

MEMBANGUN SISTEM LINUX MANDRAKE MINIMAL MENGUNAKAN INISIAL DISK RAM

Wagito

*Jurusan Teknik Informatika
STMIK AKAKOM, Yogyakarta*

ABSTRACT

Minimal Linux system is commonly used for special systems like router, gateway, Linux installer and diskless Linux system. Minimal Linux system is a Linux system that use a few facilities of all Linux capabilities. Mandrake Linux, as one of Linux distribution is able to perform minimal Linux system.

RAM is a computer resource that especially used as main memory. A part of RAM's function can be changed into disk called RAM disk. This RAM disk can be used to run the Linux system. This paper explains how the Mandrake Linux system can be run in a RAM disk. The RAM disk which is used to run Linux system is loaded at the moment of initialization of Linux system. This RAM disk is called initial RAM disk (initrd). The Linux kernel which support initrd and a boot loader are needed to develop this system. The initrd file system that content Linux root system is created in this research. This initrd file is loaded into the RAM disk using a boot loader.

This experiment shows that the Linux system can be run in the RAM disk. The Linux system that can be run is minimal Linux system because of the RAM disk capacity limitation.

Keywords : minimal Linux, RAM disk, initrd

1. PENDAHULUAN

Nama Linux biasanya digunakan untuk menunjuk tiga hal yang mirip tetapi sebetulnya berbeda. Tiga hal tersebut adalah yaitu kernel Linux, sistem operasi Linux dan distribusi Linux (distro Linux). Sebagian besar orang, menggunakan istilah Linux sebagai suatu kumpulan dari sistem operasi, kernel, perangkat lunak sistem dan perangkat lunak aplikasi secara kolektif [5].

Sistem Linux didasarkan pada suatu perangkat lunak aras paling rendah yang mengatur semua perangkat keras komputer dan semua fungsi aras rendah. Perangkat lunak ini disebut kernel Linux. Fungsi-fungsi aras rendah akan digunakan oleh program lain sedemikian, sehingga pengembang program dapat menitikberatkan perhatiannya pada program yang dikembangkan sendiri. Kernel Linux mempunyai kemampuan umum sistem operasi modern seperti multitasking, thread, memori maya, pustaka berbagi (*shared library*), pemuatan permintaan, kemampuan berbagi file, copy-on-write executable, manajemen memori, pemuatan modul, video frame buffer dan jaringan TCP/IP.

Linux merupakan sistem operasi yang mulai banyak digunakan oleh pengguna komputer saat ini. Sebagai sistem operasi, saat ini sudah banyak aplikasi yang berjalan di atas sistem Linux baik berupa bahasa pemrograman, multimedia,

pengolah kata, spreadsheet, pengolah grafik, basis data maupun paket-paket komputer server. Semua aplikasi tersebut umumnya sudah tersedia pada distribusi Linux, hanya beberapa paket saja yang mungkin harus didownload dari internet.

Sebagai sistem operasi, Linux juga menyediakan bermacam-macam utilitas. Utilitas yang disediakan sistem operasi Linux meliputi shell (menyediakan *command shell* untuk menjalankan program), menyalin file, menghapus file, menjalankan peralatan jaringan dan masih banyak lagi. Sistem operasi Linux dapat melakukan tugas multi user maupun multi tasking. Linux dilengkapi dengan semua paket yang digunakan untuk tugas-tugas tersebut baik konektivitas jaringan, paket-paket komunikasi maupun interoperasi dengan sistem operasi lain.

Sistem operasi Linux juga didukung oleh pengembang-pengembang perangkat lunak yang ada di dunia. Perusahaan perangkat lunak menambahkan banyak sekali perangkat lunak tambahan, seperti sistem XWindow (XFree86), desktop (Gnome, KDE), permainan dan sebagainya. Semua perangkat lunak yang dikompilasi berdasarkan sistem operasi Linux disebut distribusi Linux.

Linux yang pertama kali dibangun oleh mahasiswa Finlandia bernama Linus Torvalds ini, merupakan sistem operasi Linux yang dibangun berdasarkan standar POSIX. Linux mula-mula dirancang untuk komputer pribadi berprosesor Intel 80386. Sekarang sistem Linux dapat dijalankan pada seluruh komputer yang ada di dunia baik yang berprosesor Intel maupun prosesor merk lain [5].

Saat ini banyak digunakan sistem Linux untuk keperluan khusus misalnya router, gateway, Linux diskless dan sebagainya. Untuk keperluan khusus ini, tidak diperlukan semua fasilitas yang dimiliki sistem operasi Linux. Dalam hal ini hanya diperlukan apa yang disebut sistem Linux minimal. Linux minimal pada dasarnya merupakan sistem Linux yang hanya menggunakan fasilitas secara terbatas dari sekian banyak fasilitas yang ada pada Linux. Linux Mandrake sebagai salah satu distro sistem operasi Linux yang banyak digunakan juga mampu untuk dibentuk sistem Linux minimal.

Umumnya, Linux minimal yang ada saat ini dibangun menggunakan media simpanan berupa disket. Beberapa yang sudah mengimplementasikan Linux minimal dengan disket adalah Freesco yang mana Linux minimal digunakan untuk aplikasi router [2]. Implementasi lain Linux minimal untuk router adalah FloopyFW [12]. Ada satu kelemahan ketika disket dipakai sebagai media simpan sistem Linux yaitu dalam hal kecepatan. Sistem Linux minimal yang dijalankan pada disket umumnya bekerja lambat terutama saat akan menjalankan utilitas yang disimpan pada disket.

Ada salah satu sumber daya komputer yang dapat digunakan untuk menggantikan disket sebagai media simpan sistem Linux yaitu RAM. Saat ini kapasitas RAM yang terpasang di komputer sangat tinggi dan harganya murah. Sebagian dari kapasitas RAM dapat diubah menjadi seperti disk. Sebagian RAM yang diubah menjadi disk ini disebut dengan disk RAM. Disk RAM ini dapat dipakai untuk menjalankan Linux. Tentu saja tidak semua kapasitas RAM yang tersedia pada komputer akan diubah menjadi disk RAM, karena aplikasi yang berjalan pada komputer tersebut juga memerlukan kapasitas tertentu.

Linux mempunyai kemampuan untuk membuat suatu disk RAM yang disebut inisial disk RAM (*initrd*) yang kapasitasnya mencapai 32 Mbyte. Inisial disk RAM ini akan diciptakan pada saat Linux memuat kernel. Kapasitas sebesar 32 Mbyte sudah cukup untuk menjalankan Linux dengan fasilitas yang disesuaikan. Tentu saja tidak semua utilitas yang dipunyai Linux akan disertakan, hanya sebagian saja yang disesuaikan dengan keperluan.

Pembahasan makalah ini akan dititikberatkan kepada bagaimana membangun sistem inisial disk RAM tersebut untuk menjalankan Linux secara minimal pada komputer pribadi. Untuk membuat sistem Linux ini digunakan distro Linux Mandrake 10.0 yang merupakan salah satu distro Linux dari sekian banyak distro yang ada. Distro Linux Mandrake sangat mudah digunakan karena distro ini menyediakan utilitas-utilitas tambahan khusus yang sangat memudahkan pengguna komputer.

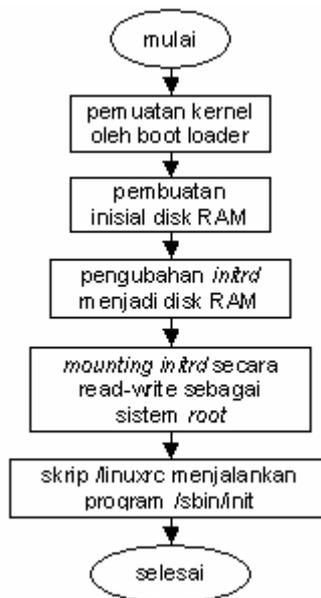
Manfaat sistem Linux pada disk RAM adalah untuk membuat sumber daya jaringan komputer tertentu yang tidak terlalu banyak memerlukan utilitas Linux. Contoh tentang ini adalah komputer router yang dibuat tanpa menggunakan harddisk. Sistem Linux pada router hanya memerlukan sedikit fasilitas dari Linux. Pengubahan ini akan menghilangkan kelemahan dalam hal kecepatan pada sistem router satu disket. Manfaat lain yang dapat diperoleh dengan adanya sistem Linux minimal dalam disk RAM antara lain sistem thin client pada terminal server, proses rekovery sistem dan sebagainya.

2. PROSES PEMUATAN SISTEM LINUX

Inisial disk RAM (*initrd*) menyediakan kemampuan untuk memuat suatu disk RAM melalui suatu proses yang dilakukan oleh boot loader. Disk RAM ini kemudian dapat *dimount* sebagai sistem file *root*, sehingga program dapat dijalankan dari *root* ini. Sebetulnya *initrd* dirancang untuk menjalankan sistem *startup* pada Linux yang terjadi dalam dua tahap yaitu memuat kernel yang berisi konfigurasi minimum driver yang sudah terkompilasi dan memuat modul-modul tambahan dari *initrd* [1].

Pada sistem Linux yang sesungguhnya, sistem file *root* yang ada pada *initrd* akan diganti oleh sistem file *root* yang baru yang dapat *dimount* dari divais lain yang berbeda. Kemudian, *root* pada *initrd* dapat dipindahkan pada suatu direktori untuk selanjutnya *diunmount*.

Pada rancangan ini dibuat suatu modifikasi yang mana *root* dari *initrd* tidak akan digantikan oleh *root* lain, tetapi tetap dipertahankan untuk menjalankan sistem Linux, sehingga sistem Linux tetap akan berjalan pada disk RAM. Supaya dapat menjalankan sistem Linux secara baik, pada *initrd* harus sudah mengandung semua utilitas yang diperlukan oleh sistem Linux. Dengan modifikasi ini, jalannya proses booting sistem Linux dapat dijelaskan dengan diagram alir berikut ini.



Gambar 1. Proses booting pada sistem Linux modifikasi

Langkah utama untuk menjalankan sistem Linux dalam disk RAM adalah langkah memuat kernel menggunakan boot loader dan membuat inisial disk RAM. Ada beberapa boot loader yang dapat digunakan, salah satu diantaranya adalah *Loadlin*. *Loadlin* merupakan perangkat lunak bebas yang dapat digunakan untuk memuat kernel Linux dari *command prompt* [4]. Dengan perangkat lunak ini, Linux dapat dijalankan dari *command prompt*. Tentu saja diperlukan perangkat lunak yang menyediakan *command prompt*. Untungnya semua sistem windows mempunyai fasilitas *command prompt*, sehingga rancangan ini dapat berjalan sesuai yang diinginkan. Paling tidak perangkat lunak *Loadlin* sudah dicoba dijalankan pada windows 95, windows 98 dan windows Millenium. Ada suatu catatan dari pengembang *Loadlin* bahwa untuk ukuran kernel Linux yang lebih besar dari satu Mbyte, harus digunakan *Loadlin* versi 1.6c [9]. Dengan demikian perlu dilihat apakah kernel yang digunakan Linux Mandrake 10.0 punya ukuran sebesar ini.

Loadlin merupakan perangkat lunak *command-line* yang tidak perlu dipasang melalui proses instalasi yang rumit. Untuk menjalankan *Loadlin* cukup menyalin file *loadlin.exe* pada sistem serta menjalankan *Loadlin* dari *command prompt*. Ada beberapa opsi parameter yang perlu diperhatikan dalam menjalankan *Loadlin*. Berikut ini diberikan beberapa opsi parameter yang digunakan untuk menjalankan sistem Linux yang akan dibuat [7,3].

```
ini trd=<file_image>
```

Opsi parameter *inltrd* digunakan untuk menunjukkan file apa yang akan digunakan sebagai inisial disk RAM. *file_image* merupakan file yang digunakan untuk membuat inisial disk RAM. File ini yang nantinya akan dirancang sebagai sistem Linux menggunakan disk RAM.

```
root=<nilai_s_root>
```

Opsi parameter *root* digunakan untuk mengarahkan lokasi divais sistem file *root*. Pada sistem Linux yang menggunakan disk RAM, opsi ini diarahkan untuk menunjuk disk RAM. Dalam sistem Linux yang menggunakan *devfs*, disk RAM diwakili oleh suatu file `/dev/rd/0` [8], sedangkan pada sistem Linux yang tidak menggunakan *devfs*, disk RAM diwakili oleh file `/dev/ram0`. Perlu diketahui bahwa semua divais yang ada pada sistem Linux, diwakili oleh suatu file.

`init=<skrip_p_init>`

Opsi parameter *init* menentukan skrip file apa yang akan digunakan untuk menjalankan inisialisasi sistem. Jika tidak menyebutkan nama skrip file inisialisasi, maka skrip file yang dijalankan untuk inisialisasi sistem Linux adalah `/linuxrc`. Untuk sistem Linux yang banyak beredar saat ini, skrip ini digunakan untuk menjalankan skrip file `/sbin/init`.

`devfs=<mount atau nomount>`

Opsi parameter *devfs* digunakan untuk menentukan apakah sistem Linux akan menggunakan sistem *devfs* atau tidak. Sistem *devfs* merupakan suatu sistem untuk mendefinisikan divais file pada Linux [6]. Ada dua pilihan untuk opsi *devfs* yaitu *mount* atau *nomount*. Jika opsi ini diberi nilai *mount*, maka sistem Linux akan menggunakan sistem *devfs*, sedangkan jika opsi ini diberi nilai *nomount*, maka sistem Linux tidak akan menggunakan sistem *devfs*.

`vga=<mode>`

Opsi parameter *vga* akan menentukan mode konsol yang akan digunakan untuk inisialisasi Linux. Jika opsi ini tidak disebutkan, maka inisialisasi Linux akan menggunakan mode teks.

Opsi parameter yang lain dalam kaitannya dengan menjalankan *Loadlin* masih cukup banyak. Dokumentasi lengkap tentang opsi parameter dalam menjalankan *Loadlin* dapat dilihat dari dokumen `params.doc` yang tersedia pada situs pengembang *Loadlin*. Dokumen lain yang dapat dijadikan acuan untuk menjalankan *Loadlin* adalah `BootPrompt-HOWTO.html` yang disediakan oleh beberapa situs internet.

3. MEMBANGUN INISIAL DISK RAM (*INITRD*)

Inisial disk RAM (*initrd*) dapat dibangun berdasarkan sistem Linux yang sudah ada. Sistem yang dibangun, akan menggunakan distro Mandrake 10.0 sebagai acuan, sehingga semua file sistem diambil dari distro Linux ini. Sebagai asumsi, bahwa sistem Linux yang akan dibangun ini adalah sistem Linux yang berjalan pada disk RAM dan sistem Linux ini menggunakan *devfs*. Dengan asumsi ini, kernel harus mendukung dua hal tersebut yaitu *devfs* dan *initrd*. Kernel Linux Mandrake 10.0 sudah mendukung dua hal tersebut, sehingga langsung dapat digunakan dalam percobaan ini. Sistem *initrd* yang dapat dibuat berdasarkan kernel Linux ini mempunyai nilai maksimum 32 Mbyte atau kira-kira setara 30 disket. Pada penelitian ini hanya akan digunakan sebesar seperempat kapasitas maksimum yang diijinkan atau sekitar 8 Mbyte.

Untuk membangun inisial disk RAM, harus dibuat file *image* yang berisi sistem Linux. Sistem file ini dibangun berdasarkan distro Linux yang digunakan sebagai acuan. Membangun file *image* disk RAM dilakukan dengan cara membuat sistem file dalam suatu divais blok, menyalin semua file yang diperlukan dan

menyalin isi divais blok pada file *initrd*. Dengan kernel yang ada, paling tidak ada tiga tipe divais yang dapat digunakan yaitu disk floppy, disk RAM dan divais *loopback* [1]. Pada makalah ini akan digambarkan bagaimana membangun file *image* disk RAM dengan divais *loopback*. Beberapa langkah yang harus dilakukan untuk membangun sistem ini dapat diuraikan sebagai berikut.

3.1 Membangun File *Image initrd*

Untuk membangun File *image initrd* diperlukan dua utilitas yang secara umum sudah disediakan oleh sistem Linux standar yaitu *dd* dan *mke2fs*. Utilitas *dd* digunakan untuk menyalin suatu file (dari input standar tertentu menjadi output standar tertentu pula) menggunakan ukuran blok input dan output tertentu [10]. Sedangkan utilitas *mke2fs* digunakan untuk memformat file menjadi file sistem, sehingga dapat digunakan dalam sistem Linux [11].

Utilitas *dd* ini akan membaca input satu blok dalam satu waktu menggunakan ukuran blok tertentu. Output blok akan mempunyai ukuran yang sama persis dengan ukuran blok yang dibaca dari input. Ukuran blok serta jumlah blok dapat ditentukan menggunakan parameter *bs* dan *count* [10].

```
# dd if=<file_in> of=<file_out> bs=<blocksize> count=<count>
```

Maksud perintah tersebut adalah menyalin isi input standar dari *file_in* dan menuliskan hasilnya pada *file_out* sebagai output standar sejumlah *blocksize* byte dalam satu waktu dan dilakukan pembacaan sebanyak *count* kali. Dengan demikian ukuran *image* menjadi (*count* × *blocksize*) byte. *file_in* maupun *file_out* dapat berupa input maupun output standar. Ukuran *bs* dapat ditentukan dengan satuan *b* (512), *k* (1024), atau *M* (1024 × 1024). Ukuran ini harus disesuaikan dengan kebutuhan kapasitas disk RAM yang akan diciptakan.

Selanjutnya file *image* harus diformat supaya menjadi file sistem. File sistem yang biasanya digunakan dalam sistem Linux adalah *ext2* atau *ext3*. Untuk membuat file sistem ini, Linux menyediakan perintah *mke2fs*. Secara umum bentuk perintah *mke2fs* adalah sebagai berikut [11].

```
# mke2fs -F -b <blocksize> file
```

Perintah ini akan memformat file dengan ukuran blok sebesar *blocksize*. Parameter *file* dalam hal ini merupakan hasil keluaran dari perintah *dd* tersebut yaitu *file_out*. *file_out* ini merupakan file *image* yang diperlukan untuk membangun inisial disk RAM. Opsi *-F* kadang-kadang perlu disertakan supaya *mke2fs* tetap dijalankan meskipun file bukan suatu divais. Kadang-kadang disertakan pula opsi *-L* untuk memberi label pada file sistem tersebut. Dengan cara ini, maka file sudah berubah menjadi suatu file sistem yang siap diisi dengan program atau utilitas yang diperlukan untuk membuat sistem Linux termasuk dalam hal ini, susunan direktori.

3.2 Membangun File Sistem *Root*

Untuk membangun file sistem *root*, paling mudah adalah dengan membuat kembaran dari sistem Linux yang sudah ada. Langkah pertama untuk membangun file sistem *root* adalah menyiapkan suatu direktori khusus yang akan dialokasikan sebagai tempat untuk membangun sistem ini. Misalnya, direktori dimaksud diberi nama *initrd* dan diletakkan di bawah */tmp*. Perintah yang digunakan adalah.

```
# mkdir /tmp/initrd
```

Nama direktori sebetulnya bebas, tidak harus seperti di atas dan letaknya juga bebas. File sistem *image* yang sudah dibuat pada langkah sebelumnya, *dimount* pada direktori ini menggunakan opsi *loop*.

Untuk membuat kembaran sistem file *root* dari sistem Linux acuan, cukup menyalin semua file yang diperlukan dalam direktori */tmp/initrd* tersebut. Karena yang akan dibuat adalah sistem Linux minimal, maka harus dipilih file apa saja yang diperlukan supaya sistem ini dapat berjalan. Selain file-file tersebut, tentu saja pustaka yang diperlukan oleh file yang bersangkutan juga harus disertakan. Untuk mengetahui pustaka apa saja yang diperlukan suatu file, dapat digunakan perintah *ldd*. Perintah ini akan menunjukkan file pustaka yang digunakan secara berbagi oleh file yang dimaksud. File-file pustaka ini harus disalin pada direktori yang bersesuaian [13].

Jika semua file yang diperlukan sudah disalin pada direktori */tmp/initrd*, maka file *image initrd* dapat diunmount. Dengan cara ini, maka file *image initrd* sudah berisi file sistem *root* yang diperlukan untuk menjalankan sistem Linux dalam disk RAM. File *image initrd* selanjutnya dapat dimampatkan dengan *gzip* supaya ukurannya menjadi lebih kecil.

4. IMPLEMENTASI DAN PEMBAHASAN

Pada bagian ini diberikan contoh implementasi dari langkah-langkah yang sudah diuraikan pada bagian sebelumnya. Misalnya diinginkan untuk membangun sistem Linux menggunakan disk RAM dengan kapasitas 8 Mbyte.

Ada dua langkah utama untuk membangun sistem Linux yang berjalan pada disk RAM dan dijalankan dari windows yaitu membangun file *image initrd* dan menyusun file pada direktori Windows.

4.1 Langkah Membangun File *Image initrd*

Untuk membangun file *image initrd* diperlukan beberapa langkah. Secara umum langkah-langkah yang dilakukan dalam membangun file *image initrd* diuraikan pada bagian berikut ini.

a. Membuat file *image* sistem *initrd*

File *image* sistem *initrd* dibuat menggunakan utilitas *dd*. Dengan utilitas ini, dapat dibuat suatu file *image* dengan ukuran sesuai dengan yang diperlukan. Dalam penelitian ini dibuat file *image* sebesar 8 Mbyte.

```
# dd if=/dev/zero of=initrd bs=8M count=1
```

Perintah ini, akan menyebabkan dibuatnya suatu file *image initrd* dengan ukuran 8 Mbyte. File ini dibuat berdasarkan input standar divais */dev/zero*.

b. Format file *image initrd*

File *image* yang sudah dibuat menggunakan utilitas *dd* harus diformat menggunakan supaya menjadi suatu sistem file yang didukung oleh Linux. Sistem file yang bisa digunakan dalam file *image* adalah *ext2*, sehingga proses format dilakukan dengan perintah berikut ini.

```
# mke2fs -F -b 1024 -L Linuxmini initrd
```

Dengan perintah ini, maka file *image initrd* akan diformat menjadi sistem file *ext2* dengan ukuran masing-masing blok sebesar 1024 dan diberi label *linuxmin*. Ukuran blok 1024 merupakan ukuran minimal yang dapat ditempati oleh suatu file atau direktori. Jika ukuran file lebih kecil dari 1024 byte, maka file tersebut akan disimpan dalam ruang sebesar 1024 Byte.

- c. Membuat lokasi untuk *mounting initrd*
Langkah berikutnya adalah membuat suatu direktori sebagai tempat untuk melakukan proses *mount* file image. Direktori yang dibuat pada penelitian ini adalah `/tmp/initrd`. Nama direktori bukan merupakan hal yang dipentingkan, sehingga nama tersebut dapat diubah sesuai dengan yang diinginkan.

```
# mkdir /tmp/initrd
```

Perintah ini akan menyebabkan dibuatnya direktori `/tmp/initrd` sebagai lokasi yang disiapkan untuk membangun sistem file *root*.
- d. Mounting file image *initrd*
Setelah direktori siap digunakan, file *image* dapat *dimount* pada direktori tersebut. Proses *mounting* didasarkan pada opsi *loop* dan sistem file *ext2*. Opsi *loop* merupakan opsi yang biasa digunakan untuk proses *mounting* suatu file *image*.

```
# mount -o loop -t ext2 initrd /tmp/initrd
```

Perintah ini akan menyebabkan file *image initrd* *dimount* pada direktori `/tmp/initrd` menggunakan opsi *loop* dan file sistem *ext2*. Selanjutnya dapat disusun suatu sistem file *root* pada direktori tersebut.
- e. Membuat susunan direktori sistem Linux
Berikut ini diberikan contoh bagaimana susunan direktori yang merupakan kembaran sistem Linux. Sistem Linux umumnya mempunyai susunan direktori `/bin`, `/etc`, `/lib`, `/proc`, `/tmp`, `/boot`, `/home`, `/mnt`, `/root`, `/usr`, `/dev`, `/initrd`, `/opt`, `/sbin`, dan `/var`. Untuk membuat direktori ini, silakan pindah pada direktori `/tmp/initrd`, kemudian dibuat direktori-direktori tersebut dengan perintah `mkdir` untuk tiap-tiap direktori.

```
# cd /tmp/initrd
# mkdir bin
# mkdir boot
```

dan seterusnya sampai seluruh direktori terbentuk.
- f. Menyalin semua file yang diperlukan
Berikut ini disajikan tabel seluruh file yang diperlukan untuk menjalankan sistem Linux minimal pada disk RAM.

Tabel 1. Daftar File Pada Direktori /bin

awk	chmod	echo	login	sleep
basename	chgrp	egrep	ls	sort
bash	cp	fgrep	mkdir	stty
bash2	cut	gawk	mount	su
cat	date	grep	mv	touch
conso lechars	dd	gunzip	rbash	true
dnsdomainname	df	gzip	rm	umount
domainname	dmesg	id	rmdir	uname
nsdomainname	chown	kill	sed	usleep
ypdomainname	hostname	ln	sh	zcat

Tabel 2. Daftar File Pada Direktori /sbin

Clock	halt	mi ngetty	setsysfont
Consol etype	hwclock	mi nilogd	shutdown
Depmod	init	mke2fs	sulogin
Devfsd	initlog	modprobe	swapoff
Fsck	insmod	pidof	swapon
fsck.ext2	killall5	poweroff	sysctl
Fuser	losetup	reboot	telinit
Getkey	lsmo d	runlevel	update

Tabel 3. Daftar File Pada Direktori /etc

adjtime	issue.net	profile	rc.local
bashrc	ld.so.cache	rc	rc.sysinit
devfsd.conf	ld.so.conf	rc0.d	redhat-release
fstab	mandrake-release	rc1.d	resolv.conf
group	modprobe.conf	rc2.d	securetty
gshadow	modules.devfs	rc3.d	shadow
info-dir	motd	rc4.d	shells
inittab	mtab	rc5.d	sysctl.conf
issue	passwd	rc6.d	

Tabel 4. Daftar File Pada Direktori /etc/pam.d

Chfn	kbd rate	other	su
Chsh	login	passwd	system-auth

Tabel 5. Daftar File Pada Direktori /etc/security/console.apps

halt	net_monitor	reboot	xserver
kbd rate	poweroff	shutdown	

Tabel 6. Daftar File Pada Direktori /etc/security

access.conf	group.conf	pam_env.conf
console.perms	limits.conf	time.conf

Tabel 7. Daftar File Pada Direktori /lib

ld-2.3.3.so	libcom_err.so.2	libproc.so.3.1.15
ld-linux.so.2	libcom_err.so.2.1	libpthread-0.10.so
libblkid.so.1	libcrypt-2.3.3.so	libpthread.so.0
libblkid.so.1.0	libcrypt.so.1	librt-2.3.3.so
libc-2.3.3.so	libext2fs.so.2.4	librt.so.1
libc.so.6	libgpm.so.1	libtermcap.so.2
libdl-2.3.3.so	libgpm.so.1.19.0	libtermcap.so.2.0.8
libdl.so.2	libnss_files-2.3.3.so	libthread_db-1.0.so
libe2p.so.2	libnss_files.so.2	libthread_db.so.1
libe2p.so.2.3	libpam_misc.so.0	libuuid.so.1
libm-2.3.3.so	libpam_misc.so.0.77	libuuid.so.1.2
libm.so.6	libpam.so.0	libz.so.1
libnsl-2.3.3.so	libpam.so.0.77	libz.so.1.2.1
libnsl.so.1	libpcre.so.0	

Tabel 8. Daftar File Pada Direktori /usr/bin

chvt	killall	locale
dircolors	ldd	tty

Tabel 9. Daftar File Pada Direktori /usr/lib

libconsole.so.0	libglib-1.2.so.0	libconsole.so.0.0.0
libctutils.so.0	libglib-1.2.so.0.0.10	libctutils.so.0.0.0

Tabel 10. Daftar File Pada Direktori /lib/security

pam_console.so	pam_console_apply_devfsd.so	pam_env.so
pam_limits.so	pam_cracklib.so	pam_permit.so
pam_nologin.so	pam_securetty.so	pam_stack.so
pam_unix_acct.so	pam_unix_passwd.so	pam_unix.so
pam_unix_auth.so	pam_unix_session.so	

Selain semua file yang disebutkan pada Tabel 1 sampai dengan Tabel 10 di atas, masih ada beberapa file yang diperlukan untuk konfigurasi minimal sistem Linux Mandrake. File-file tersebut berupa skrip yang diletakkan pada direktori /etc/rc.d dan /etc/rc.d/init.d. File skrip pada direktori /etc/rc.d digunakan untuk melakukan inisialisasi sistem Linux Mandrake yang meliputi skrip rc, rc.local, rc.modules dan rc.sysinit. File skrip pada direktori /etc/rc.d/init.d digunakan untuk menyiapkan Linux Mandrake supaya dapat memberi layanan khusus. Pada sistem Linux ini disertakan beberapa file skrip untuk keperluan layanan minimal yaitu devfsd, functions, halt, killall, single dan usb. Supaya dapat menjalankan layanan minimal diperlukan beberapa file konfigurasi yang diletakkan pada direktori /etc/sysconfig yang meliputi file autofscck, clock, i18n, init, keyboard dan network.

Selain file konfigurasi, pada sistem Linux ini juga diperlukan beberapa file yang berhubungan dengan font pada konsol. File-file tersebut diletakkan pada direktori /etc/sysconfig/console. Meskipun demikian, file-file ini tidak terlalu dipentingkan dalam inisialisasi sistem Linux.

Dari sekian banyak file yang diperlukan supaya sistem Linux dapat bekerja, ada beberapa file yang harus dimodifikasi disebabkan sistem Linux yang akan dijalankan menggunakan disk RAM sebagai tempat sistem file *root*. File yang dimodifikasi ada dua macam yaitu: /etc/fstab dan /etc/rc.d/rc.sysinit.

File /etc/fstab dimodifikasi sedemikian, sehingga berisi baris-baris sebagai berikut

```
/dev/ram0 / ext2 defaults 1 1
none /proc proc defaults 0 0
```

Baris di atas menunjukkan bahwa /dev/ram0 akan dimount sebagai file sistem *ext2*. /dev/ram0 ini akan berfungsi sebagai *root*.

Selanjutnya modifikasi pada /etc/rc.d/rc.sysinit dilakukan untuk mencegah pemeriksaan terhadap divais /dev/ram0. Pemeriksaan ini biasanya dilakukan terhadap divais blok yang akan dimount pada sistem Linux. Karena divais /dev/ram/0 bukan merupakan divais blok, pemeriksaan akan menimbulkan kesalahan. Kesalahan yang terjadi bukan disebabkan kesalahan divais, tetapi kesalahan penerapan prosedur pada objek yang

salah.. Beberapa baris perintah dan skrip /etc/rc.d/rc.sysinit harus dibuat menjadi tidak aktif. Baris-baris yang dimaksud adalah sebagai berikut.

```
#ROOTFSTYPE=`awk '/\// && ($3 != ~ /rootfs/){print $3}' /proc/mounts`
# if [ -z "$fastboot" -a "$X$ROOTFSTYPE" != "Xnfs" ]; then
#
#     gprintf "Checking root filesystem\n"
#     fsck -T -a $fsckoptions /
# fi
```

g. Unmounting file image *initrd*

Setelah semua file yang diperlukan sistem file *root* dimasukkan pada direktori tempat file *image* dimount, proses selanjutnya adalah *unmount* direktori tersebut. Proses *unmount* ini akan menyebabkan semua file yang dimasukkan pada direktori akan terbawa masuk dalam file *image*,

```
# unmount /tmp/initrd
```

Perintah ini menyebabkan file *image initrd* yang sebelumnya dimount pada direktori /tmp/initrd menjadi diunmount. Proses ini akan menyebabkan seluruh isi /tmp/initrd terbawa ke dalam file *image initrd*. Dengan demikian, file *image initrd* sudah berisi sistem file *root*.

h. Memampatkan file *image initrd*

Biasanya file *image* yang digunakan sebagai inisial disk RAM ukurannya cukup besar. Dalam penelitian ini ukurannya 8 Mbyte. File ini dapat dimampatkan menggunakan utilitas *gzip* supaya ukurannya menjadi lebih kecil. Pemampatan ini dapat mencapai ukuran sepertiga dari ukuran semula.

```
# gzip initrd
```

Perintah ini menyebabkan file *image initrd* dimampatkan, sehingga ukuran file menjadi lebih kecil. File yang dihasilkan secara otomatis diberi ekstensi *.gz, sehingga akan terbentuk *initrd.gz*. File ini merupakan *image* disk RAM yang diinginkan.

4.2 Langkah Menyusun Direktori Pada Windows

Langkah ini dilakukan untuk menyiapkan sistem Windows supaya dapat menjalankan sistem Linux. Misalkan file-file yang akan digunakan untuk menjalankan sistem Linux pada disk RAM disimpan dalam folder *c:\linux*. Folder ini harus berisi beberapa file yaitu

vmlinuz : merupakan kernel Linux

initrd.gz : file *image* disk RAM yang berisi sistem file *root* Linux

loadlin.exe : program yang berfungsi sebagai boot loader

boot.bat : file batch untuk menjalankan sistem Linux

Kernel *vmlinuz* dapat diperoleh dari direktori /boot yang ada pada sistem Linux acuan yang dipakai. Kernel ini sudah mendukung beberapa hal yang diperlukan pada sistem Linux ini yaitu mendukung *devfs* dan *initrd*. File *image initrd.gz* diperoleh dari langkah sebelumnya, dan hasilnya dapat disalin pada folder tersebut. File *loadlin.exe* dapat diperoleh dari situs yang menyediakan dan dapat didownload secara bebas.

Dari empat file tersebut, file yang harus dibuat adalah boot.bat. Isi dari file boot.bat adalah sebagai berikut.

```
cd c:\linux
loadlin vmlinuz initrd=initrd.gz root=/dev/ram0
init=/sbin/init devfs=mount
```

Perintah pertama digunakan untuk beralih menuju folder c:\linux, sedangkan perintah kedua digunakan untuk memuat kernel vmlinuz dan inisial disk RAM initrd.gz. Dari perintah tersebut diketahui bahwa beberapa parameter yang dipakai yaitu: sebagai *root* adalah /dev/ram0, sebagai program *init* digunakan /sbin/init, dan sistem Linux menggunakan *devfs*. Untuk menjalankan sistem Linux ini cukup diketikkan c:\linux\boot.bat dari *command prompt* sistem Windows.

5. SIMPULAN

Secara umum dapat disimpulkan bahwa adalah mungkin untuk membangun sistem Linux menggunakan inisial disk RAM. Sistem ini menggunakan sebagian memori yang diubah menjadi sistem *root* Linux pada saat kernel dimuat.

Sistem Linux yang dapat dibuat menggunakan inisial disk RAM masih berupa sistem Linux minimal, hal ini disebabkan adanya keterbatasan kernel dalam menciptakan ukuran inisial disk RAM. Untuk sistem Linux yang lebih besar masih perlu pengembangan lebih lanjut.

PUSTAKA

- [1] Almesberger, W. (1996) *Using the initial RAM disk (initrd)*, http://pficheux.free.fr/eyrolles/linux_embarque/docs_externes
- [2] Baughman, L. dan Storozhebykh, S.V. (2002), *Freesco Manual*, <http://www.freesco.net>
- [3] Fawcett, T. (2002) *The Linux Bootdisk HOWTO*, <http://www.tldp.org/Bootdisk-HOWTO/>
- [4] Fischer, C. (2001) *The Loadlin+Win95/98/ME mini-HOWTO*, <http://www.cyxla.com/howtos>
- [5] Gabungan Kelompok Kerja. (2004) *Sistem Operasi*, www.ilmukomputer.com
- [6] Gooch, R. (2002) *Linux Devfs (Device File System) FAQ*, <http://www.atnf.csiro.au/~rgooch>
- [7] Gortmaker, P. (2003) *The Linux BootPrompt-HowTo*, <http://www.tldp.org/HOWTO/BootPrompt-HOWTO.html>
- [8] Kamer, H. (2001) *devfs*, www.vandekamer.com
- [9] Lermen, H. (1996) *LOADLIN User Guide*, <http://elserv ffm.fgan.de/~lermen>
- [10] Rubin, P. (2004) *dd Manual*, Free Software Foundation
- [11] Theodore Ts'o. (2003) *mke2fs Manual*, <http://e2fsprogs.sourceforge.net>
- [12] Tomasez. (2005) *FloopyFW Documentation*, <http://www.zelow.no/floppyfw/>
- [13] _____. (2000) *Ldd, Linux Programmer's Manual*, Manual Linux