

SKRIPSI

**ANALISIS DAN PERBANDINGAN KINERJA *RESTFUL API* PADA BAHASA
PEMROGRAMAN *PHP 5*, *PHP 7*, *PYTHON v3.7* DAN *PYTHON v3.8***

Disusun dan diajukan oleh

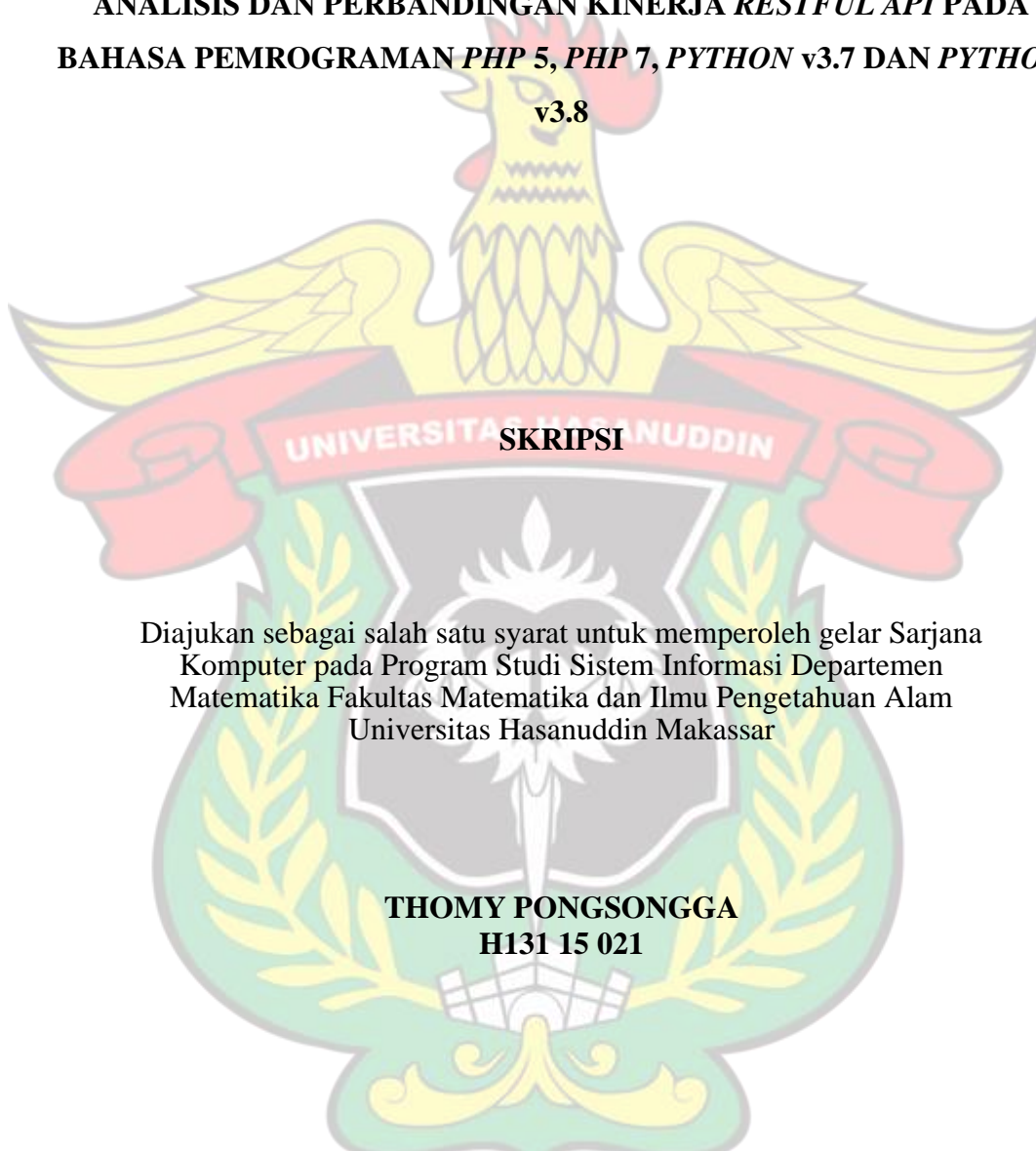
**THOMY PONGSONGGA
H131 15 021**



**PROGRAM STUDI SISTEM INFORMASI
DEPARTEMEN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS HASANUDDIN
MAKASSAR
2021**

**ANALISIS DAN PERBANDINGAN KINERJA *RESTFUL API* PADA
BAHASA PEMROGRAMAN *PHP 5, PHP 7, PYTHON v3.7* DAN *PYTHON***

v3.8



SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana
Komputer pada Program Studi Sistem Informasi Departemen
Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Hasanuddin Makassar

**THOMY PONGSONGGA
H131 15 021**

**PROGRAM STUDI SISTEM
INFORMASI DEPARTEMEN
MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS HASANUDDIN
MAKASSAR**

2021

PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini :

Nama : Thomy Pongsongga

NIM : H13115021

Program Studi : Sistem Informasi

Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul :

Analisis dan Perbandingan Kinerja Restful API pada Bahasa Pemrograman PHP 5,
PHP 7, Python v3.7 dan Python v3.8

Adalah karya tulisan saya sendiri, bukan merupakan pengambilan alihan tulisan orang lain dan bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan isi skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Makassar, 02 Juni 2021



Thomy Pongsongga
Thomy Pongsongga

LEMBAR PENGESAHAN TUGAS AKHIR

**ANALISIS DAN PERBANDINGAN KINERJA RESTFUL API PADA
BAHASA PEMROGRAMAN PHP 5, PHP 7, PYTHON v3.7 DAN PYTHON
v3.8**

Disusun dan diajukan oleh

**THOMY PONGSONGGA
H131 15 021**

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka
Penyelesaian Studi Program Sarjana pada Program Studi Sistem Informasi Fakultas
Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin
Pada tanggal 02 Juni 2021
dan dinyatakan telah memenuhi syarat kelulusan

Menyetujui,

Pembimbing Utama

Dr. Hendra, S.Si, M.Kom.
NIP. 197601022002121001

Pembimbing Pertama

Supri Bm Hj. Amir, S.Si., M.Eng
NIP.198805042019031012

Ketua Program Studi

Dr. Muhammad Hasbi, M.Sc
NIP. 196307201989031003



HALAMAN PENGESAHAN

Skripsi ini diajukan oleh:

Nama : THOMY PONGSONGGA
NIM : H131 15 021
Program Studi : Sistem Informasi
Judul Skripsi : ANALISIS DAN PERBANDINGAN KINERJA RESTFUL API PADA BAHASA PEMROGRAMAN PHP 5, PHP 7, PYTHON V3.7 DAN PYTHON V3.8

Telah dipertahankan di depan Tim Penguji dan diterima sebagai bagian persyaratan untuk memperoleh gelar Sarjana Komputer pada Program Studi Sistem Informasi Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

DEWAN PENGUJI

- | | | Tanda Tangan |
|---------------|--------------------------------------|--------------|
| 1. Ketua | : Dr. Hendra, S.Si, M.Kom. | (.....) |
| 2. Sekretaris | : Supri Bin Hj. Amir, S.Si., M.Eng | (.....) |
| 3. Anggota | : Dr. Eng. Armin Lawi, S.Si., M.Eng. | (.....) |
| 4. Anggota | : Dr. Muhammad Hasbi, M.Sc | (.....) |

Ditetapkan di : Makassar
Tanggal : 02 Juni 2021



KATA PENGANTAR

Segala puji dan syukur penulis panjatkan kehadirat Allah Yang Maha Kuasa karena berkat rahmat serta kehendak-Nya sehingga penulis dapat menyelesaikan penyusunan skripsi yang berjudul “ANALISIS DAN PERBANDINGAN KINERJA RESTFUL API PADA BAHASA PEMROGRAMAN PHP 5, PHP 7, PYTHON V3.7 DAN PYTHON V3.8” sebagai salah satu syarat menyelesaikan studi serta dalam rangka memperoleh gelar sarjana pada program studi Sistem Informasi fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin. Disadari bahwa apa yang disajikan pada proposal penelitian ini masih jauh dari kesempurnaan, oleh karena itu penulis mengharapkan saran dan kritik yang bersifat membangun demi kesempurnaan skripsi ini.

Selama proses pengerjaan skripsi ini, banyak dinamika yang dihadapi penulis. kendala-kendala dan permasalahan yang penulis hadapi, baik mengenai literatur maupun pelaksanaan pengambilan data di lapangan ditengah wabah pandemi Covid-19 serta dalam menganalisis data, tentunya tidak dapat diatasi tanpa bantuan dari berbagai pihak.

Selama proses pengerjaan skripsi ini, penulis telah mendapatkan banyak bantuan dari berbagai pihak, untuk itu pada kesempatan kali ini penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

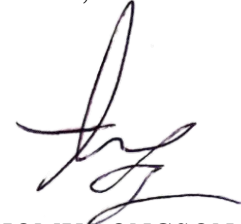
1. Kedua orang tua penulis, Ayahanda **Tarbun Zakaria**, dan Ibunda **Yuliana**, yang selalu mendoakan dan memberi semangat dan kasih sayang yang tak terkira.
2. Rektor Universitas Hasanuddin, Ibu **Prof. Dr. Dwia Aries Tina Pulubuhu** beserta jajarannya.
3. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam, **Dr. Eng. Amiruddin** beserta jajarannya.
4. Ketua Departemen Matematika FMIPA, **dr. Nurdin, S.Si., M.Si**, dan juga **Dr. Muhammad Hasbi, M.Sc** sebagai ketua Program Studi Sistem Informasi Universitas Hasanuddin.

5. Bapak **Dr. Hendra, S.Si., M.Kom** sebagai pembimbing utama yang telah memberikan arahan, ide, motivasi serta dukungan kepada penulis dalam banyak hal.
6. Bapak **Supri Bin Hj. Amir S.Si., M.Eng** sebagai pembimbing pertama yang senantiasa memberikan masukan kepada penulis.
7. Bapak **Dr. Eng. Armin Lawi, S.Si., M.Eng** dan Bapak **Dr. Muhammad Hasbi, M.Sc** sebagai tim penguji atas saran dan masukan pada penelitian yang telah dilakukan oleh penulis.
8. Seluruh Bapak dan Ibu dosen FMIPA Universitas Hasanuddin yang telah mendidik dan memberikan ilmunya sehingga penulis mampu menyelesaikan program sarjana. Serta para staf yang telah membantu dalam pengurusan berkas serta administrasi.
9. Saudara-saudara **Ilkom Boys (Hendrianto, Nur Multazam, Muh. Reskyawan Eka Cipta, Baish Fajar Luthfi, Muh. Emir Toripuji Abas, Ahmad Fikri Aslam Suharto, Muh. Ikshan, Gidion Pennikay, James Songli Patandianan, Adrianto Sapurta Mansur, Adrian Saputra, Achmad Muchtadin, Muh. Fachrul Riza, Andi Muhammad Asdar, Bima Satria Yuda Muh., Arjudi Antonius, Muh. Anugrah Trisnanda, Muh. Iskandar Zulkarnaen Tajuddin, Muh. Hasbi Hasnan Habib, Muh. Anwar Sadad Muhimmi, Ilham, Edi Anugrah, Muh Aminuddin, Syaheruddin, Faatir Anshary)** yang telah menemani penulis selama perkuliahan, saling memberi motivasi dan bantuan, meluangkan waktu dan berbagi suka-duka serta kebersamaan selama menuntut ilmu.
10. Keluarga besar **Ilmu Komputer Unhas 2015** yang setia menemani dan membantu penulis selama menjalani pendidikan. Serta kakak-kakak dan adik-adik **Ilmu Komputer 2014, 2016, 2017, 2018, 2019.2020** yang telah banyak membantu, semoga tetap semangat dalam mengejar impian.
11. Rekan-rekan **KKN Uhnas Sinjai Gel. 101** yang telah menjadi keluarga baru selama KKN dan menjadikan KKN sebagai momen yang berkesan.

12. Serta semua pihak yang telah banyak berpartisipasi, baik baik secara langsung maupun secara tidak langsung dalam penyusunan skripsi ini yang tidak sempat penulis sebutkan satu per satu.

Akhirnya penulis berharap, semoga skripsi ini dapat memberikan manfaat dan dapat menambah wawasan bagi penulis dan bagi pembaca umumnya. Semoga Allah Yang Maha Kuasa senantiasa membimbing melimpahkan rahmatnya bagi kita semua.

Makassar, 02 Juni 2021



THOMY PONGSONGGA

NIM. H13115021

**PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK
KEPENTINGAN AKADEMIS**

Sebagai civitas akademik Universitas Hasanuddin, saya yang bertanda tangan di bawah ini:

Nama : THOMY PONGSONGGA
NIM : H131 15 021
Program Studi : Sistem Informasi
Departemen : Matematika
Fakultas : Matematika dan Ilmu Pengetahuan Alam
Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Hasanuddin **Hak Prediktor Royalti Noneksklusif** (*Non-exclusive Royalty-Free Right*) atas tugas akhir saya yang berjudul:

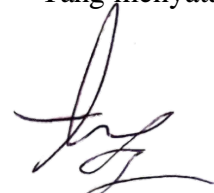
**" ANALISIS DAN PERBANDINGAN KINERJA RESTFUL API
PADA BAHASA PEMROGRAMAN PHP 5, PHP 7, PYTHON
v3.7 DAN PYTHON v3.8 "**

beserta perangkat yang ada (jika diperlukan). Terkait dengan hal diatas, maka pihak Universitas Hasanuddin berhak menyimpan, mengalih-media/format-kan, mengola dalam bentuk pangkalan data (*database*), merawat dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Dibuat di Makassar pada 02 Juni 2021

Yang menyatakan



(THOMY PONGSONGGA)

ABSTRAK

Kecepatan aplikasi server menjadi semakin penting agar dapat menangani banyak permintaan. Saat ini terdapat beberapa bahasa pemrograman yang populer dalam pengembangan aplikasi web seperti *PHP* dan *Python*. *Application Programming Interface* (API) merupakan antarmuka yang dibangun oleh pengembang sistem supaya sebagian atau keseluruhan fungsi sistem dapat diakses secara programatis. Sementara *Representational State Transfer* (REST) merupakan salah satu gaya arsitektur dari pengembangan API yang menggunakan *Hypertext Transfer Protocol* (HTTP) dalam melakukan komunikasi data. Penelitian ini menganalisis dan membandingkan kinerja *REST API* pada Bahasa Pemrograman *PHP 5*, *PHP 7*, *Python v3,7* dan *Python v3.8*. Perbandingan kinerja dilakukan dengan cara mengukur kecepatan respon time pada hasil benchmark setiap bahasa pemrograman. Pengukuran dilakukan dengan dua metode, yaitu *GET method* dan *POST method* (Input dan Output data) dengan menggunakan data sampel 2000 data dummy. Pada bahasa pemrograman *PHP* menggunakan CodeIgniter sebagai framework dengan menggunakan web server Apache dan framework Flask pada *Python* dengan menggunakan web server Apache.

Kata kunci : *PHP, Python, REST API, Get, Post*

ABSTRACT

Server application speed is becoming increasingly important in order to handle multiple requests. Currently there are several popular programming language in the development of web applications such as *PHP* and *Python*. *Application Programming Interface* (API) is an interface built by system developers so that part or all of the system's functions can be accessed programmatically. Meanwhile, *Representational State Transfer* (REST) is one of the architectural styles of API development that uses *Hypertext Transfer Protocol* (HTTP) to communicate data. This study analyzes and compares the performance of the *REST API* in the programming languages *PHP 5*, *PHP 7*, *Python v3,7* and *Python v3.8*. The performance comparison is done by measuring the response time speed of the benchmark results of each programming language. Measurements were carried out by two methods, namely the *GET method* and the *POST method* (Input and Output data) using a sample of 2000 dummy data. The programming *PHP language* uses CodeIgniter as a framework using the Apache web server and the Flask framework in *Python* using the Apache web server.

Keywords : *PHP, Python, REST API, Get, Post*

DAFTAR ISI

HALAMAN JUDUL.....	i
PERNYATAAN KEASLIAN.....	ii
HALAMAN PERSETUJUAN PEMBIMBING	iii
HALAMAN PENGESAHAN.....	iv
KATA PENGANTAR	v
PERSETUJUAN PUBLIKASI KARYA ILMIAH.....	viii
ABSTRAK	ix
ABSTRACT.....	x
DAFTAR ISI.....	xi
DAFTAR GAMBAR	xiii
DAFTAR TABEL.....	xvi
DAFTAR LAMPIRAN.....	xviii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan Penelitian	2
1.4 Manfaat Penelitian	2
1.5 Batasan Masalah	2
BAB II TINJAUAN PUSTAKA.....	3
2.1 Landasan Teori	3
2.1.1 <i>Web</i> Aplikasi	3
2.1.2 <i>Restful</i> API	4
2.1.3 PHP	5
2.1.4 CodeIgniter.....	7
2.1.5 MySQL.....	8
2.1.6 Python	10
2.1.7 Flask	12
2.2 Kerangka Konseptual.....	13
BAB III METODE PENELITIAN.....	14
1.1 Tahapan Penelitian.....	14

1.2 Waktu dan Tempat Penelitian.....	15
1.3 Sumber Data	15
1.4 Instrumen Penelitian	15
BAB IV HASIL DAN PEMBAHASAN	16
4.1 Tahapan Penelitian.....	16
4.1.1 Restful API.....	16
4.1.2 Pengaturan Software	19
4.1.2.1 Postman	19
4.1.2.2 Apache Benchmark	20
4.2 Hasil Percobaan	21
4.2.1 PHP	21
4.2.2 Python	37
BAB V.....	66
KESIMPULAN DAN SARAN.....	66
5.1 Kesimpulan	66
5.2 Saran.....	67
DAFTAR PUSTAKA	68
LAMPIRAN.....	69

DAFTAR GAMBAR

Gambar 2.1 Alur kerja web server	3
Gambar 2.2 Alur kerja framework CodeIgniter	8
Gambar 2.3 Kerangka konseptual	13
Gambar 3.1 Flowchart.....	14
Gambar 4.1 GET Method.....	16
Gambar 4.2 POST Method.....	17
Gambar 4.3 Koneksi ke database	17
Gambar 4.4 GET Method.....	18
Gambar 4.5 POST Method.....	18
Gambar 4.6 Koneksi ke database	18
Gambar 4.7 Tampilan dari aplikasi Postman	19
Gambar 4.8 Tampilan dari Apache Benchmark.....	20
Gambar 4.9 Perbandingan Time per Request pada GET Method antara PHP 5 dan PHP 7	23
Gambar 4.10 Perbandingan Penggunaan RAM pada GET Method antara PHP 5 dan PHP 7	24
Gambar 4.11 Perbandingan Penggunaan CPU pada GET Method antara PHP 5 dan PHP 7	25
Gambar 4.12 Perbandingan Time Taken pada GET Method antara PHP 5 dan PHP 7.....	26
Gambar 4.13 Perbandingan Request per Second pada GET Method antara PHP 5 dan PHP 7.....	27
Gambar 4.14 Perbandingan Transfer Rate pada GET Method antara PHP 5 dan PHP 7.....	28
Gambar 4.15 Perbandingan Time per Request pada POST Method antara PHP 5 dan PHP 7.....	31
Gambar 4.16 Perbandingan Penggunaan RAM pada POST Method antara PHP 5 dan PHP 7.....	32
Gambar 4.17 Perbandingan Penggunaan CPU pada POST Method antara PHP 5 dan PHP 7.....	33
Gambar 4.18 Perbandingan Time Taken pada POST Method antara PHP 5 dan PHP 7.....	34
Gambar 4.19 Perbandingan Request per Second pada POST Method antara PHP 5 dan PHP 7.....	35
Gambar 4.20 Perbandingan Transfer Rate pada POST Method antara PHP 5 dan PHP 7	36
Gambar 4.21 Perbandingan Time per Request pada GET Method antara Python v3.7 dan Python v3.8.....	39

Gambar 4.22 Perbandingan Penggunaan RAM pada GET Method antara Python v3.7 dan Python v3.8.....	40
Gambar 4.23 Perbandingan Penggunaan CPU pada GET Method antara Python v3.7 dan Python v3.8.....	41
Gambar 4.24 Perbandingan Time Taken pada GET Method antara Python v3.7 dan Python v3.8	42
Gambar 4.25 Perbandingan Request per Second pada GET Method antara Python v3.7 dan Python v3.8.....	43
Gambar 4.26 Perbandingan Transfer Rate pada GET Method antara Python v3.7 dan Python v3.8.....	44
Gambar 4.27 Perbandingan Time per Request pada POST Method antara Python v3.7 dan Python v3.8.....	47
Gambar 4.28 Perbandingan Penggunaan RAM pada POST Method antara Python v3.7 dan Python v3.8.....	48
Gambar 4.29 Perbandingan Penggunaan CPU pada POST Method antara Python v3.7 dan Python v3.8.....	49
Gambar 4.30 Perbandingan Time Taken pada POST Method antara Python v3.7 dan Python v3.8.....	50
Gambar 4.31 Perbandingan Request per Second pada POST Method antara Python v3.7 dan Python v3.8.....	51
Gambar 4.32 Perbandingan Transfer Rate pada POST Method antara Python v3.7 dan Python v3.8.....	52
Gambar 4.33 Perbandingan Time per Request pada GET Method antara PHP 5, PHP 7, Python v3.7 dan Python v 3.8	54
Gambar 4.34 Perbandingan Penggunaan RAM pada GET Method antara PHP 5, PHP 7, Python v3.7 dan Python v 3.8	55
Gambar 4.35 Perbandingan Penggunaan CPU pada GET Method antara PHP 5, PHP 7, Python v3.7 dan Python v 3.8	56
Gambar 4.36 Perbandingan Time Taken pada GET Method antara PHP 5, PHP 7, Python v3.7 dan Python v 3.8	57
Gambar 4.37 Perbandingan Request per Second pada GET Method antara PHP 5, PHP 7, Python v3.7 dan Python v 3.8	58
Gambar 4.38 Perbandingan Transfer Rate pada GET Method antara PHP 5, PHP 7, Python v3.7 dan Python v 3.8	59
Gambar 4.39 Perbandingan Time per Request pada POST Method antara PHP 5, PHP 7, Python v3.7 dan Python v3.8	60
Gambar 4.40 Perbandingan Penggunaan RAM pada POST Method antara PHP 5, PHP 7, Python v3.7 dan Python v 3.8	61
Gambar 4.41 Perbandingan Penggunaan CPU pada POST Method antara PHP 5, PHP 7, Python v3.7 dan Python v 3.8	62
Gambar 4.42 Perbandingan Time Taken pada POST Method antara PHP 5, PHP 7, Python v3.7 dan Python v 3.8	63

Gambar 4.43 Perbandingan Request per Second pada POST Method antara PHP 5, PHP 7, Python v3.7 dan Python v 3.8	64
Gambar 4.44 Perbandingan Transfer Rate pada POST Method antara PHP 5, PHP 7, Python v3.7 dan Python v 3.8	65

DAFTAR TABEL

Tabel 4.1 Hasil benchmark GET Method pada PHP 5	22
Tabel 4.2 Hasil benchmark GET Method pada PHP 7	22
Tabel 4.3 Perbandingan Time per Request pada GET Method antara PHP 5 dan PHP 7	23
Tabel 4.4 Perbandingan Penggunaan RAM pada GET Method antara PHP 5 dan PHP 7	24
Tabel 4.5 Perbandingan Penggunaan CPU pada GET Method antara PHP 5 dan PHP 7	25
Tabel 4.6 Perbandingan Time Taken pada GET Method antara PHP 5 dan PHP 7	26
Tabel 4.7 Perbandingan Request per Second pada GET Method antara PHP 5 dan PHP 7	27
Tabel 4.8 Perbandingan Transfer Rate pada GET Method antara PHP 5 dan PHP 7	28
Tabel 4.9 Hasil benchmark POST Method pada PHP 5.	30
Tabel 4.10 Hasil benchmark POST Method pada PHP 7.	30
Tabel 4.11 Perbandingan Time per Request pada POST Method antara PHP 5 dan PHP 7	31
Tabel 4.12 Perbandingan Penggunaan RAM pada POST Method antara PHP 5 dan PHP 7	32
Tabel 4.13 Perbandingan Penggunaan CPU pada POST Method antara PHP 5 dan PHP 7	33
Tabel 4.14 Perbandingan Time Taken pada POST Method antara PHP 5 dan PHP 7.....	34
Tabel 4.15 Perbandingan Reques per Second pada POST Method antara PHP 5 dan PHP 7	35
Tabel 4.16 Perbandingan Transfer Rate pada POST Method antara PHP 5 dan PHP 7.....	36
Tabel 4.17 Hasil benchmark GET Method pada Python v3.7	38
Tabel 4.18 Hasil benchmark GET Method pada Python v3.8.	38
Tabel 4.19 Perbandingan Time per Request pada GET Method antara Python v3.7 dan Python v3.8.....	39
Tabel 4.20 Perbandingan Penggunaan RAM pada GET Method antara Python v3.7 dan Python v3.8.....	40
Tabel 4.21 Perbandingan Penggunaan CPU pada GET Method antara Python v3.7 dan Python v3.8.....	41
Tabel 4.22 Perbandingan Time Taken pada GET Method antara Python v3.7 dan Python v3.8	42

Tabel 4.23 Perbandingan Request per Second pada GET Method antara Python v3.7 dan Python v3.8.....	43
Tabel 4.24 Perbandingan Transfer Rate pada GET Method antara Python v3.7 dan Python v3.8	44
Tabel 4.25 Hasil benchmark POST Method pada Python v3.7	46
Tabel 4.26 Hasil benchmark POST Method pada Python v3.8	46
Tabel 4.27 Perbandingan Time per Request pada POST Method antara Python v3.7 dan Python v3.8.....	47
Tabel 4.28 Perbandingan Penggunaan RAM pada POST Method antara Python v3.7 dan Python v3.8.....	48
Tabel 4.29 Perbandingan Penggunaan CPU pada POST Method antara Python v3.7 dan Python v3.8.....	49
Tabel 4.30 Perbandingan Time Taken pada POST Method antara Python v3.7 dan Python v3.8	50
Tabel 4.31 Perbandingan Request per Second pada POST Method antara Python v3.7 dan Python v3.8.....	51
Tabel 4.32 Perbandingan Transfer Rate pada POST Method antara Python v3.7 dan Python v3.8	52
Tabel 4.33 Perbandingan Time per Request pada GET Method antara PHP 5, PHP 7, Python v3.7 dan Python v 3.8	53
Tabel 4.34 Perbandingan Penggunaan RAM pada GET Method antara PHP 5, PHP 7, Python v3.7 dan Python v 3.8	54
Tabel 4.35 Perbandingan Penggunaan CPU pada GET Method antara PHP 5, PHP 7, Python v3.7 dan Python v 3.8	56
Tabel 4.36 Perbandingan Time Taken pada GET Method antara PHP 5, PHP 7, Python v3.7 dan Python v 3.8	57
Tabel 4.37 Perbandingan Request per Second pada GET Method antara PHP 5, PHP 7, Python v3.7 dan Python v 3.8	58
Tabel 4.38 Perbandingan Transfer Rate pada GET Method antara PHP 5, PHP 7, Python v3.7 dan Python v 3.8	59
Tabel 4.39 Perbandingan Time per Request pada POST Method antara PHP 5, PHP 7, Python v3.7 dan Python v3.8	60
Tabel 4.40 Perbandingan Penggunaan RAM pada POST Method antara PHP 5, PHP 7, Python v3.7 dan Python v 3.8	61
Tabel 4.41 Perbandingan Penggunaan CPU pada POST Method antara PHP 5, PHP 7, Python v3.7 dan Python v 3.8	62
Tabel 4.42 Perbandingan Time Taken pada POST Method antara PHP 5, PHP 7, Python v3.7 dan Python v 3.8	63
Tabel 4.43 Perbandingan Request per Second pada POST Method antara PHP 5, PHP 7, Python v3.7 dan Python v 3.8.....	64
Tabel 4.44 Perbandingan Transfer Rate pada POST Method antara PHP 5, PHP 7, Python v3.7 dan Python v 3.8	65

DAFTAR LAMPIRAN

Lampiran 1. Sintaks PHP GET Method.....	69
Lampiran 2. Sintaks PHP POST Method.....	70
Lampiran 3. Sintaks PHP koneksi ke database	70
Lampiran 4. Sintaks Python GET Method.....	71
Lampiran 5. Sintaks Python POST Method.....	72
Lampiran 6. Sintaks Python koneksi ke database	72

BAB I PENDAHULAN

1.1 Latar Belakang

Aplikasi berbasis web adalah suatu aplikasi yang diakses menggunakan penjelajah web melalui suatu jaringan seperti Internet atau intranet. Ia juga merupakan suatu aplikasi perangkat lunak komputer yang dikodekan dalam bahasa yang didukung penjelajah web dan bergantung pada penjelajah tersebut untuk menampilkan aplikasi. Aplikasi web menjadi populer karena kemudahan tersedianya aplikasi klien untuk mengaksesnya, penjelajah web, yang kadang disebut sebagai suatu thin client (klien tipis). Kemampuan untuk memperbarui dan memelihara aplikasi web tanpa harus mendistribusikan dan menginstalasi perangkat lunak pada kemungkinan ribuan komputer klien merupakan alasan kunci popularitasnya.

Dalam rangka menangani peningkatan jumlah pengguna yang terhubung ke internet, teknologi server web terus dikembangkan. Salah satu teknologi server web yang dimaksud adalah bahasa pemrograman server. Tidak hanya untuk meningkatkan kemudahan pembuatan aplikasi, tetapi juga untuk meningkatkan kinerja dari aplikasi tersebut.

Kecepatan aplikasi server menjadi semakin penting. Aplikasi yang kinerjanya baik adalah aplikasi yang dapat menangani banyak permintaan dari pengguna tanpa mengkonsumsi banyak sumber daya komputasi dari perangkat server, dan dengan demikian menekan biaya yang harus dikeluarkan oleh pengelola aplikasi tersebut. Saat ini terdapat beberapa bahasa pemrograman yang populer dalam pengembangan aplikasi web seperti *PHP* dan *Python*.

Penelitian ini bertujuan menganalisis serta membandingkan kinerja *Restful API* pada bahasa pemrograman *PHP* dan *Python*. Diharapkan penelitian ini dapat membantu para pengambil keputusan dalam menentukan bahasa pemrograman yang tepat dalam rangka membangun sebuah aplikasi web, khususnya aplikasi yang berbasis Representational State Transfer (*REST*).

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas maka yang menjadi rumusan masalah dalam penelitian ini yaitu membandingkan hasil kinerja *Restful API* untuk Get dan Post method (Input/Output data) dengan perbandingan hasil benchmark dari bahasa pemrograman *PHP 5*, *PHP 7*, *Python v3.7* dan *Python v3.8*.

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut:

1. Mengetahui keuntungan dari penerapan *Restful API* pada bahasa pemrograman *PHP* dan *Python* dalam mengelolah Input/Output data.
2. Menganalisis bagaimana kinerja *Restful API* pada bahasa pemrograman *PHP* dan *Python*.
3. Mengetahui perbedaan kecepatan kinerja dalam mengelolah Input/Output data pada *Restful API* antara bahasa pemrograman *PHP* dengan *Python*.
4. Menganalisis kehandalan kinerja *Restful API* pada *PHP* dan *Python* melalui hasil *benchmark*.

1.4 Manfaat Penelitian

Hasil dari penelitian ini diharapkan dapat memberikan manfaat dengan adanya penelitian ini agar dapat membedakan kecepatan dalam pengelolaan data *Restful API* pada *PHP* dan *Python* untuk memudahkan pengambilan keputusan dalam menentukan bahasa pemrograman untuk mengelolah Input/Output data yang lebih cepat dalam rangka membangun sebuah aplikasi.

1.5 Batasan Masalah

Adapun batasan masalah yang dilakukan agar pengerjaan tugas akhir ini tidak terlalu luas. Maka masalah yang akan dibahas dibatasi pada:

1. Penelitian ini berfokus membandingkan kinerja *Restful API* pada bahasa pemrograman yaitu *PHP* dan *Python* dalam mengelolah Input/Output data.
2. Penelitian ini hanya menggunakan framework CodeIgniter pada *PHP* dengan menggunakan web server Apache dan framework Flask pada *Python* dengan menggunakan web server Apache

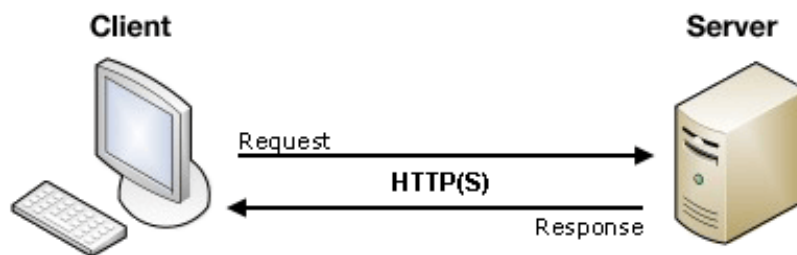
BAB II TINJAUAN PUSTAKA

2.1 Landasan Teori

2.1.1 Web Aplikasi

Pada awalnya aplikasi *web* dibangun dengan hanya menggunakan Bahasa yang disebut HTML (*Hyper Text Markup Language*). Pada perkembangan berikutnya, sejumlah skrip dan objek dikembangkan untuk memperluas kemampuan HTML. Web Aplikasi adalah adalah suatu aplikasi yang diakses menggunakan penjelajah web melalui suatu jaringan seperti Internet atau intranet. Ia juga merupakan suatu aplikasi perangkat lunak komputer yang dikodekan dalam bahasa yang didukung penjelajah web (seperti ASP, Perl, Java, Java Script, PHP, Python, Ruby, dll) dan bergantung pada penjelajah tersebut untuk menampilkan aplikasi.

Fungsi utama dari aplikasi web yaitu digunakan untuk mengolah data secara cepat melalui jaringan internet. Dimana nantinya pengguna bisa mengakses data atau informasi yang mereka butuhkan melalui smartphone, laptop, dan pc dengan syarat perangkatnya harus sudah terinstall browser dan terkoneksi ke jaringan internet atau local. Jika di perhatikan pada Gambar 2.1 terdapat 2 bagian penting daripada aplikasi web, yaitu server dan client. (Guntoro, 2019).



Gambar 2.1 Alur kerja web server

- Client: Komputer, PC, Laptop, dan smartphone yang terhubung ke sebuah jaringan bisa dikategorikan sebagai client pada web app. Perangkat-perangkat tersebut bisa mengakses aplikasi web melalui peramban (browser) contohnya google chrome, opera, dan lainnya.

- **Server:** Merupakan perangkat komputer dengan spesifikasi tertentu dan bertugas menyimpan data-data atau informasi dari sebuah aplikasi web sehingga client bisa dengan mudah mengaksesnya.

2.1.2 *Restful API*

RESTful API merupakan implementasi dari API (*Application Programming Interface*). REST (*Representational State Transfer*) adalah suatu arsitektur metode komunikasi yang menggunakan protokol HTTP untuk pertukaran data dan metode ini sering diterapkan dalam pengembangan aplikasi. Dimana tujuannya adalah untuk menjadikan sistem yang memiliki performa yang baik, cepat dan mudah untuk di kembangkan (*scale*) terutama dalam pertukaran dan komunikasi data (Irsan, 2018).

RESTFUL API memiliki 4 komponen penting diantaranya adalah (Alam, 2018):

- *URL DESIGN*
- *RESTful API* diakses menggunakan protokol HTTP. Penamaan dan struktur URL yang konsisten akan menghasilkan API yang baik dan mudah untuk dimengerti developer. URL API biasa disebut endpoint dalam pemanggilannya.
- *HTTP Verbs*

Setiap request yang dilakukan terdapat metode yang dipakai agar server mengerti apa yang sedang di *request client*

a) GET

GET adalah metode HTTP *Request* yang paling mudah, metode ini digunakan untuk membaca atau mendapatkan data dari sumber.

b) POST

POST adalah metode HTTP *Request* yang digunakan untuk membuat data baru dengan menyisipkan data dalam *body* saat *request* dilakukan.

c) PUT

PUT adalah metode HTTP *Request* yang biasanya digunakan untuk melakukan *update data resource*.

d) DELETE

DELETE adalah metode HTTP *Request* yang digunakan untuk menghapus suatu data pada *resource*.

- *HTTP Response Code*

HTTP *response code* adalah kode standarisasi dalam menginformasikan hasil request kepada *client*. Secara umum terdapat 3 kelompok yang biasa kita jumpai pada RESTful API yaitu:

- a. 2XX: adalah response code yang menampilkan bahwa request berhasil.
- b. 4XX: adalah response code yang menampilkan bahwa request mengalami kesalahan pada sisi client.
- c. 5XX: adalah response code yang menampilkan bahwa request mengalami kesalahan pada sisi server.

- *Format Response*

Setiap *request* yang dilakukan *client* akan menerima data *response* dari server, response tersebut biasanya berupa data XML ataupun JSON. Setelah mendapatkan data response tersebut barulah client bisa menggunakannya dengan cara memarsing data tersebut dan diolah sesuai kebutuhan.

2.1.3 PHP

PHP adalah bahasa pemrograman yang sering disisipkan ke dalam HTML. PHP sendiri berasal dari kata *Hypertext Preprocessor*. Sejarah PHP pada awalnya merupakan kependekan dari *Personal Home Page* (Situs personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama *Form Interpreted* (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari web.

Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI. Dengan perilisan kode sumber ini menjadi sumber terbuka, maka banyak pemrograman yang tertarik untuk ikut mengembangkan PHP. Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini, interpreter PHP sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan. Pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang interpreter PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1998,

perusahaan tersebut merilis interpreter baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0 dan singkatan PHP diubah menjadi akronim berulang PHP : *Hypertext Preprocessing*.

Pada pertengahan tahun 1999, Zend merilis interpreter PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi. Pada Juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek. Versi terbaru dari bahasa pemrograman PHP adalah versi 5.6.4 yang resmi dirilis pada tanggal 18 Desember 2014 yang kemudian di perbarui lagi dengan versi teranyar yaitu PHP 7 yang di 17 Desember 2015.

Bahasa pemrograman ini menggunakan sistem *server-side*. *Server-side programming* adalah jenis bahasa pemrograman yang nantinya *script/program* tersebut akan dijalankan/diproses oleh server. Kelebihannya adalah mudah digunakan, sederhana, dan mudah untuk dimengerti dan dipelajari. Semenjak PHP menjadi bahasa pemrograman yang open source, pengembang tidak perlu menunggu sampai dengan *update* terbaru rilis.

Pengguna PHP akan lebih baik jika menggunakan versi terbaru. Sehingga jika ada rilis terbaru Anda harus menyesuaikan sistem Anda dengan versi PHP yang paling baru. Meskipun harus menggunakan versi terbaru, biaya untuk maintenance dan web *development* sangat terjangkau.

Bahasa pemrograman PHP membantu Anda untuk mengembangkan aplikasi berbasis web yang cukup kompleks, handal, dan cepat. Tergantung dari spesifikasi bisnis, penggunaan hosting, tingkat pengalaman, kebutuhan aplikasi, dan pengembangan timeframe. Selain itu ada banyak PHP frameworks yang dapat Anda pilih. Untuk membuat sistem berbasis web tidak harus menggunakan PHP. Namun karena fiturnya yang menarik dan dirasa memudahkan, maka kebanyakan pengembangan menggunakan bahasa pemrograman ini.

Sebelum ada PHP, pengembang kebanyakan menggunakan bahasa pemrograman HTML. Bahasa pemrograman ini sudah dapat membuat tampilan website, akan tetapi tampilan yang dihasilkan masih biasa saja. Biasa saja karena website masih bersifat statis atau tetap, tidak bisa menyesuaikan tampilan dengan kondisi pengguna.

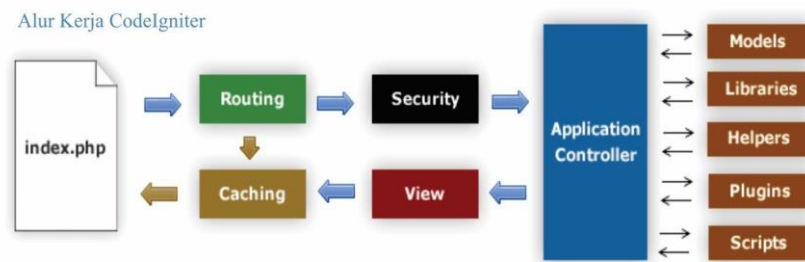
Semenjak adanya PHP maka pembuatan website dapat dikembangkan dan diatur agar menjadi website yang dinamis. Dimana website dapat menyesuaikan tampilan sesuai dengan perangkat yang digunakan oleh pengguna. Tidak hanya itu, website juga dapat melakukan input output dengan database. PHP sendiri biasanya disisipkan ke dalam kode HTML dan dikombinasikan dengan bahasa pemrograman lain, seperti CSS (Rini Sovia, 2011).

2.1.4 CodeIgniter

Codeigniter adalah sebuah framework untuk web yang dibuat dalam format PHP. Format yang dibuat ini selanjutnya dapat digunakan untuk membuat sistem aplikasi web yang kompleks. Codeigniter dapat mempercepat proses pembuatan web karena semua class dan modul yang dibutuhkan sudah ada dan programmer hanya tinggal menggunakannya kembali pada aplikasi web yang akan dibuat dan pada Gambar 2.2 akan di jelaskan bagaimana alur kerja dari framework CodeIgniter.

Berikut ini adalah fitur-fitur penting yang ada di *CodeIgniter* (Dina, 2018):

- Sistem berbasis *Model-View-Controller* (MVC)
- Sangat Ringan dan Butuh Sedikit Sumber Daya (Memori + CPU)
- Mensupport berbagai macam database dan kaya fitur
- Mendukung *Query Builder* untuk mengakses database
- Validasi Form dan Data
- Mengamankan website dari XSS (*Cross Site Scripting*)
- *Manajemen Session*



Gambar 2.2 Alur kerja framework CodeIgniter

- *Index.php*: *Index.php* disini berfungsi sebagai file pertama dalam program yang akan dibaca oleh program.
- *The Router*: Router akan memeriksa HTTP request untuk menentukan hal apa yang harus dilakukan oleh program.
- *Cache File*: Apabila dalam program sudah terdapat “*cache file*” maka file tersebut akan langsung dikirim ke browser. File cache inilah yang dapat membuat sebuah website dapat di buka dengan lebih cepat. Cache file dapat melewati proses yang sebenarnya harus dilakukan oleh program codeigniter.
- *Security*: Sebelum file controller di *load* keseluruhan, HTTP *request* dan data yang disubmit oleh *user* akan disaring terlebih dahulu melalui fasilitas security yang dimiliki oleh *codeigniter*.
- *Controller*: Controller akan membuka file model, *core libraries*, *helper* dan semua *resources* yang dibutuhkan dalam program tersebut.
- *View*: Hal yang terakhir akan dilakukan adalah membaca semua program yang ada dalam view file dan mengirimkannya ke browser supaya dapat dilihat. Apabila *file view* sudah ada yang di “*cache*” maka *file view* baru yang belum ter-*cache* akan mengupdate file *view* yang sudah ada (Prabowo, 2015).

2.1.5 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (database management system) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL. Tidak sama dengan proyek-proyek seperti Apache, dimana perangkat lunak dikembangkan oleh

komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing (Anggaeni dan Sujatmiko, 2013).

SQL merupakan bahasa pemrograman yang perlu Anda pahami karena dapat merelasikan antara beberapa tabel dengan database maupun antar database. Ada tiga bentuk SQL yang perlu Anda ketahui, yaitu Data Definition Language (DDL), Data Manipulation Language (DML), dan Data Control Language (DCL) (Yasin, 2019).

- *Data Definition Language (DDL)*

DDL berguna pada saat Anda ingin mendefinisikan data di dalam database. Terdapat beberapa query yang dikelompokkan ke dalam DDL, yaitu:

Argumen DDL di atas perlu Anda pahami karena merupakan dasar penggunaan SQL di bagian awal pembuatan database. Contohnya saja jika belum menjalankan perintah “CREATE”, Anda belum bisa melanjutkan penggunaan argumen yang lainnya.

- *Data Manipulation Language (DML)*

DML dapat Anda pakai setelah menjalankan perintah DDL. DML berfungsi untuk memanipulasi, mengubah, atau mengganti isi dari database yang sudah ada.

INSERT : Dipakai untuk memasukkan data ke dalam tabel pada database.

UPDATE : Dipakai untuk mengubah data yang ada di dalam tabel pada database.

DELETE : Dipakai untuk menghapus data di dalam tabel pada database.

- *Data Control Language (DCL)*

Jika Anda sudah mempunyai user dan ingin mengatur hak akses masing-masing user, Anda sebaiknya memahami berbagai macam jenis DCL dan cara penggunaannya. DCL berguna untuk memberikan hak akses database, mendefinisikan space, mengalokasikan space, dan melakukan audit penggunaan database.

GRANT : Dipakai untuk memberikan izin kepada user untuk mengakses database.

REVOKE : Dipakai untuk membatalkan izin user untuk mengakses database.

COMMIT : Dipakai untuk menetapkan penyimpanan pada database.

ROOLBACK : Dipakai untuk membatalkan penyimpanan pada database.

Sebagai suatu pengelola database terbesar dan paling banyak digunakan tentunya MySQL ini memiliki fitur atau kapabilitas tertentu. Salah satu yang paling dicari oleh para pengguna MySQL adalah kemampuannya yang multi-platform dan berlisensi GPL, sehingga dapat digunakan oleh komputer hampir di semua OS. Kinerjanya juga dianggap cukup tinggi dalam hal memproses query – query yang ada meskipun masih terbatas pada database dalam jumlah tertentu.

Beberapa fitur lain yang ada pada MySQL saat ini tersedianya tipe data yang sangat beragam seperti Float, Double, Char, Date dan lain – lain. MySQL juga mendukung penggunaan field sebagai index serta memiliki tingkat keamanan yang cukup bagus dengan adanya Subnetmask, nama Host serta sandi yang terenkripsi (Hermawan, 2019).

2.1.6 Python

Python adalah sebuah bahasa pemrograman dinamis yang sering digunakan dalam pengembangan aplikasi pada berbagai domain. Menurut Hall (2005, p10) Python pada dasarnya memiliki *style* yang serupa dengan *pseuocode*, perbedaannya hanyalah Python dapat dijalankan di komputer dan menampilkan hasil. Bahasa Python mudah dimengerti (serupa dengan bahasa Inggris), dan tidak perlu *compiling*, maka dari itu kita dapat menjalankan *script* Python begitu kita menyimpannya tanpa memerlukan langkah lanjutan.

Python diciptakan oleh Guido van Rossum pertama kali di *Centrum Wiskunde & Informatica* (CWI) di Belanda pada awal tahun 1990-an. Bahasa python terinspirasi dari bahasa pemrograman ABC. Sampai sekarang, Guido masih menjadi penulis utama untuk python, meskipun bersifat *open source* sehingga ribuan orang juga berkontribusi dalam mengembangkannya. Di tahun 1995, Guido melanjutkan pembuatan python di *Corporation for National Research Initiative* (CNRI) di Virginia Amerika, di mana dia merilis beberapa versi dari python. Pada Mei 2000, Guido dan tim Python pindah ke BeOpen.com dan membentuk tim BeOpen PythonLabs. Di bulan Oktober pada tahun yang

sama, tim python pindah ke Digital Creation (sekarang menjadi Perusahaan Zope). Pada tahun 2001, dibentuklah Organisasi Python yaitu *Python Software Foundation* (PSF). PSF merupakan organisasi nirlaba yang dibuat khusus untuk semua hal yang berkaitan dengan hak intelektual Python. Perusahaan Zope menjadi anggota sponsor dari PSF. Berikut adalah versi mayor dan minor python berikut tanggal rilisnya.

- Python 1.0 – Januari 1994
- Python 1.2 – 10 April 1995
- Python 1.3 – 12 Oktober 1995
- Python 1.4 – 25 Oktober 1996
- Python 1.5 – 31 Desember 1997
- Python 1.6 – 5 September 2000
- Python 2.0 – 16 Oktober 2000
- Python 2.1 – 17 April 2001
- Python 2.2 – 21 Desember 2001
- Python 2.3 – 29 Juli 2003
- Python 2.4 – 30 Nopember 2004
- Python 2.5 – 19 September 2006
- Python 2.6 – 1 Oktober 2008
- Python 2.7 – 3 Juli 2010
- Python 3.0 – 3 Desember 2008
- Python 3.1 – 27 Juni 2009
- Python 3.2 – 20 Februari 2011
- Python 3.3 – 29 September 2012
- Python 3.4 – 16 Maret 2014
- Python 3.5 – 13 September 2015
- Python 3.6 – 23 Desember 2016
- Python 3.7 – 27 Juni 2018
- Python 3.8 – 14 Oktober 2019
- Python 3.9 – 5 Oktober 2020

Python merupakan salah satu bahasa pemrograman dan penunjang pada perkembangan FOSS dan Linux. Tidak hanya kedua itu, di

perkembangan teknologi lainnya, Python hadir sebagai salah satu teknologi mumpuni yang patut kita pelajari. Biasanya Python sudah tertanam di distro - distro tertentu seperti Ubuntu, Fedora, dan Slackware. Python sendiri merupakan bahasa pemrograman yang mendukung paradigma object oriented programming ataupun scripting. Python hadir di ranah sistem operasi, virtualisasi, jaringan komputer, grafika komputer, kecerdasan buatan, teknologi web, game, dan lain sebagainya.

Bahasa pemrograman ini dibuat oleh Guido van Rossum dari Amsterdam, Belanda. Pada awalnya, motivasi pembuatan bahasa pemrograman ini adalah untuk bahasa skrip tingkat tinggi pada sistem operasi terdistribusi Amoeba. Bahasa pemrograman ini menjadi umum digunakan untuk kalangan engineer seluruh dunia dalam pembuatan perangkat lunaknya, bahkan beberapa perusahaan menggunakan Python sebagai pembuat perangkat lunak komersial.

Aplikasi bahasa Python

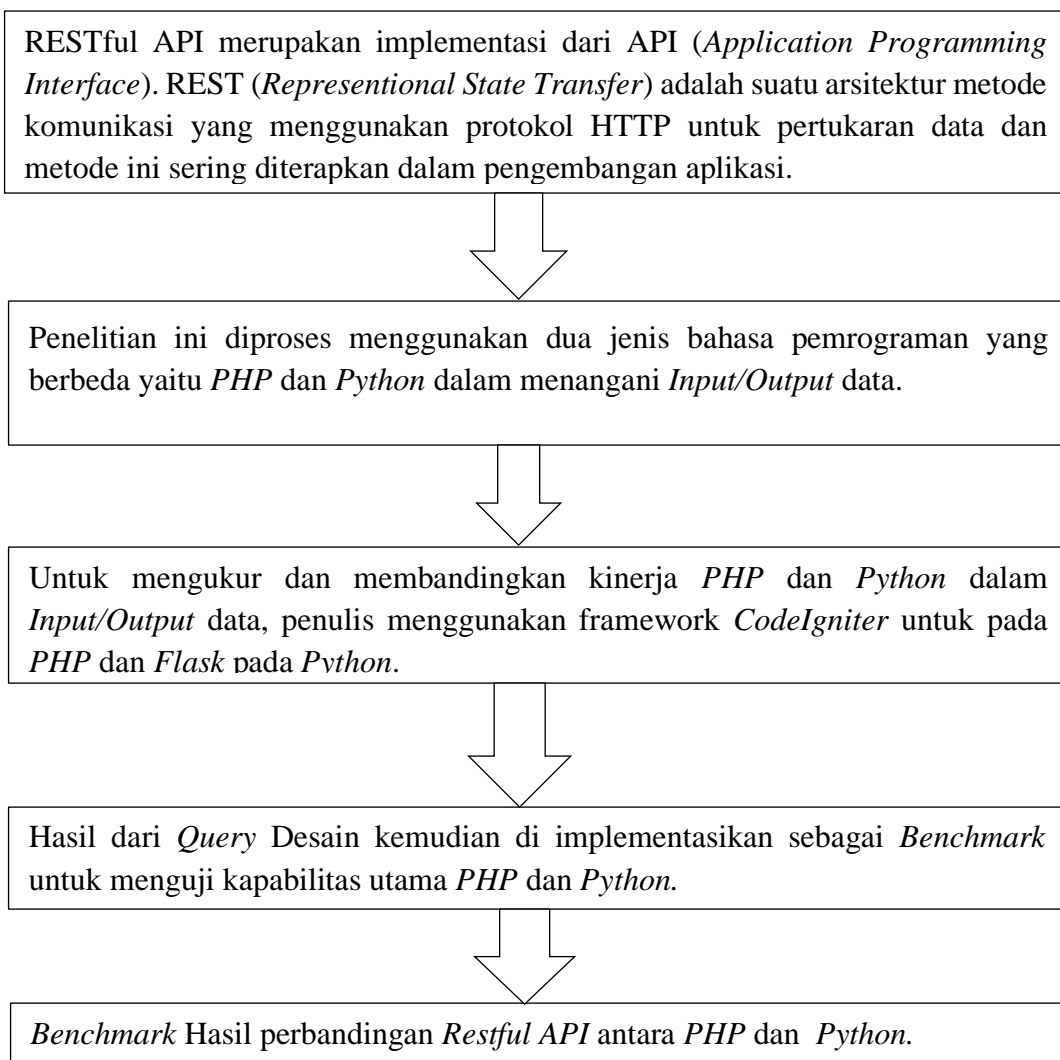
1. Perangkat bantu shell. Tugas - tugas sistem administrator, program baris perintah.
2. Kerja bahasa ekstensi. Antarmuka untuk pustaka C/C++, kustomisasi.
3. Pembuatan prototipe secara cepat/pembuatan sistem aplikasi. Prototipe yang dapat dibuang atau sesuai dengan permintaan.
4. Modul berdasarkan bahasa pemrograman. Pengganti dari penulisan parser khusus.
5. Antarmuka pengguna grafis. Penggunaan GUI API sederhana dan canggih.
6. Pengaksesan basisdata. Penyimpanan objek tetap, antarmuka sistem SQL.
7. Pemrograman terdistribusi. Penggunaan API mekanisme client/server terintegrasi.
8. Skrip internet. Skrip CGI, antarmuka HTTP, Aplet WWW, dan lainnya.

2.1.7 Flask

Flask adalah micro web framework yang ditulis dalam bahasa pemrograman Python dan berdasarkan *Werkzeug toolkit* dan *template engine Jinja2*. Berlisensi *BDS*. Flask disebut micro framework karena tidak membutuhkan alat-alat tertentu atau pustaka (Flask.pocoo, 2013). Flask tidak memiliki database abstraction layer,

validasi form, atau komponen lain di mana sudah ada pustaka pihak ketiga yang menyediakan fungsi umum. Namun, Flask mendukung ekstensi yang dapat menambahkan fitur aplikasi seolah-olah mereka diimplementasikan dalam Flask itu sendiri. Ekstensi yang ada untuk object-relational mapper, validasi form, penanganan unggahan, berbagai teknologi otentikasi terbuka, dan beberapa alat-alat yang terkait kerangka umum. Ekstensi diperbarui jauh lebih teratur daripada inti program Flask (Flask.pocoo, 2010).

2.2 Kerangka Konseptual



Gambar 2.3 Kerangka konseptual