

# ANALISIS PERBEDAAN PERFORMANSI SISTEM OPERASI SERVER FREEBSD 8.2 DAN FEDORA 15 PADA DATABASE MYSQL 5.5

Ari Nugroho (ablehkaka@gmail.com)  
 Bebas Widada (bbswdd@yahoo.com)  
 Wawan Laksito YS (wlaksito@yahoo.com)

## ABSTRAK

Pemilihan sistem operasi server diperlukan untuk mengoptimalkan kinerja dari aplikasi yang menggunakan database. Penelitian ini bertujuan untuk menguji perbedaan kinerja sistem operasi server antara FreeBSD 8.2 dengan Fedora 15 dengan metode uji independent samples test. Perubahan konfigurasi pada file sysctl.conf digunakan untuk optimasi parameter kernel sistem operasi pada saat berjalan dan file my.cnf digunakan untuk membagi sumber daya memori (RAM) untuk menjalankan perintah menyesuaikan kebutuhan yang ada dengan kapasitas RAM yang digunakan. Pengukuran kinerja dilakukan menggunakan aplikasi Sysbench dengan memberi 20 syntax yang berbeda dimana perbedaan terletak pada pemberian nilai num-threads. Nilai tersebut diasumsikan sebagai banyaknya jumlah klien yang melakukan perintah kepada server. Sampel data yang dianalisa mengambil nilai rata-rata waktu eksekusi setiap perintah pada hasil pengukuran yang dilakukan dengan aplikasi Sysbench. Untuk mengetahui kenormalan distribusi sampel data dilakukan pengujian dengan metode uji normalitas Kolmogorov-Smirnov. Dan untuk mengetahui homogenitas varian sampel data dilakukan dengan metode uji homogenitas Lavene. Pengujian hipotesis dilakukan dengan metode uji independent samples t-test. Uji ini digunakan untuk mengetahui ada atau tidaknya perbedaan rata-rata antara dua kelompok sampel yang tidak berhubungan. Hasil dari penelitian diketahui bahwa nilai rata-rata waktu setiap eksekusi perintah pada sistem operasi server Fedora 15 lebih tinggi dibanding FreeBSD 8.2, dengan selisih sebesar 0,346 detik, akan tetapi hal itu masih masuk dalam nilai toleransi (taraf kepercayaan/Confidence Interval of the Difference = 95%), sehingga dapat disimpulkan bahwa nilai rata-rata waktu setiap eksekusi perintah pada kedua sistem operasi adalah sama (tidak ada perbedaan yang signifikan).

Kata kunci : server, optimasi, kinerja, uji

## I. PENDAHULUAN

### 1.1 Latar Belakang

Pada umumnya perusahaan atau instansi dalam kegiatan operasionalnya masih menggunakan sistem operasi server yang mereka kenal saja, tanpa mau mencari sistem operasi yang lebih tepat digunakan untuk aplikasi yang digunakan pada perusahaan/instansi tersebut. Performa server yang handal sangat dibutuhkan untuk menunjang aplikasi yang membutuhkan penyimpanan database. Untuk meningkatkan performa komputer tidak hanya dilakukan dengan meningkatkan spesifikasi perangkat kerasnya, akan tetapi dapat juga dilakukan dengan merubah konfigurasi bawaan menjadi konfigurasi yang dapat disesuaikan dengan kebutuhan. Berdasarkan hal tersebut, maka dibutuhkan analisa perbandingan perbedaan performansi antara dua sistem operasi server, yaitu server FreeBSD 8.2 dengan server Fedora 15 terhadap database MySQL 5.5 yang diinstal didalamnya. Dimana kedua

sistem operasi server tersebut sangat populer dan banyak dipakai pada saat ini. FreeBSD 8.2 sebagai perwakilan dari platform unix, sedangkan Fedora 15 sebagai perwakilan dari platform linux

### 1.2 Perumusan Masalah

Perumusan masalah penelitian ini apakah ada perbedaan yang signifikan pada rata-rata waktu eksekusi setiap perintah antara sistem operasi server FreeBSD 8.2 dan Fedora 15.

### 1.3 Pembatasan Masalah

Pembatasan masalah dalam melakukan penelitian ini adalah:

1. Dalam pembangunan server menggunakan sistem operasi server FreeBSD 8.2 dan server Fedora 15 dengan database MySQL 5.5 didalamnya.
2. Penelitian dilakukan dengan menggunakan fitur yang dimiliki oleh software Sysbench, yaitu pengukuran

kinerja server database (OLTP / *Online Transaction Processing benchmark*).

- Analisa data dilakukan dengan mengambil data dari Sysbench dan dianalisa menggunakan pengujian *independent sample t-test*.

#### 1.4 Tujuan

Tujuan dari penelitian ini adalah melakukan pengujian kinerja dengan membuat analisa perbandingan rata-rata waktu eksekusi setiap perintah antara sistem operasi server FreeBSD 8.2 dan Fedora 15 pada database MySQL 5.5, menggunakan aplikasi Sysbench sebagai alat pengukur dengan fitur pengujian kinerja server database MySQL (OLTP / *Online Transaction Processing benchmark*).

#### 1.5 Manfaat

- Manfaat teoritis  
Hasil dari penelitian ini dapat digunakan sebagai referensi dan wacana keilmuan untuk mengembangkan penelitian terkait dengan faktor yang menjadi prioritas utama dan berpengaruh terhadap kinerja (performa) server.
- Manfaat praktis  
Penelitian dan analisis data perbedaan sistem operasi server ini diharapkan dapat memberikan kemudahan dalam menentukan pilihan sistem operasi server *open source* yang tepat untuk digunakan.

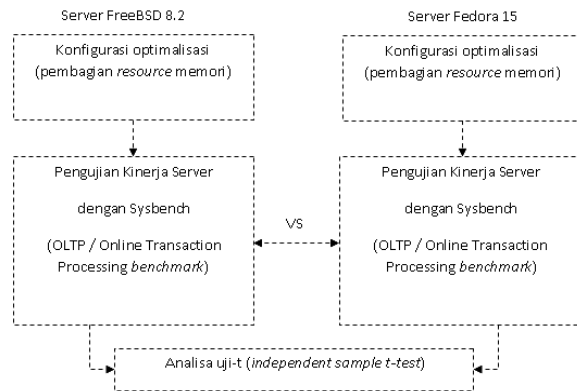
#### 1.6 Hipotesis

Penelitian ini menggunakan hipotesis sebagai berikut:

- Ho : Tidak ada perbedaan kinerja server database (OLTP / *Online Transaction Processing benchmark*) antara sistem operasi server FreeBSD 8.2 dan Fedora 15.
- Ha : Terdapat perbedaan kinerja server database (OLTP / *Online Transaction Processing benchmark*) antara sistem operasi server FreeBSD 8.2 dan Fedora 15.

#### 1.7 Kerangka Pikir

Kerangka pikir dapat dilihat seperti pada gambar 1.



Gambar 1. Kerangka Pikiran

## II. METODE PENELITIAN

Metodologi penelitian merupakan tahap-tahap yang dilalui peneliti mulai dari perumusan masalah sampai kesimpulan yang membentuk sebuah alat yang sistematis. Metode penelitian ini sangat diperlukan untuk mendapatkan data yang tepat dan akurat demi kesempurnaan hasil.

### 2.1 Langkah Penelitian

Langkah-langkah dalam penelitian ini adalah sebagai berikut:

- Mempersiapkan alat dan lingkungan penelitian. Pada tahap ini semua hardware yang dibutuhkan dirangkai.
- Melakukan setting konfigurasi optimasi pada file **sysctl.conf** dan file **my.cnf** dimasing-masing sistem operasi. Konfigurasi yang diberikan untuk masing-masing sistem operasi adalah sama.
- Melakukan proses pengumpulan data dengan menggunakan aplikasi Sysbench sebagai alat ukurnya. Perintah yang diberikan sebanyak 20 *syntax*, dengan perbedaan pada "*num-threads*" yang diasumsikan sebagai jumlah dari klien yang melakukan *request* kepada server.
- Melakukan pengujian data menggunakan analisa kuantitatif uji *independent samples test*. Untuk melakukan uji tersebut harus melalui pengujian asumsi klasik, yaitu uji normalisasi dengan menggunakan metode uji Kolmogorov-Smirnov dan uji homogenitas dengan menggunakan metode uji Lavene.
- Menarik kesimpulan dari hasil pengujian terhadap hipotesa yang telah dibuat dengan menggunakan kaidah-kaidah yang berlaku. Pada tahap ini menentukan apakah hipotesa dapat diterima atau ditolak.

## 2.2 Bahan dan Materi

Bahan dan materi penelitian yang tersedia di dalam penelitian berupa data yang dikeluarkan oleh software sysbench yang diperoleh dengan melakukan beberapa macam pengujian pemberian beban yang berbeda-beda kapasitasnya pada MySQL 5.5. Nilai yang diambil untuk dianalisa adalah nilai rata-rata waktu eksekusi setiap perintah pada hasil pengukuran yang dilakukan dengan menggunakan aplikasi Sysbench.

## 2.3 Prosedur Penelitian

Prosedur penelitian adalah langkah-langkah yang digunakan sebagai alat untuk mengumpulkan data dan menjawab pertanyaan-pertanyaan dalam penelitian. Prosedur penelitian ini menggunakan metode-metode sebagai berikut:

### 1. Pembangunan server

Pembangunan server merupakan kegiatan persiapan lingkungan penelitian. Dalam pembangunan server ini menggunakan sistem operasi server FreeBSD 8.2 dan Fedora 15 server dimana setiap sistem operasi server di instal database MySQL 5.5 sebagai penyimpanan database serta Sysbench Benchmark Tool sebagai software pengujinya. Kedua sistem operasi diatas dibangun pada *personal computer* dengan spesifikasi hardware yang sama, dengan setting konfigurasi optimalisasi yang sama pula, sehingga akan lebih adil dalam penerapan pengujiannya. Pengujian dilakukan secara bergantian antara dua sistem operasi server tersebut dengan cara mengganti hardisk yang telah terinstal masing-masing sistem operasi server.

### 2. Metode pengumpulan data

Dalam penelitian analisa perbandingan performansi sistem operasi server ini, metode pengumpulan data yang digunakan adalah :

#### a. Metode Kepustakaan.

Metode pengumpulan data dengan cara membaca buku-buku, jurnal ilmiah di internet, dan lain-lain untuk mendapatkan dasar-dasar teori yang diperlukan dalam penelitian dan penyusunan skripsi.

#### b. Metode Sysbench Benchmark Tool

Metode pengumpulan data dengan cara melakukan simulasi beban kerja server dengan menggunakan aplikasi Sysbench.

Tabel 1. Pengujian Beban pada Sysbench

No	Jumlah Threads	No	Jumlah Threads
1	1	11	450
2	10	12	500
3	50	13	550
4	100	14	600
5	150	15	650
6	200	16	700
7	250	17	750
8	300	28	800
9	350	19	850
10	400	20	900

## III. TINJAUAN PUSTAKA

Menggunakan konfigurasi default yang disediakan dengan MYSQL, itu memakan memori di atas 140MB. Setelah beberapa *tweak*, penggunaan memori hanya 45MB, itu dapat menghemat 100MB. Untuk meringankan beban, perlu melakukan beberapa *tweaking*. Pertama menonaktifkan dua dari server modul MYSQL menggunakan - InnoDB dan BDB. Jadi pertama yang harus dilakukan adalah memasuki SSH dan mengedit file konfigurasi MYSQL di / etc / mysql / my.cnf.[1]

Setelan konfigurasi bawaan sistem operasi belum tentu sesuai dengan apa yang dibutuhkan. Sehingga untuk mendapatkan server yang optimal dibutuhkan perubahan konfigurasi *default setting* yang ada. Hal terpenting sebelum melakukan optimasi dan *compile* kernel adalah mengetahui semua perangkat yang ada pada mesin server atau komputer serta memastikan apakah perangkat tersebut benar-benar dibutuhkan.[2]

Teknik yang paling umum untuk mengukur kinerja database adalah mengambil pendekatan *black box* yang mengukur Transaksi Per Second (TPS) sebuah aplikasi yang mampu mengeksekusi terhadap database. Dalam skenario ini "Transaksi" adalah unit eksekusi aplikasi klien memanggil terhadap database.[3]

Menurut Alexey Kopytov [4] Sysbench Benchmark Tool adalah perangkat lunak yang digunakan untuk mengukur performa sebuah sistem operasi. Sysbench adalah sebuah modular, *cross platform*, *multi-thread tool* benchmark yang digunakan untuk mengevaluasi parameter sistem operasi yang penting pada sistem database yang berjalan dibawah beban intensif. Desain sysbench adalah dengan menjalankan beberapa jumlah

*thread* yang spesifik dan mereka mengeksekusi *request* secara parallel.

Sysbench adalah salah satu aplikasi yang tersedia untuk umum dan biasanya digunakan untuk pengukuran (*benchmarking*) di MySQL Engineering.[5]

#### IV. HASIL DAN PEMBAHASAN

##### 4.1 Setting Konfigurasi Optimasi Sistem Operasi Server

Setelan konfigurasi bawaan sistem operasi belum tentu sesuai dengan apa yang dibutuhkan. Sehingga untuk mendapatkan server yang optimal dibutuhkan perubahan konfigurasi *default setting* yang ada.

1. Mengedit file **/etc/sysctl.conf** yang berguna untuk konfigurasi parameter kernel pada saat runtime. Dengan perintah: **nano /etc/sysctl.conf**. Kemudian menambahkan baris berikut :

**security.bsd.see\_other\_uids=0**

#Digunakan untuk tidak mengizinkan seorang user melihat proses user lain

**net.inet.tcp.sendspace=65536**

#Digunakan untuk mensetting ukuran default pengiriman TCP

**net.inet.tcp.recvspace=65536**

#Digunakan untuk mensetting ukuran default penerimaan TCP

**kern.ipc.somaxconn=2048**

#Digunakan untuk membatasi ukuran antrian koneksi TCP baru

**net.inet.icmp.drop\_redirect=1**

#Digunakan untuk mengaktifkan hapus pesan kesalahan

**net.inet.icmp.log\_redirect=0**

#Digunakan untuk menonaktifkan rekaman pesan kesalahan

**net.inet.ip.redirect=0**

#Digunakan untuk menonaktifkan pengalihan IP

**kern.maxfiles=112328**

#Digunakan untuk membatasi jumlah maksimal file deskriptor yang terbuka

**kern.maxfilesperproc=31095**

#Digunakan untuk membatasi jumlah maksimal file deskriptor yang terbuka per proses

**net.inet.tcp.blackhole=1**

#Digunakan untuk mengaktifkan paket tcp pada port tertutup di cabut

**net.inet.udp.blackhole=1**

#Digunakan untuk mengaktifkan paket udp pada port tertutup di cabut

**net.inet.icmp.icmplim\_output=0**

#Digunakan untuk menonaktifkan mengontrol jumlah maksimum icmp *unreachable* dan paket rst tcp yang dikirim kembali setiap detik

**net.inet.tcp.msl=3000**

#Digunakan untuk mengatur waktu tunggu ack dalam reply ke syn-ack

**vfs.read\_max=64**

#Digunakan untuk menentukan batas maksimum VFS (*Virtual File System*) guna peningkatan kinerja *hardware* dalam pembacaan database

**vfs.lorunningspace=6291456**

#Digunakan untuk menentukan berapa banyak antrian penulisan I/O untuk pengontrol disk seluruh sistem pada saat jalur (*traffict*) longgar (rendah)

**vfs.hirunningspace=12388608**

#Digunakan untuk menentukan berapa banyak antrian penulisan I/O untuk pengontrol disk seluruh sistem pada saat jalur (*traffict*) padat (tinggi)

**vfs.ufs.dirhash\_maxmem=38554432**

#Digunakan untuk menentukan alokasi sumber daya memori (*resource memory*) maksimum untuk mengakses file yang besar

2. Mengedit file **my.cnf** digunakan untuk membagi sumber daya (*resource memory*) dari RAM (*Random Access Memory*) untuk menjalankan perintah (*request*) menyesuaikan kebutuhan yang ada dengan kapasitas RAM yang digunakan. Menambahkan baris berikut:

**[client]**

**#password = your\_password**

#Digunakan untuk menentukan password (di setting nonaktif)

**port = 3306**

#Digunakan untuk menentukan port yang digunakan mysql

**socket = /tmp/mysql.sock**

#Digunakan untuk menentukan lokasi mysql socket

**[mysqld]**

**port = 3306**

#Digunakan untuk menentukan port yang digunakan mysql

**socket = /tmp/mysql.sock**

#Digunakan untuk menentukan lokasi mysql socket

**key\_buffer = 175M**

#Digunakan untuk menentukan alokasi sumber daya memori (*resource memory*) untuk menjalankan proses perintah sql

**join\_buffer = 16M**

#Digunakan untuk menentukan alokasi sumber daya memori (*resource memory*) untuk menjalankan proses penggabungan buffer

**#record\_buffer = 2M**

**sort\_buffer = 5M**

#Digunakan untuk menentukan alokasi sumber daya memori (*resource memory*) untuk menjalankan proses pengurutan myisamchk (*indexing*)

**sort\_buffer\_size = 5M**

#Digunakan untuk menentukan alokasi sumber daya memori (*resource memory*) untuk menjalankan proses pengurutan myisamchk (*indexing*)

**net\_buffer\_length = 1M**

#Digunakan untuk menentukan alokasi sumber daya memori (*resource memory*) untuk menentukan panjang maksimum perintah (*query*)

**read\_buffer\_size = 5M**

#Digunakan untuk menentukan alokasi sumber daya memori (*resource memory*) untuk menjalankan proses pembacaan

**read\_rnd\_buffer\_size = 2M**

#Digunakan untuk menentukan alokasi sumber daya memori (*resource memory*) untuk menjalankan proses *sorting*

**table\_cache = 2000**

#Digunakan untuk menentukan alokasi sumber daya memori (*resource memory*) untuk menentukan jumlah maksimum *cache* dalam mengakses tabel mysql

**max\_allowed\_packet = 160M**

#Digunakan untuk menentukan alokasi sumber daya memori (*resource memory*) untuk menentukan kapasitas maksimum penyimpanan data

**#skip-networking**

**server-id = 1**

#Digunakan untuk menjalankan (*mengaktifkan*) replikasi *server slave*

**#skip-name-resolve**

**bulk\_insert\_buffer\_size = 20M**

#Digunakan untuk menentukan alokasi sumber daya memori (*resource memory*) untuk menentukan kapasitas proses *insert* data yang besar

**connect\_timeout = 10**

#Digunakan untuk menentukan batas waktu putus koneksi

**max\_connections = 1000**

#Digunakan untuk menentukan batas koneksi *client*

**max\_connect\_errors = 1000**

#Digunakan untuk menentukan batas kesalahan koneksi

**max\_heap\_table\_size = 228M**

#Digunakan untuk menentukan alokasi sumber daya memori (*resource memory*) untuk menentukan kapasitas maksimum tumpukan antrian akses pada tabel

**myisam\_sort\_buffer\_size = 64M**

#Digunakan untuk menentukan alokasi sumber daya memori (*resource memory*) untuk menentukan kapasitas proses perintah pengurutan pada myisam

**default\_time\_zone = +7:00**

#Digunakan untuk menentukan zona/area waktu

**open\_files\_limit = 8192**

#Digunakan untuk menentukan batas file deskriptor untuk mysqld

**query\_cache\_type = 1**

#Digunakan untuk mengaktifkan *query chase*

**query\_cache\_limit = 4M**

#Digunakan untuk menentukan batas ukuran setiap *query chase*

**query\_cache\_size = 128M**

#Digunakan untuk menentukan alokasi sumber daya memori (*resource memory*) untuk menentukan kapasitas *query chase*

**query\_prealloc\_size = 16384**

#Digunakan untuk menentukan alokasi sumber daya memori (*resource memory*) untuk menentukan kapasitas alokasi memori selama proses eksekusi *query*

**query\_alloc\_block\_size = 16384**

#Digunakan untuk menentukan alokasi sumber daya memori (*resource memory*) untuk menjalankan fragmentasi memori

**thread\_cache = 32**

#Digunakan untuk menentukan kapasitas maksimum setiap *thread chase*

**thread\_cache\_size = 286**

#Digunakan untuk menentukan kapasitas maksimum untuk menyimpan *thread chase*

**thread\_concurrency = 2**

#Digunakan untuk menentukan jumlah *thread* pada waktu bersamaan

**lower\_case\_table\_names = 1**

#Digunakan untuk menjalankan pemberian nama tabel dengan huruf kecil

**log-error = /var/log/mysql-error.log**

#Digunakan untuk menentukan lokasi penyimpanan file rekaman kesalahan

**slow-query-log-file = /var/log/mysql-slow.log**

#Digunakan untuk menentukan lokasi penyimpanan file rekaman proses eksekusi *query* yang lambat

**[mysqldump]**

**quick**

**max\_allowed\_packet = 16M**

**[mysql]**

**no-auto-rehash**

**# Remove the next comment character if you are not familiar with SQL**

**safe-updates**

**[isamchk]**

**key\_buffer = 64M**

#Digunakan untuk menentukan alokasi sumber daya memori (*resource memory*) untuk menjalankan perintah sql *index blocks*

**sort\_buffer\_size = 64M**

#Digunakan untuk menentukan alokasi sumber daya memori (*resource memory*) untuk menjalankan proses pengurutan data

**read\_buffer = 16M**

#Digunakan untuk menentukan alokasi sumber daya memori (*resource memory*) untuk menjalankan proses pembacaan data

**write\_buffer = 16M**

#Digunakan untuk menentukan alokasi sumber daya memori (*resource memory*) untuk menjalankan proses penulisan data

**[myisamchk]**

#Digunakan untuk mesin penyimpanan (*storage engine*)

**key\_buffer = 64M**

#Digunakan untuk menentukan alokasi sumber daya memori (*resource memory*) untuk menjalankan perintah sql *index blocks*

**sort\_buffer\_size = 64M**

#Digunakan untuk menentukan alokasi sumber daya memori (*resource memory*) untuk menjalankan proses pengurutan data

**read\_buffer = 16M**

#Digunakan untuk menentukan alokasi sumber daya memori (*resource memory*) untuk menjalankan proses pembacaan data

**write\_buffer = 16M**

#Digunakan untuk menentukan alokasi sumber daya memori (*resource memory*) untuk menjalankan proses penulisan data

**# >>TUNNING<<**

**[mysqld\_safe]**

**#safe-updates**

**nice = -5**

**#[mysqlhotcopy]**

**#interactive-timeout**

Kemudian mereboot server untuk mengaktifkan semua konfigurasi dengan perintah:

**reboot**

## 4.2 Pengumpulan Data

Data diperoleh dengan menjalankan aplikasi Sysbench, dengan memberikan *num-thread* yang berbeda-beda dimana mengandung arti banyaknya jumlah klien yang melakukan permintaan (*request*) kepada server. Adapun langkah-langkahnya adalah sebagai berikut:

1. Langkah yang pertama harus dilakukan adalah menjalankan proses **prepare**, yaitu melakukan tindakan persiapan untuk pengujian yang membutuhkan mereka, dalam hal ini menciptakan dan mengisi database test untuk oltp test dengan nama sbtest dan mengisikan catatan (*record*) sebesar 1Mb. Adapun *syntax*-nya adalah:

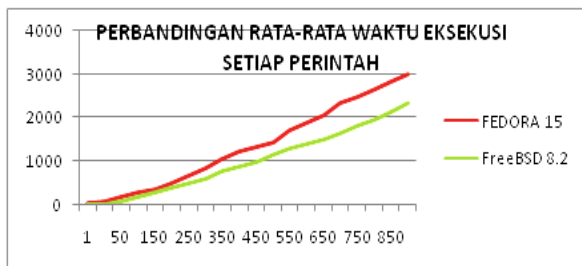
```
sysbench --test=oltp --db-driver=mysql --mysql-db=sbtest --mysql-table-engine=innodb --oltp-table-size=1000000 --mysql-host=127.0.0.1 --mysql-user=root --mysql-password=abl123 --mysql-db=sbtest prepare
```

2. Menjalankan perintah **run** untuk melakukan pengujian *benchmark* yang sebenarnya dengan mode test *complex*, membatasi waktu eksekusi kurang lebih 60 detik, tidak membatasi *request* yang diberikan, jumlah klien bervariasi dan memberikan catatan (*record*) sebesar 1Mb. Perbedaan perintah terletak pada nilai *num-threads*. Adapun *syntax* yang diberikan adalah sebagai berikut:

```
sysbench --test=oltp --oltp-test-mode=complex --db-driver=mysql --init-rng=on --mysql-host=127.0.0.1 --mysql-port=3306 --mysql-db=sbtest --mysql-user=root --mysql-password=abl --max-time=60 --oltp-read-only=off --max-requests=0 --num-threads=(?) --oltp-table-size=1000000 --mysql-table-engine=innodb --mysql-engine-trx=yes run
```

**Tabel 2. Nilai Rata-Rata Waktu Eksekusi Setiap Perintah**

Jumlah Threads	Fedora 15 (ms)	FreeBSD 8.2 (ms)
1	47,46	4,52
10	88,2	20,89
50	172,95	91,47
100	273,78	178,41
150	368,86	286,74
200	496,59	387,74
250	659,47	507,48
300	852,95	592,42
350	1035,7	776,18
400	1210,93	861,1
450	1332,48	986,87
500	1439,9	1164,18
550	1723,72	1286,43
600	1898,27	1414,5
650	2053,78	1519,15
700	2349,2	1643,6
750	2467,36	1818,53
800	2635,51	1960,52
850	2802,92	2136,28
900	2992,15	2340,84



**Gambar 3. Grafik Rata-Rata Waktu Eksekusi**

Dari tabel diatas diambil rata-rata waktu eksekusi setiap perintah pada Fedora 15 adalah 1345,109 ms (1,345 detik) dan pada FreeBSD 8.2 adalah 998,8925 ms (0,998 detik). Sehingga selisih rata-rata waktu diantara keduanya adalah 346,2165 ms (0,346 detik).

#### 4.3 Analisa Data

Data yang telah diperoleh kemudian dianalisa dengan melakukan pengujian:

##### 1. Uji Normalitas

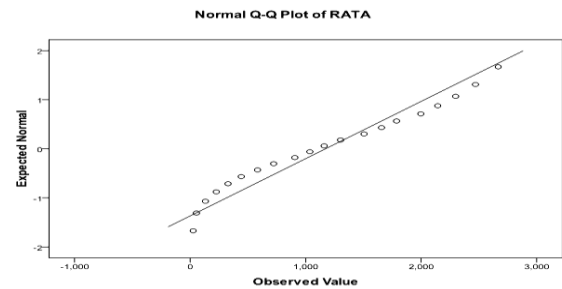
Uji Normalitas digunakan untuk mengetahui apakah suatu data berdistribusi normal atau tidak (Setiaji, 2008: 41). Untuk mengetahui data normal atau tidak salah satunya dengan menggunakan uji kenormalan Kolmogorov Smirnov Test (K-S).

Uji (K-S) dilakukan dengan membuat hipotesis:

Ho : data berdistribusi normal

Ha : data tidak berdistribusi normal

Nilai taraf signifikansi ( $\alpha$ ) = 0,05. Jika nilai signifikansi ( $p$ ) >  $\alpha$  maka Ho diterima, dan jika nilai signifikansi ( $p$ ) <  $\alpha$ , maka Ho ditolak. Hasil uji (K-S) dapat dilihat pada gambar dan tabel dibawah ini.



**Gambar 3. Hasil Uji Normalitas**

**Tabel 3. Hasil Uji Normalitas**

	Kolmogorov-Smirnov <sup>a</sup>			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
RATA	.104	20	.200	.940	20	.242

a. Lilliefors Significance Correction

\*. This is a lower bound of the true significance.

Berdasarkan hasil uji, diperoleh nilai signifikansi ( $p$ ) = 0,200. Hal ini dapat ditarik kesimpulan bahwa Ho diterima, yang berarti data variabel normalitas berdistribusi normal karena nilai signifikansi ( $p$ ) > 0,05.

##### 2. Uji Homogenitas

Uji Homogenitas digunakan untuk mengetahui varian dua populasi sama (homogen) atau tidak (heterogen). Dengan demikian perbedaan yang terjadi dalam hipotesis benar-benar berasal dari perbedaan antara kelompok, bukan akibat dari perbedaan yang terjadi di dalam kelompok.

Uji homogenitas menggunakan hipotesis:

Ho : variansi pada populasi yang sama (homogen)

Ha : variansi pada populasi yang berbeda (heterogen)

Nilai taraf signifikansi ( $\alpha$ ) = 0,05. Jika nilai signifikansi ( $p$ ) >  $\alpha$  maka Ho diterima, dan jika nilai signifikansi ( $p$ ) <  $\alpha$ , maka Ho ditolak.

**Tabel 4. Hasil Uji Homogenitas**

Test of Homogeneity of Variance					
		Levene Statistic	df1	df2	Sig.
Nilai Rata-Rata	Based on Mean	2.117	1	38	.154
	Based on Median	2.030	1	38	.162
	Based on Median and with adjusted df	2.030	1	35.645	.163
	Based on trimmed mean	2.089	1	38	.157

Berdasarkan hasil uji homogenitas varian dengan uji Lavene diatas, diperoleh nilai probabilitas (signifikansi) dengan *equal variance assumed* adalah 0,154 lebih besar dari nilai taraf signifikansi ( $\alpha$ ) = 0,05, maka  $H_0$  diterima, jadi dapat disimpulkan bahwa kedua varian (varian kelompok 1 (Fedora 15) dan kelompok 2 (FreeBSD 8.2) adalah sama).

### 3. Uji Hipotesis

Analisis statistik yang digunakan untuk menguji hipotesis adalah analisa uji *independent samples t-test*. Uji ini digunakan untuk mengetahui ada atau tidaknya perbedaan rata-rata antara dua kelompok sampel yang tidak berhubungan. Jika ada perbedaan, rata-rata manakah yang lebih tinggi. Data yang digunakan biasanya berskala interval atau rasio. Dalam penelitian ini menggunakan hipotesis:

$H_0$  : Tidak ada perbedaan kinerja server database (OLTP / *Online Transaction Processing benchmark*) antara sistem operasi server FreeBSD 8.2 dan server Fedora 15.

$H_a$  : Terdapat perbedaan kinerja server database (OLTP / *Online Transaction Processing benchmark*) antara sistem operasi server FreeBSD 8.2 dan server Fedora 15.

**Tabel 5. Independent Samples Test**

Independent Samples Test				
		Nilai Rata-Rata		
		Equal variances assumed	Equal variances not assumed	
Levene's Test for Equality of Variances	F	2.117		
	Sig.	.154		
t-test for Equality of Means	t	1.265	1.265	
	df	38	35.499	
	Sig. (2-tailed)	.214	.214	
	Mean Difference	346.21650	346.21650	
	Std. Error Difference	273.73624	273.73624	
	95% Confidence Lower Interval of the Difference	-207.93355	-209.21854	
	Upper	900.36655	901.65154	

Dari tabel *Independent Sample T Test* diatas diketahui bahwa nilai  $t_{hitung}$  adalah

1,265. Tabel distribusi t dicari pada  $\alpha = 5\%$  :  $2 = 2,5\%$  (uji 2 sisi) dengan derajat kebebasan (df)  $n-2$  atau  $40-2 = 38$ . Dengan pengujian 2 sisi (signifikansi = 0,025) hasil diperoleh untuk  $t_{tabel}$  sebesar 2,02439 (lihat pada tabel t) atau dapat dicari di Ms Excel dengan cara pada cell kosong ketik =TINV(0,05;38).

Kriteria pengujian  $H_0$  akan diterima jika  $-t_{tabel} < t_{hitung} < t_{tabel}$  dan  $H_0$  akan ditolak jika  $-t_{hitung} < -t_{tabel}$  atau  $t_{hitung} > t_{tabel}$ . Berdasarkan probabilitas  $H_0$  diterima jika  $p\text{ value (Sig.)} > 0,05$  dan  $H_0$  ditolak jika  $p\text{ value (Sig.)} < 0,05$ . Langkah selanjutnya adalah membandingkan  $t_{hitung}$  dengan  $t_{tabel}$  dan probabilitas. Dalam hasil pengujian diatas diperoleh nilai  $t_{hitung} < t_{tabel}$  ( $1,265 < 2,02439$ ) dan  $p\text{ value}$  ( $0,214 > 0,05$ ) maka  $H_0$  diterima. Hal ini berarti bahwa tidak ada perbedaan yang signifikan pada rata-rata waktu setiap eksekusi perintah antara sistem operasi server Fedora 15 dengan sistem operasi server FreeBSD 8.2.

Nilai pada 95% *Confidence Interval of the Difference* adalah rentang perbedaan yang ditoleransi. Pada penelitian ini menggunakan taraf kepercayaan sebesar 95%. Jadi dengan menggunakan taraf kepercayaan 95%, rentang selisih rata-rata waktu setiap eksekusi perintah pada sistem operasi server Fedora 15 dan sistem operasi server FreeBSD 8.2 adalah dari -207,93355 sampai 900,36655. Jadi walaupun terdapat selisih rata-rata pada kedua sistem operasi tersebut yang dapat dilihat pada nilai *Mean Difference* = 346,21650, hal itu masuk dalam rentang perbedaan yang ditoleransi. Sehingga  $H_0$  diterima yang artinya tidak ada perbedaan yang signifikan pada rata-rata waktu setiap eksekusi perintah antara sistem operasi server Fedora 15 dengan sistem operasi server FreeBSD 8.2.

## V. PENUTUP

### 5.1 Kesimpulan

Tidak ada perbedaan yang signifikan pada waktu rata-rata setiap eksekusi perintah antara sistem operasi server Fedora 15 dan FreeBSD 8.2. Dari hasil pengukuran diketahui bahwa nilai rata-rata waktu setiap eksekusi perintah pada sistem operasi server Fedora 15 lebih tinggi dibanding FreeBSD 8.2, dengan selisih sebesar 0,346 detik, akan tetapi hal itu



masih masuk dalam nilai toleransi (taraf kepercayaan/*Confidence Interval of the Difference* = 95%), sehingga dapat disimpulkan bahwa nilai rata-rata waktu setiap eksekusi perintah pada kedua sistem operasi adalah sama (tidak ada perbedaan yang signifikan). Dengan demikian kinerja kedua sistem operasi server tersebut diatas adalah sama.

## 5.2 Saran

Dalam membuat aplikasi software yang menggunakan server, hendaknya juga memperhatikan sistem operasi server yang akan digunakan untuk mendukung aplikasi yang dibuat sehingga dapat berjalan optimal. Merujuk pada hasil penelitian ini, sistem operasi server FreeBSD 8.2 dapat menjadi pilihan utama dibanding Fedora 15 walaupun hanya unggul sedikit lebih cepat pada rata-rata waktu eksekusi setiap perintah.

## DAFTAR PUSTAKA

- [1] Peter, 2009, Fine Tuning MySQL (Reducing Memory Usage), <http://www.zimbio.com/Linux/articles/692/Fine+Tuning+MySQL+Reducing+Memory+Usage>, diakses tanggal 05 November 2012 pukul 10.00 WIB
- [2] Devilian Network, 2009, Optimasi dan Compile Kernel FreeBSD, <http://www.devilian.net/2009/09/optimasi-dan-compile-kernel-freebsd>, diakses tanggal 15 Mei 2011 pukul 19.10 WIB.
- [3] MySQL AB, 2005, A MySQL Technical White Paper: *MySQL Performance Benchmark – Measuring MySQL's Scalability and Throughput*, Cupertino-California.
- [4] Alexey Kopytov, 2004, Sysbench : a system performance benchmark, <http://sysbench.sourceforge.net>, diakses tanggal 27 Februari 2013 pukul 10.00 WIB
- [5] Rich Schwerin, 2010, MySQL 5.5 is GA - interview with Tomas Ulin, *Oracle Magazine*, <http://dev.mysql.com/tech-resources/interviews/thomas-ulin-mysql-55.html>, diakses tanggal 27 Februari 2013 pukul 11.00 WIB