



ISSN 0215 - 9617

TEKNO

VOLUME 09 /No.57/DESEMBER 2011

- **PEMANFAATAN CLUSTER PC UNTUK OPERASI PERKALIAN MATRIKS**, *Arie S.M. Lumenta, ST, MT.*
- **PERANCANGAN ALAT UKUR KECEPATAN DAN ARAH ANGIN**, *Ir. Benefit S. Narasiang*
- **KAJIAN TEORITIS PENGARUH SUDUT SUDU POMPA SENTRIFUGAL TERHADAP HEAD POMPA**, *I Nyoman Gede, ST, MT.*
- **ANALISIS DAN PERANCANGAN APLIKASI DORMITORY MANAGEMENT BERBASIS WEB DENGAN MENGGUNAKAN METODE USDP**, *Stanley Karouw, ST, MTI*
- **MITIGASI RESIKO KEAMANAN INFORMASI DENGAN PENINGKATAN KESADARAN PENGGUNA : STUDI KASUS KEAMANAN SISTEM INFORMASI**, *Jimmy R. Robot, ST, MTI*
- **ALGORITMA PENENTUAN JARAK DENGAN SENSOR IMU ACCELEROMETER**, *Ir. Benefit S. Narasiang*
- **PERENCANAAN STRATEGIS SI/TI BERBASIS MANFAAT BISNIS DENGAN MENGGUNAKAN METODOLOGI BE VISSTA PLANNING (BVP) STUDI KASUS : DINAS PENDAPATAN DAERAH SULAWESI UTARA**, *Stanley Karouw, ST, MTI*
- **SOLAR COOKER**, *Benny Lokombanua Maluegha*
- **AGEN PENJUALAN BERBASIS TEKNOLOGI INFORMASI : MENGUKUR DAMPAK DARI CRM BERBASIS TI PADA EFEKTIVITAS PENJUALAN**, *Nancy J Tuturoong, ST, MKom.*
- **ALAT PENDETEKSI TINGKAT STRESS PADA MANUSIA**, *Novi M. Tulung*
- **DETECTION EARLY BREAST CANCER BY USING DIGITAL INFRARED IMAGE BASED ON ASYMMETRY THERMAL**, *Dringhuzen Jekke Mamahit, ST, M.Eng.*
- **CONSTRUCTING AN AUTOMATIC WEATHER STATION**, *Benny Lokombanua Maluegha*
- **PEMBUATAN BUKA TUTUP PINTU SECARA OTOMATIS**, *Novi M. Tulung*

DEPARTEMEN PENDIDIKAN NASIONAL
UNIVERSITAS SAM RATULANGI
FAKULTAS TEKNIK

PEMANFAATAN CLUSTER PC UNTUK OPERASI PERKALIAN MATRIKS

ARIE S. M. LUMENTA, ST., MT.

Jurusan Teknik Elektro, Fakultas Teknik Universitas Sam Ratulangi Manado

Abstrak

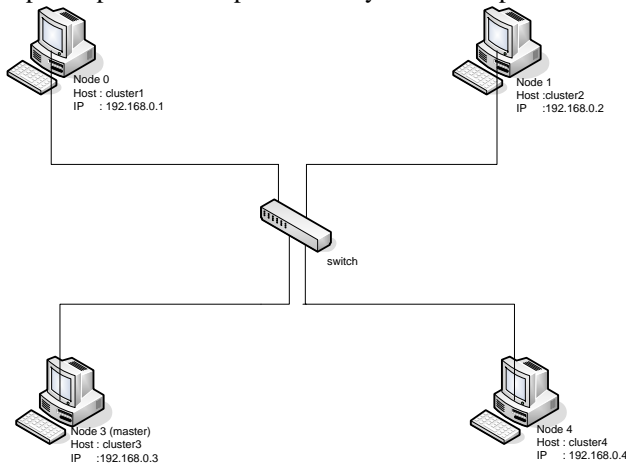
PC cluster pada dasarnya adalah sistem penyatuan beberapa PC atau server ke dalam satu kesatuan komputasi. Masing-masing PC tersebut akan merupakan unit kerja yang bekerja secara integral dengan PC-PC yang lain dalam menjalankan suatu tugas secara paralel. Dengan demikian beban untuk masing-masing PC akan menjadi jauh berkurang, dan konsekuensinya lower end PC dapat diberdayakan untuk mengerjakan tugas yang memerlukan komputasi intensif.

Dengan terbangunnya PC cluster tersebut, berbagai permasalahan komputasi dapat menjadi semakin ringan apalagi komputasi berskala besar seperti proses komputasi menyelesaikan perkalian matrix yang menggunakan banyak variabel dan banyak persamaan dengan adanya sistem PC cluster ini dapat memungkinkan proses komputasi menjadi lebih cepat jika dibandingkan dengan hanya di proses pada satu komputer saja.

Latar Belakang Masalah

PC cluster pada dasarnya adalah sistem penyatuan beberapa PC atau server ke dalam satu kesatuan komputasi. Masing-masing PC tersebut akan merupakan unit kerja yang bekerja secara integral dengan PC-PC yang lain dalam menjalankan suatu tugas secara paralel. Dengan demikian beban untuk masing-masing PC akan menjadi jauh berkurang, dan konsekuensinya lower end PC dapat diberdayakan untuk mengerjakan tugas yang memerlukan komputasi intensif.

Dengan terbangunnya PC cluster tersebut, berbagai permasalahan komputasi dapat menjadi semakin ringan apalagi komputasi berskala besar seperti proses komputasi menyelesaikan perkalian



Gambar 1 Tampilan konfigurasi jaringan LAN.

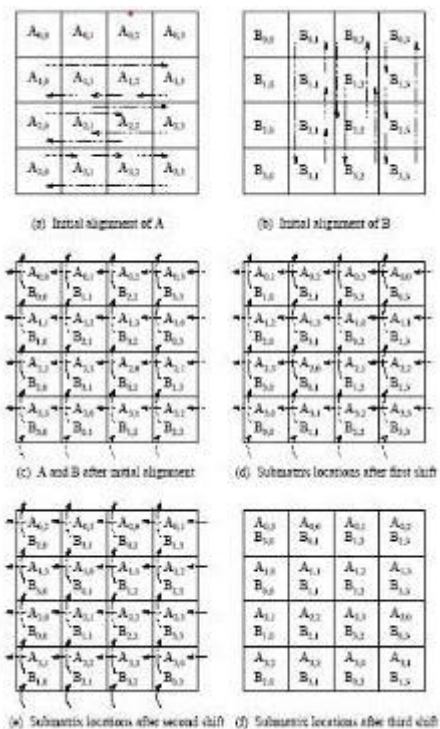
Dalam gambar diatas, tiap node client dan node master akan saling terhubung dengan sebuah switch, dimana tiap node akan mengerjakan suatu permasalahan bersama dalam hal ini, tiap node akan mengeksekusi sebuah program yang akan dieksekusi oleh node master.

Pengujian Sistem Menggunakan Sample Program MPICH

matrix yang menggunakan banyak variabel dan banyak persamaan dengan adanya sistem PC cluster ini dapat memungkinkan proses komputasi menjadi lebih cepat jika dibandingkan dengan hanya di proses pada satu komputer saja.

Rancangan Konfigurasi Jaringan LAN

Sistem cluster yang telah dibuat, dapat diterapkan pada jenis jaringan yang kecil hingga besar. Sistem dapat bekerja mulai dari pada jaringan yang dapat membentuk sebuah square yaitu $2n$. Berdasarkan penelitian, maka penulis membangun konfigurasi jaringan LAN disertai dengan contoh konfigurasi IP address masing-masing client adalah sebagai berikut:



Sebelum menguji sistem cluster menggunakan perkalian matriks, sistem ini perlu di uji apakah tiap

node sudah bisa bekerja sama memproses suatu masalah bersama. Dalam aplikasi MPICH sudah ada program bawaan yang bisa digunakan untuk menguji apakah sistem sudah berjalan dengan baik atau belum.

Pengujian dilakukan dengan proses kompilasi program tersebut dan melakukan running terhadap program tersebut:

```

root@cluster3: /mirror/mpich-1.2.7p1/examples
File Edit View Terminal Help
root@cluster3:~# cd /mirror/mpich-1.2.7p1/examples/
root@cluster3:/mirror/mpich-1.2.7p1/examples# mpirun -np 1 cpi
Process 0 on cluster3
pi is approximately 3.1416009869231254, Error is 0.0000083333333323
wall clock time = 0.000118
root@cluster3:/mirror/mpich-1.2.7p1/examples# mpirun -np 3 cpi
Process 0 on cluster3
Process 2 on cluster2
pi is approximately 3.1416009869231249, Error is 0.0000083333333318
wall clock time = 0.007249
Process 1 on cluster1
root@cluster3:/mirror/mpich-1.2.7p1/examples#

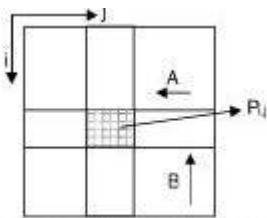
```

Gambar 2 pengujian cluster menggunakan program sample.

Proses Perkalian Matriks

Proses perkalian matriks dibuat dengan menggunakan algoritma canon. Algoritma canon merupakan salah satu algoritma yang dapat digunakan untuk melakukan perkalian matriks dengan pemakaian memory yang efisien. Berikut merupakan detail dari algoritma canon.

- Inisialisasi prosesor P_{ij} dengan elemen a_{ij} dan b_{ij} ($0 \leq i < n; 0 \leq j < n$)
- Elemen dipindahkan dari posisi awal sebagai berikut; matriks A baris ke- i diputar satu putaran ke kiri dan matriks B kolom ke- j diputar satu putaran keatas sehingga akan dihasilkan matriks baru dengan elemen $a_{i,j+i}$ dan $b_{i+j,j}$



Gambar 3 perputaran elemen matriks A dan B

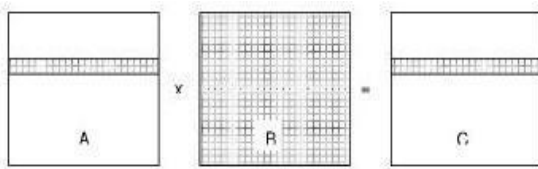
- Setiap prosesor P_{ij} melakukan perkalian masing – masing elemennya.
- Setelah proses 3 selesai, kemudian putaran matriks kembali dilakukan, dimana elemen baris ke- i matriks A akan diputar satu putaran ke kiri dan elemen kolom ke-1 matriks B akan diputar satu putaran ke atas.
- Kemudian setiap prosesor P_{ij} kembali melakukan perkalian elemen yang baru hasil

putaran pada langkah 4 dan hasilnya disimpan pada variabel jumlah total perkalian ini.

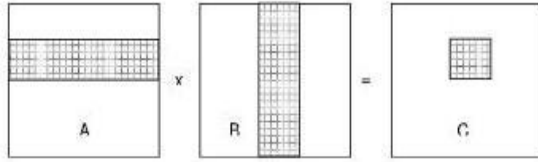
- Langkah 4 dan 5 dilakukan secara berulang sampai dengan seluruh elemen baris dan kolom diputar ($n-1$ putaran baris dan kolom)

Berikut desain foster untuk algoritma tersebut :

Partitioning & Agglomeration. Algoritma ini menggunakan teknik checkerboard untuk melakukan partisi data kedalam blok matriks yang lebih kecil. Misalkan matriks A dan B dibagi menjadi masing – masing p blok matriks. Masing – masing proses dari $P_{0,0}$ hingga $P_{\sqrt{p}-1, \sqrt{p}-1}$ akan memiliki blok A_{ij} dan B_{ij} . Agglomerasi dilakukan dengan orientasi baris, dimana setiap prosesor akan melakukan komputasi terhadap seluruh elemen C dan ini memerlukan akses terhadap seluruh elemen B (gambar 4.3). sesuai dengan teknik checkerboard diatas, setiap prosesor akan melakukan komputasi blok matriks C dengan ukuran $\left(\frac{n}{\sqrt{p}}\right) \times \left(\frac{n}{\sqrt{p}}\right)$, untuk melakukan komputasi elemen ini prosesor memerlukan referensi dari elemen matriks B sebanyak $\left(\frac{n}{\sqrt{p}}\right)$ baris dan $\left(\frac{n}{\sqrt{p}}\right)$ kolom (gambar 4)



Gambar a



Gambar b

Gambar 4 perbandingan banyaknya elemen A dan B yang perlu dihitung oleh setiap prosesor

Perkalian berdasarkan baris, setiap prosesor akan menghitung sebanyak n/p baris dari c dan perlu untuk mengakses sebanyak n/p baris A dan seluruh elemen B. Pada algoritma cannon, setiap prosesor akan melakukan komputasi sebanyak $\left(\frac{n}{\sqrt{p}}\right) \times \left(\frac{n}{\sqrt{p}}\right)$ blok matriks C. proses ini akan mengakses sebanyak $\left(\frac{n}{\sqrt{p}}\right)$ baris A dan B.

Setiap blok/proses akan melakukan komputasi sebanyak $\left(\frac{2n^3}{\sqrt{p}}\right)$, dimana setiap blok akan mengakses sebanyak $2n\left(\frac{n}{\sqrt{p}}\right)$ elemen. Rasio antara

komputasi dan komunikasi yang terjadi adalah $\frac{2n^3/p}{2n^3/p} = \frac{n}{\sqrt{p}}$

Communication, diperlukan untuk melakukan perputaran elemen matriks A dan B. Setiap proses pada baris ke-1 matriks A akan memindahkan isi blok tersebut ke kiri sebanyak 1 putaran, demikian juga pada kolom matriks B akan memindahkan isi bloknya sebanyak satu putaran ke atas

Gambar 5 proses komunikasi yang terjadi pada algoritma cannon dengan menggunakan sebanyak 16 prosesor/blok

Pada ilustrasi di atas terlihat proses komunikasi terjadi ketika algoritma melakukan inisialisasi elemen matriks dengan melakukan pergeseran elemen baris matriks A ke kiri dan kolom matriks B ke atas. Dan setiap blok melakukan perkalian elemennya. Dan kembali diputar dengan arah yang sama sampai dengan baris ke $n-1$ dan kolom ke $n-1$ juga. Proses putaran elemen ini melibatkan sebanyak $A_{i,(j+i) \bmod \sqrt{p}}$ dan $B_{i,(i+j) \bmod \sqrt{p},j}$.

Berikut cuplikan kode program untuk melakukan putaran elemen matriks sekaligus menggantikan nilai block dengan elemen yang baru tersebut :

```

/...../
/...../
/*putar blok & isikan nilai perhitungan*/
MPI_Cart_shift(GRID_COMM,1,-1,&source,&dest);
MPI_Sendrecv_replace(TA,S2,MPI_INT,dest,0,source,
0,GRID_COMM,&status);

```

```

MPI_Cart_shift(GRID_COMM,0,-1,&source,&dest);
MPI_Sendrecv_replace(TB,S2,MPI_INT,dest,0,source,
0,GRID_COMM,&status);
/...../

```

Cuplikan kode berikut merupakan topologi virtual yang digunakan pada algoritma cannon ini.

```

/...../
/...../
/*untuk buat topologi*/
M=(int)sqrt(nproc);
S=N/M; /*dimensi blok*/
S2=S*S; /*dimensi blok*/
dims[0]=dims[1]=M; /*dimensi topologi*/
periods[0]=periods[1]=1;
MPI_Cart_create(MPI_COMM_WORLD,2,dims,periods,0,&GRID_COMM);

MPI_Comm_rank(GRID_COMM,&grid_rank);
MPI_Cart_coords(GRID_COMM,grid_rank,2,coords);
myid=grid_rank;
source=coords[0];
dest=coords[1];
/...../

```

Proses Pengujian Perkalian Matriks Pada Sistem Cluster

Percobaan dilakukan pada 4 (empat) node yang mana terdapat node yang berfungsi sebagai master tempat input dan output dihasilkan dan node yang berfungsi sebagai slave tempat untuk melakukan sharing sumber daya.

Percobaan matriks dilakukan melalui console (shell). Perlu dilakukan beberapa kali percobaan untuk membuktikan kehandalan sistem pada saat eksekusi pada tabel 1 disajikan data hasil percobaan sistem menggunakan perkalian matriks.

Untuk melihat performa perkalian matriks pada cluster menggunakan algoritma cannon dilakukan perbandingan percobaan menggunakan 1 prosesor dan 4 prosesor. Hal ini disebabkan karena dalam melakukan perkalian matriks secara paralel di perlukan suatu topologi yang dapat membentuk sebuah square (1, 4, 9, 16, dst) agar dapat menjalankan algoritma cannon tersebut.

Pada penelitian ini penulis mengambil ukuran elemen matriks mulai dari elemen 100, 500, 1000 untuk setiap kali percobaan. Kecepatan sebuah sistem memproses data diukur untuk setiap percobaan dan di bandingkan

dengan pengujian pada 1 node. Berikut tabel hasil pengujian.

Tabel 1 Hasil Pengujian

| N | Jumlah prosesor | |
|------|-----------------|-----------|
| | NP 1 | NP 4 |
| 100 | 3.814936 | 20.017021 |
| 500 | 9.414158 | 17.109651 |
| 1000 | 14.475219 | 22.642907 |

Pada tabel diatas diperoleh bahwa pengujian sistem PC cluster menggunakan perkalian matriks pada 4(empat) prosesor menghasilkan speed up yang lebih cepat dibandingkan dengan pengujian pada 1(satu) prosesor. Berdasarkan tabel diatas tidak ditemukan

besaran dalam speed up yang dihasilkan dari proses komputasi, hal ini disebabkan karena berdasarkan hukum amdahl peningkatan kecepatan dari hasil proses pemrograman paralel tidak terdapat besaran yang menginsialisasikan speed up tersebut. Dalam sebuah sistem cluster tidak ada ukuran pasti bilamana sebuah komputasi digolongkan sebagai komputasi berskala besar atau komputasi berskala kecil, ukuran skalabilitas dari sebuah komputasi di tentukan dari perbandingan antara komputasi serial dan paralel. Berikut adalah hasil output dari sistem.

```

root@cluster3: /mirror/mpich-1.2.7p1
File Edit View Terminal Help
root@cluster3:/mirror/mpich-1.2.7p1# mpirun -np 1 canon
Process 0 of 1 on cluster3
speed up: 4.228000
root@cluster3:/mirror/mpich-1.2.7p1# mpirun -np 4 canon
Process 0 of 4 on cluster3
speed up: 21.900000
Process 2 of 4 on cluster2
Process 1 of 4 on cluster1
Process 3 of 4 on cluster1
root@cluster3:/mirror/mpich-1.2.7p1#

```

Gambar 6 Proses perkalian matriks dengan n = 100

```

root@cluster3: /mirror/mpich-1.2.7p1
File Edit View Terminal Tabs Help
root@cluster3: /mirror/mpich-1.2.7p1
root@cluster3: /mirror/mpich-1.2.7p1
root@cluster3:/mirror/mpich-1.2.7p1# /mirror/mpich-1.2.7p1/bin/mpirun -np 1 cano
n
Process 0 of 1 on cluster3
speed up: 4.237000
root@cluster3:/mirror/mpich-1.2.7p1# /mirror/mpich-1.2.7p1/bin/mpirun -np 4 cano
n
Process 0 of 4 on cluster3
speed up: 21.322000
Process 2 of 4 on cluster2
Process 1 of 4 on cluster1
Process 3 of 4 on cluster1
root@cluster3:/mirror/mpich-1.2.7p1#

```

Gambar 7 proses perkalian matriks dengan N = 500

```

root@cluster3: /mirror/mpich-1.2.7p1
File Edit View Terminal Tabs Help
root@cluster3: /mirror/mpich-1.2.7p1 x root@cluster3: /mirror/mpich-1.2.7p1 x
root@cluster3:/mirror/mpich-1.2.7p1# mpirun -np 1 canon
Process 0 of 1 on cluster3
speed up: 13.832000
root@cluster3:/mirror/mpich-1.2.7p1# mpirun -np 4 canon
Process 0 of 4 on cluster3
speed up: 24.656000
Process 2 of 4 on cluster2
Process 1 of 4 on cluster1
Process 3 of 4 on cluster1
root@cluster3:/mirror/mpich-1.2.7p1#

```

Gambar 8 Proses perkalian matriks dengan $N = 1000$

Berdasarkan penelitian Jika salah satu node terputus maka sistem akan membuat topologi dengan memanfaatkan 3 node lainnya yang akan membuat topologi square, namun jika ada 2 node yang terputus maka sistem tidak bisa lagi membuat topologi square menggunakan 2 node lainnya. Sehingga sistem tidak akan bisa dijalankan. Hal ini disebabkan karena topologi tersebut dibuat berdasarkan algoritma canon. Seperti yang sudah jelaskan diatas topologi virtual yang dibuat oleh algoritma canon akan mendistribusikan instruksi – instruksi dari PC master ke tiap – tiap PC slave dengan membuat sebuah square

agar algoritma tersebut dapat dijalankan untuk memecahkan perkalian matriks secara paralel tersebut.

Dan ketika jumlah PC yang tergabung dalam cluster tidak cukup untuk membuat topologi virtual yang dibangun oleh algoritma canon, maka sistem akan mengeluarkan error yang akan memberitahukan bahwa sistem cluster dengan implementasi perkalian matriks tidak dapat dijalankan, meskipun dalam cluster tanpa implementasi sistem sudah dapat dikatakan berhasil.

```

root@cluster3: /mirror/mpich-1.2.7p1
File Edit View Terminal Help
root@cluster3:~# cd /mirror/mpich-1.2.7p1/
root@cluster3:/mirror/mpich-1.2.7p1# mpirun -np 1 canon
Process 0 of 1 on cluster3
speed up: 15.320000
root@cluster3:/mirror/mpich-1.2.7p1# mpirun -np 4 canon
ssh: connect to host cluster1 port 22: Connection timed out
p0_4563: p4_error: Child process exited while making connection to remote process on cluster1: 0
root@cluster3:/mirror/mpich-1.2.7p1#

```

Gambar 9 Sistem cluster error

Penutup

Sistem PC cluster Merupakan sekumpulan komputer yang bekerja sama, dihubungkan oleh jaringan, sehingga dapat dipandang sebagai sebuah kesatuan, cluster komputer ini dikoordinasi oleh sebuah komputer induk yang bertugas untuk mendistribusikan pekerjaan kepada masing-masing komputer lainnya.

Komputasi dilakukan secara paralel menggunakan pemrograman paralel dengan menggunakan bahasa pemrograman C dan MPI.

Sistem cluster dapat meningkatkan speed up jauh lebih baik dari pada pengerjaan pada sistem serial. Jadi semakin banyak node yang dapat terhubung dalam sebuah cluster maka peningkatan kemampuan pun semakin optimal.

Perkalian matriks secara paralel menggunakan algoritma canan sebuah sistem harus membentuk sebuah square (1, 4, 9, 16, dst) untuk dapat membuat topologi dari algoritma tersebut. Namun dalam hal ini jika terdapat 4 node pemroses dan jika sebuah node mati maka sistem akan membuat topologi square menggunakan 3 node lainnya, tapi apabila ada 2 node yang mati maka system tidak dapat membuat topologi menggunakan 2 node sisa, sehingga sistem tidak bisa dijalankan.

Daftar Pustaka

Akbar A.H .2009. Tugas akhir mata kuliah pemrosesan paralel Algoritme fox untuk perkalian matriks. Bogor

Anonymous. 2007. Rancangan dan implementasi sistem keamanan pada paralel prosesing komputasi dengan menggunakan PC Cluster. Bandung

Argentini Gianluca. A generalization of Amdahl's law and relative conditions of parallelism. New Technologies and Models, Riello Group, Legnago (VR), Italy. 2002

Gropp William dan Lusk Ewing. Installation and User's Guide to MPICH, a Portable Implementation of MPI Version 1.2.7 The ch p4 device for Workstation Networks.

Hukum amdahl

http://en.wikipedia.org/wiki/Amdahl%27s_law

Hill. D. Mark. Amdahl's law in muticore era. University of Wisconsin-Madison. 2008

Hwang Kai. ADVANCE COMPUTER ARCHITECTURE. University of southern California.1993

Kinerja sistem paralel

[Http://repository.binus.ac.id/content/IF532/IF53285383.doc](http://repository.binus.ac.id/content/IF532/IF53285383.doc)

Kusno. 2001 Arsitektur Komputer dan Ukuran Kompleksitas. Jurusan matematika Fakultas MIPA Universitas Jember.

[Membuat Paralel Environment Menggunakan Mpich](http://blog.its.ac.id/kholis/2007/10/27/membuat-paralel-environment-menggunakan-mpich/)

[http://blog.its.ac.id/kholis/2007/10/27/membuat-paralel-environment-menggunakan mpich/](http://blog.its.ac.id/kholis/2007/10/27/membuat-paralel-environment-menggunakan-mpich/)

SSH tanpa password

<http://blog.its.ac.id/kholis/2007/10/27/ssh-tanpa-password/>

Network File System (NFS),

[http://lecturer.eepisits.edu/~dphoto/kuliah/jarkom/Buku Jaringan Komputer-Dphoto.pdf](http://lecturer.eepisits.edu/~dphoto/kuliah/jarkom/Buku_Jaringan_Komputer-Dphoto.pdf)