BAB IV

HASIL DAN PENGUJIAN

4.1 Pengujian Terhadap *Eucalyptus Cloud*

4.1.1 Pengujian Konektifitas Pada *Server*

Pengujian konektifitas berfungsi untuk mengetahui apakah komputer *server* dapat terhubung dengan internet atau komputer lainnya Pengujian ini dapat dilakukan dengan melakukan *ping* atau dengan mengakses *website* diluar *localhost*. Berikut adalah contoh *ping* terhadap *server google*, yang menandakan komputer *server controller* dapat terhubung dengan jaringan internet:

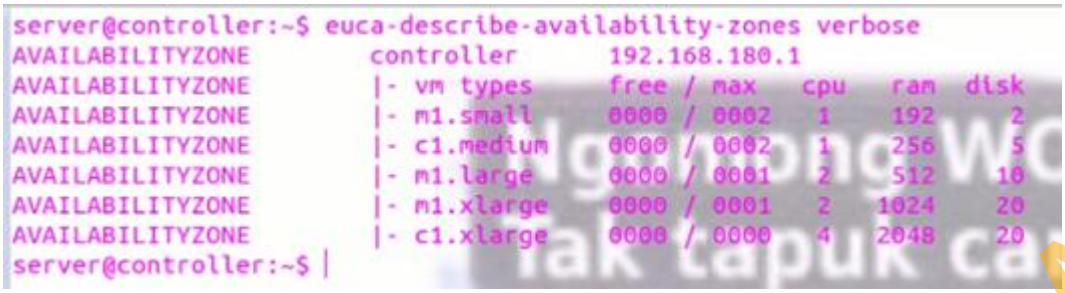
```
server@controller:~$ ping google.com
PING google.com (103.11.30.27) 56(84) bytes of data.
64 bytes from cache.google.com (103.11.30.27): icmp_req=1 ttl=57 time=12.3 ms
64 bytes from cache.google.com (103.11.30.27): icmp_req=2 ttl=57 time=12.8 ms
64 bytes from cache.google.com (103.11.30.27): icmp_req=3 ttl=57 time=12.4 ms
64 bytes from cache.google.com (103.11.30.27): icmp_req=4 ttl=57 time=12.7 ms
^C
--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3002ms
rtt min/avg/max/mdev = 12.389/12.624/12.890/0.205 ms
```

Gambar 4.1 Pengujian Konektifitas Pada *Server*

Pada Gambar 4.1 dapat kita lihat bahwa komputer *server* dapat melakukan cek koneksi dengan perintah *ping* terhadap *google*, hal ini berarti komputer *server* ini sudah terhubung dengan jaringan internet.

4.1.2 Memastikan *Eucalyptus* Berjalan Dengan Baik

Perngujian *Eucalyptus system* dapat berjalan dengan baik atau tidak dapat dilihat menggunakan perintah berikut ini:



Gambar 4.2 Pengujian *Eucalyptus Cloud*

Pada Gambar 4.2 terlihat bahwa semua komponen *eucalyptus* berjalan dengan baik dengan ditampilkannya daftar dari *resource* yang disediakan oleh *eucalyptus*. Untuk mengetahui *resource* yang sesuai dengan aplikasi yang akan gunakan, misalnya kita akan menjalankan *operating system* dengan RAM sebesar 512. Maka kita dapat menggunakan m1.large untuk sistem operasi *tersebut.*

Jika mengalami xxx mode pada kolom *free / max*, maka ada bagian yang tidak tersinkronisasi atau tidak berjalan pada sistem *eucalyptus*. Untuk itu dapat melakukan *restart* terhadap aplikasi yang tidak berjalan atau mengecek ulang apakah semua konfigurasi sudah benar atau tidak. Selain itu hal yang dapat dilakukan untuk mengecek apakah sistem berjalan dengan benar atau tidak. Bisa dilihat dari *error log* yang tersedia di */var/log/eucalyptus/*, disana dapat dilihat aplikasi manakah yang tidak berjalan dengan semestinya atau sedang mengalami *error*.

4.1.3 Pengujian Terhadap *Console* Dari *Cloud Client* Yang Berjalan

Pengujian terhadap *console* dilakukan karena penulis ingin memastikan apakah terjadi *error* saat *cloud client* sedang berjalan, jika tidak ada *error* maka *output*-nya haruslah seperti ini :

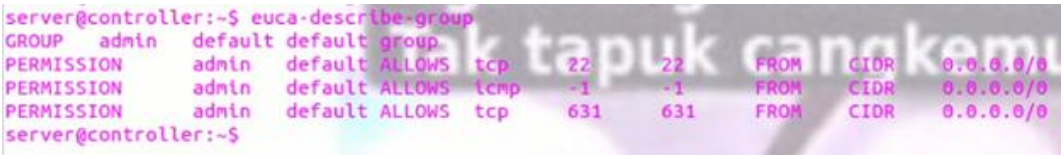
Gambar 4.3 Pengujian Terhadap *Console* Dari *Cloud Client*

Pada Gambar 4.3 dapat kita ketahui *output* dari *konsole instance* yang sedang berjalan, kita dapat mengetahui informasi dari *client* yang sedang berjalan seperti *kernel*, *memory*, *bios* dan sebagainya.

Menggunakan perintah *euca-get-console-ouptut id_instance* untuk menampilkan *output* seperti pada Gambar 4.3. Pastikan bahwa *output* dari *console* berjalan dengan baik dan tidak terjadi *error* atau kesalahan saat *cloud client* mulai berjalan.

4.1.4 Mamastikan Apakah *Rule* Berjalan Dengan Baik

Pengujian terhadap *rule* atau *access-list* dapat dilakukan dengan melihat apakah *rule* yang dipasang sudah sesuai dengan keperluan yang diterapkan oleh *administrator*, jalankah perintah *euca-describe-group* untuk melihat keseluruhan *rule* yang sudah ditetapkan, dan pastikan tidak ada *error* saat menjalankan perintah tersebut, jika *eucalyptus* tidak mengalami *error* maka *output* yang dihasilkan adalah sebagai berikut ini :



Gambar 4.4 Pengujian Terhadap *Access-List*

Pada Gambar 4.4 dapat diketahui *rule* yang telah diterapkan pada *server* ini. *Rule* tersebut meliputi *ssh* pada *port* 22, *icmp* pada *port* -1 serta *remote desktop* pada *port* 631 dengan tujuan dan sumber dari mana saja dengan tanda 0.0.0.0/0.

4.1.5 Pengujian Terhadap *Image*

Image adalah hal penting yang harus diperhatikan sebelum *cloud client* dijalankan. Untuk melakukan pengujian terhadap *image* kita dapat menjalankan perintah *euca-describe-images* sehingga menghasilkan *output* sebagai berikut :

Gambar 4.5 Pengujian Terhadap *Image*

Pada Gambar 4.5 dapat kita ketahui semua daftar dari *image* yang dapat digunakan atau tidak, disana juga terdapat semua informasi untuk *image* tersebut. Pada Gambar 4.5 kita dapat melihat sebuah *image* yang mempunyai EKI dan ERI dimana kedua komponen tersebut digunakan untuk menjalankan *instance* nantinya, EKI adalah kepanjangan dari *Eucalyptus Kernel Image* sedangkan ERI adalah kepanjangan dari *Eucalyptus Ram Image*.

Kemudian pastikan bahwa *image* yang sudah di-*upload* tidak mempunyai status *failed*, karena hanya *image* yang mempunyai status *available*-lah yang

dapat dijalankan, Untuk itu pastikan apakah *image* sudah berada pada keadaan *available* atau tidak.

4.1.6 Pengujian Terhadap Ketersediaan *Instance*

Pengujian terhadap *instance* dapat dilakukan perintah *euca-describe-instance* untuk mengetahui *instance* mana saja yang dapat berjalan dengan baik dan tidak mengalami *error*. Jika *instance* berjalan dengan baik dan tidak mengalami *error* maka *output* dari *instance* tersebut adalah sebagai berikut :

Gambar 4.6 Pengujian Terhadap *Instance*

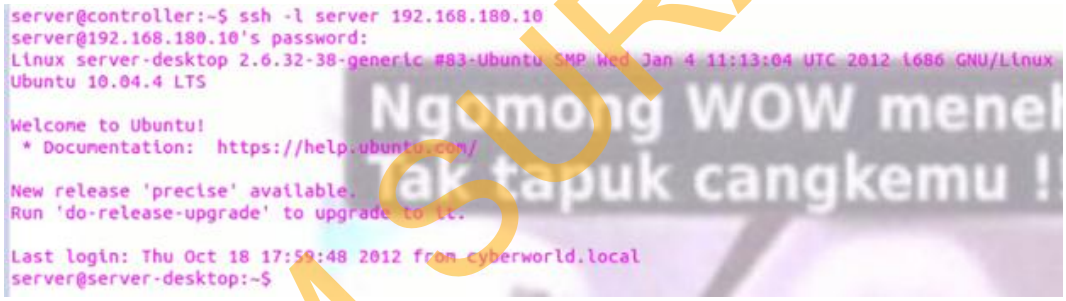
Pada Gambar 4.6 dapat diketahui bahwa sedang ada sebuah *instance* yang berjalan. Dapat diketahui apakah *instance* tersebut berjalan dengan melihat *state* dari instance tersebut. Jika *state* tersebut *running* maka *instance* tersebut berjalan, apabila *state* tersebut *terminate* maka maka *instance* tersebut berada dalam keadaan mati.

Pastikan *instance* tersebut mempunyai status *running*. Jika statusnya *terminate* atau *pending* maka sudah dapat dipastikan *instance* anda berlum berjalan dengan semestinya. Untuk mengetahui *error* dari *instance* yang mempunyai dua status diatas anda dapat mengecek pada *log error* di /var/log/eucalyptus/nova.log.

4.1.7 Pengujian Terhadap *Instance* Yang Sedang Berjalan

Untuk mengetahui apakah *instance* sudah berjalan dengan baik dan dapat diakses oleh *user* yang bersangkutan. Pengujian dapat dilakukan dengan melakukan *remote* terhadap *instance* yang sedang berjalan, karena pada penelitian kali *administrator* menggunakan *linux Ubuntu desktop* menjadi *instance* yang sedang berjalan. Maka cara untuk mengaksesnya adalah dengan melakukan *remote ssh* terhadap *instance* tersebut.

Jika tidak mempunyai masalah dengan *instance* yang sedang berjalan dan terhadap *rule* yang telah disediakan maka seharusnya dapat memperoleh *output* sebagai berikut :



Gambar 4.7 Pengujian Terhadap *Instance* Yang Sedang Berjalan

Pada gambar 4.7 dapat dilihat bahwa komputer server dapat melakukan koneksi secara remote dengan menggunakan protokol *ssh*. Untuk menggunakan *ssh* cukup menjalankan perintah *ssh –l username IP_yang dituju*.

4.1.8 Pengujian Terhadap *Range* Untuk *Client*

Untuk melakukan pengujian terhadap *range* dari IP yang telah disediakan sebelumnya, pengujian ini dapat dilakukan dengan perintah *euca-describe-address*. Pastikan *range* dari IP tersebut sesuai dengan konfigurasi yang sudah

diterapkan sebelumnya. Pada penelitian ini *range* yang digunakan adalah 192.168.180.0/28, sehingga *output* yang didapatkan adalah sebagai berikut :

Gambar 4.8 Pengujian Terhadap *Range* Untuk *Client*

Pada Gambar 4.8 dapat kita lihat bahwa IP yang tersedia mulai dari 192.168.180.3 - 192.168.180.10. IP tersebut nantinya akan dialokasikan untuk *instance* yang akan berjalan, seperti contoh IP 192.168.180.10 digunakan oleh *instance* dengan *id* i-4A5C0946.

4.2 Pengujian *OpenStack Cloud*

4.2.1 Pengujian Terhadap *Cloud Controller*

Pengujian ini dilakukan untuk mengetahui apakah setiap aplikasi yang berada pada *server contoller* berjalan dengan baik, termasuk konektifitas jaringan komputer serta sinkronisasi antara *server controller* dengan *server* lain.

A. Pengujian Konektifitas Pada *Server*

Pengujian konektifitas berfungsi untuk mengetahui apakah komputer *server* dapat terhubung dengan internet atau komputer lainnya, pengujian ini dapat dilakukan dengan melakukan *ping* atau dengan mengakses *website* diluar

localhost. berikut ini adalah contoh *ping* terhadap *server google*, yang menandakan komputer *server controller* dapat terhubung dengan jaringan internet:

```
server@controller:~$ ping google.com
PING google.com (103.11.30.27) 56(84) bytes of data.
64 bytes from cache.google.com (103.11.30.27): icmp_req=1 ttl=57 time=12.3 ms
64 bytes from cache.google.com (103.11.30.27): icmp_req=2 ttl=57 time=12.8 ms
64 bytes from cache.google.com (103.11.30.27): icmp_req=3 ttl=57 time=12.4 ms
64 bytes from cache.google.com (103.11.30.27): icmp_req=4 ttl=57 time=12.7 ms
^C
--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3002ms
rtt min/avg/max/mdev = 12.389/12.624/12.890/0.205 ms
```

Gambar 4.9 Pengujian Konektifitas Pada *Server* 1

Pada Gambar 4.9 dapat dilihat bahwa komputer *server* dapat melakukan cek koneksi dengan perintah *ping* terhadap *google*, hal ini berarti komputer *server* ini dapat terhubung dengan jaringan internet.

B. Pengujian Terhadap *Keystone*

Pengujian ini dilakukan untuk mengetahui apakah aplikasi *keystone* berjalan dengan baik. Untuk mengetahuinya dapat melakukan perintah berikut untuk melihat daftar *key* yang sudah dibuat dan dapat digunakan.

```
server@controller:~$ keystone user-list
+----------------------------------+---------+-------------------+--------+
|                id                | enabled |       email       |  name  |
+----------------------------------+---------+-------------------+--------+
| 4ceff7f497604884922b99999249a5d3 | True    | admin@foobar.com  | admin  |
| ba3c7255b7df41339383aef042cde2fa | True    | nova@foobar.com   | nova   |
| cad8c6084ecd40118ece0c2d94df9cf5 | True    | glance@foobar.com | glance |
| ce97035b64ea4ed9b335991c699814c5 | True    | swift@foobar.com  | swift  |
+----------------------------------+---------+-------------------+--------+
```

Gambar 4.10 *Keystone User List*

```
server@controller:~$ keystone tenant-list
+----------------------------------+---------+---------+
|                id                |   name  | enabled |
+----------------------------------+---------+---------+
| 1e4214cb7f9847408da5547e3c65e150 | service | True    |
| 1fc660d9bf8f41b18ac498cf4a18a261 | admin   | True    |
+----------------------------------+---------+---------+
```

Gambar 4.11 *Keystone Tenant List*

```
server@controller:~$ keystone role-list
+----------------------------------+---------+
|                id                |   name  |
+----------------------------------+---------+
| 5009d6d48bc14d3c8786049d31da4fc1 | Member  |
| ab02dd90cae140f8a2670e6f43630f79 | admin   |
+----------------------------------+---------+
```

Gambar 4.12 *Keystone Role List*

```
server@controller:~$ keystone service-list
+----------------------------------+----------+-------------+----------------------------+
|                id                |   name   |     type    |        description         |
+----------------------------------+----------+-------------+----------------------------+
| 0e57a0a39bc74bcf903b2f4521b28da5 | ec2      | ec2         | EC2 Service                |
| 1b66f8d977464dd0b944bf056b6c00f0 | nova     | compute     | OpenStack Compute Service  |
| 5c7b7ca2c3944fde8315b6622996c83b | swift    | object-store| OpenStack Storage Service  |
| 7c657b8ca37148bca6387af7d22eef34 | glance   | image       | OpenStack Image Service    |
| 9dd00bd106404c01b53d638416d8f99a | volume   | volume      | OpenStack Volume Service   |
| e8eeed522a9345d3ad3aa9c2682981e9 | keystone | identity    | OpenStack Identity Service |
+----------------------------------+----------+-------------+----------------------------+
```

Gambar 4.13 Keystone Service-List

Pada Gambar 4.10 - 4.13 dapat diketahui bahwa *keystone* berjalan dengan baik. Data-data tersebut yang nantinya akan dijadikan *kredential* untuk melakukan konfigurasi lebih lanjut terhadap openstack.

C. Pengujian Terhadap *Glance*

Pengujian ini dilakukan untuk mengetahui apakah *glance* berfungsi dengan baik. Pengujian ini sekaligus dapat dilakukan untuk melakukan pengujian terhadap *image*.

```
server@controller:~$ glance index
ID                                   Name                 Disk Format       Container Format    Size
------------------------------------ -------------------- ----------------- ----------------- -------------
df7fca6f-7a93-45c6-b50f-d410bda51a6b ubuntu10.04-desktop  qcow2             ovf                 2663972864
```

Gambar 4.14 *Glance Index*

Dari Gambar 4.14 dapat diketahui bahwa *image* yang akan digunakan sudah berhasil diupload. Administrator dapat mengetahui semua informasi tentang

image tersebut. Informasi tersebut diantaranya adalah id, nama, format, container format, dan size.

D. Pengujian Terhadap *Nova*

Pengujian ini dilakukan untuk mengetahui apakah *nova* berserta seluruh *service*-nya berfungsi dengan baik. Untuk mengetahui hal tersebut dapat menggukan perintah *sudo nova-manage list*. Jika *service*-nya berjalan, maka pada tab "*state*" kita akan diberikan output ":)" tapi apabila aplikasinya tidak berjalan maka kita akan diberikan *output* "XXX". Berikut ini adalah gambar *service nova* yang berjalan dengan benar.

```
server@controller:~$ sudo nova-manage service list
2013-01-10 08:24:42 DEBUG nova.utils [req-34f4932b-597c-4643-b9fe-2b11698240a7 None None] backend <module 'nova.db.sqlalchemy.api'
es/nova/db/sqlalchemy/api.pyc'> from (pid=21921) __get_backend /usr/lib/python2.7/dist-packages/nova/utils.py:658
Binary           Host                                 Zone             Status     State Updated_At
nova-consoleauth controller                           nova             enabled    :-)   2013-01-10 01:24:41
nova-network     controller                           nova             enabled    :-)   2013-01-10 01:24:41
nova-cert        controller                           nova             enabled    :-)   2013-01-10 01:24:41
nova-scheduler   controller                           nova             enabled    :-)   2013-01-10 01:24:41
nova-volume      controller                           nova             enabled    :-)   2013-01-10 01:24:42
nova-compute     controller                           nova             enabled    :-)   2013-01-10 01:24:41
nova-compute     node                                 nova             enabled    :-)   2013-01-10 01:24:39
```

Gambar 4.15 *Nova Service Controller*

Dari Gambar 4.15 dapat diketahui bahwa *nova* berjalan dengan baik. Hal ini dapat diketahui dari daftar *binary* yang ada, dimana *binary* adalah program pendukung dari *nova* untuk menjalankan *openstack*.

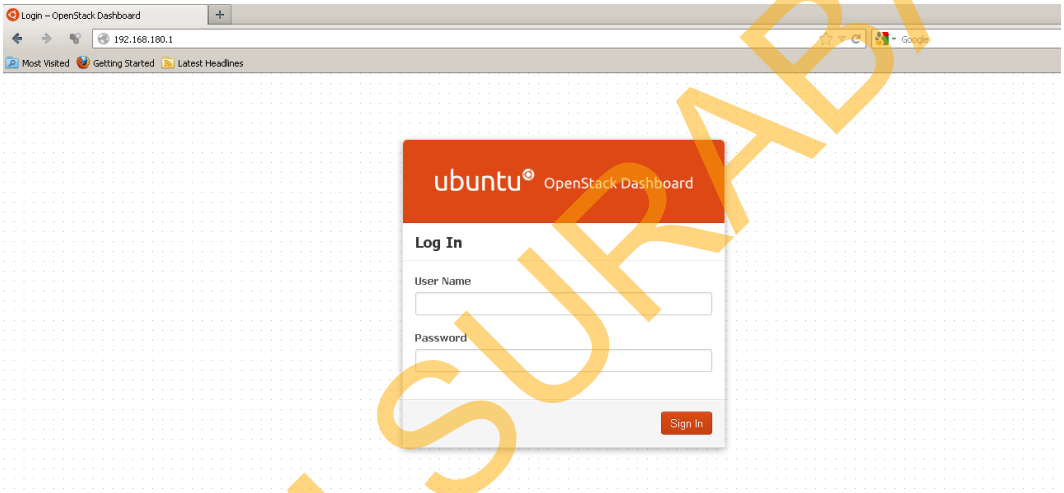
E. Pengujian Terhadap *Openstack Dashboard*

Pengujian ini dilakukan untuk mengetahui apakan *openstack dashboard* berjalan dengan baik. Untuk menguji *openstack dasboard* kita dapat menuliskan alamat dari *server* yang ter*install openstack dashboard*. Pada kasus kali ini penulis menggunakan *server controller* dengan IP 192.168.180.1 sehingga untuk

menguji *openstack dashboard* dapat dilakukan lewat *browser* dengan menuliskan alamat berikut kepada *browser*:

http://192.168.180.1/

Jika *openstack dashboard* anda berkerja dengan baik maka seharusnya mendapat halaman *login* dan anda dapat *login* kedalam *openstack dashboard* dan juga dapat melakukan konfigurasi degan *openstack dashboard* tersebut. Contoh halaman *login openstack dashboard* adalah sebagai berikut:



Gambar 4.16 *Login Openstack Dashboard Page*

Dari Gambar 4.16 dapat diketahui bahwa *web server* dari *OpenStack* berjalan dengan baik. Pada Gambar 4.16 adalah halaman awal *dashboard* yang digunakan untuk *login user* termasuk *administrator* yang konfigurasinya dapat dijalankan dari *web server* tersebut apabila *user* mempunyai hak *administrator.*

F. Pengujian Terhadap *Swift*

Pengujian ini dilakukan untuk mengetahui apakah *swift* berserta seluruh *service*-nya berfungsi dengan baik. Cara untuk mengetahuinya adalah degan melakukan perintah berikut:

swift -v -V 2.0 -A http://127.0.0.1:5000/v2.0/ -U service:swift -K swift stat

Bisa juga mengganti IP 127.0.0.1 dengan IP dari *server swift*. Jika mengkonfigurasinya dengan baik maka akan mendapatkan *output* sebagai berikut:

```
server@controller:~$ swift -v -V 2.0 -A http://127.0.0.1:5000/v2.0/ -U service:swift -K swift stat
StorageURL: http://192.168.180.1:8080/v1/AUTH_1e4214cb7f9847408da5547e3c65e150
Auth Token: 9ebefa4f717a41e48e89df1f07c23a0b
   Account: AUTH_1e4214cb7f9847408da5547e3c65e150
Containers: 1
   Objects: 0
     Bytes: 0
Accept-Ranges: bytes
X-Trans-Id: tx64c1386ee915490180c3e224d82e3f72
```

Gambar 4.17 *Swift Test*

Pada Gambar 4.17 dapat diketahui bahwa bahwa *swift* sudah berjalan dengan baik. Hal tersebut dapat dilihat dari informasi yang disediakan oleh *swift.* Jika mendapatkan *output containers : 0, objects : 0, bytes : 0,* tidak perlu panik, *swift* tetap bekerja dengan baik walaupun mengeluarkan *output 0,* hal itu terjadi karena belum melakukan *upload image* atau menambahkan *containers* serta *object* kedalam *swift*.

4.2.2 Pengujian Terhadap *Node Controller*

Pengujian ini dilakukan untuk mengetahui apakah setiap aplikasi yang berada pada *server contoller* berjalan dengan baik, termasuk konektifitas jaringan komputer serta sinkronisasi antara *server controller* dengan *server* lain.

A. Pengujian Konektifitas Pada *Server*

Pengujian konektifitas berfungsi untuk mengetahui apakah komputer *server* dapat terhubung dengan internet atau komputer lainnya, pengujian ini dapat dilakukan dengan melakukan *ping* atau dengan mengakses *website* diluar *localhost*. berikut ini adalah contoh *ping* terhadap *server google*, yang

menandakan komputer *server controller* dapat terhubung dengan jaringan internet:

```
server@controller:~$ ping google.com
PING google.com (103.11.30.27) 56(84) bytes of data.
64 bytes from cache.google.com (103.11.30.27): icmp_req=1 ttl=57 time=12.3 ms
64 bytes from cache.google.com (103.11.30.27): icmp_req=2 ttl=57 time=12.8 ms
64 bytes from cache.google.com (103.11.30.27): icmp_req=3 ttl=57 time=12.4 ms
64 bytes from cache.google.com (103.11.30.27): icmp_req=4 ttl=57 time=12.7 ms
^C
--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3002ms
rtt min/avg/max/mdev = 12.389/12.624/12.890/0.205 ms
```

Gambar 4.18 Pengujian Konektifitas Pada *Server* 2

Pada Gambar 4.18 dapat kita lihat bahwa komputer *server* dapat melakukan cek koneksi dengan perintah *ping* terhadap *google*. Hal ini berarti komputer *server* ini dapat terhubung dengan jaringan internet.

B. Pengujian Pada *Nova-Compute*

Pengujian ini dilakukan untuk mengetahui apakah *nova-compute* berfungsi dengan baik. Untuk mengetahui hal tersebut kita dapat menggukan perintah *sudo nova-manage list*, jika *service*nya berjalan maka pada *tab* "*state*" kita akan diberikan *output* ":)" tapi apabila aplikasinya tidak berjalan maka kita akan diberikan output "XXX". Berikut ini adalah gambar *service nova* yang berjalan dengan benar.

```
server@controller:~$ sudo nova-manage service list
2013-01-10 08:24:42 DEBUG nova.utils [req-34f4932b-597c-4643-b9fe-2b11698240a7 None None] backend <module 'nova.db.sqlalchemy.api'
es/nova/db/sqlalchemy/api.pyc'> from (pid=21921) __get_backend /usr/lib/python2.7/dist-packages/nova/utils.py:658
Binary           Host                                 Zone             Status     State Updated_At
nova-consoleauth controller                           nova             enabled    :-)   2013-01-10 01:24:41
nova-network     controller                           nova             enabled    :-)   2013-01-10 01:24:41
nova-cert        controller                           nova             enabled    :-)   2013-01-10 01:24:41
nova-scheduler   controller                           nova             enabled    :-)   2013-01-10 01:24:41
nova-volume      controller                           nova             enabled    :-)   2013-01-10 01:24:42
nova-compute     controller                           nova             enabled    :-)   2013-01-10 01:24:41
nova-compute     node                                 nova             enabled    :-)   2013-01-10 01:24:39
```

Gambar 4.19 *Nova Service Node*

Pada Gambar 4.19 dapat diketahui bahwa *nova-compute* pada komputer *node* berjalan dengan baik. Hal ini dapat dilihat dari *tab host*. Pada bagian *tab host*

paling bawah, terdapat kita lihat *hostname* "*node*". *Node* adalah *hostname* dari komputer *node* atau *server* kedua yang menjalankan *service nova-compute*.

4.2.3 Pengujian Terhadap *Cloud Client* (*Instance*)

Pengujian ini dilakukan untuk mengetahui apakah *client* / *instance* dapat berfungsi dengan baik, pengujian ini meliputi pengujian akses terhadap *instance*, Sedangkan untuk pengujian koneksi kita dapat menggunakan perintah *ssh*, berikut adalah gambar dari pengujian akses dengan *ssh*:

```
server@controller:~$ ssh -l client 192.168.4.2
client@192.168.4.2's password:
Linux client-desktop 2.6.32-33-generic #70-Ubuntu SMP Thu Jul 7 21:09:46 UTC 2011 i686 GNU/Linux
Ubuntu 10.04.3 LTS

Welcome to Ubuntu!
 * Documentation:  https://help.ubuntu.com/

285 packages can be updated.
221 updates are security updates.

New release 'precise' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Thu Jan 10 08:35:32 2013 from 192.168.4.1
```

Gambar 4.20 Uji Akses SSH Pada *Client*

Pada Gambar 4.20 dapat dilihat bahwa komputer *server* dapat melakukan koneksi secara *remote* dengan menggunakan protokol *ssh*, untuk menggunakan *ssh* kita cukup menjalankan perintah *ssh –l username IP_yang ditujuh*.

4.3 *Benchmarking* Pada *Instance Eucalyptus*

Pengujian ini dilakukan untuk menghasilkan nilai data *performance* dari *benchmarking instance Eucalyptus* agar dapat diproses dalam analisa statistika. Pengujian ini meliputi *benchmarking memory* (RAM), *disk* (Harddisk) dan *processor* pada *instance Eucalyptus*.

4.3.1 *Benchmarking Memory* Pada *Instance Eucalyptus*

Setelah melakukan *benchmarking* terhadap *memory* (RAM) *instance* yang berjalan pada *Eycalyptus* maka didapatkan data *performance* sebagai berikut:

Tabel 4.1 Hasil *Benchmarking Memory* Pada *Instance Eucalyptus* (MB/s)

Type	Average		Type	Average	
Bechmark	Integer	Floating Point	Bechmark	Integer	Floating Point
1	4067,65	4177,05	16	4059,18	4169,17
2	4055,37	4172,56	17	4050,55	4166,39
3	4060,58	4173,65	18	4051,05	4165,95
4	4056,49	4175,86	19	4050,99	4164,64
5	4058,18	4168,86	20	4045,4	4165,33
6	4053,1	4171,69	21	4046,21	4163,04
7	4057,44	4175,6	22	4051,28	4165,62
8	4058,84	4169,91	23	4047,06	4166,07
9	4062,58	4174,35	24	4045,69	4161,3
10	4059,07	4173,28	25	4046,57	4164,46
11	4056,47	3962,97	26	4047,82	4163,81
12	4051,22	4062,845	27	4046,93	4163,06
13	4045,97	4162,72	28	4054,52	4168,7
14	4048,89	4167,37	29	4059,05	4171,85
15	4044,92	4167,66	30	4051,75	4167,38

Dari Tabel 4.1 dapat dilihat bahwa pada penelitian kali ini penulis melakukan *benchmarking* terhadap *memory* (RAM) pada *instance Eucalyptus* dengan menghasilkan dua variabel yaitu *integer* dan *floating point*.

4.3.2 *Benchmarking Disk* Pada *Instance Eucalyptus*

Setelah melakukan *benchmarking* terhadap *disk* (*Harddisk) instance* yang berjalan pada *Eucalyptus*, maka didapatkan data *performance* sebagai berikut:

Tabel 4.2 Hasil *Benchmarking Disk* Pada *Instance Eucalyptus* (MB/s)

Size	512		Size	512	
Iozone	Read	Write	Iozone	Read	Write
1	15,01	580,8	16	263,605	300,035
2	15,55	514,36	17	139,4075	450,0525
3	15,24	583,56	18	15,21	600,07
4	15,53	588,33	19	15,6	586,82
5	14,55	577,62	20	15,13	607,41
6	14,935	560,765	21	15,23125	579,2275
7	15,32	543,91	22	15,28	576,28
8	14,955	559,735	23	15,43	545,15
9	14,59	575,56	24	15,3325	551,045
10	14,64	541,08	25	15,235	556,94
11	15,11	562,27	26	15,1375	562,835
12	15,58	583,46	27	15,04	568,73
13	15,4	595,74	28	15,09	599,67
14	14,47	586,2	29	15,14	630,61
15	7,235	293,1	30	15,15	593,14

Dari Tabel 4.2 dapat dilihat bahwa pada penelitian kali ini penulis melakukan *benchmarking* terhadap *disk* (*harddisk*) pada *instance Eucalyptus* dengan menghasilkan dua variabel yaitu *read* dan *write*.

4.3.3 *Benchmarking Processor* Pada *Instance Eucalyptus*

Setelah melakukan *benchmarking* terhadap *processor instance* yang berjalan pada *Eucalyptus*, maka didapatkan data *performance* sebagai berikut:

Tabel 4.3 Hasil *Benchmarking Processor* Pada *Instance* Eucalyptus

C-Ray	Processor	C-Ray	Processor
1	377,37	16	377,16
2	377,4	17	377,17
3	377,3625	18	377,09
4	377,43	19	377,13
5	377,355	20	377,11
6	377,28	21	377,12
7	377,255	22	377,115
8	377,23	23	377,19
9	377,26	24	377,15
10	377,25	25	377,17
11	377,04	26	377,18
12	377,185	27	377,16
13	377,33	28	377,15
14	377,54	29	377,3
15	188,77	30	377,225

Dari Tabel 4.3 dapat dilihat bahwa pada penelitian kali ini penulis melakukan *benchmarking* terhadap *processor* pada *instance Eucalyptus*.

4.4 *Benchmarking* Pada *Instance OpenStack*

Pengujian ini dilakukan untuk menghasilkan nilai data *performance* dari *benchmarking instance OpenStack* agar dapat diproses dalam analisa statistika. Pengujian ini meliputi *benchmarking memory* (RAM), *disk* (Harddisk) dan *processor* pada *instance OpenStack*.

4.4.1 *Benchmarking Memory* Pada *Instance OpenStack*

Setelah melakukan *benchmarking* terhadap *memory* (RAM) *instance* yang berjalan pada *OpenStack*, maka didapatkan data *performance* sebagai berikut:

Tabel 4.4 Hasil *Benchmarking Memory* Pada *Instance OpenStack* (MB/s)

Type	Average		Type	Average	
Bechmark	Integer	Floating Point	Bechmark	Integer	Floating Point
1	3957,87	4067,69	16	3965,1	4070,82
2	3965,33	4072,61	17	3959,44	4068,85
3	3957,07	4059,87	18	3962,77	4071,58
4	3954,14	4067,9	19	3960,07	4066,34
5	3952,26	4072,94	20	3960,86	4065,35
6	3961,51	4073,75	21	3952,94	4070,75
7	3964,55	4066,62	22	3955,9	4068
8	3960,79	4064,59	23	3959,77	4068,66
9	3965,42	4067,76	24	3963,68	4064,59
10	3957,755	4069,615	25	3966,26	4078,45
11	3950,09	4071,47	26	3954,18	4061,4
12	3963,38	4069,89	27	3959,12	4073,92
13	3956,64	4058,89	28	3953,98	4064,91
14	3958,73	4059,84	29	3957,97	4066,25
15	3953,86	4068,05	30	3960,15	4081,1

Dari Tabel 4.4 dapat dilihat bahwa pada penelitian kali ini penulis melakukan *benchmarking* terhadap *memory* (RAM) pada *instance OpenStack* dengan menghasilkan dua variabel yaitu *integer* dan *floating point*.

4.4.2 *Benchmarking Disk* Pada *Instance OpenStack*

Setelah melakukan *benchmarking* terhadap *disk* (*harddisk*) *instance* yang berjalan pada *OpenStack*, maka didapatkan data *performance* sebagai berikut:

Tabel 4.5 Hasil *Benchmarking Disk* Pada *Instance OpenStack* (MB/s)

Size	512		Size	512	
Iozone	Read	Write	Iozone	Read	Write
1	86,56	112,92	16	122,75	107,57
2	110,1	107,39	17	126,68	112,88
3	124,6	111,28	18	120,01	108,36
4	119,24	109,93	19	125,38	110,18
5	122,88	108,13	20	124,49	112,77
6	123,12	116,46	21	126,09	111,19
7	123,97	108,26	22	122,02	116,32
8	126,18	111,84	23	121,03	109,13
9	123,22	110,74	24	119,98	104,87
10	123,91	113,22	25	121,44	109,86
11	124,93	107,4	26	119,02	105,76
12	113,61	115,22	27	116,6	113,73
13	125,41	105,36	28	123,57	114,46
14	122,98	114,14	29	120,61	112,18
15	125,91	107,93	30	124,53	113,37

Dari Tabel 4.5 dapat dilihat bahwa pada penelitian kali ini penulis melakukan *benchmarking* terhadap *disk* (*harddisk*) pada *instance OpenStack* dengan menghasilkan dua variabel yaitu *read* dan *write*.

4.4.3 *Benchmarking Processor* Pada *Instance OpenStack*

Setelah melakukan *benchmarking* terhadap *processor instance* yang berjalan pada *OpenStack*, maka didapatkan data *performance* sebagai berikut:

Tabel 4.6 Hasil *Benchmarking Processor* Pada *Instance OpenStack*

C-Ray	Processor	C-Ray	Processor
1	747,98	16	748,06
2	747,82	17	748,06
3	747,91	18	747,78
4	748,16	19	748,16
5	747,85	20	748,02
6	747,63	21	748
7	747,9	22	748,1
8	747,87	23	748,36
9	748,1	24	747,91
10	748,01	25	748,13
11	747,81	26	748,12
12	747,92	27	747,91
13	747,7	28	747,8
14	748,05	29	748,17
15	748,17	30	747,87

Dari Tabel 4.6 dapat dilihat bahwa pada penelitian kali ini penulis melakukan *benchmarking* terhadap *processor* pada *instance OpenStack*.

4.5 Penghitungan Statistika

Penghitungan statistika ini dilakukan untuk mengetahui apakah *performance memory* (RAM), *disk* (*harddisk*) dan *processor* dari *Eucalyptus* lebih baik dari *OpenStack* ataupun sebaliknya.

4.5.1 Perhitungan Statistika Data *Performance* Pada *Memory*

A. Prosedur Perhitungan

1. Menjumlahkan hasil dari *benchmarking memory* pada *integer* ($\sum_{\text{integer}}$) dan *floating point* ($\sum_{\text{floating point}}$)
2. Mencari rata-rata dari hasil penjumlahan pada *integer* ($\bar{X}_{integer}$) dan *floating point* ($\bar{X}_{floating\ point}$)
3. Mengurangkan X_i dengan rata-rata ($\bar{X}$) dari penjumlahan *benchmarking memory* pada *integer* dan *floating point*
4. Mengkuadratkan hasil dari pengurangan X_i dengan rata-rata ($\bar{X}$) pada *integer* dan *floating point*. $(X_i - \bar{X})^2$
5. Mencari pendugaan titik (S^2) pada tiap *integer* dan *floating point*.
6. Menghitung variansi (F) pada tiap *integer* dan *floating point*.
7. Mencari uji rataan (T') pada tiap *integer* dan *floating point*.

B. Perhitungan Statistika

Untuk dapat mengambil keputusan mana *cloud* yang lebih baik antara *Eucalyptus* dan *OpenStack* dari sisi *memory* (RAM). Penulis melakukan perhitungan uji rata-rata berikut untuk membandingkan keunggulan dari masing-masing.

1. Tipe Data *Integer*

Berikut ini adalah perhitungan uji rata-rata terhadap data hasil *benchmarking memory* pada tipe data *integer* dari *Eucalyptus*:

Tabel 4.7 Perhitungan Data *Performance Integer Eucalyptus*

Type	Average		
Bechmark	Integer	Xi - Xbar	(Xi - Xbar)^2
1	4067,650	14,623	213,832
2	4055,370	2,343	5,490
3	4060,580	7,553	57,048
4	4056,490	3,463	11,992
5	4058,180	5,153	26,553
6	4053,100	0,073	0,005
7	4057,440	4,413	19,475
8	4058,840	5,813	33,791
9	4062,580	9,553	91,260
10	4059,070	6,043	36,518
11	4056,470	3,443	11,854
12	4051,220	-1,807	3,265
13	4045,970	-7,057	49,801
14	4048,890	-4,137	17,115
15	4044,920	-8,107	65,723
16	4059,180	6,153	37,859
17	4050,550	-2,477	6,136
18	4051,050	-1,977	3,909
19	4050,990	-2,037	4,149
20	4045,400	-7,627	58,171
21	4046,210	-6,817	46,471
22	4051,280	-1,747	3,052
23	4047,060	-5,967	35,605
24	4045,690	-7,337	53,832
25	4046,570	-6,457	41,693
26	4047,820	-5,207	27,113
27	4046,930	-6,097	37,173
28	4054,520	1,493	2,229
29	4059,050	6,023	36,277
30	4051,750	-1,277	1,631
Jumlah	121590,820	0,010	1039,022
Rata-rata	4053,027	0,000	34,634
		S^2	35,828

Berikut ini adalah perhitungan uji rata-rata terhadap data hasil *benchmarking memory* pada tipe data *integer* dari *OpenStack*:

Tabel 4.8 Perhitungan Data *Performance Integer OpenStack*

Type	Average		
Bechmark	Integer	Xi - Xbar	(Xi - Xbar)^2
1	3957,870	-1,183	1,399
2	3965,330	6,277	39,401
3	3957,070	-1,983	3,932
4	3954,140	-4,913	24,138
5	3952,260	-6,793	46,145
6	3961,510	2,457	6,037
7	3964,550	5,497	30,217
8	3960,790	1,737	3,017
9	3965,420	6,367	40,539
10	3957,755	-1,298	1,685
11	3950,090	-8,963	80,335
12	3963,380	4,327	18,723
13	3956,640	-2,413	5,823
14	3958,730	-0,323	0,104
15	3953,860	-5,193	26,967
16	3965,100	6,047	36,566
17	3959,440	0,387	0,150
18	3962,770	3,717	13,816
19	3960,070	1,017	1,034
20	3960,860	1,807	3,265
21	3952,940	-6,113	37,369
22	3955,900	-3,153	9,941
23	3959,770	0,717	0,514
24	3963,680	4,627	21,409
25	3966,260	7,207	51,941
26	3954,180	-4,873	23,746
27	3959,120	0,067	0,004
28	3953,980	-5,073	25,735
29	3957,970	-1,083	1,173
30	3960,150	1,097	1,203
Jumlah	118771,585	-0,005	556,330
Rata-rata	3959,053	0,000	18,544
		S^2	19,184

Uji variansi:

$H_0 : \sigma_1^2 = \sigma_2^2$, hal ini berarti kedua data bersifat *homogen*

$H_1 : \sigma_1^2 \neq \sigma_2^2$, hal ini berarti kedua data bersifat *heterogen*

Taraf nyata $\alpha = 0{,}05$

Statistika uji : $F = \frac{S_1^2}{S_2^2} = \frac{35{,}828}{19{,}184} = 1{,}867$

dimana $S^2 = \frac{\sum(X-\bar{X})^2}{n-1}$

Daerah kritis : tolak H_0, jika $F < F_{1-\alpha/2;(V1,V2)}$ dan $F > F_{\alpha/2;(V1,V2)}$

dimana $F_{1-\alpha/2;(V1,V2)} = F_{1-0{,}05/2;(29,29)} = F_{1-0{,}25;(29,29)} = F_{0{,}975;(29,29)} = \frac{1}{F_{0{,}025;(29,29)}} = \frac{1}{2{,}07} = 0{,}483$

$F_{\alpha/2;(V1,V2)} = F_{0{,}05/2;(29,29)} = F_{0{,}025;(29,29)} = 2{,}07$

Sehingga tolak H_0 jika $F < 0{,}483$ dan $F > 2{,}07$

Kesimpulan: terima H_0, karena $F = 1{,}867 < 2{,}07$. Hal ini berarti kedua data tersebut bersifat *homogen*. Karena itu dilakukan uji rata-rata terhadap data *integer* sebagai berikut:

Uji rata-rata:

$H_0 : \mu_1 = \mu_2$, hal ini berarti rata-rata kedua data sama

$H_1 : \mu_1 \neq \mu_2$, hal ini berarti rata-rata kedua data tidak sama

dimana μ_1 adalah rata-rata dari *integer Eucalyptus* dan μ_2 adalah rata-rata dari *integer OpenStack.*

Taraf nyata $\alpha = 0{,}05$

Statistik uji: $T = \frac{(\bar{X}_1 - \bar{X}_2) - d_0}{S_p \sqrt{1/n_1 + 1/n_2}}$

$S_p^2 = \frac{(n_1-1)S_1^2 + (n_2-1)S_2^2}{n_1+n_2-2} = \frac{(30-1)(35{,}828)+(30-1)(19{,}184)}{30+30-2} = \frac{(29)(35{,}828)+(29)(19{,}184)}{58} = \frac{1039{,}012+556{,}336}{58} = \frac{1595{,}348}{58} = 27{,}506$

$S_p = \sqrt{27{,}506} = 5{,}245$

Sehingga: $T = \frac{(4053{,}027-3959{,}053)-0}{5{,}245\sqrt{1/30+1/30}} = \frac{(93{,}974)-0}{5{,}245\sqrt{2/30}} = \frac{93{,}974}{5{,}245\sqrt{0{,}0667}} =$

$\frac{93{,}974}{(5{,}245).(0{,}258)} = \frac{93{,}974}{1{,}354} = 69{,}405$

Daerah kritis : tolak H_0 jika $T < -T_{\alpha/2;v}$ dan $T > T_{\alpha/2;v}$

dimana v = n1 + n2 – 2 = 30 + 30 – 2 = 58

$T_{\alpha/2;v} = T_{0{,}05/2;58} = T_{0{,}025;58} = 2{,}002$

sehingga: Tolak H_0 jika $T < -2{,}002$ dan $T > 2{,}002$

Kesimpulan: tolak H_0, hal ini berarti rata-rata kedua data tidak sama. Karena T = 69.405 > 2.002, maka rata-rata μ_1 lebih besar dari pada μ_2.

Hal ini dapat disimpulkan bahwa *performance memory* pada *Eucalyptus* itu lebih baik dari *OpenStack*, jika tipe data yang di proses adalah *integer*. Hal ini disebabkan karena rata-rata *performance memory Eucalyptus* pada tipe data *integer* lebih besar dari pada *performance memory OpenStack*.

2. Tipe Data *Floating Point*

Berikut ini adalah perhitungan uji rata-rata terhadap data hasil *benchmarking memory* pada tipe data *floating point* dari *Eucalyptus*:

Tabel 4.9 Perhitungan Data *Performance Floating Point Eucalyptus*

Type	Average		
Bechmark	Floating Point	Xi - Xbar	(Xi - Xbar)^2
1	4177,050	18,945	358,913
2	4172,560	14,455	208,947
3	4173,650	15,545	241,647
4	4175,860	17,755	315,240
5	4168,860	10,755	115,670
6	4171,690	13,585	184,552
7	4175,600	17,495	306,075
8	4169,910	11,805	139,358
9	4174,350	16,245	263,900
10	4173,280	15,175	230,281
11	3962,970	-195,135	38077,668
12	4062,845	-95,260	9074,468
13	4162,720	4,615	21,298
14	4167,370	9,265	85,840
15	4167,660	9,555	91,298
16	4169,170	11,065	122,434
17	4166,390	8,285	68,641
18	4165,950	7,845	61,544
19	4164,640	6,535	42,706
20	4165,330	7,225	52,201
21	4163,040	4,935	24,354
22	4165,620	7,515	56,475
23	4166,070	7,965	63,441
24	4161,300	3,195	10,208
25	4164,460	6,355	40,386
26	4163,810	5,705	32,547
27	4163,060	4,955	24,552
28	4168,700	10,595	112,254
29	4171,850	13,745	188,925
30	4167,380	9,275	86,026
Jumlah	124743,145	-0,005	50701,850
Rata-rata	4158,105	0,000	1690,062
		S^2	1748,340

Berikut ini adalah perhitungan uji rata-rata terhadap data hasil *benchmarking memory* pada tipe data *floating point* dari *OpenStack*:

Tabel 4.10 Perhitungan Data *Performance Floating Point OpenStack*

Type	Average		
Bechmark	Floating Point	Xi - Xbar	(Xi - Xbar)^2
1	4067,690	-0,725	0,526
2	4072,610	4,195	17,598
3	4059,870	-8,545	73,017
4	4067,900	-0,515	0,265
5	4072,940	4,525	20,476
6	4073,750	5,335	28,462
7	4066,620	-1,795	3,222
8	4064,590	-3,825	14,631
9	4067,760	-0,655	0,429
10	4069,615	1,200	1,440
11	4071,470	3,055	9,333
12	4069,890	1,475	2,176
13	4058,890	-9,525	90,726
14	4059,840	-8,575	73,531
15	4068,050	-0,365	0,133
16	4070,820	2,405	5,784
17	4068,850	0,435	0,189
18	4071,580	3,165	10,017
19	4066,340	-2,075	4,306
20	4065,350	-3,065	9,394
21	4070,750	2,335	5,452
22	4068,000	-0,415	0,172
23	4068,660	0,245	0,060
24	4064,590	-3,825	14,631
25	4078,450	10,035	100,701
26	4061,400	-7,015	49,210
27	4073,920	5,505	30,305
28	4064,910	-3,505	12,285
29	4066,250	-2,165	4,687
30	4081,100	12,685	160,909
Jumlah	122052,455	0,005	744,067
Rata-rata	4068,415	0,000	24,802
		S^2	25,657

Uji variansi:

$H_0 : \sigma_1^2 = \sigma_2^2$, hal ini berarti kedua data bersifat *homogen*.

$H_1 : \sigma_1^2 \neq \sigma_2^2$, hal ini berarti kedua data bersifat *heterogen*.

Taraf nyata $\alpha = 0{,}05$

Statistika uji : $F = \frac{S_1^2}{S_2^2} = \frac{1748{,}340}{25{,}657} = 68{,}14$

dimana $S^2 = \frac{\sum(X-\bar{X})^2}{n-1}$

Daerah kritis : tolak H_0, jika $F < F_{1-\alpha/2;(V1,V2)}$ dan $F > F_{\alpha/2;(V1,V2)}$

dimana $F_{1-\alpha/2;(V1,V2)} = F_{1-0{,}05/2;(29,29)} = F_{1-0{,}25;(29,29)} = F_{0{,}975;(29,29)} = \frac{1}{F_{0{,}025;(29,29)}} = \frac{1}{2{,}07} = 0{,}483$

$F_{\alpha/2;(V1,V2)} = F_{0{,}05/2;(29,29)} = F_{0{,}025;(29,29)} = 2{,}07$

Sehingga tolak H_0 jika $F < 0{,}483$ dan $F > 2{,}07$

Kesimpulan: tolak H_0, karena $F = 68{,}14 > 2{,}07$. hal ini berarti kedua data tersebut bersifat *heterogen*. Karena itu dilakukan uji rata-rata terhadap data *floating point* sebagai berikut:

Uji rata-rata:

$H_0 : \mu_1 = \mu_2$, hal ini berarti rata-rata kedua data sama

$H_1 : \mu_1 \neq \mu_2$, hal ini berarti rata-rata kedua data tidak sama

dimana μ_1 adalah rata-rata dari *floating point Eucalyptus* dan μ_2 adalah rata-rata dari *floating point OpenStack*

Taraf nyata $\alpha = 0{,}05$

Statistik uji: $T = \frac{\left(\bar{X}_1 - \bar{X}_2\right) - d_0}{\sqrt{S_1^2/n_1 + S_2^2/n_2}} = \frac{(4158{,}105 - 4068{,}415) - 0}{\sqrt{1748{,}340/30 + 25{,}657/30}} =$

$$\frac{(89{,}69) - 0}{\sqrt{58{,}278 + 0{,}855}} = \frac{89{,}69}{\sqrt{59{,}133}} = \frac{89{,}69}{7{,}702} = 32{,}316$$

Daerah kritis : tolak H_0 jika $T < -T_{\alpha/2;v}$ dan $T > T_{\alpha/2;v}$

dimana $v = \frac{(S_1^2/n_1 + S_2^2/n_2)^2}{\frac{(S_1^2/n_2)^2}{n_1 - 1} + \frac{(S_2^2/n_2)^2}{n_2 - 1}} = \frac{(1748{,}340/30 + 25{,}657/30)^2}{\frac{(1748{,}340/30)^2}{30-1} + \frac{(25{,}657/30)^2}{30-1}} =$

$$\frac{(58{,}278 + 0{,}855)^2}{\frac{(58{,}278)^2}{29} + \frac{(0{,}855)^2}{29}} = \frac{(59{,}133)^2}{\frac{3396{,}325}{29} + \frac{0{,}731}{29}} = \frac{3496{,}712}{117{,}115 + 0{,}0252} = \frac{3496{,}712}{117{,}139}$$

$= 29{,}851$

$T_{\alpha/2;v} = T_{0{,}05/2;30} = T_{0{,}025;30} = 2{,}042$

sehingga: Tolak H_0 jika $T < -2{,}042$ dan $T > 2{,}042$

Kesimpulan: tolak H_0, hal ini berarti rata-rata kedua data tidak sama. Karena $T = 32.316 > 2.042$, maka rata-rata μ_1 lebih besar dari pada μ_2.

Hal ini dapat disimpulkan bahwa *performance memory* pada *OpenStack* itu lebih baik dari *Eucalyptus*, jika tipe data yang di proses adalah *floating point*. Hal ini disebabkan karena rata-rata *performance memory OpenStack* pada tipe data *integer* lebih besar dari pada *performance memory Eucalyptus*.

4.5.2 Perhitungan Statistika Data *Performance* Pada *Disk*

A. Prosedur Perhitungan

1. Menjumlahkan hasil dari *benchmarking disk* pada *read performance* ($\sum_{read}$) dan *write performance* ($\sum_{write}$)
2. Mencari rata-rata dari hasil penjumlahan pada *read performance* ($\bar{X}_{read}$) dan *write performance* ($\bar{X}_{write}$)
3. Mengurangkan X_i dengan rata-rata ($\bar{X}$) dari penjumlahan *benchmarking disk* pada *read performance* dan *write performance*
4. Mengkuadratkan hasil dari pengurangan X_i dengan rata-rata ($\bar{X}$) pada *read performance* dan *write performance*. $(X_i - \bar{X})^2$
5. Mencari pendugaan titik (S^2) pada *read performance* dan *write performance*
6. Menghitung variansi (F) pada *read performance* dan *write performance*
7. Mencari uji rataan (T') pada *read performance* dan *write performance*

B. Hasil Perhitungan

Untuk dapat mengambil keputusan mana *cloud* yang lebih baik antara *Eucalyptus* dan *OpenStack* dari sisi *disk* (*harddisk*). Penulis melakukan perhitungan uji rata-rata berikut untuk membandingkan keunggulan dari masing-masing.

1. *Read Performance*

Berikut ini adalah perhitungan terhadap uji rata-rata data hasil *benchmarking disk* (*Harddisk*) pada *read performance* dari *Eucalyptus*:

Tabel 4.11 Perhitungan Data *Performance Read Performance Eucalyptus*

Size	512		
Iozone	Read	Xi - Xbar	(Xi - Xbar)^2
1	15,010	-0,104	0,011
2	15,550	0,436	0,190
3	15,240	0,126	0,016
4	15,530	0,416	0,173
5	14,550	-0,564	0,318
6	14,935	-0,179	0,032
7	15,320	0,206	0,042
8	14,955	-0,159	0,025
9	14,590	-0,524	0,275
10	14,640	-0,474	0,225
11	15,110	-0,004	0,000
12	15,580	0,466	0,217
13	15,400	0,286	0,082
14	14,470	-0,644	0,415
15	14,655	-0,459	0,211
16	14,840	-0,274	0,075
17	15,025	-0,089	0,008
18	15,210	0,096	0,009
19	15,600	0,486	0,236
20	15,130	0,016	0,000
21	15,231	0,117	0,014
22	15,280	0,166	0,028
23	15,430	0,316	0,100
24	15,333	0,218	0,048
25	15,235	0,121	0,015
26	15,138	0,023	0,001
27	15,040	-0,074	0,005
28	15,090	-0,024	0,001
29	15,140	0,026	0,001
30	15,150	0,036	0,001
Jumlah	453,406	-0,014	2,772
Rata-rata	15,114	0,000	0,092
		S^2	0,096

Berikut ini adalah perhitungan uji rata-rata terhadap data hasil *benchmarking disk* (*harddisk*) pada *read performance* dari *OpenStack*:

Tabel 4.12 Perhitungan Data *Performance Read Performance OpenStack*

Size	512		
Iozone	Read	Xi - Xbar	(Xi - Xbar)^2
1	86,560	-34,467	1187,974
2	110,100	-10,927	119,399
3	124,600	3,573	12,766
4	119,240	-1,787	3,193
5	122,880	1,853	3,434
6	123,120	2,093	4,381
7	123,970	2,943	8,661
8	126,180	5,153	26,553
9	123,220	2,193	4,809
10	123,910	2,883	8,312
11	124,930	3,903	15,233
12	113,610	-7,417	55,012
13	125,410	4,383	19,211
14	122,980	1,953	3,814
15	125,910	4,883	23,844
16	122,750	1,723	2,969
17	126,680	5,653	31,956
18	120,010	-1,017	1,034
19	125,380	4,353	18,949
20	124,490	3,463	11,992
21	126,090	5,063	25,634
22	122,020	0,993	0,986
23	121,030	0,003	0,000
24	119,980	-1,047	1,096
25	121,440	0,413	0,171
26	119,020	-2,007	4,028
27	116,600	-4,427	19,598
28	123,570	2,543	6,467
29	120,610	-0,417	0,174
30	124,530	3,503	12,271
Jumlah	3630,820	0,010	1633,922
Rata-rata	121,027	0,000	54,464
		S^2	56,342

Uji variansi:

$H_0 : \sigma_1^2 = \sigma_2^2$, hal ini berarti kedua data bersifat *homogen*.

$H_1 : \sigma_1^2 \neq \sigma_2^2$, hal ini berarti kedua data bersifat *heterogen*.

Taraf nyata $\alpha = 0{,}05$

Statistika uji : $F = \frac{S_1^2}{S_2^2} = \frac{56{,}342}{0{,}096} = 586{,}896$

Dimana $S^2 = \frac{\Sigma(X-\bar{X})^2}{n-1}$

Daerah kritis : tolak H_0, jika $F < F_{1-\alpha/2;(V1,V2)}$ dan $F > F_{\alpha/2;(V1,V2)}$

dimana $F_{1-\alpha/2;(V1,V2)} = F_{1-0{,}05/2;(29,29)} = F_{1-0{,}025;(29,29)} = F_{0{,}975;(29,29)} = \frac{1}{F_{0{,}025;(29,29)}} = \frac{1}{2{,}07} = 0{,}483$

$F_{\alpha/2;(V1,V2)} = F_{0{,}05/2;(29,29)} = F_{0{,}025;(29,29)} = 2{,}07$

Sehingga tolak H_0 jika $F < 0{,}483$ dan $F > 2{,}07$

Kesimpulan: tolak H_0, karena $F = 586{,}896 > 2{,}07$. hal ini berarti kedua data tersebut bersifat *heterogen*. Karena itu dilakukan uji rata-rata terhadap data *read performance* sebagai berikut:

Uji rata-rata:

$H_0 : \mu_1 = \mu_2$, hal ini berarti rata-rata kedua data sama

$H_1 : \mu_1 \neq \mu_2$, hal ini berarti rata-rata kedua data tidak sama

dimana μ_1 adalah rata-rata dari *read performance Eucalyptus* dan μ_2 adalah rata-rata dari *read performance OpenStack*

Taraf nyata $\alpha = 0{,}05$

Statistik uji: $T = \frac{\left(\bar{X}_1 - \bar{X}_2\right) - d_0}{\sqrt{S_1^2/n_1 + S_2^2/n_2}} = \frac{(121{,}027 - 15{,}114) - 0}{\sqrt{56{,}342/30 + 0{,}096/30}} = \frac{(105{,}913) - 0}{\sqrt{1{,}878 + 0{,}0032}} =$

$\frac{105{,}913}{\sqrt{1{,}8812}} = \frac{105{,}913}{1{,}371} = 77{,}220$

Daerah kritis : tolak H_0 jika $T < -T_{\alpha/2;v}$ dan $T > T_{\alpha/2;v}$

dimana $v = \frac{(S_1^2/n_1 + S_2^2/n_2)^2}{\frac{(S_1^2/n_2)^2}{n_1 - 1} + \frac{(S_2^2/n_2)^2}{n_2 - 1}} = \frac{(56{,}342/30 + 0{,}096/30)^2}{\frac{(56{,}342/30)^2}{30-1} + \frac{(0{,}096/30)^2}{30-1}} =$

$\frac{(1{,}878 + 0{,}0032)^2}{\frac{(1{,}878)^2}{29} + \frac{(0{,}0032)^2}{29}} = \frac{(1{,}8812)^2}{\frac{3{,}527}{29} + \frac{0{,}00001024}{29}} = \frac{3496{,}712}{\frac{3{,}527}{29}} = \frac{3496{,}712}{0{,}122} = 29{,}007$

$T_{\alpha/2;v} = T_{0{,}05/2;30} = T_{0{,}025;30} = 2{,}042$

sehingga: Tolak H_0 jika $T < -2{,}042$ dan $T > 2{,}042$

Kesimpulan: tolak H_0, hal ini berarti rata-rata kedua data tidak sama. Karena T = 77.220 > 2.042, maka rata-rata μ_1 lebih besar dari pada μ_2.

Hal ini dapat disimpulkan bahwa *read disk performance* pada 512MB *OpenStack* itu lebih baik dari *Eucalyptus*. Hal ini disebabkan karena rata-rata *read disk performance* pada 512MB *OpenStack* lebih besar dari pada *Eucalyptus*.

2. *Write Performance*

Berikut ini adalah perhitungan uji rata-rata terhadap data hasil *benchmarking disk* (*harddisk*) pada *write performance* dari *Eucalyptus*:

Tabel 4.13 Perhitungan Data *Performance Write Performance Eucalyptus*

Size	512		
Disk Test	Write	Xi - Xbar	(Xi - Xbar)^2
1	580,800	4,443	19,740
2	514,360	-61,997	3843,628
3	583,560	7,203	51,883
4	588,330	11,973	143,353
5	577,620	1,263	1,595
6	560,765	-15,592	243,110
7	543,910	-32,447	1052,808
8	559,735	-16,622	276,291
9	575,560	-0,797	0,635
10	541,080	-35,277	1244,467
11	562,270	-14,087	198,444
12	583,460	7,103	50,453
13	595,740	19,383	375,701
14	586,200	9,843	96,885
15	589,668	13,311	177,169
16	593,135	16,778	281,501
17	596,603	20,246	409,880
18	600,070	23,713	562,306
19	586,820	10,463	109,474
20	607,410	31,053	964,289
21	579,228	2,870	8,240
22	576,280	-0,077	0,006
23	545,150	-31,207	973,877
24	551,045	-25,312	640,697
25	556,940	-19,417	377,020
26	562,835	-13,522	182,844
27	568,730	-7,627	58,171
28	599,670	23,313	543,496
29	630,610	54,253	2943,388
30	593,140	16,783	281,669
Jumlah	17290,723	0,013	16113,021
Rata-rata	576,357	0,000	537,101
		S^2	555,621

Berikut ini adalah perhitungan uji rata-rata terhadap data hasil *benchmarking disk* (*harddisk*) pada *write performance* dari *OpenStack*:

Tabel 4.14 Perhitungan Data *Performance Write Performance OpenStack*

Size	512		
Disk Test	Write	Xi - Xbar	(Xi - Xbar)^2
1	112,920	2,158	4,657
2	107,390	-3,372	11,370
3	111,280	0,518	0,268
4	109,930	-0,832	0,692
5	108,130	-2,632	6,927
6	116,460	5,698	32,467
7	108,260	-2,502	6,260
8	111,840	1,078	1,162
9	110,740	-0,022	0,000
10	113,220	2,458	6,042
11	107,400	-3,362	11,303
12	115,220	4,458	19,874
13	105,360	-5,402	29,182
14	114,140	3,378	11,411
15	107,930	-2,832	8,020
16	107,570	-3,192	10,189
17	112,880	2,118	4,486
18	108,360	-2,402	5,770
19	110,180	-0,582	0,339
20	112,770	2,008	4,032
21	111,190	0,428	0,183
22	116,320	5,558	30,891
23	109,130	-1,632	2,663
24	104,870	-5,892	34,716
25	109,860	-0,902	0,814
26	105,760	-5,002	25,020
27	113,730	2,968	8,809
28	114,460	3,698	13,675
29	112,180	1,418	2,011
30	113,370	2,608	6,802
Jumlah	3322,850	-0,010	300,035
Rata-rata	110,762	0,000	10,001
		S^2	10,346

Uji variansi:

$H_0 : \sigma_1^2 = \sigma_2^2$, hal ini berarti kedua data bersifat *homogen*.

$H_1 : \sigma_1^2 \neq \sigma_2^2$, hal ini berarti kedua data bersifat *heterogen*.

Taraf nyata $\alpha = 0,05$

Statistika uji : $F = \frac{S_1^2}{S_2^2} = \frac{555,621}{10,346} = 53,704$

Dimana $S^2 = \frac{\Sigma(X-\bar{X})^2}{n-1}$

Daerah kritis : tolak H_0, jika $F < F_{1-\alpha/2;(V1,V2)}$ dan $F > F_{\alpha/2;(V1,V2)}$

dimana $F_{1-\alpha/2;(V1,V2)} = F_{1-0,05/2;(29,29)} = F_{1-0,025;(29,29)} = F_{0,975;(29,29)} = \frac{1}{F_{0,025;(29,29)}} = \frac{1}{2,07} = 0,483$

$F_{\alpha/2;(V1,V2)} = F_{0,05/2;(29,29)} = F_{0,025;(29,29)} = 2,07$

Sehingga tolak H_0 jika $F < 0,483$ dan $F > 2,07$

Kesimpulan: tolak H_0, karena $F = 53,704 > 2,07$. hal ini berarti kedua data tersebut bersifat *heterogen*. Karena itu dilakukan uji rata-rata terhadap data *write performance* sebagai berikut:

Uji rata-rata:

$H_0 : \mu_1 = \mu_2$, hal ini berarti rata-rata kedua data sama

$H_1 : \mu_1 \neq \mu_2$, hal ini berarti rata-rata kedua data tidak sama

dimana μ_1 adalah rata-rata dari *write performance Eucalyptus* dan μ_2 adalah rata-rata dari *write performance OpenStack*.

Taraf nyata $\alpha = 0,05$

Statistik uji: $T = \frac{\left(\bar{X}_1 - \bar{X}_2\right) - d_0}{\sqrt{S_1^2/n_1 + S_2^2/n_2}} = \frac{(576{,}357 - 110{,}762) - 0}{\sqrt{555{,}621/30 + 10{,}346/30}} = \frac{(465{,}595) - 0}{\sqrt{\frac{565{,}967}{30}}}$

$= \frac{465{,}595}{\sqrt{18{,}865}} = \frac{465{,}595}{4{,}343} = 107{,}195$

Daerah kritis : tolak H_0 jika $T < -T_{\alpha/2;v}$ dan $T > T_{\alpha/2;v}$

dimana $v = \frac{(S_1^2/n_1 + S_2^2/n_2)^2}{\frac{(S_1^2/n_2)^2}{n_1 - 1} + \frac{(S_2^2/n_2)^2}{n_2 - 1}} = \frac{(555{,}621/30 + 10{,}346/30)^2}{\frac{(555{,}621/30)^2}{30-1} + \frac{(10{,}346/30)^2}{30-1}} =$

$\frac{(\frac{565{,}967}{30})^2}{\frac{(18{,}521)^2}{29} + \frac{(0{,}345)^2}{29}} = \frac{(18{,}865)^2}{\frac{343{,}03}{29} + \frac{0{,}119}{29}} = \frac{355{,}888}{\frac{343{,}149}{29}} = \frac{355{,}888}{11{,}833} = 30{,}076$

$T_{\alpha/2;v} = T_{0{,}05/2;30} = T_{0{,}025;30} = 2{,}042$

sehingga: Tolak H_0 jika $T < -2{,}042$ dan $T > 2{,}042$

Kesimpulan: tolak H_0, hal ini berarti rata-rata kedua data tidak sama. Karena T = 107.195 > 2.042, maka rata-rata μ_1 lebih besar dari pada μ_2.

Hal ini dapat disimpulkan bahwa *write disk performance Eucalyptus* itu lebih baik dari *OpenStack*. Hal ini disebabkan karena rata-rata *write disk performance Eucalyptus* lebih besar dari pada *OpenStack*.

4.5.3 Perhitungan Statistika Data *Performance* Pada *Processor*

A. Prosedur Perhitungan

1. Menjumlahkan hasil dari *benchmarking processor* ($\sum$)
2. Mencari rata-rata dari hasil penjumlahan *processor* ($\bar{X}$)
3. Mengurangkan X_i dengan rata-rata ($\bar{X}$) dari penjumlahan *benchmarking processor*
4. Mengkuadratkan hasil dari pengurangan X_i dengan rata-rata dari selisih angka. $(X_i - \bar{x})^2$
5. Mencari pendugaan titik (S^2)
6. Menghitung variansi (F)
7. Mencari uji rataan (T)

B. Hasil Perhitungan

Untuk dapat mengambil keputusan mana *cloud* yang lebih baik antara *Eucalyptus* dan *OpenStack* dari sisi *processor*. Penulis melakukan perhitungan uji rata-rata berikut untuk membandingkan keunggulan dari masing-masing.

Berikut ini adalah perhitungan uji rata-rata terhadap data hasil *benchmarking processor* dari *Eucalyptus*:

Tabel 4.15 Perhitungan Data *Performance Processor Eucalyptus*

C-Ray	Processor	Xi - Xbar	(Xi - Xbar)^2
1	377,37	0,13	0,02
2	377,40	0,16	0,03
3	377,36	0,12	0,02
4	377,43	0,19	0,04
5	377,36	0,12	0,01
6	377,28	0,04	0,00
7	377,26	0,01	0,00
8	377,23	-0,01	0,00
9	377,26	0,02	0,00
10	377,25	0,01	0,00
11	377,04	-0,20	0,04
12	377,19	-0,06	0,00
13	377,33	0,09	0,01
14	377,54	0,30	0,09
15	377,35	0,11	0,01
16	377,16	-0,08	0,01
17	377,17	-0,07	0,00
18	377,09	-0,15	0,02
19	377,13	-0,11	0,01
20	377,11	-0,13	0,02
21	377,12	-0,12	0,01
22	377,12	-0,13	0,02
23	377,19	-0,05	0,00
24	377,15	-0,09	0,01
25	377,17	-0,07	0,00
26	377,18	-0,06	0,00
27	377,16	-0,08	0,01
28	377,15	-0,09	0,01
29	377,30	0,06	0,00
30	377,23	-0,01	0,00
Jumlah	11317,058	-0,143	0,393
Rata-rata	377,235	-0,005	0,013
		S^2	0,014

Berikut ini adalah perhitungan uji rata-rata terhadap data hasil *benchmarking processor* dari *OpenStack*:

Tabel 4.16 Perhitungan Data *Performance Processor OpenStack*

C-Ray	Processor	Xi - Xbar	(Xi - Xbar)^2
1	747,980	0,000	0,000
2	747,820	-0,160	0,026
3	747,910	-0,070	0,005
4	748,160	0,180	0,032
5	747,850	-0,130	0,017
6	747,630	-0,350	0,123
7	747,900	-0,080	0,006
8	747,870	-0,110	0,012
9	748,100	0,120	0,014
10	748,010	0,030	0,001
11	747,810	-0,170	0,029
12	747,920	-0,060	0,004
13	747,700	-0,280	0,078
14	748,050	0,070	0,005
15	748,170	0,190	0,036
16	748,060	0,080	0,006
17	748,060	0,080	0,006
18	747,780	-0,200	0,040
19	748,160	0,180	0,032
20	748,020	0,040	0,002
21	748,000	0,020	0,000
22	748,100	0,120	0,014
23	748,360	0,380	0,144
24	747,910	-0,070	0,005
25	748,130	0,150	0,022
26	748,120	0,140	0,020
27	747,910	-0,070	0,005
28	747,800	-0,180	0,032
29	748,170	0,190	0,036
30	747,870	-0,110	0,012
Jumlah	22439,330	-0,070	0,766
Rata-rata	747,978	-0,002	0,026
		S^2	0,026

Uji variansi:

$H_0 : \sigma_1^2 = \sigma_2^2$, hal ini berarti kedua data bersifat *homogen*.

$H_1 : \sigma_1^2 \neq \sigma_2^2$, hal ini berarti kedua data bersifat *heterogen*.

Taraf nyata $\alpha = 0{,}05$

Statistika uji : $F = \frac{S_1^2}{S_2^2} = \frac{0{,}026}{0{,}014} = 1{,}857$

dimana $S^2 = \frac{\sum(X-\bar{X})^2}{n-1}$

Daerah kritis : tolak H_0, jika $F < F_{1-\alpha/2;(V1,V2)}$ dan $F > F_{\alpha/2;(V1,V2)}$

dimana $F_{1-\alpha/2;(V1,V2)} = F_{1-0{,}05/2;(29,29)} = F_{1-0{,}25;(29,29)} = F_{0{,}975;(29,29)} = \frac{1}{F_{0{,}025;(29,29)}} = \frac{1}{2{,}07} = 0{,}483$

$F_{\alpha/2;(V1,V2)} = F_{0{,}05/2;(29,29)} = F_{0{,}025;(29,29)} = 2{,}07$

Sehingga tolak H_0 jika $F < 0{,}483$ dan $F > 2{,}07$

Kesimpulan: terima H_0, karena $F = 1{,}857 < 2{,}07$. hal ini berarti kedua data tersebut bersifat *homogen*. Karena itu dilakukan uji rata-rata terhadap data *processor* sebagai berikut:

Uji rata-rata:

$H_0 : \mu_1 = \mu_2$, hal ini berarti rata-rata kedua data sama

$H_1 : \mu_1 \neq \mu_2$, hal ini berarti rata-rata kedua data tidak sama

dimana μ_1 adalah rata-rata dari *processor OpenStack* dan μ_2 adalah rata-rata dari *processor Eucalyptus*.

Taraf nyata $\alpha = 0{,}05$

Statistik uji: $T = \frac{(\bar{X}_1 - \bar{X}_2) - d_0}{S_p\sqrt{1/n_1 + 1/n_2}}$

$S_p^2 = \frac{(n_1-1)S_1^2 + (n_2-1)S_2^2}{n_1+n_2-2} = \frac{(30-1)(0{,}026)+(30-1)(0{,}014)}{30+30-2} = \frac{(29)(0{,}026)+(29)(0{,}014)}{58} = \frac{0{,}754+0{,}406}{58} = \frac{1{,}16}{58} = 0{,}02$

$S_p = \sqrt{0{,}02} = 0{,}141$

Sehingga: $T = \frac{(747{,}978-377{,}235)-0}{0{,}141\sqrt{1/30+1/30}} = \frac{(370{,}743)-0}{0{,}141\sqrt{2/30}} = \frac{370{,}743}{(0{,}141)\sqrt{0{,}0667}} =$

$\frac{370{,}743}{(0{,}141).(0{,}258)} = \frac{370{,}743}{0{,}036378} = 10191{,}407$

Daerah kritis : tolak H_0 jika $T < -T_{\alpha/2;v}$ dan $T > T_{\alpha/2;v}$

dimana v = n1 + n2 – 2 = 30 + 30 – 2 = 58

$T_{\alpha/2;v} = T_{0{,}05/2;58} = T_{0{,}025;58} = 2{,}002$

sehingga: Tolak H_0 jika $T < -2{,}002$ dan $T > 2{,}002$

Kesimpulan: tolak H_0, hal ini berarti rata-rata kedua data tidak sama. Karena T = 10191.407 > 2.002, maka rata-rata μ_1 lebih besar dari pada μ_2.

Hal ini dapat disimpulkan bahwa *performance processor OpenStack* itu lebih baik dari *Eucalyptus*. Hal ini disebabkan karena rata-rata *performance processor OpenStack* lebih besar dari pada *Eucalyptus*.