



TUGAS AKHIR

**PENGEMBANGAN SISTEM
INFORMASI MANAJEMEN
TUGAS AKHIR DI PROGRAM
STUDI INFORMATIKA ITK
DENGAN ARSITEKTUR
*MICROSERVICE***

Ridwan Thamrin
NIM. 11181074

Syamsul Mujahidin, S.Kom., M.Eng
Muchammad Chandra Cahyo Utomo, M.Kom.

Program Studi Informatika
Jurusan Matematika dan Teknologi Informasi
Program Studi Informatika ITK
Balikpapan, 2022

LEMBAR PERSETUJUAN

Proposal Tugas Akhir dengan judul :

**“PENGEMBANGAN SISTEM INFORMASI MANAJEMEN TUGAS
AKHIR DI PROGRAM STUDI INFORMATIKA ITK DENGAN
ARSITEKTUR *MICROSERVICE*”**

Yang disusun oleh :

Ridwan Thamrin

NIM. 11171074

Telah diperiksa dan disetujui oleh dosen pembimbing:

Dosen Pembimbing Utama

Dosen Pembimbing Pendamping

Syamsul Mujahidin, S.Kom., M.Eng
NIP 199002182019031009

Muchammad Chandra Cahyo Utomo,
M.Kom.

NIP. 199205202019031013

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa atas berkat dan anugerah-Nya sehingga saya dapat menyelesaikan proposal tugas akhir yang berjudul :

“PENGEMBANGAN SISTEM INFORMASI MANAJEMEN TUGAS AKHIR DI PROGRAM STUDI INFORMATIKA ITK DENGAN ARSITEKTUR *MICROSERVICE*”

Proposal tugas akhir ini merupakan salah satu syarat yang harus ditempuh untuk menyelesaikan Program Sarjana di Program Studi Informatika Jurusan Matematika dan Teknologi Informasi, Program Studi Informatika ITK (ITK) Balikpapan. Untuk itu penulis mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Bapak Syamsul Mujahidin, S.Kom., M.Eng selaku Dosen Pembimbing Utama dan Bapak Muchammad Chandra Cahyo Utomo, M.Kom.selaku Dosen Pembimbing Pendamping.
2. Ibu Nisa Rizqiya Fadhliana, S.Kom., M.T, selaku Koordinator Tugas Akhir Program Studi Informatika Jurusan Matematika dan Teknologi Informasi ITK
3. Bapak dan Ibu Seluruh Dosen serta Tendik Informatika Jurusan Matematika dan Teknologi Informasi ITK.
4. Serta semua pihak yang terlibat dalam penyusunan proposal tugas akhir ini.

Penulis menyadari bahwa penyusunan proposal tugas akhir ini masih jauh dari sempurna, karena itu penulis mengharapkan segala kritik dan saran yang membangun. Semoga tugas akhir ini dapat bermanfaat bagi kita semua. Atas perhatiannya kami ucapkan terima kasih.

Balikpapan, 14 Februari 2022

Penyusun

PENGEMBANGAN SISTEM INFORMASI MANAJEMEN TUGAS AKHIR DI PROGRAM STUDI INFORMATIKA ITK DENGAN ARSITEKTUR MICROSERVICE

Nama Mahasiswa : Ridwan Thamrin
NIM : 11181074
Dosen Pembimbing Utama : Syamsul Mujahidin, S.Kom., M.Eng
Dosen Pembimbing Pendamping : Muchammad Chandra Cahyo Utomo, M.Kom.

ABSTRAK

Tugas akhir atau skripsi merupakan persyaratan utama seorang mahasiswa untuk mendapatkan status sarjana (S1) di setiap Perguruan Tinggi Negeri (PTN) maupun Perguruan Tinggi Swasta (PTS) di Indonesia. Dalam pengerjaan tugas akhir ini pihak kampus juga memiliki andil dalam penyelesaiannya. Pihak kampus perlu melakukan pengelolaan atau manajemen Tugas Akhir dengan baik dan benar untuk menghindari terjadinya permasalahan pada setiap tahapannya. Dengan penerapan sistem informasi tugas akhir diruang lingkup Institusi Perguruan Tinggi diharapkan mampu meningkatkan kinerja yang lebih efektif dan efisien untuk membantu proses menghasilkan lulusan yang baik. Berdasarkan permasalahan di atas maka diperlukannya Sistem Informasi Manajemen Tugas Akhir yang baik di suatu perguruan tinggi. Pada penelitian ini mengambil studi kasus pada Program Studi Informatika ITK, dikarenakan kurangnya sistem pengelolalan Tugas Akhir pada Program Studi tersebut. Banyaknya pelayanan yang disediakan pada sistem informasi ini, maka pemilihan Arsitektur *Microservice* tepat digunakan dalam pengerjaan sistem ini. Pola arsitektur *Microservice* adalah pengembangan sistem dari *Service Oriented Architecture* (SOA) dengan sekelompok komponen layanan independen (otonom) yang terbentuk dari suatu aplikasi. Pengembangan ini menggunakan pendekatan *Microservice* yang terdiri dari 8 tahapan, yaitu identifikasi masalah, studi literatur, *requirements*, *planning*, *iteration initialization*, *design*, *implementation*, *API Getaway*, *system testing*, *retrospective*, yang tentunya ditutup dengan pengujian tingkat kelayakan sistem. *API Gateway* ini akan menghasilkan beberapa *service* dalam pembuatan sistem informasi manajemen Tugas Akhir ini nantinya. Diharapkan penelitian ini dapat mempermudah manajemen Tugas Akhir di Program Studi Informatika ITK dengan mengembangkan Sistem Informasi Manajemen Tugas Akhir menggunakan arsitektur *Microservice*.

Kata kunci: *Microservice, Sistem Informasi Manajemen Tugas Akhir, Tugas Akhir*

DAFTAR ISI

Kata Pengantar	i
Abstrak	i
Daftar Isi.....	ii
Daftar Gambar.....	iv
Daftar Tabel	v
BAB 1 PENDAHULUAN	1
1.1 Latar belakang	1
1.2 Perumusan Masalah.....	4
1.3 Tujuan Penelitian.....	4
1.4 Batasan Penelitian	5
1.5 Manfaat Penelitian.....	5
1.6 Kerangka Pemikiran	5
BAB 2 TINJAUAN PUSTAKA.....	8
2.1 Tugas Akhir.....	8
2.2 Sistem Informasi Tugas Akhir	8
2.3 Program Studi Informatika ITK	9
2.4 Arsitektur <i>Microservice</i>	11
2.5 <i>Web Service</i>	13
2.6 Laravel	13
2.7 <i>REST API</i>	14
2.8 <i>Microsoft SQL Server</i>	15
2.9 <i>Testing Jmeter Apache</i>	15
2.10 JSON	16
2.11 Penelitian Terdahulu	17
BAB 3 METODE PENELITIAN	22
3.1 Garis Besar Penelitian	22
3.2 Diagram Alir Penelitian	22
3.3 Prosedur Penelitian.....	23
3.3.1 Identifikasi Masalah	24
3.3.2 Studi Literatur.....	24

3.3.3 <i>Requirements</i>	24
3.3.4 <i>Planning</i>	28
3.3.5 <i>Iteration Initialization</i>	31
3.3.6 <i>Design</i>	31
3.3.7 <i>Implementation</i>	39
3.3.8 <i>System Testing</i>	39
3.3.9 <i>Retrospective</i>	40
3.4 Rencana Jadwal Penelitian	40
Daftar Pustaka	39
Lampiran A	39

DAFTAR GAMBAR

Gambar 1. 1 Kerangka Pemikiran	6
Gambar 2. 1 <i>Arsitektur Microservice</i>	12
Gambar 2. 2 Tampilan awal <i>Jmeter Apache</i>	16
Gambar 3. 1 Diagram Alir Pengembangan Sistem Manajemen Tugas Akhir	23
Gambar 3. 2 <i>Use Case</i> SIM-TA Informatika ITK	29
Gambar 3. 3 <i>Database</i> SIM TA Informatika ITK.....	38

DAFTAR TABEL

Tabel 2. 1 Penelitian Terdahulu	18
Tabel 3. 1 Rangkuman <i>User Story</i>	25
Tabel 3. 2 Daftar Iterasi	30
Tabel 3. 3 Deskripsi Format API Bimbingan	32
Tabel 3. 4 Deskripsi <i>API Service</i> Bimbingan.....	33
Tabel 3. 5 Deskripsi Format API Sempro	33
Tabel 3. 6 Deskripsi <i>API Service</i> Sempro.....	33
Tabel 3. 7 Deskripsi Format API Jadwal Sempro.....	34
Tabel 3. 8 Deskripsi <i>API Service</i> Jadwal Sempro.....	34
Tabel 3. 9 Deskripsi Format API SemHas	34
Tabel 3. 10 Deskripsi <i>API Service</i> SemHas	35
Tabel 3. 11 Deskripsi Format API Jadwal SemHas	35
Tabel 3. 12 Deskripsi <i>API Service</i> Jadwal SemHas.....	36
Tabel 3. 13 Deskripsi Format API Proposal	36
Tabel 3. 14 Deskripsi <i>API Service</i> Proposal	36
Tabel 3. 15 Deskripsi Format API Download Proposal.....	37
Tabel 3. 16 Deskripsi <i>API Service</i> Proposal	37
Tabel 3. 17 Rencana Jadwal Penelitian.....	41

BAB 1

PENDAHULUAN

Pada bab pendahuluan, akan dijelaskan terkait latar belakang penelitian ini dilakukan. Secara garis besar, penelitian ini akan membuat Pengembangan Sistem Informasi Manajemen Tugas Akhir dengan metode *Microservice* di Program Studi Informatika ITK. Setelah itu, akan dijelaskan juga terkait rumusan masalah, tujuan dan manfaat dari penelitian ini. Permasalahan yang ada pada Program Studi Informatika ITK akan disusun ke dalam kerangka pemikiran penelitian yang juga disajikan pada bab ini.

1.1 Latar belakang

Pada umumnya seorang mahasiswa pada perguruan tinggi dituntut untuk segera mungkin untuk menyelesaikan masa studinya. Fase akhir masa studi sebagai seorang mahasiswa biasanya diberi tugas akhir yang biasa disebut skripsi. Tugas akhir atau skripsi merupakan persyaratan utama seorang mahasiswa untuk mendapatkan status sarjana (S1) di setiap Perguruan Tinggi Negeri (PTN) maupun Perguruan Tinggi Swasta di Indonesia. Yang di mana skripsi berisi sebuah karangan ilmiah yang wajib ditulis oleh seorang mahasiswa tingkat akhir untuk menyelesaikan masa studinya dan memperoleh gelar kesarjanaan (Seto, Wondo and Mei, 2020).

Selain daripada itu, skripsi juga menjadi salah satu bentuk pembuktian kemampuan akademik mahasiswa bersangkutan dalam penelitian dengan topik yang sesuai dengan bidang studinya. Atau bisa dikatakan bahwa skripsi adalah suatu karya ilmiah yang disusun oleh seorang mahasiswa berdasarkan hasil penelitian dengan menggunakan data primer atau data sekunder yang penulisannya terikat pada metodologi ilmiah yang benar (Z.M, 2012). Proses penyusunan skripsi pada hakikatnya dilakukan secara individual oleh setiap mahasiswa dengan memanfaatkan semua pengetahuan dan keterampilan yang pernah diperoleh sebelumnya untuk diterapkan dalam menggali

permasalahan yang ada. Dalam penyusunan skripsi dibutuhkan sumber yang cukup banyak dan tentunya mahasiswa tersebut dapat mengetahui sesuatu dan menambah pengetahuan tentang apa yang akan mereka kerjakan (Simatupang and Muhammad, 2019).

Dalam pengerjaan tugas akhir ini pihak kampus juga memiliki andil dalam penyelesaiannya. Tahapan tugas akhir atau skripsi berbeda di setiap jenjang pendidikan dan perguruan tinggi, namun pada umumnya proses penyusunan tugas akhir dan skripsi ini melalui beberapa tahapan seperti pendaftaran peserta skripsi sebagai mahasiswa yang mengambil mata kuliah tugas akhir atau skripsi, pengajuan proposal skripsi beserta judul, penelitian, seminar proposal skripsi, dan sidang skripsi (Suwita, 2020) . Pihak kampus perlu melakukan pengelolaan atau manajemen Tugas Akhir dengan baik dan benar untuk menghindari terjadinya permasalahan pada setiap tahapannya. Dengan penerapan sistem informasi tugas akhir diruang lingkup Institusi Perguruan Tinggi diharapkan mampu meningkatkan kinerja yang lebih efektif dan efisien untuk membantu proses menghasilkan lulusan yang baik(Putra, Sitanggang and Istiadi, 2019).

Pemakaian teknologi komputer dalam segala bidang kehidupan sehari-hari tidak akan dapat dihindari. Selain memberikan kemudahan, juga dapat mengerjakan pekerjaan dengan cepat, tepat dan akurat. Bahkan, penggunaan teknologi komputer tersebut akan menjadi syarat utama untuk menunjukkan kualitas sesuatu bidang maupun instansi dan menjadi modal terpenting dalam memenangkan persaingan. Pada sebuah instansi pendidikan tentunya sangat membutuhkan sistem informasi dalam mengatur kegiatan akademik yang cepat, efektif, efisien, akurat untuk meningkatkan standar mutu pendidikan pada instansi pendidikan tersebut(Wibawa and F., 2017).

Program Studi Informatika ITK (ITK) termasuk ke dalam Perguruan Tinggi Negeri yang telah diresmikan oleh Presiden Republik Indonesia Susilo Bambang Yudhoyono pada tanggal 6 Oktober 2014 sebagai salah satu perguruan tinggi negeri baru di Indonesia yang berdiri di Kalimantan Timur. ITK saat ini masih ke dalam proses perguruan tinggi yang sedang berkembang (Lembaga Penelitian dan Pengabdian Kepada Masyarakat,

2019). Dalam perkembangannya ITK banyak mengotomatisasi setiap proses yang menyangkut kegiatan perkuliahan. Seperti halnya pembuatan LMS yang saat ini sudah bisa digunakan setiap mahasiswa untuk mengontrol kegiatan perkuliahan setiap harinya. Namun untuk pengelolaan Tugas Akhir atau Skripsi masih dilakukan dengan cara konvensional. Seperti halnya proses pemilihan dosen pembimbing yang belum jelas tahapannya. Para mahasiswa sering kali merasa kebingungan pada tahapan tersebut. Penyampaian informasi yang masih manual menjadi salah satu permasalahan juga. Selain itu, semua tahapan tersebut pencatatannya maupun pendataannya masih dilakukan secara manual.

Dengan beberapa permasalahan tersebut maka perlu dilakukan penelitian untuk merancang dan membangun manajemen sistem informasi manajemen Tugas Akhir yang dapat membantu mahasiswa dalam melakukan proses pengelolaan Tugas Akhir yang lebih mudah dan efisien. Dalam hal ini teknologi *web application* akan dipilih karena bisa digunakan pada berbagai macam *platform (cross platform)* yang terhubung oleh internet. Kemudian dalam proses pengembangan teknologi *web application* tadi akan menggunakan pendekatan *Microservice*. *Microservice* memberikan kebebasan bagi pengembang untuk mengembangkan *software* dalam waktu yang cepat. Perkembangan arsitektur *Microservice* juga didukung oleh kurang handalnya arsitektur *monolith* dalam menangani *system failure*. Dikarenakan setiap dalam satu aplikasi memiliki satu *codebase* yang sama, akan menjadi suatu hal yang pasti bahwa arsitektur *monolith* juga menjadi *single point of failure* yang akan terjadi jika salah satu layanan di *codebase* yang sama mengalami malfungsi atau *error*. Berdasarkan penelitian Yuri Chandra yang berjudul “Implementasi Arsitektur *Microservice* Pada Aplikasi Web Pengajaran Agama Islam *Home Pesantren*” menghasilkan Penerapan komunikasi antar *Microservice* sehingga dapat menjadi sebuah kesatuan adalah dengan menerapkan dua teknik komunikasi yang berbeda, yaitu *REST API* untuk mendukung *synchronous communication* dan RabbitMQ untuk mendukung *asynchronous communication* (Yuri Chandra Tri Putra, Thomas Adi Purnomo Sidi and Joseph Eric Samodra, 2020). Begitu juga dengan

penelitian yang dilakukan oleh Muhhamad Fathan yang berjudul “Analisis dan Desain Arsitektur *Microservice* dengan *GraphQL* Sebagai *API Gateway* untuk Sistem Informasi Akademik AIS UIN Jakarta” yang menyatakan bahwa penerapan arsitektur *Microservice* pada sistem AIS tersebut dapat meningkatkan kinerja aplikasi dan memudahkan jika adanya pengembangan pada aplikasi tersebut (Radhiyan, 2020). Oleh karena itu, penelitian pengembangan sistem informasi manajemen Tugas Akhir dengan Metode *Microservice* pada studi kasus program studi Informatika ITK perlu dilakukan untuk dapat mempercepat dan mempermudah setiap aspek dalam pengelolaan Tugas Akhir.

1.2 Perumusan Masalah

Berdasarkan permasalahan yang diangkat dalam latar belakang terkait sistem informasi manajemen pengelolaan tugas akhir yang dirumuskan ke dalam pertanyaan yaitu

1. Bagaimana membangun Sistem Informasi Manajemen Tugas Akhir di Program Studi Informatika ITK dengan menggunakan pendekatan *Microservice*.
2. Bagaimana pengujian hasil Sistem Informasi Manajemen Tugas Akhir yang dikembangkan dengan pendekatan *Microservice* untuk menguji tingkat kelayakan sistem.

1.3 Tujuan Penelitian

Penelitian ini dilakukan atas dasar tujuan yang ingin dicapai, yaitu Untuk mempermudah manajemen Tugas Akhir di Program Studi Informatika ITK dengan mengembangkan Sistem Informasi Manajemen Tugas Akhir menggunakan pendekatan *Microservice*.

1.4 Batasan Penelitian

Batasan dari penelitian pada rancang bangun manajemen Tugas Akhir yaitu sebagai berikut :

1. Sistem Informasi Manajemen Tugas Akhir berbasis *web*, dikembangkan menggunakan bahasa pemrograman PHP dengan bantuan *framework* Laravel.
2. Pengembangan Sistem Informasi Manajemen Tugas Akhir berbasis *web* menggunakan basis data dari *SQL Server*.
3. Pengembangan Sistem Informasi Manajemen Tugas Akhir berbasis *web* dilakukan dengan pendekatan *Microservice*.
4. *Stakeholder* Sistem Manajemen Tugas Akhir adalah Program Studi Informatika ITK.
5. Pada penelitian ini proses pengujian menggunakan *Jmeter Apache testing* untuk pengujian fitur.

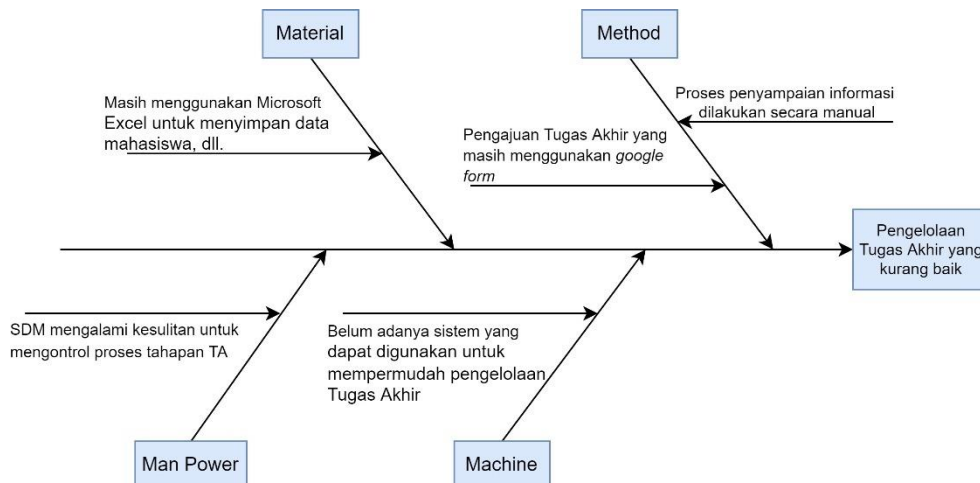
1.5 Manfaat Penelitian

Manfaat yang didapatkan dari penelitian pada manajemen Tugas Akhir yaitu sebagai berikut :

1. Bagi mahasiswa ITK, memberikan kemudahan untuk mendapatkan pelayanan terkait dengan pemilihan dosen pembimbing.
2. Bagi unit tenaga Pendidikan (tendik) jurusan, memberikan kemudahan dalam proses pengajuan Tugas Akhir
3. Membantu meningkatkan efektivitas dan efisiensi manajemen pengelolaan Tugas Akhir di ITK

1.6 Kerangka Pemikiran

Kerangka pemikiran pada penelitian ini dimodelkan menggunakan diagram sebab akibat atau *fishbone diagram* yang menjelaskan alur berpikir dalam pengerjaan penelitian berdasarkan masalah hingga penyelesaian yang diusulkan.



Gambar 1. 1 Kerangka Pemikiran

Pada Gambar 1.1 terdapat diagram yang menjelaskan terkait beberapa hal yang menjadi kategori permasalahan. Kategori permasalahan tersebut yang pada akhirnya menjadi latar belakang dari dilakukannya penelitian. Dalam diagram tersebut terdapat beberapa kategori permasalahan yaitu *Manpower*, *Method*, *Machine* dan *Material* .

Kategori *Man Power* berhubungan dengan manusia di mana dalam kategori ini permasalahan yang terjadi yaitu keterbatasan sumber daya manusia yang ada melingkupi hal seperti *human error* pada unit, petugas ataupun mahasiswa saat melakukan proses yang berhubungan dengan mengelola data Tugas Akhir. *Human error* juga berhubungan dengan penyampaian informasi terkait Tugas Akhir yang kurang baik. Disisi lain sebagai Perguruan Tinggi Negeri baru ITK memiliki potensi untuk terus berkembang salah satunya dari sisi jumlah mahasiswa yang akan terus bertambah. Jika *procedure operasional* layanan pengelolaan Tugas Akhir untuk mahasiswa ini tetap secara manual, maka hal ini akan menjadi permasalahan yang menyulitkan.

Kategori *Method* menjelaskan tentang permasalahan proses atau metode. Pada proses layanan baik itu proses layanan pembuatan surat yang diajukan oleh mahasiswa ditingkat akademik ataupun jurusan masih secara

manual di mana mahasiswa datang ke akademik ataupun jurusan kemudian menulis data. Proses pengajuan TA yang ada di lingkup kampus Program Studi Informatika ITK, belum terintegrasi dengan suatu sistem atau *software* khusus semuanya dikerjakan secara manual tanpa adanya *template* dari sistem sehingga proses pengerjaannya cukup memakan waktu. Pendistribusian informasi terkait Tugas akhir juga masih menggunakan *e-mail* ataupun surat fisik. Oleh sebab itu untuk pengelolaan seperti pengajuan TA maupun penyampaian informasi membutuhkan waktu yang cukup lama. Dari metode dan proses yang berjalan dapat dikatakan tidak efektif dan efisien.

Setelah itu kategori *Material* yang didapatkan berdasarkan material atau barang yang digunakan. Dalam hal ini buku digunakan sebagai media penyimpanan dan pencatatan data terhadap pengajuan pembuatan surat oleh mahasiswa. Buku ini sewaktu – waktu bisa saja hilang, rusak ataupun penuh akan data. Sehingga merepotkan mahasiswa untuk melakukan pengajuan pembuatan surat dan petugas yang akan melakukan pembuatan surat, karena data – data yang diperlukan berada di dalam buku tersebut .

Selanjutnya Kategori *Machine*, menjelaskan tentang permasalahan teknologi. Masalah yang ada dalam faktor teknologi yaitu tidak adanya sistem atau *software* manajemen yang mengintegrasikan seluruh kegiatan pengelolaan Tugas Akhir. Selama ini teknologi yang digunakan adalah teknologi konvensional seperti *email*, *microsoft office*, *google drive* dan *google form*. Cara ini dinilai kurang efektif dan efisien.

Dari penjelasan kategori permasalahan yang ada di atas, maka dilakukanlah penelitian ini dengan harapan dapat menyelesaikan masalah utama yaitu belum adanya Sistem Informasi Manajemen Tugas Akhir. Dari penelitian ini nantinya memiliki target menghasilkan Sistem Informasi manajemen Tugas Akhir berbasis *web* pada ITK. Sistem tersebut nantinya untuk mendukung proses pengelolaan Tugas Akhir.

BAB 2

TINJAUAN PUSTAKA

Pada bab ini akan menjelaskan terkait tinjauan pustaka dan studi literatur yang digunakan pada penelitian mengenai rancang bangun manajemen Tugas Akhir berbasis *web* dengan metode *Microservice* di Program Studi Informatika ITK. Tujuannya untuk mempelajari literatur yang berkaitan dengan penelitian. Teori yang dibahas antara lain Tugas Akhir, Sistem Informasi Tugas Akhir, Program Studi Informatika ITK, Metode *Microservice*, *Web Service*, *Laravel*, *REST API*, *Microsoft SQL Server*, *Testing Jmeter Apache*, dan beberapa penelitian terdahulu.

2.1 Tugas Akhir

Tugas Akhir atau biasa dikatakan skripsi merupakan salah satu syarat kelulusan yang ditetapkan di masing-masing Program Studi/Fakultas ataupun Universitas, yang merupakan tuntutan dari sebuah formulasi kurikulum yang diberlakukan. Skripsi juga menjadi satu syarat yang harus dipenuhi oleh calon sarjana untuk bisa mendapatkan gelar sarjana. Oleh karena itu, mahasiswa harus menyusun target-target tertentu untuk bisa mempertanggungjawabkan tugas akhirnya tersebut di hadapan dosen pembimbing dan dosen penguji (Machmud, 2016). Skripsi merupakan suatu karangan ilmiah yang wajib ditulis oleh seorang mahasiswa tingkat akhir sebagai persyaratan untuk menyelesaikan masa studinya. Skripsi juga menjadi salah satu bukti kemampuan akademik mahasiswa, selain sebagai persyaratan akhir pendidikan skripsi juga menjadi salah satu syarat untuk mendapatkan gelar sarjana (Seto, Wondo and Mei, 2020).

2.2 Sistem Informasi Tugas Akhir

Sistem merupakan himpunan dari sesuatu benda nyata atau abstrak (*a set of things*) yang terdiri dari bagian-bagian atau komponen-komponen yang saling berkaitan, berhubungan, yang tentunya saling mendukung, yang secara keseluruhan bersatu dalam satu kesatuan untuk mencapai sebuah tujuan tertentu secara efisien dan efektif. Untuk pengertian Informasi sendiri merupakan sebuah data yang telah

diolah sesuai dengan keperluan tertentu (Suwita, 2020). Maka dari itu dapat disimpulkan bahwa sistem informasi merupakan suatu himpunan maupun komponen-komponen yang di dalamnya berupa data yang sudah diolah menjadi suatu informasi yang terintegrasi dengan tujuan dibuatnya sistem tersebut (Wibawa and F., 2017). Dengan adanya sistem informasi yang dapat manajemen Tugas Akhir dalam suatu instansi Pendidikan perguruan tinggi akan membuat *monitoring* pelaksanaan dan pengelolaan tugas akhir dapat dilakukan dengan cepat, tepat dan akurat, serta mampu meningkatkan efisiensi dan efektivitas kerja pihak-pihak yang terkait. Selain itu juga dapat mengurangi penggunaan kertas, sehingga perguruan tinggi dapat menekan biaya pengadaan kertas (Simatupang and Muhammad, 2019).

2.3 Program Studi Informatika ITK

Program Studi ITK merupakan salah satu perguruan tinggi yang berada di Kalimantan Timur, tepatnya berada di Jalan Soekarno Hatta KM.15, Kota Balikpapan. Program Studi Informatika ITK atau yang biasa disingkat dengan ITK adalah perguruan tinggi negeri yang berdiri sejak tahun 2012 dan disahkan melalui Peraturan Presiden No. 125 Tahun 2014. Sasaran utama ITK saat ini adalah pemenuhan pelayanan standar minimum perguruan tinggi sesuai dengan Tri Dharma Perguruan Tinggi yaitu pendidikan, penelitian dan pengabdian masyarakat (Lembaga Penelitian dan Pengabdian Kepada Masyarakat, 2019). Salah satu Program Studi yang ada di ITK ialah Program Studi Informatika. Program Studi Informatika berdiri pada tahun 2016 bersamaan dengan Program Studi Teknik Industri dan Teknik Lingkungan. Kehadiran program studi Informatika di Institut Teknologi Kalimantan diharapkan dapat berpartisipasi aktif dalam mendukung

Buku Putih Indonesia 2005-2025 di bidang Teknologi Informasi dan Komunikasi. Adapun visi, misi dan tujuan ITK adalah sebagai berikut:

A. Visi

Rumusan visi institusi Program Studi Informatika adalah “*Menjadi program studi unggul di bidang Informatika yang inovatif dan kreatif dalam poros Kalimantan pada tahun 2025*” (website if.itk.ac.id, 2019)

B. Misi

Sebagai upaya dalam mewujudkan visi tersebut, maka misi Program Studi Informatika ITK adalah (website if.itk.ac.id, 2019) :

1. Menyelenggarakan sistem pendidikan yang efektif, efisien, dan berkelanjutan dalam rangka menghasilkan lulusan sarjana Informatika
2. Menghasilkan lulusan yang memiliki kompetensi di bidang Informatika, berjiwa wirausaha (*entrepreneur*) dan dapat berperan positif di tingkat nasional dan internasional (*world class*)
3. Meningkatkan kontribusi dan kolaborasi dengan berbagai pihak dalam masyarakat dengan mengembangkan produk dan layanan dalam bidang Informatika di tingkat regional, nasional maupun internasional

C. Tujuan

Dari keseluruhan visi dan misi program studi Informatika ITK tersebut, maka didapatkan beberapa tujuan yang ingin dicapai sebagai berikut:

1. Menghasilkan lulusan yang memiliki kompetensi di bidang Informatika, berjiwa wirausaha (*entrepreneur*) dan dapat dipercaya sehingga mampu bekerja sama dan memberikan kontribusi di tingkat nasional dan internasional (*world class*), melalui kurikulum yang disusun dengan

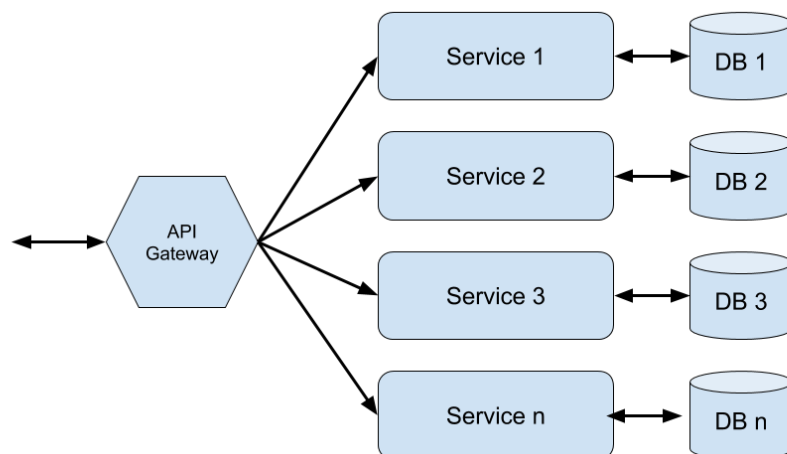
mempertimbangkan model kurikulum Informatika pada tingkat nasional dan internasional.

2. Menjalankan sistem pendidikan dengan penjaminan mutu sesuai standar nasional dan internasional.
3. Melibatkan civitas *academica* Prodi Informatika dalam penelitian yang dapat memperkaya keilmuan di bidang komputasi dalam rangka mengisi dan menunjang pembangunan regional maupun nasional.
4. Melibatkan civitas *academica* Prodi Informatika dalam pengabdian masyarakat dalam bentuk pembinaan, bimbingan dan konsultasi dalam rangka meningkatkan peran serta masyarakat dalam pembangunan potensi.
5. Meningkatkan kontribusi dan kolaborasi dengan berbagai pihak dalam masyarakat dengan mengembangkan produk dan layanan hasil inovasi dan kreasi dalam bidang Informatika di tingkat regional, nasional maupun internasional.
6. Mengembangkan sertifikasi kompetensi di bidang Informatika di tingkat regional, nasional maupun internasional.

2.4 Arsitektur *Microservice*

Pendekatan *Microservice* merupakan salah satu pola arsitektur yang muncul saat ini, yang dibangun di sekitar konsep inti berikut: unit yang dikerahkan secara terpisah, komponen layanan, dan memiliki sistem yang bersifat terdistribusi. Pola arsitektur *Microservice* adalah pengembangan sistem dari *Service Oriented Architecture* (SOA) dengan sekelompok komponen layanan independen (otonom) yang terbentuk dari suatu aplikasi. Keuntungan terbesar dari penerapan pola arsitektur ini adalah komponen yang mudah diterapkan dan diuji serta kemampuan untuk menskalakan secara horizontal. Dapat dimisalkan jika kita mengumpulkan data dari sumber independen dan memprosesnya (atau setidaknya menggabungkannya) adalah salah satu pilihan yang baik adalah menerapkan pendekatan ini yang bertujuan untuk pemrosesan data waktu nyata dan memvisualisasikan data tersebut (Damyanov, 2019).

Microservice juga merupakan tren baru yang meningkat pesat, karena meningkatkan fleksibilitas untuk menggabungkan teknologi yang berbeda, mengurangi kompleksitas dengan menggunakan layanan yang ringan dan modular, dan meningkatkan skalabilitas dan ketahanan sistem secara keseluruhan. Dalam definisi gaya arsitektur *Microservice* adalah pendekatan untuk mengembangkan aplikasi tunggal sebagai *suite* layanan kecil, masing-masing berjalan dalam prosesnya sendiri dan berkomunikasi dengan mekanisme ringan (misalnya, *HTTP* resource API). Aplikasi kemudian tersusun dari sejumlah layanan (*service-based application*) yang bekerja secara kohesif untuk menyediakan fungsionalitas yang kompleks. Karena keunggulan arsitektur *Microservice*, banyak pengembang berniat untuk mengubah aplikasi monolitik tradisional menjadi *service-based application* (Hu, de Laat and Zhao, 2019) Penggunaan *Microservice* juga didasarkan pada pemenuhan kebutuhan beberapa sistem yang memiliki *server-side*, yang dimana memungkinkan dimiliki oleh beberapa *user* dan mungkin perlu mengekspos API untuk pengembangan sistem tersebut. *Server-side* disini dimaksud jika pengembangan sistem lanjutannya bisa menggunakan API yang sudah dibuat pada sistem sebelumnya (Ortiz *et al.*, 2019).



Gambar 2. 1 Arsitektur *Microservice*

2.5 Web Service

Web service merupakan sebuah entitas komputasi yang dapat diakses melalui jaringan internet maupun intranet melalui standar protokol tertentu dalam platform dan antarmuka bahasa pemrograman yang independen. Yang di mana biasanya digunakan sebagai suatu fasilitas yang disediakan oleh suatu *website* untuk menyediakan layanan (dalam bentuk informasi) kepada sistem lain, sehingga sistem lain dapat berinteraksi dengan sistem tersebut melalui layanan-layanan (*service*) yang disediakan oleh suatu sistem yang menyediakan *web service* (Maulidiansyah, Rakhman and Ramdhani, 2017). *Web Service* adalah aplikasi komputasi terdistribusi melalui Internet yang dapat diakses melalui satu set XML homogen antarmuka. W3C (*World Wide Web Consortium*) mendefinisikan layanan Web sebagai "perangkat lunak" sistem yang dirancang untuk mendukung interoperabilitas interaksi mesin-ke-mesin melalui jaringan. Sebuah *Web Service* terdiri dari satu set komputasi atau aktivitas fisik dengan sejumlah sumber daya untuk memenuhi kebutuhan penggunanya (Tun and Onn, 2020). Atau dapat diartikan juga sebagai kumpulan protokol komunikasi korespondensi yang berkembang dan ditetapkan yang terdiri dari *Extensible Markup Language* (XML), *Simple Object Application Protocol* (SOAP), *Universal Description Discovery and Integration* (UDDI) dan *Web Services Description Language* (WSDL) melalui *Hypertext Transfer Protokol* (HTTP) (Hammoudeh and Al-Ajlan, 2020).

2.6 Laravel

Laravel adalah salah satu *framework* PHP yang paling dapat digunakan untuk programmer pemula dan *amateur*. Ini dapat mengurangi waktu pengembangan aplikasi web dan pemasaran dengan metode PHP berorientasi objek modern. Sintaks ekspresif dan fungsi modernnya menarik bagi pengembang yang ingin membuat aplikasi yang kuat. Menggunakan *framework* yang dapat memudahkan proses pengembangan karena menyediakan beberapa modul dengan koneksi yang bisa dikerjakan bersamaan. Laravel menyediakan fitur-fitur canggih seperti lapisan abstrak *database* ekspresif dan injeksi ketergantungan (Subecz, 2021) .

Framework Laravel adalah *framework* CLI (*Command-Line Interface*), migrasi dan artisan CLI yang menawarkan seperangkat alat dan arsitektur aplikasi yang menggabungkan banyak fitur terbaik dari *framework* seperti *Codeigniter*, *Yii*, ASP.NET MVC, dll. Laravel juga menganut konsep MVC, yang di mana berarti membuat program terstruktur dengan memisahkan logika menjadi 3 bagian, yaitu model, *view* dan *controller* (Santoso, Sinaga and Zuhdi, 2021). Atau biasa dikatakan bahwa Laravel merupakan pengembangan *website* berbasis MVP yang ditulis dengan bahasa PHP yang dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal, biaya pemeliharaan, dan untuk meningkatkan pengalaman bekerja dengan aplikasi dengan menyediakan sintaks yang ekspresif, jelas dan menghemat waktu pengerjaan juga tentunya (Hermanto, Yusman and Nagara, 2019). Namun untuk struktur pola MVC pada laravel sedikit berbeda pada struktur pola MVC pada umumnya. Pada Laravel terdapat *routing* yang memberikan koneksi antara *request* dari *user* dan *controller*. Jadi *controller* tidak langsung menerima *request* tersebut (Purnama Sari and Wijanarko, 2020).

2.7 REST API

REST API merupakan sebuah implementasi dari *API* (*Application Programming Interface*). *REST* (*Representational State Transfer*) adalah suatu arsitektur metode komunikasi yang menggunakan protokol *HTTP* untuk pertukaran data pada sistemnya. *REST API* diakses menggunakan protokol *HTTP* (*Hypertext Transfer Protocol*), Dikarenakan hal tersebut, maka diperlukan penamaan dan struktur URL yang baik dan mudah dimengerti dalam penggunaannya. *Endpoint* dalam pemanggilannya yang biasa dikenal sebagai URL API. Contoh pemanggilan URL yang baik adalah seperti berikut: *users*, *users/123*, *users/123/photos* dan seterusnya (Perdana, 2018). Gaya arsitektur *REST* semakin populer tetapi ada banyak perdebatan dan kekhawatiran yang berkembang tentang pemodelan layanan *web REST* (*REST API*). *REST API* menjelaskan satu set sumber daya, dan satu set operasi yang dapat dipanggil pada sumber daya tersebut. Operasi di *REST API* dapat dipanggil dari klien *HTTP* apa pun, termasuk kode *JavaScript* sisi klien yang berjalan di browser web. Klien *HTTP* menggunakan jalur relatif terhadap jalur

dasar yang mengidentifikasi sumber daya di *REST API* yang diakses klien. Jalur ke sumber daya dapat bersifat hierarkis, dan struktur jalur yang dirancang dengan baik dapat membantu konsumen *REST API* memahami sumber daya yang tersedia di dalam *REST API* tersebut. Setiap sumber daya di *REST API* memiliki serangkaian operasi yang dapat dipanggil oleh klien *HTTP*. Operasi di *REST API* memiliki nama dan metode *HTTP* (seperti *GET*, *POST*, atau *DELETE*). Nama operasi harus bersifat *unique* di semua sumber daya di *REST API* tersebut. Selain itu, satu sumber daya hanya dapat memiliki satu operasi yang ditentukan untuk metode *HTTP* tertentu (Surwase, 2016).

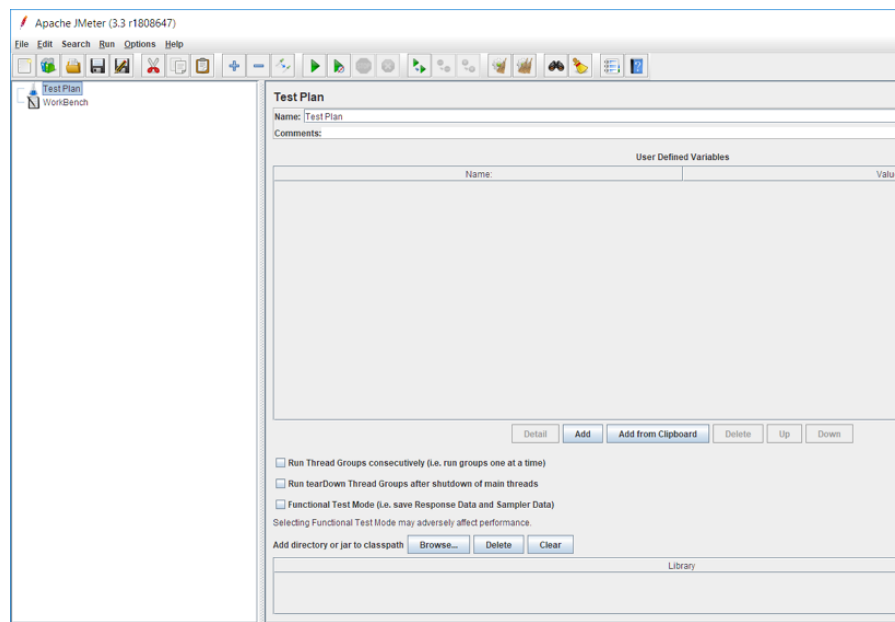
2.8 Microsoft SQL Server

Microsoft SQL Server merupakan suatu sistem manajemen basis data relasional, memiliki MS-SQL dan *Transact-SQL* sebagai bahasa pemrograman terstruktur utamanya. MS-SQL mengandalkan aljabar relasional yang terutama digunakan untuk penyisipan data, modifikasi, penghapusan dan pengambilan, serta untuk pengendalian akses data. *Query* yang paling sering digunakan adalah kueri pengambilan data melalui perintah *SELECT* (Luca and Ā, 2014). Untuk bahasa *query* yang digunakan pada MS-SQL Server merupakan *Transact-SQL* yang berasal dari implementasi SQL standar ANSI/ISO. MS-SQL Server juga sering digunakan pada skala bisnis kecil sampai menengah, tetapi lambat laun berkembang juga digunakan pada skala besar (Izzaty, Astuti and Cholimah, 1967).

2.9 Testing Jmeter Apache

JMeter adalah aplikasi desktop, yang dirancang untuk menguji dan mengukur kinerja dan perilaku fungsional aplikasi klien/server, seperti aplikasi *web* atau aplikasi FTP. Sejauh ini, ini adalah salah satu aplikasi pengujian *open-source* yang paling banyak digunakan dan didistribusikan secara bebas yang dapat ditawarkan oleh *Net*. Ini murni berbasis *Java* dan sangat dapat dikembangkan melalui API (*Application Programming Interface*) yang disediakan. *JMeter* bekerja dengan bertindak sebagai "sisi klien" dari aplikasi "klien/server". Ini mengukur waktu respons dan semua sumber daya server lainnya seperti beban CPU, penggunaan memori, dan penggunaan sumber daya. Dalam hal ini, *JMeter* dapat digunakan

secara efektif untuk otomatisasi pengujian fungsional (Halili, 2008). Digunakan untuk menguji kinerja baik pada sumber daya statis dan dinamis (*web services*)(SOAP / REST), Web bahasa dinamis - PHP, *Java*, ASP.NET, File, dll -, *Java Objects*, Pangkalan Data dan Pertanyaan, FTP Server dll). Hal ini dapat digunakan untuk mengimplementasikan beban berat pada server, sekelompok server, jaringan atau objek untuk menguji kekuatan atau untuk menganalisis kinerja secara keseluruhan di bawah jenis beban yang berbeda (Permatasari, 2020).



Gambar 2. 2 Tampilan awal *Jmeter Apache*

2.10 JSON

JSON (*JavaScript Object Notation*) adalah format data semi terstruktur ringan yang populer berdasarkan tipe data bahasa pemrograman *JavaScript*. Ini telah menjadi format pertukaran data utama di *World Wide Web* dalam beberapa tahun terakhir. JSON juga mendapatkan popularitas dalam penelitian komunitas basis data. Sebagai format data semi terstruktur, JSON tidak hanya dapat diintegrasikan dalam sistem database tradisional, tetapi juga banyak digunakan dalam sistem *database* NoSQL. Model data menggambarkan struktur data dasar dan semantik dari data yang mendasarinya, sehingga merupakan aspek fundamental dan kunci untuk format data apa pun termasuk data JSON. Karena model data juga

merupakan kunci dan fondasi untuk teknologi manajemen data lainnya, seperti pengindeksan data, kueri data, pencarian data, pemetaan data, integrasi data, dan penambangan data, bagaimana merancang model yang efisien dan kuat untuk data JSON diperlukan dan penting untuk topik penelitian terkait lainnya, seperti integrasi data, pencarian dan kueri data, kualitas dan evaluasi data, dll (Lv, Yan and He, 2018). Keuntungan dari pemakaian JSON antara lain penggunaan Bahasa yang sederhana dan banyak dipakai oleh pengembang, tipe sistem yang sederhana, dan Sebagian besar penggunaan API sekarang menggunakan data JSON. Atau dalam kata lain JSON merupakan format teks untuk pemberian label supata data terstruktur. Itu berasal dari literal objek *JavaScript*. Seperti *JavaScript*, JSON dapat mewakili empat tipe primitif (*string*, angka, *boolean*, dan *null*) dan dua tipe terstruktur seperti objek maupun *array* (Code, no date).

2.11 Penelitian Terdahulu

Berikut merupakan rangkuman hasil penelitian terdahulu yang telah dikaji oleh beberapa penulis sebagai bahan referensi pada penelitian ini.

Tabel 2. 1 Penelitian Terdahulu

No.	Judul Penelitian	Peneliti Dan Tahun	Metode	Hasil Penelitian
1.	Implementasi Arsitektur <i>Microservices</i> Pada Rancang Bangun Aplikasi Marketplace Berbasis Web	(Sinambela and Coastera, 2021)	<i>Microservice</i>	Aplikasi <i>marketplace</i> menggunakan arsitektur <i>microservices</i> telah berhasil dibangun dengan hasil persentase pengujian <i>black box</i> 100% dari 25 aktivitas berhasil dan hasil persentase pengujian <i>endpoint</i> api 100% dari 29 aktivitas berhasil. Pada penelitian ini memiliki 3 aktor yaitu Admin, Penjual dan Pembeli.
2.	Implementasi Arsitektur <i>Microservice</i> Pada Aplikasi <i>Web</i> Pengajaran Agama Islam <i>Home</i> Pesantren	(Yuri Chandra Tri Putra, Thomas Adi Purnomo Sidi and Joseph Eric Samodra, 2020)	<i>Microservice</i>	Penerapan komunikasi antar <i>microservice</i> sehingga dapat menjadi sebuah kesatuan adalah dengan menerapkan dua teknik komunikasi yang berbeda, yaitu <i>REST API</i> untuk mendukung <i>synchronous communication</i> dan RabbitMQ untuk mendukung <i>asynchronous communication</i> . Pada penelitian ini memiliki 3 aktor yaitu pengurus pesantren, admin dan anak-anak pesantren.

No.	Judul Penelitian	Peneliti Dan Tahun	Metode	Hasil Penelitian
3.	Analisis dan Desain Arsitektur <i>Microservice</i> dengan GraphQL Sebagai <i>API Gateway</i> untuk Sistem Informasi Akademik AIS UIN Jakarta	(Radhiyan, 2020)	<i>Microservice</i> dan GraphQL	Analisis dan desain arsitektur <i>microservices</i> dengan GraphQL sebagai <i>API Gateway</i> untuk sistem informasi akademik AIS UIN Jakarta dilakukan dengan metode <i>decomposition by business capabilities</i> , artinya metode tersebut dapat digunakan untuk analisis dan desain arsitektur <i>microservices</i> untuk sistem informasi akademik AIS UIN Jakarta. Pada penelitian ini memiliki 3 aktor yaitu admin,Dosen dan mahasiswa.
4.	Aplikasi Apotek Berbasis Web Menggunakan Arsitektur <i>Microservices</i> (Studi Kasus Apotek Glen, Kab.Toba)	(Microservices <i>et al.</i> , 2021)	<i>Microservices</i>	Aplikasi menggunakan arsitektur <i>microservices</i> merupakan keputusan yang tepat apabila ingin membangun aplikasi dalam skala yang besar dikarenakan aplikasi dibangun terdiri dari beberapa <i>service</i> yang masing - masing memiliki satu tanggung jawab. Berbeda dengan arsitektur monolitik yang semua layanan berada

No.	Judul Penelitian	Peneliti Dan Tahun	Metode	Hasil Penelitian
				dalam satu komponen sehingga satu program akan menangani semua layanan dalam aplikasi.
5.	Desain Sistem Terfederasi Dengan Pendekatan <i>Microservice Architecture</i> Pada Kasus Studi Sistem Pelaporan Pajak	(Darmayantie, 2020)	<i>Microservice</i>	Teknologi <i>web service</i> berbasis <i>REST-API</i> digunakan untuk menjamin interoperabilitas data dan informasi pada masing-masing sistem.
6.	<i>Optimizing Service Placement for Microservice Architecture in Clouds</i>	(Hu, de Laat and Zhao, 2019)	<i>Microservice</i>	Dalam makalah ini, kami menyelidiki masalah penempatan layanan untuk arsitektur layanan mikro di awan. Untuk menemukan partisi berkualitas tinggi dari aplikasi berbasis layanan, kami mengusulkan dua algoritma partisi: Partisi Biner dan Partisi K, yang didasarkan pada algoritma kontraksi acak yang dirancang dengan baik..
7.	Rancang Bangun <i>Back-End</i> “Siap”: Sistem Informasi Aspirasi Dan Pengaduan Masyarakat Berbasis Web	(Jayanto, 2017)	<i>Microservice Springboot</i>	Dalam membangun aplikasi SIAP yang berbentuk <i>microservice</i> maka kebutuhan fungsional harus dipecah menjadi beberapa <i>microservice</i> yang saling berhubungan

No.	Judul Penelitian	Peneliti Dan Tahun	Metode	Hasil Penelitian
	Dengan Menggunakan Metode <i>Microservice Springboot</i>			sehingga membentuk satu kesatuan aplikasi sesuai dengan proses bisnis yang telah ditentukan.
8.	Arsitektur <i>Microservice</i> untuk Resiliensi Sistem Informasi	(Suryotrisongko, 2017)	<i>Microservice</i>	Pola komunikasi <i>REST</i> Service dengan format JSON memberikan kemudahan dalam implementasi dan juga tidak membebani server
9.	Aplikasi <i>Taking Order (Front End)</i> Menggunakan <i>Microservice</i> Pada Pt. Xyz	(Rachman, 2018)	<i>Microservice</i> dan <i>DSRM</i>	Pengembangan sistem aplikasi <i>taking order</i> sudah dilakukan dan menghasilkan aplikasi <i>taking order (front end)</i> menggunakan <i>microservice</i> pada PT XYZ untuk diimplementasikan di PT XYZ
10.	Data Aggregation in <i>Microservice</i> Architecture	(Damyanov, 2019)	<i>Microservice</i>	Pemrosesan data yang terdistribusi seperti pada <i>Lazy Evaluation</i> , LINQ, dan <i>tuple</i> membuat pemrosesan data yang tidak memperhatikan urutan, ini sangat penting untuk sistem yang memerlukan data yang ditampilkan secara <i>real-time</i> .

BAB 3

METODE PENELITIAN

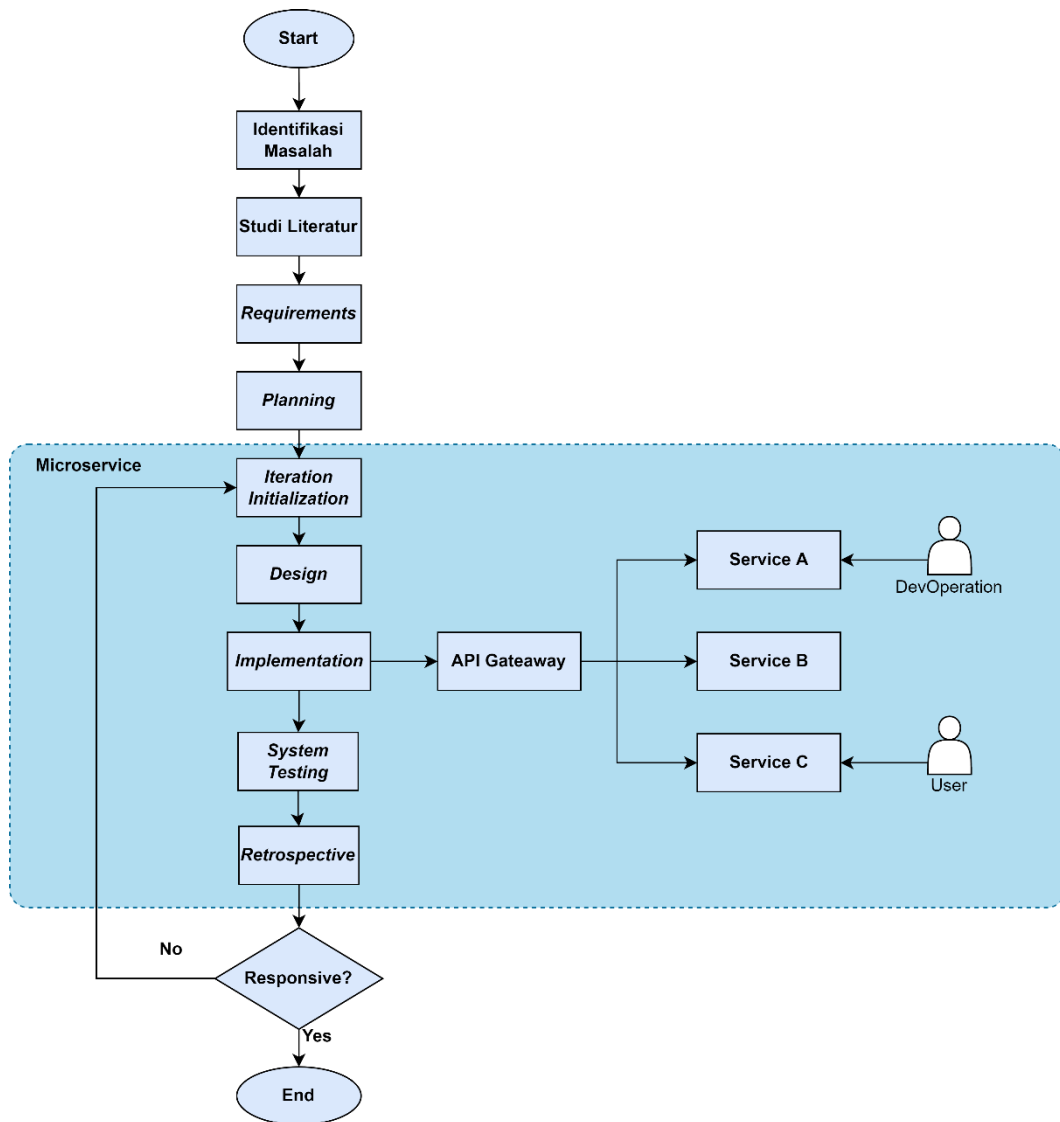
Pada bab ini menjelaskan mengenai tahapan-tahapan yang dilakukan selama penelitian meliputi, garis besar penelitian, prosedur penelitian, diagram alir penelitian dan rencana jadwal penelitian.

3.1 Garis Besar Penelitian

Secara garis besar penelitian ini dilakukan dengan metode *Microservice* yang terdiri dari beberapa tahapan yaitu identifikasi masalah, studi literatur, *requirements, planning, iteration initialization, design, implementation, system testing* dan *retrospective*. Tahapan-tahapan tersebut akan menjadi acuan dalam proses penelitian pengembangan sistem informasi manajemen Tugas Akhir di Program Studi Informatika ITK.

3.2 Diagram Alir Penelitian

Adapun diagram alir dari penelitian ini adalah sebagai berikut :



Gambar 3. 1 Diagram Alir Pengembangan Sistem Manajemen Tugas Akhir

Pada Gambar 3.1 ditunjukkan diagram alir dalam penelitian ini. Terdiri dari 8 tahapan, yaitu identifikasi masalah, studi literatur, *requirements*, *planning*, *iteration initialization*, *design*, *implementation*, *API Getaway*, *system testing*, *retrospective*, yang tentunya ditutup dengan pengujian tingkat kelayakan sistem.

3.3 Prosedur Penelitian

Berikut adalah prosedur penelitian yang berisi penjelasan untuk lebih mendetailkan setiap langkah-langkah pengerjaan pada penelitian ini :

3.3.1 Identifikasi Masalah

Langkah awal yang dilakukan dalam penelitian ini merupakan identifikasi masalah yang ada pada Program Studi Informatika ITK. Tahapan ini dilakukan untuk mendapatkan gambaran besar mengenai latar belakang penelitian yang diangkat, tujuan dilakukannya penelitian, dan penjelasan mengapa penelitian ini harus dilakukan. Pengindentifikasian masalah dilakukan dengan metode observasi dan wawancara kepada pihak terkait. Penggunaan metode *MicroService* pada penelitian ini cocok digunakan pada pembangunan sistem informasi manajemen TA ini, dikarenakan merupakan metode ini dapat membagi berbagai macam *Service* dalam server. Hal ini dapat mempermudah proses penyaluran data atau informasi secara *real-time*.

3.3.2 Studi Literatur

Langkah selanjutnya merupakan studi literatur. Tahap ini berisi kegiatan mempelajari literatur yang berkaitan dengan penelitian. Mulai dari pencarian sumber teori dan literatur yang digunakan berasal dari buku, jurnal, prosiding, buku panduan maupun *website* yang dapat dipercaya kebenaran datanya. Pada penelitian ini, studi literatur yang dilakukan berkaitan dengan Program Studi Informatika ITK, SIM-TA, *Web Service*, *Microservice*, *PHP*, *Laravel*, *UML*, *JSON*, dan *Jmeter Apache*.

3.3.3 Requirements

Tahap ini berisi kegiatan menggali informasi dengan metode wawancara kepada pihak koordinator program studi Informatika dengan kebutuhan yang diperlukan untuk diterapkan ke dalam sistem, dan kemudian hasil dari wawancara tersebut akan dianalisis sehingga menghasilkan dokumen *user story*. Nantinya dokumen *user story* akan digunakan sebagai acuan bagi pengembang untuk merancang serta membangun sistem agar sesuai dengan kebutuhan dari *stackholder* yaitu Program Studi Informatika ITK. *User story* pada pengembangan sistem informasi manajemen TA Informatika ITK adalah sebagai berikut.

Tabel 3. 1 Rangkuman *User Story*

ID	Judul	Deskripsi	<i>Acceptance Criteria</i>
US-1	<i>Login</i>	Seluruh Pengguna dapat melakukan <i>Login</i>	<i>Login</i> dapat dilakukan oleh seluruh pengguna, namun memiliki keterbatasan hak akses sesuai dengan <i>role</i> -nya
US-2	<i>Logout</i>	Seluruh pengguna dapat melakukan <i>Logout</i> untuk keluar dari sistem	Seluruh Pengguna
US-3	Tambah Daftar Bimbingan	Fitur ini digunakan untuk menambah daftar bimbingan maupun dosen pembimbing	<i>Super admin</i> , admin (Tendik Jurusan), Mahasiswa, Dosen
US-4	Lihat Daftar Bimbingan	Fitur ini digunakan untuk menampilkan daftar bimbingan maupun dosen pembimbing	Seluruh Pengguna, dengan batasan data pribadi sendiri
US-5	Edit Daftar Bimbingan	Fitur ini digunakan untuk mengubah data daftar bimbingan maupun dosen pembimbing	<i>Super admin</i> , admin (Tendik Jurusan), Dosen
US-6	Hapus Daftar Bimbingan	Fitur ini digunakan untuk menghapus data daftar bimbingan maupun dosen pembimbing	<i>Super admin</i> , admin (Tendik Jurusan), Dosen
US-7	Daftar Sempro	Fitur ini digunakan untuk menambah daftar sempro	Hanya Mahasiswa dengan batasan data pribadi sendiri
US-8	Lihat Data Sempro	Fitur ini digunakan untuk menampilkan data daftar sempro	<i>Super admin</i> , admin (Tendik Jurusan), Koorprodi, Dosen

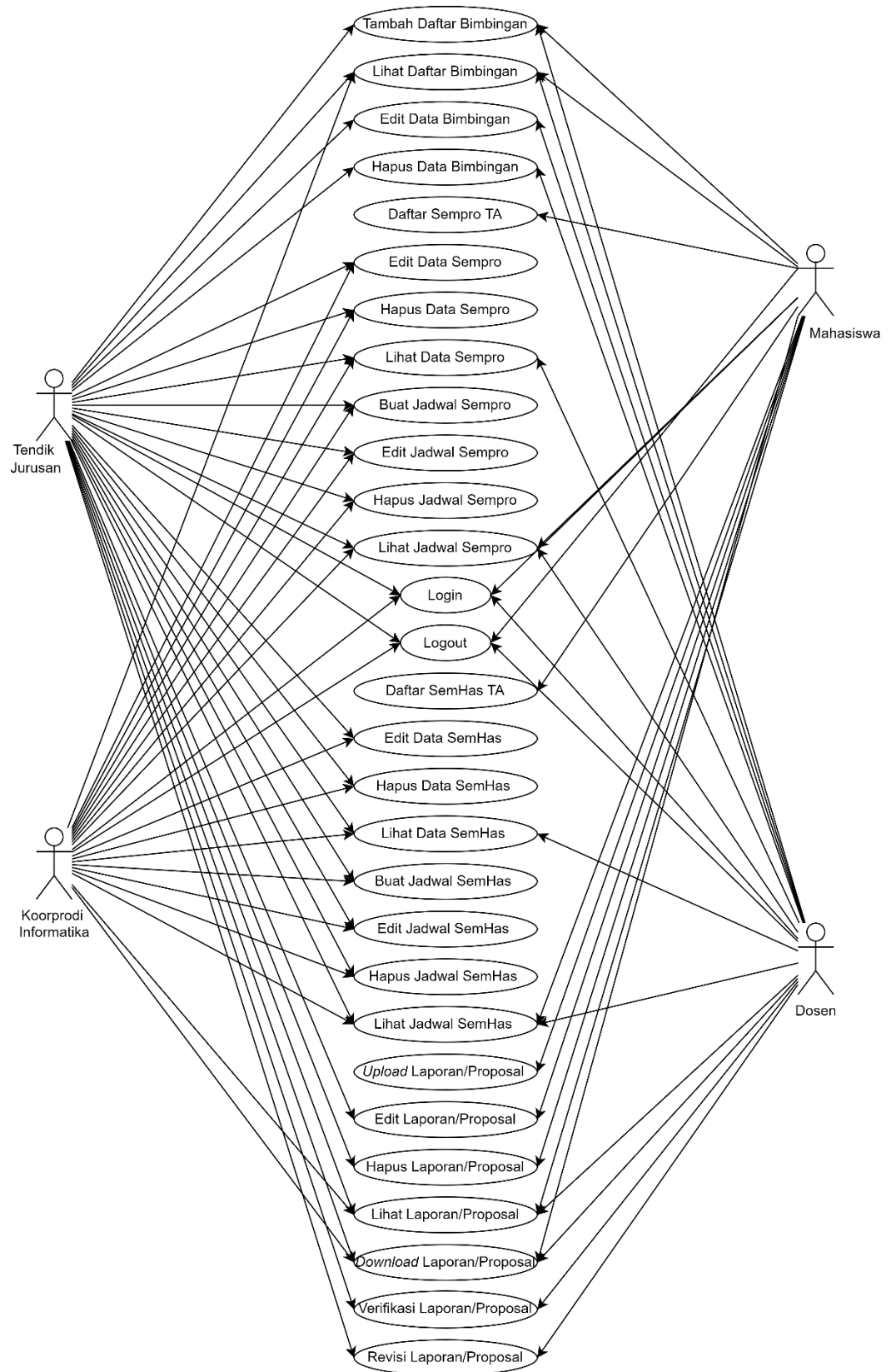
US-9	Edit Daftar Sempro	Fitur ini digunakan untuk mengubah data daftar sempro	<i>Super admin</i> , admin (Tendik Jurusan), Koorprodi
US-10	Hapus Daftar Sempro	Fitur ini digunakan untuk menghapus data daftar sempro	<i>Super admin</i> , admin (Tendik Jurusan), Koorprodi
US-11	Buat Jadwal Sempro	Fitur ini digunakan untuk menambah data jadwal sempro	<i>Super admin</i> , admin (Tendik Jurusan), Koorprodi
US-12	Lihat Jadwal Sempro	Fitur ini digunakan untuk menampilkan data jadwal sempro	Seluruh pengguna
US-13	Edit Jadwal Sempro	Fitur ini digunakan untuk mengubah data jadwal sempro	<i>Super admin</i> , admin (Tendik Jurusan), Koorprodi
US-14	Hapus Jadwal Sempro	Fitur ini digunakan untuk menghapus data jadwal sempro	<i>Super admin</i> , admin (Tendik Jurusan), Koorprodi
US-15	Daftar SemHas TA	Fitur ini digunakan untuk menambah data daftar semHas TA	Hanya mahasiswa
US-16	Lihat Data SemHas	Fitur ini digunakan untuk menampilkan data daftar semHas	<i>Super admin</i> , admin (Tendik Jurusan), Koorprodi, dan Dosen
US-17	Edit Data SemHas	Fitur ini digunakan untuk mengubah data daftar semHas TA	<i>Super admin</i> , admin (Tendik Jurusan), Koorprodi
US-18	Hapus Data SemHas	Fitur ini digunakan untuk menghapus data daftar semHas	<i>Super admin</i> , admin (Tendik Jurusan), Koorprodi

US-19	Buat Jadwal SemHas	Fitur ini digunakan untuk menambah data jadwal semHas TA	<i>Super admin</i> , admin (Tendik Jurusan), Koorprodi
US-20	Lihat Jadwal SemHas	Fitur ini digunakan untuk menampilkan data jadwal semHas	Seluruh pengguna
US-21	Edit Jadwal SemHas	Fitur ini digunakan untuk mengubah data jadwal semHas	<i>Super admin</i> , admin (Tendik Jurusan), Koorprodi
US-22	Hapus Jadwal SemHas	Fitur ini digunakan untuk menghapus data jadwal semHas	<i>Super admin</i> , admin (Tendik Jurusan), Koorprodi
US-23	<i>Upload</i> Laporan/Proposal	Fitur ini digunakan untuk menambahkan/meng- <i>upload</i> laporan maupun proposal mahasiswa	Mahasiswa
US-24	Edit Laporan/Proposal	Fitur ini digunakan untuk mengubah laporan maupun proposal mahasiswa	<i>Super admin</i> , admin (Tendik Jurusan) dan mahasiswa
US-25	Hapus Laporan/Proposal	Fitur ini digunakan untuk menghapus laporan maupun proposal mahasiswa	<i>Super admin</i> , admin (Tendik Jurusan) dan mahasiswa
US-26	Lihat Laporan/Proposal	Fitur ini digunakan untuk menampilkan laporan maupun proposal mahasiswa	Seluruh pengguna
US-27	Download Laporan/Proposal	Fitur ini digunakan untuk mengekspor laporan maupun proposal mahasiswa	Seluruh pengguna

US-28	Verifikasi Laporan/Proposal	Fitur ini digunakan untuk memvalidasi laporan maupun proposal mahasiswa	<i>Super admin</i> , admin (Tendik Jurusan) dan Dosen
US-29	Revisi Laporan/Proposal	Fitur ini digunakan untuk memberikan revisi laporan maupun proposal mahasiswa	<i>Super admin</i> , admin (Tendik Jurusan) dan Dosen

3.3.4 Planning

Langkah selanjutnya pada tahap ini berisi kegiatan merencanakan secara garis besar bagaimana sistem akan dibuat dan menentukan fungsionalitas secara keseluruhan yang akan dikembangkan pada sistem yang akan dikerjakan berdasarkan dari *user story* yang telah dihasilkan pada proses *requirements*. Pada tahap ini juga akan dilakukan perencanaan tingkat prioritas fitur – fitur apa saja yang akan dibuat pada sistem dengan cara membuat kumpulan *task* berdasarkan *user story*, yang dimana kumpulan *task* ini dapat terdiri dari *task – task* kecil yang akan dikategorikan. Setelah daftar perencanaan selesai dibuat, tentunya akan dilanjutkan perencanaan jadwal perancangan dan pembangunan sistem. Berikut merupakan gambaran besar *Use Case* pada SIM TA Informatika ITK.



Gambar 3. 2 Use Case SIM-TA Informatika ITK

Gambar 3.2 menunjukkan bahwa *Use Case* SIM TA Informatika ITK memiliki 29 *Use Case*. Terdapat 4 aktor yakni *staff* Tendik Jurusan, Koorprodi, Dosen dan mahasiswa, masing-masing memiliki jumlah *Use Case* berbeda. Penjelasan terkait deskripsi lengkap serta aktor dari tiap *Use Case* ditunjukkan Tabel 3.1, maka didapatkan beberapa iterasi seperti table berikut:

Tabel 3. 2 Daftar Iterasi

No	User Story ID	Judul	Story Point
ITERASI 1			
1.	US-1	<i>Login</i>	3
2.	US-2	<i>Logout</i>	3
4.	US-3	Tambah daftar bimbingan	3
5.	US-4	Lihat daftar bimbingan	3
6.	US-5	Edit data bimbingan	3
7.	US-6	Hapus data bimbingan	2
Total			17
ITERASI 2			
1.	US-7	Daftar Sempro	2
2.	US-8	Lihat data sempro	3
3.	US-9	Edit daftar sempro	3
4.	US-10	Hapus daftar sempro	2
Total			10
ITERASI 3			
1.	US-11	Buat jadwal sempro	3
2.	US-12	Lihat jadwal sempro	3
3.	US-13	Edit jadwal sempro	3
4.	US-14	Hapus jadwal sempro	2
Total			11
ITERASI 4			
1.	US-15	Daftar semhas TA	2
2.	US-16	Lihat data semhas	3
3.	US-17	Edit data semhas	3

4.	US-18	Hapus data semhas	2
Total			10
ITERASI 5			
1.	US-19	Buat jadwal semhas	3
2.	US-20	Lihat jadwal semhas	3
3.	US-21	Edit jadwal semhas	3
4.	US-22	Hapus jadwal semhas	2
Total			11
ITERASI 6			
1.	US-23	<i>Upload</i> Laporan/Proposal	3
2.	US-24	Edit Laporan/Proposal	3
3.	US-25	Hapus Laporan/Proposal	2
4.	US-26	Lihat Laporan/Proposal	3
5.	US-27	Download Laporan/Proposal	3
6.	US-28	Verifikasi Laporan/Proposal	3
7.	US-29	Revisi Laporan/Proposal	3
Total			20

3.3.5 *Iteration Initialization*

Selanjutnya pada tahap *iteration initialization* dilakukan inisialisasi terhadap iterasi yang dilakukan dalam proses pengembangan Sistem Informasi Manajemen pengelolaan tugas akhir. Penggunaan iterasi bertujuan untuk memberikan batasan pada pengerjaan agar lebih efisien serta dapat menunjukkan *progress* dari pengembangan Sistem Informasi Manajemen Tugas Akhir. Pada proses iterasi dilakukan pembagian terhadap tahapan pengembangan sistem menjadi unit kecil. Dimulai dari pemilihan *tasks* yang akan menjadi fokus utama iterasi

3.3.6 *Design*

Selanjutnya pada tahap ini dilakukan proses perancangan fungsi, kelas, alur data dan model dari sistem yang akan di implementasikan terhadap iterasi yang sedang berjalan. Design rancangan sistem hanya memenuhi kebutuhan *stack holder*

saat ini tanpa membuat sebuah asumsi apa yang akan dibutuhkan dimasa yang akan datang. Dalam proses design harus dilakukan sesederhana mungkin karena akan berdampak pada waktu pengembangan yang lebih cepat dan lebih sedikit terkait kode yang digunakan. Yang dimana waktu tersebut sangat bermanfaat pada proses iterasi pengembangan karena dapat mempersingkat waktu iterasi. Begitu juga sebaliknya ketika desain tampilan yang rumit akan mempengaruhi pada waktu pengembangan yang lebih lama. Selain mendesain tampilannya, pada sistem ini perlu adanya design database yang nantinya dapat digunakan untuk proses penggunaan sistem yang akan dikerjakan. *ERD (Entity Relationship Data)* digunakan untuk merancang basis data yang akan dibangun sebagai media pengelolaan data dari sistem. *ERD (Entity Relationship Data)* nantinya akan digunakan sebagai acuan dalam pembuatan basis data dari sistem. Kemudian desain tampilan sistem digunakan untuk menggambarkan secara garis besar bagaimana nantinya tampilan dari sistem akan dibuat. Berikut merupakan perancangan *design website* menggunakan sistem *REST API* pada SIM-TA ini.

Tabel 3. 3 Deskripsi Format API Bimbingan

<i>Transport Protocol</i>	:	<i>HTTP</i>
<i>Content Type</i>	:	<i>JSON</i>
<i>Interaction Type</i>	:	<i>Synchronous</i>
<i>Security Policy</i>	:	<i>Token Key</i>
<i>Format Request Template</i>	:	{ <div> “bimbingan_id”: “ “, “action”: “ “, “data”: “ “ </div> }
<i>Format Response Template</i>	:	{ <div> “bimbingan_id”: “ “, “response_code”: “ “, “response_message”: “ “ </div> }

Berikut merupakan detail dan *endpoint* API yang disajikan pada Tabel 3.3:

Tabel 3. 4 Deskripsi API Service Bimbingan

API	Deskripsi
/Bimbingan	Membaca semua data bimbingan
/Bimbingan/edit/id	Mengedit data bimbingan
/Bimbingan/store	Mengirim data bimbingan
/Bimbingan/id	Melihat detail bimbingan

Tabel 3. 5 Deskripsi Format API Sempro

<i>Transport Protocol</i>	:	<i>HTTP</i>
<i>Content Type</i>	:	JSON
<i>Interaction Type</i>	:	<i>Synchronous</i>
<i>Security Policy</i>	:	<i>Token Key</i>
<i>Format Request Template</i>	:	{ “sempro_id”: “ “, “action”: “ “, “data”: “ “ }
<i>Format Response Template</i>	:	{ “sempro_id”: “ “, “response_code”: “ “, “response_message”: “ “ }

Berikut merupakan detail dan *endpoint* API yang disajikan pada Tabel 3.5:

Tabel 3. 6 Deskripsi API Service Sempro

API	Deskripsi
/Sempro	Membaca semua data Sempro
/Sempro/edit/id	Mengedit data Sempro
/Sempro/store	Mengirim data Sempro
/Sempro/id	Melihat detail Sempro

Tabel 3. 7 Deskripsi Format API Jadwal Sempro

<i>Transport Protocol</i>	:	<i>HTTP</i>
<i>Content Type</i>	:	JSON
<i>Interaction Type</i>	:	<i>Synchronous</i>
<i>Security Policy</i>	:	<i>Token Key</i>
<i>Format Request Template</i>	:	{ “Jsempro_id”: “ “, “action”: “ “, “data”: “ “ }
<i>Format Response Template</i>	:	{ “Jsempro_id”: “ “, “response_code”: “ “, “response_message”: “ “ }

Berikut merupakan detail dan *endpoint* API yang disajikan pada Tabel 3.7:

Tabel 3. 8 Deskripsi API Service Jadwal Sempro

API	Deskripsi
/JSempro	Membaca semua data Jadwal Sempro
/JSempro/edit/id	Mengedit data Jadwal Sempro
/JSempro/store	Mengirim data Jadwal Sempro
/JSempro/id	Melihat detail Jadwal Sempro

Tabel 3. 9 Deskripsi Format API SemHas

<i>Transport Protocol</i>	:	<i>HTTP</i>
<i>Content Type</i>	:	JSON
<i>Interaction Type</i>	:	<i>Synchronous</i>
<i>Security Policy</i>	:	<i>Token Key</i>
<i>Format Request Template</i>	:	{ “semhas_id”: “ “, “action”: “ “, }

		<pre> “data”: “ “ } </pre>
<i>Format Response</i>	<i>Template</i>	<pre> : { “semhas_id”: “ “, “response_code”: “ “, “response_message”: “ “ } </pre>

Berikut merupakan detail dan *endpoint* API yang disajikan pada Tabel 3.9:

Tabel 3. 10 Deskripsi API Service SemHas

API	Deskripsi
/Semhas	Membaca semua data SemHas
/Semhas/edit/id	Mengedit data SemHas
/Semhas/store	Mengirim data SemHas
/Semhas/id	Melihat detail SemHas

Tabel 3. 11 Deskripsi Format API Jadwal SemHas

<i>Transport Protocol</i>	:	<i>HTTP</i>
<i>Content Type</i>	:	JSON
<i>Interaction Type</i>	:	<i>Synchronous</i>
<i>Security Policy</i>	:	<i>Token Key</i>
<i>Format Request</i>	<i>Template</i>	<pre> : { “Jsemhas_id”: “ “, “action”: “ “, “data”: “ “ } </pre>
<i>Format Response</i>	<i>Template</i>	<pre> : { “Jsemhas_id”: “ “, “response_code”: “ “, “response_message”: “ “ } </pre>

		}
--	--	---

Berikut merupakan detail dan *endpoint* API yang disajikan pada Tabel 3.11:

Tabel 3. 12 Deskripsi API Service Jadwal SemHas

API	Deskripsi
/JSemhas	Membaca semua data Jadwal SemHas
/JSemhas/edit/id	Mengedit data Jadwal SemHas
/JSemhas/store	Mengirim data Jadwal SemHas
/JSemhas/id	Melihat detail Jadwal SemHas

Tabel 3. 13 Deskripsi Format API Proposal

<i>Transport Protocol</i>	:	<i>HTTP</i>
<i>Content Type</i>	:	JSON
<i>Interaction Type</i>	:	<i>Syncronous</i>
<i>Security Policy</i>	:	<i>Token Key</i>
<i>Format Request Template</i>	:	{ “proposal_id”: “ “, “action”: “ “, “data”: “ “ }
<i>Format Response Template</i>	:	{ “proposal_id”: “ “, “response_code”: “ “, “response_message”: “ “ }

Berikut merupakan detail dan *endpoint* API yang disajikan pada Tabel 3.13:

Tabel 3. 14 Deskripsi API Service Proposal

API	Deskripsi
/Proposal	Membaca semua data Proposal
/Proposal/edit/id	Mengedit data Proposal

/Proposal/store	Mengirim data Proposal
/Proposal/id	Melihat detail Proposal

Tabel 3. 15 Deskripsi Format API Download Proposal

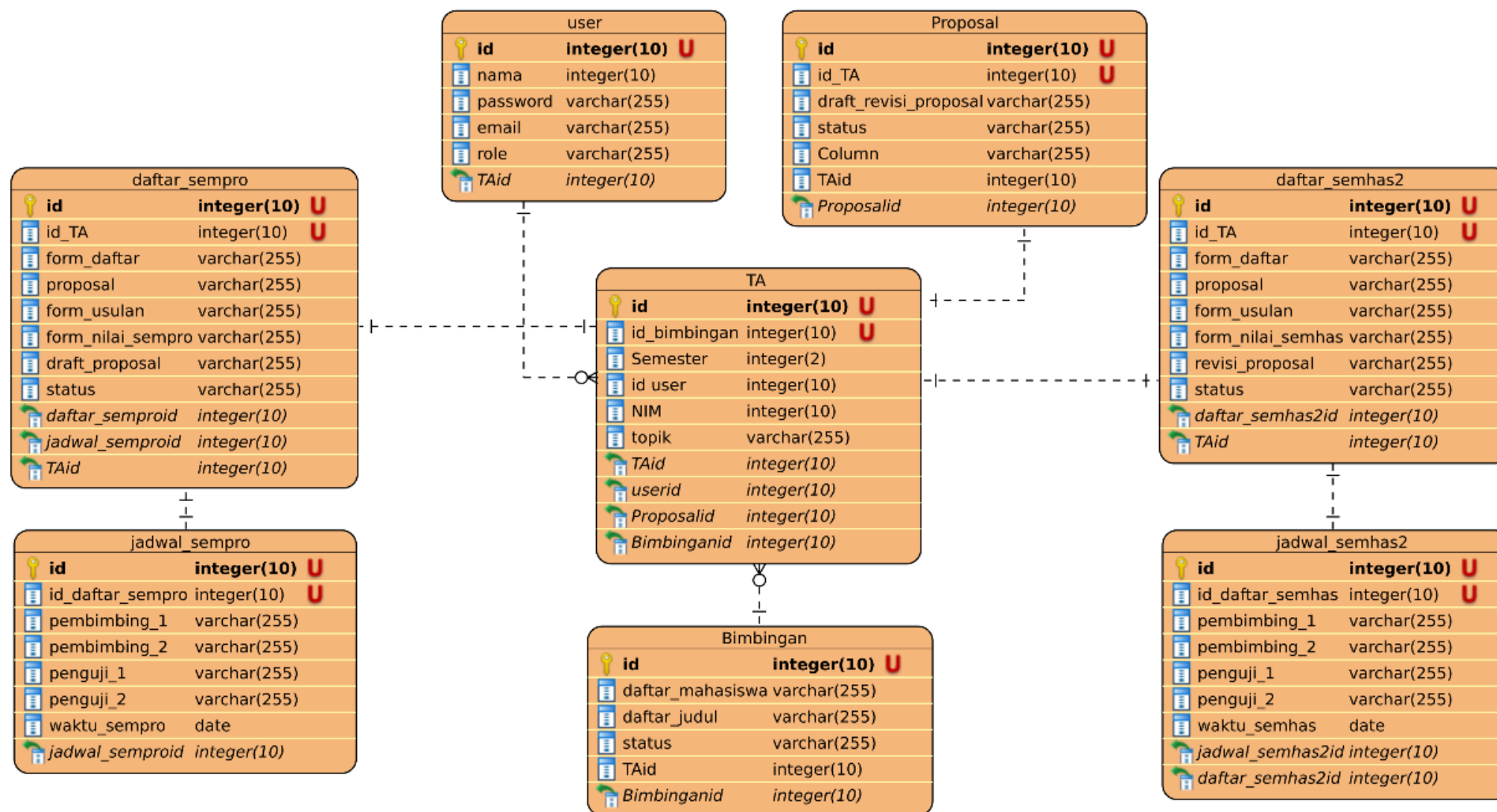
<i>Transport Protocol</i>	:	<i>HTTP</i>
<i>Content Type</i>	:	JSON
<i>Interaction Type</i>	:	<i>Synchronous</i>
<i>Security Policy</i>	:	<i>Token Key</i>
<i>Format Request Template</i>	:	{ “proposal_id”: “ “, “action”: “ “, “data”: “ “ }
<i>Format Response Template</i>	:	{ “proposal_id”: “ “, “response_code”: “ “, “response_message”: “ “ }

Berikut merupakan detail dan *endpoint* API yang disajikan pada Tabel 3.15:

Tabel 3. 16 Deskripsi API Service Proposal

API	Deskripsi
/download/id	Mengunduh proposal sesuai dengan id yang telah dipilih

Berdasarkan beberapa tabel API di atas, maka dibentuklah *Design database website* SIM TA Informatika ITK menggunakan fitur *migration* pada *Laravel*. Berikut merupakan desain yang dirancang untuk *website* SIM TA Informatika ITK



Gambar 3. 3 Database SIM TA Informatika IT

3.3.7 Implementation

Selama proses implementation, dilakukan proses Coding atau penulisan kode program, *unit testing* untuk setiap unit dan proses pengimplementasian metode *microservice*. Pada coding, berisi penulisan spesifikasi API. Yang dimana dalam API tersebut berisi daftar *end point*, *request* dan *response* serta informasi pendukung lainnya seperti pada rancangan pendukung penyusunan API. Kemudian pada tahap berikutnya ialah perancangan *design User Interface*, yang terbagi untuk beberapa *service*. Yang dimana untuk *service* utama akan dipegang oleh *developer* atau pihak pengembang. Dan untuk *service* terakhir akan dapat diakses oleh siapapun. Yang pada akhirnya setiap *service* akan tersusun kembali ke dalam masing-masing *database* yang berbeda. Kemudian setelah pendistribusian *service* telah dilakukan, maka tahap berikutnya ialah *unit testing* untuk setiap unit nya. Penggunaan *Code Coverage* untuk *unit testing* dapat membantu dalam proses penguji cobaan fungsi setiap *code*. Hasil tolak ukur tingkat presentasi kode yang telah diuji akan menyatakan tingkat kelayakan sistem. Proses ini akan menilai apakah fungsi atau fitur yang dibuat sudah benar atau belum. Jika belum, maka akan dilakukan proses *refactor* yaitu proses penulisan kembali kode program untuk memperbaiki kesalahan atau ketidaksesuaian dari penulisan sebelumnya. Dikarenakan pendistribusian *service* telah dilakukan, pengerjaan proses *refactor* akan lebih terfokus sesuai pembagian tersebut.

3.3.8 System Testing

Selanjutnya pada tahap ini berisi pengujian terhadap sistem yang telah dirancang dan dibangun. Tahap ini menguji fitur-fitur dan kesesuaian dengan kebutuhan *stackholder* berdasarkan *user story* yang diberikan pada tahap awal pengerjaan maupun keinginan dari *stackholder* yang disampaikan selama tahap pengerjaan berlangsung. Pada tahap ini akan dilakukan pengujian juga apakah fitur-fitur yang saling berkaitan telah dapat berjalan dengan baik tanpa ada *bug* atau *error* pada sistem. Pengujian dilakukan satu kali dalam tiga minggu dengan menggunakan proses aplikasi *Jmeter Apache* bersama dengan *stakeholder* untuk melakukan pengujian terhadap *system* yang telah dikembangkan. Pengujian ini dilakukan untuk keseluruhan fitur.

Dimulai dari pengujian *load*, yang di mana menguji lama waktunya sistem dalam mengelola data yang dimasukkan. Kemudian melakukan pengujian terhadap data yang ingin dibaca pada sistem, seperti menkonfigurasi *GET Login* maupun *Method Post* pada sistem. Pengujian ini memiliki beberapa faktor penilaian seperti lamanya waktu saat *loading*, menguji data saat di akses dan juga bisa dipengaruhi oleh faktor eksternal seperti tingkat banyaknya *user* pada saat mengakses sistem ini. Kemudian masuk ke dalam *stress testing* untuk menguji sistem pada saat menerima banyaknya data dan menguji sistem pada saat diberikan batas persyaratan yang telah ditentukan. Kemudian dapat juga melakukan *spike testing* dan *capacity testing*, yang bertujuan untuk memastikan skalabilitas dan stabilitas sistem yang telah dibuat. Apabila tidak ditemukan kesalahan atau *error* pada proses pengujian ini maka akan dilanjutkan ke tahap *retrospective*.

3.3.9 Retrospective

Tahap ini adalah tahap pengambilan kesimpulan terkait sistem yang telah dibangun, apakah sistem ini telah bisa digunakan sepenuhnya atau masih terdapat beberapa hal yang perlu diperbaiki kembali. Tahap ini akan menjadi diskusi akhir apakah sistem telah dapat memasuki tahap *production* atau belum, dimana apabila masih terdapat ketidaksesuaian dengan kebutuhan atau keinginan dari *stackholder*. Jika belum, maka tahap ini harus kembali kepada tahap *iteration and initialization* hingga pihak *stackholder* telah setuju secara keseluruhan dengan sistem yang telah dirancang bangun. Proses ini akan berhenti apabila sudah tidak ditemukan permasalahan pada Sistem Informasi Manajemen pengelolaan Tugas Akhir serta sudah memenuhi semua kebutuhan atau keinginan dari *stakeholder*.

3.4 Rencana Jadwal Penelitian

Berdasarkan keseluruhan tahap pengembangan Sistem Informasi Manajemen Tugas Akhir Informatika ITK ini, maka dapat disusun rencana penjadwalan penelitian sebagai berikut:

Tabel 3. 17 Rencana Jadwal Penelitian

Kegiatan	Bulan Ke-															
	1				2				3				4			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Identifikasi Masalah																
Studi Literatur																
<i>Requirements</i>																
<i>Planning</i>																
<i>Iteration Initialization</i>																
<i>Design</i>																
<i>Implementation</i>																
<i>System testing</i>																
<i>Retrospective</i>																
Penulisan Laporan																

DAFTAR PUSTAKA

- Code, L. (no date) ‘Что Takoe Takoe *REST API*?’, *Toster*. Available at: [HTTPS://toster.ru/q/136265](https://toster.ru/q/136265).
- Damyanov, I. (2019) ‘Data aggregation in microservice architecture’, *International journal of online and biomedical engineering*, 15(12), pp. 81–87. doi: 10.3991/ijoe.v15i12.11095.
- Darmayantie, A. (2020) ‘Desain Sistem Terfederasi Dengan Pendekatan Microservice Architecture Pada Kasus Studi Sistem Pelaporan Pajak’, *Jurnal Ilmiah Informatika Komputer*, 25(1), pp. 50–63. doi: 10.35760/ik.2020.v25i1.2523.
- Halili, E. H. (2008) *Apache JMeter*, Packt Publishing.
- Hammoudeh, M. A. A. and Al-Ajlan, A. S. (2020) ‘Implementing web services using PHP soap approach’, *International Journal of Interactive Mobile Technologies*, 14(10), pp. 35–45. doi: 10.3991/ijim.v14i10.14391.
- Hermanto, B., Yusman, M. and Nagara, N. (2019) ‘Sistem Informasi Manajemen Keuangan pada PT. Hulu Balang Mandiri Menggunakan Framework Laravel’, *Jurnal Komputasi*, 7(1), pp. 17–26. doi: 10.23960/komputasi.v7i1.2051.
- Hu, Y., de Laat, C. and Zhao, Z. (2019) ‘Optimizing service placement for microservice architecture in clouds’, *Applied Sciences (Switzerland)*, 9(21). doi: 10.3390/app9214663.
- Izzaty, R. E., Astuti, B. and Cholimah, N. (1967) ‘~~濟無~~No Title No Title No Title’, *Angewandte Chemie International Edition*, 6(11), 951–952., pp. 5–24.
- Jayanto, D. P. (2017) ‘Informasi Web Application Backend Development of “ Siap ”: “ Sistem Informasi Aspirasi Dan Pengaduan Masyarakat ” Application Based on Web Using Microservice Springboot’, p. 156.
- Luca, V. and Ā, O. T. P. (2014) ‘Query Optimization Techniques in *Microsoft SQL Server*’, *Database Systems Journal*, 5(2), pp. 33–48.
- Lv, T., Yan, P. and He, W. (2018) ‘Survey on JSON Data Modelling’, *Journal of Physics: Conference Series*, 1069(1). doi: 10.1088/1742-6596/1069/1/012101.
- Machmud, M. (2016) ‘Tuntunan Penulisan TUGAS AKHIR Berdasarkan Prinsip Dasar Penelitian Ilmiah’, *Nucleic Acids Research*, pp. 1–266.
- Maulidiansyah, R., Rakhman, D. F. and Ramdhani, M. A. (2017) ‘Aplikasi Pelaporan Kerusakan Jalan Tol Menggunakan Layanan Web Service Berbasis Android’, *Jurnal ISTEK*, 10(1), pp. 117–123.
- Microservices, A. *et al.* (2021) ‘APLIKASI APOTEK BERBASIS WEB MENGGUNAKAN’, 1(2), pp. 22–28.

- Ortiz, G. *et al.* (2019) ‘Real-time context-aware microservice architecture for predictive analytics and smart decision-making’, *IEEE Access*, 7, pp. 183177–183194. doi: 10.1109/ACCESS.2019.2960516.
- 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), pp. 100–104. doi: 10.30743/infotekjar.v3i1.563.
- Permatasari, D. I. (2020) ‘Pengujian Aplikasi menggunakan metode Load Testing dengan Apache JMeter pada Sistem Informasi Pertanian’, *Jurnal Sistem dan Teknologi Informasi (JUSTIN)*, 8(1), p. 135. doi: 10.26418/justin.v8i1.34452.
- Purnama Sari, D. and Wijanarko, R. (2020) ‘Implementasi Framework Laravel pada Sistem Informasi Penyewaan Kamera (Studi Kasus di Rumah Kamera Semarang)’, *Jurnal Informatika dan Rekayasa Perangkat Lunak*, 2(1), p. 32. doi: 10.36499/jinrpl.v2i1.3190.
- Putra, I. M., Sitanggang, I. S. and Istiadi, M. A. (2019) ‘SIMETA ILKOM: Sistem Informasi Manajemen Tugas Akhir Program Studi S1 Ilmu Komputer IPB’, *Jurnal Ilmu Komputer dan Agri-Informatika*, 5(2), p. 109. doi: 10.29244/jika.5.2.109-118.
- Rachman, T. (2018) ‘*濟無*No Title No Title No Title’, *Angewandte Chemie International Edition*, 6(11), 951–952., pp. 10–27.
- Radhiyan, M. F. (2020) ‘Analisis dan Desain Arsitektur Microservices dengan GraphQL Sebagai API Gateway untuk Sistem Informasi Akademik AIS UIN Jakarta (Studi Kasus : AIS untuk Mahasiswa) Analisis dan Desain Arsitektur Microservices dengan GraphQL Sebagai API Gateway untuk Sis’.
- Available at:
[HTTps://repository.uinjkt.ac.id/dspace/handle/123456789/56187](https://repository.uinjkt.ac.id/dspace/handle/123456789/56187).
- Santoso, G. B., Sinaga, T. M. and Zuhdi, A. (2021) ‘MVC Implementation In Laravel Framework For Development Web-Based E-Commerce Applications’, *Intelmatiks*, 1(1), pp. 37–42. doi: 10.25105/itm.v1i1.7867.
- Seto, S. B., Wondo, M. T. S. and Mei, M. F. (2020) ‘Hubungan Motivasi Terhadap Tingkat Stress Mahasiswa Dalam Menulis Tugas Akhir (Skripsi)’, *Jurnal Basicedu*, 4(3), pp. 733–739. doi: 10.31004/basicedu.v4i3.431.
- Simatupang, J. and Muhammad, M. (2019) ‘Sistem Aplikasi Pengelolaan Tugas Akhir Berbasis Mobile’, *It Journal Research and Development*, 3(2), pp. 66–75. doi: 10.25299/itjrd.2019.vol3(2).2339.
- Sinambela, A. and Coastera, F. F. (2021) ‘Bangun Aplikasi Marketplace’, 9(1), pp. 1–13.
- Subecz, Z. (2021) ‘Web-development with Laravel framework’, *Gradus*, 8(1), pp. 211–218. doi: 10.47833/2021.1.csc.006.
- Surwase, V. (2016) ‘*REST API* Modeling Languages -A Developer ’ s Perspective

Related papers *REST API Modeling Languages - A Developer ' s Perspective*, *IJSTE - International Journal of Science Technology and Engineering*, 2(10), pp. 634–637. Available at: [HTTPS://www.academia.edu/27064725/REST_API_Modeling_Languages_A_Developers_Perspective?bulkDownload=thisPaper-topRelated-sameAuthor-citingThis-citedByThis-secondOrderCitations&from=cover_page](https://www.academia.edu/27064725/REST_API_Modeling_Languages_A_Developers_Perspective?bulkDownload=thisPaper-topRelated-sameAuthor-citingThis-citedByThis-secondOrderCitations&from=cover_page).

- Suryotrisongko, H. (2017) 'Arsitektur Microservice untuk Resiliensi Sistem Informasi', *Sisfo*, 06(02), pp. 231–246. doi: 10.24089/j.sisfo.2017.01.006.
- Suwita, F. S. (2020) 'Pengembangan Sistem Informasi Tugas Akhir dan Skripsi (SIMITA)', *Jurnal Teknologi dan Informasi (JATI)*, 10, pp. 71–82. doi: 10.34010/jati.v10i1.
- Tun, U. and Onn, H. (2020) 'International Journal of Advanced Trends in Computer Science and Engineering Available Online at [HTTP://www.warse.org/IJATCSE/static/pdf/file/ijatcse6491.12020.pdf](http://www.warse.org/IJATCSE/static/pdf/file/ijatcse6491.12020.pdf) JARI – Malaysian Sign Language Translator', 9(1).
- Wibawa, J. C. and F., M. R. (2017) 'Pengembangan Sistem Informasi Penjadwalan dan Manajemen Keuangan Kegiatan Seminar dan Sidang Skripsi/Tugas Akhir (Studi Kasus Program Studi Sistem Informasi UNIKOM)', *Jurnal Teknik Informatika dan Sistem Informasi*, 3(1), pp. 150–168. doi: 10.28932/jutisi.v3i1.585.
- Yuri Chandra Tri Putra, Thomas Adi Purnomo Sidi and Joseph Eric Samodra (2020) 'Implementasi Arsitektur Microservice pada Aplikasi Web Pengajaran Agama Islam Home Pesantren', *Jurnal Informatika Atma Jogja*, 1(1), pp. 88–97.
- Z.M, T. (2012) 'Analisis Prokrastinasi Tugas Akhir (Skripsi)', *formatif: jurnal ilmiah pendidikan MIPA*, 2(1), pp. 82–89. Available at: [HTTP://journal.lppmunindra.ac.id/index.php/Formatif/article/view/87/85%0A](http://journal.lppmunindra.ac.id/index.php/Formatif/article/view/87/85%0A).

Lampiran A