

## BAB VI PENGUJIAN DAN ANALISIS

### 6.1 Alat Pengujian

Untuk melakukan pengujian, penulis menggunakan sebuah perangkat lunak tambahan berupa aplikasi Apache Jmeter. Perangkat lunak ini murni menggunakan Java dan didesain untuk melakukan pengujian beban dan mengukur performa aplikasi *Web Server*. Pengujian performa dapat dilakukan dengan menggunakan sumber daya statis dan dinamis seperti halnya data, *Servlets*, *Perl Scripts*, *Java Objects*, *Database*, *Query*, *FTP server*, dan lainnya. Apache Jmeter dapat digunakan untuk mensimulasikan beban tinggi pada sebuah *server*, jaringan, atau objek yang ingin di uji kemampuannya. Selain itu, aplikasi ini juga dapat menampilkan analisis dalam bentuk gambar dari performa atau perilaku *server*, *Script*, atau objek ketika dilakukan tes beban [NON-12].

Grup *thread* adalah tahap awal yang perlu diperhatikan sebelum melakukan pengujian. Tahap ini penting karena seluruh *controller* dan *samplers* berada di bawah grup *thread*. Unsur lainnya, misalnya *listener*, juga dapat ditempatkan langsung di bawah grup *thread* dan hal ini berlaku untuk seluruh grup *thread*. Grup *thread* mengontrol jumlah *thread* dan kemudian Jmeter yang akan melaksanakan pengujian. Kontrol grup *thread* memungkinkan kita untuk mengatur jumlah *thread*, mengatur periode *ramp-up*, dan mengatur jumlah perulangan eksekusi tes.

Setiap *thread* akan mengeksekusi pengujian secara keseluruhan dan berjalan independen dari *thread* lainnya. Beberapa *thread* yang digunakan dapat mensimulasikan koneksi secara bersamaan untuk aplikasi *server*. Periode *ramp-up* menentukan berapa lama eksekusi Jmeter yang dilakukan dengan jumlah *thread* yang ditentukan. Jika yang digunakan adalah 10 *thread* dan periode *ramp-up* adalah 100 detik, maka JMeter akan menggunakan 100 detik untuk menjalankan 10 *thread* tersebut. Setiap *thread* akan dimulai 10 (100/10) detik setelah *thread* sebelumnya dimulai. Jika ada 30 *thread* dan periode *ramp-up* 120 detik, maka setiap *thread* berturut-turut akan tertunda 4 detik. Untuk mengawali pengujian,

sebaiknya menggunakan nilai yang sama antara *ramp-up* dan jumlah *thread* agar terhindar dari beban kerja yang sangat besar.

## 6.2 Pengujian

Dalam penelitian ini, pengujian dilakukan dengan menggunakan tiga skenario *stress load*. Proses pelaksanaan skenario pengujian mengacu pada tahapan-tahapan yang telah dijelaskan pada Bab III. Untuk mendapatkan data yang lebih akurat, penulis melakukan setiap skenario pengujian sebanyak lima kali percobaan, sedangkan waktu yang diperlukan untuk melakukan satu percobaan adalah sepuluh menit. Dalam rentang waktu sepuluh menit, sistem virtualisasi *server* ini sangat reaktif terhadap perubahan beban kerja CPU HVM satu yang terjadi di setiap menitnya. Sistem virtualisasi *server* ini menggunakan parameter batas maksimal beban kerja CPU HVM satu pada angka 80%. Selain itu, berdasarkan penjelasan mengenai cara penggunaan aplikasi Apache Jmeter sebelumnya, penulis perlu mengontrol grup *thread* untuk melakukan pengujian *stress load* terhadap layanan HTTP dari mesin virtual. Berikut kontrol grup *thread* yang ditentukan penulis:

- Jumlah *Thread* (klien) = 50
- Periode *Ramp-Up* (dalam detik) = 1
- Jumlah Perulangan = Selamanya

Kontrol grup *thread* di atas ditentukan berdasarkan kemampuan ideal sistem virtualisasi *server* ini dalam menangani permintaan layanan HTTP. Pengujian yang diharapkan penulis yaitu Apache Jmeter dapat menghasilkan beban kerja tinggi CPU yang sesuai dengan kapasitas sumber daya komputasi mesin virtual. Dari parameter yang ditentukan di atas, dapat diartikan bahwa terdapat lima puluh permintaan klien yang harus dilayani dalam waktu satu detik dan proses tersebut berjalan seterusnya. Untuk membatasi pengujian *stress load* dalam waktu sepuluh menit, penulis melakukan hal tersebut dengan menjalankan dan memberhentikan aplikasi secara manual.

Seluruh data dari hasil pengujian diambil melalui berkas *log.txt* yang dibuat penulis dan *summary report* dari aplikasi Apache Jmeter. Data yang diambil dari berkas *log.txt* yaitu seluruh beban kerja CPU dari HVM satu, VM

satu, VM dua, dan VM tiga, beserta proses perpindahan mesin virtual yang terjadi, yaitu migrasi dan remigrasi. Migrasi adalah proses perpindahan mesin virtual dari HVM satu menuju HVM dua atau HVM tiga, sedangkan remigrasi adalah proses perpindahan mesin virtual dari HVM dua atau HVM tiga menuju HVM satu. Untuk data yang diambil dari *summary report* aplikasi Apache Jmeter yaitu jumlah *samples* yang dikirim oleh klien dari setiap percobaan. Seluruh data yang diperoleh kemudian dianalisis dan selanjutnya digunakan untuk menarik kesimpulan dan saran. Berikut penjelasan detail dari hasil pengujian serta proses analisis yang dilakukan pada pengujian skenario satu, skenario dua, dan skenario tiga.

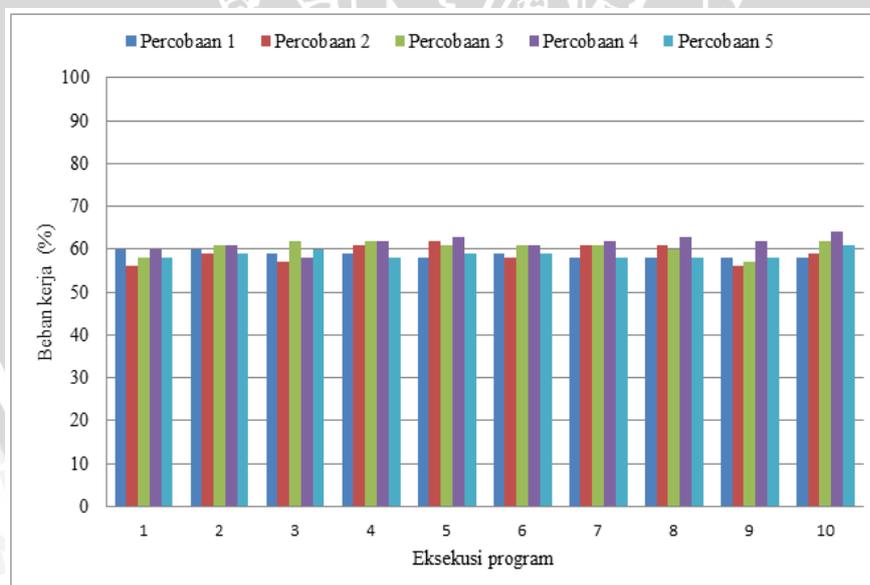
### 6.2.1 Skenario Satu

Pada pengujian skenario satu, *stress load* yang dilakukan hanya ditujukan pada satu mesin virtual yaitu VM 1. Tujuan pengujian skenario satu adalah untuk mengetahui perilaku sistem ketika permintaan akses klien tinggi pada salah satu mesin virtual, khususnya VM 1. Pengujian *stress load* skenario satu dilakukan dengan memberikan banyak permintaan HTTP melalui sebuah komputer klien dalam rentang waktu sepuluh menit. Analisis yang akan dilakukan dari pengujian skenario satu adalah analisis perubahan beban kerja CPU HVM satu. Berikut hasil dari pengujian skenario satu setelah dilakukan lima kali percobaan:

**Tabel 6.1** Hasil Pengujian Skenario Satu

Per- co- baan	Jumlah Permintaan Layanan HTTP			Rata-rata Beban Kerja CPU (%)			Beban Kerja CPU HVM Satu (%)			Jumlah Perpindahan	
	VM 1	VM 2	VM 3	VM 1	V M 2	V M 3	M a k	M i n	Rata- rata	Mig- rasi	Re- mig- rasi
1	547893	0	0	54,8	1	1	60	58	58,7	0	0
2	558247	0	0	55,1	1	1	62	56	59	0	0
3	573783	0	0	57,2	1	1	62	57	60,5	0	0
4	580089	0	0	58,2	1	1	64	58	61,6	0	0
5	544303	0	0	55,5	1	1	61	58	58,8	0	0

Dari lima kali percobaan pengujian skenario satu, dapat dilihat dalam Tabel 6.1 bahwa beban kerja CPU meningkat hingga mencapai beban kerja tertinggi pada angka 64% dan terendah pada angka 56%. Berdasarkan data yang sama, beban kerja CPU HVM satu sebelum dilakukan pengujian *stress load* berkisar diantara angka 8 hingga 12%. Mesin virtual satu sebagai *server target* juga mengalami kenaikan beban kerja CPU yaitu berkisar diantara angka rata-rata tertinggi 58,2% dan terendah 54,8%. Untuk dua mesin virtual yang tidak dilakukan pengujian *stress load*, beban kerja CPU tidak berubah yaitu pada angka 1%. Data-data lengkap mengenai pengujian skenario satu dapat dilihat dalam lampiran yang disertakan penulis pada akhir penulisan penelitian ini. Tabel 6.1 di atas membuktikan bahwa aplikasi Jmeter cukup dapat memberikan beban kerja tinggi pada CPU HVM satu. Selain itu, dapat disimpulkan juga bahwa dengan menjalankan tiga mesin virtual, mesin fisik dapat menangani akses klien dalam jumlah yang cukup besar yaitu sebanyak 580.089 akses dalam rentang waktu sepuluh menit.



**Gambar 6.1** Grafik Beban Kerja CPU HVM Satu Skenario Satu

Gambar 6.1 menunjukkan grafik perubahan beban kerja CPU *server* fisik secara lengkap pada pengujian skenario satu. Dari grafik tersebut dapat dilihat bahwa tidak ada beban kerja yang melebihi angka 80% sehingga pada skenario satu ini praktis tidak ada proses perpindahan mesin virtual terjadi.

### 6.2.2 Skenario Dua

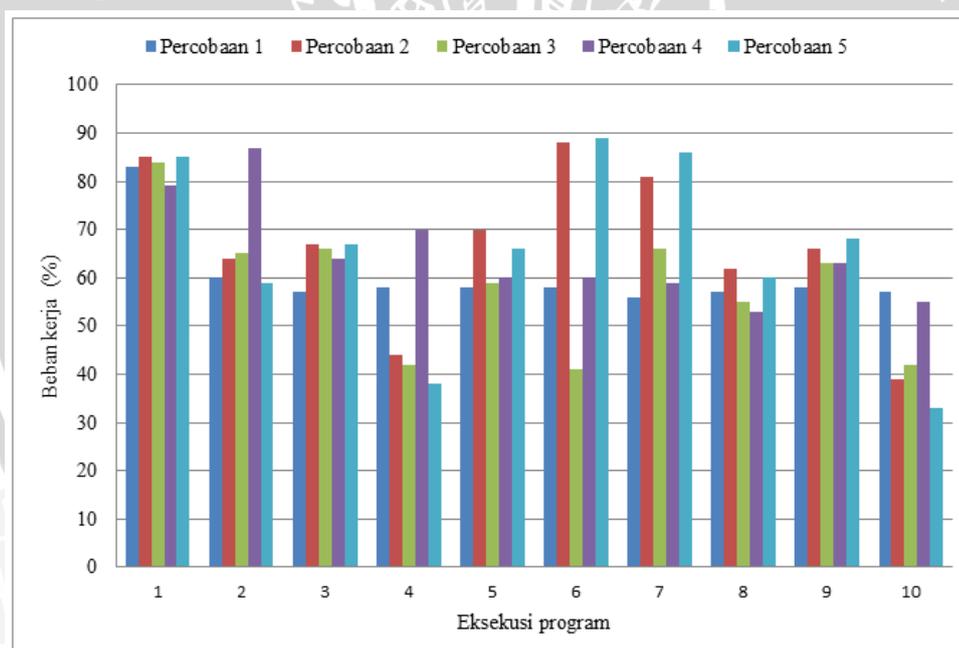
Pada pengujian skenario dua, *stress load* ditujukan pada dua buah mesin virtual yaitu VM 1 dan VM 2. Tujuan dilakukannya pengujian skenario dua adalah untuk mengetahui perilaku sistem ketika permintaan akses klien tinggi pada dua buah mesin virtual, khususnya VM 1 dan VM 2. Selain itu, pengujian skenario dua ini juga diharapkan mampu memberikan beban kerja tinggi *server* fisik yang melebihi angka 80% penggunaan CPU sehingga dapat memicu proses perpindahan mesin virtual. Pengujian *stress load* skenario dua dilakukan dengan memberikan banyak permintaan HTTP melalui dua buah komputer klien yang berbeda dalam waktu sepuluh menit. Analisis yang akan dilakukan dari pengujian skenario dua ini adalah analisis perubahan beban kerja CPU HVM satu dan perpindahan mesin virtual. Berikut hasil pengujian skenario dua setelah dilakukan lima kali percobaan:

**Tabel 6.2** Hasil Pengujian Skenario Dua

Per- co- baan	Jumlah Permintaan Layanan HTTP			Rata-rata Beban Kerja CPU (%)			Beban Kerja CPU HVM Satu (%)			Jumlah Perpindahan	
	VM 1	VM 2	VM 3	VM 1	VM 2	VM 3	M a k	M i n	Rata- rata	Mi- grasi	Re- mig- rasi
1	555035	447441	0	55,5	53,3	0,8	83	56	60,2	1	0
2	393036	409520	0	38,5	44,7	0,5	88	39	66,6	2	1
3	561803	595308	0	49,3	53,3	0,8	84	41	58,3	1	0
4	511303	516922	0	46,8	53,2	0,8	87	53	65	1	0
5	373467	384991	0	35,7	46	0,5	89	33	65,1	2	1

Dari pengujian skenario dua, dapat dilihat dalam Tabel 6.2 bahwa beban kerja CPU *server* fisik berubah lebih dinamis yaitu dengan nilai tertinggi 89% dan nilai terendah 33%. Nilai tertinggi didapat ketika HVM satu menanggung dua beban kerja mesin virtual di menit-menit awal percobaan dan nilai terendah didapat ketika salah satu mesin virtual telah dipindahkan ke mesin lainnya. Pada skenario dua ini, sistem berhasil memindahkan mesin virtual dengan baik setelah

beban kerja HVM satu melebihi batas maksimum yaitu pada angka 80%. Dalam lima kali percobaan yang telah dilakukan, terdapat tiga percobaan dengan jumlah perpindahan mesin virtual sebanyak satu kali dan dua percobaan sebanyak tiga kali. Hal tersebut dikarenakan tidak stabilnya kemampuan mesin virtual dalam melayani permintaan akses layanan HTTP. Pada saat itu terlihat bahwa mesin virtual sempat sesekali *down* dan kemudian dapat melanjutkan layanannya kembali. Hal tersebut juga diperkuat dengan berkurangnya jumlah salah satu akses klien yaitu turun pada angka terendah 393.036 akses dalam waktu sepuluh menit. Idealnya, jika mesin virtual dapat menangani akses klien dengan baik maka tidak ada proses remigrasi yang terjadi karena beban mesin virtual stabil pada HVM dua. Menurut penulis, masalah ini disebabkan karena kurangnya alokasi sumber daya komputasi yaitu *memory* untuk setiap mesin virtual sehingga jumlah akses klien yang dapat ditangani pun terbatas.



**Gambar 6.2** Grafik Beban Kerja CPU HVM Satu Skenario Dua

Gambar 6.2 di atas menunjukkan grafik perubahan beban kerja CPU *server* fisik secara lengkap pada pengujian skenario dua. Dari grafik tersebut dapat dilihat bahwa beban kerja *server* fisik berubah lebih dinamis daripada grafik skenario pengujian satu. Pada menit pertama terlihat bahwa beban kerja CPU melebihi batas angka 80%, terkecuali pada percobaan ke empat. Pada saat itu

proses migrasi mesin virtual dilakukan oleh sistem yang kemudian mengakibatkan beban kerja HVM satu turun di bawah angka 80%. Namun, pada menit ke enam terlihat bahwa beban kerja *server* fisik kembali mengalami peningkatan di atas angka 80% untuk percobaan dua dan lima. Hal ini dikarenakan terjadinya proses remigrasi mesin virtual pada menit ke empat karena mesin virtual tidak dapat melayani akses layanan HTTP dengan stabil sehingga mengakibatkan beban kerja CPU menurun.

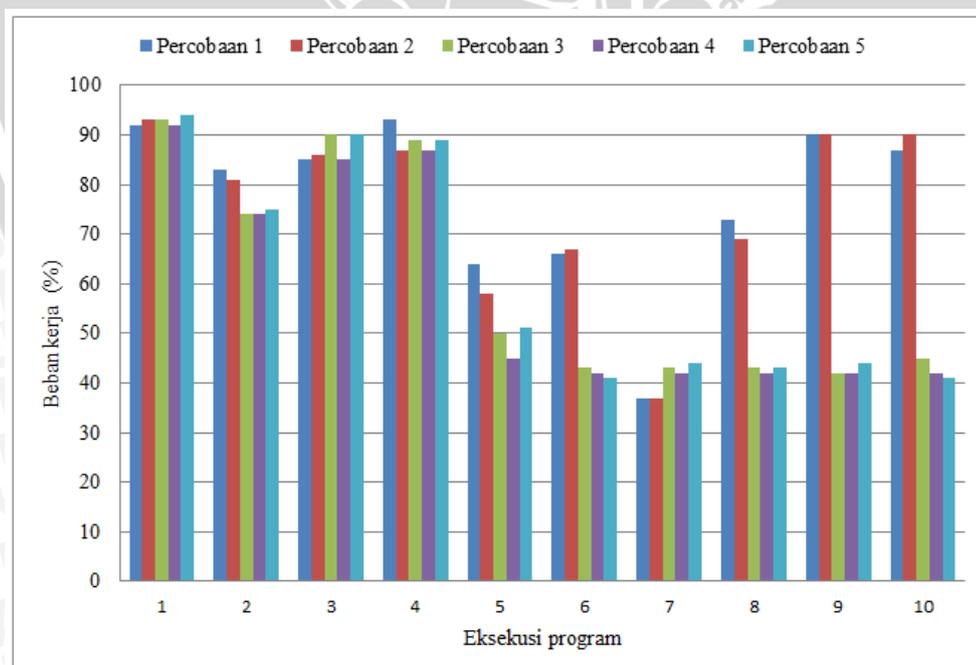
### 6.2.3 Skenario Tiga

Pada pengujian skenario tiga, *stress load* ditujukan pada tiga buah mesin virtual yaitu VM 1, VM 2, dan VM 3. Tujuan dilakukannya pengujian skenario tiga adalah untuk mengetahui perilaku sistem ketika permintaan akses klien tinggi pada tiga buah mesin virtual. Pengujian skenario tiga ini juga diharapkan mampu menghasilkan perpindahan penuh mesin virtual yaitu kondisi dimana setiap HVM menjalankan satu mesin virtual didalamnya. Pengujian *stress load* skenario tiga dilakukan dengan cara memberikan banyak permintaan HTTP melalui tiga buah komputer klien yang berbeda dalam waktu sepuluh menit. Analisis yang akan dilakukan dari pengujian skenario tiga ini adalah analisis perubahan beban kerja CPU HVM satu dan perpindahan mesin virtual. Berikut hasil dari pengujian skenario tiga setelah dilakukan lima kali percobaan:

**Tabel 6.3** Hasil Pengujian Skenario Tiga

Per- co- baan	Jumlah Permintaan Layanan HTTP			Rata-rata Beban Kerja CPU (%)			Beban Kerja CPU HVM Satu (%)			Jumlah Perpindahan	
	VM 1	VM 2	VM 3	V M 1	V M 2	V M 3	M a k	M i n	Rata- rata	Mig- rasi	Re- mig- rasi
1	281278	401919	1044853	38,2	48	44	93	37	77	3	1
2	275483	406419	1038404	36,7	47,2	40	93	37	75.8	3	1
3	376330	595922	974693	45	58,2	37,8	93	42	61.2	2	1
4	549609	433288	1025355	48	51,6	38	92	42	59.3	2	0
5	546906	422553	958707	51	52,3	39,6	94	41	61.2	2	0

Dari pengujian skenario tiga, dapat dilihat bahwa beban kerja CPU *server* fisik berubah lebih dinamis dari pengujian skenario dua. Selain dikarenakan bertambahnya jumlah mesin virtual yang diakses oleh klien, hal ini juga disebabkan oleh masih tidak stabilnya penanganan permintaan layanan HTTP. Berdasarkan Tabel 6.3, terlihat bahwa HVM satu menembus angka tertinggi 94% ketika HVM satu menjalankan tiga mesin virtual dan 37% untuk angka terendah ketika HVM satu telah memindahkan dua mesin virtual ke HVM lain. Masalah yang sama seperti pengujian skenario dua juga terjadi pada proses perpindahan mesin virtual skenario tiga. Tercatat bahwa selama sepuluh menit terdapat tiga percobaan dengan dua kali migrasi dan satu remigrasi, sedangkan untuk proses idealnya hanya tercatat dua percobaan yaitu pada percobaan empat dan percobaan lima dengan dua kali proses migrasi saja. Hal ini juga diperkuat dengan tidak seimbangannya jumlah akses antara tiga komputer klien yaitu tercatat sejumlah 1.044.853 untuk akses yang terbesar dan 275.483 untuk akses yang terkecil. Meskipun demikian, proses pengujian dapat dibilang berhasil karena sistem dapat mensimulasikan beban kerja dinamis *server* fisik dan juga proses pemindahan mesin virtual tetap dilakukan dengan baik oleh sistem virtualisasi *server* ini.



**Gambar 6.3** Grafik Beban Kerja CPU HVM Satu Skenario Tiga

Gambar 6.3 menunjukkan grafik perubahan beban kerja CPU *server* fisik secara lengkap pada pengujian skenario tiga. Sama seperti pada pengujian skenario dua, grafik tersebut berubah sangat dinamis di setiap menitnya karena terjadinya proses perpindahan mesin virtual. Pada menit pertama terlihat bahwa beban kerja CPU pada seluruh percobaan melebihi batas angka 80% dan bahkan menembus angka 90%. Pada saat itu HVM satu menanggung beban kerja dari seluruh mesin virtual yang berada di dalamnya. Setelah salah satu mesin virtual dipindahkan, beban kerja sedikit menurun namun pada menit ke tiga dan keempat naik kembali di atas angka 80%. Hal tersebut dikarenakan masih terdapat dua mesin virtual dengan beban kerja tinggi pada HVM satu sehingga proses migrasi pun perlu dilakukan kembali. Setelah itu terlihat beban kerja CPU HVM satu turun di bawah angka 80% dan pada kondisi ini setiap HVM menanggung satu *server* fisik di dalamnya. Khusus pada percobaan satu dan dua, proses remigrasi terjadi pada menit ke tujuh dan kemudian diikuti dengan proses migrasi kembali pada menit ke sepuluh.

### 6.3 Analisis

Berdasarkan hasil pengujian yang telah dilakukan melalui pengujian skenario satu, skenario dua, dan skenario tiga, data menunjukkan bahwa sistem virtualisasi *server* dapat berjalan secara fungsional. Ketika HVM satu menanggung beban kerja yang melebihi batas maksimal penggunaan CPU yaitu pada angka 80%, maka secara otomatis sistem dapat memindahkan mesin virtual yang memiliki beban kerja tertinggi. Berikut salah satu contoh proses migrasi mesin virtual dengan beban kerja CPU tertinggi pada pengujian skenario dua percobaan satu:

Sat Dec 22 12:21:01 EST 2012	TIDAK ADA MIGRASI (3VM)	HVM1 CPU = 9%	HVM1 MEM = 65%	VM1 LOAD = 2%	VM2 LOAD = 2%	VM3 LOAD = 0%
Sat Dec 22 12:22:02 EST 2012	MIGRASI (VM1 KE HVM2)	HVM1 CPU = 83%	HVM1 MEM = 65%	VM1 LOAD = 41%	VM2 LOAD = 40%	VM3 LOAD = 0%
Sat Dec 22 12:23:01 EST 2012	PROSES PERPINDAHAN BERJALAN	HVM1 CPU = 60%	HVM1 MEM = 52%			
Sat Dec 22 12:24:01 EST 2012	PROSES PERPINDAHAN BERJALAN	HVM1 CPU = 57%	HVM1 MEM = 52%			
Sat Dec 22 12:25:01 EST 2012	TIDAK ADA MIGRASI (2VM)	HVM1 CPU = 58%	HVM1 MEM = 52%	VM1 LOAD = 58%	VM2 LOAD = 56%	VM3 LOAD = 1%

**Gambar 6.4** Log Proses Migrasi Mesin Virtual

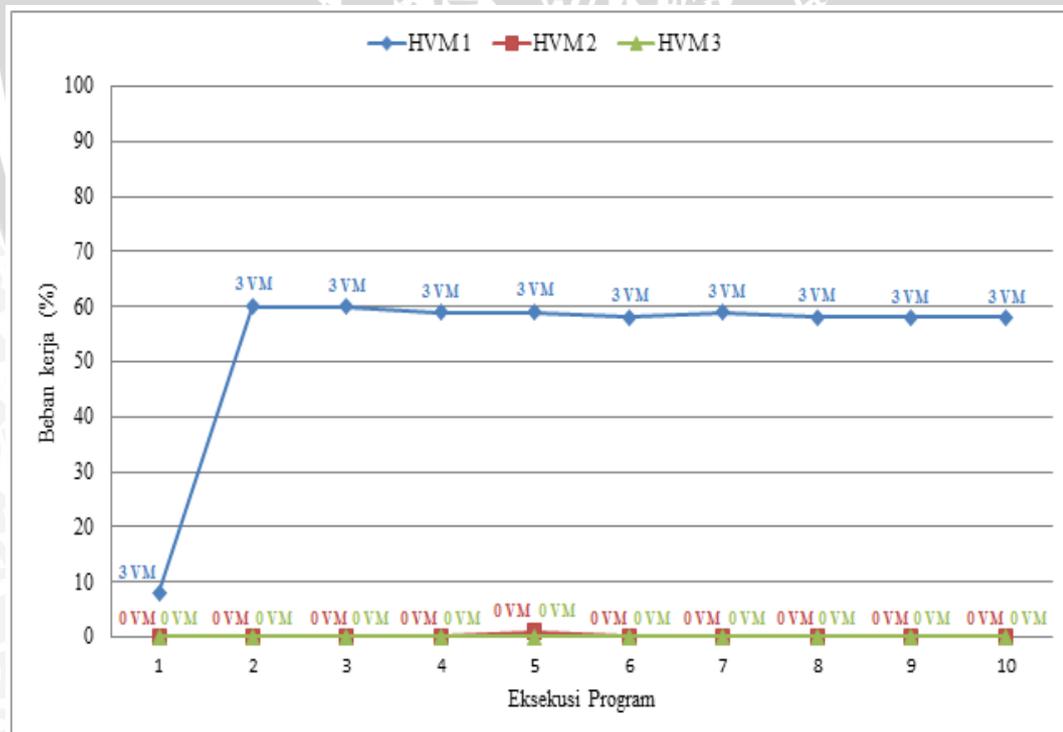
Sebaliknya, ketika *server* fisik atau mesin virtual menanggung beban kerja yang rendah, maka sistem tetap dapat mempertahankan tiga mesin virtual untuk

berjalan di HVM satu atau memindahkan kembali mesin virtual yang berada di server fisik lain untuk menuju kembali ke HVM satu. Berikut salah satu contoh proses remigrasi mesin virtual yang terjadi pada pengujian skenario dua percobaan satu:

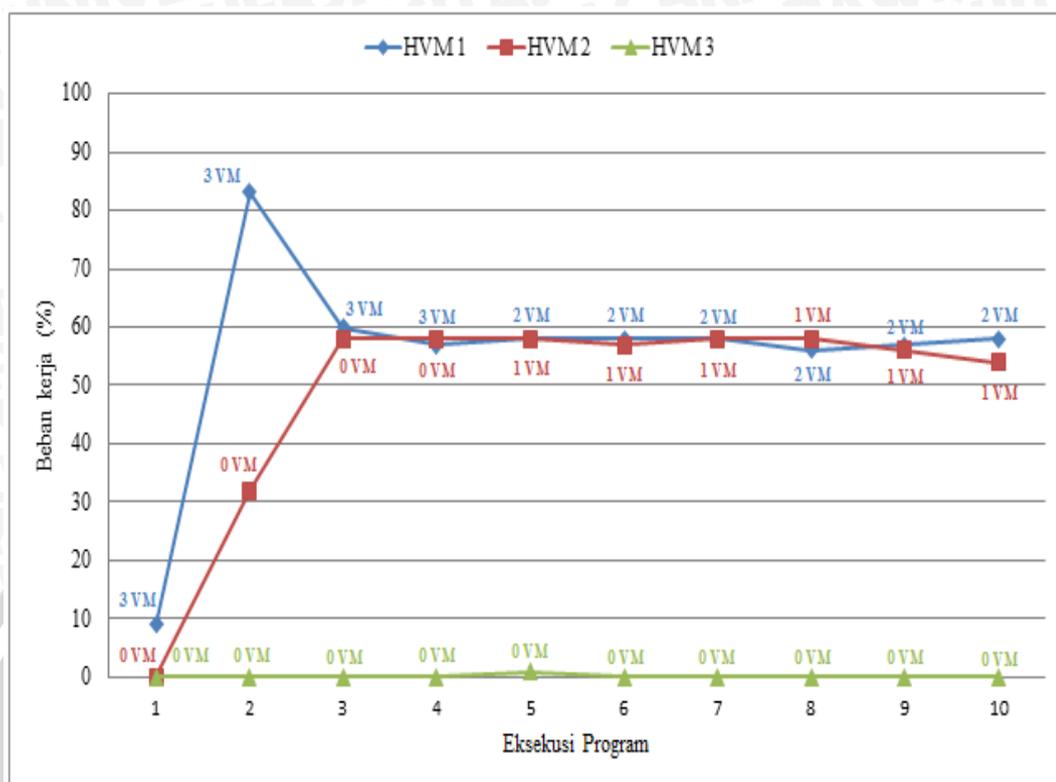
Sat Dec 22 12:31:01 EST 2012	TIDAK ADA MIGRASI (2VM)	HVM1 CPU = 57%	HVM1 MEM = 54%	VM1 LOAD = 54%	VM2 LOAD = 55%	VM3 LOAD = 1%
Sat Dec 22 12:32:01 EST 2012	REMIGRASI (VM1 KE HVM1)	HVM1 CPU = 7%	HVM1 MEM = 54%	VM1 LOAD = 3%	VM2 LOAD = 2%	VM3 LOAD = 2%
Sat Dec 22 12:33:01 EST 2012	PROSES PERPINDAHAN BERJALAN	HVM1 CPU = 7%	HVM1 MEM = 69%			
Sat Dec 22 12:34:01 EST 2012	PROSES PERPINDAHAN BERJALAN	HVM1 CPU = 7%	HVM1 MEM = 69%			
Sat Dec 22 12:35:01 EST 2012	TIDAK ADA MIGRASI (3VM)	HVM1 CPU = 7%	HVM1 MEM = 69%	VM1 LOAD = 2%	VM2 LOAD = 2%	VM3 LOAD = 2%

**Gambar 6.5** Log Proses Remigrasi Mesin Virtual

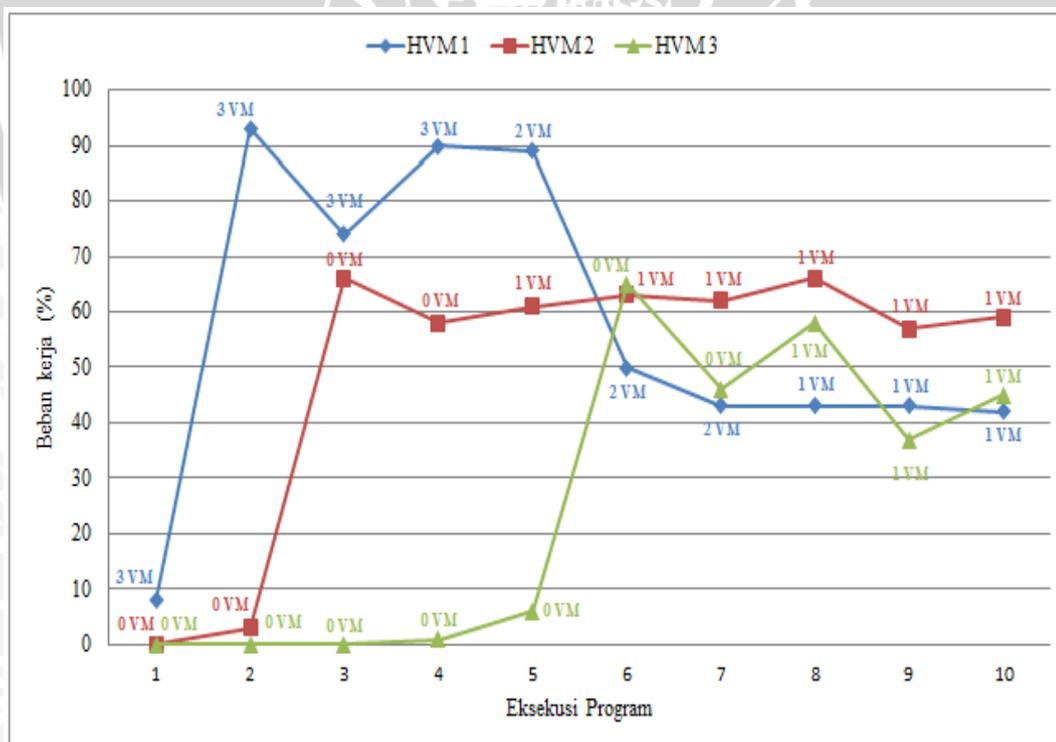
Selain itu, terlihat juga penurunan tingkat beban kerja CPU HVM satu setelah proses perpindahan mesin virtual dilakukan di setiap percobaan. Data-data lengkap mengenai proses perpindahan mesin virtual yang terjadi di seluruh percobaan, penulis sertakan dalam lampiran di akhir penulisan penelitian ini. Berikut grafik perubahan beban kerja CPU HVM satu, HVM dua, dan HVM tiga pada salah satu percobaan dari setiap skenario pengujian:



**Gambar 6.6** Grafik Beban Kerja CPU HVM Skenario Satu Percobaan Satu



Gambar 6.7 Grafik Beban Kerja CPU HVM Skenario Dua Percobaan Satu



Gambar 6.8 Grafik Beban Kerja CPU HVM Skenario Tiga Percobaan Tiga

Hasil lainnya yang dapat disimpulkan dari Gambar 6.4 dan Gambar 6.5 yaitu sistem virtualisasi *server* ini memiliki *response time* selama satu menit dan waktu pindah selama dua menit. *Response time* tersebut diperoleh berdasarkan data-data yang menunjukkan bahwa sistem mengeksekusi program di setiap satu menit. Dengan begitu, setiap saat ketika sistem mendapatkan pemicu proses perpindahan, yaitu beban kerja CPU di atas angka 80%, maka sistem ini akan tanggap untuk melakukan proses perpindahan mesin virtual pada menit berikutnya. Sedangkan untuk waktu migrasi dan remigrasi, sistem ini membutuhkan waktu paling lama dua menit untuk melakukan perpindahan mesin virtual. Hal tersebut dikarenakan sistem virtualisasi *server* tidak dapat melakukan monitoring beban kerja CPU mesin virtual selama proses perpindahan dilakukan. Pada rentang waktu itu, mesin virtual dapat dikatakan dalam kondisi tidak tersedia. Namun pada kenyataannya, dengan penggunaan teknik *live migration* mesin virtual dan jaringan lokal maka proses perpindahan mesin virtual dapat berlangsung begitu cepat sehingga layanan *server* virtual tetap dapat diakses.

