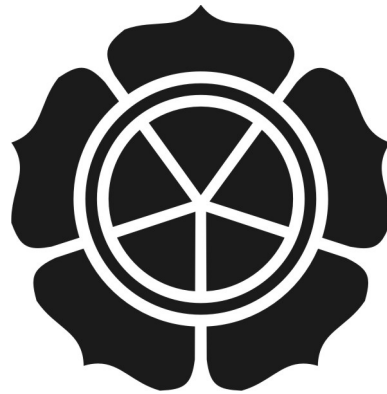


**ANALISIS PENERAPAN *PHP-FPM* DAN *HHVM* SEBAGAI  
*INTERPRETER* KODE PROGRAM *PHP***

**SKRIPSI**



disusun oleh

**Rizqon Sadida**

**11.11.5381**

**JURUSAN TEKNIK INFORMATIKA  
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER  
AMIKOM YOGYAKARTA  
YOGYAKARTA  
2015**

**ANALISIS PENERAPAN *PHP-FPM* DAN *HHVM* SEBAGAI  
*INTERPRETER* KODE PROGRAM *PHP***

**SKRIPSI**

untuk memenuhi sebagian persyaratan  
mencapai derajat Sarjana S1  
pada jurusan Teknologi Informasi



disusun oleh

**Rizqon Sadida**

**11.11.5381**

**JURUSAN TEKNIK INFORMATIKA  
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER  
AMIKOM YOGYAKARTA  
YOGYAKARTA  
2015**

**PERSETUJUAN**

**SKRIPSI**

**ANALISIS PENERAPAN *PHP-FPM* DAN *HHVM* SEBAGAI  
*INTERPRETER* KODE PROGRAM PHP**

yang disusun oleh

**Rizqon Sadida**

**11.11.5381**

Telah disetujui oleh Dosen Pembimbing Skripsi  
pada tanggal 8 November 2015

**Dosen Pembimbing,**

**Andi Sunyoto M. Kom**  
**NIK. 190302052**

**PENGESAHAN**  
**SKRIPSI**  
**ANALISIS PENERAPAN PHP-FPM DAN HHVM SEBAGAI**  
**INTERPRETER PADA KODE PROGRAM PHP**

yang disusun oleh

**Rizqon Sadida**

**11.11.5381**

Telah dipertahankan di depan Dewan Penguji  
pada tanggal 23 November 2015

Susunan Dewan Penguji

**Nama Penguji**

**Tanda Tangan**

**Bayu Setiaji M. Kom**

**NIK. 190302216**

**Mei P Kurniawan M. Kom**

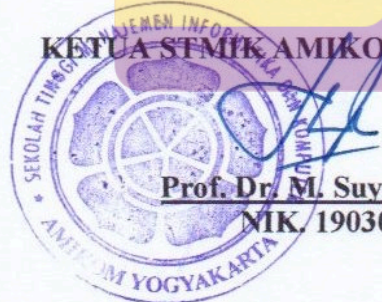
**NIK. 190302187**

**Andi Sunyoto M. Kom**

**NIK. 190302052**

Skripsi ini telah diterima sebagai salah satu persyaratan  
untuk memperoleh gelar Sarjana Komputer  
Tanggal 23 November 2015

**KETUA STMIK AMIKOM YOGYAKARTA**



**Prof. Dr. M. Suyanto, M.M.**

**NIK. 190302001**

## PERNYATAAN

Saya yang bertandatangan dibawah ini menyatakan bahwa, skripsi ini merupakan karya saya sendiri (ASLI), dan isi dalam skripsi ini tidak terdapat karya yang pernah diajukan oleh orang lain untuk memperoleh gelar akademis di suatu institusi pendidikan tinggi manapun, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis dan/atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Segala sesuatu yang terkait dengan naskah dan karya yang telah dibuat adalah menjadi tanggungjawab saya pribadi.

Yogyakarta, .....

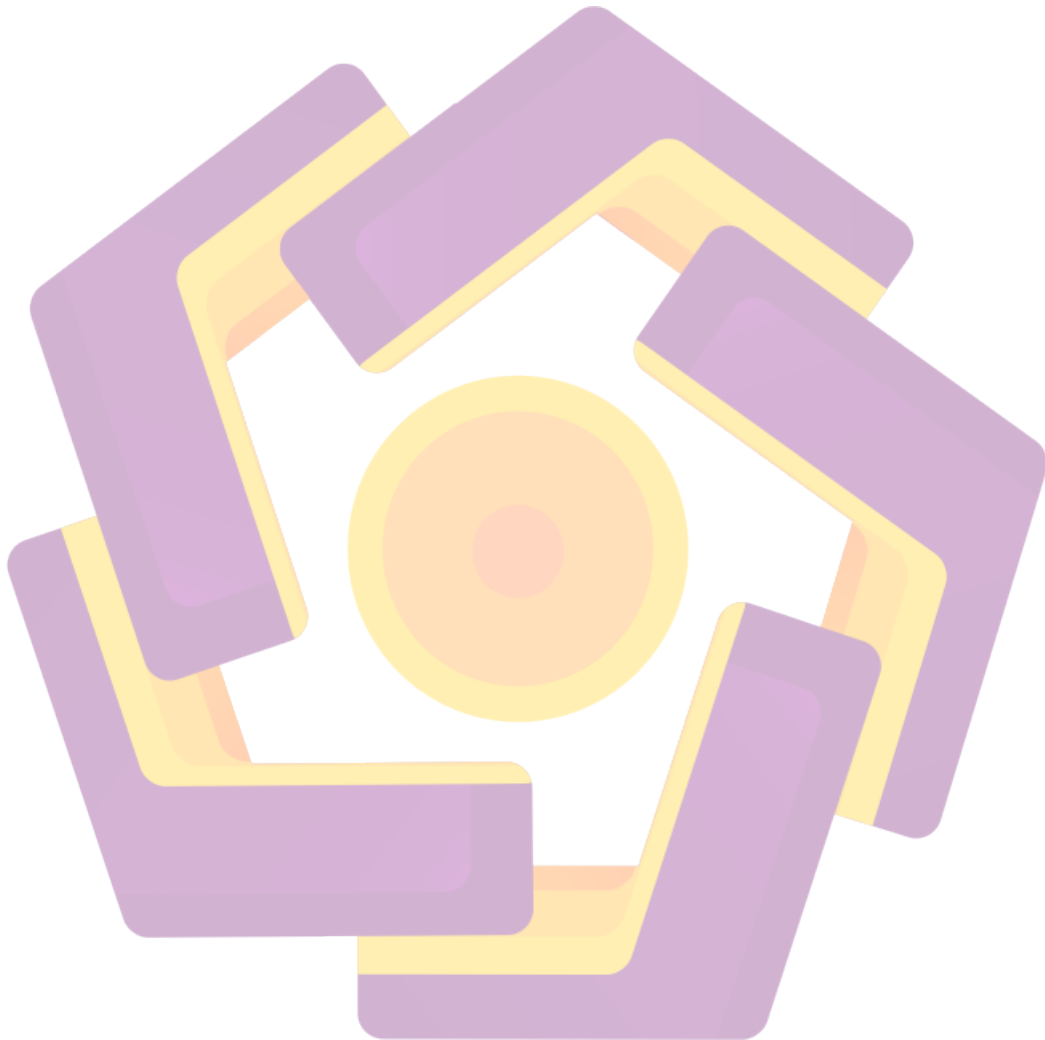
Rizqon Sadida

NIM. 11.11.5381

## **MOTTO**

“Make things as simple as possible, but not simpler.”

- Albert Einstein



## PERSEMBAHAN

Sembah sujud serta syukur kepada Allah SWT, atas segala nikmat hidup dan kesempatan menggenggam ilmu, sehingga penulis dapat menyelesaikan skripsi yang berjudul “Analisis Penerapan *PHP-FPM* dan *HHVM* Sebagai Interpreter kode Program *PHP*”. Oleh karena itu, pada kesempatan ini penulis sangat ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Kedua orang tua saya yang senantiasa mengingatkan saya untuk sholat lima waktu, potong kuku, dan menanyakan “kapan sidang?” sehingga akhirnya saya bisa disidang beneran.
2. Untuk dosen pembimbing saya, pak Andi. Terimakasih banyak sudah sabar dalam membimbing saya. Walaupun saya sempet libur skripsi selama setahun.
3. Untuk temen-temen PAG (Rian, Herman, Agung, Rendy, Randy, Kukuh, Desi, Mapong, Bang Jib, Guntur, Kiki, Hanox, Yogi, Fani, Nyo2, Hari, Eko, Dimas, Faiz, Agnes, sing rung kesebut ojo nesul!).
4. Buat temen-temen di Kadol (Robin, Indrak, Fuzak, Alpin, Kikik, Mira, Farandi, Amad, Laka, bang Pian dll), Mohon bimbingannya buat nge-up website.
5. Buat temen-temen di Fossil, makasih udah sharing, berbagi ilmu sehingga saya bisa mendapat inspirasi judul skripsi ini, tetep dukung open source dab!
6. Buat temen-temen Maju Jaya Abadi (Obet, Mas Huda, Mas Remick, Yudhi, koe kapan yud?). berkat kalian saya jadi punya banyak pengalaman lomba, ayo kapan lomba meneh? Wingi aku skripsian je.
7. Buat pak Doni, Makasih pak Don udah di kasih jalan Toool...
8. Terima Kasih buat (sebut saja mawar). Yang sudah memberi saya semangat dan doa. Walau sekarang kita udah jarang main lagi.
9. Kepada petani kopi seluruh Indonesia, saya ucapkan terimakasih. Mungkin tanpa kalian saya tidak bisa ngopi sambil bikin skripsi.
10. Untuk temen-temen yang kenal maupun yang gak kenal, terimakasih sudah mengingatkan saya dengan “skripsi piye?”. “kapan sidang?”, “DO wae”, “skripsimu sampai mana?”, “ayo karaoke wae.”, “bandungan sek kon”.

## KATA PENGANTAR

Puji syukur kehadiran Tuhan Yang Maha Esa atas segala rahmatNYA sehingga skripsi ini dapat tersusun hingga selesai. Penulis juga mengucapkan banyak terimakasih atas bantuan dari pihak yang telah berkontribusi dengan memberikan sumbangan baik materi maupun pikirannya.

Skripsi ini telah penulis susun dengan maksimal dan mendapatkan bantuan dari berbagai pihak sehingga dapat memperlancar pembuatan skripsi ini. Untuk itu penulis menyampaikan banyak terima kasih kepada semua pihak yang telah berkontribusi dalam pembuatan skripsi ini.

Terlepas dari semua itu, penulis menyadari sepenuhnya bahwa masih ada kekurangan baik dari segi susunan kalimat maupun tata bahasanya. Oleh karena itu dengan tangan terbuka kami menerima segala saran dan kritik dari pembaca agar penulis dapat memperbaiki skripsi ini.

Akhir kata penulis berharap semoga skripsi tentang interpreter php ini dapat memberikan manfaat maupun inspirasi terhadap pembaca.

Yogyakarta, 7 Desember 2015

Rizqon Sadida  
NIM 11.11.5381



## DAFTAR ISI

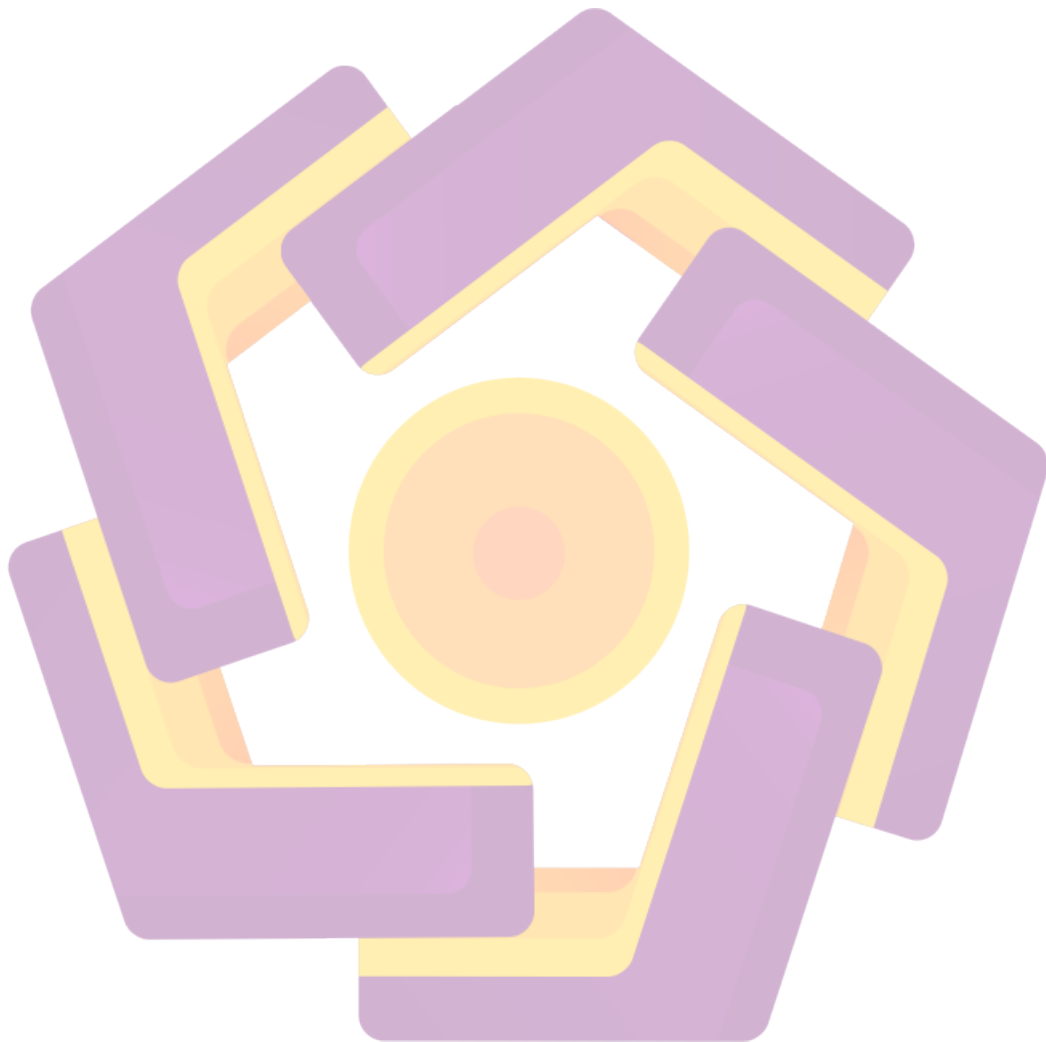
PERNYATAAN .....	iv
MOTTO .....	v
PERSEMBAHAN.....	vi
KATA PENGANTAR .....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xiv
INTISARI.....	xvi
<i>ABSTRACT</i> .....	xvii
<b>BAB 1 PENDAHULUAN</b> .....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Penelitian.....	2
1.4 Tujuan Penelitian.....	3
1.5 Metode Penelitian.....	3
1.5.1 Metode Pengumpulan Data.....	3
1.5.2 Metode Analisis.....	4
1.6 Sistematika Penulisan.....	4
<b>BAB 2 LANDASAN TEORI</b> .....	6
2.1 Tinjauan Pustaka.....	6
2.2 Landasan Teori.....	6
2.2.1 Pengertian Virtual Server.....	6
2.2.2 Pengertian Interpreter dan Compiler.....	8
2.2.3 Pengertian Webserver.....	8
2.2.4 Pengertian Web Server Benchmarking.....	9
2.2.5 Pengertian PHP-FPM.....	9
2.2.6 Pengertian HHVM.....	10
2.3 Metode Analisis <i>Test-Driven Infrastructure</i> .....	11
2.4 Langkah-Langkah Analisis.....	12
<b>BAB 3 METODE PENELITIAN</b> .....	14

3.1	Tinjauan Umum .....	14
3.2	Analisis Masalah .....	14
3.3	Alur Penelitian .....	15
3.4	Rancangan <i>Server</i> Beserta <i>PHP5-FPM</i> dan <i>HHVM</i> .....	16
3.5	<i>Variable</i> Penelitian.....	17
3.5.1	Kecepatan Dalam Mengeksekusi Benchmark Script.....	17
3.5.2	Konsumsi Resource Dalam Mengeksekusi Benchmark Script.....	17
3.5.3	Kecepatan Web Server Dalam Implementasi Program PHP .....	17
3.5.4	Konsumsi Resource Server Dalam Implementasi Program PHP .....	17
3.6	Alat dan Bahan Peneliiian.....	17
3.6.1	Analisa Perangkat Keras (Hardware).....	18
3.6.2	Analisis Perangkat Lunak (software).....	18
3.7	Rancangan Penggunaan <i>Glances Server Monitoring Server</i> .....	20
3.7.1	CPU (Central Processing Unit).....	20
3.7.2	RAM (Random Access Memory) .....	22
3.8	Rancangan Penggunaan <i>ApacheBench</i> .....	23
3.9	Rancangan Penggunaan <i>XHProf</i> .....	24
3.10	Rancangan Pada Kedua <i>Server</i> .....	24
3.10.1	Instalasi Sistem Operasi Paket-Paket Yang dibutuhkan .....	24
3.11	Rancangan Pada Komputer <i>Remote</i> .....	26
3.12	Rancangan Pengambilan Data .....	27
3.12.1	Pengambilan Data Kecepatan Kode Program PHP .....	27
3.12.2	Pengambilan Data Resource Menggunakan <i>XHProf</i> .....	29
3.12.3	Pengambilan Data Kecepatan Web Server .....	30
3.12.4	Pengambilan Data Konsumsi Resource Server.....	32
BAB 4	HASIL DAN PEMBAHASAN .....	33
4.1	Analisis Benchmark Kode Program <i>PHP</i> .....	33
4.1.1	Analisis Fungsi <i>Ackermann(7)</i> .....	33
4.1.2	Analisis Fungsi <i>Ary()</i> .....	34
4.1.3	Analisis Fungsi <i>Ary2()</i> .....	35
4.1.4	Analisis Fungsi <i>Ary3()</i> .....	36

4.1.5	Analisis Fungsi Fibo(30) .....	37
4.1.6	Analisa Fungsi Hash(500).....	38
4.1.7	Analisa Fungsi Hash1(50000).....	39
4.1.8	Analisa Fungsi Heapsort(20000) .....	41
4.1.9	Analisa Fungsi Mandel().....	42
4.1.10	Analisa Fungsi Mandel2().....	43
4.1.11	Analisa Fungsi Matrix(20).....	44
4.1.12	Analisa Fungsi Nestedloop(12).....	45
4.1.13	Analisa Fungsi Sieve(30).....	46
4.1.14	Analisa Fungsi Simple() .....	47
4.1.15	Analisa Fungsi Simplecall().....	48
4.1.16	Analisa Fungsi Simpleucall().....	49
4.1.17	Analisa Fungsi Simpleudcall().....	50
4.1.18	Analisa Fungsi Strcat() .....	51
4.1.19	Analisa Waktu Tempuh Keseluruhan Pada Script Benchmark .....	52
4.2	Analisis <i>Resource</i> Pada Kode Program <i>PHP</i> .....	54
4.2.1	Analisis Walltime.....	54
4.2.2	Analisis CPU Time .....	56
4.2.3	Analisis Memory Usage.....	57
4.2.4	Analisis Peak Memory .....	58
4.3	Analisis Kecepatan <i>Interpreter</i> Terhadap <i>Web Server</i> .....	60
4.3.1	Analisa Jumlah Request per Second .....	60
4.3.2	Analisa <i>Transfer Rate</i> .....	61
4.3.3	Analisa Time per Request.....	63
4.3.4	Analisis Maximum Time Taken .....	64
4.4	Analisa Konsumsi <i>Resource Server</i> Terhadap Jumlah Koneksi.....	65
4.4.1	Analisa Penggunaan CPU .....	65
4.4.2	Analisa Penggunaan RAM.....	67
BAB 5	KESIMPULAN DAN SARAN .....	69
5.1	Kesimpulan .....	69
5.2	Saran .....	70

DAFTAR PUSTAKA.....71

LAMPIRAN



## DAFTAR TABEL

Tabel 3.1 Spesifikasi Laptop <i>Remote</i> .....	18
Tabel 3.2 Spesifikasi <i>Server Vultr</i> .....	18
Tabel 3.3 Spesifikasi <i>Software</i> Komputer / Laptop .....	19
Tabel 3.4 Spesifikasi <i>Software Server</i> Satu .....	19
Tabel 3.5 Spesifikasi <i>Software Server</i> Dua.....	19
Tabel 3.6 Parameter <i>CPU</i> pada <i>Glances</i> .....	21
Tabel 3.7 Parameter <i>Alert CPU</i> Pada <i>Glances</i> .....	22
Tabel 3.8 Parameter <i>RAM</i> pada <i>Glances</i> .....	22
Tabel 3.9 Parameter <i>alert RAM</i> pada <i>Glances</i> .....	23
Tabel 3.10 Paramater yang disediakan <i>Apachebench</i> .....	23
Tabel 3.11 Sistem Operasi Kedua <i>Server</i> .....	25
Tabel 3.12 Paket yang diinstall pada kedua <i>Server</i> .....	25
Tabel 3.13 Kebutuhan Sistem Operasi pada Komputer <i>Remote</i> .....	26
Tabel 3.14 Kebutuhan paket yang digunakan dalam Penelitian.....	27
Tabel 3.15 Fungsi pada <i>zend/Bench.php</i> .....	27
Tabel 3.16 Dependency Perintah <i>ApacheBench</i> .....	31
Tabel 4.1 Data Pengujian Fungsi <i>Ackermann(7)</i> .....	33
Tabel 4.2 Data Pengujian Fungsi <i>Ary()</i> .....	34
Tabel 4.3 Data Pengujian Fungsi <i>Ary2()</i> .....	35
Tabel 4.4 Data Pengujian Fungsi <i>Ary3()</i> .....	36
Tabel 4.5 Data Pengujian Fungsi <i>Fibo()</i> .....	37
Tabel 4.6 Data Pengujian Pada Fungsi <i>Hash1()</i> .....	39
Tabel 4.7 Data Pengujian Fungsi <i>Hash2()</i> .....	40
Tabel 4.8 Data Pengujian Fungsi <i>Heapsort()</i> .....	41
Tabel 4.9 Data Pengujian Fungsi <i>Mandel()</i> .....	42
Tabel 4.10 Data Pengujian Fungsi <i>Mandel2()</i> .....	43
Tabel 4.11 Data Pengujian Fungsi <i>Matrix(20)</i> .....	44
Tabel 4.12 Data Pengujian <i>Nestedloop(12)</i> .....	45
Tabel 4.13 Data Pengujian Fungsi <i>Sieve(30)</i> .....	46

Tabel 4.14 Data Pengujian Fungsi <i>Simple()</i> .....	47
Tabel 4.15 Data Pengujian Fungsi <i>Simplecall()</i> .....	48
Tabel 4.16 Data Pengujian Fungsi <i>Simpleucall()</i> .....	49
Tabel 4.17 Data Pengujian Fungsi <i>Simpleudcall</i> .....	50
Tabel 4.18 Data Pengujian Fungsi <i>Strcat()</i> .....	52
Tabel 4.19 Data Pengujian Keseluruhan Terhadap <i>Script Benchmark</i> .....	53
Tabel 4.20 Analisa <i>Walltime</i> Terhadap <i>Script Benchmark</i> .....	55
Tabel 4.21 Analisa CPU Time Terhadap <i>Script Benchmark</i> .....	56
Tabel 4.22 Analisa Memory Terhadap <i>Script Benchmark</i> .....	57
Tabel 4.23 Analisis Peak Memory Terhadap <i>Script Benchmark</i> .....	59
Tabel 4.24 Data Jumlah <i>Request per Second</i> .....	61
Tabel 4.25 Data <i>Request Rate (Kbytes/Second)</i> .....	62
Tabel 4.26 Data <i>Time per Request</i> .....	63
Tabel 4.27 Data <i>Maximum Time Taken</i> .....	64
Tabel 4.28 Data Konsumsi <i>CPU</i> Terhadap <i>Interpreter PHP-FPM</i> .....	65
Tabel 4.29 Data Konsumsi <i>CPU</i> Terhadap <i>Interpreter HHVM</i> .....	65
Tabel 4.30 Penggunaan <i>RAM PHP-FPM</i> .....	67
Tabel 4.31 Penggunaan <i>RAM HHVM</i> .....	68

## DAFTAR GAMBAR

Gambar 2.1 Cara Kerja <i>Virtual Server</i> .....	7
Gambar 2.2 Proses Kerja <i>Interpreter</i> dan <i>Compiler</i> .....	8
Gambar 2.3 Proses Kerja <i>Web Sever</i> .....	9
Gambar 2.4 Proses Interpretasi <i>PHP-FPM</i> .....	10
Gambar 2.5 Proses <i>HPPc</i> .....	11
Gambar 2.6 <i>JIT (Just In Time) Compiler</i> .....	11
Gambar 3.1 Alur Penelitian .....	15
Gambar 3.2 Rancangan <i>Server, Host 1</i> dan <i>Host 2</i> Serta <i>Laptop Remote</i> .....	16
Gambar 3.3 Script Penggunaan <i>XHProf</i> .....	29
Gambar 3.4 Perintah Penggunaan <i>ApacheBench</i> .....	31
Gambar 4.1 Grafik Pengujian Fungsi <i>Ackermann(7)</i> .....	34
Gambar 4.2 Grafik Pengujian Fungsi <i>Ary()</i> .....	35
Gambar 4.3 Grafik Pengujian Fungsi <i>Ary2()</i> .....	36
Gambar 4.4 Grafik Pengujian Fungsi <i>Ary3()</i> .....	37
Gambar 4.5 Grafik Perbandingan Fungsi <i>Fibo()</i> .....	38
Gambar 4.6 Grafik Pengujian Fungsi <i>Hash(500)</i> .....	39
Gambar 4.7 Grafik Pengujian Fungsi <i>Hash1(50000)</i> .....	40
Gambar 4.8 Grafik Pengujian Fungsi <i>Heapsort()</i> .....	41
Gambar 4.9 Grafik Pengujian <i>Mandel()</i> .....	42
Gambar 4.10 Grafik Pengujian Fungsi <i>Mandel2()</i> .....	43
Gambar 4.11 Grafik Pengujian Fungsi <i>Matrix(20)</i> .....	44
Gambar 4.12 Grafik Perbandingan Fungsi <i>Nestedloop()</i> .....	45
Gambar 4.13 Grafik Pengujian Fungsi <i>Sieve(30)</i> .....	47
Gambar 4.14 Grafik Pengujian Fungsi <i>Simple()</i> .....	48
Gambar 4.15 Grafik Pengujian Fungsi <i>Simplecall()</i> .....	49
Gambar 4.16 Grafik Pengujian Fungsi <i>Simpleucall()</i> .....	50
Gambar 4.17 Grafik Pengujian Fungsi <i>Simpleudcall()</i> .....	51
Gambar 4.18 Grafik Pengujian <i>Strcat()</i> .....	52

Gambar 4.19 Grafik Pengujian Keseluruhan Terhadap <i>Script Benchmark</i> .....	54
Gambar 4.20 Grafik <i>Walltime</i> Pada Proses <i>Profiling</i> .....	55
Gambar 4.21 Grafik <i>CPU Time</i> Pada Proses <i>Profiling</i> .....	57
Gambar 4.22 Grafik Konsumsi <i>RAM</i> Terhadap <i>Script Benchmark</i> .....	58
Gambar 4.23 <i>Memory Peak HHVM</i> dan <i>PHP-FPM</i> .....	60
Gambar 4.24 Grafik Perbandingan <i>Request/Second</i> .....	61
Gambar 4.25 Grafik Perbandingan <i>Transfer Rate</i> .....	62
Gambar 4.26 Grafik Perbandingan <i>Time per Request</i> .....	63
Gambar 4.27 Grafik Perbandingan <i>Maximum Time Taken</i> .....	65
Gambar 4.28 Perbandingan Penggunaan <i>CPU User</i> Terhadap <i>Web Server</i> .....	66
Gambar 4.29 Perbandingan Penggunaan <i>CPU System</i> Terhadap <i>Web Server</i> .....	67
Gambar 4.30 Grafik Penggunaan <i>RAM</i> Terhadap <i>Web Server</i> .....	68





## INTISARI

Seiring dengan pertumbuhan pengguna internet di dunia, yang dilansir oleh internet.org bahwa pengguna internet di dunia pada saat ini mencapai 31.7 miliar orang dan dari tahun ke tahun pengguna internet di dunia mengalami peningkatan sebanyak 7.6% [3]. Dengan pertumbuhan pengguna internet ini, pemilik *website* yang jumlah pengunjungnya terus bertambah di haruskan untuk mengalokasikan *server* baru agar *service* mereka tetap bejalan. Hal ini berdampak pada biaya yang harus di keluarkan semakin besar [4].

Dari permasalahan diatas dapat disimpulkan bahwa sistem yang di *develop* tidak di desain untuk mensupport jumlah *request* yang besar dan *concurrency* yang banyak. Sebagai alternatif, seorang *system engineer* dapat mengkonversi sistem yang mereka gunakan dengan teknologi yang tepat dan efisien. Salah satu solusi mengatasi permasalahan tersebut adalah dengan menggunakan *fastCGI*. Namun pada implementasinya *fastCGI* tidak secara langsung di tunjukkan untuk program *PHP*. Melainkan harus menggunakan aplikasi tambahan seperti *PHP-FPM* atau *HHVM* [6]. Aplikasi tambahan ini berfungsi sebagai interpreter yang menghubungkan kode program *PHP* dengan *fastCGI* yang menghubungkan interpreter dengan bahasa mesin.

Masalah baru muncul seiring banyaknya teknologi baru, hal ini menimbulkan kebingungan *system engineer* dalam menentukan sistem mana yang akan di gunakan, oleh sebab itu perlu diadakannya penelitian tentang penerapan *PHP-FPM* dan *HHVM* sebagai *interpreter* kode program *PHP*. Penelitian yang akan di lakukan meliputi benchmark kecepatan kode program *PHP*, kecepatan request *webserver*, dan konsumsi *resource* server yang dihasilkan oleh masing-masing *software* tersebut, sehingga nantinya diharapkan dapat menjawab *interpreter* mana yang akan di gunakan oleh system engineer untuk mengatasi permasalahan yang dialami.

*Kata Kunci: HHVM, PHP-FPM, PHP, Interpreter, Benchmark*

## **ABSTRACT**

*Along with the growth of internet users in the world, offered by internet.org that internet users in the world at this time reached 19.7 billion people and over the years the world's internet users has increased by as much as 7.6% [3]. The owner of the website that the number of visitors continues to grow, need to allocate new servers in order to keep their service running. This has an impact on the costs that should be getting bigger and bigger[4].*

*From the above it can be concluded that the problems of the system design does not develop to support a large number of request and concurrency. Alternatively, a system engineer can convert the system they use with the right technology. One of the solutions to overcome these problems is to use fastCGI. However, in the implementation, fastCGI is not directly in the PHP program to show it. Instead it should use additional applications such as PHP-FPM or HHVM [6]. Additional application serves as an interpreter that connects with fastCGI PHP code that connects the interpreters with machine language.*

*New problems emerged over the large number of new technologies, this led to a flurry of system engineer in determining which system will be in use, therefore it is necessary to hold the research about the application of PHP-FPM and HHVM as the interpreter source code PHP. In doing research that will include benchmark speed, the speed of PHP code request webserver, resource consumption and server produced by each of the software, so that later the hoped answer which interpreters would be in use by the system engineer to resolve problems experienced.*

*Keyword: HHVM, PHP-FPM, PHP, Interpreter, Benchmark*