

SKRIPSI

**MANAJEMEN DATA TUGAS AKHIR MAHASISWA FAKULTAS MIPA
MENGUNAKAN *RESTFUL API* SISTEM INFORMASI UNIVERSITAS
HASANUDDIN**

Disusun dan diajukan oleh

HENDRIANTO

H13115 315



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

**MANAJEMEN DATA TUGAS AKHIR MAHASISWA FAKULTAS MIPA
MENGUNAKAN *RESTFUL API* SISTEM INFORMASI UNIVERSITAS
HASANUDDIN**

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

**HENDRIANTO
H13115315**

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

MAKASSAR

JUNI 2021

HALAMAN PERNYATAAN KEOTENTIKAN

Yang bertanda tangan di bawah ini :

Nama : Hendrianto
NIM : H131 15 315
Progran Studi : Sistem Informasi
Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul :

**MANAJEMEN DATA TUGAS AKHIR MAHASISWA FAKULTAS
MIPA MENGGUNAKAN *RESTFUL API* SISTEM INFORMASI
UNIVERSITAS HASANUDDIN**

Adalah benar hasil karya saya sendiri bukan merupakan pengambilan alihan tulisan orang lain dan belum pernah dipublikasikan dalam bentuk apapun.

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

Makassar, 02 Juni 2021



Hendrianto

NIM. H13115315

**MANAJEMEN DATA TUGAS AKHIR MAHASISWA FAKULTAS MIPA
MENGUNAKAN *RESTFUL API* SISTEM INFORMASI UNIVERSITAS
HASANUDDIN**

Disusun dan diajukan oleh

HENDRIANTO
H13115315

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka
Penyelesaian studi Program Sarjana 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 Pendamping

Supri Bin 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 : Hendrianto
NIM : H131 15 315
Program Studi : Sistem Informasi
Judul Skripsi : Manajemen Data Tugas Akhir Mahasiswa Fakultas
MIPA Menggunakan Restful API Sistem Informasi
universitas Hasanuddin

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

1. Ketua : Dr. Hendra, S.Si, M.Kom.
2. Sekretaris : Supri Bin Hj Amir, S.Si., M.Eng
3. Anggota : Edy Saputra R, S.Si., M.Si.
4. Anggota : Rozalina Amran, S.T.,M.Eng.

Tanda Tangan

(.....)
(.....)
(.....)
(.....)

Ditetapkan di : Makassar
Tanggal : 02 Juni 2021



KATA PENGANTAR

AlhamdulillahRobbila'lamin, segala puji senantiasa dipanjatkan untuk menyampaikan rasa syukur kepada **Allah SWT** karena atas segala limpahan berkat dan rahmat yang diberikan kepada penulis sehingga dapat menyelesaikan penulisan skripsi ini. . Salam dan sholawat InsyaAllah senantiasa tercurah kepada **Nabi Muhammad Shallallahu'alaihi Wasallam** yang merupakan teladan dalam menjalankan kehidupan dunia.

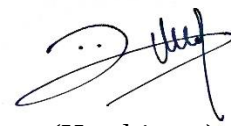
Skripsi dengan judul “**Manajemen Data Tugas Akhir Mahasiswa Fakultas MIPA Menggunakan Restful API Sistem Informasi universitas Hasanuddin**” dilakukan dalam rangka memenuhi salah satu syarat untuk mendapatkan gelar Sarjana Strata 1 pada program studi Sistem Informasi Departemen Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin Makassar. Penulis ucapkan terima kasih yang sebesar-besarnya kepada kedua orang tua penulis, Ayahanda **Subair S.H** dan Ibunda **Dra. Masnah** atas didikan dan cinta kasih serta doa dan nasehat yang selalu setia diberikan kepada penulis. Penulis juga mengucapkan terima kasih yang sedalam-dalamnya kepada seluruh keluarga besar penulis atas doa dan dukungan yang selalu diberikan selama ini. Disamping itu, penulis juga ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada :

1. Rektor Universitas Hasanuddin, Ibu **Prof. Dr. Dwia Aries Tina Pulubuhu** beserta jajarannya.
2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam, **Dr. Eng. Amiruddin** beserta jajarannya.
3. 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.
4. Bapak **Dr. Hendra, S.Si, M.Kom.** sebagai pembimbing utama yang telah banyak memberikan arahan, ide, motivavsi serta dukungan kepada penulis dalam banyak hal.

5. Bapak **Supri Bin Hj Amir, S.Si., M.Eng.** sebagai pembimbing pertama yang senantiasa memberikan masukan kepada penulis.
6. Bapak **Edy Saputra R, S.Si., M.Si.** dan Ibu **Rozalina Amran, S.T, M.Sc** sebagai tim penguji atas saran dan masukan pada penelitian yang telah dilakukan oleh penulis.
7. 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 administrasi.
8. Seluruh teman-teman seperjuangan **Program Studi Sistem Informasi 2015** yang telah mendukung dan berjuang bersama dalam suka dan duka selama ini. Serta kakak-kakak dan adik-adik **Sistem Informasi 2014, 2016, 2017, 2018** yang telah banyak membantu, semoga tetap semangat dalam mengejar impian.
9. Rekan-rekan **KKN Reguler Unhas Gel. 99 Kecamatan Camba** yang telah menjadi keluarga baru selama KKN dan menjadikan KKN sebagai momen yang berkesan.

Akhir kata, penulis berterima kasih atas segala bantuan, doa, dan dukungan yang diberikan kepada penulis dan berharap agar Allah SWT membalas segala kebaikan pihak yang telah membantu. Semoga tulisan ini dapat memberikan manfaat bagi para mahasiswa, khususnya bagi Mahasiswa Fakultas Matematika dan Ilmu Pengetahuan Alam jurusan Matematika dan bagi Perguruan Tinggi.

Makassar, 02 Juni 2021



(Hendrianto)

ABSTRAK

Pengajuan tugas akhir di Universitas Hasanuddin khususnya fakultas MIPA dilaksanakan melalui beberapa tahap sesuai dengan prosedur yang telah ditetapkan, yang meliputi pengajuan tugas akhir ke pembimbing, persetujuan judul tugas akhir, pemberian status kelayakan, penentuan dosen penguji, hingga penerbitan surat penugasan pembimbing dan Penguji. Seluruh proses ini masih dilakukan secara manual sehingga dapat menyita cukup banyak waktu dan juga tenaga. Skripsi ini membahas mengenai pembangunan sistem informasi manajemen data tugas akhir yang bermanfaat untuk mempermudah pengelolaan kegiatan pengajuan tugas akhir. Pembuatan *website* ini menggunakan bahasa pemrograman *PHP* serta *MySQL* sebagai *database* untuk menyimpan data tugas akhir yang telah diajukan. *Restful API* digunakan untuk mengambil data dosen dari database simpeg Unhas serta data berupa nomor surat dari database persuratan unhas. Pengujian dilakukan dengan menggunakan metode pengujian *Blackbox*. Dari hasil pembuatan dan pengujian, maka diperoleh hasil berupa sebuah *website* yang mampu melakukan pengelolaan kegiatan tugas akhir mahasiswa sesuai dengan prosedur yang ada.

Kata Kunci : Tugas Akhir, *website*, *PHP*, *MySQL*, *Restful API*, *Blackbox*.

ABSTRACT

Submission of final assignments at Hasanuddin University, especially the Faculty of Mathematics and Natural Sciences, is carried out in several stages according to predetermined procedures, which include submission of the final project to the supervisor, approval of the title of the final project, granting eligibility status, determining the examiner lecturer, and the issuance of the assignment letter for the supervisor and examiner. This whole process is still done manually, so it could take up a lot of time and effort. This thesis discusses the development of an information system of final project data management that is useful for facilitating the management of final assignment submission. Making this website by using the programming language *PHP* and *MySQL* as a database to store data for the final project that has been submitted. *Restful API* is used to collect lecturer data from the Unhas simpeg database and data in the form of letter numbers from the Unhas correspondence database. Testing is done using the Blackbox testing method. From the test results, obtained a website that is able to manage student final project activities in accordance to the procedures.

Keyword : Final Assignment, *website*, *PHP*, *MySQL*, *Restful API*, *Blackbox*.

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PERNYATAAN KEOTENTIKAN.....	ii
HALAMAN PENGESAHAN.....	iv
KATA PENGANTAR	v
ABSTRAK	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	2
1.3 Tujuan.....	2
1.4 Manfaat.....	2
1.5 Batasan Masalah.....	2
1.6 Organisasi Skripsi.....	3
BAB II TINJAUAN PUSTAKA.....	4
2.1 Landasan Teori.....	4
2.1.1 Web Aplikasi.....	4
2.1.2 PHP	5
2.1.3 Codeigniter.....	7
2.1.4 <i>Restful</i> API	9
2.1.5 MySQL.....	11
2.2 Kerangka Pikir.....	13
BAB III METODE PENELITIAN.....	15
3.1 Tahapan Penelitian	15
3.2 Perencanaan Sistem	16
3.3 Waktu dan Lokasi Penelitian.....	16

3.4	Sumber Data	17
3.5	Instrumen Penelitian.....	17
BAB IV HASIL DAN PEMBAHASAN		18
4.1	Desain Sistem dengan Diagram UML.....	18
4.1.1	<i>Use Case Diagram</i>	18
4.1.2	<i>Activity Diagram</i>	19
4.2	Implementasi <i>Restful API</i>	19
4.3	Menampilkan Notifikasi ke Email.....	22
4.4	Penjelasan Sistem	23
4.4.1	Halaman <i>Login</i>	23
4.4.2	<i>Dashboard Mahasiswa</i>	24
4.4.3	<i>Dashboard Pembimbing</i>	27
4.4.4	<i>Dashboard Kepala Laboratorium</i>	28
4.4.5	<i>Dashboard Ketua Prodi</i>	29
4.4.6	<i>Dashboard Departemen</i>	30
4.4.7	<i>Dashboard Fakultas</i>	32
4.5	Pengujian Sistem	34
BAB V KESIMPULAN DAN SARAN.....		46
5.1	Kesimpulan.....	46
5.2	Saran	46
DAFTAR PUSTAKA		47
LAMPIRAN		49

DAFTAR GAMBAR

Gambar 2.1 Hubungan Server dan Client	5
Gambar 2.2 Alur Kerja Framework Codeigniter	9
Gambar 2.3 Alur Kerja Restful API.....	11
Gambar 3.1 Flowchart penelitian.....	15
Gambar 3.2 Perancangan Sistem.....	16
Gambar 4.1 Use Case Diagram.....	18
Gambar 4.2 Activity Diagram.....	19
Gambar 4.3 Login ke server API	20
Gambar 4.4 ambil data dosen dari database simpeg Universitas Hasanuddin.....	21
Gambar 4.5 pengambilan nomor surat	21
Gambar 4.6 Konfigurasi Library PHPmailer	22
Gambar 4.7 Konfigurasi pada Controller.....	23
Gambar 4.8 Halaman Login.....	24
Gambar 4.9 Dashboard Mahasiswa.....	25
Gambar 4.10 Halaman tugas akhir mahasiswa	26
Gambar 4.11 Form pengajuan tugas akhir	26
Gambar 4.12 Menu Pembimbing.....	27
Gambar 4.13 Detail tugas akhir	28
Gambar 4.14 Menu Kepala Laboratorium	28
Gambar 4.15 detal tugas akhir untuk kepala laboratorium	29
Gambar 4.16 Halaman ketua prodi	29
Gambar 4.17 Ketua prodi mengusulkan penguji.....	30
Gambar 4.18 Menu departemen.....	31
Gambar 4.19 Departemen menerbitkan surat usulan	31
Gambar 4.20 Cetak surat yang telah diterbitkan	32

Gambar 4.21 Menu Fakultas	33
Gambar 4.22 departemen menerbitkan surat penugasan.....	33
Gambar 4.23 Cetak surat penugasan.....	34

DAFTAR TABEL

Tabel 4.1 Hasil pengujian halaman login.....	35
Tabel 4.2 Hasil Pengujian dashboard mahasiswa	36
Tabel 4.3 Hasil pengujian dashboard pembimbing.....	38
Tabel 4.4 Hasil pengujian Dashboard kepala laboratorium	39
Tabel 4.5 Hasil pengujian dashboard ketua prodi.....	41
Tabel 4.6 Hasil pengujian dashboard departemen	42
Tabel 4.7 Hasil pengujian dashboard fakultas	44

BAB I

PENDAHULUAN

1.1 Latar Belakang

Saat ini, perkembangan teknologi informasi semakin berkembang pesat. Hal ini menyebabkan semakin cepatnya informasi diproses dan diterima sehingga berakibat pada munculnya berbagai macam teknologi serta aplikasi yang dapat mempermudah berbagai pekerjaan dan permasalahan.

Tugas Akhir (skripsi) merupakan mata kuliah yang wajib harus dilalui oleh seluruh mahasiswa untuk mendapatkan gelar sarjananya. mata kuliah ini berupa proyek mandiri yang dilakukan oleh mahasiswa dan dibimbing oleh dosen pembimbing. Tugas akhir sebagai sarana untuk mengetahui seberapa jauh kemampuan mahasiswa dalam menguasai ilmu yang diberikan selama kuliah dan layak untuk mengabdikan ke masyarakat sesuai dengan kompetensi yang telah diajarkan oleh kampus. Sebelum menyusun tugas akhir, ada beberapa tahapan – tahapan yang harus dilalui terlebih dahulu. Antara lain : mengusulkan topik atau judul tugas akhir ke calon dosen pembimbing, penyetujuan judul oleh dosen pembimbing, penyusunan proposal, seminar proposal, seminar hasil, dan sidang tugas akhir.

Dalam penerapannya, proses pengurusan tugas akhir di Universitas Hasanuddin khususnya fakultas MIPA, masih menggunakan cara manual, misalkan untuk mengusulkan judul tugas akhir, mahasiswa harus bertemu langsung dengan dosen pembimbing, belum lagi jika harus mendapatkan persetujuan dari kepala laboratorium, maka harus menunggu kepala laboratorium untuk hadir di kampus, sehingga hal ini cukup menyita waktu. oleh karena itu, dibutuhkan sebuah aplikasi penunjang yang dapat membantu dalam proses pengajuan dan pengurusan tugas akhir yang dapat diakses disetiap waktu tanpa batasan waktu dan tempat.

Berdasarkan uraian diatas, maka solusi dari permasalahan ini adalah dengan membangun suatu sistem manajemen tugas akhir berbasis web sebagai fasilitas untuk membantu dalam pengurusan tugas akhir yang dapat diakses oleh koordinator

laboratorium, pihak fakultas, dosen pembimbing serta mahasiswa yang mengambil tugas akhir, sehingga atas permasalahan diatas, dibuatlah skripsi yang berjudul **“Manajemen Tugas Akhir Fakultas Mipa Universitas Hasanuddin Berbasis Web Menggunakan *Restful API*”**. Sistem manajemen tugas akhir ini diharapkan mampu menjadi fasilitas yang mendukung dalam pengerjaan Tugas Akhir di fakultas MIPA Universitas Hasanuddin.

1.2 Rumusan Masalah

Berdasarkan uraian diatas maka yang menjadi rumusan masalah dalam penelitian ini yaitu bagaimana mengembangkan aplikasi Manajemen Tugas Akhir berbasis *web* di Fakultas MIPA Universitas Hasanuddin.

1.3 Tujuan

Tujuan dari penelitian ini adalah agar dapat membangun sebuah aplikasi manajemen Tugas Akhir berbasis web yang dapat digunakan oleh mahasiswa, dosen serta staf pegawai di lingkungan Fakultas MIPA Universitas Hasanuddin dalam pengurusan tugas akhir.

1.4 Manfaat

Hasil dari penelitian ini diharapkan dapat memberikan manfaat berupa :

1. Sebagai salah satu sarana atau fasilitas yang mendukung dalam manajemen Tugas Akhir mahasiswa.
2. Memberikan kemudahan kepada mahasiswa dalam mengajukan tugas akhir kepada pembimbing.

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. Hanya dilakukan pada ruang lingkup Fakultas MIPA Universitas Hasanuddin.
2. Aplikasi dibangun menggunakan *Framework Codeigniter*.
3. *Database* yang digunakan adalah *MySQL*.

1.6 Organisasi Skripsi

Sistematika penulisan skripsi ini adalah sebagai berikut:

BAB I : Pendahuluan

Bab ini membahas mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan dan manfaat penulisan, serta organisasi skripsi.

BAB II : Tinjauan Pustaka

Bab ini membahas mengenai landasan teori, konsep dasar yang mendasari pokok permasalahan dalam tulisan ini.

BAB III : Metodologi Penelitian

Bab ini berisi waktu dan tempat penelitian, tahapan penelitian, rancangan sistem, sumber data, dan instrumen penelitian.

BAB IV : Hasil dan Pembahasan

Bab ini menguraikan tentang perancangan solusi serta implementasi dari masalah-masalah yang telah dianalisis. Pada bagian ini juga akan ditentukan bagaimana sistem dirancang, dibangun, diuji, dan disesuaikan dengan hasil penelitian.

BAB V : Penutup

Bab ini berisi kesimpulan dan saran yang merupakan jawaban yang melatar belakangi masalah pada Bab 1, dan saran untuk perbaikan menindak lanjuti hasil penelitian yang nantinya akan berguna bagi pengembangan sistem ini kedepannya.

BAB II

TINJAUAN PUSTAKA

2.1 Landasan Teori

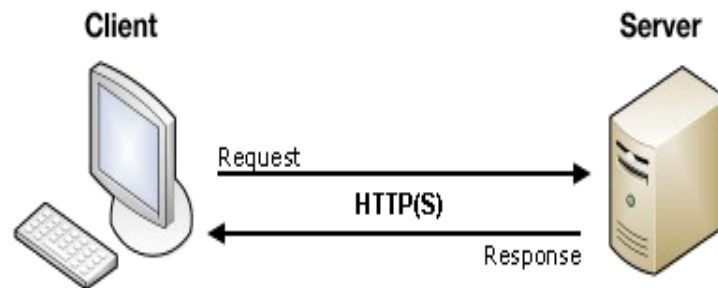
2.1.1 Web Aplikasi

Aplikasi berbasis web atau disebut juga *web base*, merupakan jenis aplikasi berbasis teknologi *web browser*. Artinya, aplikasi ini berjalan jika ada *browser* dan koneksi internet. Aplikasi berbasis *web* punya beberapa ciri sebagai berikut (Hilman, 2016).

- Aplikasi berbasis *web* tidak membutuhkan penginstalan karena untuk mengaksesnya hanya memerlukan peramban atau *browser* dan jaringan internet.
- Selama ada *browser* dan jaringan internet, aplikasi berbasis web dapat diakses dengan mudah di sistem operasi apapun. Sistem operasi *Windows*, *MacOS*, *iOS*, *Android*, *Linux*, atau sistem operasi lainnya selama ada *browser* dan bisa tersambung ke internet, maka aplikasi *web* dapat diakses dengan mudah.
- Karena bisa diakses dengan berbagai sistem operasi, maka aplikasi berbasis *web* bisa dengan mudah diakses melalui berbagai perangkat seperti perangkat PC desktop, laptop, *smartphone*, ataupun tablet.
- Aplikasi berbasis *web* tidak membutuhkan spesifikasi perangkat terlalu tinggi. Hal ini karena proses dilakukan pada web server penyedia aplikasi *web* tersebut.
- Karena memanfaatkan teknologi jaringan internet, aplikasi *web* membutuhkan jaringan yang stabil. Tujuannya agar proses menjalankan aplikasi tidak terganggu dan dapat berjalan dengan baik.
- Aplikasi web membutuhkan sistem keamanan yang baik. Terutama soal *server* penyedia aplikasi *web*. Jangan sampai *server down* sehingga mengakibatkan aplikasi berbasis *web* tidak bisa berjalan dengan baik.

Fungsi utama dari aplikasi *web* yaitu digunakan untuk mengolah data secara cepat melalui jaringan internet. Dimana nantinya pengguna bisa mengakses data atau

informasi yang dibutuhkan melalui *smartphone*, laptop, dan pc dengan syarat perangkatnya harus sudah ter-*install browser* dan terkoneksi ke jaringan internet atau lokal. Jika di perhatikan pada gambar 2.1, terdapat dua bagian penting daripada aplikasi *web*, yaitu *server* dan *client*. (Guntoro, 2019).



Gambar 2.1 Hubungan *Server* dan *Client*

- *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 PHP

PHP adalah bahasa pemrograman *script server-side* yang didesain untuk pengembangan *web*. Selain itu, PHP juga bisa digunakan sebagai bahasa pemrograman umum. PHP di kembangkan pada tahun 1995 oleh Rasmus Lerdorf, dan sekarang dikelola oleh *The PHP Group*. Situs resmi PHP beralamat di <http://www.php.net>. PHP disebut bahasa pemrograman *server side* karena PHP diproses pada komputer *server*. Hal ini berbeda dibandingkan dengan bahasa pemrograman *client-side* seperti *JavaScript* yang diproses pada *web browser client*. (andre, 2019).

PHP termasuk dalam produk *open source*. Jadi pengguna dapat merubah *source code* dan mendistribusikannya secara bebas. PHP juga diedarkan secara gratis. PHP juga dapat berjalan di berbagai *web server* misalnya IIS, *Apache*, PWS, dll. PHP dikembangkan dan ditulis ulang oleh pembuat aslinya, Rasmus di dalam bahasa C untuk meningkatkan kecepatannya. Oleh sebab itu pemrograman PHP mirip dengan pemrograman bahasa C. Adapun kelebihan - kelebihan PHP adalah sebagai berikut (Anggaeni dan Sujatmiko, 2013) :

- PHP mudah dibuat dan kecepatan akses tinggi
- PHP dapat berjalan dalam *web server* yang berbeda pula. PHP dapat berjalan disistem operasi Unix, Windows 98 dan NT, Machintosh.
- PHP diedarkan secara gratis
- PHP juga dapat berjalan pada *web server Microsoft Personal Web Server*, *Apache*, IIS, Xitami dan sebagainya.
- PHP termasuk bahasa yang *embedded* (bisa diletakkan atau ditempel di HTML).
- PHP termasuk *server-side programming*

Pada awalnya PHP merupakan singkatan dari *Personal Home Page*. Sesuai dengan namanya, PHP digunakan untuk membuat *website* pribadi. Dalam beberapa tahun perkembangannya, PHP menjelma menjadi bahasa pemrograman *web* yang *powerful* dan tidak hanya digunakan untuk membuat halaman *web* sederhana, tetapi juga *website* populer yang digunakan oleh jutaan orang seperti wikipedia, wordpress, joomla, dll. Saat ini PHP adalah singkatan dari PHP: *Hypertext Preprocessor*, sebuah kepanjangan rekursif, yakni permainan kata dimana kepanjangannya terdiri dari singkatan itu sendiri: PHP: *Hypertext Preprocessor*. PHP dapat digunakan dengan gratis (*free*) dan bersifat *Open Source*. PHP dirilis dalam lisensi *PHP License*, sedikit berbeda dengan lisensi GNU General Public License (GPL) yang biasa digunakan untuk proyek *Open Source* (andre, 2019).

PHP berjalan pada sisi *server* sehingga PHP disebut juga sebagai bahasa *server side scripting*. Artinya bahwa untuk menjalankan PHP, wajib adanya *web server*. PHP

ini bersifat *open source* sehingga dapat dipakai secara Cuma-cuma dan mampu lintas *platform*, yaitu dapat berjalan pada sistem operasi *Windows* maupun *Linux*. PHP juga dibangun sebagai modul pada *apache web server* dan sebagai *binary* yang dapat berjalan sebagai CGI (Agus dan Feni, 2012).

2.1.3 Codeigniter

Codeigniter adalah sebuah *framework* php yang bersifat *open source* dan menggunakan metode MVC (*Model, View, Controller*). *codeigniter* bersifat *free* alias tidak berbayar jika digunakan. *framework codeigniter* di buat dengan tujuan sama seperti *framework* lainnya yaitu untuk memudahkan *developer* atau *programmer* dalam membangun sebuah aplikasi berbasis *web* tanpa harus membuat nya dari awal. MVC adalah teknik atau konsep yang memisahkan komponen utama menjadi tiga komponen yaitu *model, view* dan *controller* (Hadi, 2019).

- *Model* merupakan bagian penanganan yang berhubungan dengan pengolahan atau manipulasi database. seperti misalnya mengambil data dari *database*, meng-*input* dan pengolahan *database* lainnya. semua intruksi yang berhubungan dengan pengolahan *database* di letakkan di dalam *model*.
- *View* merupakan bagian yang menangani halaman *user interface* atau halaman yang muncul pada *user*. tampilan dari *user interface* di kumpulkan pada *view* untuk memisahkannya dengan *controller* dan *model* sehingga memudahkan *web designer* dalam melakukan pengembangan tampilan halaman *website*.
- *Controller* merupakan kumpulan intruksi aksi yang menghubungkan *model* dan *view*, jadi user tidak akan berhubungan dengan *model* secara langsung, intinya dari *view* kemudian *controller* yang mengolah intruksi.

Ada beberapa kelebihan *CodeIgniter* (CI) dibandingkan dengan *Framework* PHP lain, yaitu sebagai berikut (Muhardian, 2018).

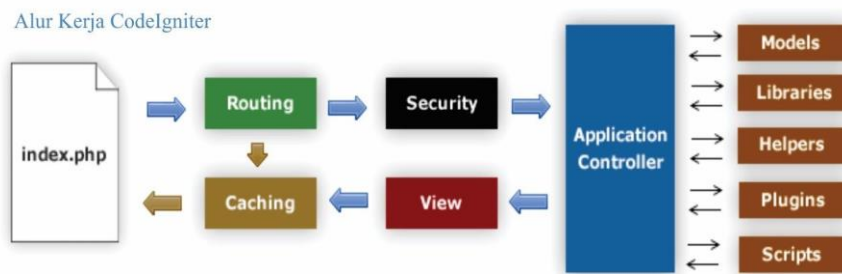
- Performa cepat: *Codeigniter* merupakan *framework* yang paling cepat dibanding *framework* yang lain. Karena tidak menggunakan *template engine* dan ORM yang dapat memperlambat proses.

- Konfigurasi yang minim (*nearly zero configuration*): tentu saja untuk menyesuaikan dengan *database* dan keleluasaan *routing* tetap diizinkan melakukan konfigurasi dengan mengubah beberapa *file* konfigurasi seperti *database.php* atau *autoload.php*, namun untuk menggunakan *codeigniter* dengan *setting* standar, hanya diperlukan mengubah sedikit saja pada *file* di folder *config*.
- Memiliki banyak komunitas: Komunitas CI di indonesia cukup ramai, tutorialnya pun mudah dicari.
- Dokumentasi yang lengkap: *Codeigniter* disertai dengan *user_guide* yang berisi dokumentasi yang lengkap.
- Mudah dipelajari pemula: Bagi pemula, CI sangat mudah dipelajari. Karena CI tidak terlalu bergantung pada *tool* tambahan seperti *composer*, ORM, *Template Engine*, dll.

Berikut ini adalah fitur-fitur penting yang ada di *CodeIgniter* (Amalia, 2018).

- Sistem berbasis *Model-View-6* (MVC).
- Sangat Ringan dan Butuh Sedikit Sumber Daya (*Memory* + CPU).
- Men-support 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*.

Untuk lebih memahami bagaimana alur kerja dari *codeigniter*, dapat dilihat pada gambar 2.2.



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).

2.1.4 Restful API

RESTful API didasarkan pada teknologi *state transfer* (*representational state transfer* / *REST*), gaya arsitektur dan pendekatan komunikasi yang sering digunakan

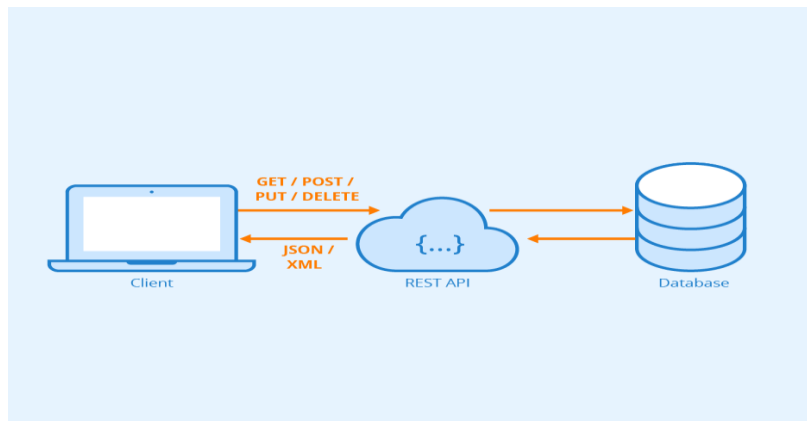
dalam pengembangan layanan *web*. Meskipun REST dapat digunakan di hampir semua protokol, tapi biasanya memanfaatkan HTTP ketika digunakan untuk *Web API*. Hal ini membantu pengembang *web* tidak perlu menginstal *library* atau perangkat lunak tambahan untuk memanfaatkan desain REST API. Design REST API pertama kali diperkenalkan oleh Dr. Roy Fielding dalam disertasi doktor tahun 2000-nya. REST API terkenal karena fleksibilitasnya yang luar biasa. Data tidak terikat dengan metode dan sumber daya, REST memiliki kemampuan untuk menangani beberapa jenis panggilan, mengembalikan format data yang berbeda dan bahkan mengubah secara struktural tentunya dengan implementasi yang benar. REST yang digunakan oleh *browser* dapat dianggap sebagai bahasa internet. Dengan meningkatnya penggunaan *cloud*, API muncul untuk mengekspos layanan *web*. REST adalah pilihan logis untuk membangun API yang memungkinkan pengguna untuk terhubung dan berinteraksi dengan layanan *cloud*. API telah banyak digunakan oleh situs-situs seperti *Amazon*, *Google*, *LinkedIn* dan *Twitter*. (Yudana, 2019).

Dalam pengaplikasiannya, REST lebih banyak digunakan untuk *web service* yang berorientasi pada *resource*. Maksudnya orientasi pada *resource* adalah orientasi yang menyediakan *resource-resource* sebagai layanannya dan bukan kumpulan dari aktifitas yang mengolah *resource* itu. Beberapa contoh *web service* yang menggunakan REST adalah : *Flickr API*, *YouTube API*, dan *Amazon API* (Hariyanto dkk, 2013).

Gambar 2.3 memperlihatkan bagaimana alur kerja dari REST API. API secara eksplisit memanfaatkan metodologi HTTP yang ditentukan oleh protokol RFC 2616. Permintaan ke API bisa menggunakan *GET* untuk mengambil sumber daya, *PUT* untuk mengubah status atau memperbarui sumber daya, yang dapat berupa objek, file, atau blok, *POST* untuk membuat sumber daya itu, dan *DELETE* untuk menghapusnya. API dapat dikatakan “RESTful” jika memiliki fitur berikut (Yudana, 2019):

- *Client – server* : *client* menangani *front end* dan *server* menangani *back end* dan keduanya dapat diganti secara independen satu sama lain.

- *Stateless* : Tidak ada data klien yang disimpan di *server* ketika ada permintaan dan status sesi disimpan di klien.
- *Cacheable* : Klien dapat *men-cache* respon (seperti *browser* yang *men-cache* elemen statis halaman *web*) untuk meningkatkan kinerja.



Gambar 2.3 Alur Kerja *Restful* API

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 dijual 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 di pahami karena dapat merelasikan antara beberapa tabel dengan database maupun antar database. Ada tiga bentuk SQL yang perlu diketahui, yaitu *Data Definition Language* (DDL), *Data Manipulation Language*(DML), dan *Data Control Language* (DCL) (Yasin, 2019).

- *Data Definition Language (DDL)*

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

CREATE : digunakan untuk membuat, termasuk di dalamnya membuat database baru, tabel baru view baru, dan kolom baru.

ALTER : berfungsi untuk mengubah struktur tabel yang telah dibuat. Mencakup di dalamnya mengubah nama tabel, menambah kolom, mengubah kolom, menghapus kolom, dan memberikan atribut pada kolom.

DROP : berfungsi untuk menghapus database atau tabel.

- *Data Manipulation Language (DML)*

DML dapat dipakai setelah menjalankan perintah DDL. DML berfungsi untuk memanipulasi, mengubah, atau mengganti isi dari database (tabel) 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 sudah mempunyai user dan ingin mengatur hak akses masing-masing *user*, pengguna 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*. Terdapat beberapa perintah DCL yang perlu diketahui, yaitu:

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 (Riyadi, 2019).

2.2 Kerangka Pikir

Kerangka pikir merupakan model konseptual tentang bagaimana teori berhubungan dengan berbagai faktor yang telah diidentifikasi. Sehingga pada penelitian ini memerlukan kerangka konseptual agar mempermudah dalam proses penelitian. Maka dapat digambarkan kerangka konseptual sebagai berikut:

membangun aplikasi manajemen tugas akhir mahasiswa menggunakan *Restful API* Fakultas MIPA Universitas Hasanuddin.



Aplikasi ini dibangun dengan menggunakan *Codeigniter* untuk membangun aplikasi *front-end* (HTML:CSS) dan *back-end* (PHP), serta *MySQL* untuk tempat penyimpanan data tugas akhir.



Untuk mendapatkan data akademik, data dosen, serta data surat, digunakan *Restful API* untuk komunikasi data.



Hasil akhir dari penelitian ini berupa sebuah aplikasi berbasis *web* yang dapat diakses dengan menggunakan koneksi internet.