

**PENGEMBANGAN SERVICE MASTER PADA APLIKASI
SIKADCLOUD (STUDI KASUS PT SEVIMA)**

(Skripsi)

Oleh

**AULIA AHMAD NABIL
NPM 1817051074**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2022**

**PENGEMBANGAN SERVICE MASTER PADA APLIKASI
SIKADCLOUD (STUDI KASUS PT SEVIMA)**

Oleh

AULIA AHMAD NABIL

Skripsi

**Sebagai Salah Satu Syarat untuk Mendapat Gelar
SARJANA KOMPUTER**

Pada

**Jurusan Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2022**

ABSTRAK

PENGEMBANGAN SERVICE MASTER PADA APLIKASI SIKADCLOUD (STUDI KASUS PT SEVIMA)

Oleh

AULIA AHMAD NABIL

SiakadCloud adalah produk utama SEVIMA sebagai solusi manajemen akademik terintegrasi terlengkap, aman dan terbukti efektif dalam memfasilitasi manajemen pendidikan tinggi dan pelaporan PDDIKTI. SiakadCloud saat ini sedang dimigrasikan ke arsitektur *microservice*. Arsitektur *microservice* membuat aplikasi SiakadCloud harus dibagi menjadi bagian-bagian kecil. Penelitian ini bertujuan untuk mengembangkan sebuah layanan yang dapat digunakan untuk mengelola data master pada aplikasi SiakadCloud. Layanan ini disebut layanan master. Pengembangan dalam penelitian ini menggunakan metode *waterfall*. Pengembangan layanan dilakukan dengan menggunakan *framework* Lumen. Penelitian ini menghasilkan sebuah layanan yang dapat mengelola data master untuk digunakan dan menjadi standar untuk pengembangan layanan lainnya.

Kata Kunci: API, Lumen, Microservice, PHPUnit, Swagger.

ABSTRACT

DEVELOPMENT OF SERVICE MASTER IN SIAKADCLOUD APPLICATION (CASE STUDY PT SEVIMA)

Oleh

AULIA AHMAD NABIL

SiakadCloud is SEVIMA's main product as an integrated academic management solution that is the most complete, secure and proven effective in facilitating higher education management and PDDIKTI reporting. SiakadCloud is currently being migrated to a microservice architecture. The microservice architecture makes the SiakadCloud application must be divided into small parts. This research is intended to develop a service that can be used to manage master data in the SiakadCloud application. The service is called the service master. The development in this research uses the waterfall method. Service development is made using the Lumen framework. This research produces a service that can manage master data for use and become a standard for the development of other services.

Keywords: API, Lumen, Microservice, PHPUnit, Swagger.

Judul Skripsi : **PENGEMBANGAN SERVICE MASTER PADA APLIKASI SIAKAD CLOUD (STUDI KASUS PT SEVIMA)**

Nama Mahasiswa : **Aulia Ahmad Nabil**

NPM : **1817051074**

Program Studi : **S1 Ilmu Komputer**

Jurusan : **Ilmu Komputer**

Fakultas : **Matematika dan Ilmu Pengetahuan Alam**



MENYETUJUI

1. Komisi Pembimbing

Didik Kurniawan, S.Si., M.T.
NIP. 198004192005011004

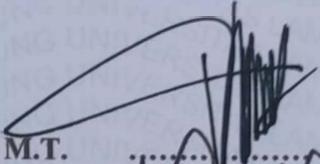
Astria Hijriani, S.Kom., M.Kom.
NIP. 198103082008122002

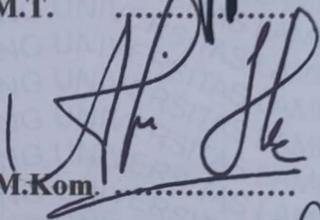
2. Ketua Jurusan Ilmu Komputer

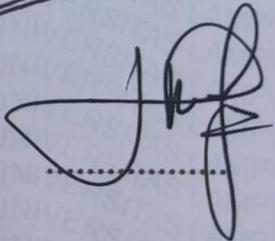
Didik Kurniawan, S.Si., M.T.
NIP. 198004192005011004

MENGESAHKAN

1. Tim Penguji

Ketua : Didik Kurniawan, S.Si., M.T. 

Penguji I
Sekretaris : Astria Hijriani, S.Kom., M.Kom. 

Penguji II
Bukan Pembimbing : Dwi Sakethi, S.Si., M.Kom. 

2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



Dr. Eng. Surpto Dwi Yuwono, S.Si., M.T.
NIP: 197407052000031001

Tanggal Lulus Ujian Skripsi: 19 Mei 2022

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Aulia Ahmad Nabil

NPM : 1817051074

Dengan ini menyatakan bahwa skripsi saya yang berjudul **“Pengembangan Service Master Pada Aplikasi SiakadCloud (Studi Kasus PT SEVIMA)”** merupakan karya saya sendiri dan bukan karya orang lain. Semua tulisan yang tertuang dalam skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti skripsi saya merupakan hasil penjiplakan atau dibuat orang lain, maka bersedia menerima sanksi berupa pencabutan gelar yang telah saya terima.

Bandar Lampung, 20 Juni 2022



Nabil

NPM. 1817051074

RIWAYAT HIDUP



Penulis dilahirkan di Bandar Lampung, pada tanggal 10 Mei 2000, sebagai anak pertama dari empat bersaudara.

Penulis menyelesaikan pendidikan formal di MIN 5 Bandar Lampung dan selesai pada tahun 2012.

Kemudian pendidikan menengah pertama di MTs N 2 Bandar Lampung yang diselesaikan pada tahun

2015, lalu melanjutkan ke pendidikan menengah atas di SMA Kebangsaan yang diselesaikan pada tahun 2018.

Pada tahun 2018 penulis terdaftar sebagai mahasiswa Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung melalui jalur SBMPTN. Selama menjadi mahasiswa, penulis melakukan beberapa kegiatan antara lain.

1. Menjadi anggota Adapter Himpunan Mahasiswa Jurusan Ilmu Komputer pada periode 2018/2019.
2. Menjadi anggota bidang Keilmuan Himpunan Mahasiswa Jurusan Ilmu Komputer periode 2018/2019.

3. Mengikuti perlombaan bidang *programming*, *capture the flag*, dan *MTQ* di berbagai acara perlombaan tingkat mahasiswa maupun umum.
4. Menjadi Kepala Bidang Keilmuan Himpunan Mahasiswa Jurusan Ilmu Komputer periode 2019/2020
5. Menjadi Asisten Dosen Jurusan Ilmu Komputer tahun 2019 hingga 2021.
6. Mengikuti ujian sertifikasi dan mendapat sertifikat *Programmer* dan *Junior Web Developer* oleh Lembaga Sertifikasi Profesi Informatika pada tahun 2020 dan 2021.
7. Melaksanakan Kerja Praktek pada bulan Februari periode 2020/2021 di PT Arsi Enarcon.
8. Melaksanakan KKN Kebangsaan di Desa Pandan Makmur, Kecamatan Geragai, Kabupaten Tanjung Jabung Timur, Jambi pada tahun 2020/2021 dengan program kerja pembuatan web desa.
9. Penerima Beasiswa Bright Scholarship pada tahun 2019.
10. Mengikuti Program Kampus Merdeka di PT SEVIMA pada bulan September 2021 – Februari 2022.

MOTTO

11. “Jika kamu tidak sanggup menahan lelahnya belajar maka kamu harus sanggup menahan perihnya kebodohan.”

(Imam Syafi'i)

12. “Jangan takut mencoba hal yang baru, gapailah impianmu. Tapi ingatlah, tak peduli kemana kamu pergi, keluarga tempatmu kembali.”

(Anonim)

13. “Jika kamu ingin hidup yang sulit maka kamu harus membuat pilihan yang mudah.”

(Anonim)

14. “Work in silence and let success make the noise.”

(Anonim)

PERSEMBAHAN

Alhamdulillahillobbilamin

Puji dan syukur tercurahkan kepada Allah Subhanahu Wa Ta'ala atas segala Rahmat dan Karunia-Nya sehingga saya dapat menyelesaikan skripsi ini. Shalawat serta salam selalu tercurahkan kepada Nabi Muhammad SAW.

Kupersembahkan karya ini kepada:

Kedua Orang Tuaku Tercinta

Yang senantiasa memberikan yang terbaik, dan melantunkan do'a yang selalu menyertaiku. Kuucapkan pula terima kasih sebesar-besarnya karena telah mendidik dan membesarkanku dengan cara yang dipenuhi kasih sayang, dukungan, dan pengorbanan yang belum bisa terbalaskan.

Seluruh Keluarga Besar Ilmu Komputer 2018

Yang selalu memberikan semangat dan dukungan.

Almamater Tercinta, Universitas Lampung dan Jurusan Ilmu Komputer

Tempat bernaung mengemban semua ilmu untuk menjadi bekal hidup.

SANWACANA

Puji syukur kehadirat Allah SWT atas berkah, rahmat dan hidayat-Nya, serta petunjuk dan pedoman dari Rasulullah Nabi Muhammad Sholallahu Alaihi Wasallam penulis dapat menyelesaikan skripsi yang berjudul “Pengembangan Service Master Pada Aplikasi SiakadCloud (Studi Kasus PT SEVIMA)” dengan baik dan lancar.

Terima kasih penulis ucapkan kepada semua pihak yang telah membantu dan berperan besar dalam menyusun skripsi ini, antara lain.

1. Kedua orang tua tercinta yang selalu memberi dukungan, do’a, semangat, motivasi, dan kasih sayang yang luar biasa tak terhingga. Semua yang telah kalian berikan tidak akan pernah mampu untukku balas. Semoga Allah SWT selalu memberikan kebahagiaan dan keberkahan dalam kehidupan kalian di dunia dan akhirat.
2. Bapak Didik Kurniawan, S.Si., M.T. sebagai pembimbing utama yang telah memberikan arahan, ide, kritik serta saran kepada penulis sehingga dapat menyelesaikan skripsi ini dengan baik.

3. Ibu Astria Hijriani, S.Kom., M.Kom. sebagai pembimbing kedua yang juga selalu dapat memberikan waktu untuk membimbing penulis dalam memberikan ide, kritik serta saran untuk dapat menyelesaikan skripsi ini.
4. Bapak Dwi Sakethi, S.Si., M.Kom. sebagai pembahas yang telah memberikan masukan yang bermanfaat dalam perbaikan skripsi ini.
5. Bapak Ardiansyah, S.Kom., M.Kom. selaku pembimbing akademik penulis yang selalu mendukung peningkatan akademik penulis.
6. Bapak Didik Kurniawan, S.Si., M.T. selaku ketua Jurusan Ilmu Komputer FMIPA Universitas Lampung.
7. Bapak Dr. Eng. Suropto Dwi Yuwono, S.Si., M.T. selaku Dekan FMIPA Universitas Lampung.
8. Bapak Dr. rer. nat. Akmal Junaidi, M.Sc. selaku Sekretaris Jurusan Ilmu Komputer FMIPA Universitas Lampung.
9. Ibu Ade Nora Maela dan Bang Zainuddin yang telah membantu segala urusan administrasi penulis di Jurusan Ilmu Komputer.
10. Bapak dan Ibu Dosen Jurusan Ilmu Komputer FMIPA Universitas Lampung yang telah memberikan ilmu dan pengalaman dalam hidup untuk menjadi lebih baik.
11. Teman-teman di SEVIMA yang banyak memberikan ilmu-ilmu baru selama magang.
12. Teman-teman grup Mas Pikri dan Mbak Intan, yaitu Fikri Mulya Permana, Ratih Indah Wardani, Rahmadila Nurjannah, Suci Hikmawati, Intania Rahmadhilla, dan Dewi Lestari yang selalu memberikan semangat dan dukungan.

13. Keluarga Ilmu Komputer 2018 yang tidak bisa penulis sebut satu persatu. Keluarga kedua penulis, rekan kelompok, rekan diskusi, rekan bercanda, dan telah memberi arti dan warna serta pengalaman tak ternilai semasa duduk di bangku kuliah.
14. Seluruh kakak tingkat dan adik tingkat Ilmu Komputer yang tidak bisa disebutkan satu persatu yang telah menjadi warna selama masa perkuliahan penulis.
15. Teman-teman Himakom dan Brighter yang sudah mengajarkan banyak hal dalam berorganisasi, memberikan banyak pengalaman.

Penulis menyadari bahwa skripsi ini masih jauh dari kata sempurna, semoga skripsi ini membawa manfaat dan keberkahan bagi semua civitas Ilmu Komputer Universitas Lampung aamiin ya rabbal aalamiin.

Bandar Lampung, 20 Juni 2022

Aulia Ahmad Nabil
NPM. 1817051074

DAFTAR ISI

| | Halaman |
|---|----------------|
| DAFTAR ISI | iv |
| DAFTAR TABEL | vi |
| DAFTAR GAMBAR | viii |
| I. PENDAHULUAN | 1 |
| 1.1. Latar Belakang | 1 |
| 1.2. Rumusan Masalah..... | 3 |
| 1.3. Batasan Masalah | 3 |
| 1.4. Tujuan Penelitian | 3 |
| 1.5. Manfaat Penelitian | 3 |
| II. TINJAUAN PUSTAKA | 5 |
| 2.1. Penelitian Terdahulu | 5 |
| 2.1.1. Perancangan <i>Application Programming Interface</i> (API) Berbasis Web menggunakan Gaya Arsitektur <i>Representational State Transfer</i> (REST) untuk Pengembangan Sistem Informasi Administrasi Pasien Klinik Perawatan Kulit. | 5 |
| 2.1.2. Pengembangan REST API pada Aplikasi SAKU (Solusi Akuntansi UMKM) Menggunakan <i>Framework</i> Lumen. | 6 |
| 2.1.3. Implementasi REST Web <i>Service</i> untuk <i>Sales Order</i> dan <i>Sales Tracking</i> Berbasis <i>Mobile</i> | 7 |
| 2.2. Uraian Landasan Teori..... | 8 |
| 2.2.1. API..... | 8 |
| 2.2.2. Aplikasi..... | 9 |
| 2.2.3. Basis Data | 9 |
| 2.2.4. SIAKAD | 10 |
| 2.2.5. REST API | 10 |
| 2.2.6. JSON | 11 |
| 2.2.7. <i>Framework</i> Lumen | 11 |
| 2.2.8. PHP | 12 |
| 2.2.9. Redis | 12 |
| 2.2.10.JWT (JSON Web Token)..... | 12 |
| 2.2.11.Unit <i>Testing</i> | 13 |
| 2.2.12.Swagger..... | 13 |
| 2.2.13.Arsitektur <i>Microservice</i> | 14 |
| 2.2.14.Arsitektur Monolitik..... | 14 |

| | |
|---|------------|
| 2.2.15.UML (Unified Modeling Language)..... | 14 |
| 2.2.16.Metode Waterfall..... | 15 |
| 2.2.17.Black Box Testing..... | 16 |
| III. METODE PENELITIAN | 17 |
| 3.1. Waktu dan Tempat Penelitian..... | 17 |
| 3.2. Perangkat Penelitian | 18 |
| 3.2.1. Perangkat Keras | 18 |
| 3.2.2. Perangkat Lunak | 18 |
| 3.3. Tahapan Penelitian..... | 19 |
| 3.3.1. Studi Literatur | 20 |
| 3.3.2. Pengumpulan Data..... | 20 |
| 3.3.3. Pengembangan Sistem | 21 |
| 3.3.4. Skenario Pengujian Sistem | 64 |
| IV. HASIL DAN PEMBAHASAN | 65 |
| 4.1. Hasil Implementasi | 65 |
| 4.1.1. Struktur API..... | 66 |
| 4.1.2. Dokumentasi API..... | 73 |
| 4.2. Pengujian Sistem..... | 79 |
| 4.2.1. <i>Unit Test</i> | 80 |
| 4.2.2. <i>Performance Test</i> | 125 |
| V. SIMPULAN DAN SARAN..... | 141 |
| 5.1. Simpulan | 141 |
| 5.2. Saran | 141 |
| DAFTAR PUSTAKA..... | 144 |

DAFTAR TABEL

| Tabel | Halaman |
|--|----------------|
| Tabel 3.1 Waktu Penelitian | 17 |
| Tabel 3.2 Rencana Pengembangan Sistem..... | 25 |
| Tabel 3.3 Kamus data entitas ak_bidang_studi..... | 32 |
| Tabel 3.4 Kamus data entitas ak_keluarga_mhs | 33 |
| Tabel 3.5 Kamus data entitas ak_mahasiswa..... | 33 |
| Tabel 3.6 Kamus data entitas ak_mahasiswa_keluar | 35 |
| Tabel 3.7 Kamus data entitas ak_mahasiswa_pt..... | 36 |
| Tabel 3.8 Kamus data entitas ak_pegawai | 37 |
| Tabel 3.9 Kamus data entitas ak_pegawai_pt..... | 38 |
| Tabel 3.10 Kamus data entitas lv_agama..... | 40 |
| Tabel 3.11 Kamus data entitas lv_almamater | 40 |
| Tabel 3.12 Kamus data entitas lv_bidang_ilmu..... | 41 |
| Tabel 3.13 Kamus data entitas lv_bidang_ilmu_pt..... | 41 |
| Tabel 3.14 Kamus data entitas lv_gelombang | 42 |
| Tabel 3.15 Kamus data entitas lv_gelombang_pt | 42 |
| Tabel 3.16 Kamus data entitas lv_hobi..... | 43 |
| Tabel 3.17 Kamus data entitas lv_jalur_pendaftaran..... | 43 |
| Tabel 3.18 Kamus data entitas lv_jalur_pendaftaran_pt..... | 44 |
| Tabel 3.19 Kamus data entitas lv_jenis_tinggal | 44 |
| Tabel 3.20 Kamus data entitas lv_jenjang_pendidikan..... | 45 |
| Tabel 3.21 Kamus data entitas lv_jenjang_pendidikan_pt | 45 |
| Tabel 3.22 Kamus data entitas lv_kelas_perkuliahahan | 46 |
| Tabel 3.23 Kamus data entitas lv_kota | 47 |
| Tabel 3.24 Kamus data entitas lv_minat..... | 47 |

| | |
|--|----|
| Tabel 3.25 Kamus data entitas lv_pekerjaan..... | 48 |
| Tabel 3.26 Kamus data entitas lv_pekerjaan_opt | 48 |
| Tabel 3.27 Kamus data entitas lv_penghasilan | 49 |
| Tabel 3.28 Kamus data entitas lv_penghasilan_opt..... | 49 |
| Tabel 3.29 Kamus data entitas lv_sistem_kuliah..... | 50 |
| Tabel 3.30 Kamus data entitas lv_sistem_kuliah_pt..... | 51 |
| Tabel 3.31 Kamus data entitas lv_status_mahasiswa | 51 |
| Tabel 3.32 Kamus data entitas lv_status_mahasiswa_opt | 52 |
| Tabel 3.33 Kamus data entitas lv_status_pegawai | 52 |
| Tabel 3.34 Kamus data entitas lv_suku..... | 53 |
| Tabel 3.35 Kamus data entitas lv_transport..... | 53 |
| Tabel 3.36 Kamus data entitas ms_jabatan_fungsional | 54 |
| Tabel 3.37 Kamus data entitas ms_jabatan_struktural..... | 54 |
| Tabel 3.38 Kamus data entitas ms_jenis_pegawai..... | 55 |
| Tabel 3.39 Kamus data entitas ms_negara..... | 56 |
| Tabel 3.40 Kamus data entitas ms_pangkat..... | 56 |
| Tabel 3.41 Kamus data entitas ms_periode..... | 57 |
| Tabel 3.42 Kamus data entitas ms_pt | 57 |
| Tabel 3.43 Kamus data entitas ms_tahun_ajaran..... | 58 |
| Tabel 3.44 Kamus data entitas ms_tahun_kurikulum | 58 |
| Tabel 3.45 Kamus data entitas ms_unit | 59 |
| Tabel 3.46 Kamus data entitas ms_universitas | 60 |
| Tabel 3.47 Kamus data entitas ms_universitas_opt | 61 |
| Tabel 3.48 Kamus data entitas ms_universitas_prodi..... | 62 |
| Tabel 3.49 Kamus data entitas ms_yayasan..... | 62 |
| Tabel 3.50 Skenario Pengujian <i>Equivalence Partitioning</i> pada <i>Service Master</i> ... | 64 |
| Tabel 4.1 Daftar API..... | 67 |
| Tabel 4.2 Hasil Pengujian Unit Test | 92 |

DAFTAR GAMBAR

| Gambar | Halaman |
|---|----------------|
| Gambar 2.1 Metodologi <i>Waterfall</i> (Pressman & Maxim, 2006). | 15 |
| Gambar 3.1 Diagram Alir Penelitian. | 19 |
| Gambar 3.2 Pembagian <i>Service</i> Pada SiakadCloud 2.0..... | 23 |
| Gambar 3.3 <i>Use Case</i> Diagram <i>Service</i> Master..... | 27 |
| Gambar 3.4 <i>Activity</i> Diagram Menampilkan Data Mahasiswa..... | 28 |
| Gambar 3.5 <i>Activity</i> Diagram Menambahkan Data Mahasiswa. | 29 |
| Gambar 3.6 <i>Entity Relationship</i> Diagram <i>Service</i> Master. | 31 |
| Gambar 4.1. Struktur Framework. | 65 |
| Gambar 4.2. Desain URL Pada <i>Service</i> Master. | 66 |
| Gambar 4.3. Dokumentasi API Pada <i>Service</i> Master. | 73 |
| Gambar 4.4. Dokumentasi API Untuk Menampilkan Daftar Mahasiswa..... | 74 |
| Gambar 4.5. Dokumentasi API Untuk Menambahkan Data Mahasiswa..... | 75 |
| Gambar 4.6. Dokumentasi API Menampilkan Detail Mahasiswa. | 76 |
| Gambar 4.7. Dokumentasi API Mengubah Data Mahasiswa. | 78 |
| Gambar 4.8. Dokumentasi API Menghapus Data Mahasiswa. | 79 |
| Gambar 4.9. Pengujian Mendapatkan Daftar Mahasiswa..... | 80 |
| Gambar 4.10. Pengujian Mendapatkan Detail Mahasiswa. | 81 |
| Gambar 4.11. Pengujian Mendapatkan Detail Mahasiswa Dengan ID Yang Salah | 82 |
| Gambar 4.12. Pengujian Mendapatkan Detail Mahasiswa Dengan <i>Path</i> Yang Salah. | 82 |
| Gambar 4.13. Pengujian Mengakses <i>Resource</i> Yang Salah. | 83 |
| Gambar 4.14. Pengujian Menambahkan Data Mahasiswa..... | 83 |
| Gambar 4.15. Pengujian Mengubah Data Mahasiswa. | 84 |

| | |
|--|-----|
| Gambar 4.16. Pengujian Menghapus Data Mahasiswa..... | 84 |
| Gambar 4.17. Pengujian Menambahkan Data Mahasiswa Tanpa <i>Body</i> | 85 |
| Gambar 4.18. Pengujian Mengakses <i>Resource</i> Dengan <i>Invalid</i> Token. | 85 |
| Gambar 4.19. Hasil Pengujian <i>Stress Testing</i> Menggunakan <i>Tools</i> k6. | 140 |

I. PENDAHULUAN

1.1. Latar Belakang

PT Sentra Vidya Utama (SEVIMA) adalah perusahaan konsultan dan pengembang teknologi informasi yang didirikan pada tahun 2004. SEVIMA memiliki pengalaman lebih dari 15 tahun dalam mengimplementasikan sistem informasi di perguruan tinggi dan telah membantu lebih dari 400 perguruan tinggi.

SiakadCloud adalah produk unggulan SEVIMA sebagai solusi manajemen akademik terintegrasi yang terlengkap, aman dan terbukti efektif dalam memudahkan manajemen perguruan tinggi dan pelaporan PDDIKTI. Saat ini terdapat 174 universitas yang terdaftar sebagai pengguna SEVIMA SiakadCloud.

Sistem Informasi Akademik (SIKAD) SiakadCloud ini telah terintegrasi dengan modul/sistem lain seperti Sistem Informasi Kepegawaian Perguruan Tinggi untuk pelaporan SISTER, Sistem Informasi Penerimaan Mahasiswa Baru (SIM PMB), sistem keuangan mahasiswa, sistem *Computer Based Test* (CBT), Pelaporan *Feeder* Pangkalan Data Pendidikan Tinggi (PDDIKTI), Pelaporan *Education Management Information System* (EMIS) dan sistem informasi lainnya yang digunakan dalam ruang lingkup perguruan tinggi.

SiakadCloud dibangun menggunakan arsitektur monolitik. Arsitektur monolitik merupakan sebuah arsitektur di mana dalam pembuatan aplikasi semua komponen terbentuk sebagai satu kesatuan kode yang tidak dapat dipisahkan satu dan lainnya. Dampak dari arsitektur monolitik ini yaitu; penurunan performa ketika aplikasi menjadi semakin besar dan banyak yang

mengakses, untuk menggunakan teknologi baru kode aplikasi harus ditulis ulang secara keseluruhan, jika terjadi error pada salah satu bagian maka dapat mempengaruhi keseluruhan aplikasi. Dampak-dampak tersebut akan sangat mempengaruhi baik di sisi *end user* maupun *programmer* di kemudian hari. Contoh kasus yang sudah terjadi pada SEVIMA SiakadCloud ini yaitu ketika waktu pengisian KRS, *traffic* menjadi sangat tinggi, sehingga membuat seluruh aplikasi menjadi *down*. Hal ini mempengaruhi mahasiswa lainnya yang hanya sekedar ingin melihat nilai ataupun jadwal kuliah menjadi tidak bisa.

SiakadCloud saat ini akan dilakukan migrasi ke arsitektur *microservice*. Arsitektur *microservice* membuat aplikasi SiakadCloud harus dibagi menjadi bagian-bagian kecil atau *service-service* yang mana setiap *service* memiliki skema *database* sendiri. Setiap *service* harus berhubungan satu sama lain. Komunikasi yang digunakan antar *service* memerlukan protokol yang ringan agar komunikasi antar *service* bisa berjalan dengan baik.

Berdasarkan uraian di atas, aplikasi SiakadCloud 2.0 akan dibagi menjadi beberapa *service*. SiakadCloud 2.0 belum mempunyai sebuah *service* yang dapat mengelola data-data master dalam aplikasi, maka pada penelitian ini akan dikembangkan sebuah *service* yang dapat digunakan untuk manajemen data-data master yang dapat digunakan oleh *service* lainnya dan juga menjadi standarisasi pengembangan *service* selanjutnya. *Service* tersebut dinamakan dengan *service* master. *Service* master ini nantinya paling sering digunakan oleh *service* lain dalam mendapatkan data master, sehingga *service* master dikembangkan terlebih dahulu dibandingkan dengan *service* lainnya. *Service* master ini akan dibuat dengan menggunakan *framework* Lumen karena SDM pada PT SEVIMA banyak yang berfokus pada bahasa pemrograman PHP.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah dikemukakan, maka permasalahan yang dapat dirumuskan pada penelitian ini yaitu bagaimana mengembangkan sebuah *service* yang dapat digunakan sebagai manajemen data master untuk *service-service* lainnya menggunakan *framework* Lumen.

1.3. Batasan Masalah

Agar penelitian lebih terfokus, berikut ini beberapa batasan masalah dalam penelitian ini:

1. *Service* yang dikembangkan adalah *service* master.
2. *Service* yang dikembangkan menggunakan *framework* Lumen dan menggunakan basis data Postgresql.
3. *Service* dikembangkan untuk pengembang internal aplikasi SiakadCloud.
4. Menggunakan metode *black-box testing* dengan *tools* PHPUnit untuk memastikan kesesuaian antara *request* dan *response*.
5. Menggunakan Swagger untuk membuat dokumentasi API.
6. Menggunakan JSON sebagai standar format dalam komunikasi data.

1.4. Tujuan Penelitian

Tujuan dilakukannya penelitian ini adalah untuk menghasilkan sebuah *service* sebagai manajemen data master untuk dapat digunakan oleh *service* lainnya dan menjadi standarisasi pengembangan *service* selanjutnya.

1.5. Manfaat Penelitian

Manfaat dari penelitian ini adalah:

1. Dapat menjadi referensi bagi pengembang *service* lain dalam pengembangan *service-service* selanjutnya.

2. Memudahkan bagi pengembang *service* lainnya dalam mendapatkan data master yang dapat digunakan pada *service* yang dikembangkan.
3. Memudahkan bagi *programmer frontend* dalam mendapatkan informasi *request* dan *response* yang dibutuhkan.
4. Dapat menjadi referensi bagi penelitian-penelitian berikutnya yang berhubungan dengan *microservice*.

II. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Beberapa penelitian terdahulu yang digunakan sebagai referensi dalam penelitian ini adalah sebagai berikut.

2.1.1. Perancangan *Application Programming Interface* (API) Berbasis Web menggunakan Gaya Arsitektur *Representational State Transfer* (REST) untuk Pengembangan Sistem Informasi Administrasi Pasien Klinik Perawatan Kulit.

Dalam kedokteran, penggunaan komputer untuk mendukung prosedur dan metode pemberian pelayanan kesehatan disebut manajemen komputer. Bantuan komputer dalam administrasi pelayanan kesehatan dengan bantuan konsultan teknologi informasi biasanya berbentuk sistem informasi. Saat mengembangkan ini, konsultan teknologi informasi harus memperhitungkan bahwa jumlah kasus yang dihadapi klien mereka akan terus bertambah. Oleh karena itu, diperlukan konsep pengembangan yang baik agar sistem informasi dapat beradaptasi. Memisahkan logika *database* dari antarmuka logis dari sistem informasi dianggap sebagai cara yang ideal untuk memecahkan masalah ini. Dengan cara ini anda dapat mengembangkan setiap logika satu per satu dan menggunakan lebih banyak fitur.

Penelitian ini menerapkan gaya arsitektur REST ketika mengembangkan API sebagai *backend* untuk sistem informasi

perawatan pasien klinik dermatologi. API yang dikembangkan menggunakan Javascript *Object Notation* (JSON) sebagai format standar untuk *transfer* data dan JSON Web Token (JWT) sebagai kode otentikasi pengguna sistem. Studi ini menunjukkan bahwa pengembangan API telah berhasil dilakukan untuk mengelola pasien di klinik perawatan kulit dan REST telah diterapkan untuk memfasilitasi pengembangan desain API. Studi ini membangun *backend* sistem informasi manajemen pasien klinik dermatologi berbasis REST API. Pengujian API dilakukan dalam tiga tahap: pengujian otentikasi menggunakan JWT pada beberapa *server* internal, pengujian API menggunakan metode *Equivalence Partitioning*, dan pengujian fungsi sistem.

2.1.2. Pengembangan REST API pada Aplikasi SAKU (Solusi Akuntansi UMKM) Menggunakan *Framework* Lumen.

Sebuah usaha memiliki bagian penting untuk dikerjakan yaitu melakukan pencatatan arus keuangan. Namun, dalam praktiknya pemilik UMKM tidak memahami aturan pencatatan pembukuan keuangan, sehingga UMKM sering terkendala dengan proses pencatatan arus keuangan. Sistem informasi akuntansi menjadi salah satu hal penting yang dapat mendukung kelancaran dalam kegiatan operasional perusahaan, baik itu usaha besar ataupun usaha kecil. Sistem informasi memiliki beberapa platform diantaranya *desktop*, *web*, dan *mobile*. Agar sebuah aplikasi dapat berjalan di *platform* berbeda dengan menggunakan *database* yang sama, dibutuhkan jembatan yang dapat menghubungkan perbedaan *platform* tersebut. Salah satu jembatan yang dapat menghubungkannya adalah API (*Application Programming Interface*).

Hasil penelitian ini yaitu membuat REST API pada aplikasi saku (solusi akuntansi UMKM). Berdasarkan hasil perancangan sistem

yang telah dibuat, sistem ini diimplementasikan menggunakan *framework* lumen dan *database* mysql, serta menggunakan metode REST.

2.1.3. Implementasi REST Web Service untuk Sales Order dan Sales Tracking Berbasis Mobile.

Divisi Penjualan adalah ujung tombak perusahaan yang mempunyai tugas utama untuk memasarkan produk. Dengan menggunakan sistem informasi penjualan berbasis *desktop/web*, perusahaan sudah dapat mencatat transaksi penjualan. Namun ada beberapa masalah yang dihadapi oleh perusahaan dan tidak dapat terselesaikan dengan sistem informasi penjualan berbasis *web/desktop*. Masalah yang pertama adalah tentang pengawasan pegawai yang bertugas untuk menawarkan barang ke pelanggan. Masalah yang dihadapi adalah perusahaan merasa kesulitan untuk memastikan apakah tenaga penjual tersebut benar mendatangi pelanggan sesuai dengan target perusahaan. Untuk itu dibutuhkan sebuah aplikasi berbasis *mobile* yang dapat mencatat posisi dari tenaga penjual ketika mengunjungi pelanggan. Masalah kedua yang dihadapi adalah perusahaan tidak dapat memantau secara *real-time* barang apa saja yang dipesan oleh pelanggan. Saat ini yang terjadi adalah tenaga penjual akan mencatat barang yang dibutuhkan oleh pelanggan, jika dibutuhkan informasi tentang ketersediaan barang yang dipesan maka tenaga penjual akan menanyakan informasi tersebut melalui telepon. Tenaga penjual juga tidak tahu pasti ketersediaan stok barang secara *real-time*.

Hasil dari penelitian ini adalah untuk menerapkan teknologi Web Services dengan arsitektur REST yang dapat digunakan oleh berbagai macam jenis *client* seperti aplikasi *mobile*, aplikasi web, dan aplikasi *desktop* yang dapat membantu perusahaan untuk melakukan pelacakan atau *tracking* terhadap tenaga penjual yang ditugaskan

untuk menawarkan barang atau penagihan ke pelanggan. Dengan REST Web Services yang akan dibuat diharapkan perusahaan akan dapat memastikan bahwa semua tenaga penjual akan mengunjungi pelanggan sesuai dengan target yang sudah ditentukan oleh perusahaan. Selain itu penelitian ini juga akan menerapkan cara yang lebih cepat untuk memasukan data pesanan barang ke dalam sistem dengan memanfaatkan kamera pada perangkat *mobile* untuk pembacaan *barcode*.

2.2. Uraian Landasan Teori

2.2.1. API

Application programming interface (API) merupakan suatu dokumentasi yang terdiri dari *interface*, fungsi, kelas, struktur dan sebagainya untuk membangun sebuah perangkat lunak (Fajar, 2015). Dengan adanya API ini, maka memudahkan *programmer* untuk “membongkar” suatu *software*, kemudian dapat dikembangkan atau diintegrasikan dengan perangkat lunak yang lain. API dapat dikatakan sebagai penghubung suatu aplikasi dengan aplikasi lainnya yang memungkinkan *programmer* menggunakan sistem *function* (Pranata et al., 2018). Proses ini dikelola melalui sistem operasi. Keunggulan dari API ini adalah memungkinkan suatu aplikasi dengan aplikasi lainnya dapat saling berhubungan dan berinteraksi.

2.2.2. Aplikasi

Menurut (Abdurahman & Riswaya, 2014), aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut. Tujuannya mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut. Aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputasi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan.

2.2.3. Basis Data

Menurut (Swara & Pebriadi, 2016), basis data atau *database* merupakan kumpulan data yang disusun secara sistematis di dalam sebuah perangkat keras (komputer) sehingga dapat diolah menggunakan perangkat lunak untuk menjadi sebuah informasi yang berguna.

Menurut (Munawaroh, 2005), *Database PostgreSQL* merupakan salah satu alternatif solusi bagi pengguna *database* yang mendukung banyak *platform* dan bebas lisensi. PostgreSQL termasuk sebagai *database server* yang handal dengan berbagai macam fitur-fitur pendukungnya, sehingga menjadikan *database* ini begitu ideal sebagai media penyimpanan dari aplikasi sistem informasi. PostgreSQL dikembangkan oleh University of California di Berkeley *Computer Science Department*. Dengan sifatnya yang *open source* menjadikan pula *database* ini dapat dikembangkan sesuai dengan kebutuhan.

2.2.4. SIAKAD

Sistem Informasi Akademik (SIAKAD) merupakan sistem yang mengolah data dan melakukan proses kegiatan akademik yang melibatkan antara mahasiswa, dosen, dan administrasi akademik (Harleni & Marisa, 2018). Sistem informasi Akademik melakukan kegiatan proses administrasi mahasiswa dalam melakukan kegiatan administrasi akademik, melakukan proses pada transaksi belajar-mengajar antara dosen dan mahasiswa, melakukan proses administrasi akademik baik yang menyangkut kelengkapan dokumen dan biaya yang muncul pada kegiatan registrasi administrasi akademik. Sistem Informasi Akademik (SIAKAD) selain merupakan sumber daya informasi di kampus, juga dapat digunakan sebagai sarana media komunikasi antara dosen dan mahasiswa, mahasiswa dengan mahasiswa dosen dengan pejabat kampus terkait dan siapa saja yang ada di lingkungan kampus tersebut.

2.2.5. REST API

REST API adalah sebuah implementasi dari API (*Application Programming Interface*). REST (*Representational State Transfer*) adalah suatu arsitektur metode komunikasi yang menggunakan protokol HTTP untuk pertukaran data (Perdana, 2018). HTTP method terdiri dari GET, POST, PUT, PATCH, DELETE (Yanti & Rihyanti, 2021).

a. Kelebihan

Menurut (Halili & Ramadani, 2018), berikut ini kelebihan dari REST API :

1. Mendukung komunikasi *stateless*.
2. *Bandwidth* yang ringan, dikarenakan menggunakan format pesan JSON.

3. Dapat digunakan oleh berbagai *client*.

b. Kekurangan

Menurut (Halili & Ramadani, 2018), berikut ini kekurangan dari REST API :

1. Tidak cocok untuk data dalam jumlah besar.
2. *Request* REST (terutama GET) tidak cocok untuk data dalam jumlah besar.
3. Latensi dalam waktu pemrosesan permintaan dan penggunaan *bandwidth*.

2.2.6. JSON

JSON (Javascript *Object Notation*) merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan. Selain JSON, terdapat XML yang memiliki fungsi sama yaitu alat komunikasi antar aplikasi, integrasi data, dan komunikasi aplikasi eksternal dengan luaran. JSON lebih unggul dari XML, mulai dari kecepatan, penulisan yang lebih mudah dan *coding* untuk *parsing* yang lebih ringkas dan sederhana (Qibtiyah & Rahayu, 2017).

2.2.7. Framework Lumen

Lumen merupakan salah satu *micro framework* yang dikembangkan oleh Taylor Otwell, pengembang yang berada di balik *framework* PHP Laravel. Lumen lebih dikhususkan untuk pembuatan API, yang mengakibatkan beberapa fitur yang ada di Laravel dipangkas dan tidak tersedia di Lumen. Namun demikian, Lumen memiliki performa yang lebih baik dalam menangani permintaan (*request*) dibandingkan Laravel (Surahman et al., 2018).

2.2.8. PHP

Menurut (Haviluddin et al., 2016), PHP adalah sebuah bahasa pemrograman yang perintahnya dilaksanakan *server* dan kemudian hasilnya ditampilkan pada komputer *client*. PHP merupakan HTML *embedded*, yaitu *sintaks* PHP yang dituliskan bersamaan dengan sintaks HTML. Sehingga PHP dan HTML merupakan sinergi dua bahasa pemrograman yang saling menguatkan.

2.2.9. Redis

Redis merupakan *database* NoSQL yang menyimpan data dalam bentuk pasangan *key-value* dan memiliki sistem *in-memory* dalam pengambilan datanya (Febriyani et al., 2019). Sehingga Redis memiliki kemampuan pembacaan data yang lebih cepat dibandingkan NoSQL lainnya. Walaupun Redis dikenal sebagai sebuah *database*, tetapi Redis juga dapat berperan sebagai *cache* dan *message broker*. Mekanisme kerja Redis sebagai *message broker* menerapkan model *publish-subscribe*, yaitu *topic-based*.

2.2.10. JWT (JSON Web Token)

JSON Web Token adalah sebuah token berbentuk string JSON yang sangat padat (ukurannya), informasi mandiri yang gunanya sendiri untuk melakukan sistem autentikasi dan pertukaran informasi. Karena bentuknya kecil, token JWT dapat dikirim melalui URL, parameter HTTP POST atau di dalam *header* HTTP, dan juga karena ukurannya yang kecil maka dapat ditransmisikan dengan lebih cepat. Disebut informasi mandiri karena isi dari token yang dihasilkan memiliki informasi dari pengguna yang dibutuhkan, sehingga tidak perlu *query* ke basis data lebih dari satu kali (Rahmatulloh et al., 2018).

JWT bekerja di mana token ini seperti *password*, jadi ketika *user* berhasil melakukan *login* maka *server* akan memberikan sebuah token. Nanti token tersebut akan disimpan oleh *user* pada *local storage* atau *cookies browser* dan bila *user* ingin mengakses halaman halaman tertentu maka harus menyertakan token tersebut. Untuk itu *user* akan mengirim balik token yang dikirim di awal sebagai bukti bila *user* sudah melakukan *login*.

Struktur JWT terdiri dari tiga bagian yang dipisahkan oleh titik (.), yaitu *header*, *payload*, dan *signature*. Oleh karena itu, JWT biasanya terlihat seperti berikut ini (xxxxx.yyyyy.zzzzz).

2.2.11. Unit Testing

Menurut (Rizkyana et al., 2021), Unit *testing* merupakan salah satu metode pengujian perangkat lunak yang digunakan dengan cara menguji unit terkecil dari sebuah kode. Kegunaan unit *testing* bagi pengembang perangkat lunak adalah dapat memperkecil kesalahan atau *bug* yang ada pada sistem. Pengujian unit *testing* dilakukan hingga sistem sudah memenuhi syarat sesuai rancangan. Untuk membuktikan bahwa pengujian unit *testing* dapat efektif digunakan dalam pengembangan *service*, maka dalam penelitian ini dilakukan pengujian *service* dengan unit *testing* yaitu menggunakan PHPUnit.

2.2.12. Swagger

Menurut (Syaryadhi, 2018), swagger adalah suatu *tools framework* untuk merancang, membuat dokumentasi dan mengakses Restful API berdasarkan spesifikasi dari Open API. Bahasa dan format *file* yang digunakan untuk membangun rancangan Restful API menggunakan swagger adalah YAML (*Yet Another Markup Language*). Swagger berperan dalam memberikan spesifikasinya kepada Open API

Initiative (OAI) yang menjadi standar dalam pengembangan Restful API.

2.2.13. Arsitektur Microservice

Arsitektur *microservice* merupakan alternatif arsitektur yang lebih terukur dan lebih fleksibel. Pada arsitektur *microservice*, sistem informasi dirancang untuk terdistribusi dan menyediakan layanan secara lebih fokus dan spesifik (Munawar & Hodijah, 2018). Permasalahan besar akan dipecah menjadi beberapa solusi kecil yang disusun dalam satu *service*, di mana setiap *service* memiliki tanggung jawabnya sendiri. Dengan pendekatan ini, suatu sistem informasi akan terdiri dari beberapa *service* yang dapat dikelola dan didistribusikan secara *independent*, hal ini akan lebih memudahkan sistem untuk beradaptasi terhadap perubahan kebutuhan.

2.2.14. Arsitektur Monolitik

Arsitektur monolitik merupakan sebuah arsitektur di mana dalam pembuatan aplikasi, semua komponen menjadi satu kesatuan. Pemilihan arsitektur monolitik dikarenakan latensi komunikasi antar modulnya sangat cepat, karena berada di dalam sebuah *server* yang sama serta kemudahan dalam mendistribusikan aplikasi ke *server* karena aplikasi terbungkus menjadi satu bagian (Ponce et al., 2019).

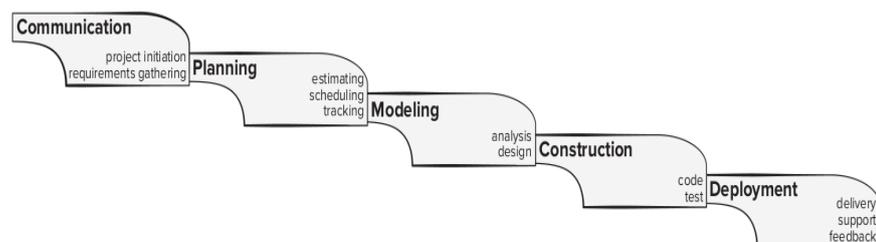
2.2.15. UML (Unified Modeling Language)

Unified Modeling Language (UML) adalah standar bahasa yang digunakan untuk menjelaskan dengan jelas *requirement*, pembuatan desain serta analisis dan arsitektur dari pemrograman yang berorientasi objek tersebut digambarkan dengan jelas dikarenakan

dalam dunia industri standar bahasa ini banyak digunakan. UML ada dikarenakan dalam pemodelan visual sangat dibutuhkan untuk menspesifikasikan, menggambarkan, membangun, dan mendokumentasikan dari sisi sistem perangkat lunak. UML sendiri juga memberikan standar penulisan sebuah sistem *blueprint*, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema *database*, dan komponen-komponen yang diperlukan dalam sistem *software* (Suendri, 2018).

2.2.16. Metode Waterfall

Metode *waterfall* adalah metode kerja yang menekankan fase-fase yang berurutan dan sistematis. Disebut *waterfall* karena proses mengalir satu arah “ke bawah” seperti air terjun. Metode *waterfall* ini harus dilakukan secara berurutan sesuai dengan tahap yang ada. Menurut (Pressman & Maxim, 2006), metode *waterfall* memiliki 5 fase pengembangan yaitu *Communication*, *Planning*, *Modeling*, *Construction*, dan *Deployment*. Secara *flow* dapat dilihat pada Gambar 2.1.



Gambar 2.1 Metodologi *Waterfall* (Pressman & Maxim, 2006).

Communication adalah tahap penginisiasian proyek atau bagaimana proyek yang akan dibuat. *Planning* sendiri adalah membuat rencana kerja dan proyek yang telah diinisiasikan. Kemudian masuk ke dalam tahap *Modeling* yang dilakukan untuk membuat desain analisis dari proyek agar mendapatkan gambaran kasar dari proyek. Setelah selesai melakukan *Modeling*, maka dilakukan *Construction* untuk

mengimplementasikan desain yang dibuat ke dalam kode program dan melakukan pengujian. Tahap akhir dari metode *waterfall* adalah *deployment* yang bertujuan untuk menyebarkan sistem yang sudah dibuat, sering juga disebut melakukan *hosting*. Saat proses *deployment* berjalan, pengembang sistem tetap harus memantau sistem untuk mengetahui apakah ada *error* pada sistem di kemudian hari, *error* sendiri bisa ditemukan oleh pengembang maupun para pengguna yang memberikan *feedback* kepada pengembang.

2.2.17. Black Box Testing

Black box testing merupakan pengujian perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program untuk mengetahui apakah fungsi, masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan (Cholifah et al., 2018). Metode *Black box testing* merupakan salah satu metode yang mudah digunakan karena hanya memerlukan batas bawah dan batas atas dari data yang diharapkan, Estimasi banyaknya data uji dapat dihitung melalui banyaknya *field* data *entry* yang akan diuji, aturan entri yang harus dipenuhi serta kasus batas atas dan batas bawah yang memenuhi. Metode ini dapat diketahui jika fungsionalitas masih dapat menerima masukan data yang tidak diharapkan maka menyebabkan data yang disimpan kurang valid.

III. METODE PENELITIAN

3.1. Waktu dan Tempat Penelitian

Penelitian dilakukan di PT SEVIMA yang beralamat di Medokan Asri Tengah MA-2 Blok Q No 16, Medokan Ayu, Kec. Rungkut, Kota SBY, Jawa Timur. Penelitian ini dilakukan pada semester genap Tahun Ajaran 2021/2022. Waktu penelitian dapat dilihat pada Tabel 3.1 berikut.

Tabel 3.1 Waktu Penelitian

| Tahapan | Kegiatan | 2021 | | | | | | | | | | | | | | | | 2022 | | | | | | | |
|---------------------|---|----------|---|---|---|---------|---|---|---|---------|---|---|---|----------|---|---|---|---------|---|---|---|---|---|---|---|
| | | Septembe | | | | Oktober | | | | Novembe | | | | Desember | | | | Januari | | | | | | | |
| | | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | | | | | | |
| | | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | | | | |
| Penelitian Awal | Studi Literatur Pengumpulan Data | ■ | ■ | ■ | ■ | | | | | | | | | | | | | | | | | | | | |
| | <i>Communication Planning Modelling</i> | | | | | ■ | ■ | ■ | ■ | | | | | | | | | | | | | | | | |
| Penelitian Lanjutan | <i>Construction Testing Deployment</i> | | | | | | | | | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | | | | | | | | |
| Penelitian Akhir | Penulisan Laporan | | | | | | | | | | | | | | | | | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |

Dapat dilihat pada tabel 3.1, waktu penelitian dibagi menjadi 3 tahapan yaitu penelitian awal, penelitian lanjutan dan penelitian akhir. Tahap pertama adalah penelitian awal dilakukan dari September 2021 s/d Oktober 2021 dengan kegiatan yang bertujuan untuk melakukan analisis, mendapatkan

kebutuhan sistem, dan melakukan pemodelan sistem. Tahap kedua adalah penelitian lanjutan yang dilaksanakan dari November 2021 s/d awal bulan Januari 2022 dengan kegiatan yang bertujuan untuk mengimplementasikan model sistem yang telah dibuat pada penelitian awal dan kemudian akan dilakukan *testing* dengan metode *black box testing* menggunakan PHPUnit. Tahap ketiga yang merupakan tahap akhir penelitian yaitu penulisan laporan yang bertujuan untuk memberikan dokumentasi terkait penelitian yang dilakukan.

3.2. Perangkat Penelitian

Spesifikasi perangkat yang digunakan pada penelitian ini adalah sebagai berikut.

3.2.1. Perangkat Keras

Perangkat keras yang digunakan dalam penelitian ini adalah sebuah laptop dengan spesifikasi sebagai berikut.

- a. *System Manufacturer* : HP
- b. *System Model* : HP Elitebook 2170p
- c. *Processor* : Intel® Core™ i5-3427U CPU @
1.80GHz (4 CPUs)
- d. *Installed RAM* : 8 GB
- e. *System Type* : 64 bit

3.2.2. Perangkat Lunak

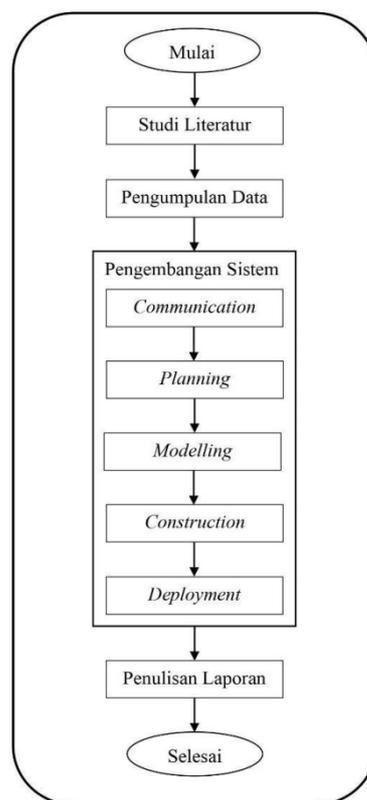
Perangkat lunak yang digunakan dalam penelitian ini adalah sebagai berikut.

- a. Sistem Operasi Ubuntu 20.04.3
- b. Visual Studio Code *version* 1.63.2
- c. Apache *version* 2.4.41

- d. PHP *version* 7.4.3
- e. PostgreSQL *version* 12.9
- f. Firefox *Browser version* 95.0.1
- g. Insomnia *version* 2021.6.0 sebagai REST API *client*
- h. *Framework* Lumen *version* 8.0
- i. Git *version control system version* 2.2.5.1

3.3. Tahapan Penelitian

Tahapan penelitian pada penelitian ini ditunjukkan pada Gambar 3.1. Gambar 3.1 merupakan diagram alir penelitian. Tahapan penelitian dibagi menjadi 3 bagian yaitu pengumpulan data, pengembangan sistem, dan penulisan laporan.



Gambar 3.1 Diagram Alir Penelitian.

3.3.1. Studi Literatur

Tahapan yang pertama adalah studi literatur. Studi literatur adalah tahapan yang dilakukan untuk memahami konsep *microservices*, REST API, memahami bagaimana struktur dari SiakadCloud versi sekarang dan juga mengidentifikasi permasalahan yang sering muncul.

3.3.2. Pengumpulan Data

Pengumpulan data pada penelitian ini didapatkan dari 2 sumber yaitu observasi dan studi pustaka.

a. Studi Pustaka

Studi pustaka merupakan langkah awal dalam metode pengumpulan data. Studi pustaka merupakan metode pengumpulan data yang diarahkan kepada pencarian data dan informasi melalui dokumen-dokumen, baik dokumen tertulis, foto-foto, gambar, maupun dokumen elektronik yang berkaitan dengan REST API maupun *microservice* yang dapat mendukung dalam proses pengembangan sistem.

b. Observasi

Observasi merupakan langkah kedua dalam pengumpulan data setelah penulis melakukan studi pustaka. Observasi dilakukan agar penulis dapat mengetahui dengan lebih pasti kondisi permasalahan yang sebenarnya pada SiakadCloud versi sekarang. Observasi dilakukan dengan cara mempelajari aplikasi yang telah

dibuat sebelumnya seperti arsitektur, *database*, struktur, fitur-fitur yang ada dan juga *gate* yang sering diakses oleh pengguna.

3.3.3. Pengembangan Sistem

Tahap selanjutnya yaitu melakukan pengembangan sistem, pada tahap ini menggunakan metode *waterfall*. Mulai dari *Communication*, *Planning*, *Modeling*, *Construction*, dan *Deployment*.

1. *Communication*

Communication adalah tahap inisialisasi proyek seperti menganalisis permasalahan yang dihadapi dan mengumpulkan data-data yang diperlukan, serta membantu mendefinisikan fitur dan fungsi dari aplikasi. Tahap ini dilakukan dengan cara berdiskusi dengan *programmer* sebelumnya dan juga mengumpulkan data-data tambahan baik yang ada di jurnal, artikel, maupun dari internet. Dari tahapan ini, didapatkan hasil berupa analisis sistem yaitu:

a. Analisis Masalah

Berdasarkan penjelasan yang diberikan oleh *programmer* sebelumnya, permasalahan yang ada pada aplikasi SiakadCloud versi sekarang adalah penggunaanya yang sudah mencapai angka 213.379.484 *user*/tahun ditambah aplikasinya sudah cukup besar dan mempunyai banyak fitur di dalamnya yang mana aplikasi tersebut dibangun dengan menggunakan arsitektur monolitik. Hal ini mengakibatkan *server down* pada beberapa waktu di mana banyak *user* yang menggunakan SiakadCloud secara bersamaan. Kasus yang sering terjadi adalah ketika perkuliahan di kampus memasuki masa pengisian KRS. Ketika semua mahasiswa mengakses

fitur KRS secara bersama-sama, maka akan terjadi *server down*, karena terlalu banyak *load* ataupun *request* yang dilakukan dalam satu waktu. Hal ini menyulitkan bagi mahasiswa lain yang hanya ingin sekedar melihat nilai ataupun data-data mereka yang terdapat di dalam aplikasi.

b. Analisis Kebutuhan Sistem

Analisis kebutuhan sistem dilakukan untuk menyelesaikan masalah yang telah dijelaskan sebelumnya. Untuk mengatasi masalah tersebut maka akan SiakadCloud versi sekarang atau SiakadCloud 2.0 perlu dilakukan *refactoring* arsitekturnya dari yang sebelumnya menggunakan arsitektur monolitik diubah ke arsitektur *microservice*. Pembagian *service* pada SiakadCloud 2.0 ditunjukkan pada Gambar 3.2.



Gambar 3.2 Pembagian *Service* Pada SiakadCloud 2.0.

Pada penelitian ini akan dikembangkan *service* master yang akan menjadi referensi *service-service* selanjutnya dan juga untuk manajemen data master yang dapat digunakan pada *service* lainya. Kebutuhan fungsional merupakan kebutuhan aplikasi yang berhubungan dengan proses *input* dan *output* pada aplikasi. Kebutuhan fungsional dari pengembangan *service* master pada aplikasi SiakadCloud 2.0 ini adalah:

- 1) *Service* dapat manajemen data-data master.
- 2) *Service* dapat mengirimkan data master ke *service* lain.
- 3) Terdapat pengecekan hak akses.

Kebutuhan fungsional *service* akan dijelaskan lebih jelas pada *use case* diagram yang akan dibahas pada tahap *modeling*.

Adapun kebutuhan secara non-fungsional yaitu:

- 1) *Service* memberikan respon yang cepat dan tepat.
- 2) Terdapat dokumentasi api yang dapat memudahkan *frontend engineer*.
- 3) Memiliki tingkat keamanan yang baik.

2. *Planning*

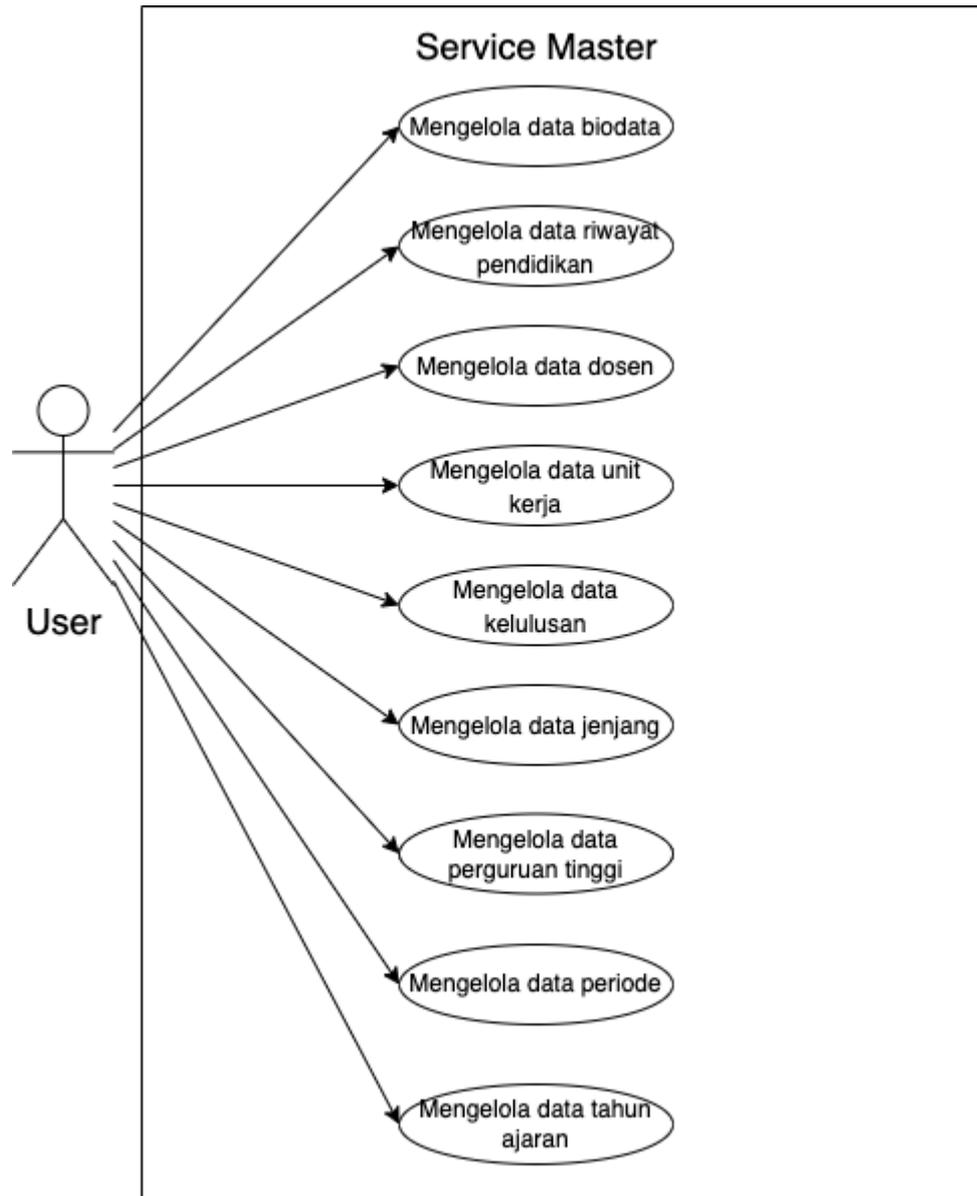
Tahap pengembangan sistem berikutnya adalah *planning*. *Planning* menjelaskan tentang estimasi tugas-tugas teknis yang akan dilakukan, produk kerja yang ingin dihasilkan, penjadwalan kerja yang akan dilaksanakan, dan *tracking* proses pengerjaan sistem. Tahapan ini akan menghasilkan *Gantt Chart* dari pengembangan sistem yang dapat dilihat pada Tabel 3.2.

3. *Modeling*

Modeling sendiri menjadi gambaran kasar sistem sebelum diimplementasi pada sebuah bahasa pemrograman. *Modeling* merupakan tahap perancangan dan pemodelan arsitektur sistem yang berfokus pada perancangan *Use case Diagram*, *Activity Diagram*, *ER Diagram*. Tujuannya untuk lebih memahami gambaran besar dari apa yang akan dikerjakan.

a. *Use Case Diagram*

Terdapat 1 aktor yang bertindak sebagai *user*, di mana *user* ini memiliki *role* nya masing-masing. *Role* tersebut diatur pada *service* auth sehingga pada *service* master hanya perlu melakukan pengecekan hak aksesnya saja yang diambil dari token *role*. *User* dapat melakukan kelola data biodata, riwayat pendidikan, dosen, unit kerja, kelulusan, jenjang, perguruan tinggi, periode dan tahun ajaran. *Use case* diagram pada *service* master dapat dilihat pada Gambar 3.3.



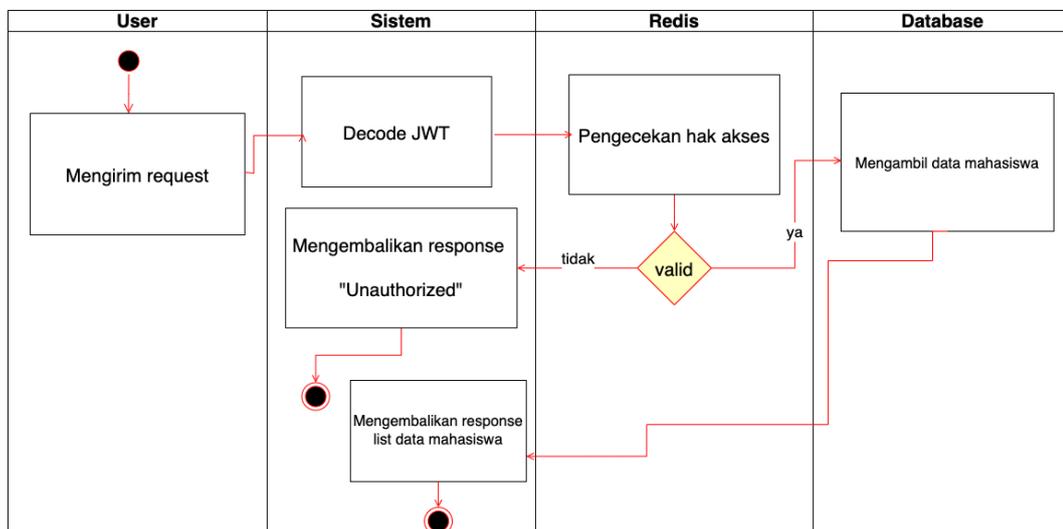
Gambar 3.3 *Use Case Diagram Service Master.*

b. Activity Diagram

Activity diagram merupakan gambaran dari proses-proses yang terjadi pada sebuah sistem. *Activity* diagram adalah pengembangan dari *Use Case* diagram yang memiliki alur aktivitas. *Activity* diagram pada *service* master memiliki proses pengelolaan yang tidak jauh berbeda, sehingga *activity* diagram yang dibuat hanya proses untuk menampilkan dan memasukkan data biodata mahasiswa saja.

1) Activity Diagram Menampilkan Data Mahasiswa

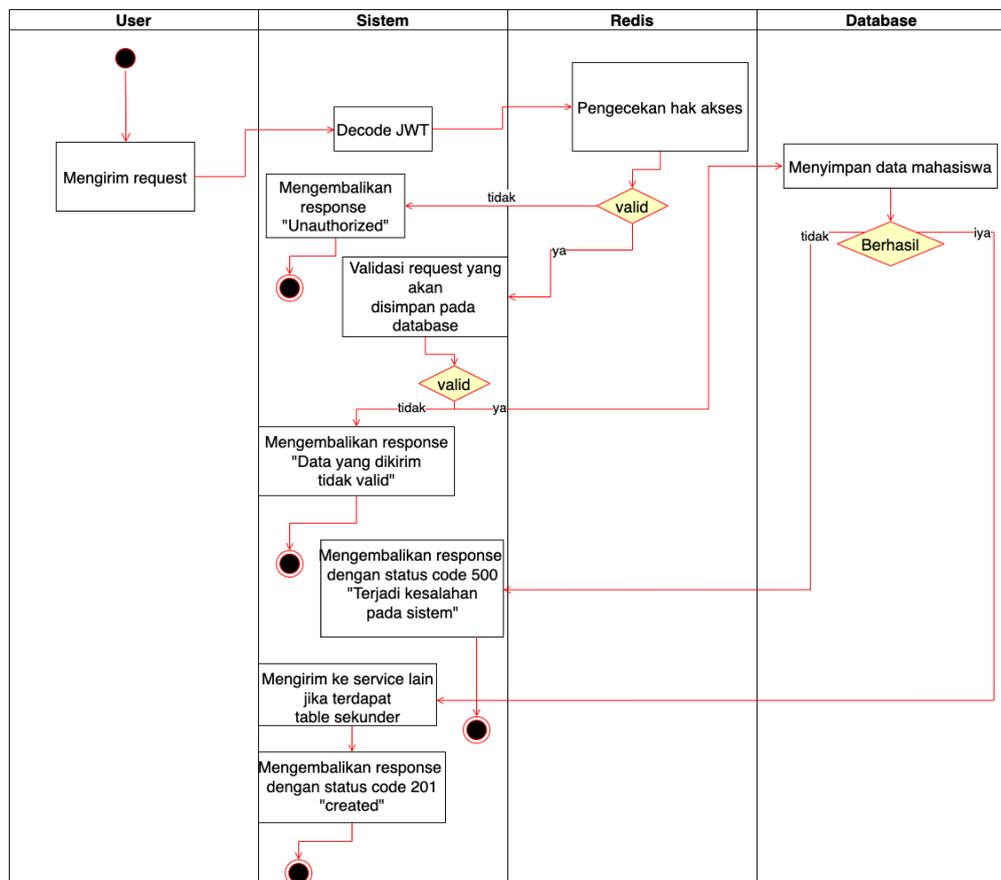
Activity yang pertama yaitu proses untuk menampilkan data mahasiswa. *User* yang ingin mendapatkan data mahasiswa harus membawa token *role* dulu yang mana token tersebut didapatkan dari *service* auth. *User* mengirimkan *request* ke *service* master kemudian pada *service* master maka akan diarahkan pada *middleware* pengecekan hak akses terlebih dahulu. Token yang sudah didecode maka akan mendapatkan *id_role* yang mana *id_role* akan dicek pada *redis*. Jika *role* tersebut valid maka *service* akan mengambil data dari *database* kemudian dari data yang dibawa akan dikembalikan sebagai *response* dalam bentuk JSON. Namun jika *role* tersebut tidak valid, maka *service* akan mengembalikan *response unauthorized*. Untuk lebih jelas *activity* diagram menampilkan data mahasiswa dapat dilihat pada Gambar 3.4.



Gambar 3.4 Activity Diagram Menampilkan Data Mahasiswa.

2) Activity Diagram Menambahkan Data Mahasiswa

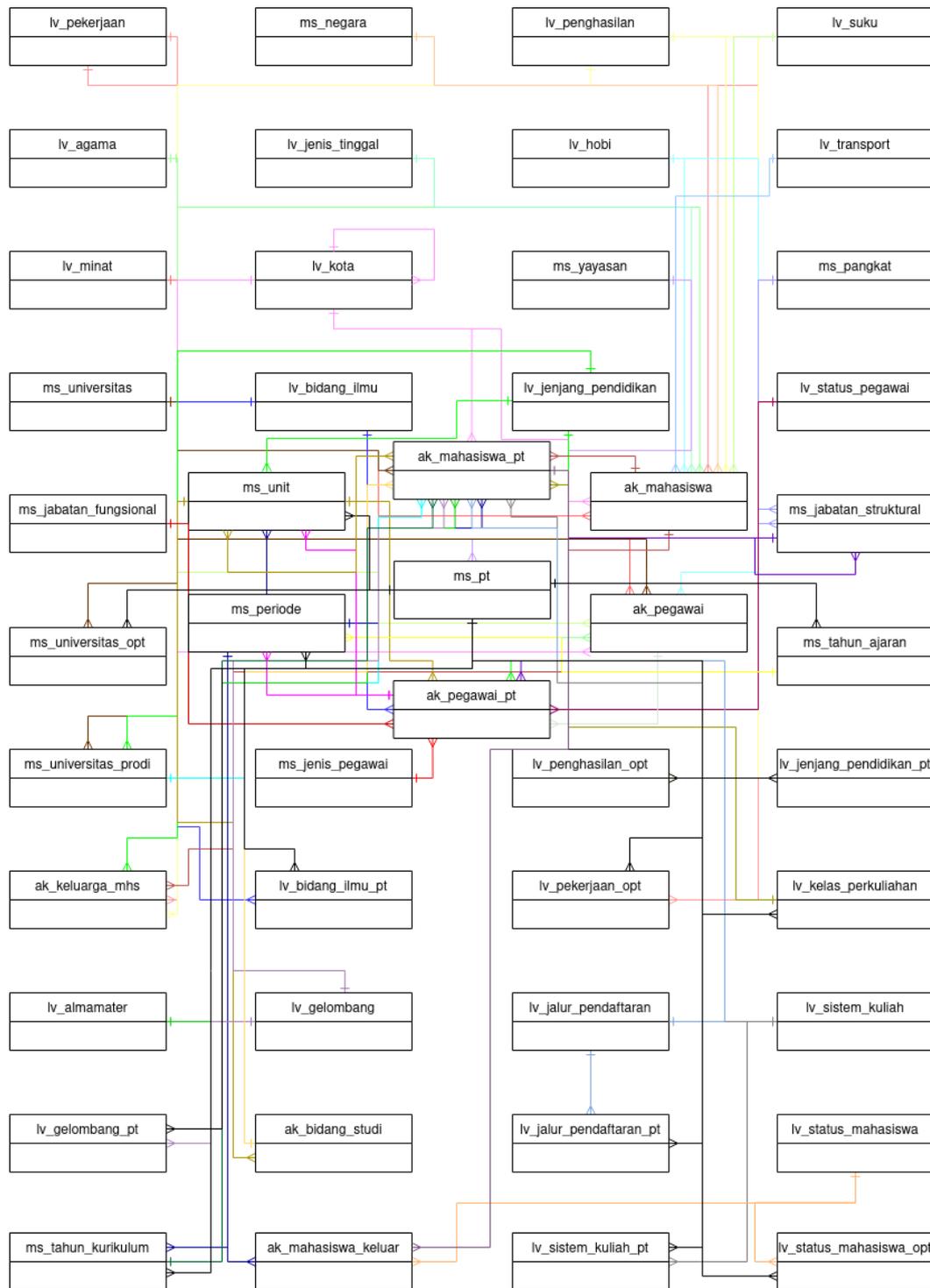
Activity diagram selanjutnya yaitu proses untuk menambahkan data mahasiswa. Proses menambahkan data mahasiswa terdapat 3 pengecekan, yang pertama yaitu validasi hak akses, pengecekan setelah melakukan *query* ke *database*. Jika gagal pada proses validasi *request* maka *service* akan mengembalikan *response* 400 dengan pesan *bad request*. Jika gagal pada proses *query* ke *database*, maka akan mengembalikan *response* 500 dengan pesan terjadi kesalahan pada sistem. Jika semua proses pengecekan berhasil maka *service* master akan mengirimkan ke table sekunder pada *service* lain yang membutuhkan data tersebut. Untuk lebih jelas *activity* diagram menambahkan data mahasiswa dapat dilihat pada Gambar 3.5.



Gambar 3.5 Activity Diagram Menambahkan Data Mahasiswa.

c. *Entity Relationship Diagram*

Entity relationship diagram merupakan suatu model atau rancangan untuk membuat *database* yang menggambarkan data yang memiliki hubungan atau relasi dalam bentuk sebuah desain. Service master memiliki 47 entitas, sehingga pada desain tidak disertakan atributnya karena memakan banyak ruang. Atribut pada setiap entitas akan dijelaskan pada kamus data. Berikut merupakan desain ERD pada *service* master.



Gambar 3.6 Entity Relationship Diagram Service Master.

d. Kamus Data

Kamus data digunakan untuk menjelaskan lebih detail mengenai atribut dari setiap entitas pada ERD, sehingga masing-masing atribut dapat menjelaskan maksudnya. Berikut merupakan kamus data yang merupakan penjelasan lebih lanjut dari ERD yang sudah dibuat.

1) Kamus Data Entitas `ak_bidang_studi`

Kamus data entitas `ak_bidang_studi` menjelaskan masing-masing atribut yang terdapat pada entitas `ak_bidang_studi`. Kamus data entitas `ak_bidang_studi` dapat dilihat pada tabel 3.3.

Tabel 3.3 Kamus data entitas `ak_bidang_studi`

| Nama Field | Tipe Data | Ukuran | Batasan |
|-----------------------------|---------------|--------|--|
| <code>id</code> | integer | | primary key |
| <code>id_unit</code> | integer | | foreign key dari <code>ms_unit</code> , unique, not null |
| <code>kode_bidang</code> | varchar | 5 | unique, not null |
| <code>nama_bidang</code> | varchar | 100 | not null |
| <code>nama_bidang_en</code> | varchar | 100 | |
| <code>created_at</code> | timestam p | | |
| <code>created_by</code> | integer | | |
| <code>updated_at</code> | timestam p | | |
| <code>updated_by</code> | integer | | |
| <code>updated_ip</code> | inet | | |
| <code>updated_path</code> | varchar | 50 | |

2) Kamus Data Entitas `ak_keluarga_mhs`

Kamus data entitas `ak_keluarga_mhs` menjelaskan masing-masing atribut yang terdapat pada entitas `ak_keluarga_mhs`. Kamus data entitas `ak_keluarga_mhs` dapat dilihat pada tabel 3.4.

Tabel 3.4 Kamus data entitas ak_keluarga_mhs

| Nama Field | Tipe Data | Ukuran | Batasan |
|-------------------|------------------|---------------|---|
| id | integer | | primary key |
| id_mahasiswa | integer | | foreign key dari ak_mahasiswa, unique |
| status_keluarga | char | 1 | not null, unique |
| nama | varchar | 200 | not null |
| tgl_akhir | date | | |
| alamat | varchar | 255 | |
| telepon | varchar | 20 | |
| email | varchar | 100 | |
| nik | varchar | 100 | |
| is_nik_aktif | char | 1 | not null |
| id_jenjang | integer | | foreign key dari lv_jenjang_pendidikan |
| id_pekerjaan | integer | | foreign key dari lv_pekerjaan |
| id_penghasilan | integer | | foreign key dari lv_penghasilan |
| instansi_kerja | varchar | 30 | |
| status_kondisi | char | 1 | |
| status_kerabat | char | 1 | |
| created_at | timestamp | | |
| created_by | integer | | |
| updated_at | timestamp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

3) Kamus Data Entitas ak_mahasiswa

Kamus data entitas ak_mahasiswa menjelaskan masing-masing atribut yang terdapat pada entitas ak_mahasiswa. Kamus data entitas ak_mahasiswa dapat dilihat pada tabel 3.5.

Tabel 3.5 Kamus data entitas ak_mahasiswa

| Nama Field | Tipe Data | Ukuran | Batasan |
|-------------------|------------------|---------------|----------------|
| id | integer | | primary key |
| nama | varchar | 200 | not null |
| gelar_depan | varchar | 30 | |

| Nama Field | Tipe Data | Ukuran | Batasan |
|------------------------|------------------|---------------|-----------------------------------|
| gelar_belakang | varchar | 30 | |
| jenis_kelamin | char | 1 | not null |
| tempat_lahir | varchar | 100 | not null |
| tgl_lahir | date | | not null |
| alamat | varchar | 255 | |
| telepon | varchar | 20 | |
| hp | varchar | 20 | |
| hp_2 | varchar | 20 | |
| email | varchar | 100 | |
| gol_darah | varchar | 2 | |
| status_nikah | char | 1 | |
| nik | varchar | 30 | |
| no_kk | varchar | 30 | |
| rt | varchar | 5 | |
| rw | varchar | 5 | |
| dusun | varchar | 100 | |
| desa | varchar | 100 | |
| no_kps | varchar | 20 | |
| id_agama | integer | | foreign key dari lv_agama |
| id_negara | integer | | foreign key dari ms_negara |
| id_kota | integer | | foreign key dari lv_kota |
| id_kecamatan | integer | | foreign key dari lv_kota |
| id_suku | integer | | foreign key dari lv_suku |
| id_pekerjaan | integer | | foreign key dari lv_pekerjaan |
| id_penghasilan | integer | | foreign key dari lv_penghasilan |
| id_tinggal | integer | | foreign key dari lv_jenis_tinggal |
| id_transport | integer | | foreign key dari lv_transport |
| id_hobi | integer | | foreign key dari lv_hobi |
| id_minat | integer | | foreign key dari lv_minat |
| no_rekening | varchar | 50 | |
| berat_badan | varchar | 3 | |
| tinggi_badan | varchar | 3 | |
| instansi_kerja | varchar | 100 | |
| email_ortu | varchar | 100 | |
| show_hp | char | 1 | not null |
| show_kota | char | 1 | not null |
| show_hobi | char | 1 | not null |
| show_minat | char | 1 | not null |
| is_valid_email | char | 1 | not null |
| id_file_akta_kelahiran | uuid | | |
| created_at | timestamp | | |

| Nama Field | Tipe Data | Ukuran | Batasan |
|-------------------|------------------|---------------|----------------|
| created_by | integer | | |
| updated_at | timestamp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

4) Kamus Data Entitas ak_mahasiswa_keluar

Kamus data entitas ak_mahasiswa_keluar menjelaskan masing-masing atribut yang terdapat pada entitas ak_mahasiswa_keluar. Kamus data entitas ak_mahasiswa_keluar dapat dilihat pada tabel 3.6.

Tabel 3.6 Kamus data entitas ak_mahasiswa_keluar

| Nama Field | Tipe Data | Ukuran | Batasan |
|-------------------|------------------|---------------|--|
| id | integer | | primary key |
| id_mahasiswa_pt | integer | | not null, foreign key dari ak_mahasiswa_pt, unique |
| id_periode | integer | | not null, foreign key dari ms_periode |
| id_status | integer | | not null, foreign key dari lv_status_mahasiswa |
| tgl_sk | date | | |
| no_sk | varchar | 100 | |
| id_file_sk | uuid | | |
| created_at | timestamp | | |
| created_by | integer | | |
| updated_at | timestamp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

5) Kamus Data Entitas ak_mahasiswa_pt

Kamus data entitas ak_mahasiswa_pt menjelaskan masing-masing atribut yang terdapat pada entitas ak_

mahasiswa_pt. Kamus data entitas ak_mahasiswa_pt dapat dilihat pada tabel 3.7.

Tabel 3.7 Kamus data entitas ak_mahasiswa_pt

| Nama Field | Tipe Data | Ukuran | Batasan |
|----------------------|------------------|---------------|---|
| id | integer | | primary key |
| id_mahasiswa | integer | | not null, foreign key pada ak_mahasiswa |
| nim | varchar | 24 | not null |
| id_periode | integer | | not null, foreign key pada ms_periode |
| id_unit | integer | | not null, foreign key pada ms_unit |
| id_tahun | integer | | not null, foreign key pada ms_tahun_kurikulum |
| id_jalur | integer | | foreign key dari lv_jalur_pendaftaran |
| id_gelombang | integer | | foreign key dari lv_gelombang |
| id_sistem | integer | | foreign key dari lv_sistem_kuliah |
| id_kelas | integer | | foreign key dari lv_kelas_perkuliahan |
| id_bidang | integer | | foreign key dari ak_bidang_studi |
| id_periode_akhir | integer | | foreign key dari ms_periode |
| email_kampus | varchar | 100 | |
| tgl_daftar | date | | |
| nilai_tpa | numeric | 5,2 | |
| nilai_kesehatan | numeric | 5,3 | |
| nilai_psikotes | numeric | 5,4 | |
| nilai_wawancara | numeric | 5,5 | |
| jenis_transfer | char | 1 | |
| kode_transfer | varchar | 2 | |
| kode_pendidikan_asal | varchar | 5 | |
| asal_smu | varchar | 50 | |
| alamat_smu | varchar | 1000 | |
| telepon_smu | varchar | 15 | |
| jurusan_smu | varchar | 30 | |
| thn_lulus_smu | integer | | |
| nem_smu | numeric | 5,2 | |
| no_ijasah_smu | varchar | 50 | |
| nisn | varchar | 20 | |
| nupn | varchar | 20 | |
| id_kota_smu | integer | | foreign key dari lv_kota |
| id_propinsi_smu | integer | | foreign key dari lv_kota |
| tgl_transfer | date | | |
| nim_lama | varchar | 24 | |
| asal_pt | varchar | 100 | |

| Nama Field | Tipe Data | Ukuran | Batasan |
|----------------------------|------------------|---------------|---------------------------------------|
| prodi_pt | varchar | 100 | |
| nim_pt | varchar | 20 | |
| ipk_asal | numeric | 3,2 | |
| sks_asal | integer | | |
| npsn | varchar | 20 | |
| id_periode_transf er | integer | | foreign key dari ms_periode |
| id_unit_asal | integer | | foreign key dari ms_unit |
| id_tahun_asal | integer | | foreign key dari ms_tahun_kurikulum |
| id_universitas | integer | | foreign key dari ms_universitas |
| id_universitas_pr odi | integer | | foreign key dari ms_universitas_prodi |
| nirm | varchar | 20 | |
| nirl | varchar | 20 | |
| id_almamater | integer | | foreign key dari lv_almamater |
| is_valid_email | char | 1 | |
| id_file_transkrip_ asal | uuid | | |
| id_file_ijazah_ak hir | uuid | | |
| id_file_surat_pind ah | uuid | | not null |
| created_at | timesta mp | | |
| created_by | integer | | |
| updated_at | timesta mp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

6) Kamus Data Entitas ak_pegawai

Kamus data entitas ak_pegawai menjelaskan masing-masing atribut yang terdapat pada entitas ak_pegawai.

Kamus data entitas ak_pegawai dapat dilihat pada tabel 3.8.

Tabel 3.8 Kamus data entitas ak_pegawai

| Nama Field | Tipe Data | Ukuran | Batasan |
|-------------------|------------------|---------------|----------------|
| id | integer | | primary key |
| nama | varchar | 100 | not null |

| Nama Field | Tipe Data | Ukuran | Batasan |
|-------------------|------------------|---------------|---------------------------------|
| gelar_depan | varchar | 30 | |
| gelar_belakang | varchar | 100 | |
| jenis_kelamin | char | 1 | not null |
| tgl_lahir | date | | |
| tempat_lahir | varchar | 100 | |
| alamat | varchar | 100 | |
| telepon | varchar | 20 | |
| email | varchar | 100 | |
| id_agama | integer | | foreign key dari lv_agama |
| id_universitas | integer | | foreign key dari ms_universitas |
| id_kota | integer | | foreign key dari lv_kota |
| id_hobi | integer | | foreign key dari lv_hobi |
| id_minat | integer | | foreign key dari lv_minat |
| show_hp | char | 1 | not null |
| show_email | char | 1 | not null |
| show_kota | char | 1 | not null |
| show_hobi | char | 1 | not null |
| show_minat | char | 1 | not null |
| is_valid_email | char | 1 | not null |
| id_file_ttd | uuid | | |
| created_at | timestamp | | |
| created_by | integer | | |
| updated_at | timestamp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

7) Kamus Data Entitas ak_pegawai_pt

Kamus data entitas ak_pegawai_pt menjelaskan masing-masing atribut yang terdapat pada entitas ak_pegawai_pt. Kamus data entitas ak_pegawai_pt dapat dilihat pada tabel 3.9.

Tabel 3.9 Kamus data entitas ak_pegawai_pt

| Nama Field | Tipe Data | Ukuran | Batasan |
|-------------------|------------------|---------------|---------------------------------------|
| id | integer | | primary key |
| id_pegawai | integer | | not null, foreign key dari ak_pegawai |
| nip | varchar | 20 | not null |
| nidn | varchar | 30 | |

| Nama Field | Tipe Data | Ukuran | Batasan |
|-------------------|---------------|--------|--|
| nupn | varchar | 50 | |
| nidk | varchar | 50 | |
| email_kampus | varchar | 100 | |
| id_unit | integer | | foreign key dari ms_unit |
| id_jenjang | integer | | foreign key dari lv_jenjang_pendidikan |
| id_pangkat | integer | | foreign key dari ms_pangkat |
| id_fungsional | integer | | foreign key dari ms_jabatan_fungsional |
| id_struktural | integer | | foreign key dari ms_jabatan_struktural |
| id_jenis | integer | | foreign key dari ms_jenis_pegawai |
| id_status | integer | | foreign key dari lv_status_pegawai |
| id_bidang | integer | | foreign key dari lv_bidang_ilmu |
| tugas_luar | varchar | 100 | not null |
| is_dosen_luar | char | 1 | not null |
| is_pengasuh | char | 1 | not null |
| is_pembina_ukm | char | 1 | not null |
| is_pembina_ekskul | char | 1 | not null |
| kuota_pa | integer | | |
| kuota_pembimbing | integer | | |
| is_valid_email | char | 1 | not null |
| created_at | timestam p | | |
| created_by | integer | | |
| updated_at | timestam p | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

8) Kamus Data Entitas lv_agama

Kamus data entitas lv_agama menjelaskan masing-masing atribut yang terdapat pada entitas lv_agama. Kamus data entitas lv_agama dapat dilihat pada tabel 3.10.

Tabel 3.10 Kamus data entitas lv_agama

| Nama Field | Tipe Data | Ukuran | Batasan |
|-------------------|------------------|---------------|------------------|
| id | integer | | primary key |
| kode_agama | integer | | not null, unique |
| nama_agama | varchar | 30 | not null |
| created_at | timestamp | | |
| created_by | integer | | |
| updated_at | timestamp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

9) Kamus Data Entitas lv_almamater

Kamus data entitas lv_almamater menjelaskan masing-masing atribut yang terdapat pada entitas lv_almamater. Kamus data entitas lv_almamater dapat dilihat pada tabel 3.11.

Tabel 3.11 Kamus data entitas lv_almamater

| Nama Field | Tipe Data | Ukuran | Batasan |
|-------------------|------------------|---------------|----------------|
| id | integer | | primary key |
| nama_almamater | varchar | 20 | not null |
| created_at | timestamp | | |
| created_by | integer | | |
| updated_at | timestamp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

10) Kamus Data Entitas lv_bidang_ilmu

Kamus data entitas lv_bidang_ilmu menjelaskan masing-masing atribut yang terdapat pada entitas lv_bidang_ilmu. Kamus data entitas lv_bidang_ilmu dapat dilihat pada tabel 3.12.

Tabel 3.12 Kamus data entitas lv_bidang_ilmu

| Nama Field | Tipe Data | Ukuran | Batasan |
|-------------------|------------------|---------------|------------------|
| id | integer | | primary key |
| kode_bidang | varchar | 5 | not null, unique |
| nama_bidang | varchar | 50 | not null |
| created_at | timestamp | | |
| created_by | integer | | |
| updated_at | timestamp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

11) Kamus Data Entitas lv_bidang_ilmu_pt

Kamus data entitas lv_bidang_ilmu_pt menjelaskan masing-masing atribut yang terdapat pada entitas lv_bidang_ilmu_pt. Kamus data entitas lv_bidang_ilmu_pt dapat dilihat pada tabel 3.13.

Tabel 3.13 Kamus data entitas lv_bidang_ilmu_pt

| Nama Field | Tipe Data | Ukuran | Batasan |
|-------------------|------------------|---------------|---|
| id | integer | | primary key |
| id_pt | integer | | not null, foreign key dari ms_pt, unique |
| id_bidang | integer | | not null, foreign key dari lv_bidang_ilmu, unique |
| created_at | timestamp | | |
| created_by | integer | | |
| updated_at | timestamp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

12) Kamus Data Entitas lv_gelombang

Kamus data entitas lv_gelombang menjelaskan masing-masing atribut yang terdapat pada entitas lv_gelombang. Kamus data entitas lv_gelombang dapat dilihat pada tabel 3.14.

Tabel 3.14 Kamus data entitas lv_gelombang

| Nama Field | Tipe Data | Ukuran | Batasan |
|-------------------|------------------|---------------|------------------|
| id | integer | | primary key |
| kode_gelombang | integer | | not null, unique |
| nama_gelombang | varchar | 100 | not null |
| created_at | timestamp | | |
| created_by | integer | | |
| updated_at | timestamp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

13) Kamus Data Entitas lv_gelombang_pt

Kamus data entitas lv_gelombang_pt menjelaskan masing-masing atribut yang terdapat pada entitas lv_gelombang_pt. Kamus data entitas lv_gelombang_pt dapat dilihat pada tabel 3.15.

Tabel 3.15 Kamus data entitas lv_gelombang_pt

| Nama Field | Tipe Data | Ukuran | Batasan |
|-------------------|------------------|---------------|---|
| id | integer | | primary key |
| id_pt | integer | | not null, foreign key dari ms_pt, unique |
| id_gelombang | integer | | not null, foreign key dari lv_gelombang, unique |
| created_at | timestamp | | |
| created_by | integer | | |
| updated_at | timestamp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

14) Kamus Data Entitas lv_hobi

Kamus data entitas lv_hobi menjelaskan masing-masing atribut yang terdapat pada entitas lv_hobi. Kamus data entitas lv_hobi dapat dilihat pada tabel 3.16.

Tabel 3.16 Kamus data entitas lv_hobi

| Nama Field | Tipe Data | Ukuran | Batasan |
|-------------------|------------------|---------------|----------------|
| id | integer | | primary key |
| nama_hobi | varchar | 100 | not null |
| created_at | timestamp | | |
| created_by | integer | | |
| updated_at | timestamp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

15) Kamus Data Entitas lv_jalur_pendaftaran

Kamus data entitas lv_jalur_pendaftaran menjelaskan masing-masing atribut yang terdapat pada entitas lv_jalur_pendaftaran. Kamus data entitas lv_jalur_pendaftaran dapat dilihat pada tabel 3.17.

Tabel 3.17 Kamus data entitas lv_jalur_pendaftaran

| Nama Field | Tipe Data | Ukuran | Batasan |
|-------------------|------------------|---------------|------------------|
| id | integer | | primary key |
| nama_jalur | varchar | 100 | not null, unique |
| keterangan | varchar | 1000 | |
| kode_transfer | varchar | 2 | not null |
| created_at | timestamp | | |
| created_by | integer | | |
| updated_at | timestamp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

16) Kamus Data Entitas lv_jalur_pendaftaran_pt

Kamus data entitas lv_jalur_pendaftaran_pt menjelaskan masing-masing atribut yang terdapat pada entitas lv_jalur_pendaftaran_pt. Kamus data entitas lv_jalur_pendaftaran_pt dapat dilihat pada tabel 3.18.

Tabel 3.18 Kamus data entitas lv_jalur_pendaftaran_pt

| Nama Field | Tipe Data | Ukuran | Batasan |
|-------------------|------------------|---------------|---|
| id | integer | | primary key |
| id_pt | integer | | not null, foreign key dari ms_pt, unique |
| id_jalur | integer | | not null, foreign key dari lv_jalur_pendaftaran, unique |
| created_at | timestamp | | |
| created_by | integer | | |
| updated_at | timestamp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

17) Kamus Data Entitas lv_jenis_tinggal

Kamus data entitas lv_jenis_tinggal menjelaskan masing-masing atribut yang terdapat pada entitas lv_jenis_tinggal. Kamus data entitas lv_jenis_tinggal dapat dilihat pada tabel 3.19.

Tabel 3.19 Kamus data entitas lv_jenis_tinggal

| Nama Field | Tipe Data | Ukuran | Batasan |
|-------------------|------------------|---------------|------------------|
| id | integer | | primary key |
| kode_tinggal | integer | | not null, unique |
| nama_tinggal | varchar | 100 | not null |
| created_at | timestamp | | |
| created_by | integer | | |
| updated_at | timestamp | | |
| updated_by | integer | | |

| Nama Field | Tipe Data | Ukuran | Batasan |
|--------------|-----------|--------|---------|
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

18) Kamus Data Entitas lv_jenjang_pendidikan

Kamus data entitas lv_jenjang_pendidikan menjelaskan masing-masing atribut yang terdapat pada entitas lv_jenjang_pendidikan. Kamus data entitas lv_jenjang_pendidikan dapat dilihat pada tabel 3.20.

Tabel 3.20 Kamus data entitas lv_jenjang_pendidikan

| Nama Field | Tipe Data | Ukuran | Batasan |
|-----------------|-----------|--------|------------------|
| id | integer | | primary key |
| kode_jenjang | varchar | 5 | not null, unique |
| nama_jenjang | varchar | 100 | not null |
| nama_jenjang_en | varchar | 100 | |
| is_akademik | char | 1 | not null |
| is_univ | char | 1 | not null |
| urutan | integer | | |
| created_at | timestamp | | |
| created_by | integer | | |
| updated_at | timestamp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

19) Kamus Data Entitas lv_jenjang_pendidikan_pt

Kamus data entitas lv_jenjang_pendidikan_pt menjelaskan masing-masing atribut yang terdapat pada entitas lv_jenjang_pendidikan_pt. Kamus data entitas lv_jenjang_pendidikan_pt dapat dilihat pada tabel 3.21.

Tabel 3.21 Kamus data entitas lv_jenjang_pendidikan_pt

| Nama Field | Tipe Data | Ukuran | Batasan |
|------------|-----------|--------|--|
| id | integer | | primary key |
| id_pt | integer | | not null, foreign key dari ms_pt, unique |

| Nama Field | Tipe Data | Ukuran | Batasan |
|-------------------|------------------|---------------|--|
| id_jenjang | integer | | not null, foreign key dari lv_jenjang_pendidikan, unique |
| max_cuti | integer | | |
| max_studi | integer | | |
| masa_studi | integer | | |
| default_nilai | numeric | 3,2 | |
| kode_nim | varchar | | |
| kode_emis | varchar | | |
| kode_emis_pasca | varchar | | |
| deskripsi | varchar | | |
| is_pt | char | 1 | not null |
| created_at | timestamp | | |
| created_by | integer | | |
| updated_at | timestamp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

20) Kamus Data Entitas lv_kelas_perkuliahan

Kamus data entitas lv_kelas_perkuliahan menjelaskan masing-masing atribut yang terdapat pada entitas lv_kelas_perkuliahan. Kamus data entitas lv_kelas_perkuliahan dapat dilihat pada tabel 3.22.

Tabel 3.22 Kamus data entitas lv_kelas_perkuliahan

| Nama Field | Tipe Data | Ukuran | Batasan |
|-------------------|------------------|---------------|--|
| id | integer | | primary key |
| id_pt | integer | | not null, foreign key dari ms_pt, unique |
| kode_kelas | varchar | 5 | not null, unique |
| nama_kelas | varchar | 50 | not null |
| created_at | timestamp | | |
| created_by | integer | | |
| updated_at | timestamp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

21) Kamus Data Entitas lv_kota

Kamus data entitas lv_kota menjelaskan masing-masing atribut yang terdapat pada entitas lv_kota. Kamus data entitas lv_kota dapat dilihat pada tabel 3.23.

Tabel 3.23 Kamus data entitas lv_kota

| Nama Field | Tipe Data | Ukuran | Batasan |
|--------------|-----------|--------|--------------------------|
| id | integer | | primary key |
| id_parent | integer | | foreign key dari lv_kota |
| kode_kota | varchar | 10 | not null, unique |
| nama_kota | varchar | 100 | not null |
| level | char | 1 | not null |
| kode_iso | varchar | 5 | |
| kode_dikti | varchar | 10 | |
| created_at | timestamp | | |
| created_by | integer | | |
| updated_at | timestamp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

22) Kamus Data Entitas lv_minat

Kamus data entitas lv_minat menjelaskan masing-masing atribut yang terdapat pada entitas lv_minat. Kamus data entitas lv_minat dapat dilihat pada tabel 3.24.

Tabel 3.24 Kamus data entitas lv_minat

| Nama Field | Tipe Data | Ukuran | Batasan |
|--------------|-----------|--------|-------------|
| id | integer | | primary key |
| nama_minat | varchar | 100 | not null |
| created_at | timestamp | | |
| created_by | integer | | |
| updated_at | timestamp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

23) Kamus Data Entitas lv_pekerjaan

Kamus data entitas lv_pekerjaan menjelaskan masing-masing atribut yang terdapat pada entitas lv_pekerjaan. Kamus data entitas lv_pekerjaan dapat dilihat pada tabel 3.25.

Tabel 3.25 Kamus data entitas lv_pekerjaan

| Nama Field | Tipe Data | Ukuran | Batasan |
|----------------|-----------|--------|------------------|
| id | integer | | primary key |
| kode_pekerjaan | integer | | not null, unique |
| nama_pekerjaan | varchar | 100 | not null |
| created_at | timestamp | | |
| created_by | integer | | |
| updated_at | timestamp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

24) Kamus Data Entitas lv_pekerjaan_opt

Kamus data entitas lv_pekerjaan_opt menjelaskan masing-masing atribut yang terdapat pada entitas lv_pekerjaan_opt. Kamus data entitas lv_pekerjaan_opt dapat dilihat pada tabel 3.26.

Tabel 3.26 Kamus data entitas lv_pekerjaan_opt

| Nama Field | Tipe Data | Ukuran | Batasan |
|-------------------|-----------|--------|---|
| id | integer | | primary key |
| id_pt | integer | | not null, foreign key dari ms_pt, unique |
| id_pekerjaan | integer | | not null, foreign key dari lv_pekerjaan, unique |
| kode_emis | varchar | 2 | |
| kode_emis_mhs | varchar | 2 | |
| kode_emis_lulusan | varchar | 2 | |
| created_at | timestamp | | |
| created_by | integer | | |

| Nama Field | Tipe Data | Ukuran | Batasan |
|--------------|-----------|--------|---------|
| updated_at | timestamp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

25) Kamus Data Entitas lv_penghasilan

Kamus data entitas lv_penghasilan menjelaskan masing-masing atribut yang terdapat pada entitas lv_penghasilan. Kamus data entitas lv_penghasilan dapat dilihat pada tabel 3.27.

Tabel 3.27 Kamus data entitas lv_penghasilan

| Nama Field | Tipe Data | Ukuran | Batasan |
|------------------|-----------|--------|------------------|
| id | integer | | primary key |
| kode_penghasilan | varchar | | not null, unique |
| nama_penghasilan | varchar | 100 | not null |
| poin_bidik_misi | integer | | not null |
| created_at | timestamp | | |
| created_by | integer | | |
| updated_at | timestamp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

26) Kamus Data Entitas lv_penghasilan_opt

Kamus data entitas lv_penghasilan_opt menjelaskan masing-masing atribut yang terdapat pada entitas lv_penghasilan_opt. Kamus data entitas lv_penghasilan_opt dapat dilihat pada tabel 3.28.

Tabel 3.28 Kamus data entitas lv_penghasilan_opt

| Nama Field | Tipe Data | Ukuran | Batasan |
|------------|-----------|--------|--|
| id | integer | | primary key |
| id_pt | integer | | not null, foreign key dari ms_pt, unique |

| Nama Field | Tipe Data | Ukuran | Batasan |
|-------------------|------------------|---------------|---|
| id_penghasilan | integer | | not null, foreign key dari lv_penghasilan, unique |
| kode_emis | varchar | 2 | |
| created_at | timestamp | | |
| created_by | integer | | |
| updated_at | timestamp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

27) Kamus Data Entitas lv_sistem_kuliah

Kamus data entitas lv_sistem_kuliah menjelaskan masing-masing atribut yang terdapat pada entitas lv_sistem_kuliah. Kamus data entitas lv_sistem_kuliah dapat dilihat pada tabel 3.29.

Tabel 3.29 Kamus data entitas lv_sistem_kuliah

| Nama Field | Tipe Data | Ukuran | Batasan |
|-------------------|------------------|---------------|------------------|
| id | integer | | primary key |
| kode_sistem | integer | | not null, unique |
| nama_sistem | varchar | 100 | not null |
| keterangan | varchar | 1000 | |
| is_reguler | char | 1 | not null |
| created_at | timestamp | | |
| created_by | integer | | |
| updated_at | timestamp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

28) Kamus Data Entitas lv_sistem_kuliah_pt

Kamus data entitas lv_sistem_kuliah_pt menjelaskan masing-masing atribut yang terdapat pada entitas lv_sistem_kuliah_pt. Kamus data entitas lv_sistem_kuliah_pt dapat dilihat pada tabel 3.30.

Tabel 3.30 Kamus data entitas lv_sistem_kuliah_pt

| Nama Field | Tipe Data | Ukuran | Batasan |
|--------------|-----------|--------|---|
| id | integer | | primary key |
| id_pt | integer | | not null, foreign key dari ms_pt, unique |
| id_sistem | integer | | not null, foreign key dari lv_sistem_kuliah, unique |
| created_at | timestamp | | |
| created_by | integer | | |
| updated_at | timestamp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

29) Kamus Data Entitas lv_status_mahasiswa

Kamus data entitas lv_status_mahasiswa menjelaskan masing-masing atribut yang terdapat pada entitas lv_status_mahasiswa. Kamus data entitas lv_status_mahasiswa dapat dilihat pada tabel 3.31.

Tabel 3.31 Kamus data entitas lv_status_mahasiswa

| Nama Field | Tipe Data | Ukuran | Batasan |
|--------------|-----------|--------|------------------|
| id | integer | | primary key |
| kode_status | varchar | 2 | not null, unique |
| nama_status | varchar | 50 | not null |
| created_at | timestamp | | |
| created_by | integer | | |
| updated_at | timestamp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

30) Kamus Data Entitas lv_status_mahasiswa_opt

Kamus data entitas lv_status_mahasiswa_opt menjelaskan masing-masing atribut yang terdapat pada entitas

lv_status_mahasiswa_opt. Kamus data entitas lv_status_mahasiswa_opt dapat dilihat pada tabel 3.32.

Tabel 3.32 Kamus data entitas lv_status_mahasiswa_opt

| Nama Field | Tipe Data | Ukuran | Batasan |
|--------------|-----------|--------|--|
| id | integer | | primary key |
| id_pt | integer | | not null, foreign key dari ms_pt, unique |
| id_status | integer | | not null, foreign key dari lv_status_mahasiswa, unique |
| kode_emis | varchar | 2 | |
| is_diajukan | char | 1 | not null |
| created_at | timestamp | | |
| created_by | integer | | |
| updated_at | timestamp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

31) Kamus Data Entitas lv_status_pegawai

Kamus data entitas lv_status_pegawai menjelaskan masing-masing atribut yang terdapat pada entitas lv_status_pegawai. Kamus data entitas lv_status_pegawai dapat dilihat pada tabel 3.33.

Tabel 3.33 Kamus data entitas lv_status_pegawi

| Nama Field | Tipe Data | Ukuran | Batasan |
|--------------|-----------|--------|-------------|
| id | integer | | primary key |
| kode_status | varchar | 2 | not null |
| nama_status | varchar | 100 | not null |
| is_keluar | char | 1 | not null |
| created_at | timestamp | | |
| created_by | integer | | |
| updated_at | timestamp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

32) Kamus Data Entitas lv_suku

Kamus data entitas lv_suku menjelaskan masing-masing atribut yang terdapat pada entitas lv_suku. Kamus data entitas lv_suku dapat dilihat pada tabel 3.34.

Tabel 3.34 Kamus data entitas lv_suku

| Nama Field | Tipe Data | Ukuran | Batasan |
|-------------------|------------------|---------------|----------------|
| id | integer | | primary key |
| kode_suku | varchar | | not null |
| nama_suku | varchar | 100 | not null |
| created_at | timestamp | | |
| created_by | integer | | |
| updated_at | timestamp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

33) Kamus Data Entitas lv_transport

Kamus data entitas lv_transport menjelaskan masing-masing atribut yang terdapat pada entitas lv_transport. Kamus data entitas lv_transport dapat dilihat pada tabel 3.35.

Tabel 3.35 Kamus data entitas lv_transport

| Nama Field | Tipe Data | Ukuran | Batasan |
|-------------------|------------------|---------------|------------------|
| id | integer | | primary key |
| kode_transport | integer | | not null, unique |
| nama_transport | varchar | 100 | not null |
| created_at | timestamp | | |
| created_by | integer | | |
| updated_at | timestamp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

34) Kamus Data Entitas ms_jabatan_fungsional

Kamus data entitas ms_jabatan_fungsional menjelaskan masing-masing atribut yang terdapat pada entitas ms_jabatan_fungsional. Kamus data entitas ms_jabatan_fungsional dapat dilihat pada tabel 3.36.

Tabel 3.36 Kamus data entitas ms_jabatan_fungsional

| Nama Field | Tipe Data | Ukuran | Batasan |
|-----------------|-----------|--------|------------------|
| id | integer | | primary key |
| kode_fungsional | varchar | 5 | not null, unique |
| nama_fungsional | varchar | 100 | not null |
| created_at | timestamp | | |
| created_by | integer | | |
| updated_at | timestamp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

35) Kamus Data Entitas ms_jabatan_struktural

Kamus data entitas ms_jabatan_struktural menjelaskan masing-masing atribut yang terdapat pada entitas ms_jabatan_struktural. Kamus data entitas ms_jabatan_struktural dapat dilihat pada tabel 3.37.

Tabel 3.37 Kamus data entitas ms_jabatan_struktural

| Nama Field | Tipe Data | Ukuran | Batasan |
|-----------------|-----------|--------|--|
| id | integer | | primary key |
| kode_struktural | varchar | 10 | not null, unique |
| nama_struktural | varchar | 100 | not null |
| nama_singkat | varchar | 50 | |
| id_parent | integer | | foreign key dari ms_jabatan_struktural |
| id_pangkat_min | integer | | foreign key dari ms_pangkat |
| id_pangkat_max | integer | | foreign key dari ms_pangkat |
| kode_eselon | varchar | 3 | |
| keterangan | varchar | 255 | |
| is_pimpinan | char | 1 | not null |
| info_left | integer | | not null |
| info_right | integer | | not null |

| Nama Field | Tipe Data | Ukuran | Batasan |
|-------------------|------------------|---------------|----------------|
| created_at | timestamp | | |
| created_by | integer | | |
| updated_at | timestamp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

36) Kamus Data Entitas ms_jenis_pegawai

Kamus data entitas ms_jenis_pegawai menjelaskan masing-masing atribut yang terdapat pada entitas ms_jenis_pegawai. Kamus data entitas ms_jenis_pegawai dapat dilihat pada tabel 3.38.

Tabel 3.38 Kamus data entitas ms_jenis_pegawai

| Nama Field | Tipe Data | Ukuran | Batasan |
|-------------------|------------------|---------------|----------------|
| id | integer | | primary key |
| kode_jenis | varchar | 5 | not null |
| nama_jenis | varchar | 100 | not null |
| tipe | char | 1 | |
| created_at | timestamp | | |
| created_by | integer | | |
| updated_at | timestamp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

37) Kamus Data Entitas ms_negara

Kamus data entitas ms_negara menjelaskan masing-masing atribut yang terdapat pada entitas ms_negara. Kamus data entitas ms_negara dapat dilihat pada tabel 3.39.

Tabel 3.39 Kamus data entitas ms_negara

| Nama Field | Tipe Data | Ukuran | Batasan |
|-------------------|------------------|---------------|------------------|
| id | integer | | primary key |
| kode_negara | varchar | 5 | not null, unique |
| nama_kota | varchar | 100 | not null |
| kode_emis | varchar | 2 | |
| created_at | timestamp | | |
| created_by | integer | | |
| updated_at | timestamp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

38) Kamus Data Entitas ms_pangkat

Kamus data entitas ms_pangkat menjelaskan masing-masing atribut yang terdapat pada entitas ms_pangkat. Kamus data entitas ms_pangkat dapat dilihat pada tabel 3.40.

Tabel 3.40 Kamus data entitas ms_pangkat

| Nama Field | Tipe Data | Ukuran | Batasan |
|-------------------|------------------|---------------|------------------|
| id | integer | | primary key |
| kode_pangkat | varchar | 5 | not null, unique |
| nama_pangkat | varchar | 100 | not null |
| created_at | timestamp | | |
| created_by | integer | | |
| updated_at | timestamp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

39) Kamus Data Entitas ms_periode

Kamus data entitas ms_periode menjelaskan masing-masing atribut yang terdapat pada entitas ms_periode. Kamus data entitas ms_periode dapat dilihat pada tabel 3.41.

Tabel 3.41 Kamus data entitas ms_periode

| Nama Field | Tipe Data | Ukuran | Batasan |
|---------------------|-----------|--------|--|
| id | integer | | primary key |
| id_pt | integer | | not null, foreign key dari ms_pt, unique |
| kode_periode | varchar | 5 | not null, unique |
| nama_periode | varchar | 100 | not null |
| nama_singkat | varchar | 50 | |
| id_tahun_ajaran | integer | | not null, foreign key dari ms_tahun_ajaran |
| id_ketua_ujian | integer | | foreign key dari ak_pegawai_pt |
| tgl_awal | date | | |
| tgl_akhir | date | | |
| tgl_awal_uts | date | | |
| tgl_akhir_uts | date | | |
| tgl_awal_uas | date | | |
| tgl_akhir_uas | date | | |
| bulan_awal_tagihan | varchar | 6 | |
| bulan_akhir_tagihan | varchar | 6 | |
| pertemuan_kuliah | integer | | |
| minimal_absensi | numeric | 5,2 | |
| is_aktif | char | 1 | not null |
| created_at | timestamp | | |
| created_by | integer | | |
| updated_at | timestamp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

40) Kamus Data Entitas ms_pt

Kamus data entitas ms_pt menjelaskan masing-masing atribut yang terdapat pada entitas ms_pt. Kamus data entitas ms_pt dapat dilihat pada tabel 3.42.

Tabel 3.42 Kamus data entitas ms_pt

| Nama Field | Tipe Data | Ukuran | Batasan |
|-------------|-----------|--------|-------------|
| id | integer | | primary key |
| kode_pt | varchar | 20 | not null |
| nama_pt | varchar | 100 | not null |
| is_fakultas | char | 1 | not null |
| is_jurusan | char | 1 | not null |

| Nama Field | Tipe Data | Ukuran | Batasan |
|--------------|-----------|--------|-----------------------------|
| id_yayasan | integer | | foreign key dari ms_yayasan |
| created_at | timestamp | | |
| created_by | integer | | |
| updated_at | timestamp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

41) Kamus Data Entitas ms_tahun_ajaran

Kamus data entitas ms_tahun_ajaran menjelaskan masing-masing atribut yang terdapat pada entitas ms_tahun_ajaran. Kamus data entitas ms_tahun_ajaran dapat dilihat pada tabel 3.43.

Tabel 3.43 Kamus data entitas ms_tahun_ajaran

| Nama Field | Tipe Data | Ukuran | Batasan |
|-------------------|-----------|--------|--|
| id | integer | | primary key |
| id_pt | integer | | not null, foreign key dari ms_pt, unique |
| tahun_ajaran | varchar | 4 | not null, unique |
| nama_tahun_ajaran | varchar | 100 | not null |
| created_at | timestamp | | |
| created_by | integer | | |
| updated_at | timestamp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

42) Kamus Data Entitas ms_tahun_kurikulum

Kamus data entitas ms_tahun_kurikulum menjelaskan masing-masing atribut yang terdapat pada entitas ms_tahun_kurikulum. Kamus data entitas ms_tahun_kurikulum dapat dilihat pada tabel 3.44.

Tabel 3.44 Kamus data entitas ms_tahun_kurikulum

| Nama Field | Tipe Data | Ukuran | Batasan |
|------------|-----------|--------|-------------|
| id | integer | | primary key |

| Nama Field | Tipe Data | Ukuran | Batasan |
|-----------------|-----------|--------|--|
| id_pt | integer | | not null, foreign key dari ms_pt, unique |
| tahun_kurikulum | varchar | 4 | not null, unique |
| nama_kurikulum | varchar | 100 | not null |
| tgl_awal | date | | |
| tgl_akhir | date | | |
| id_periode | integer | | foreign key dari ms_periode |
| created_at | timestamp | | |
| created_by | integer | | |
| updated_at | timestamp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

43) Kamus Data Entitas ms_unit

Kamus data entitas ms_unit menjelaskan masing-masing atribut yang terdapat pada entitas ms_unit. Kamus data entitas ms_unit dapat dilihat pada tabel 3.45.

Tabel 3.45 Kamus data entitas ms_unit

| Nama Field | Tipe Data | Ukuran | Batasan |
|---------------|-----------|--------|--|
| id | integer | | primary key |
| id_pt | integer | | not null, foreign key dari ms_pt, unique |
| nama_unit | varchar | 100 | not null |
| nama_unit_en | varchar | 100 | |
| nama_singkat | varchar | 50 | |
| id_parent | integer | | foreign key dari ms_unit |
| id_jenjang | integer | | foreign key dari lv_jenjang_pendidikan |
| id_ketua | integer | | foreign key dari ak_pegawai_pt |
| id_sekretaris | integer | | foreign key dari ak_pegawai_pt |
| id_pembantu_1 | integer | | foreign key dari ak_pegawai_pt |
| id_pembantu_2 | integer | | foreign key dari ak_pegawai_pt |
| id_pembantu_3 | integer | | foreign key dari ak_pegawai_pt |
| id_pembantu_4 | integer | | foreign key dari ak_pegawai_pt |
| jenis_unit | char | 1 | not null |
| alamat | varchar | 100 | |
| telepon | varchar | 20 | |
| email | varchar | 100 | |
| website | varchar | 100 | |
| gelar | varchar | 10 | |
| gelar_en | varchar | 10 | |

| Nama Field | Tipe Data | Ukuran | Batasan |
|--------------------|-----------|--------|-----------------------------|
| deskripsi_gelar | varchar | 100 | |
| deskripsi_gelar_en | varchar | 100 | |
| akreditasi | varchar | 2 | |
| sk_akreditasi | varchar | 100 | |
| tgl_akreditasi | date | | |
| tgl_sk_pendirian | date | | |
| id_periode_berdiri | integer | | foreign key dari ms_periode |
| sks_lulus_min | integer | | |
| ipk_lulus_min | numeric | 3,2 | |
| batas_sks_awal | integer | | |
| jumlah_pembimbing | integer | | |
| jumlah_penguji | integer | | |
| id_jenis_ta | char | 1 | |
| kode_nim | varchar | 3 | |
| foto | varchar | 100 | |
| keterangan | varchar | 5000 | |
| visi | varchar | 5000 | |
| misi | varchar | 5001 | |
| kompetensi | varchar | 5002 | |
| cp | varchar | 5003 | |
| is_aktif | char | 1 | not null |
| is_eksternal | char | 1 | not null |
| info_left | integer | | not null |
| info_right | integer | | not null |
| created_at | timestamp | | |
| created_by | integer | | |
| updated_at | timestamp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

44) Kamus Data Entitas ms_universitas

Kamus data entitas ms_universitas menjelaskan masing-masing atribut yang terdapat pada entitas ms_universitas. Kamus data entitas ms_universitas dapat dilihat pada tabel 3.46.

Tabel 3.46 Kamus data entitas ms_universitas

| Nama Field | Tipe Data | Ukuran | Batasan |
|------------|-----------|--------|-------------|
| id | integer | | primary key |

| Nama Field | Tipe Data | Ukuran | Batasan |
|-------------------|------------------|---------------|----------------|
| kode_universitas | varchar | 20 | not null |
| nama_universitas | varchar | 100 | not null |
| alamat | varchar | 100 | |
| telepon | varchar | 20 | |
| created_at | timestamp | | |
| created_by | integer | | |
| updated_at | timestamp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

45) Kamus Data Entitas ms_universitas_opt

Kamus data entitas ms_universitas_opt menjelaskan masing-masing atribut yang terdapat pada entitas ms_universitas_opt. Kamus data entitas ms_universitas_opt dapat dilihat pada tabel 3.47.

Tabel 3.47 Kamus data entitas ms_universitas_opt

| Nama Field | Tipe Data | Ukuran | Batasan |
|-------------------|------------------|---------------|---|
| id | integer | | primary key |
| id_pt | integer | | not null, foreign key dari ms_pt, unique |
| id_universitas | integer | | not null, foreign key dari ms_universitas, unique |
| is_pmb | char | 1 | not null |
| created_at | timestamp | | |
| created_by | integer | | |
| updated_at | timestamp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

46) Kamus Data Entitas ms_universitas_prodi

Kamus data entitas ms_universitas_prodi menjelaskan masing-masing atribut yang terdapat pada entitas ms_universitas_prodi. Kamus data entitas ms_universitas_prodi dapat dilihat pada tabel 3.48.

Tabel 3.48 Kamus data entitas ms_universitas_prodi

| Nama Field | Tipe Data | Ukuran | Batasan |
|----------------|-----------|--------|---|
| id | integer | | primary key |
| id_universitas | integer | | not null, foreign key dari ms_universitas, unique |
| kode_prodi | varchar | 10 | not null, unique |
| nama_prodi | varchar | 100 | not null |
| id_jenjang | integer | | foreign key dari lv_jenjang_pendidikan |
| alamat | varchar | 100 | |
| telepon | varchar | 20 | |
| created_at | timestamp | | |
| created_by | integer | | |
| updated_at | timestamp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

47) Kamus Data Entitas ms_yayasan

Kamus data entitas ms_yayasan menjelaskan masing-masing atribut yang terdapat pada entitas ms_yayasan. Kamus data entitas ms_yayasan dapat dilihat pada tabel 3.49.

Tabel 3.49 Kamus data entitas ms_yayasan

| Nama Field | Tipe Data | Ukuran | Batasan |
|--------------|-----------|--------|-------------|
| id | integer | | primary key |
| nama_yayasan | varchar | 100 | not null |
| created_at | timestamp | | |
| created_by | integer | | |
| updated_at | timestamp | | |
| updated_by | integer | | |
| updated_ip | inet | | |
| updated_path | varchar | 50 | |

4. Construction

Construction merupakan proses penerjemahan bentuk desain atau model yang sudah dibuat pada tahap sebelumnya menjadi bentuk kode atau bahasa yang yang dapat dimengerti oleh mesin. Hasil dari *Construction* dijelaskan pada bab 4 hasil dan pembahasan. Setelah melakukan koding, maka dilakukan pengujian dengan metode *Black Box Testing*. Skenario pengujian dapat dilihat pada poin 3.5 tentang Skenario Pengujian Sistem. Tujuannya untuk menemukan kesalahan yang mungkin terjadi untuk nantinya diperbaiki. Setelah dilakukan testing maka tahap selanjutnya yaitu membuat *API documentation* yang akan dibuat dengan Swagger.

5. Deployment

Tahapan terakhir dari metode *waterfall* adalah *deployment* yang merupakan tahapan membuat *service* yang sudah dikembangkan menjadi *live* atau biasa disebut dengan *hosting*, sehingga memudahkan *programmer service* lain dalam penggunaannya jika dibutuhkan dalam proses pengembangannya. Setelah melakukan tahap *deployment*, pengembang mengimplementasikan CI (*Continuous Integration*)/CD (*Continuous Delivery*) agar setiap terjadi perubahan kode pada *service* tidak langsung di-*push* ke dalam *production*, namun masuk ke dalam fase *review* dan *staging* terlebih dahulu. Di mana perubahan atau penambahan kode tersebut dapat dilakukan *test* sebelum kode diimplementasikan ke dalam fase *production*. Sehingga melalui tes otomatis ini akan lebih mudah mendeteksi *bug* pada tahap pengembangan awal, jadi tidak perlu khawatir dengan munculnya *error* pada detik-detik terakhir.

3.3.4. Skenario Pengujian Sistem

Pengujian sistem pada penelitian ini yaitu menggunakan metode *black box testing*. Jenis *black box testing* yang digunakan pada penelitian ini adalah *Equivalence Partitioning*. *Equivalence Partitioning* merupakan metode pengujian pada *black box testing* dengan menggunakan skenario uji, hasil yang diharapkan dan hasil pengujian untuk melihat apakah sistem berjalan sesuai atau tidak (Adi Pranata et al., 2018). *Tools* yang akan digunakan dalam pengujian ini yaitu PHPUnit. Berikut merupakan skenario dalam pengujian *service master*.

Tabel 3.50 Skenario Pengujian *Equivalence Partitioning* pada *Service Master*

| Kode Uji | Daftar Pengujian | Kasus Uji | Hasil yang Diharapkan |
|----------------|-------------------------|--|--|
| Path | Validasi path | Request method GET pada endpoint /mahasiswa | Sukses mendapatkan list mahasiswa |
| | | Request method GET pada endpoint /mahasiswa/{id} | Sukses mendapatkan detail mahasiswa |
| | | Request method GET pada endpoint /mahasiswa/{wrong_id} | Error dengan status kode 404 - not found |
| | | Request method GET pada endpoint /mahasiswa/{id}/detail | Error dengan status kode 404 - not found |
| | | Request method GET pada endpoint /salah | Error dengan status kode 500 - internal server error |
| Request method | Validasi request method | Request method GET pada endpoint /mahasiswa | Sukses mendapatkan list mahasiswa |
| | | Request method GET pada endpoint /mahasiswa/{id} | Sukses mendapatkan detail mahasiswa |
| | | Request method POST pada endpoint /mahasiswa | Sukses menambahkan mahasiswa |
| | | Request method PUT pada endpoint /mahasiswa/{id} | Sukses mengubah data mahasiswa |
| | | Request method DELETE pada endpoint /mahasiswa/{id} | Sukses menghapus mahasiswa |
| Request body | Validasi request body | Request method POST pada endpoint /mahasiswa tanpa body | Error dengan status kode 400 - bad request |
| Security | Validasi Token | Request method GET pada endpoint /mahasiswa dengan invalid token | Error dengan status kode 401 - Unauthorized |

V. SIMPULAN DAN SARAN

5.1. Simpulan

Adapun kesimpulan dari penelitian yang telah dilakukan adalah sebagai berikut.

1. Telah berhasil dikembangkan *service* master pada aplikasi SiakadCloud 2.0 dengan menggunakan *framework* Lumen dan basis data PostgreSQL.
2. *Service* master merupakan sebuah *service* yang dapat mengelola data-data master dalam aplikasi. Data pada *service* master akan digunakan oleh *service* lainnya.
3. Terdapat 236 *endpoint* yang dapat digunakan dan setiap *endpoint* tersebut sudah ada di dalam dokumentasi API yang telah dibuat menggunakan Swagger.
4. Hasil pengujian dari unit *test* menunjukkan bahwa terdapat 518 *test case* yang sudah sesuai dan dari *performance test* menunjukkan bahwa terdapat 45548 *request* yang berhasil dari total 45589 *request* yang dikirim, di mana setiap *request* dapat mencapai paling maksimal 480 VUs dan test dijalankan dengan durasi waktu 10 menit.

5.2. Saran

Adapun saran dari penelitian yang telah dilakukan adalah sebagai berikut.

1. Proses *invalidate* token sebaiknya token disimpan ke dalam sebuah in *memory database* salah satunya yaitu Redis. Redis menyimpan semua datanya di memori utama *server*. Redis berbeda dengan *database* yang

menyimpan data di *hard disk*. Sehingga Redis dapat memberikan kinerja yang lebih baik.

2. Proses pengecekan JWT sebaiknya dilakukan pada *service* API Gateway. API Gateway meneruskan *request* yang dibutuhkan ke *service* yang dituju, sehingga pada setiap *service* tidak perlu melakukan pengecekan token lagi.
3. Proses validasi *request* sebaiknya dilakukan di bagian *controller* bukan di bagian *service*.

DAFTAR PUSTAKA

DAFTAR PUSTAKA

- Abdurahman, H., & Riswaya, A. R. (2014). Aplikasi Pinjaman Pembayaran Secara Kredit Pada Bank Yudha Bhakti STMIK Mardira Indonesia, Bandung. *Computech & Bisnis*, 8(2), 61–69.
- Cholifah, W. N., Yulianingsih, Y., & Sagita, S. M. (2018). Pengujian Black Box Testing pada Aplikasi Action & Strategy Berbasis Android dengan Teknologi Phonegap. *STRING (Satuan Tulisan Riset Dan Inovasi Teknologi)*, 3(2), 206.
- Fajar, M. (2015). Pembangunan Aplikasi Informasi, Pengaduan, Kritik, Dan Saran Seputar Kota Cimahi Pada Platform Android. *Jurnal Ilmiah Komputer Dan Informatika (KOMPUTA)*, 9.
- Febriyani, F., Pramukantoro, E. S., & Bachtiar, F. A. (2019). Perbandingan Kinerja Redis, Mosquitto, dan MongoDB sebagai Message Broker pada IoT Middleware | *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*. J-Ptiik.Ub.Ac.Id, 03(07), 6816–6823.
- Halili, F., & Ramadani, E. (2018). Web Services: A Comparison of Soap and Rest Services. *Modern Applied Science*, 12(3), 175.
- Harleni, H., & Marisa, M. (2018). Sistem Informasi Akademik (SIKAD) STIKES Perintis Padang. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 5(3), 305.
- Haviluddin, Haryono, A. T., & Rahmawati, D. (2016). *Aplikasi Program PHP dan MySQL*. Mulawarman University Press, 53(9), 1699.
- Munawar, G., & Hodijah, A. (2018). Analisis Model Arsitektur Microservice Pada Sistem Informasi DPLK. *Sinkron: Jurnal Dan Penelitian Teknik Informatika*, 3(1), 232–239.
- Munawaroh, S. (2005). Mengeksplorasi Database PostgreSQL dengan PgAdmin III. *Jurnal Teknologi Informasi DINAMIK*, X(2), 103–107.
- Perdana, M. A. K. (2018). Pengembangan REST API Layanan Penyimpanan menggunakan Metode Rapid Application Development (Studi kasus PT. XYZ). *InfoTekJar (Jurnal Nasional Informatika Dan Teknologi Jaringan)*, 3(1), 100–104.
- Ponce, F., Marquez, G., & Astudillo, H. (2019). *Migrating from monolithic architecture to microservices: A Rapid Review*. Proceedings - International Conference of the Chilean Computer Science Society, SCCS, 2019-Novem

(September). <https://doi.org/10.1109/SCCC49216.2019.8966423>

- Pranata, B. A., Hijriani, A., & Junaidi, A. (2018). Perancangan Application Programming Interface (API) Berbasis WEB Menggunakan Gaya Arsitektur Representational State Transfer (REST) Untuk Pengembangan Sistem Informasi Administrasi Pasien Klinik Perawatan Kulit. *Jurnal Komputasi*, 6(1), 33-42.
- Pressman, R., & Maxim, B. (2020). *Software Engineering: A Practitioner's Approach 9th Edition*. Kybernetes, 29(9/10), 8-9.
- Qibtiyah, U. M., & Rahayu, S. (2017). Implementasi JSON Web Service pada Aplikasi Digital Library Politeknik Sukabumi. *Jurnal Teknologi Rekayasa*, 2(1), 9.
- Rahmatulloh, A., Sulastri, H., & Nugroho, R. (2018). Keamanan RESTful Web Service Menggunakan JSON Web Token (JWT) HMAC SHA-512. *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi (JNTETI)*, 7(2).
- Rizkyana, M. A., Herdian, C. A., & R, A. A. Y. (2021). Implementasi Unit Testing Menggunakan Metode Test-First Development. *Jurnal Multimedia Networking Informatics*, 7(1), 37-47.
- Suendri. (2018). Implementasi Diagram UML (Unified Modeling Language) Pada Perancangan Sistem Informasi Remunerasi Dosen Dengan Database Oracle (Studi Kasus: UIN Sumatera Utara Medan). *Jurnal Ilmu Komputer Dan Informatika*, 3(1), 1-9.
- Surahman, F., Al Ikhsan, H., & Kusumah, F. (2018). *Rancang Bangun Web Service untuk Transaksi Data pada Aplikasi Sahabat Jasa dengan Metode REST*. Seminar Nasional Teknologi Informasi, 1, 256-264.
- Swara, G. Y., & Pebriadi, Y. (2016). Rekayasa Perangkat Lunak Pemesanan Tiket Bioskop Berbasis Web. *Jurnal Teknoif Teknik Informatika Institut Teknologi Padang (TEKNOIF)*, 4(2).
- Syaryadhi, M. (2018). Restful Web Service Untuk Pemantauan Dan Pengendalian Peternakan Ayam Broiler. *Jurnal Komputer, Informasi Teknologi, Dan Elektro*, 3(2), 53-59.
- Yanti, S. N., & Rihyanti, E. (2021). Penerapan Rest API untuk Sistem Informasi Film Secara Daring. *Jurnal Informatika Universitas Pamulang*, 6(1), 195.