

Meningkatkan Performa dan Stabilitas Kecepatan Transfer Data pada FreeBSD dengan Kostumisasi Kernel

Handy Sunjaya, Deny Setiawan, Sandri Pratama

Program Studi Sistem Informasi, Universitas Multimedia Nusantara, Tangerang, Indonesia
handsunjaya@gmail.com, ds2010bc@gmail.com, sandri_pratama@yahoo.com

Diterima 5 Desember 2013

Disetujui 30 Desember 2013

Abstract—In this project, we demonstrated how to improve transfer rate when transferring data from a notebook installed FreeBSD operating system to another notebook installed Windows 8 operating system. The method which we used was with customizing the kernel on FreeBSD. Before we customized the kernel, we calculated the average time of the transfer speed with its default kernel. After we got the average time, then we customized its kernel and calculated the average time after we installed the custom kernel. At the end we got faster average time with the customized kernel and the performance better.

Index Terms— FreeBSD, Server, Transfer rate, Improving

I. PENDAHULUAN

FreeBSD adalah sebuah sistem operasi berbasis UNIX yang lebih mengutamakan fitur, kecepatan, dan stabilitas. FreeBSD adalah sistem operasi *open source*, *source code*-nya terbuka bagi siapa saja yang ingin mengembangkannya. Sebagian orang mungkin kurang familiar dengan sistem operasi ini, berbeda dengan Windows 8 yang hampir setiap orang mengetahuinya karena sistem operasi ini berbasis Windows dan banyak sekali penggunaannya. FreeBSD adalah sistem operasi yang minim akan *User Interface*-nya, hal ini yang membuat pengguna awam sulit untuk menggunakannya berbeda dengan sistem operasi Windows yang disajikan dengan *User Interface (UI)* yang bagus sehingga orang mudah untuk menggunakannya.

Sebuah server haruslah memiliki *performance* yang cepat. Untuk mendapatkan *performance* yang cepat maka dibutuhkan sebuah sistem operasi yang efisien baik itu dari segi penggunaan memori, pemakaian *bandwith* dan lain sebagainya. Banyak sistem operasi yang bisa digunakan untuk sebuah server seperti : *FreeBSD*, *Ubuntu Server*, dan *Windows server* yang masing-masing dari sistem operasi tersebut memiliki kelebihan dan kekurangannya tersendiri. Namun dalam penelitian ini sistem operasi yang kami gunakan untuk servernya adalah *FreeBSD*. Alasan kami memilih *FreeBSD* karena minimnya *GUI (Graphical User Interface)* dari sistem operasi ini yang kami anggap akan unggul dari segi kecepatan dibanding sistem operasi lainnya.

Dalam penelitian ini kami mencoba mengukur kecepatan transfer file dari sebuah *notebook* yang kami *install* sistem operasi *FreeBSD* sebagai servernya ke *notebook* lain yang kami *install* sistem operasi *Windows 8* sebagai *client*-nya. Setelah kami ukur dan dapat hasilnya, kami *custom kernel* dari sistem operasi *FreeBSD* yang kami *install* di *notebook* server lalu kami *install* dan kemudian kami ukur lagi kecepatan transfer *file*-nya dengan menggunakan *kernel* yang telah kami *custom*. Tujuan kami *custom kernel* dari *FreeBSD* ini adalah untuk meningkatkan kecepatan transfer *file* dari server ke *client*.

II. METODE PENELITIAN

A. System Specification

Untuk melakukan penelitian ini kami menggunakan dua *notebook*. *Notebook* pertama yang kami gunakan sebagai server dengan

spesifikasi :

- IntelCore2Duo 2.53GHz
- 100Mbps LAN
- 2GB RAM
- Sistem operasi FreeBSD 9.0.

Notebook kedua yang kami gunakan sebagai client dengan spesifikasi

- IntelCorei7 2.4 GHz
- 4GB RAM
- 100Mbps LAN
- Sistem operasi Windows 8.

B. Bahan Pendukung

Untuk dapat menghitung waktu rata-rata maka kami membutuhkan sebuah *file* dengan *size* yang cukup besar untuk memudahkan kami dalam menghitungnya. Alasan kami memilih menggunakan satu buah *file* dengan *size* yang besar adalah untuk memudahkan kami dalam menghitung waktunya, karena apabila *file* yang kami gunakan *size*-nya kecil tentu penghitungan waktu akan lebih sulit karena perbedaan waktunya sangat tipis. Karena kami membutuhkan sebuah *file* dengan *size* yang besar, maka kami menggunakan sebuah *file* video dengan format *.mkv dengan *size* sebesar 868.247 *Kilobytes*. *File* tersebut kami simpan di *notebook* satu yang kami gunakan sebagai server.

Untuk dapat mengakses *file* yang berada di server dari *client* kami membutuhkan sebuah *software* pendukung. *Software* yang kami gunakan adalah aplikasi bernama WinSCP yang kami *install* di *notebook* dua. Setelah aplikasi tersebut ter-*install* di *notebook* dua kami bisa melihat *file-file* apa saja yang ada di server dan dapat melakukan penarikan *file* tersebut dari server ke *client* dengan menggunakan kabel LAN.

C. Penarikan File

Setelah kedua *notebook* siap, kami mencoba menarik *file* tersebut dan mencatat waktu sampai *file* tersebut berhasil di-*copy* dari *notebook* satu ke *notebook* dua. Pada saat transfer *file* dari server ke *client* berlangsung kedua *notebook* sedang berada dalam status *charging* untuk memastikan semua *hardware* akan berjalan dengan kondisi normal. Kami melakukan penarikan *file* ini selama

sepuluh kali dan menghitung rata-ratanya. Kami melakukannya sebanyak sepuluh kali agar kami bisa menghitung rata-ratanya dan bisa melihat tingkat kestabilannya pada saat proses penarikan *file* dari setiap percobaan. Rata-rata yang kami dapat dalam percobaan pertama ini adalah hasil dengan menggunakan *setting-an kernel default*.

D. Customizing The Kernel

Setelah mendapat rata-rata waktu dengan *setting-an default*, kami lalu meng-*custom default kernel* dari *FreeBSD*. *Default kernel* dari sistem operasi *FreeBSD* ada di *file* 'GENERIC' yang terdapat di direktori /usr/src/sys/i386/conf/ (karena yang kami gunakan 32 bit). Kami meng-*custom* isi dari *file* 'GENERIC' tersebut dengan cara menduplikat *file* tersebut dengan nama baru dan meng-*custom* isinya dengan cara menonaktifkan modul yang kami aggap tidak diperlukan. Isi dari *file* 'GENERIC' adalah fitur-fitur dari sistem operasi *FreeBSD* dan *driver-driver* dari berbagai jenis *software*. Pada saat sistem operasi *FreeBSD* di-*install*, semua fitur-fitur dan *driver-driver* ini aktif secara otomatis dan di-*load* pada saat *notebook/PC* dinyalakan, sehingga pada *FreeBSD* kita tidak perlu meng-*install driver* untuk *hardware-hardware* kita karena semuanya sudah terpasang. Oleh karena tidak semua dari fitur-fitur dan *driver-driver* itu digunakan oleh *PC* kita, maka kita bisa menonaktifkan fitur-fitur dan *driver-driver* yang tidak kita gunakan. Kami berpikir apabila fitur-fitur dan *driver-driver* yang tidak kami gunakan itu dinonaktifkan, *performance PC* menjadi lebih baik. Berikut adalah contoh tampilan awal apabila kita membuka *file* 'GENERIC' di *FreeBSD* :

```

# GENERIC -- Generic kernel configuration file for FreeBSD/i386
#
# For more information on this file, please read the config(5) manual page,
# and/or the handbook section on Kernel Configuration Files:
#
#   http://www.FreeBSD.org/doc/en_US.ISO8859-1/books/handbook/kernelconfig-
#   fig.html
#
# The handbook is also available locally in /usr/share/doc/handbook
# if you've installed the doc distribution, otherwise always see the
# FreeBSD World Wide Web server (http://www.FreeBSD.org/) for the
# latest information.
#
# An exhaustive list of options and more detailed explanations of the
# device lines is also present in the ../conf/NOTES and NOTES files.
# If you are in doubt as to the purpose or necessity of a line, check first
# in NOTES.
#
$FreeBSD: release/9.1.0/sys/i386/conf/GENERIC 235077 2012-05-24 03:45:13Z mav
?
cpu             1486_CPU
cpu             1586_CPU
GENERIC: unmodified: line 1

```

Gambar 1

Berikut adalah contoh tampilan yang menunjukkan fitur-fitur yang ada di *FreeBSD* :

```
options SCHED_ULE # ULE scheduler
options PREEMPTION # Enable kernel thread preemption
options INET # InterNETworking
options INET6 # IPv6 communications protocols
options SCTP # Stream Control Transmission Protocol
options FFS # Berkeley Fast Filesystem
options SOFTUPDATES # Enable FFS soft updates support
options UFS_ACL # Support for access control lists
options UFS_DIRHASH # Improve performance on big directories
options UFS_GJOURNAL # Enable gjournal-based UFS journaling
options MD_ROOT # MD is a potential root device
options NFSCL # New Network Filesystem Client
options NFSD # New Network Filesystem Server
options NFSLOCKD # Network Lock Manager
options NFS_ROOT # NFS usable as /, requires NFSCL
options MSDOSFS # MSDOS Filesystem
options G9660 # ISO 9660 Filesystem
options PROCFSS # Process Filesystem (requires PSEUDOFSS)
options PSEUDOFSS # Pseudo-filesystem Framework
options GEOM_PART_GPT # GUID Partition Tables.
options GEOM_RAID # Soft RAID functionality.
options GEOM_LABEL # Provides labelization
options COMPAT_FREEBSD4 # Compatible with FreeBSD4
```

Gambar 2

Berikut adalah contoh tampilan yang menunjukkan *driver-driver* yang disediakan oleh *FreeBSD* :

```
# PCI Ethernet NICs that use the common MII bus controller code.
# NOTE: Be sure to keep the 'device miibus' line in order to use these NICs!
device miibus # MII bus support
device ae # Atansic/Atheros L2 FastEthernet
device ael # Atansic/Atheros L1 Gigabit Ethernet
device aic # Atheros AR8131/AR8132 Ethernet
device ale # Atheros AR8121/AR8113/AR8114 Ethernet
device bce # Broadcom BCM5706/BCM5709 Gigabit Ethernet
device bfe # Broadcom BCM440x 10/100 Ethernet
device bge # Broadcom BCM570x Gigabit Ethernet
device cas # Sun Cassini/Cassini+ and NS DP83065 Saturn
device dc # DEC/Intel 21143 and various workalikes
device et # Agere ET1310 10/100 Gigabit Ethernet
device fxp # Intel EtherExpress PRO/1000 (82557, 82558)
device gem # Sun GEM/Sun ERL/Apple GMAC
device hme # Sun HME (Happy Meal Ethernet)
device jme # JMicron JMC258 Gigabit/JMC260 Fast Ethernet
device lge # Level 1 LXT1001 gigabit Ethernet
device msk # Marvell/SysKonnect Yukon II Gigabit Ethernet
device nfe # nVidia nForce MCP on-board Ethernet
device nge # NatSemi DP83820 gigabit Ethernet
device nve # nVidia nForce MCP on-board Ethernet Networking
device pcn # AMD am79C97x PCI 10/100 (precedence over 'le')
device re # RealTek 8139C/-/8169/8169S/8110S
```

Gambar 3

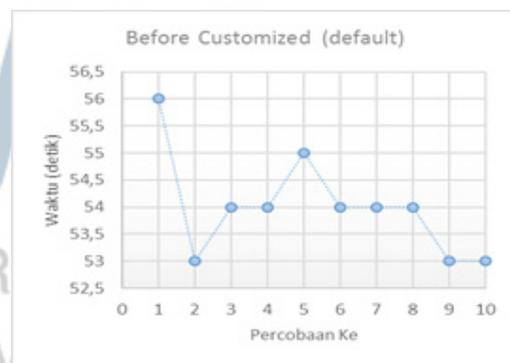
Cara untuk menonaktifkan fitur-fitur dan *driver-driver* yang kami anggap tidak diperlukan adalah dengan cara memberikan tanda '#' (tanpa kutip) di awal baris setiap fitur atau *driver* yang ingin kami nonaktifkan. Tanda '#' pada *FreeBSD* menandakan bahwa baris tersebut hanya-lah sebuah *command* sehingga baris tersebut tidak akan di-*compile* pada saat instalasi. Untuk melakukan kostumisasi *kernel* ini kami perlu mengetahui secara detil mengenai spesifikasi dari *notebook/PC* kami. Isi dari file *GENERIC* terbagi atas modul-modul seperti *PCI Ethernet*, *Wireless NIC card*, *USB Support* dan lain-lain. Di setiap modul tersebut akan tercantum jenis-jenis untuk setiap *hardware* untuk setiap modul tersebut, contohnya pada modul *Wireless NIC Card* terdapat jenis *Atheros*, *Broadcomm*, *Intel* dan lain-lain. Yang perlu kami lakukan adalah menonaktifkan jenis yang tidak sesuai dengan spesifikasi *notebook/PC* kami, misalnya *notebook* kami menggunakan *Atheros* maka kami perlu memberi tanda '#' pada *Broadcomm*, *Intel* dan yang lainnya selain *Atheros*. Oleh karena itulah diperlukan pengetahuan secara detil tentang *notebook/PC* kami dan kostumisasi untuk setiap

notebook/PC jelas berbeda.

Setelah selesai menonaktifkan fitur-fitur dan *driver-driver* yang kami anggap tidak perlu, *kernel* tersebut di-*build* dan di-*install*. Proses *customization* kami lakukan beberapa kali karena saat *kernel* di-*build* sering kali terjadi *error* karena ada kesalahan dalam menonaktifkan fitur atau *driver*. Kesalahan tersebut yaitu ada fitur atau *driver* yang seharusnya tidak dinon-aktifkan namun ternyata kami non-aktifkan serta saat pemberian tanda '#' terjadi perpindahan baris sehingga menimbulkan *error* pula. Setelah *kernel* berhasil di-*build* dan di-*install* *notebook* di-*restart* dan kami melakukan pengukuran penarikan *file* ulang dan membandingkan hasilnya dengan hasil yang sebelumnya saat menggunakan *kernel default*.

III. HASIL PENELITIAN

Adapun hasil dari percobaan pertama yang kami lakukan saat menggunakan *setting-an kernel default* adalah seperti berikut :



Tabel 1

Seperti terlihat pada tabel 1 hasil yang didapat rata-rata waktu yang dibutuhkan untuk transfer *file* video sebesar 868.247 Kilobytes adalah 54 detik. Terlihat juga dari tabel di atas jika dalam sepuluh kali percobaan waktu yang di dapat dari masing-masing percobaan berubah-ubah (tidak stabil).

Setelah *kernel* kami *custom* dan berhasil di-*build* dan di-*install* hasil yang kami dapat adalah sebagai berikut :



Tabel 2

Seperti terlihat pada tabel 2 hasil yang didapat rata-rata waktu yang dibutuhkan untuk transfer *file* video sebesar 868.247 *Kilobytes* adalah 52 detik. Terlihat juga dari tabel di atas jika dalam sepuluh kali percobaan waktu yang didapat dari masing-masing percobaan cenderung konstan (stabil).

IV. PEMBAHASAN

Seperti yang kita lihat dari hasil yang tergambar pada tabel pertama, waktu yang didapat dari sepuluh kali percobaan cenderung tidak stabil. Di awal penarikan kami membutuhkan waktu 56 detik, lalu di penarikan selanjutnya yaitu penarikan kedua sampai penarikan kesepuluh waktu yang didapat mengalami naik dan turun. Pada percobaan pertama ini waktu yang didapatkan tidak stabil. Dari sepuluh kali penarikan dipercobaan pertama ini rata-rata waktu yang dibutuhkan adalah 54 detik.

Setelah mendapat hasil tersebut kami meng-*custom kernel*-nya dan mendapatkan hasil seperti pada tabel kedua. Pada percobaan kedua waktu yang didapatkan dari penarikan pertama sampai penarikan kesepuluh cenderung stabil. Kenaikan hanya terjadi sekali di percobaan kelima yaitu 53 detik dan penurunan juga terjadi sekali di penarikan kesepuluh yaitu 51 detik. Dari percobaan kedua ini setelah *kernel*-nya di-*custom* rata-rata waktu yang dibutuhkan adalah 52 detik. Hasil ini menunjukkan waktu yang lebih stabil dari pada hasil pada tabel pertama. Selain peningkatan kestabilan rata-rata waktu yang didapat juga mengalami peningkatan rata-rata dari 54 detik menjadi 52 detik, terjadi peningkatan sekitar 2 detik dari hasil pertama.

Selain peningkatan kecepatan dan kestabilan, hasil lain yang kami peroleh setelah meng-*custom kernel* dari *FreeBSD* ini adalah waktu *booting* yang dibutuhkan oleh *FreeBSD* menjadi lebih cepat. Ini merupakan hasil yang tidak kami kami perkirakan sebelumnya. Hal ini mungkin disebabkan oleh banyak modul-modul yang tidak dibutuhkan yang kami nonaktifkan dari meng-*custom kernel* tadi oleh karena itu proses *booting* menjadi lebih cepat.

V. Simpulan

Penelitian kami ini memfokuskan kepada bagaimana meningkatkan *transfer rate* dan stabilitas saat transfer data dari *notebook* dengan sistem operasi *FreeBSD* sebagai server ke *notebook* dengan sistem operasi *Windows 8* sebagai *client* dengan cara meng-*custom file kernel*-nya. Dan hasilnya peningkatan waktu dan stabilitas pun didapat setelah meng-*custom kernel* itu. *Customization kernel* pada setiap *PC/Notebook* tentu berbeda karena setiap *PC/Notebook* memiliki *hardware* yang berbeda, oleh karena itu penting bagi kita untuk mengetahui secara detail tentang spesifikasi *PC/notebook* kita sebelum melakukan *customization kernel*. Dengan dilakukannya penelitian ini diharapkan dapat memberikan manfaat bagi kita dalam membangun sebuah server dengan sistem operasi *FreeBSD* agar dapat memberikan kecepatan dan kestabilan saat mentransfer data dari server ke klien-kliennya.

Satu hal yang bagi para peneliti yang ingin meneliti lebih jauh mengenai *customization kernel FreeBSD* adalah untuk tidak menonaktifkan *device* dalam modul yang bernama *FireWire support*, karena saat penelitian kami lakukan kami selalu mengalami *error*. Dan saat ditelusuri ternyata setelah *device* pada *FireWire support* kami aktifkan kembali *error* tersebut bisa diatasi. Jadi kami sarankan bagi para peneliti untuk tidak menonaktifkan *device* pada modul *FireWire support*. Selain itu kami juga menyarankan kepada para peneliti apabila ingin meng-*custom kernel FreeBSD*, perlu diperhatikan saat menonaktifkan setiap fitur atau *device* saat memberikan tanda '#' (tanpa kutip), jangan sampai saat setelah mengetikkan tanda '#' membuat baris baru karena terkadang saat mengetikkan tanda '#' teks akan berpindah ke baris baru. Hal ini akan

membuat *error* saat *kernel* di-*build* dan apabila hal itu terjadi *kernel* tidak bisa di-*install*. Hal-hal tersebut adalah hal-hal yang kami alami saat kami melakukan penelitian ini dan kami harap peneliti selanjutnya bisa mengantisipasi hal-hal seperti itu.

UCAPAN TERIMA KASIH

Kami mengucapkan terima kasih yang sebesar-besarnya kepada Bapak Yutinus Eko Soelistio,S.,Kom, M.M. selaku dosen mata kuliah Konsep Sistem Operasi kami yang telah memberikan kami tugas penelitian ini dan telah membimbing kami selama proses pengerjaan penelitian ini mulai dari awal sampai penelitian ini selesai.

DAFTAR PUSTAKA

- [1] A.S.Tanenbaum. Modern Operating System, Third Edition. Pearson International Edition. 2009.
- [2] Silberschatz, Galvin, Gagne. Operating System Concepts, 8th Edition. John Wiley & Sons Inc. 2008.
- [3] The FreeBSD Community. (2013-11-13). About FreeBSD [Media Online]. Alamat situs : [http:// www. freebsd. org / about.html](http://www.freebsd.org/about.html)
- [4] Cory Janssen. FREEBSD [Media Online]. Alamat situs : <http://www.techopedia.com/definition/3509/freebsd>