connections/default` yang berisi:

```
[ipv4]
method=manual
addresses1=192.168.0.?;24;192.168.0.254;
```

dengan `192.168.0.?` disesuaikan dengan alamat IP yang bersesuaian. Tabel 4.6 menjelaskan asosiasi nama *host* dan alamat IP *node* komputasi sekunder.

Tabel 4.6 Asosiasi Nama *Host* dan Alamat IP *Node* Komputasi Sekunder

Nama Host	Alamat IP
link1	192.168.0.1
link2	192.168.0.2
link3	192.168.0.3
link4	192.168.0.4
link5	192.168.0.5
link6	192.168.0.6
link7	192.168.0.7
link8	192.168.0.8
link9	192.168.0.9
link10	192.168.0.10

4.2.3.2 Konfigurasi *User mpi*

Konfigurasi *user mpi* di *node* komputasi sekunder sama dengan konfigurasi *user mpi* di *front-end*.

4.2.3.4 Konfigurasi Klien NFS

WattOS R6 tidak mencantumkan klien NFS secara *default* sejak instalasi, sehingga paket terkait harus dipasang terlebih dahulu dengan program utilitas aptitude:

```
# apt-get install nfs-common
```

Setelah instalasi paket selesai, *node* komputasi sekunder dapat dikonfigurasi untuk melakukan mounting direktori NFS dari *front-end* secara otomatis tiap *boot* dengan menambahkan entri ke berkas */etc/fstab*:

```
192.168.0.11:/home/mpi /home/mpi nfs sloppy,rw 0 0
```

4.2.3.5 Konfigurasi SSH

Node komputasi sekunder mendapatkan kunci publik SSH *user mpi* secara otomatis setelah direktori *home user mpi* di-*mount* dan konfigurasi SSH di *front-end* telah dilakukan.

4.2.3.6 Kompilasi dan Instalasi OpenMPI *Node* Komputasi Sekunder

Node komputasi sekunder memerlukan hasil kompilasi yang berbeda terhadap hasil kompilasi karena konfigurasi *path* program dan pustaka yang berbeda antara Salix64 dan WattOS, perbedaan versi *gcc* dan kernel.

Langkah kompilasi Open MPI di *node* komputasi primer sama dengan langkah kompilasi di *front-end*, namun hasil kompilasi diarsip dalam berkas yang berbeda. Pengarsipan hasil kompilasi dilakukan dengan:

```
# tar -jcvf /tmp/openmpi-1.7-ubuntu.tar.bz2 /tmp/openmpi-1.7
```

Kompilasi hanya perlu dilakukan di satu *node* komputasi sekunder, *node* komputasi sekunder yang lain dapat menggunakan arsip hasil kompilasi untuk proses instalasi. Proses instalasi di *node* komputasi sekunder dilakukan dengan:

```
# tar -jxvf openmpi-1.7-ubuntu.tar.bz2 /tmp/openmpi-1.7
# cd /tmp/openmpi-1.7
# make install
# ldconfig
```

Proses instalasi telah berhasil jika *mpirun* dan *mpicc* dikenali dalam *path* sistem dan pustaka yang berkaitan berhasil dimuat.

4.2.4 Program Komputasi Integral Metode Balok (blokice)

Program komputasi paralel integral definit dimensi tiga metode balok diimplementasikan dengan bahasa pemrograman C. Kode program blokice.c terlampir (Lampiran 3).

Kompilasi program dilakukan dengan program gcc:

```
$ gcc blokice.c -o blokice
```

Eksekusi program dilakukan dengan melalui *shell* dengan format argumen:

```
$ ./blokice EXPRESSION X_MIN X_MAX X_NUM Y_MIN Y_MAX Y_NUM
           Z_MIN Z_MAX Z_NUM PROC_COUNT HOSTFILE
```

4.2.5 Program Komputasi Integral Metode Monte Carlo (montice)

Program komputasi paralel integral definit dimensi tiga metode Monte Carlo diimplementasikan dengan bahasa pemrograman C. Kode program montice.c terlampir (Lampiran 4)

Kompilasi program dilakukan dengan program mpicc:

```
$ gcc montice.c -o montice
```

Eksekusi program dilakukan dengan melalui *shell* dengan format argumen:

```
$ ./montice EXPRESSION X_MIN X_MAX Y_MIN Y_MAX Z_MIN Z_MAX
           SAMPLE_COUNT PROC_COUNT HOSTFILE
```

4.2.6 Skrip *Shell* untuk Otomasi Pengujian

Beberapa skrip *shell* digunakan untuk melakukan otomasi pengujian secara *batch*. Skrip sample.sh (Lampiran 4) digunakan untuk menjalankan program komputasi paralel metode Monte Carlo dengan beberapa cacah sampel berbeda. Skrip sample-blok.sh (Lampiran 5) digunakan untuk menjalankan program komputasi paralel metode balok dengan beberapa cacah sampel berbeda.

Skrip numprocs.sh (Lampiran 6) digunakan untuk menjalankan program komputasi paralel metode Monte Carlo dengan beberapa cacah *proses* berbeda. Skrip nodes.sh (Lampiran 7) digunakan untuk menjalankan program komputasi paralel metode Monte Carlo dengan beberapa cacah *node* komputasi berbeda.