BAB 4

IMPLEMENTASI DAN PENGUJIAN

Pada bab ini akan dilakukan implementasi dan pengujian terhadap pembangunan Linux *Terminal Server Project* (LTSP) . Implementasi dilakukan berdasarkan analisis dan perancangan yang telah dibuat. Setelah implementasi selesai, maka selanjutnya akan dilakukan pengujian dan perbandingan kinerja (*benchmark)* pada kinerja dari Linux *Terminal Server Project* (LTSP).

4.1 Instalasi *Server Linux Terminal Server Project* (LTSP)

Untuk memulai instalasi *server* LTSP, langkah pertama yang dilakukan adalah mempersiapkan *repository*, yaitu *bundle* paket-paket program yang diperlukan oleh sistem operasi Linux Ubuntu. Sumber *repository* dapat diperoleh dari koneksi langsung dari internet dengan merujuk kepada *repository* resmi Ubuntu di alamat http://archive.canonical.com/ubuntu atau *mirror-mirror* lokal. Selanjutnya dalam instalasi yang akan dilakukan adalah dengan menggunakan *repository server* lokal dengan alamat http://kambing.ui.edu/repository/ubuntu.

Berikut ini merupakan langkah – langkah dalam proses instalasi LTSP :

1. Install *openssh-server*

```
sudo apt-get install openssh-server
```

2. Install ltsp-*server*-standalone pada *server*

```
sudo apt-get install ltsp-server-standalone
```

3. Konfigurasi dhcp di */etc/ltsp/dhcpd.conf*.

```
sudo nano /etc/ltsp/dhcp.conf
```

Tabel dibawah ini menunjukan konfigurasi pada yang terdapat pada file /etc/ltsp/dhcpd.conf :

Tabel 4.1 Konfigurasi file /etc/ltsp/dhcpd.conf

No	Jenis Konfigurasi	Analisis Konfigurasi
1	Subnet	192.168.1.0
2	Netmask	255.255.255.0
3	Range IP DHCP	192.168.1.2 – 192.168.1.20
4	Root path	/opt/ltsp/i386

Untuk lebih lengkapnya, berikut adalah isi dari konfigurasi file yang terdapat pada file /etc/ltsp/dhcp.conf :

```
#
# Default LTSP dhcpd.conf config file.
#

authoritative;

subnet 192.168.1.0 netmask 255.255.255.0 {
   range 192.168.1.2 192.168.1.20;
   option domain-name "example.com";
   option domain-name-servers 192.168.1.1;
   option broadcast-address 192.168.0.255;
   option routers 192.168.1.1;
#    next-server 192.168.1.1;
#    get-lease-hostnames true;
   option subnet-mask 255.255.255.0;
   option root-path "/opt/ltsp/i386";
   if substring( option vendor-class-identifier, 0, 9 ) = "PXEClient" {
      filename "/ltsp/i386/pxelinux.0";
   } else {
      filename "/ltsp/i386/nbi.img";
   }
}
```

4. Membuat *Environment* untuk *Client* LTSP

   Dalam pembuatan *environment* ini, penulis mengarahkan ke mirror ITB dengan alamat : *ftp://ftp.itb.ac.id*

```
sudo ltsp-build-client --mirror=ftp://ftp.itb.ac.id/pub/ubuntu --security-
mirror=ftp://ftp.itb.ac.id/pub/ubuntu
```

```
sudo ltsp-build-client –dist lucid –arch i386 --
mirror=ftp://ftp.itb.ac.id/pub/ubuntu --security-
mirror=ftp://ftp.itb.ac.id/pub/ubuntu
```

5. Membuat *user client* LTSP

```
sudo adduser client1
```

```
sudo adduser client2
```

```
sudo adduser client3
```

6. Melakukan *update-sshkey*

```
sudo ltsp-update-sshkey
```

7. Melakukan *update-kernels*

```
sudo ltsp-update-kernels
```

8. Melakukan *update-image*

```
sudo ltsp-update-image
```

9. Memulai *service* DHPC

```
sudo service dhcp3 start
```

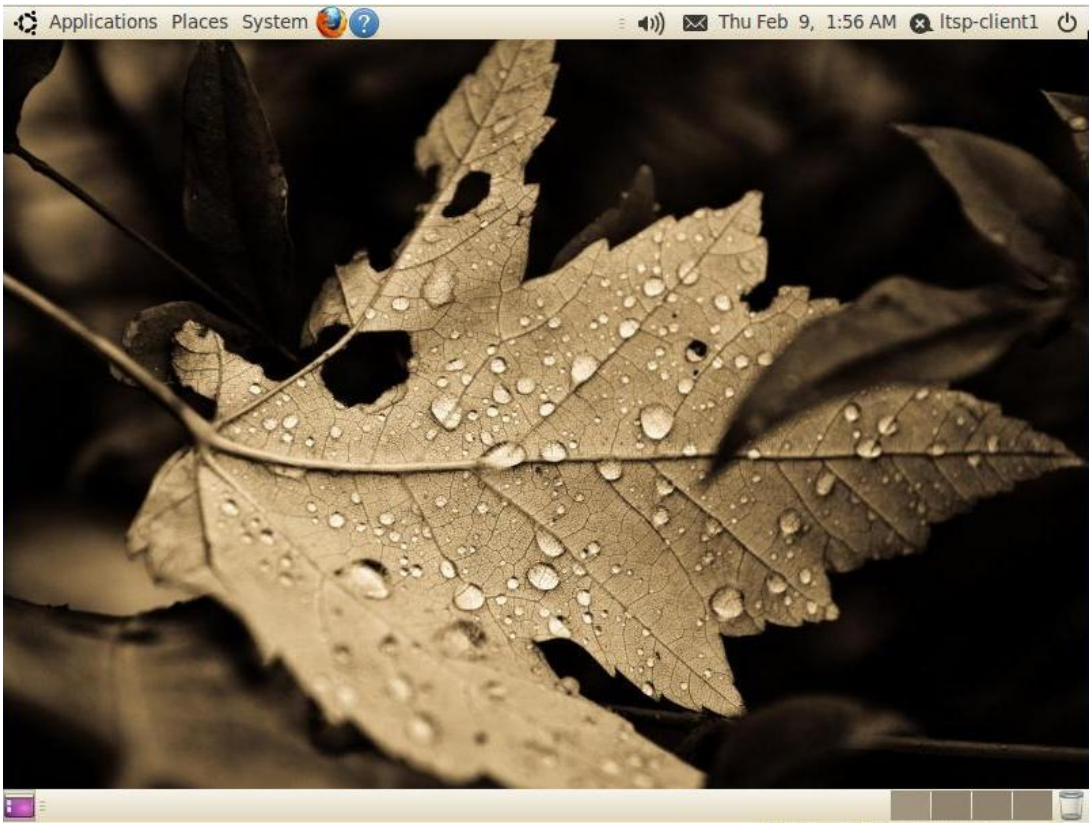
4.2 Tampilan Login dan Desktop *Linux Terminal Server Project (LTSP)*

Setelah proses instalasi LTSP berhasil dilakukan, maka *Terminal Server Project (LTSP)* yang dibangun akan dapat digunakan. Tampilan antarmuka (*interface)* dari halaman *login* dan tampilan *desktop Linux Terminal Server Project (LTSP)* dapat dilihat pada gambar-gambar dibawah ini :



Gambar 4.1 Halaman login LTSP

Pada halaman ini *user* memasukan *username* dan *password*,jika berhasil maka user akan mendapat sesi pada *server*. Tampilan *Desktop user* dapat dilihat pada gambar berikut :



Gambar 4.2 Desktop LTSP

4.3 Implementasi Optimalisasi server LTSP

Sebelum melakukan pengujian pada tiga buah *client*, terlebih dahulu dilakukan opimalisasi pada server LTSP. Optimalisasi ini dilakukan dengan cara :

1. Instalasi Kernel Versi 3.0.1

   Dalam implementasi Server LTSP ini, akan digunakan kernel versi terbaru yaitu versi 3.0.1. Dalam kernel ini terdapat perubahan fitur yang yang lebih mendukung untuk *hardware* dan *software.* Kernel versi

3.0.1 merupakan versi kernel pengembangan dari kernel versi 2.6.x.x yang memiliki kinerja yang stabil dan memiliki banyak sekali tambahan *library,plugins,*dan perbaikan *bug.*

Instalasi kernel 3.0.1 ini akan dilakukan dalam modus *Terminal* dengan menggunakan hak akses *root.* Berikut adalah langkah – langkah instalasi kernel 3.0.1 :

1. Buka *Terminal* pada sistem operasi Ubuntu
2. Download *linux-headers, linux-headers-generic, linux-headers-image* dengan menggunkan *wget* dan simpan pada *home* direktori

   ```
   cd /home/ltsp
   ```

   ```
   wget http://kernel.ubuntu.com/%7Ekernel-ppa/mainline/v3.0.1-oneiric/linux-headers-3.0.1-030001_3.0.1-030001.201108060905_all.deb
   ```

   ```
   wget http://kernel.ubuntu.com/%7Ekernel-ppa/mainline/v3.0.1-oneiric/linux-headers-3.0.1-030001-generic_3.0.1-030001.201108060905_i386.deb
   ```

   ```
   wget http://kernel.ubuntu.com/%7Ekernel-ppa/mainline/v3.0.1-oneiric/linux-image-3.0.1-030001-generic_3.0.1-030001.201108060905_i386.deb
   ```

3. Install semua bagian yang telah didownload sebelumnya dengan menggunakan perintah *sudo dpkg –i* dan install terlebih dahulu *linux-headers,* kemudian *linux-headers-generic* dan terakhir *linux-headers-image.*

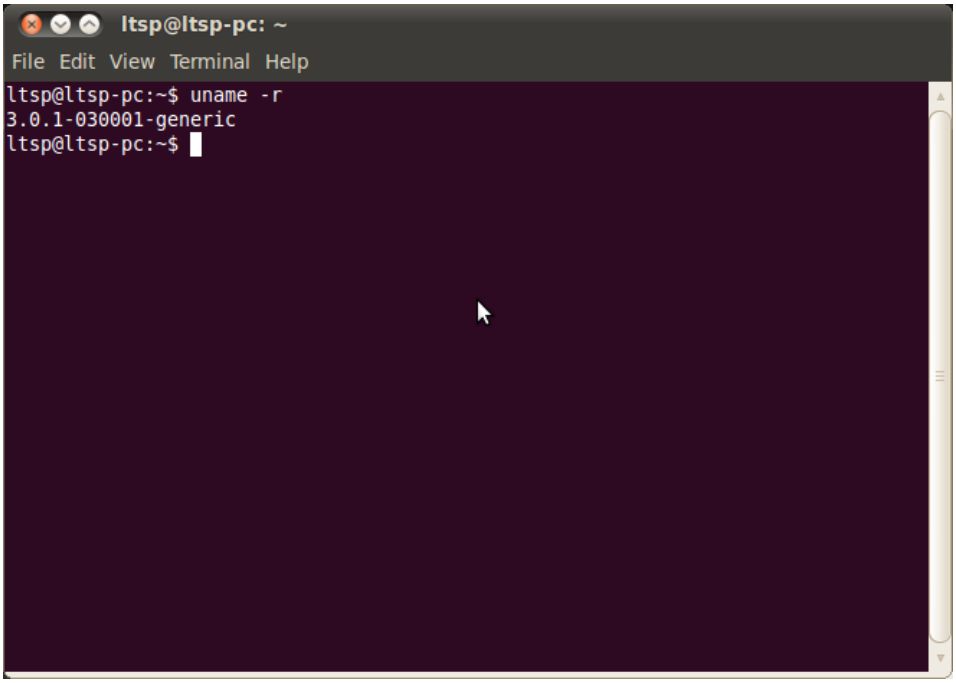
```
sudo dpkg -i linux-headers-3.0.1-030001_3.0.1-030001.201108060905_all.deb
```

```
sudo dpkg -i linux-headers-3.0.1-030001-generic_3.0.1-030001.201108060905_i386.deb
```

```
sudo dpkg -i linux-image-3.0.1-030001-generic_3.0.1-030001.201108060905_i386.deb
```

4. Restart komputer
5. Untuk pengecekan versi kernel dapat dilakukan dengan menggunakan perintah :

```
uname -r
```



Gambar 4.3 Pengecekan versi *kernel* linux

2. Pemasangan *localapps*

   Seperti yang telah diterangkan pada Bab III halaman 67, pemasangan *localapps* ini dilakuakan agar kinerja dari server LTSP tidak terlalu terganggu terutama untuk aplikasi yang membutuhkan *resource* yang besar. Implementasi localapps dalam server LTSP ini di dilakukan pada aplikasi Firefox dengan mengetikkan :

```
sudo chroot /opt/ltsp/i386 apt-get install firefox*
```

   Kemudian dilakukan kembali proses *update-image*

```
sudo ltsp-update-image
```

   Setelah itu menambahkan baris berikut pada /var/lib/tftpboot/ltsp/i386/lts.conf :

```
[default]
LOCAL_APPS=True
LOCAL_APPS_MENU=True
LOCAL_APPS_MENU_ITEMS=firefox
```

4.4 Pengujian kerja LTSP

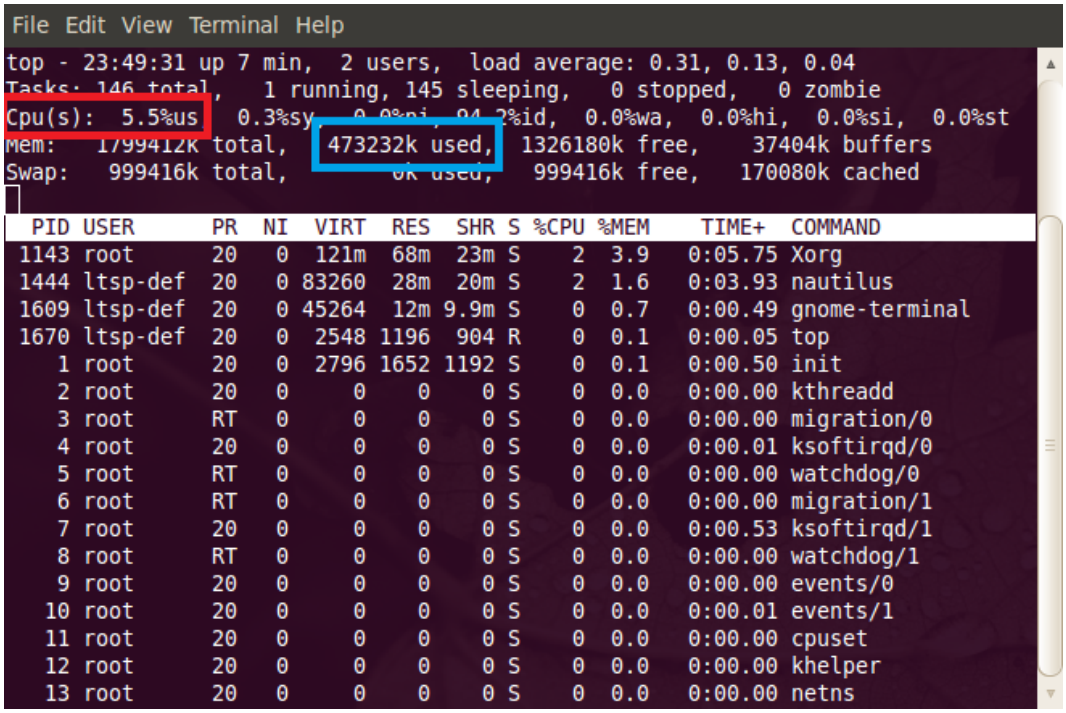
Skenario pengujian yang akan dilakukan adalah dengan melihat perbandingan kerja prosessor dan memori terhadap :

1. Sistem LTSP *default,* merupakan server LTSP yang dibangun dalam kondisi tanpa dilakukan perubahan dalam sistem kerjanya.

2. Sistem LTSP *default* yang dioptimasi dengan penggunaan *local application*.
3. Sistem LTSP yang telah dioptimalisasi, merupakan sistem yang telah dirancang seperti pada Bab III halaman 67 dan diimplementasikan pada Bab IV ini di halaman 73.

Pengujian Linux Terminal Server Project (LTSP) dilakukan dengan menggunakan program top. Program ini dijalankan pada sisi *server* untuk melihat penggunaan prosessor dan memori. Ukuran Untuk menjalankan program ini cukup dengan diaktifkan pada terminal c*onsole server* dengan mengetik top.

```
top
```



Gambar 4.4 Tampilan aplikasi *top*.

Seperti yang terlihat pada gambar 4.4, penggunaan CPU (kotak merah) memiliki ukuran dalam persen (%) dan penggunaan memori (kotak biru) dalam *KiloBytes* (KB).

Client yang digunakan dalam tahap pengujian ini adalah tiga buah CPU Pentium III, dengan *processor* 800MHZ dan memori sebesar 256MB.

Tahapan pengujian dilakukan dengan cara menyalakan *client* satu persatu sampai tiga *client* semuanya beroperasi dan diberikan tiga buah beban aplikasi, yaitu Openoffice, Gimp dan Firefox web browser. Batasan ini dilakukan karena aplikasi-aplikasi tersebut adalah aplikasi yang dirancang untuk digunakan oleh *client* dan selain itu juga menuntut penggunaan sumberdaya yang cukup besar,sehingga layak untuk dijadikan aplikasi untuk pengujian *Linux terminal server project* (LTSP).

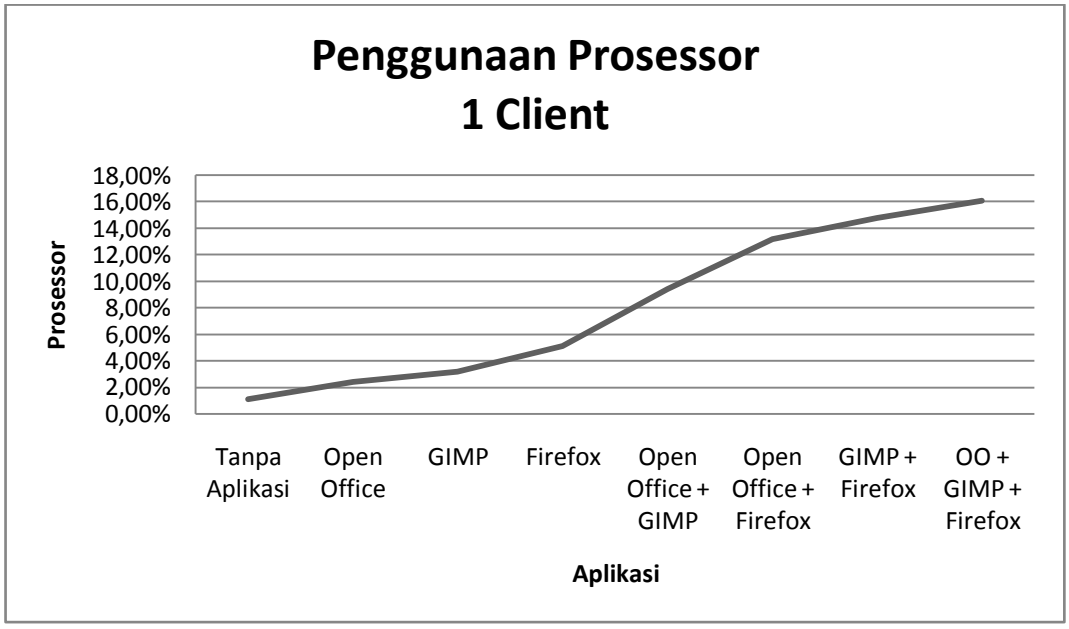
4.4.1 Pengujian LTSP kondisi *default*

Tabel dibawah menunjukan penggunaan prossesor dan memori pada keadaan satu sampai tiga *client* aktif :

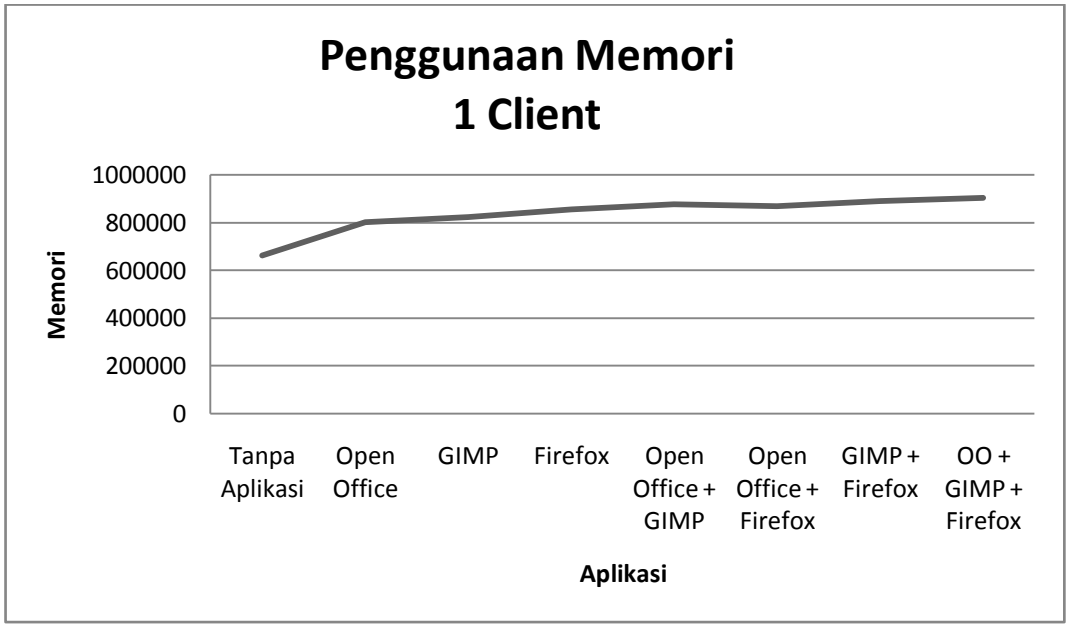
Tabel 4.2 Tabel uji pada satu *client* LTSP *default*.

No	Beban Aplikasi	Penggunaan Prosessor	Penggunaan Memori (KB)
1	Tanpa Aplikasi	1,1%	662588
2	Open Office	2,4%	802436
3	GIMP	3,2%	823716
4	Firefox	5,1%	853948
5	Open Office + GIMP	9,4%	876516
6	Open Office + Firefox	13,2%	869072
7	GIMP + Firefox	14,8%	890036
8	OO + GIMP + Firefox	16,1%	903352

Tabel diatas memperlihatkan hasil *benchmark* LTSP dengan satu *client* aktif mulai dari keadaan tidak menjalankan aplikasi sampai dengan menjalankan aplikasi Open Office,GIMP,dan firefox. Seperti yang terlihat pada tabel penggunan memory dan prossesor cukup stabil pada penggunaan satu *client*. Selanjutnya data yang didapatkan diubah dalam bentuk grafik yang dapat dilihat pada gambar berikut :



Gambar 4.5 Grafik penggunaan prosessor pada satu *client.*



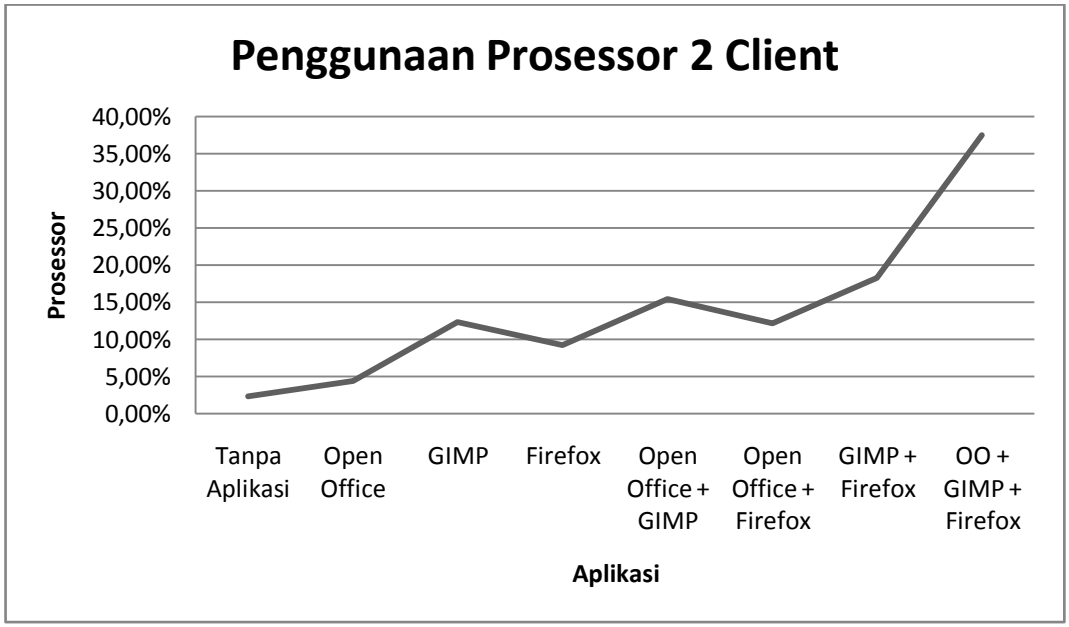
Gambar 4.6 Grafik penggunaan memori pada satu *client.*

Selanjutnya dilakukan dengan dua *client* aktif dengan skenario yang sama seperti sebelumnya. Data pengujian dapat dilihat pada tabel berikut :

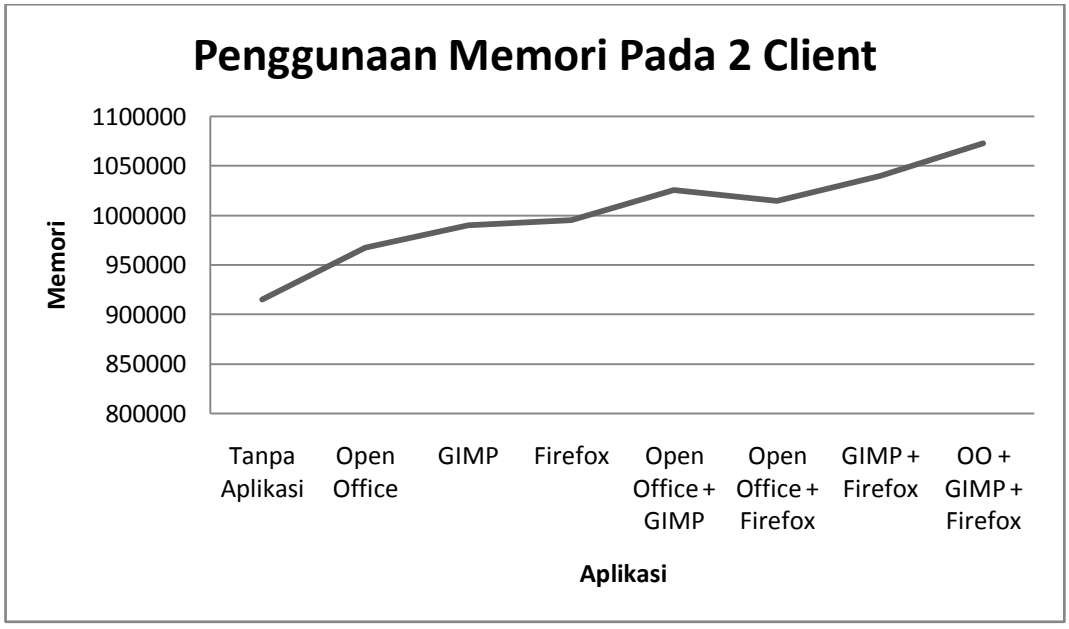
Tabel 4.3 Tabel uji pada dua *client* LTSP *default*.

No	Beban Aplikasi	Penggunaan Prosessor	Penggunaan Memori (KB)
1	Tanpa Aplikasi	2,3%	914724
2	Open Office	4,4%	967080
3	GIMP	12,3%	990184
4	Firefox	9,2%	995284
5	Open Office + GIMP	15,4%	1025384
6	Open Office + Firefox	12,1%	1014796
7	GIMP + Firefox	18,3%	1039636
8	OO + GIMP + Firefox	37,5%	1072904

Selanjutnya data yang didapatkan diubah dalam bentuk grafik yang dapat dilihat pada gambar berikut :



Gambar 4.7 Grafik penggunaan prosessor pada dua *client.*



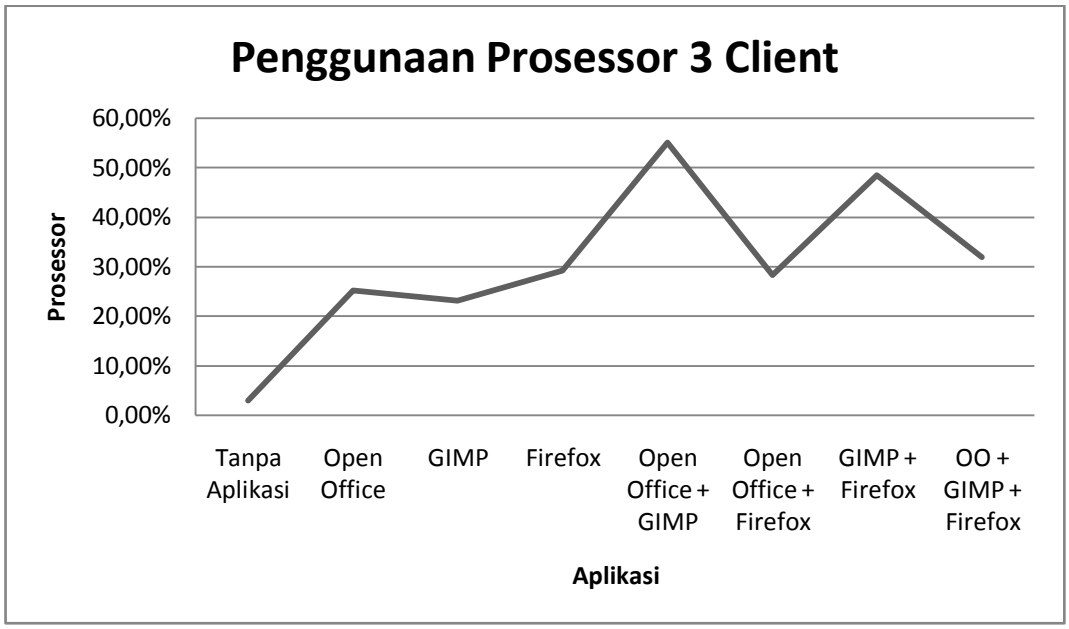
Gambar 4.8 Grafik penggunaan memori pada dua *client.*

Dan yang terakhir adalah melakukan pengujian dengan tiga *client* aktif dengan skenario yang sama seperti sebelumnya. Data pengujian dapat dilihat pada tabel berikut :

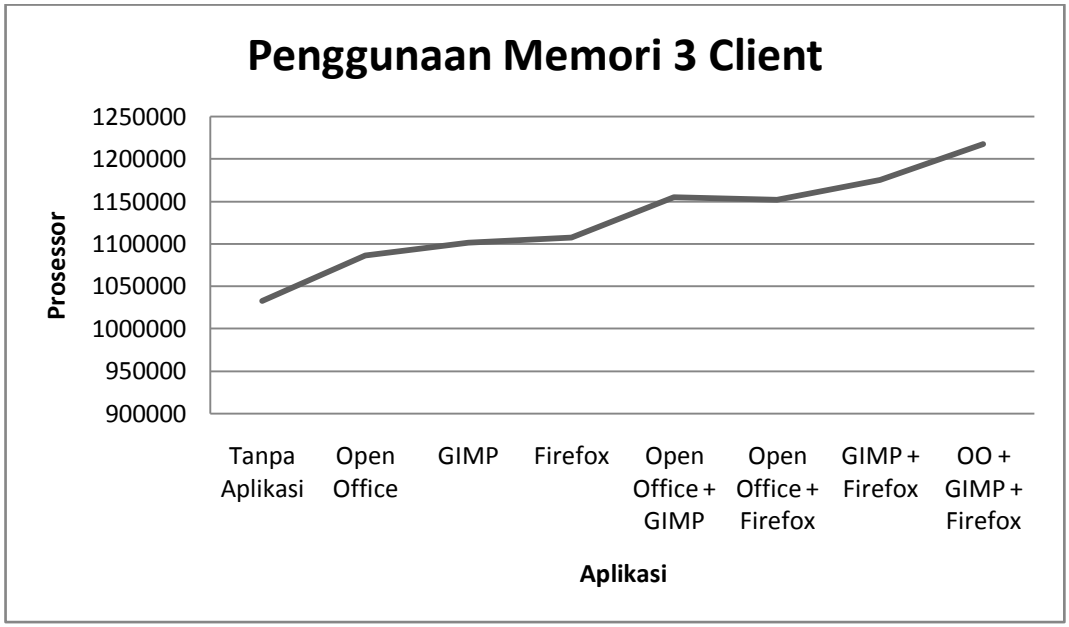
Tabel 4.4 Tabel uji pada tiga *client* LTSP *default*.

No	Beban Aplikasi	Penggunaan Prosessor	Penggunaan Memori (KB)
1	Tanpa Aplikasi	2,9%	1032432
2	Open Office	25,2%	1085940
3	GIMP	23,1%	1100924
4	Firefox	29,1%	1106816
5	Open Office + GIMP	55,0%	1154856
6	Open Office + Firefox	28,3%	1151352
7	GIMP + Firefox	48,4%	1174632
8	OO + GIMP + Firefox	31,9%	1217048

Data yang penggunaan prossesor dan memory yang didapatkan diubah dalam bentuk grafik yang dapat dilihat pada gambar berikut :



Gambar 4.9 Grafik penggunaan prosessor pada tiga *client.*



Gambar 4.10 Grafik penggunaan memori pada tiga *client.*

Dari hasil pengujian LTSP dengan sistem default yaitu dengan versi 10.04 dapat dilihat dengan jelas bahwa penggunaan memori masih cukup tinggi. Hal ini berkaitan dengan versi kernel standar dan GCC yang dimiliki oleh distribusi Linux Ubuntu versi 10.04. Kekurangan dari kernel 2.6.32-28-Generic yang dimiliki oleh Linux distribusi Ubuntu 10.04 adalah kurangnya *library* untuk proses NFS dan *upgrade-upgrade* lainnya yang dapat meningkatkan kinerja LTSP.

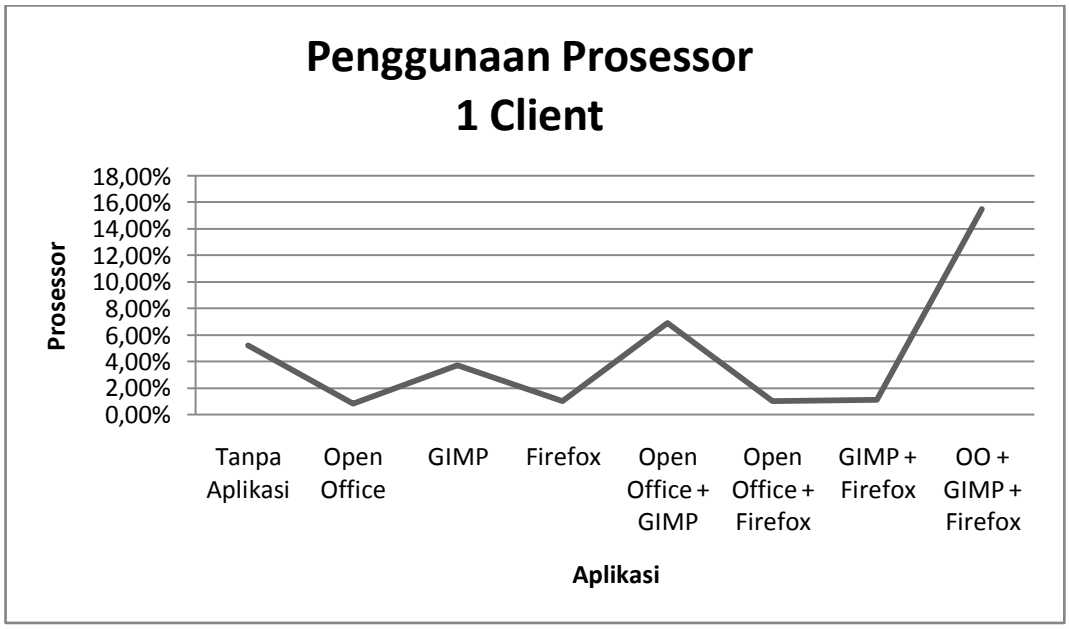
4.4.2 Pengujian LTSP *default + local application*

Berikut adalah pengujian pada LTSP *default* yang dioptimasi dengan penggunaan *local application.* Tabel dibawah menunjukan penggunaan prossesor dan memori pada keadaan satu sampai tiga *client* aktif :

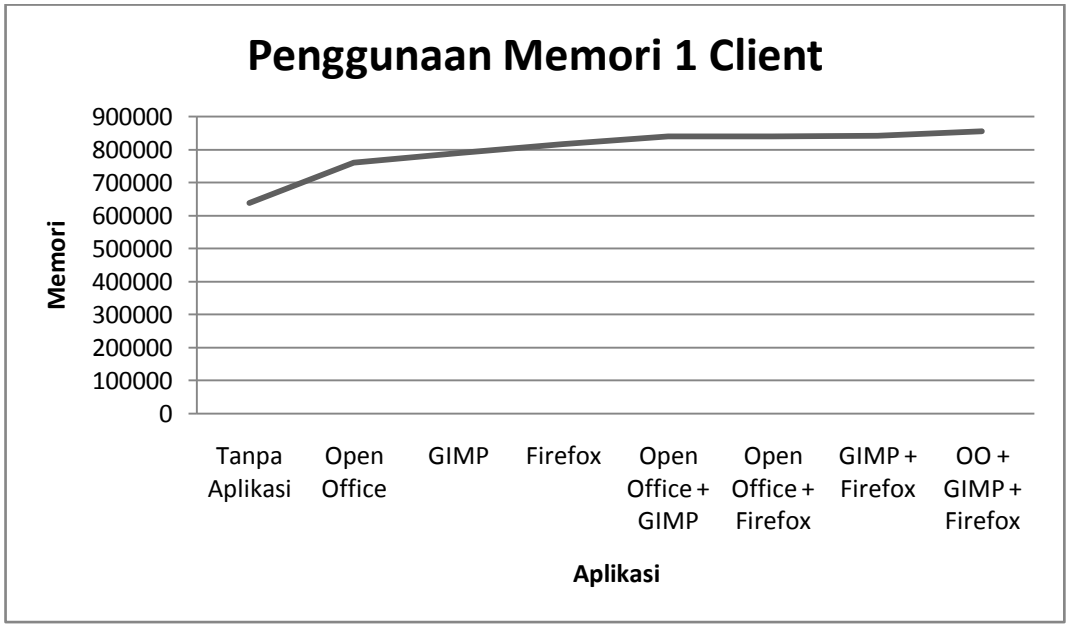
Tabel 4.5 Tabel uji pada satu *client* LTSP.

No	Beban Aplikasi	Penggunaan Prosessor	Penggunaan Memori (KB)
1	Tanpa Aplikasi	5,2%	636848
2	Open Office	0,8%	759880
3	GIMP	3,7%	787756
4	Firefox	1,0%	814736
5	Open Office + GIMP	6,9%	839344
6	Open Office + Firefox	1,0%	838772
7	GIMP + Firefox	1,1%	840252
8	OO + GIMP + Firefox	15,5%	854260

Data yang penggunaan prossesor dan memory yang didapatkan diubah dalam bentuk grafik yang dapat dilihat pada gambar berikut :



Gambar 4.11 Grafik penggunaan prosessor pada satu *client.*



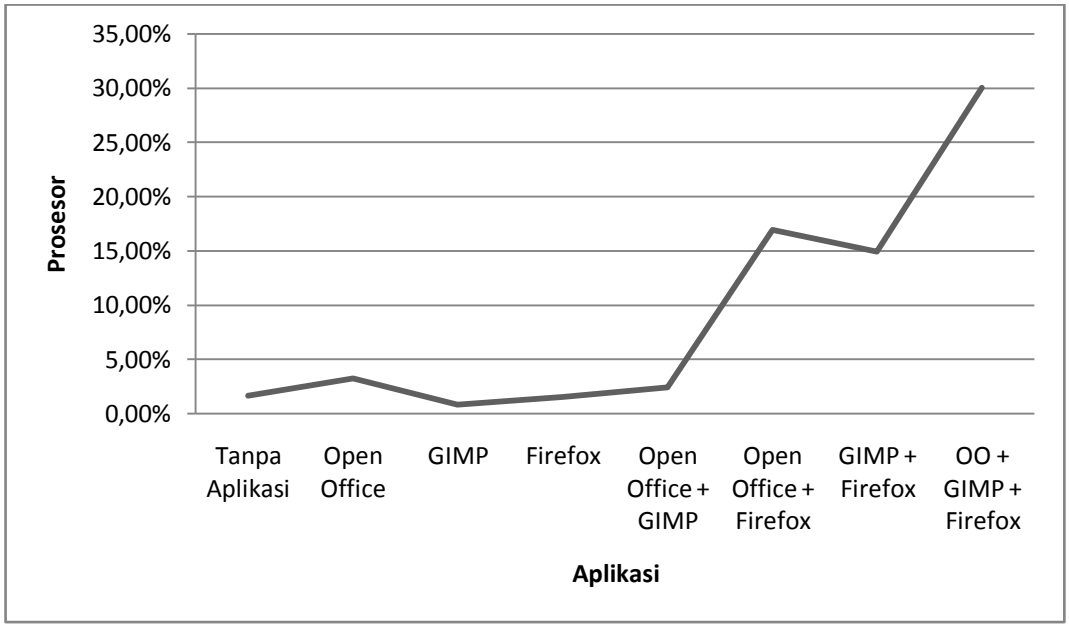
Gambar 4.12 Grafik penggunaan memori pada satu *client.*

Selanjutnya dilakukan dengan dua *client* aktif dengan skenario yang sama seperti sebelumnya. Data pengujian dapat dilihat pada tabel berikut :

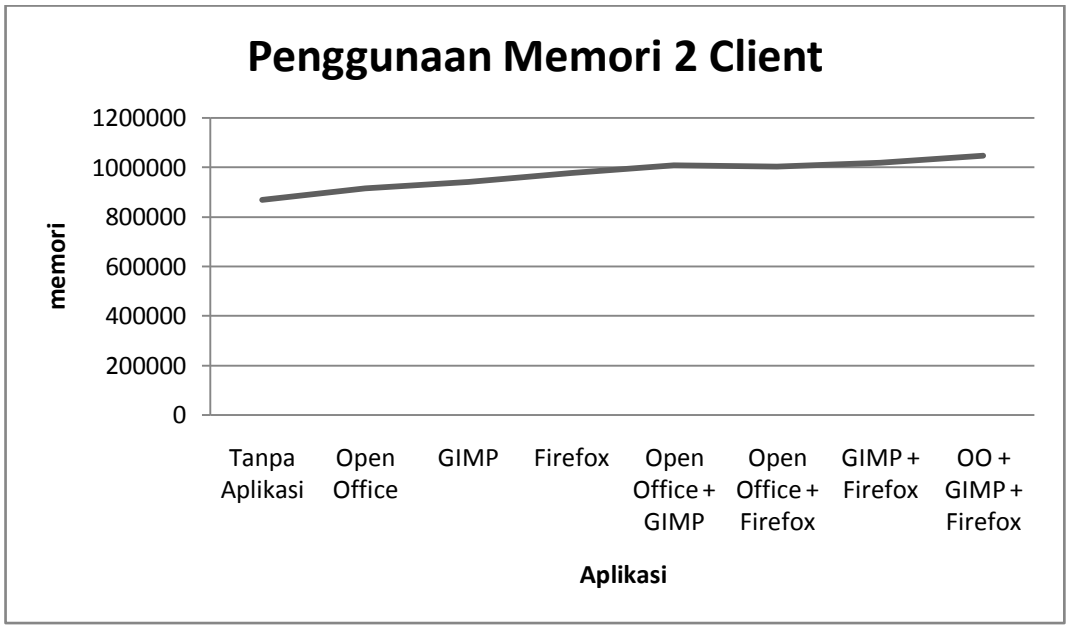
Tabel 4.6 Tabel uji pada dua *client* LTSP.

No	Beban Aplikasi	Penggunaan Prosessor	Penggunaan Memori (KB)
1	Tanpa Aplikasi	1,6%	867280
2	Open Office	3,2%	915080
3	GIMP	0,8%	940336
4	Firefox	1,5%	977608
5	Open Office + GIMP	2,4%	1006872
6	Open Office + Firefox	16,9%	1004144
7	GIMP + Firefox	14,9%	1018604
8	OO + GIMP + Firefox	30,0%	1047304

Data yang penggunaan prossesor dan memori yang didapatkan diubah dalam bentuk grafik yang dapat dilihat pada gambar berikut :



Gambar 4.13 Grafik penggunaan prosessor pada dua *client.*



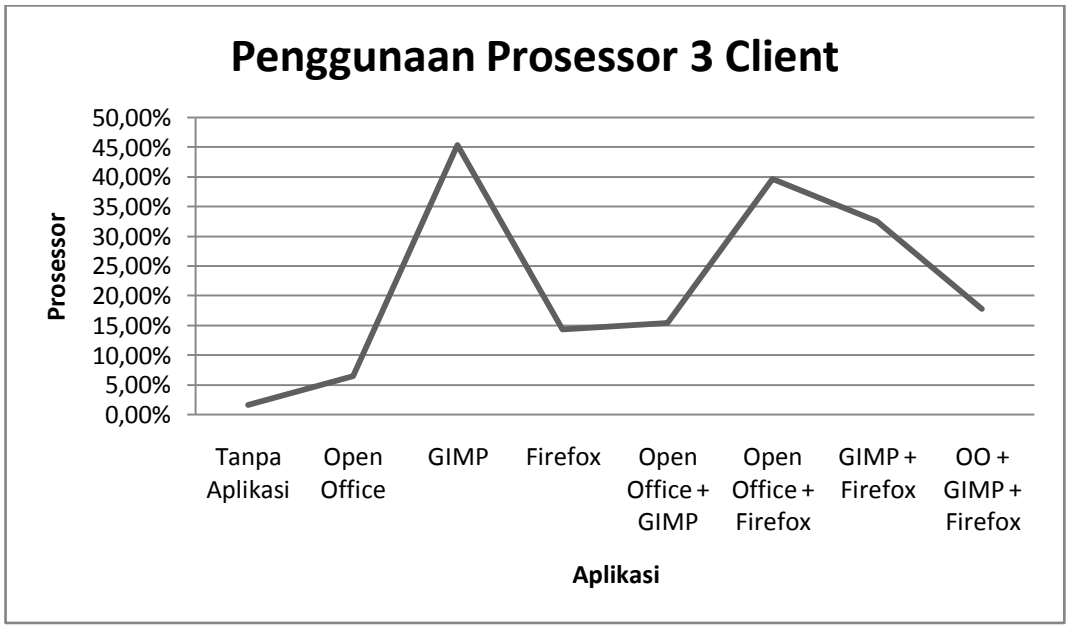
Gambar 4.14 Grafik penggunaan memori pada dua *client.*

Dan yang terakhir adalah melakukan pengujian dengan tiga *client* aktif dengan skenario yang sama seperti sebelumnya. Data pengujian dapat dilihat pada tabel berikut :

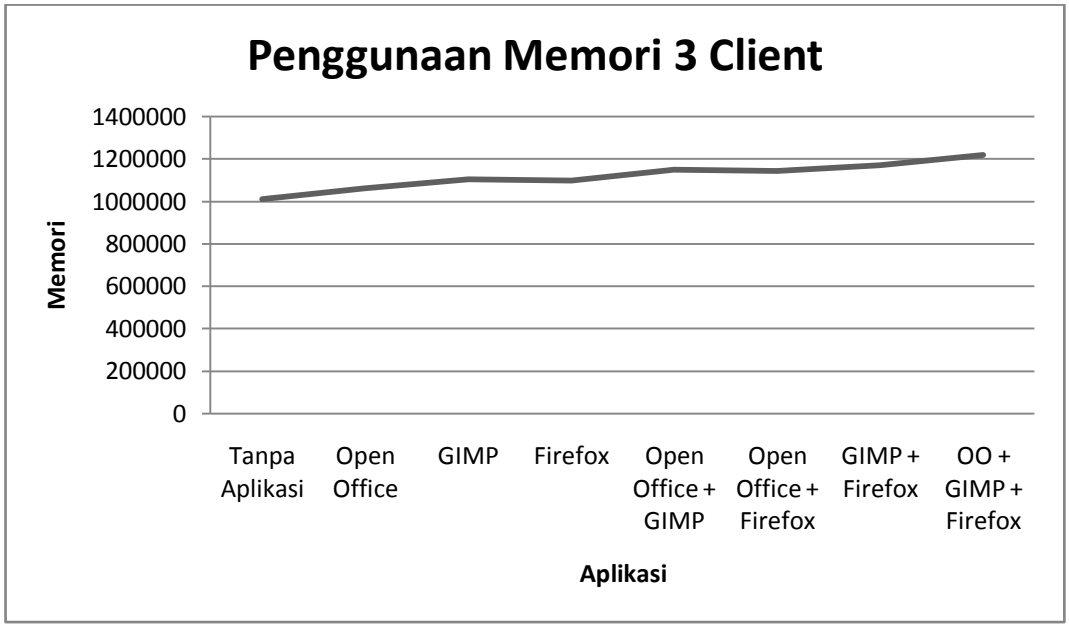
Tabel 4.7 Tabel uji pada tiga *client* LTSP.

No	Beban Aplikasi	Penggunaan Prosessor	Penggunaan Memori (KB)
1	Tanpa Aplikasi	1,6%	1009312
2	Open Office	6,4%	1060840
3	GIMP	45,3%	1102444
4	Firefox	14,3%	1097292
5	Open Office + GIMP	15,4%	1150736
6	Open Office + Firefox	39,6%	1144048
7	GIMP + Firefox	32,5%	1169332
8	OO + GIMP + Firefox	17,7%	1219200

Data yang penggunaan prossesor dan memori yang didapatkan diubah dalam bentuk grafik yang dapat dilihat pada gambar berikut :



Gambar 4.15 Grafik penggunaan prosessor pada tiga *client.*



Gambar 4.16 Grafik penggunaan memori pada tiga *client.*

Dari hasil pengujian LTSP dengan sistem default yang dioptimasi dengan pengunaan *localapps* yaitu terlihat penggunaan memori lebih baik dari kondisi *default,* hal ini dikarenakan penggunaan dari *localapps* yang memungkinkan suatu aplikasi (dalam hal ini Firefox) dijalankan secara local oleh *client.*

4.4.3 Pengujian LTSP yang telah dilakukan optimalisasi

Berikutnya akan dilakukan pengujian pada sistem LTSP yang telah mengalami optimalisasi menggunakan kernel 3.0.1 maupun pada *environment* chroot Distribusi Ubuntu 10.04.

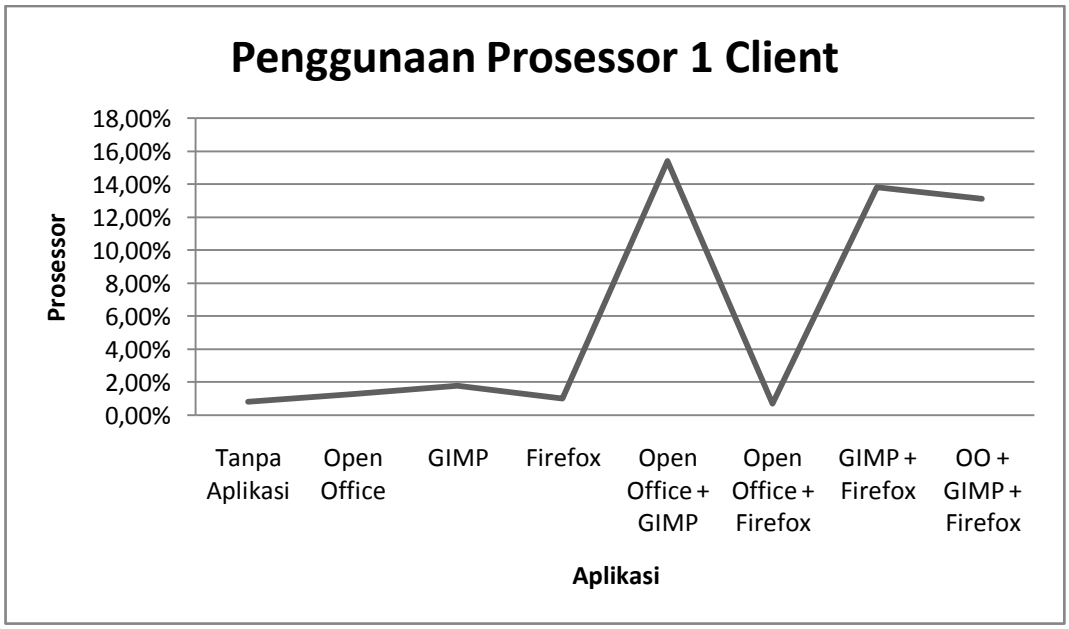
Selanjutnya akan dilakukan pengujian untuk mendapatkan data penggunaan prosessor dan memori pada LTSP yang telah dilakukan optimalisasi.

Tabel dibawah menunjukan penggunaan prossesor dan memori pada keadaan satu sampai tiga *client* aktif :

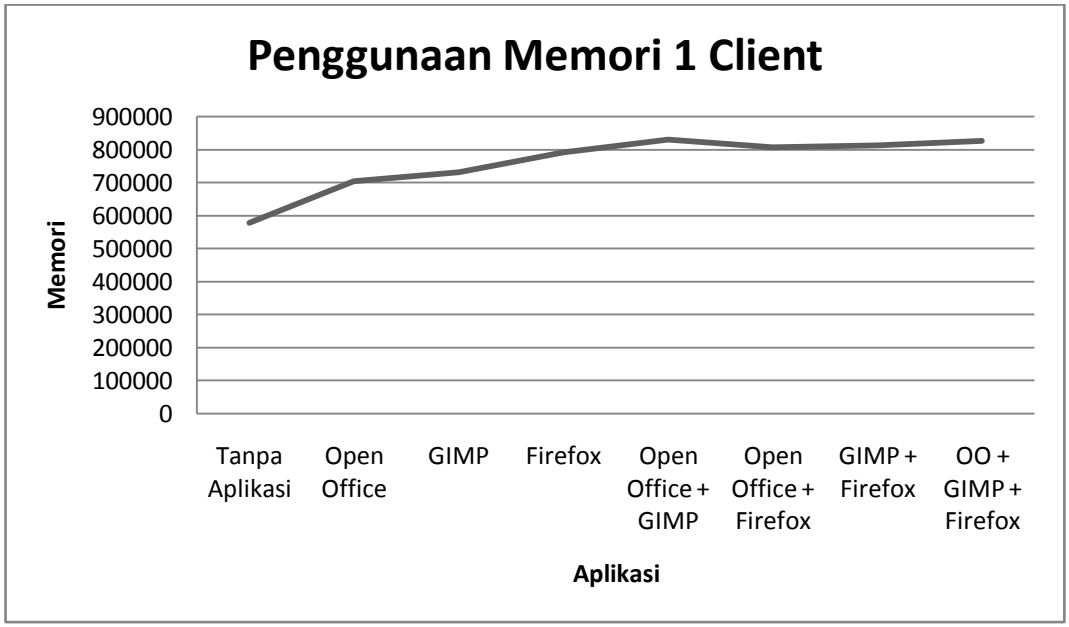
Tabel 4.8 Tabel uji pada satu *client* LTSP *optimasi*.

No	Beban Aplikasi	Penggunaan Prosessor	Penggunaan Memori (KB)
1	Tanpa Aplikasi	0,8%	577788
2	Open Office	1,3%	704264
3	GIMP	1,8%	729736
4	Firefox	1,0%	789600
5	Open Office + GIMP	15,4%	828996
6	Open Office + Firefox	0,7%	805496
7	GIMP + Firefox	13,8%	811224
8	OO + GIMP + Firefox	13,1%	825500

Data yang penggunaan prossesor dan memory yang didapatkan diubah dalam bentuk grafik yang dapat dilihat pada gambar berikut :



Gambar 4.17 Grafik penggunaan prosessor pada satu *client.*



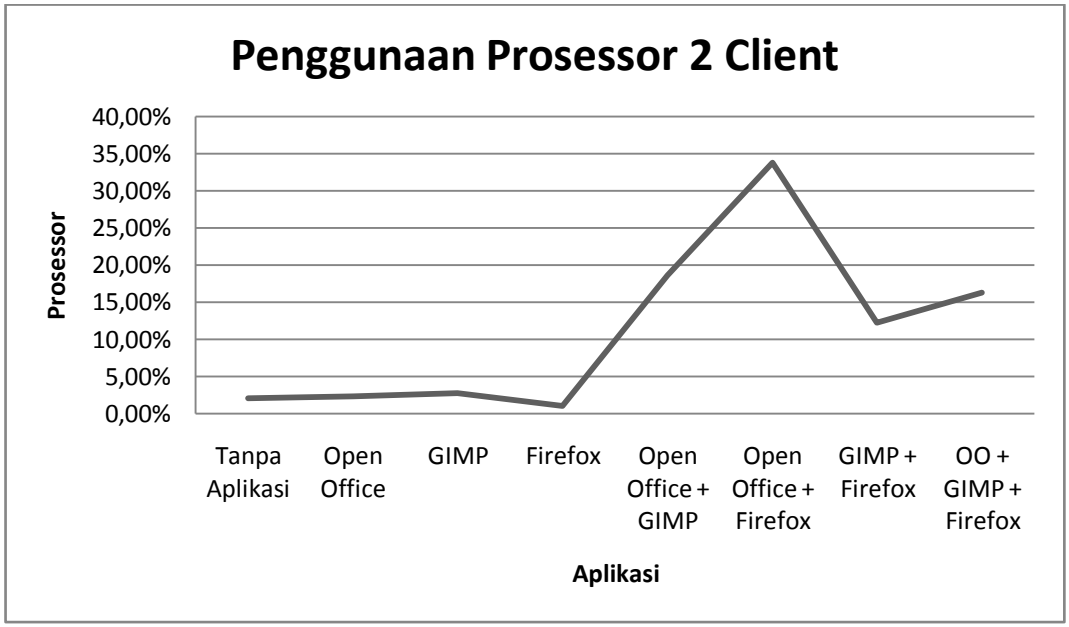
Gambar 4.12 Grafik penggunaan memori pada satu *client.*

Selanjutnya dilakukan dengan dua *client* aktif dengan skenario yang sama seperti sebelumnya. Data pengujian dapat dilihat pada tabel berikut :

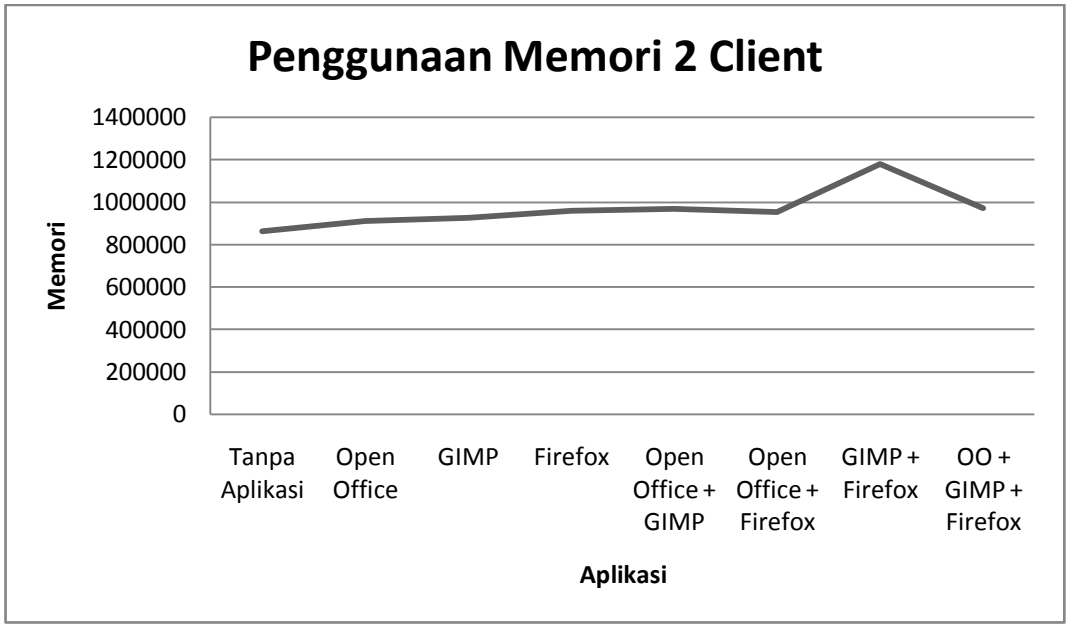
Tabel 4.9 Tabel uji pada dua *client* LTSP *optimasi*.

No	Beban Aplikasi	Penggunaan Prosessor	Penggunaan Memori (KB)
1	Tanpa Aplikasi	2,0%	862188
2	Open Office	2,3%	909580
3	GIMP	2,7%	925437
4	Firefox	1,0%	958467
5	Open Office + GIMP	18,6%	968380
6	Open Office + Firefox	33,8%	953136
7	GIMP + Firefox	12,20%	1179208
8	OO + GIMP + Firefox	16,3%	970612

Data yang penggunaan prossesor dan memory yang didapatkan diubah dalam bentuk grafik yang dapat dilihat pada gambar berikut :



Gambar 4.18 Grafik penggunaan prosessor pada dua *client.*



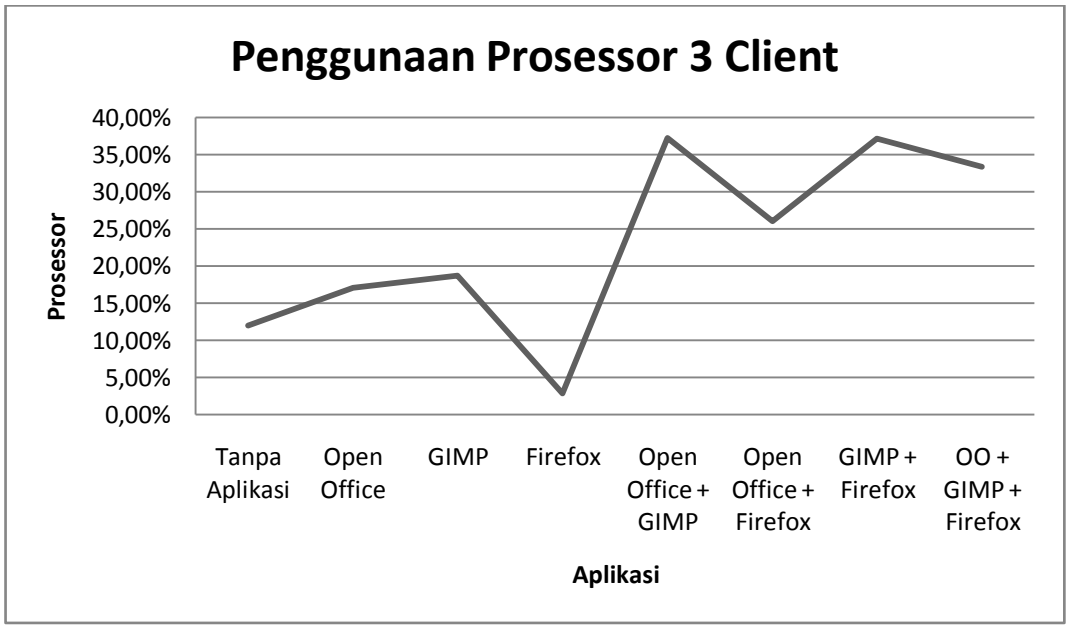
Gambar 4.19 Grafik penggunaan memori pada dua *client.*

Dan yang terakhir adalah melakukan pengujian dengan tiga *client* aktif dengan skenario yang sama seperti sebelumnya. Data pengujian dapat dilihat pada tabel berikut :

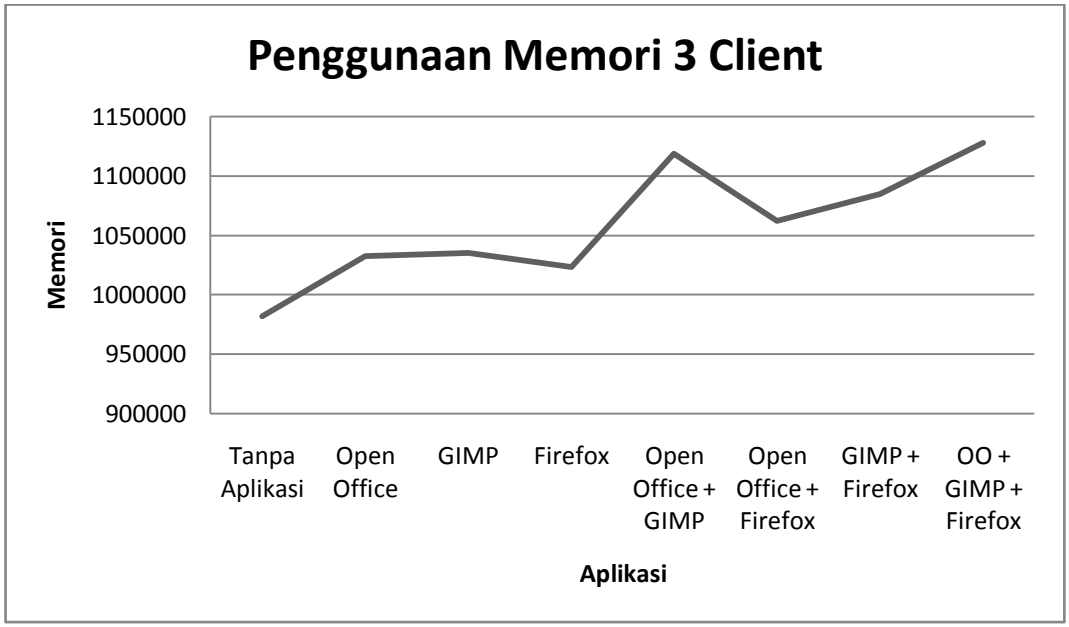
Tabel 4.10 Tabel uji pada tiga *client* LTSP .

No	Beban Aplikasi	Penggunaan Prosessor	Penggunaan Memori (KB)
1	Tanpa Aplikasi	11,9%	981908
2	Open Office	17,0%	1032356
3	GIMP	18,7%	1035478
4	Firefox	2,8%	1023540
5	Open Office + GIMP	37,2%	1119004
6	Open Office + Firefox	26,0%	1062440
7	GIMP + Firefox	37,1%	1084596
8	OO + GIMP + Firefox	33,3%	1127956

Data yang penggunaan prossesor dan memory yang didapatkan diubah dalam bentuk grafik yang dapat dilihat pada gambar berikut :



Gambar 4.20 Grafik penggunaan prosessor pada tiga *client* LTSP



Gambar 4.21 Grafik penggunaan memori pada tiga *client* LTSP

4.5 Perbandingan kinerja LTSP

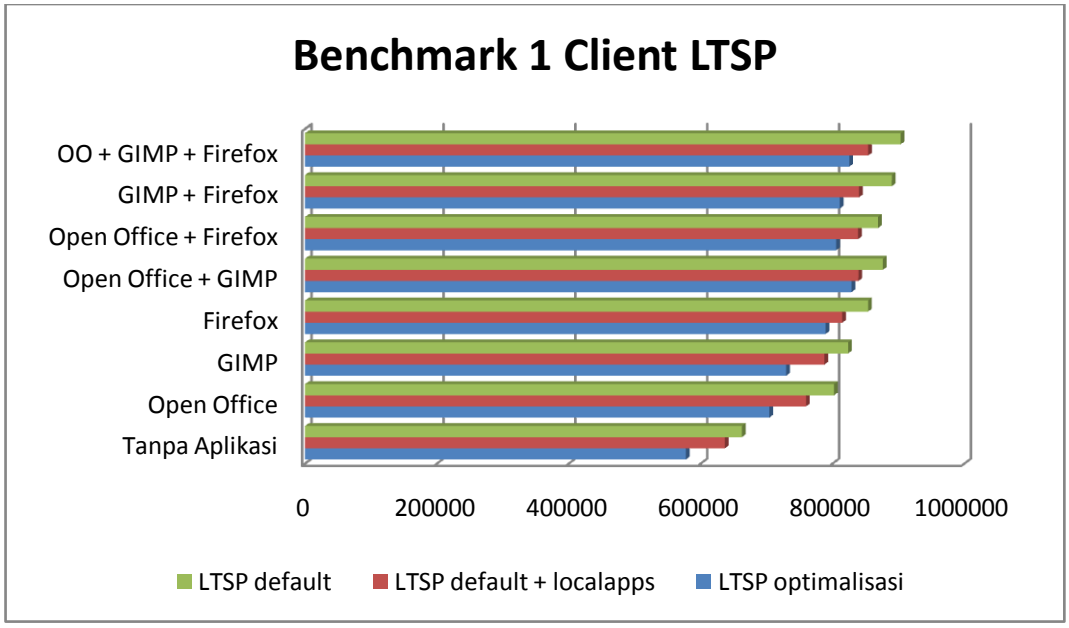
Dari hasil pengujian akan dilakukan perbandingan kerja antara sistem LTSP *default* dan yang telah dilakukan optimalisasi. Berikut adalah grafik perbandingannya :

1. Kondisi perbandingan 1 client LTSP

Tabel 4.11 Tabel *Benchmark* pada satu *client* LTSP.

No	Beban Aplikasi	LTSP default (KB)	LTSP default + *localapps* (KB)	LTSP optimalisasi (KB)
1	Tanpa Aplikasi	662588	636848	577788
2	Open Office	802436	759880	704264
3	GIMP	823716	787756	729736
4	Firefox	853948	814736	789600
5	Open Office + GIMP	876516	839344	828996

6	Open Office + Firefox	869072	838772	805496
7	GIMP + Firefox	890036	840252	811224
8	OO + GIMP + Firefox	903352	854260	825500



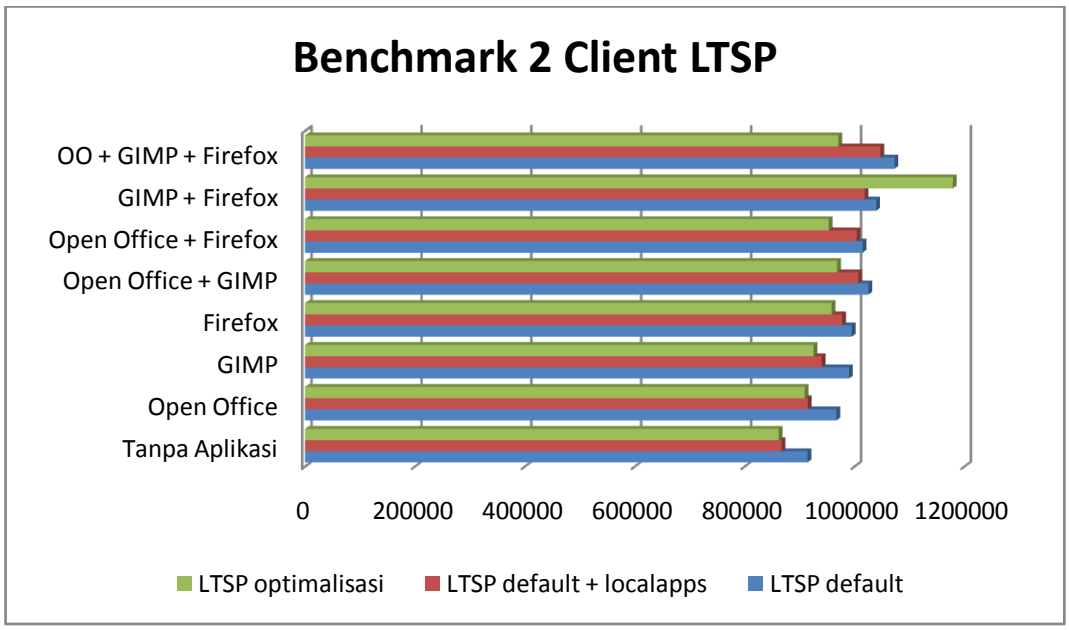
Gambar 4.22 Grafik perbandingan memori pada satu *client* LTSP.

2. Kondisi perbandingan 2 client LTSP

Tabel 4.12 Tabel *Benchmark* pada dua *client* LTSP.

No	Beban Aplikasi	LTSP default (KB)	LTSP default + localapps (KB)	LTSP optimalisasi (KB)
1	Tanpa Aplikasi	914724	867280	862188
2	Open Office	967080	915080	909580
3	GIMP	990184	940336	925437
4	Firefox	995284	977608	958467
5	Open Office + GIMP	1025384	1006872	968380
6	Open Office + Firefox	1014796	1004144	953136

7	GIMP + Firefox	1039636	1018604	1179208
8	OO + GIMP + Firefox	1072904	1047304	970612



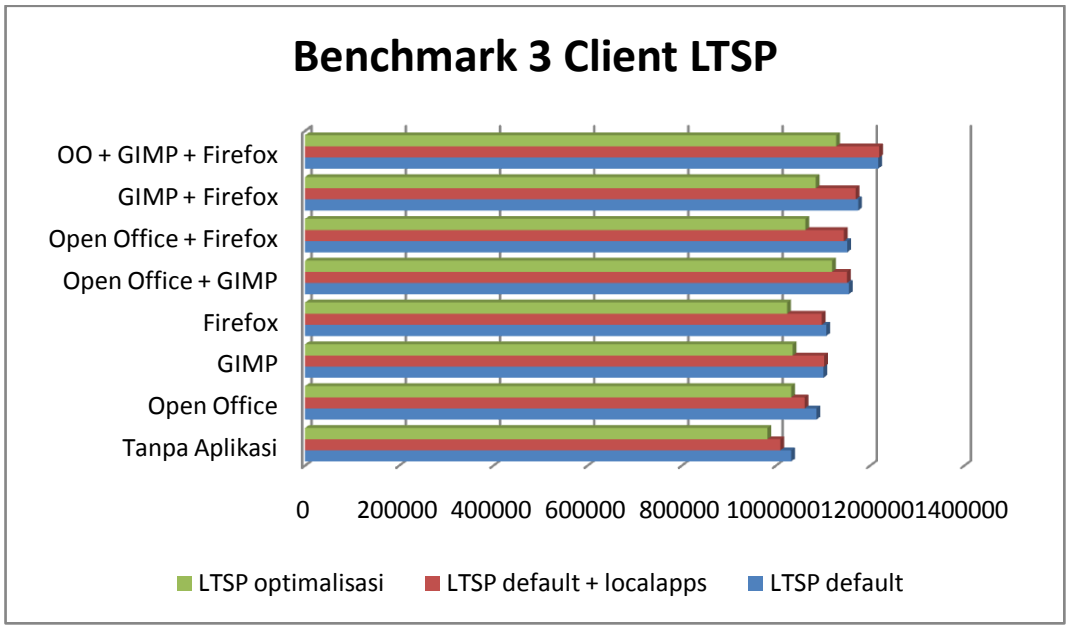
Gambar 4.23 Grafik perbandingan memori pada dua *client* LTSP.

3. Kondisi perbandingan 3 client LTSP

Tabel

No	Beban Aplikasi	LTSP default (KB)	LTSP default + localapps (KB)	LTSP optimalisasi (KB)
1	Tanpa Aplikasi	1032432	1009312	981908
2	Open Office	1085940	1060840	1032356
3	GIMP	1100924	1102444	1035478
4	Firefox	1106816	1097292	1023540
5	Open Office + GIMP	1154856	1150736	1119004
6	Open Office + Firefox	1151352	1144048	1062440
7	GIMP + Firefox	1174632	1169332	1084596
8	OO + GIMP + Firefox	1217048	1219200	1127956

4.13 Tabel *Benchmark* pada dua *client* LTSP.



Gambar 4.24 Grafik perbandingan memori pada dua *client* LTSP.

Dari hasil data pengujian dan perbandingan grafik yang didapatkan, dapat dilihat bahwa modifikasi LTSP menggunakan kernel 3.01 dan modifikasi pada chroot memiliki efesiensi dan kerja yang lebih baik dibandingkan dengan menggunakan kondisi *default* LTSP, hal ini disebabkan oleh versi kernel 3.0.1 merupakan versi kernel terbaru yang memiliki fitur-fitur tambahan untuk mendukung kinerja LTSP..

Kinerja LTSP yang telah dioptimalisasi lebih baik, hal ini disebabkan fitur dan perbaikan yang ditambahkan pada server LTSP. Perbaikan dan fitur-fitur tambahan tersebut adalah :

1. *Local applications support*, yaitu fitur dimungkinkan untuk memilih

aplikasi mana yang dijalankan oleh *server* dan yang dijalankan secara lokal oleh *client*.

2. *Integrated LTSP-Cluster support.*

3. *Optional fully-featured dhcp client integration (udhcpc). Scripting xorg.conf* terbaru