

PROPOSAL SKRIPSI

IMPLEMENTASI ARSITEKTUR MICROSERVICE DAN RESTFUL API PADA
PENGEMBANGAN SISTEM MANAJEMEN SKRIPSI



Disusun Oleh :

Ahmad Khairi Ramadan

170411100099

Dosen Pembimbing 1 :	Ach. Khozaimi, S.Kom., M.Kom.	19860926 201404 1 001
Dosen Pembimbing 2 :	Yoga Dwitya Pramudita, S.Kom., M.Cs.	19840413 200812 1 002

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS TRUNOJOYO MADURA

2020

LEMBAR PERSETUJUAN PROPOSAL

Nama : Ahmad Khairi Ramadan
NRP : 17.04.1.1.1.00099
Bidang Minat : Rekayasa Perangkat Lunak
Program Studi : Informatika
Jurusan : Teknik Informatika
Dosen Pembimbing : 1. Ach. Khozaimi, S.Kom., M.Kom.
2. Yoga Dwitya Pramudita, S.Kom., M.Cs.
Judul Tugas Akhir : Implementasi Arsitektur Microservice dan RESTful API pada
Pengembangan Sistem Manajemen Skripsi

Proposal ini telah disetujui di seminar pada

Tanggal. 2021

Dosen Pembimbing 1,

Dosen Pembimbing 2,

Ach. Khozaimi, S.Kom., M.Kom.

Yoga Dwitya Pramudita, S.Kom., M.Cs.

NIP. 19860926 201404 1 001

NIP. 19840413 200812 1 002

Mengetahui,

Menyetujui,

Ketua Program Studi

Kepala Laboratorium

Teknik Informatika,

Jurusan Teknik Informatika,

Yoga Dwitya Pramudita, S.Kom., M.Cs.

Devie Rosa Anamisa, S.Kom., M.MT.

NIP. 19840413 200812 1 002

NIP. 19841104 200812 2 003

ABSTRAK

Skripsi merupakan suatu karya yang wajib dikerjakan oleh mahasiswa dengan tujuan untuk mengembangkan ilmu yang telah dipelajari selama menempuh masa kuliah. Selama ini mahasiswa di Fakultas Teknik khususnya Teknik Informatika melaksanakan proses pembuatan skripsi secara manual. Selain banyak kertas dan tinta yang digunakan, mahasiswa juga mengalami banyak kesulitan lain pada pelaksanaannya. Sistem Manajemen Skripsi (SMS) merupakan perangkat lunak yang akan dibangun dengan tujuan memudahkan proses pelaksanaan skripsi sekaligus meminimalisir penggunaan kertas dalam proses pelaksanaan skripsi. pembangunan yang dilakukan secara monolitik seperti sistem pada umumnya menyebabkan kesulitan pada penanganan layanan. Maka dalam penelitian ini akan dibangun Sistem Manajemen Skripsi dengan pendekatan microservice untuk memudahkan penanganan pada setiap layanan sistem. Pada pengembangan ini akan memanfaatkan REST API sebagai perantara komunikasi antar layanan. Pengembangan sistem dengan pendekatan microservice akan dilakukan pengujian performa untuk memastikan kemampuan sistem untuk dapat digunakan untuk mengelola proses pelaksanaan skripsi di program studi Teknik Informatika Universitas Trunojoyo Madura dengan baik.

Kata Kunci : Sistem Manajemen Skripsi, REST API, Microservices.

DAFTAR ISI

LEMBAR PERSETUJUAN PROPOSAL	i
ABSTRAK	ii
DAFTAR ISI	iii
DAFTAR GAMBAR	v
DAFTAR TABEL	vi
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.2.1. Identifikasi Masalah	2
1.2.2. Usulan Solusi.....	3
1.2.3. Pertanyaan Penelitian	3
1.3. Tujuan dan Manfaat	3
1.3.1. Tujuan Penelitian.....	3
1.3.2. Manfaat.....	3
1.4. Batasan Masalah.....	3
1.5. Metodologi Penulisan Proposal.....	4
BAB II TEORI PENUNJANG.....	5
2.1. Skripsi.....	5
2.2. Microservice.....	5
2.3. Frontend dan Backend.....	7
2.4. RESTful API	8
2.5. Pengujian Performance	10
2.7. Penelitian Terkait	11
BAB III PERANCANGAN SISTEM	14
3.1. Tahapan Penelitian	14
3.1.1. Analisis Masalah	14

3.1.2. Studi Pustaka	15
3.1.3. Analisis dan perancangan sistem.....	15
3.1.4. Implementasi Sistem	15
3.1.5. Pengujian Sistem	15
3.1.6. Dokumentasi dan Penyusunan Laporan	15
3.2. Arsitektur Sistem.....	38
3.2. Analisis dan Perancangan Sistem.....	16
3.3.1. Rancangan Usecase Diagram	Error! Bookmark not defined.
3.3.2. Rancangan Activity Diagram	22
3.4. Skenario Pengujian.....	38
3.4.1. Pengujian Functionalitas	40
3.4.2. Pengujian Performa	41
3.5. Rencana Kegiatan.....	41
DAFTAR PUSTAKA	43

DAFTAR GAMBAR

Gambar 1 Komunikasi App dan Backend Service	6
Gambar 2 Komunikasi API Gateway	7
Gambar 3 Tahapan Penelitian	14
Gambar 4 Usecase Diagram Sistem Manajemen Skripsi	17
Gambar 5 Usecase Diagram Layanan User.....	18
Gambar 6 Usecase Diagram Layanan Diskusi	19
Gambar 7 Usecase Diagram Layanan Jadwal	20
Gambar 8 Usecase Diagram Layanan Skripsi.....	21
Gambar 9 Activity Diagram Login	23
Gambar 10 Activity Diagram Mengubah Data User.....	24
Gambar 11 Activity Diagram Menghapus Data User	25
Gambar 12 Activity Diagram Mengubah Profil.....	26
Gambar 13 Activity Diagram Penambahan Data Dosen/Mahasiswa.....	27
Gambar 14 Activity Diagram Menghapus Data Dosen/Mahasiswa	28
Gambar 15 Activity Diagram Membuat Thread	29
Gambar 16 Activity Diagram Mengomentari Thread	30
Gambar 17 Activity Diagram Mendaftar	31
Gambar 18 Activity Diagram Menentukan Jadwal	32
Gambar 19 Activity Diagram Menghapus Jadwal	33
Gambar 20 Activity Diagram Pengajuan Topik.....	34
Gambar 21 Activity Diagram Mengubah File Skripsi	35
Gambar 22 Activity Diagram Menetapkan Dosen Pembimbing	36
Gambar 23 Activity Diagram Validasi Topik dan Dosen Pembimbing.....	37
Gambar 24 Activity Diagram Penilaian	38
Gambar 25 Arsitektur Sistem.....	39
Gambar 26 Siklus Hidup Pengembangan Waterfall	40

DAFTAR TABEL

Tabel 1 Contoh Penggunaan Metode dalam RESTful API.....	9
Tabel 2 Batasan Waktu Respon dari Komputer	11
Table 3 Detail Usecase Layanan User.....	18
Tabel 4 Detail Usecase Layanan Diskusi	19
Tabel 5 Detail Usecase Layanan Jadwal	20
Tabel 6 Detail Usecase Layanan Skripsi.....	22
Tabel 7 Parameter Pengujian Performa.....	41
Tabel 8 Rencana Kegiatan.....	41

BAB I PENDAHULUAN

1.1. Latar Belakang

Skripsi merupakan karya tulis ilmiah yang berisikan paparan dari pembahasan hasil penelitian suatu masalah pada bidang ilmu tertentu. Skripsi juga harus dapat dipertanggungjawabkan secara ilmiah serta dikerjakan berdasarkan aturan atau tata cara yang telah ditentukan. Tujuan dari skripsi ini sendiri adalah agar mahasiswa mampu menulis dan menyusun suatu karya tulis ilmiah sesuai dengan bidangnya masing-masing.

Selama ini mahasiswa di Fakultas Teknik khususnya Prodi Teknik Informatika melaksanakan proses pembuatan skripsi secara manual, baik dalam pelaksanaannya maupun pengelolaan data/berkas skripsi. Proses pelaksanaan skripsi secara manual ini sangat memakan waktu, tenaga, biaya dan banyak sekali menghabiskan kertas baik untuk berkas-berkas persyaratan skripsi ataupun cetak proposal dan skripsi itu sendiri. Pada pelaksanaannya mahasiswa perlu menemui pihak-pihak seperti koordinator, dosen pembimbing dan dosen penguji untuk dapat melengkapi berkas-berkas skripsi hal ini sangat memakan tenaga baik dari sisi mahasiswa ataupun dosen yang bersangkutan.

Beberapa peneliti sebelumnya telah berhasil membangun sistem yang bertujuan untuk meminimalisir permasalahan-permasalahan yang terjadi pada pelaksanaan skripsi. Di tahun 2010 Ach. Khozaimi, Firdaus Solihin dan Achmad Jauhari telah berhasil mengembangkan Sistem Informasi Tugas Akhir (SIMTAK) [1]. Penelitian ini berfokus pada pembangunan sistem yang dapat memanajemen, monitoring dan pencatatan proses skripsi serta pemberian informasi terkait skripsi. Sistem ini dikembangkan kembali pada tahun 2019 oleh Ach. Khozaimi, Sigit Susanto Putro dan Mujibur Rohman [2]. Pada penelitian tersebut mereka menerapkan metode komputasi berupa cosine similarity untuk mengurangi tingkat kesamaan judul maupun metode yang digunakan mahasiswa Dalam pelaksanaan skripsi.

Sistem yang telah dibangun pada penelitian sebelumnya menggunakan arsitektur monolitik dimana setiap layanan di dalam sistem memiliki keterkaitan yang tinggi sehingga ketika terjadi perubahan pada suatu layanan dapat mempengaruhi layanan lain. Oleh karena itu perlu dibangun sebuah sistem yang menerapkan arsitektur microservice dimana setiap layanan akan bekerja secara mandiri sehingga dapat memudahkan pengembangan lanjutan dalam membenahi sistem. Pembenahan sistem cukup dilakukan pada setiap layanan yang bermasalah tanpa perlu mengganggu kinerja layanan lain [3].

Pengembangan arsitektur monolitik adalah suatu cara pengembangan sistem dimana keseluruhan kompoen dalam sistem dikemas bersama[3]. Pada arsitektur monolitik setiap komponen atau layanan memiliki keterkaitan yang sangat tinggi menjadikan arsitektur ini sulit untuk dibenahi karena pembenahan pada suatu komponen akan mempengaruhi komponen lain. Berbeda dengan arsitektur microservice. Dalam arsitektur microservice setiap layanan dalam suatu sistem akan berdiri sendiri. sehingga ketika terjadi suatu masalah pada suatu layanan tidak akan mempengaruhi layanan lain[3]. Dalam penerapan arsitektur microservice yang setiap layanannya terpisah diperlukan suatu perantara web service[4].

Web service terbagi menjadi SOAP dan REST. SOAP merupakan sebuah XML-based mark-up language yang digunakan untuk pertukaran data object dalam network. Pengiriman data menggunakan SOAP tidak maksimal untuk data berskala besar[4]. selain itu penggunaan XML juga memiliki format yang lebih besar dibandingkan dengan JSON pada penerapan komunikasi REST. REST adalah gaya arsitektur yang sangat umum digunakan saat ini [5][6] dimana untuk proses komunikasinya menggunakan JSON. API yang mengikuti gaya REST disebut RESTful API. REST juga merupakan pendekatan praktis dalam pengembangan aplikasi web [6]. REST sangat cocok untuk digunakan berkomunikasi antar sistem yang bekerja secara independent seperti microservice.

Sistem Manajemen Skripsi (SMS) merupakan sistem yang dibangun dengan tujuan mempermudah Proses pelaksanaan skripsi dan mengubah proses pelaksanaan skripsi yang semula *Base on Paper* menjadi *Paperless* sehingga penggunaan kertas dapat terminimalisir. Pengembangan sistem manajemen skripsi akan dilakukan dengan menerapkan arsitektur microservice dengan komunikasi RESTful API agar pembenahan lanjutan dapat lebih mudah dilakukan. Hasil pengembangan sistem manajemen skripsi ini akan diuji dengan secara fungsionalitas untuk memastikan fungsi dari setiap layanan sudah berjalan dengan baik dan pengujian secara performa untuk memastikan sistem dapat bekerja dengan baik dibawah beban kerja tertentu [7].

1.2. Rumusan Masalah

1.2.1. Identifikasi Masalah

Proses pelaksanaan skripsi yang dilakukan secara manual kurang efektif baik secara pendataan ataupun pelaksanaan itu sendiri. Maka pembangunan suatu sistem untuk dapat memanajemen proses pelaksanaan skripsi sangat diperlukan.

1.2.2. Usulan Solusi

Solusi untuk permasalahan ini adalah membangun sistem manajemen skripsi menggunakan arsitektur *microservice* dengan REST API sebagai komunikasi antar layanan yang dibentuk.

1.2.3. Pertanyaan Penelitian

Pertanyaan yang akan dibahas dalam penelitian ini adalah :

1. Berapa banyak *microservice* yang perlu dibangun pada pengembangan sistem manajemen skripsi?
2. Bagaimana pengaruh banyaknya request terhadap kinerja sistem manajemen skripsi yang telah dibangun berdasarkan penggunaan RAM, penggunaan CPU dan *response time*?

1.3. Tujuan dan Manfaat

1.3.1. Tujuan Penelitian

Tujuan dilakukannya penelitian ini antara lain :

1. Merancang dan membangun sistem manajemen skripsi agar dapat memanajemen proses pelaksanaan skripsi dengan baik.
2. Menerapkan arsitektur *microservice* dan REST API pada sistem manajemen skripsi.
3. Mengetahui pengaruh banyaknya request terhadap kinerja sistem manajemen skripsi yang dibangun dengan arsitektur *microservice* dari segi penggunaan RAM, penggunaan CPU dan *response time*.

1.3.2. Manfaat

Manfaat yang diharapkan setelah sistem ini dibangun adalah :

1. Proses pelaksanaan skripsi dapat dilaksanakan secara online melalui sistem.
2. Pembenahan sistem dapat menjadi lebih mudah dengan penerapan arsitektur *microservice*.
3. bisa dijadikan penelitian lebih lanjut terhadap sistem yang telah dikembangkan.

1.4. Batasan Masalah

Berdasarkan permasalahan yang telah di rumuskan maka masalah dalam penelitian ini akan dibatasi hingga meliputi :

1. Sistem akan dibangun dengan pendekatan *microservice*
2. Komunikasi antar layanan akan menggunakan Representational State Transfer (REST)
3. Pengujian sistem yang dilakukan adalah functional testing dan performance testing.

1.5. Metodologi Penulisan Proposal

Adapun sistematika penulisan proposal skripsi ini adalah sebagai berikut :

BAB I	PENDAHULUAN	Bab ini berisikan paparan latar belakang, perumusan masalah, tujuan dan manfaat, penelitian, batasan-batasan penelitian dan metodologi penulisan proposal/skripsi.
BAB II	TEORI PENUNJANG	Bab ini membahas tentang landasan teori yang mendasari penelitian ini.
BAB III	PERANCANGAN SISTEM	Pada bab ini akan dipaparkan hasil perancangan yang dilakukan untuk penelitian ini.

BAB II TINJAUAN PUSTAKA

2.1. Skripsi

Tugas akhir yang sering disebut juga dengan skripsi merupakan suatu karya tulis ilmiah yang wajib disusun oleh mahasiswa untuk menyelesaikan studinya di program sarjana. Tugas akhir ini memiliki tujuan agar mahasiswa dapat mengaplikasikan pengalaman yang di dapatkan selama studi sehingga dapat dipergunakan untuk memecahkan suatu masalah dalam bidang studi tertentu yang disusun dalam sebuah karya tulis yang sistematis, logis, kritis, kreatif dan didasarkan pada data/informasi yang akurat serta di dukung dengan analisis yang tepat. Skripsi disusun berdasarkan hasil penelitian dengan ciri-ciri sebagai berikut [2] :

1. Harus ada permasalahan
2. Judul skripsi dipilih sendiri oleh mahasiswa atau ditentukan oleh dosen pembimbing
3. Didasarkan pada pengamatan lapangan (data primer) dan/atau analisis data sekunder
4. Harus ada ketertiban metodologi
5. Mengungkapkan adanya kenyataan baru atau kenyataan khusus
6. Dibawah bimbingan secara berkala dan teratur oleh dosen pembimbing
7. Mengikuti tata tulis karya ilmiah
8. Dilengkapi dengan abstrak
9. Dipresentasikan pada forum seminar
10. Dipertahankan dalam ujian siding lisan di depan tim dosen penguji

Mahasiswa diharapkan dapat mengaplikasikan semua pengalaman pendidikan dalam memecahkan masalah di bidang tertentu dan di tuangkan dalam penulisan dan penyusunan skripsi masing-masing.

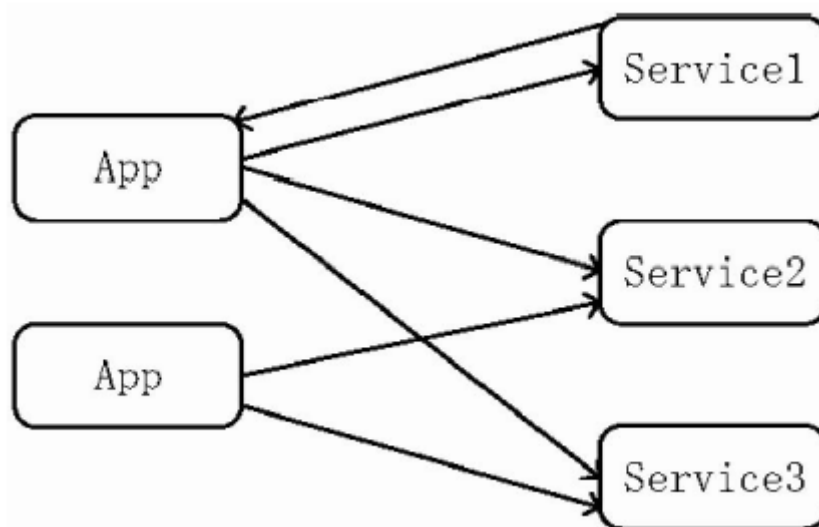
2.2. Microservice

Microservice merupakan sebuah pendekatan baru dalam mengimplementasikan sistem berorientasi layanan. Microservice membagi suatu sistem yang kompleks menjadi kumpulan layanan-layanan kecil yang berdiri sendiri [8]. Dengan pendekatan arsitektur microservice ini memungkinkan pengembangan setiap layanan secara terpisah hal ini dapat mempercepat pembangunan sistem. Dalam penerapan arsitektur microservice juga memungkinkan penggunaan teknologi yang berbeda-beda pada setiap microservice [9]. Pada arsitektur pengembangan perangkat lunak ini ditekankan pada pemecahan layanan sistem menjadi beberapa layanan sesuai dengan lingkup kebutuhan. Kemudahan yang didapatkan pada penggunaan microservice adalah ketika terjadi masalah pada salah satu layanan maka tidak akan mempengaruhi layanan lain. Selain

microservice terdapat juga layanan lain seperti monolithic. Arsitektur monolithic adalah sebuah cara mengembangkan perangkat lunak dimana semua komponen bisnis berada dalam suatu sistem yang dikemas bersama [3].

Karakteristik dan keunggulan dari arsitektur microservice adalah dapat menggunakan bermacam-macam teknologi dan dapat dengan mudah mengganti teknologi yang digunakan, ketahanan, skalabilitas dan deployment yang independent, berorientasi organisasi yang ditargetkan [9]. Untuk suatu aplikasi yang membutuhkan concurrency (konkurensi) dan capacity (kapasitas) yang tinggi maka arsitektur layanan mikro adalah pilihan yang baik. Arsitektur microservice memberikan solusi yang efektif dalam memecahkan masalah seperti proyek team yang besar, kompleks dan mengkonsumsi waktu dalam iterasi pembaruan perangkat lunak, dan perawatan yang sulit pada integrasi dan rilis aplikasi besar [3].

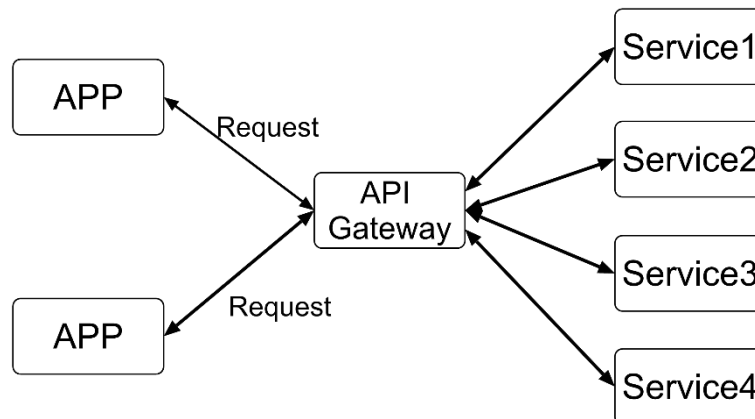
Pola komunikasi di dalam arsitektur microservices terbagi menjadi 2 bagian. Komunikasi aplikasi dan backend service seperti yang terlihat pada gambar 1. *Backend service* disini memiliki fungsi dan tugasnya masing masing. Aplikasi adalah *consumer* yang memanggil *backend service* seperti aplikasi *mobile* dan *web service client*.



Gambar 1 Komunikasi App dan Backend Service

Memanggil secara langsung terhadap backend service merupakan cara yang sangat fleksibel. Namun, hal ini memungkinkan aplikasi untuk memanggil bermacam-macam *backend service* untuk suatu *request* yang dapat menyebabkan *delay* untuk *remote calls*. *Backend service* biasanya didasarkan pada arsitektur microservice. Pembagian layanan dipengaruhi oleh perkembangan bisnis. Hal ini menyebabkan semua tipe aplikasi perlu dilakukan perbaikan untuk dapat beradaptasi dengan perubahan layanan.

Pola komunikasi lain dalam arsitektur *microservice* adalah komunikasi menggunakan API *gateway* yang dapat dilihat pada gambar 2.



Gambar 2 Komunikasi API Gateway

Pada pola komunikasi ini aplikasi dan setiap layanan perlu mengakses API *gateway* untuk berkomunikasi. Setiap *backend service* akan didaftarkan ke dalam API *gateway*. API *gateway* akan menerima request dari aplikasi dan mengarahkan pada layanan yang dituju [8]. Setiap request harus melalui API *gateway* sehingga setiap aplikasi tidak dapat secara langsung berkomunikasi dengan setiap layanan.

Pada proses komunikasi dalam arsitektur *microservice* terdapat suatu manajemen sesi agar server dapat mengenali user mana yang melakukan request. Berbeda dengan program-program monolitik pada komunikasi *microservice* ini manajemen sesi biasanya menggunakan token sebagai identitas dari user yang mana token ini dibawa oleh *cookie*. Token akan disimpan dalam database ketika user telah melakukan login kemudian akan dikirimkan ke *cookie* pada browser. Setiap request yang dilakukan akan mengambil token yang ada pada *cookie*. Dan server akan memastikan bahwa token yang dikirimkan sudah benar atau belum untuk dapat melanjutkan request[10].

2.3. Frontend dan Backend

Dengan perkembangan teknologi saat ini menciptakan suatu pembagian bentuk pada pengembangan perangkat lunak menjadi *frontend* dan *backend*. Frontend adalah bagian dari suatu website yang menjadi perantara antara interaksi pengguna dengan sistem [6]. Bagian ini biasanya menyajikan tampilan yang interaktif dan menarik untuk digunakan pengguna. Dibalik tampilan yang indah pada *frontend* terdapat juga bagian *backend* dimana bagian ini bekerja untuk menompong semua fitur yang ada dalam suatu sistem sehingga dapat berjalan dengan baik.

Backend adalah suatu ruang logis dengan fungsionalitas dan operasi dari aplikasi perangkat lunak atau sistem informasi [6]. *Backend* adalah sebuah program yang berjalan pada sisi server

dan dapat berinteraksi secara langsung dengan database [11]. Penggunaan REST-API sebagai *backend* layanan web sangat memungkinkan suatu layanan web untuk dapat diakses tanpa batasan bahasa, lingkungan ataupun platform dari sisi *frontend* [11]. Sistem backend memastikan data atau layanan yang diminta dan dikirim oleh sistem atau aplikasi frontend dikirimkan melalui metode yang diprogram. Backend terdiri dari logika aplikasi inti, database, integrasi data dan aplikasi, API, dan proses backend lainnya[6].

2.4. RESTful API

API (Application Program Interface) adalah antarmuka pemrograman aplikasi yang menyediakan suatu cara sehingga aplikasi dapat berhubungan dan berintegrasi antara satu aplikasi dengan aplikasi lain [12]. API disini memiliki peran sebagai perantara antar berbagai aplikasi yang berbeda-beda. Antarmuka pemrograman aplikasi juga dapat digunakan untuk berkomunikasi dengan berbagai macam bahasa pemrograman yang berbeda-beda. antarmuka ini dapat dijadikan sebuah dokumentasi perangkat lunak yang menjelaskan layanan-layanan yang ada dalam sebuah aplikasi atau sistem sehingga penggunaan API ini dapat mempermudah pengembang untuk mempelajari sistem juga sebagai panduan untuk penggunaan fitur-fitur yang disediakan [13].

REST (Representational State Transfer) merupakan model arsitektur perangkat lunak yang sangat umum digunakan untuk sebuah sistem terdistribusi yang fokus pada skalabilitas dan interaksi antar komponen dalam sistem dan generalisasi interface [5]. REST memisahkan perhatian dari gaya client-server tanpa masalah skalabilitas server dan memungkinkan informasi untuk bersembunyi melalui antarmuka generik untuk mengaktifkan enkapsulasi dan evolusi layanan [14]. REST merupakan pendekatan praktis dalam pengembangan aplikasi web dimana sistem pengembang perlu ditingkatkan atau membutuhkan cara yang lebih sederhana dalam berkomunikasi dengan sistem yang independen. REST tidak memiliki state dan berorientasi data. Semua permintaan bersifat independen dan server tidak menyimpan status permintaan apapun [6].

REST adalah salah satu bentuk penerapan web service. Web Service sendiri merupakan suatu bentuk sistem perangkat lunak yang dirancang untuk dapat berinteraksi antar mesin dalam suatu jaringan. Webservice memiliki interface yang dibuat dalam format yang dapat dibaca oleh mesin [4]. Web service mudah dibangun baik secara native ataupun dengan memanfaatkan framework. Pertukaran data dalam web service dilakukan dan disimpan dalam format XML/JSON karena format tersebut berbasis teks dan dapat dibaca dengan baik pada bahasa pemrograman dan sistem operasi yang berbeda-beda [15][16].

Selain REST ada juga SOAP yang merupakan sebuah XML-based mark-up language yang digunakan untuk melakukan pertukaran data objek dalam suatu jaringan. Aspek komunikasi yang didefinisikan dalam SOAP adalah : *Message Envelope*, *Encoding*, *RPC call convention*, dan bagaimana menyatukan sebuah message didalam protocol transport [15]. Sedangkan REST bekerja dengan konsep perpindahan state. REST berinteraksi melalui HTTP (Hypertext Transfer Protocol) dimana dalam prosesnya memanfaatkan file XML atau JSON. Untuk enkripsi dan integrasi, REST menggunakan SSL (*Secure Sockets Layer*) dan TLS (*Transport Layer Security*). Server dalam penerapan REST tidak akan menyimpan request apapun karena sifatnya yang independen [17][6].

Application Program Interface (API) yang mengikuti gaya REST disebut dengan RESTful API yang mana memanfaatkan *Uniform Resource Identifier* (URI) untuk merepresentasikan data [6]. RESTful API juga merupakan suatu sistem terdistribusi yang berfokus pada skalabilitas dan interaksi antar komponen yang didokumentasikan dengan metode API sehingga pengembang menjadi lebih mudah dalam mempelajari sistem yang berjalan serta dapat dijadikan sebagai panduan untuk menggunakan fitur-fitur yang disediakan. RESTful API memanfaatkan URI (Uniform Resource Identifier) yang digunakan untuk mengidentifikasi sumberdaya. Metode-metode yang digunakan di dalam REST antara lain [17][6]:

- GET untuk mendapatkan sumberdaya
- POST untuk membuat sumberdaya baru
- PUT untuk memperbarui sumberdaya berdasarkan id sumber daya
- DELETE digunakan untuk menghapus sumber daya atau kumpulan sumber daya

Contoh penggunaan metode dalam RESTful API dapat dilihat pada tabel berikut.

Tabel 1 Contoh Penggunaan Metode dalam RESTful API

Resource	Method			
	GET	POST	PUT	DELETE
/api/student	Mendapatkan list dari semua siswa	Membuat list baru dari siswa	Memperbarui list siswa	Menghapus semua siswa

/api/student/1	Mendapatkan siswa berdasarkan id siswa	Diasumsikan sebagai koleksi. Membuat siswa baru di dalamnya.	Jika siswa tersedia maka perbarui siswa tersebut. Jika tidak tersedia maka membuat siswa baru.	Menghapus siswa dengan id tersebut.
----------------	--	--	--	-------------------------------------

2.5. Pengujian Performance

Pengujian Merupakan tahapan dari pengembangan sistem yang tujuannya untuk memastikan software/aplikasi yang telah dibuat bisa berjalan sesuai dengan kebutuhan yang diharapkan. *Performance testing* (pengujian kinerja) merupakan jenis pengujian untuk memastikan perangkat lunak yang telah dibuat mampu bekerja dengan baik dibawah beban kerja yang diharapkan [7]. Pengujian kinerja dapat dilakukan pada berbagai macam tipe aplikasi perangkat lunak seperti aplikasi mobile, aplikasi web, layanan web, dll [18]. Pada umumnya pengujian kinerja akan mengarah pada kemampuan perangkat lunak dalam menyelesaikan suatu proses pada interaksi multi-user yang tinggi [18].

Terdapat beberapa tools yang dapat digunakan untuk menguji performa seperti SOAPUI dan APACHE JMETER. Kedua tools ini gratis dan dapat diakses pada website masing-masing. SOAPUI menawarkan beberapa fitur seperti *load testing*, *regression testing*, *security testing*, dll. SOAPUI berbasis GUI sehingga akan lebih mudah untuk digunakan. JMETER adalah sebuah tool dari apache yang biasa digunakan untuk menguji kinerja pada aplikasi web. Pengujian pada JMETER dilakukan berdasarkan test plan dengan menentukan thread group, request web service, loop dan deskripsi dari hasil testplan. JMETER juga menyediakan fitur remote start dimana kita dapat melakukan pengujian pada dari beberapa mesin. Umumnya jika request yang dilakukan datang dari mesin yang sama maka *webserver* akan menganggapnya sebagai *sequentially*. Namun, jika ipnya berbeda maka akan disebut sebagai *parallel way*. Ketika request yang dilakukan datang dari mesin yang berbeda-beda maka akan lebih baik menggunakan multiple Jmeter [18].

Pengujian kinerja tergolong pada pengujian non-functional testing. Pengujian ini memiliki peran penting ketika layanan bertambah dan digunakan oleh banyak user dalam suatu waktu. Pengujian kinerja adalah tugas yang cukup sulit sehingga kita memerlukan bantuan tools seperti apache Jmeter dan soapUI. Tanpa tools kita akan sangat kesulitan dalam menerapkan pengujian

ini [18]. Terdapat tiga jenis pengujian performa dalam sistem yaitu pengujian response time, penggunaan CPU dan penggunaan RAM (memori). Pada pengujian response time terdapat sebuah interpretasi mengenai batasan waktu respon [19].

Tabel 2 Batasan Waktu Respon dari Komputer

No	Waktu	Respon
1	0.1 detik	Pengguna menerima respon dari perintah yang dijalankan
2	1.0 detik	Batasan dari pengguna berfikir untuk menunggu <i>feedback</i> dari sistem
3	10 detik	Batasan akhir perhatian pengguna untuk menunggu sistem

Dari tabel 2 menunjukkan response time maksimal sistem yang baik adalah 10 detik. Semakin kecil atau semakin cepat rata-rata waktu yang dibutuhkan untuk suatu proses maka semakin baik aplikasi yang dikembangkan.

2.7. Penelitian Terkait

Pengembangan aplikasi dengan tujuan untuk manajemen tugas akhir atau skripsi telah dibahas di tahun 2010 oleh Ach. Khozaimi, bersama rekan-rekannya [1] dengan tujuan membangun sistem informasi tugas akhir (SIMTAK) agar dapat manajemen tugas akhir serta melakukan pencatatan untuk setiap kegiatan yang dilakukan mahasiswa dalam proses pelaksanaan tugas akhirnya. Penelitian ini dilanjutkan pada tahun 2019 [2] dengan menambahkan cosine similarity untuk menghindari kesamaan dokumen. Pengembangan sistem informasi tugas akhir (SIMTAK) ini yang dijadikan dasar pada pengembangan sistem manajemen skripsi (SMS).

Pengembangan aplikasi dengan pendekatan REST API dilakukan oleh Febianto Arifien, Sutarno dan Marti Riastuti di tahun 2019 [4]. Penelitian ini membahas permasalahan yang muncul yaitu sistem informasi perguruan tinggi yang semula dikembangkan dengan pendekatan SOAP mendapat masalah “maximum execution time” pada pemrosesan data besar. Sehingga dalam penelitian tersebut dibahas pengembangan perangkat lunak dengan pendekatan REST API. Berdasarkan hasil penelitian, pengembangan dengan REST-API terbukti berhasil mengintegrasikan interoperabilitas secara efektif dan berhasil mengatasi error “Maximum Execution Time” berdasarkan hasil pengujian kecepatan dan ukuran file yang telah dilakukan. Penelitian ini yang menjadi dasar atas digunakannya komunikasi REST dalam sistem yang akan digunakan ditinjau dengan efektifitasnya dalam integrasi antar komponen dalam sistem.

Penelitian penerapan RESTful API pada pengembangan sistem dilakukan oleh Bayu Wijaya Putra bersama rekan-rekannya di tahun 2019 [20] pada Sistem Informasi Manajemen Tugas Akhir.

Penelitian lain juga dilakukan oleh Sidik Priyo Utomo bersama dengan rekan-rekannya di tahun 2020 [16]. Dalam penelitian tersebut mereka membahas tentang pengembangan sistem dengan mengkombinasikan framework PHP CodeIgniter dengan menerapkan arsitektur RESTful API. Hal ini akan dijadikan dasar pembangunan sistem manajemen skripsi yang memanfaatkan framework CodeIgniter didalamnya serta dapat membantu pada penerapan RESTful API sebagai komunikasi antar komponen yang ada di dalam sistem ini.

Sebuah penelitian terkait dengan pengembangan arsitektur microservice dilakukan oleh Yuri Chandra Tri Putra, bersama rekan-rekannya pada tahun 2020 [21]. Penelitian tersebut ditujukan untuk mengatasi permasalahan skalabilitas yang muncul pada aplikasi Home Pesantren yang telah dibangun secara monolitik dengan melakukan pengkodean sistem dan menerapkan basis data secara terpisah antar microservice. Penerapan microservice menggunakan komunikasi REST API untuk mendukung *synchronous communication* dan RabbitMQ untuk mendukung *asynchronous communication*. Penelitian ini yang menjadi dasar penerapan microservice pada sistem manajemen skripsi yang nantinya juga akan menggunakan komunikasi REST API.

Penelitian terkait pengujian performa sistem dilakukan Pada tahun 2018 oleh Xian Jun Hong bersama rekan-rekannya [22]. Dalam penelitian ini dilakukan pengujian performa pada penerapan RESTful API dan RabbitMQ untuk microservice aplikasi web. Pengujian performa dilakukan terhadap kedua metode tersebut dibawah berbagai jumlah pengguna untuk membandingkan dan mengevaluasi kinerja keduanya pada keadaan yang berbeda-beda. Penelitian ini mendasari tahapan pengujian sistem dalam hal performa dengan menguji request terhadap sistem dibawah berbagai jumlah pengguna.

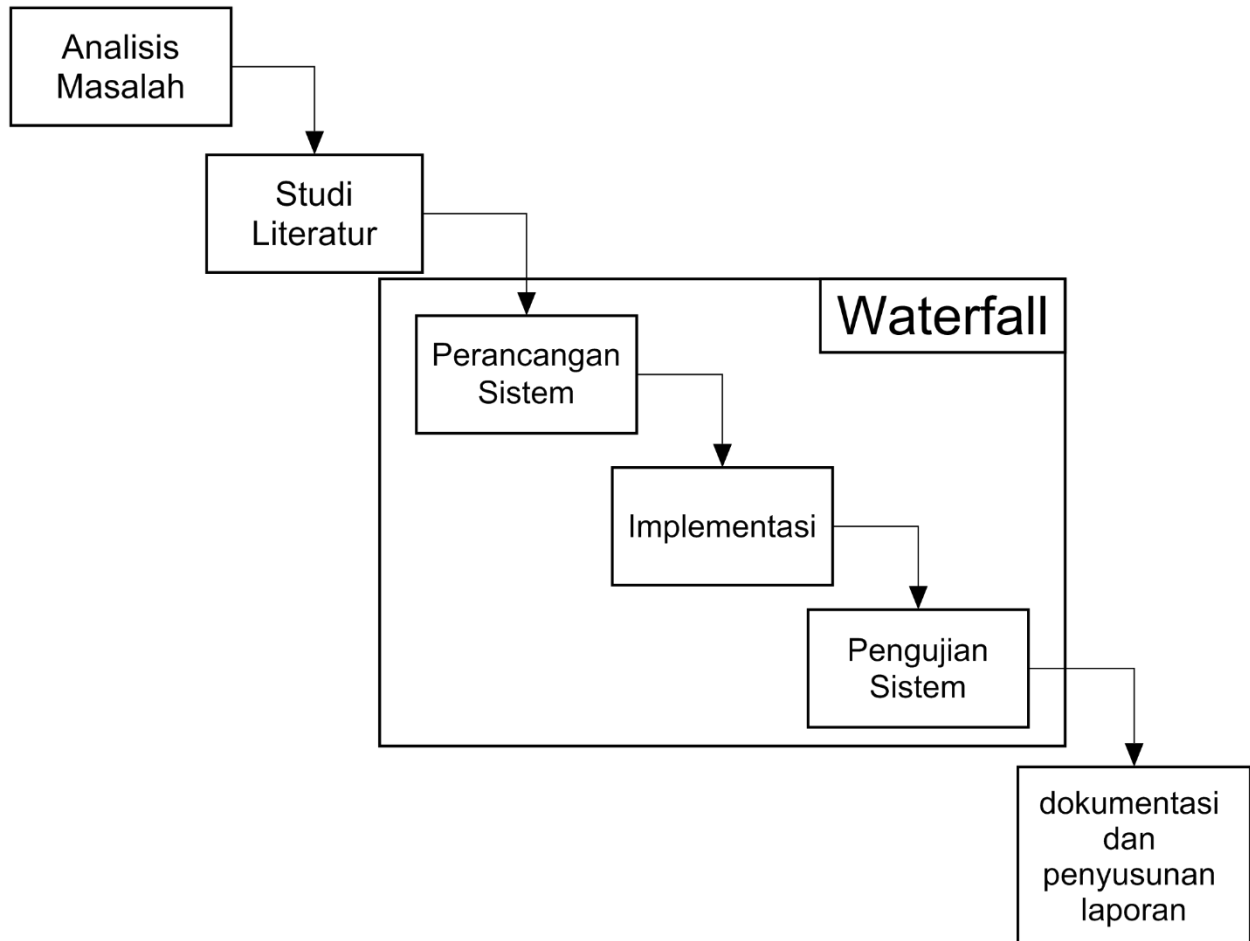
Peneliti, tahun	Metode/Kasus	Keunggulan	Kelemahan
Ach. Khozaimi, Firdaus Solihin dan Achmad Jauhari, 2010	Membangun Sistem Informasi Tugas Akhir (SIMTAK) di Program Studi Teknik Informatika, Fakultas Teknik Universitas Trunojoyo Madura	Sistem yang telah dibangun dapat membantu semua pihak yang terlibat dalam pemberian informasi dan pengarsipan document tugas akhir.	Berfokus pada perancangan dan pembuatan sistem, tidak ada pengujian sistem yang dilakukan
Ach. Khozaimi, Sigit Susanto Putro dan Mujibur Rohman, 2019	Pengembangan Aplikasi Manajemen Tugas Skripsi di Program Studi Teknik	Sistem yang telah dibuat cukup efektif dengan rata-rata hasil uji experience	Dalam perhitungan <i>cosine similarity</i> memakan waktu yang lama.

	Informatika Universitas Trunojoyo Madura)	sebesar 8,82 dari range 1 sampai 10	
Febianto Arifien, Sutarno dan Marti Riastuti, 2019	Model Interoperabilitas Web Service Feeder PDDIKTI Menggunakan Enterprise Javabeans (EJB) dan REST-API	Pengembangan aplikasi dengan pendekatan REST API berhasil mengatasi error “Maximum Execution Time” berdasarkan hasil kecepatan dan ukuran file pada Aplikasi yang sebelumnya SOAP XML	Kecepatan proses menggunakan REST api tidak sepenuhnya lebih baik dibandingkan dengan penggunaan SOAP.
Bayu Wijaya Putra, Ariansyah Saputra, Rudi Sanjaya dan Dedy Kurniawan, 2019	Implementasi Framework CodeIgniter dan Restful API pada Sistem Informasi Manajemen Tugas Akhir	Nilai SUS dari hasil pengembangan sistem mencapai 80,88	Tidak dilakukan pengujian lain selain metode SUS
Sidik Priyo Utomo, Nurul Hanifatul Alfiyah, Zulfi Achsan Sani, Mukhtar Hanafi dan Ardhin Primadewi, 2020	Perancangan RESTful Web Service Pada Sistem Informasi Terintegrasi Menggunakan FrameWork CodeIgniter	Didapatkan hasil pengujian <i>responsetime</i> yang berbanding lurus dengan banyaknya data. Arsitektur REST dimanfaatkan dengan baik dalam pengembangan sistem	Pengujian hanya dilakukan pada pertukaran data dalam database.
Yuri Chandra Tri Putra, Thomas Adi Purnomo Sidi dan Joseph Eric Samodra, 2020	Implementasi Arsitektur Microservice pada Aplikasi Web Pengajaran Agama Islam Home Pesantren	Penerapan arsitektur meicroservice dapat meningkatkan kemudahan skalabilitas aplikasi.	Belum dilakukan monitoring terhadap banyaknya API request yang dilakukan client.
Hong, Xian Jun Sik Yang, Hyun Kim, Young Han, 2018	Perbandingan RESTful API dan RabbitMQ secara performa sistem	Didapatkan penggunaan RESTful API pada microservice memiliki respon performa yang relatif baik	Belum ditambahkan situasi dan faktor yang sebenarnya terjadi untuk mendapatkan hasil uji yang lebih akurat

BAB III PERANCANGAN SISTEM

3.1. Tahapan Penelitian

Tahapan penelitian yang akan dilakukan dapat dilihat pada gambar 4.



Gambar 3 Tahapan Penelitian

Berdasarkan gambar 4. tahapan penelitian dimulai dengan analisis masalah yaitu tahap dimana permasalahan akan di definisikan, kemudian akan dilanjutkan studi pustaka dengan pengembangan sistem dengan metode waterfall yang diawali dengan analisis dan perancangan sistem kemudian implementasi, pengujian. Kemudian yang terakhir adalah dokumentasi dan penyusunan laporan.

3.1.1. Analisis Masalah

Berdasarkan hasil analisis yang telah dilakukan maka dapat ditarik kesimpulan bahwa pelaksanaan skripsi yang dilakukan secara manual kurang efisien sehingga perlu dilakukan pengembangan sistem manajemen skripsi berbasis web dengan tujuan untuk dapat

membantu manajemen dan monitoring proses pelaksanaan skripsi untuk mengatasi permasalahan skripsi sekaligus meminimalisir banyaknya penggunaan kertas pada proses pelaksanaan skripsi. Untuk mempermudah pembenahan layanan di dalam sistem, sistem manajemen skripsi perlu dikembangkan dengan pendekatan *microservice*. Dimana setiap layanan akan bekerja secara mandiri dan perbaikan dapat dilakukan pada setiap layanan kecil yang dibuat tanpa mempengaruhi layanan lain.

Namun terdapat beberapa faktor yang perlu diukur dalam penerapan *microservice* ke dalam sistem manajemen skripsi. Sistem yang akan dikembangkan perlu diukur secara fungsional dan secara performa sistem baik secara *response time*, penggunaan memori dan penggunaan CPU dalam menanggapi request dari client.

3.1.2. Studi Pustaka

Pada tahap ini semua informasi yang dibutuhkan dikumpulkan dan dipelajari, informasi berupa materi pengembangan sistem, penerapan *microservice*, penerapan REST API dan beberapa materi terkait pengembangan sistem menggunakan arsitektur *microservice*.

3.1.3. Analisis dan perancangan sistem

Pada tahap ini akan dilakukan analisis terhadap kebutuhan untuk pembangunan sistem dan dilanjutkan dengan merancang sistem yang akan dibangun dari hasil analisis yang telah dilakukan.

3.1.4. Implementasi Sistem

Di tahap ini akan dilakukan implementasi hasil dari analisis dan perancangan sistem untuk membangun sistem manajemen skripsi dengan pendekatan arsitektur *microservice* menggunakan *webservice REST API*.

3.1.5. Pengujian Sistem

Pada tahap ini sistem manajemen skripsi yang telah dibangun akan di uji untuk memastikan sistem telah siap atau belum sistem untuk digunakan. Pengujian yang dilakukan didasarkan pada skenario uji yang ditetapkan.

3.1.6. Dokumentasi dan Penyusunan Laporan

Dokumentasi dan penyusunan laporan akan dilakukan selama penelitian berlangsung mulai dari tahap studi pustaka sampai tahap pengujian sistem.

3.2. Analisis Kebutuhan

Pada bagian ini akan dijelaskan persyaratan dan kebutuhan sistem yang akan dijadikan dasar dari pengembangan sistem yang akan dibangun. Analisis yang akan dijelaskan pada sub bab ini adalah analisis kebutuhan fungsional, usecase diagram dan activity diagram.

3.2.1. Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional merupakan kebutuhan yang berisikan penjelasan mengenai fitur-fitur yang harus diterapkan dalam pengembangan sistem manajemen skripsi nantinya. Kebutuhan fungsional ini juga berisi informasi-informasi yang akan dihasilkan oleh sistem. Berikut kebutuhan fungsional dari sistem manajemen skripsi.

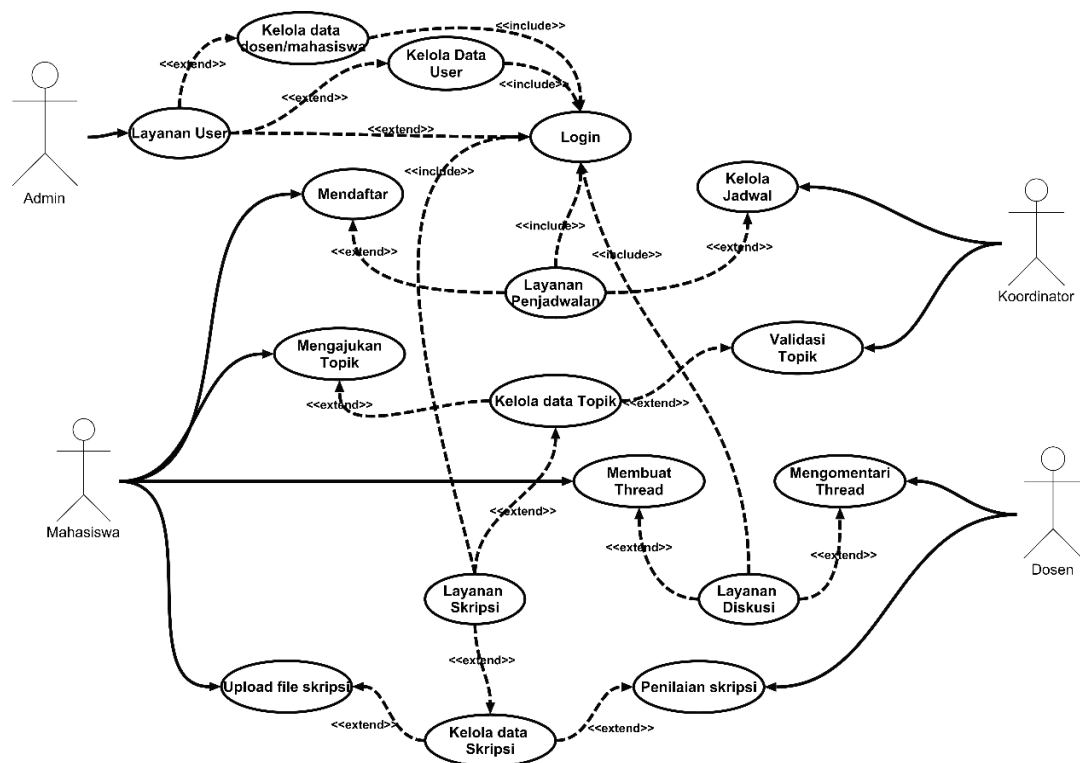
1. Sistem ini membutuhkan fitur login agar sistem dapat mengenali user yang saat ini sedang melakukan interaksi. Login ini dilakukan dengan mencocokkan inputan username dan password dari user dengan data dalam database dan user akan dialihkan ke halaman yang sesuai dengan identitas user.
2. Dibutuhkan fitur untuk melakukan pengolahan data user seperti menambah, mengubah, termasuk juga melihat data user yang hanya dapat dilakukan oleh administrator.
3. Administrator juga perlu diberi fitur untuk mengolah data dosen/mahasiswa seperti menambah, menghapus, mengubah dan melihat data dosen/mahasiswa. Penambahan dan penghapusan data dosen/mahasiswa akan mempengaruhi data user.
4. Untuk mahasiswa diperlukan fitur untuk mengajukan topik dan dosen pembimbing yang diinginkan.
5. Dibutuhkan sebuah fitur pada mahasiswa untuk mengupload dan mengubah file/berkas skripsi.
6. Koordinator prodi perlu melakukan pengecekan terhadap topik dan dosen pembimbing yang diajukan oleh mahasiswa karena dosen pembimbing memiliki kuota untuk mahasiswa bimbingan. Sehingga dibutuhkan fitur validasi terhadap topik dan penetapan dosen pembimbing.
7. Untuk mengatasi pelaksanaan bimbingan dibutuhkan sebuah fitur diskusi dimana dosen dan mahasiswa dapat membuat dan mengomentari suatu konten diskusi.
8. Mahasiswa dapat melakukan pendaftaran seminar/sidang skripsi. Untuk mendaftar sidang skripsi mahasiswa harus melalui paling tidak 6 kali bimbingan.

9. Fitur penjadwalan atau mengolah jadwal seminar/sidang yang dapat dilakukan oleh koordinator prodi. Pada masa awal penjadwalan ini koordinator dapat menentukan penguji yang untuk suatu skripsi.
10. Fitur terakhir yang perlu diterapkan adalah penilaian. Penilaian dapat dilakukan oleh dosen-dosen pembimbing dan penguji terhadap suatu skripsi. Nilai ini menentukan kelulusan skripsi mahasiswa.

Karena sistem akan dibangun secara microservices maka diperlukan pembagian komponen-komponen sistem menjadi beberapa bagian kecil. Pembagian pada sistem ini terbagi menjadi dua bagian yaitu server dan client. Pada sisi server akan dibagi menjadi 4 layanan disesuaikan dengan pengolahan datanya yaitu layanan user, layanan diskusi, layanan skripsi, dan layanan penjadwalan. Maka disini terdapat 5 bagian dari sistem yaitu 1 client dan 4 server.

3.2.2. Analisis Usecase

Analisis Usecase dilakukan untuk mendefinisikan proses-proses yang akan diterapkan kedalam sistem manajemen skripsi. Usecase ini menunjukkan bagaimana sistem akan merespon kegiatan user dan menunjukkan interaksi yang terjadi antara aktor dengan proses-proses yang ada di dalam sistem [23].

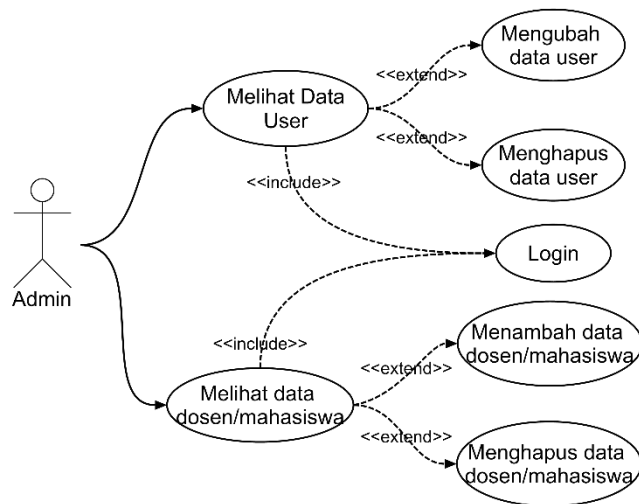


Gambar 4 Usecase Diagram Sistem Manajemen Skripsi

Gambar 4 adalah gambaran interaksi antara setiap aktor dengan usecase utama pada setiap layanan. Terdapat 4 aktor dalam sistem ini Layanan dalam usecase ini terbagi menjadi 4 dan masing-masing layanan memiliki usecasenya masing-masing.

3.2.1.1. Usecase Layanan User

Layanan user adalah salah satu layanan dalam sistem manajemen skripsi yang ditujukan untuk mengatasi pengelolaan data user termasuk data dosen dan mahasiswa. Setiap usecase yang ada dalam layanan user ini harus melewati usecase login terlebih dahulu agar sistem dapat mengenali user yang saat ini sedang berinteraksi. Usecase diagram yang akan diterapkan pada layanan user dapat dilihat pada gambar berikut.



Gambar 5 Usecase Diagram Layanan User

Pada gambar 6 menunjukkan hubungan antara setiap actor dengan setiap usecase yang ada dalam layanan user. Terdapat 4 aktor dan 8 usecase yang berhubungan dalam layanan ini. Detail dari setiap usecase dapat dilihat pada tabel berikut.

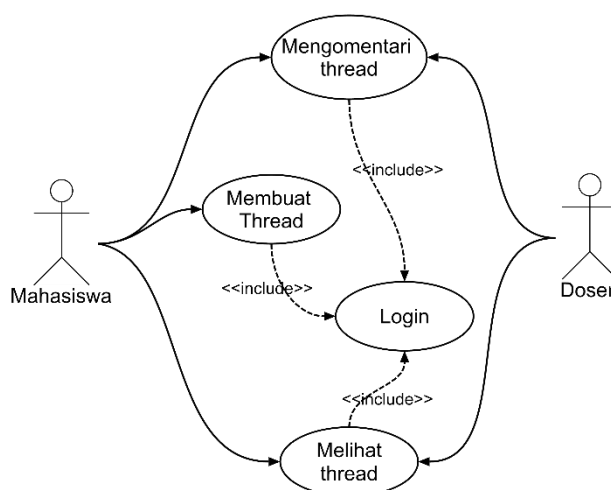
Table 3 Detail Usecase Layanan User

Nama usecase	Aktor	Tujuan
Login	Admin	Melakukan authorisasi user
Mengubah data user	Admin	Memanipulasi data-data user
Menghapus data user	Admin	Menghapus data-data user
Melihat data user	Admin	Melihat seluruh data user
Menambah data dosen/mahasiswa	Admin	Menambahkan data dosen/mahasiswa

Menghapus dosen/mahasiswa	data	Admin	Menghapus data dosen/mahasiswa
Melihat dosen/mahasiswa	data	Admin	Melihat seluruh data dosen dan mahasiswa

3.2.1.2. Usecase Diagram Layanan Diskusi

Usecase diagram untuk layanan diskusi ditunjukkan pada gambar berikut.



Gambar 6 Usecase Diagram Layanan Diskusi

Pada gambar 7 menggambarkan hubungan antara user dengan setiap case yang ada dalam layanan diskusi. Terdapat 2 user yang berperan dalam layanan diskusi ini yaitu dosen dan mahasiswa. Seluruh fitur yang ada dalam layanan ini perlu melalui login terlebih dahulu agar sistem dapat mengenali user. Suatu diskusi dimulai dari user dengan membuat sebuah thread (pesan berantai). Setiap thread dapat dikomentari baik dari sisi dosen ataupun sisi mahasiswa. Ketika akan mengomentari thread user juga perlu melihat thread yang akan dikomentari. Untuk detail setiap usecase dapat dilihat pada tabel berikut

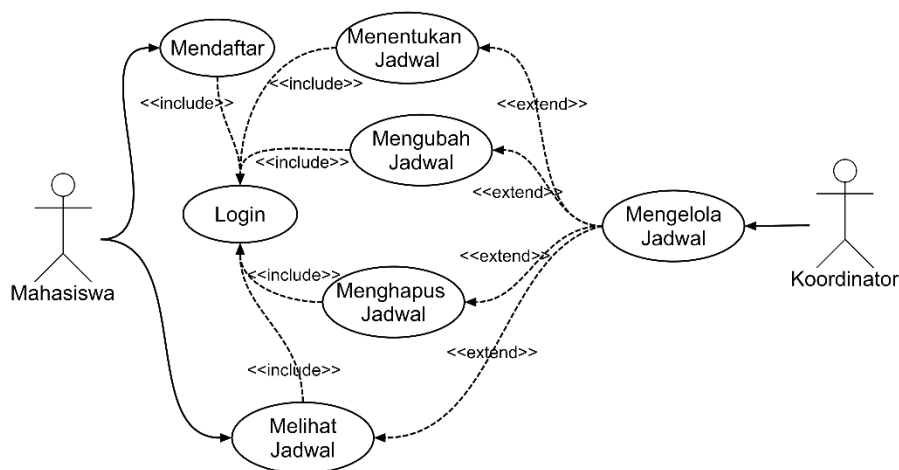
Tabel 4 Detail Usecase Layanan Diskusi

Usecase	Aktor	Tujuan
Login	Dosen dan mahasiswa	Melakukan authorisasi user
Membuat thread	Mahasiswa	Membuat thread baru

Mengomentari thread	Dosen dan mahasiswa	Mengomentari thread yang sudah ada
Melihat thread	Dosen dan mahasiswa	Melihat seluruh thread yang terkait dengan dosen/mahasiswa yang telah login

3.2.1.3. Usecase Diagram Layanan Jadwal

Usecase diagram yang akan diterapkan pada layanan jadwal ditunjukkan pada gambar berikut.



Gambar 7 Usecase Diagram Layanan Jadwal

Pada gambar 8 menggambarkan interaksi yang terjadi antara user dengan setiap aktivitas di dalam layanan jadwal. User yang terlibat dalam aktivitas ini adalah mahasiswa dan koordinator. Layanan ini akan menangani proses penjadwalan baik itu penjadwalan untuk seminar proposal ataupun sidang skripsi. Proses penjadwalan diawali dengan mahasiswa melakukan pendaftaran untuk seminar proposal atau sidang skripsi. Proses selanjutnya akan dilakukan oleh koordinator. Dalam layanan ini koordinator dapat menentukan jadwal dalam hal ruangan, waktu dan dosen pembimbing. Koordinator juga dapat mengubah dan menghapus jadwal yang telah ditentukan.

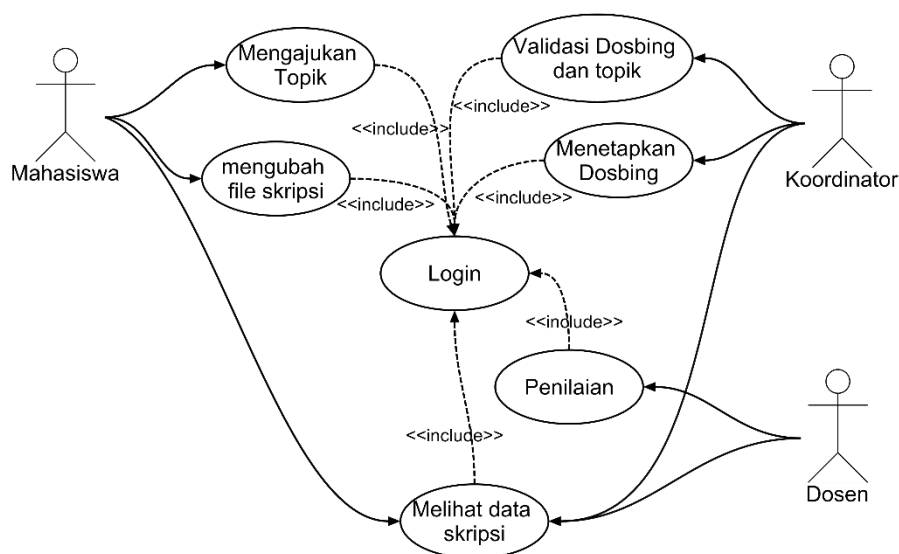
Tabel 5 Detail Usecase Layanan Jadwal

Nama Usecase	Aktor	Tujuan
Login	Mahasiswa dan Koordinator	Authorisasi user

Mendaftar	Mahasiswa	Mendaftarkan skripsi untuk sempro/sidang
Melihat jadwal	Mahasiswa dan Koordinator	Melihat jadwal yang sudah ditambahkan oleh koordinator
Menentukan jadwal	Koordinator	Menentukan jadwal untuk pendaftar seminar proposal/sidang skripsi
Mengubah jadwal	Koordinator	Mengubah jadwal yang telah ditentukan
Menghapus jadwal	Koordinator	Menghapus jadwal yang sudah dibuat

3.2.1.4. Usecase Digaram Layanan Skripsi

Usecase diagram yang akan diterapkan pada layanan skripsi dapat dilihat pada gambar berikut.



Gambar 8 Usecase Diagram Layanan Skripsi

Pada gambar 9 menunjukkan interaksi antara user dan berbagai usecase dalam layanan. Dalam layanan ini terdapat 3 user yang terlibat yaitu mahasiswa, koordinator dan dosen. mahasiswa dapat melakukan pengajuan topik, mengubah file-file skripsi sekaligus melihat data skripsi. Mahasiswa mengajukan topik sekaligus mengusulkan dosen pembimbing untuk membimbing skripsi yang akan dikerjakan. Dosen

pembimbing yang diusulkan oleh mahasiswa nantinya akan di periksa berdasarkan beban dosen. ketika beban dosen sudah penuh maka koordinator akan menetapkan dosen pembimbing. Koordinator juga dapat memvalidasi dosen pembimbing dan topik yang diajukan oleh mahasiswa. Dosen dalam layanan ini berperan sebagai penilai terhadap skripsi yang dikerjakan mahasiswa. Proses penilaian ini diberikan setelah mahasiswa melalui proses sidang skripsi.

Tabel 6 Detail Usecase Layanan Skripsi

Nama Usecase	Aktor	tujuan
Login	Mahasiswa, dosen, koordinator	Authorisasi user yang sedang menggunakan sistem
Mengajukan topik	Mahasiswa	Mengajukan topik dan dosen pembimbing
Mengubah file skripsi	Mahasiswa	Mengupload file skripsi atau mengubah file skripsi yang sudah pernah di upload
Melihat data skripsi	Mahasiswa dan dosen	Melihat seluruh data skripsi yang terkait dengan user
Menetapkan dosen pembimbing	Koordinator	Menetapkan dosen untuk suatu skripsi apabila dosen yang diajukan sudah memiliki beban penuh
Validasi dosen pembimbing	Koordinator	Untuk memvalidasi topik yang diajukan sehingga bisa dilanjutkan ke tahap skripsi
Penilaian	Dosen	Memberikan penilaian terhadap skripsi yang sudah sidang

3.2.2. Activity Diagram

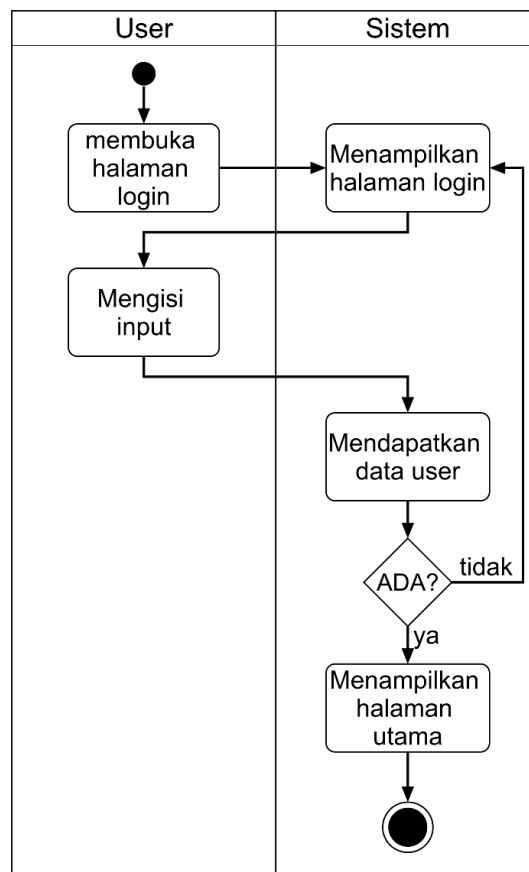
Activity Diagram adalah penggambaran model yang menunjukkan aktivitas-aktivitas yang terjadi di dalam sistem. Dalam sistem ini terbagi menjadi beberapa layanan dan setiap layanan memiliki aktivitas-aktivitas yang menyusunnya.

3.2.2.1. Activity Diagram Layanan User

Dalam Layanan User terdapat beberapa aktivitas yang terjadi yaitu login, mengubah data user, menghapus data user, mengubah profil, menambah data dosen/mahasiswa dan menghapus data dosen/mahasiswa.

a. Login

Login adalah salah satu aktivitas dalam layanan user yang ditujukan untuk mengarahkan user berdasarkan hak akses yang diberikan.

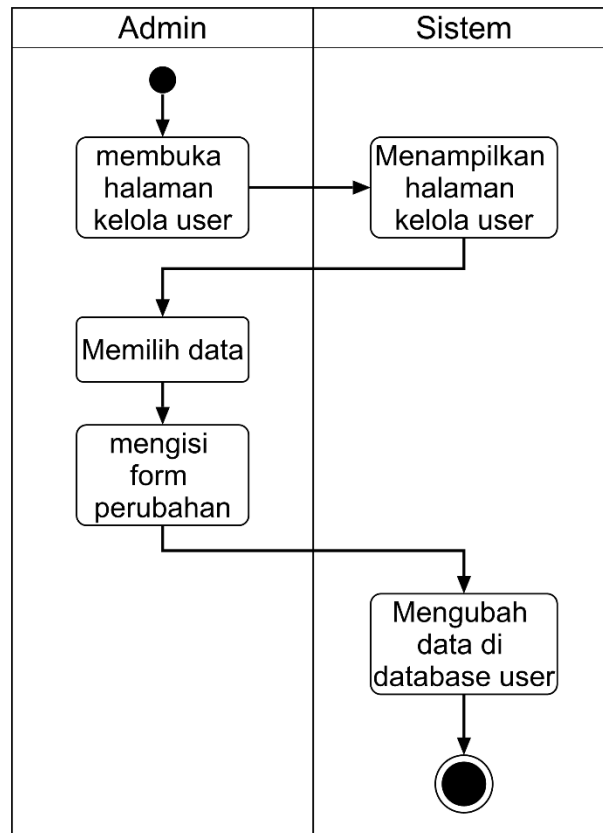


Gambar 9 Activity Diagram Login

Pada gambar 10 menunjukkan aktivitas yang terjadi pada proses login. Aktivitas ini dimulai dengan user membuka halaman login dan sistem akan merespon dengan menampilkan halaman login. User akan melanjutkan dengan mengisi inputan, inputan disini adalah username dan password. Inputan ini akan di olah dengan melakukan pengecekan dari data user jika tidak ada maka user perlu menginputkan kembali username dan password sedangkan jika data ditemukan maka sistem akan mengalihkan user ke halaman utama.

b. Mengubah Data User

Mengubah data user adalah aktivitas dalam layanan user yang ditujukan untuk mempermudah pengelolaan data user dalam hal perubahan data.

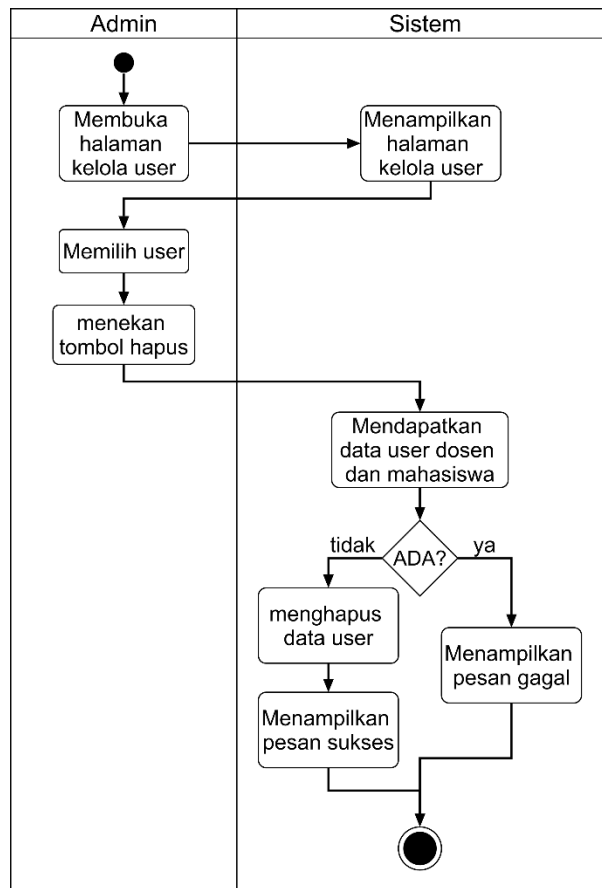


Gambar 10 Activity Diagram Mengubah Data User

Pada gambar 11 menunjukkan aktivitas-aktivitas yang terjadi dalam proses perubahan data user. Proses perubahan data user dilakukan oleh admin dengan membuka halaman kelola user, sistem akan memberikan respon berupa menampilkan halaman kelola user. Dalam halaman kelola user, admin perlu memilih data mana yang akan di ubah. Setelah memilih data yang akan di ubah, admin akan diminta untuk mengisi form perubahan data. Sistem akan mengubah data dalam database user berdasarkan form yang telah di inputkan.

c. Menghapus Data User

Dalam mengelola data user tentu perlu adanya penghapusan data user yang digunakan untuk menghapus data user yang tidak diperlukan.

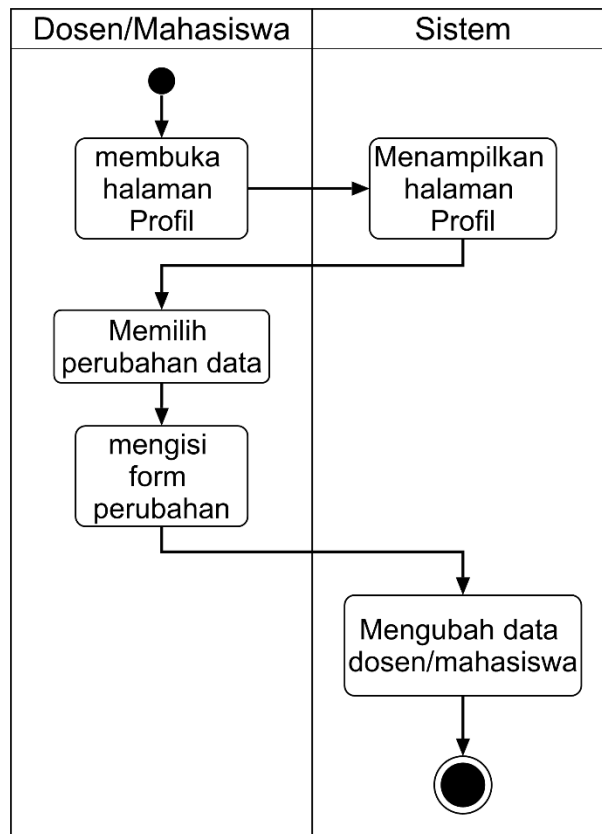


Gambar 11 Activity Diagram Menghapus Data User

Pada gambar 12 ditunjukkan bagaimana proses penghapusan data user pada layanan user. Proses ini hanya dilakukan oleh admin dengan membuka halaman kelola user dan sistem akan menampilkan halaman kelola user. Admin perlu memilih user mana yang akan di hapus. Setelah pemilihan selesai sistem akan mengambil data dosen dan mahasiswa karena kedua data ini terhubung dengan data user. Ketika user tidak ada di dalam data dosen ataupun mahasiswa maka data akan di hapus dan ditampilkan pesan sukses sedangkan ketika user ada dalam data dosen atau mahasiswa maka akan di tampilkan pesan gagal.

d. Mengubah Profil

Aktivitas perubahan profil merupakan salah satu aktivitas di dalam layanan user yang dilakukan oleh dosen dan mahasiswa untuk melengkapi identitas diri mereka.

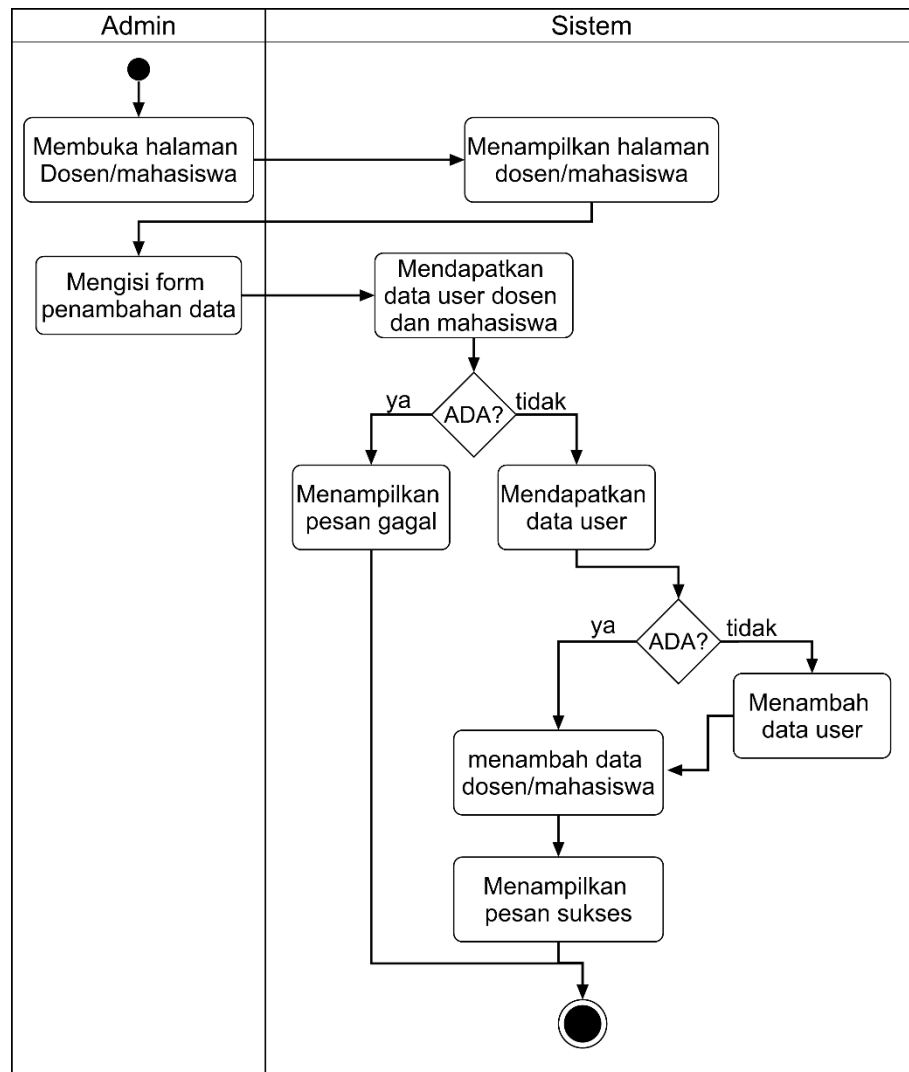


Gambar 12 Activity Diagram Mengubah Profil

Pada gambar 13 ditunjukkan aktivitas yang terjadi pada proses perubahan profil. Proses ini dilakukan oleh mahasiswa/dosen konsepnya sama diawali dengan membuka halaman profil dan sistem akan menampilkan halaman profil. User perlu memilih tombol perubahan data dan mengisi form inputan data yang akan diubah. Sistem akan mengubah data dosen/mahasiswa berdasarkan inputan yang telah diisi. Perubahan yang terjadi disini hanya berbeda tabel saja jika user yang melakukan proses ini adalah mahasiswa maka akan mengubah profil mahasiswa dalam tabel mahasiswa begitu juga pada dosen, jika perubahan dilakukan oleh user dosen maka yang dilakukan perubahan adalah data pada tabel dosen.

e. Menambah Data Dosen/Mahasiswa

Proses penambahan data pada layanan user dilakukan pada data dosen dan mahasiswa. Penambahan data dosen/mahasiswa akan secara otomatis menambahkan data user.

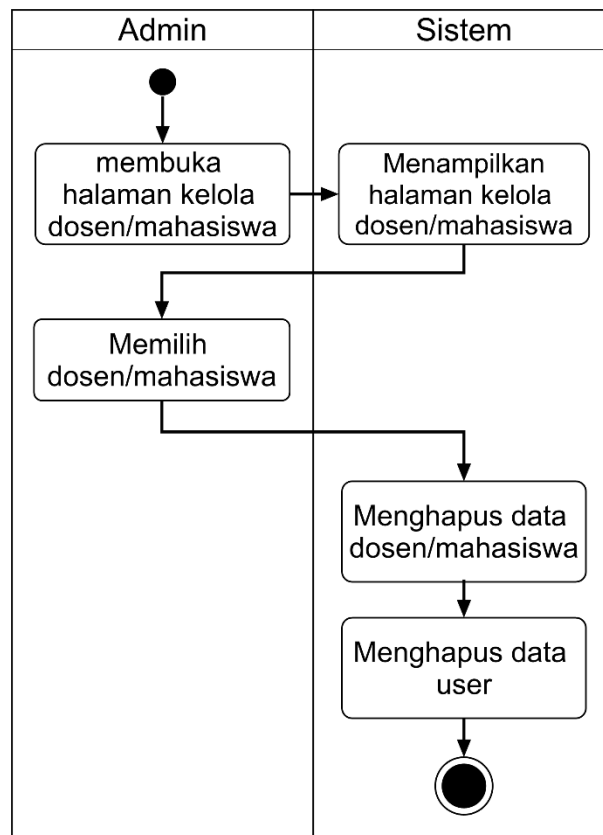


Gambar 13 Activity Diagram Penambahan Data Dosen/Mahasiswa

Pada gambar 14 telah ditunjukkan proses penambahan data dosen/mahasiswa yang terjadi pada layanan user. Proses ini dilakukan oleh admin dengan membuka halaman dosen atau mahasiswa terlebih dahulu dan sistem akan menampilkan halaman dosen atau mahasiswa. Untuk menambahkan data admin perlu mengisi form penambahan data untuk dosen dan mahasiswa kemudian sistem akan melakukan pemeriksaan data berdasarkan data dosen. Jika data dosen/mahasiswa sudah ada maka sistem akan menampilkan pesan gagal namun jika tidak ada data dosen/mahasiswa maka Sistem akan melakukan pemeriksaan lanjutan dengan data user. Ketika tidak ada user dengan username yang sama maka akan dilakukan penambahan data user terlebih dahulu kemudian menambahkan data dosen/mahasiswa namun jika sudah ada data user yang sesuai maka sistem hanya akan menambahkan data dosen/mahasiswa dan proses diakhiri dengan pesan sukses.

f. Menghapus Data Dosen/Mahasiswa

Dalam layanan user juga terdapat proses penghapusan data dosen dan mahasiswa yang dilakukan oleh admin.



Gambar 14 Activity Diagram Menghapus Data Dosen/Mahasiswa

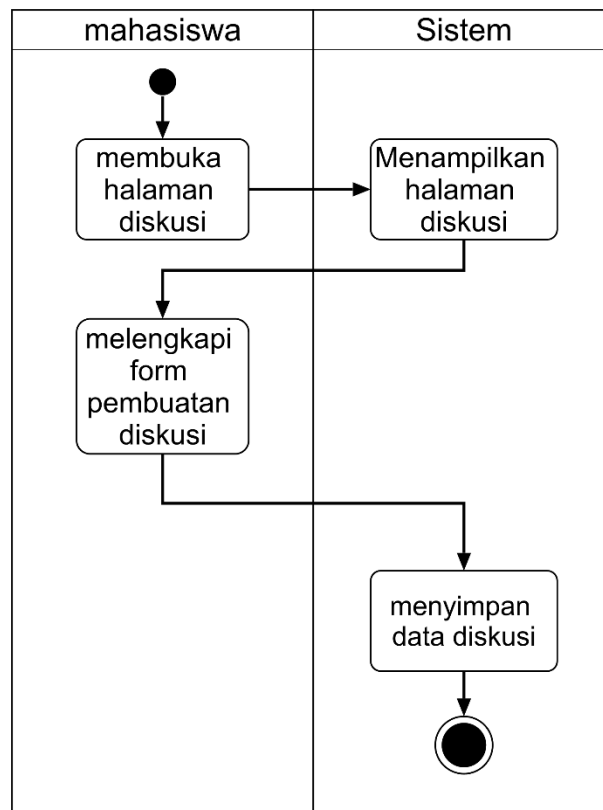
Pada gambar 15 telah ditunjukkan aktivitas yang terjadi pada proses penghapusan data dosen atau mahasiswa. Aktivitas ini dimulai dengan admin membuka halaman kelola dosen/mahasiswa dan sistem menampilkan halaman kelola dosen/mahasiswa. Admin perlu memilih dosen/mahasiswa yang akan dihapus kemudian sistem akan menghapus data yang terpilih dari tabel dosen/mahasiswa dan dilanjutkan dengan menghapus user pada tabel user. Hal ini dilakukan karena kedua tabel ini berkaitan dan tabel user adalah tabel master dari tabel mahasiswa/dosen.

3.2.2.2. Activity Diagram Layanan Diskusi

Pada layanan diskusi terdapat beberapa aktivitas yang terjadi yaitu Membuat thread dan mengomentari thread. Yang berperan dalam layanan ini adalah dosen dan mahasiswa.

a. Membuat Thread

Pada layanan diskusi diperlukan sebuah thread (pesan berantai). Pembuatan thread dilakukan oleh mahasiswa sebagai pembuka diskusi.

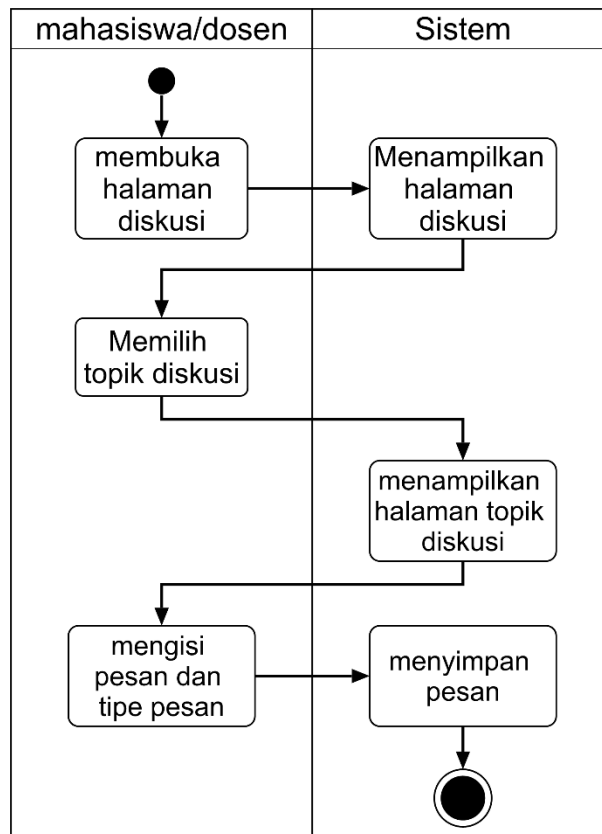


Gambar 15 Activity Diagram Membuat Thread

Pada gambar 16 ditunjukkan aktifitas pembuatan thread. Dalam aktivitas ini diawali dengan mahasiswa membuka halaman diskusi dan sistem akan menampilkan halaman diskusi. Mahasiswa perlu melengkapi form pembuatan diskusi. Dalam form ini terdapat beberapa kebutuhan seperti judul diskusi, pesan awal dan penerima pesan. Sistem akan menyimpan data diskusi yang telah dibuat. Data diskusi ini nantinya yang akan dikomentari dan dicatat.

b. Mengomentari Thread

Mengomentari suatu thread adalah aktivitas yang cukup penting dalam layanan diskusi. Aktivitas ini berfungsi untuk memberikan tanggapan dari thread yang telah dibuat oleh mahasiswa. Pemberi tanggapan dalam layanan ini adalah dosen dan mahasiswa.



Gambar 16 Activity Diagram Mengomentari Thread

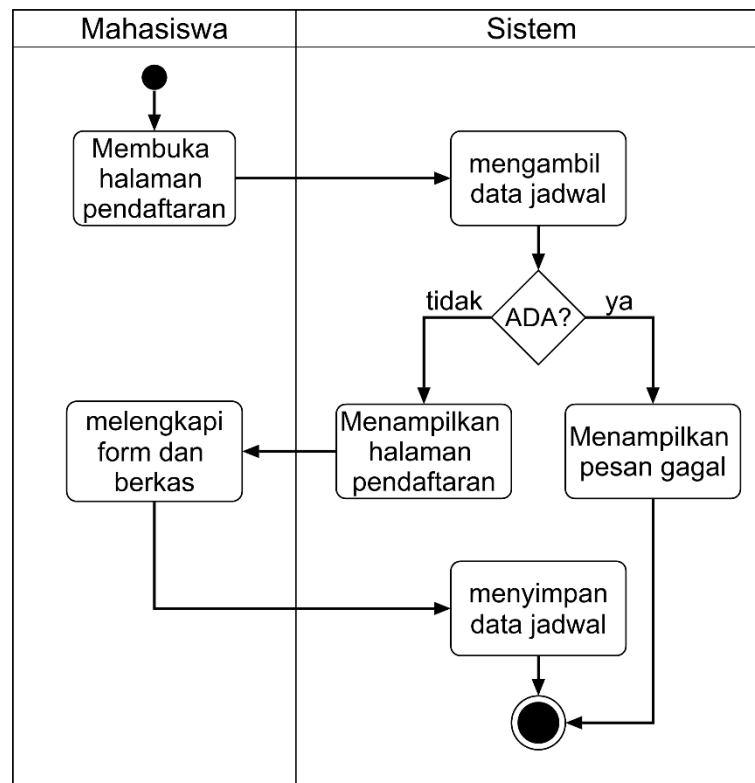
Pada gambar 17 ditunjukkan aktivitas yang terjadi pada proses pembuatan komentar dalam suatu thread. Pemberian komentar/tanggapan ini dimulai dengan mahasiswa/dosen membuka halaman diskusi dan sistem akan menampilkan halaman diskusi. Mahasiswa ataupun dosen perlu memilih topic diskusi yang akan di komentari dan sistem akan menyajikan halaman topik diskusi berdasarkan topik yang dipilih. Mahasiswa atau dosen perlu mengisi pesan dan tipe pesan. Tipe pesan digunakan untuk menentukan pesan yang dikirim perlu atau tidak untuk dicatat pada catatan bimbingan atau revisi. Sedangkan pesan digunakan untuk menuliskan tanggapan terhadap thread. Sistem akan menyimpan pesan kedalam database.

3.2.2.3. Activity Diagram Layanan Jadwal

Pada layanan jadwal terdapat beberapa aktivitas antara lain, mendaftar yang dilakukan oleh mahasiswa, menentukan jadwal dan menghapus jadwal yang dilakukan oleh koordinator.

a. Mendaftar

pada layanan jadwal dimulai dengan pendaftaran. Pada activity pendaftaran ini dilakukan oleh mahasiswa.

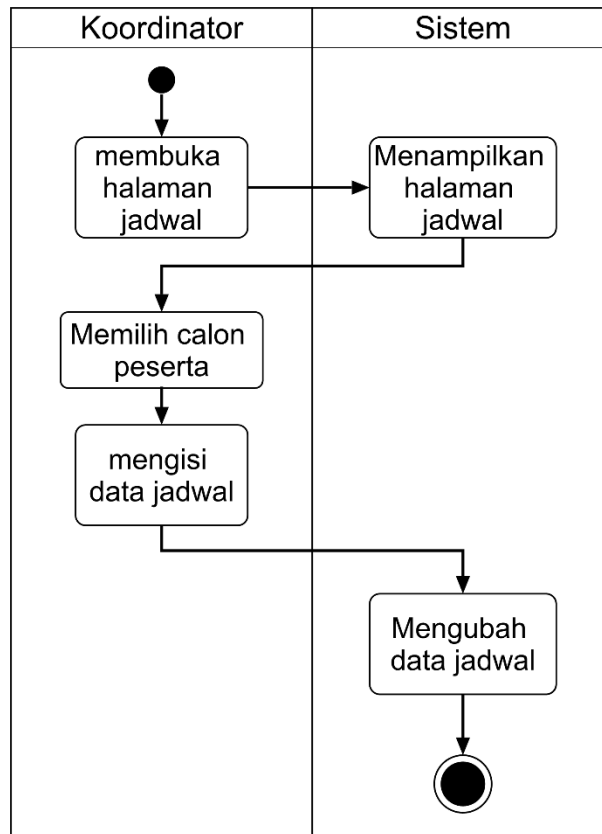


Gambar 17 Activity Diagram Mendaftar

Pada gambar 18 ditunjukkan aktivitas yang terjadi pada proses pendaftaran. Pendaftaran pada layanan ini adalah pendaftaran untuk seminar proposal dan sidang skripsi. Proses ini dimulai dengan mahasiswa membuka halaman pendaftaran dan sistem akan melakukan pengecekan berdasarkan data jadwal. Jika skripsi yang didaftarkan sudah ada dalam database jadwal maka akan di tampilkan pesan gagal. Sedangkan jika skripsi yang di daftarkan tidak ada dalam data jadwal maka sistem akan menampilkan halaman pendaftaran kemudian mahasiswa dapat mengisi form pendaftaran beserta berkas-berkas yang dibutuhkan untuk pendaftaran baik itu pendaftaran seminar proposal ataupun sidang skripsi. Sistem akan menyimpan hasil pendaftaran ini ke database untuk dijadwalkan.

b. Menentukan Jadwal

Penentuan jadwal adalah salah satu aktivitas utama yang ada pada layanan jadwal. Aktivitas ini dilakukan oleh koordinator untuk menjadwalkan suatu skripsi baik untuk melaksanakan seminar proposal ataupun sidang skripsi.

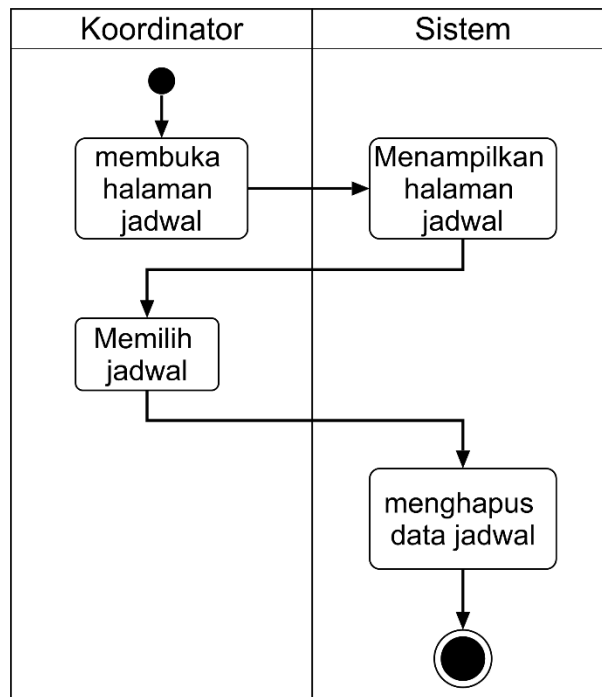


Gambar 18 Activity Diagram Menentukan Jadwal

Gambar 19 menunjukkan aktivitas yang terjadi pada proses penentuan jadwal. Penentuan jadwal dimulai dengan koordinator membuka halaman jadwal dan sistem akan menampilkan halaman jadwal. Koordinator perlu memilih calon peserta yang sudah mendaftar dan siap dijadwalkan. Koordinator harus mengisi data jadwal seperti ruangan waktu dan dosen pembimbing kemudian sistem akan menyimpan data tersebut kedalam data jadwal.

c. Menghapus Jadwal

Salah satu activity dalam layanan jadwal adalah menghapus jadwal. Aktivitas ini diperlukan untuk menghapus data-data yang tidak diperlukan.



Gambar 19 Activity Diagram Menghapus Jadwal

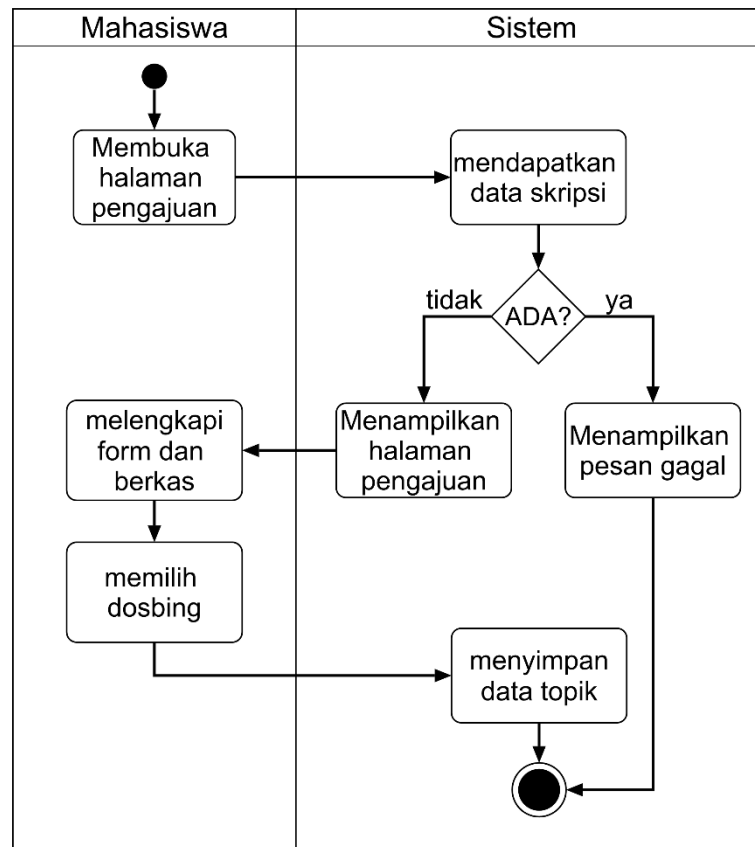
Pada gambar 20 menunjukkan aktivitas yang terjadi pada penghapusan jadwal. Aktivitas ini ditujukan untuk membantu pengelolaan data jadwal khususnya dalam hal penghapusan data jadwal. Proses ini dimulai dengan koordinator membuka halaman jadwal dan sistem akan menampilkan halaman jadwal. Koordinator akan memilih jadwal yang perlu dihapus kemudian sistem akan menghapus data jadwal yang dipilih.

3.2.2.4. Activity Diagram Layanan Skripsi

Pada layanan skripsi terdapat beberapa aktivitas yang terjadi yaitu : pengajuan topic, mengubah file skripsi, menetapkan dosen pembimbing, validasi topik dan dosen pembimbing dan penilaian.

a. Pengajuan Topik

pengajuan topik merupakan salah satu aktivitas yang dilakukan oleh mahasiswa dengan tujuan mengajukan topik untuk skripsi dimana prosesnya bisa dilihat pada gambar berikut.

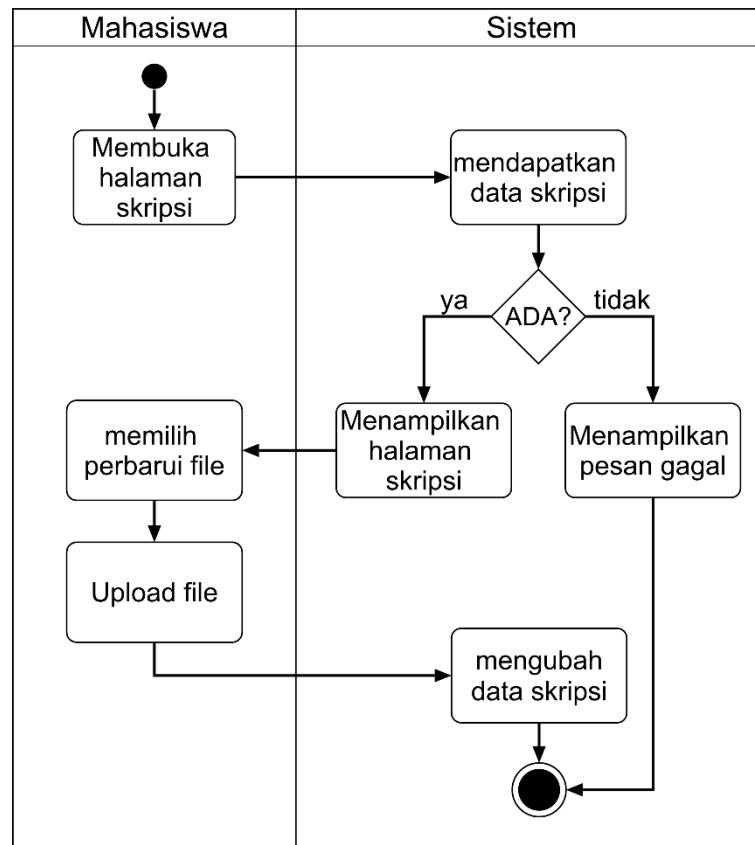


Gambar 20 Activity Diagram Pengajuan Topik

Pada gambar 21 ditunjukkan aktifitas yang terjadi pada proses pengajuan topik. Pada proses ini mahasiswa akan membuka halaman pengajuan dahulu dan sistem akan melakukan pengecekan terhadap data skripsi. Jika sudah ada skripsi yang sedang dalam proses pengerjaan mahasiswa maka akan ditampilkan pesan gagal. Sedangkan jika tidak terdapat skripsi yang terkait dengan mahasiswa maka sistem akan menampilkan halaman pengajuan topik skripsi. Mahasiswa akan diminta untuk melengkapi form-form dan berkas pengajuan topic skripsi. Pada proses ini juga mahasiswa akan mengajukan dosen pembimbing. Data yang sudah diisi mahasiswa pada pengajuan topik skripsi akan disimpan pada data topik. Data topik hanya menyimpan informasi pengajuan dari mahasiswa.

b. Mengubah File Skripsi

Mengubah File Skripsi merupakan aktivitas yang ada dalam layanan user. Aktivitas ini terjadi ketika mahasiswa perlu melakukan perubahan pada data skripsi baik itu ketika mendapatkan saran dari pembimbing ataupun ketika melakukan revisi.

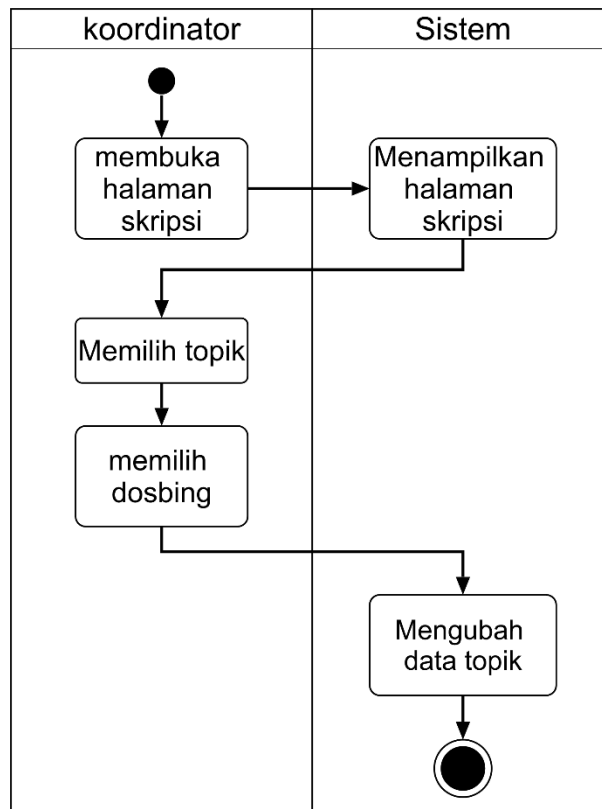


Gambar 21 Activity Diagram Mengubah File Skripsi

Pada gambar 22 menunjukkan activity diagram dari mengubah file skripsi. Proses ini diawali dengan mahasiswa membuka halaman skripsi. Disini sistem akan melakukan pemeriksaan data berdasarkan data skripsi. Jika mahasiswa belum memiliki skripsi yang sedang dikerjakan maka akan ditampilkan pesan gagal. Namun jika mahasiswa memiliki skripsi yang sedang di proses maka akan ditampilkan halaman skripsi dan mahasiswa perlu memilih perbarui file untuk mengubah file skripsi kemudian mahasiswa akan diminta untuk mengunggah file perubahan. Sistem akan mengubah data skripsi berdasarkan perubahan yang dilakukan mahasiswa.

c. Menetapkan Dosen Pembimbing

Menetapkan dosen pembimbing merupakan aktivitas dalam layanan skripsi yang dilakukan oleh koordinator. Ketika mahasiswa mengajukan topik beserta dosen yang ingin dijadikan dosen pembimbing koordinator perlu memeriksa beban yang diterima dosen yang bersangkutan. Ketika beban dosen sudah penuh maka koordinator perlu memilihkan dosen yang sesuai dengan topik mahasiswa.

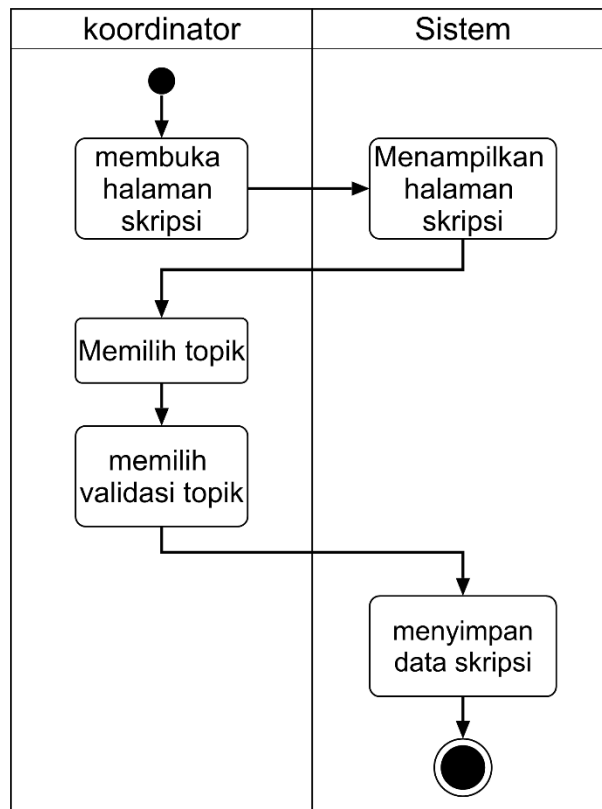


Gambar 22 Activity Diagram Menetapkan Dosen Pembimbing

Pada gambar 23 menunjukkan activity diagram dari penetapan dosen pembimbing. Proses ini dimulai dengan koordinator membuka halaman skripsi. Sistem akan menampilkan halaman skripsi. koordinator perlu memilih topik untuk melihat detail dari topik yang diajukan. Kemudian koordinator dapat memilih dosen pembimbing yang sesuai dengan topik yang diajukan oleh mahasiswa. Sistem akan mengubah data topik berdasarkan perubahan dari koordinator. Data belum tersimpan pada tabel skripsi karena belum di validasi.

d. Validasi Topik dan Dosen Pembimbing

Validasi topik dan pembimbing dilakukan dalam layanan skripsi oleh koordinator dengan tujuan memberikan izin agar topik dapat dilanjutkan ke tahap pembuatan skripsi. Aktivitas ini dilakukan setelah mahasiswa mengajukan topik skripsi dan koordinator telah menetapkan topik dan dosen pembimbing. Sehingga dilakukanlah validasi data tersebut pad aktivitas ini.

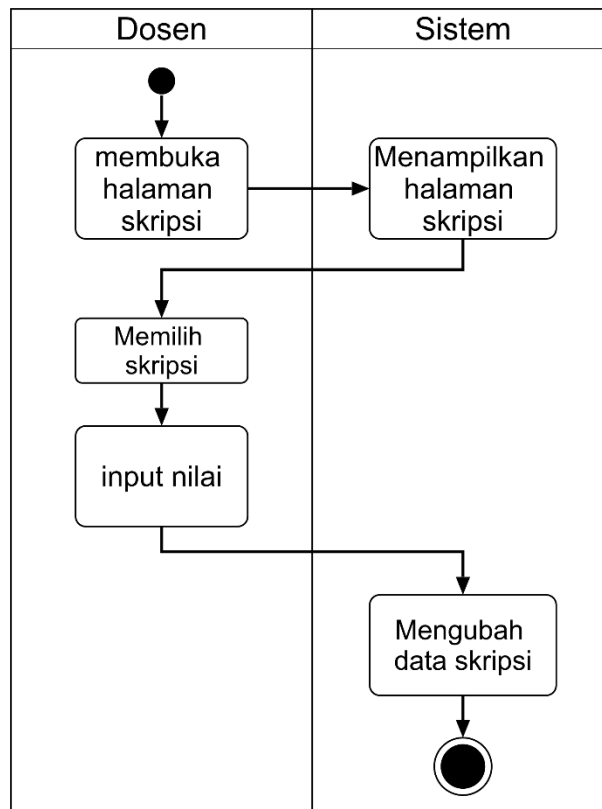


Gambar 23 Activity Diagram Validasi Topik dan Dosen Pembimbing

Pada gambar 24 ditunjukkan aktivitas yang terjadi dalam validasi topik dan dosen pembimbing. Aktivitas ini dimulai dengan koordinator membuka halaman skripsi dan sistem akan menampilkan halaman skripsi. Untuk memvalidasi suatu topik koordinator harus memilih terlebih dahulu topic yang akan di validasi dan sistem akan menyimpan data topik yang tervalidasi kedalam tabel skripsi. Disi juga koordinator perlu memastikan bahwa topik yang diajukan dan dosen pembimbing sudah sesuai dan tidak perlu di ubah. Setelah proses validasi ini mahasiswa dapat melanjutkan proses ke pembuatan skripsi dalam hal ini pembuatan proposal skripsi untuk diseminarkan terlebih dahulu.

e. Penilaian

Penilaian merupakan aktivitas terakhir dalam proses pelaksanaan skripsi. aktivitas ini dilakukan oleh dosen penguji untuk menilai suatu skripsi. penilaian dapat dilakukan setelah skripsi melalui proses sidang skripsi dan seminar proposal. Aktivitas ini yang menentukan kelulusan suatu skripsi yang diajukan oleh mahasiswa.



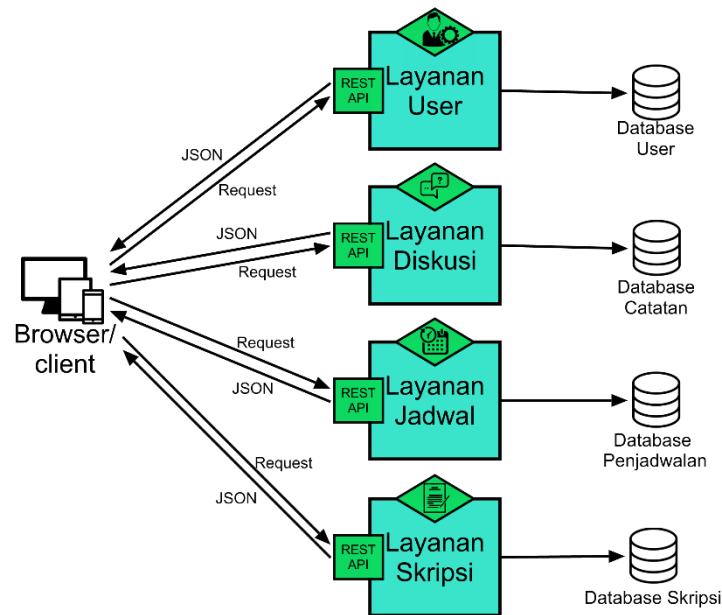
Gambar 24 Activity Diagram Penilaian

Pada gambar 25 ditunjukkan aktivitas yang terjadi dalam proses penilaian skripsi. aktivitas penilaian ini dimulai dengan dosen membuka halaman skripsi dan sistem akan menampilkan halaman skripsi. dosen perlu memilih skripsi yang akan dinilai dan memasukkan nilai untuk skripsi tersebut. Sistem akan mengubah data pada tabel skripsi berdasarkan nilai yang telah dimasukkan. Nilai ini yang dijadikan syarat kelulusan skripsi yang dilaksanakan mahasiswa.

3.3. Perancangan Sistem

3.3.1. Arsitektur Sistem

Secara garis besar gambaran arsitektur sistem yang akan diterapkan pada penelitian ini didasarkan pada arsitektur *microservices* yang di tunjukkan pada gambar 5.

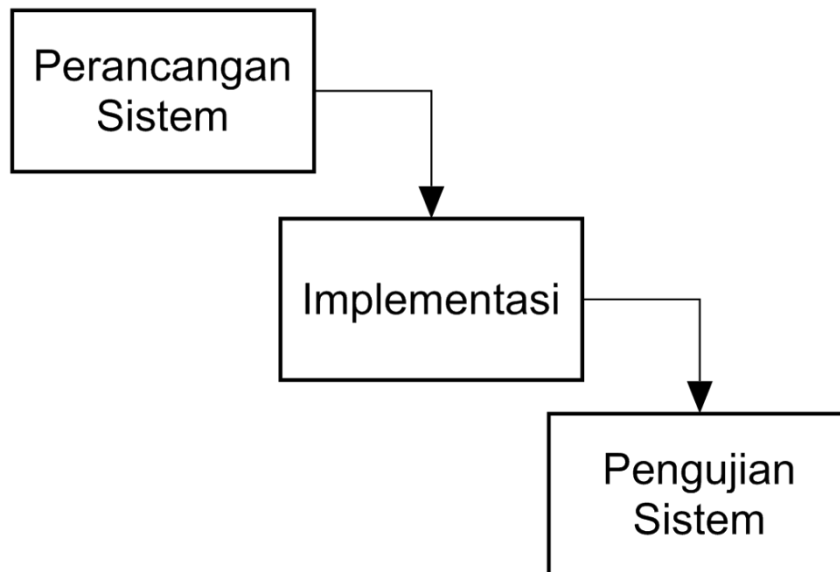


Gambar 25 Arsitektur Sistem

Pada gambar 4 menjelaskan tentang gambaran sistem yang akan dibangun. Dalam sistem ini terbagi menjadi dua bagian yaitu *frontend* dan *backend*. Browser/client adalah sisi *frontend* dimana pada bagian ini pengguna akan berinteraksi dengan halaman-halaman web. bagian lainnya adalah *backend* yang berlaku sebagai *server-side*. *Frontend* akan mengirimkan request untuk mendapatkan data dari *backend*. Layanan-layanan microservice yang bekerja pada serverside memiliki aktivitasnya masing-masing terhadap databasenya masing-masing. Setiap request terhadap suatu layanan akan diberikan respon dalam bentuk JSON yang dapat dikenali oleh banyak bahasa pemrograman.

Dalam sistem ini dibagi menjadi 4 layanan kecil yaitu layanan user, layanan diskusi, layanan jadwal dan layanan skripsi. Setiap layanan memiliki fungsinya masing-masing untuk tujuan yang berbeda-beda. Layanan user akan bertugas mengatasi pengelolaan data user termasuk data dosen dan mahasiswa. Layanan user akan terhubung langsung ke database user. Layanan diskusi akan mengatasi proses diskusi yang terjadi dalam sistem seperti proses bimbingan dan diskusi sidang, layanan ini juga berguna sebagai pencatatan proses bimbingan juga revisi. Layanan diskusi akan terhubung dengan database catatan. Layanan jadwal akan mengatasi permasalahan terkait penjadwalan mulai dari pendaftaran hingga jadwal di tetapkan. Layanan jadwal ini akan terhubung dengan database penjadwalan. Dan layanan yang terakhir adalah layanan skripsi, layanan ini bertugas mengatasi proses skripsi mulai dari pengajuan topik, dosen pembimbing hingga skripsi dinilai. Setiap layanan akan memanajemen datanya masing-masing dengan database yang berbeda- beda. Layanan skripsi akan terhubung dengan database skripsi.

Untuk metode pengembangan akan dilakukan dengan menerapkan siklus hidup pengembangan perangkat lunak *waterfall* sesuai dengan gambar berikut.



Gambar 26 Siklus Hidup Pengembangan Waterfall

Berdasarkan gambar 5 pengembangan sistem dimulai dengan perancangan dimana pada tahap ini akan mendefinisikan rancangan dari sistem yang akan dibangun. Rancangan yang akan dibuat disini adalah activity diagram dan usecase diagram yang akan digunakan. Pada tahap implementasi akan dilakukan pengembangan sistem berdasarkan desain yang telah dihasilkan. Setelah implementasi akan dilakukan pengujian terhadap sistem yang telah dibuat. Pengujian ini dilakukan dengan menguji performa sistem berdasarkan respon time, penggunaan CPU dan penggunaan RAM dengan memanfaatkan tools apache JMeter. Pengujian respon time dilakukan pada sisi client sedangkan pengujian RAM dan CPU akan dilakukan pada sisi server. Dengan mengujikan setiap fitur pada sistem manajemen skripsi yang dibangun dengan pendekatan microservice.

3.4. Skenario Pengujian

Pada tahap ini akan diberikan gambaran terkait pengujian sistem yang akan dilakukan sekaligus menunjukkan bagian-bagian yang akan di uji.

3.4.1. Pengujian Functionalitas

Pengujian yang pertama adalah fungsionalitas. Pengujian ini termasuk pengujian blackbox yang bertujuan untuk memastikan setiap fungsi yang diperlukan telah berjalan sesuai dengan kebutuhan. Fungsi yang akan diuji adalah setiap fungsi yang ada di dalam setiap layanan sistem manajemen skripsi.

3.4.2. Pengujian Performa

Pengujian performa dilakukan pada sistem manajemen skripsi untuk memastikan sistem dapat digunakan dibawah beban kerja yang diharapkan. Parameter yang akan diuji pada pengujian performa dapat dilihat pada tabel berikut.

Tabel 7 Parameter Pengujian Performa

No	Parameter	Satuan	Keterangan
1	Respon Time	Milisecond (ms)	Waktu client menunggu respon terhadap request yang dilakukan
2	CPU Usage	Persentase (%)	Penggunaan CPU pada saat melakukan suatu proses
3	RAM Usage	Byte	Besar penggunaan RAM (memori) pada saat mengeksekusi suatu request

Pengujian performa pada sisi client dilakukan dengan mencatat lama waktu menunggu (respon time) dari request. Sedangkan dari sisi server digunakan untuk menguji besarnya penggunaan CPU dan RAM ketika melakukan suatu eksekusi request. Pengujian ini dilakukan pada setiap sevice yang dibuat dengan mengujikan setiap fungsi pada layanan. Pada setiap pengujian akan dilakukan 5 tahap request. Banyaknya request yang akan dilakukan didasarkan pada pendaftar seminar proposal dan sidang skripsi yang berkisar sekitar 70%-80% dari banyaknya mahasiswa prodi informatika dalam satu angkatan. Jumlah keseluruhan mahasiswa prodi informatika dalam satu angkatan yang berkisar 230 orang sehingga 70% dari 230 adalah 161 sedangkan 80% dari 230 adalah 184. Berdasarkan perhitungan pendaftar seminar proposal atau sidang skripsi maka untuk menguji sistem ini akan digunakan request mulai dari 200, 400, 600, 800 dan 1000 request. Dari hasil pengujian akan dihitung nilai rata-rata untuk setiap tahap pada setiap parameter sehingga dapat terlihat pengaruh dari banyaknya request terhadap setiap parameter. Pengujian ini nantinya akan dilakukan dengan memanfaatkan tool Apache JMeter.

3.5. Rencana Kegiatan

Rencana kegiatan yang akan dilakukan dalam penelitian ini dapat dilihat pada tabel berikut.

Tabel 8 Rencana Kegiatan

No	Kegiatan	Bulan																							
		1				2				3				4				5				6			
		I	II	III	IV	I	II	III	IV	I	II	III	IV	I	II	III	IV	I	II	III	IV	I	II	III	IV
1	Analisa Pemasalahan																								

DAFTAR PUSTAKA

- [1] A. Khozaimi, Ach.; Solihin, Firduas; Jauhari, “PERANCANGAN DAN PEMBUATAN SISTEM INFORMASI TUGAS AKHIR (SIMTAK),” *J. Simantec*, vol. 1, no. 3, pp. 203–211, 2010, [Online]. Available: <https://simantectrunojoyo.files.wordpress.com/2014/04/6-khozaimi-firdaus-jauhari-perancangan-dan-pembuatan-sistem-informasi-tugas-akhir.pdf>.
- [2] A. Khozaimi, S. S. Putro, and M. Rohman, “Pengembangan Aplikasi Managemen Tugas Skripsi (Studi Kasus : Program Studi Teknik Informatika Universitas Trunojoyo Madura),” *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 18, no. 2, pp. 237–245, 2019, doi: 10.30812/matrik.v18i2.392.
- [3] Z. Ren *et al.*, “Migrating web applications from monolithic structure to microservices architecture,” *ACM Int. Conf. Proceeding Ser.*, 2018, doi: 10.1145/3275219.3275230.
- [4] F. Arifien and M. Riastuti, “Model Interoperabilitas Web Service Feeder PDDIKTI Menggunakan Enterprise Javabeans (EJB) dan REST-API,” vol. 3, 2019.
- [5] A. Belkhir, M. Abdellatif, R. Tighilt, N. Moha, Y. G. Gueheneuc, and E. Beaudry, “An observational study on the state of REST API uses in android mobile applications,” in *Proceedings - 2019 IEEE/ACM 6th International Conference on Mobile Software Engineering and Systems, MOBILESoft 2019*, 2019, pp. 66–75, doi: 10.1109/MOBILESoft.2019.00020.
- [6] B. M. Adam, A. Rachmat Anom Besari, and M. M. Bachtiar, “Backend Server System Design Based on REST API for Cashless Payment System on Retail Community,” *IES 2019 - Int. Electron. Symp. Role Techno-Intelligence Creat. an Open Energy Syst. Towar. Energy Democr. Proc.*, pp. 208–213, 2019, doi: 10.1109/ELECSYM.2019.8901668.
- [7] D. I. Permatasari, “Pengujian Aplikasi menggunakan metode Load Testing dengan Apache JMeter pada Sistem Informasi Pertanian,” *J. Sist. dan Teknol. Inf.*, vol. 8, no. 1, p. 135, 2020, doi: 10.26418/justin.v8i1.34452.
- [8] M. Song, C. Zhang, and E. Haihong, “An Auto Scaling System for API Gateway Based on Kubernetes,” *2018 IEEE 9th Int. Conf. Softw. Eng. Serv. Sci.*, pp. 109–112, 2018.
- [9] A. Akbulut and H. G. Perros, “Software Versioning with Microservices through the API Gateway Design Pattern,” *2019 9th Int. Conf. Adv. Comput. Inf. Technol. ACIT 2019 - Proc.*, pp. 289–292, 2019, doi: 10.1109/ACITT.2019.8779952.

- [10] Y. Zhao and X. Wan, "The Design of Embedded Web System based on REST Architecture," in *Proceedings of 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference, IAEAC 2019*, 2019, no. Iaeac, pp. 99–103, doi: 10.1109/IAEAC47372.2019.8997929.
- [11] A. Mubariz, D. Nur, E. Tungadi, M. Nur, and Y. Utomo, "Perancangan Back-End Server Menggunakan Arsitektur Rest dan Platform Node . JS (Studi Kasus : Sistem Pendaftaran Ujian Masuk Politeknik Negeri Ujung Pandang)," pp. 72–77, 2020.
- [12] R. S. Saputra, I. R. Munadi, and D. D. Sanjoyo, "Implementasi Dan Analisis Performansi Platform As a Service Untuk Api Gateway Menggunakan Kong," vol. 5, no. 3, pp. 4973–4979, 2018, [Online]. Available: <https://libraryproceeding.telkomuniversity.ac.id/index.php/engineering/article/viewFile/7883/7776>.
- [13] Sutrisno, M. Asyidiq, and S. Santoso, "Perancangan Sistem Pemasangan Iklan Online Pada Aplikasi E-Commerce (E-Gemanausa) Menggunakan Metode Restful Api Dan Framework Laravel," *J. Sains, Teknol. dan Ind.*, vol. 2, no. 2, pp. 32–38, 2019, [Online]. Available: <http://ejurnal.its.ac.id/index.php/teknik/article/view/5163/1552%0Ahttp://ejournal.lppm-unbaja.ac.id/index.php/saintek/article/view/99>.
- [14] Roy Thomas Fielding, "Architectural Styles and the Design of Network-based Software Architectures," 2000.
- [15] M. G. L. Putra and M. I. A. Putera, "Analisis Perbandingan Metode Soap Dan Rest Yang Digunakan Pada Framework Flask Untuk Membangun Web Service," *SCAN - J. Teknol. Inf. dan Komun.*, vol. 14, no. 2, pp. 1–7, 2019, doi: 10.33005/scan.v14i2.1480.
- [16] S. P. Utomo, N. H. Aliyah, Z. A. Sani, M. Hanafi, and A. Primadewi, "Perancangan RESTFul Web Service pada Sistem Informasi Terintegrasi Menggunakan FrameWork CodeIgniter," *Semin. Nas. Din. Inform.*, pp. 124–128, 2020.
- [17] R. Gunawan and A. Rahmatulloh, "JSON Web Token (JWT) untuk Authentication pada Interoperabilitas Arsitektur berbasis RESTful Web Service," *J. Edukasi dan Penelit. Inform.*, vol. 5, no. 1, p. 74, 2019, doi: 10.26418/jp.v5i1.27232.
- [18] R. K. Lenka and M. R. Dey, "Performance and Load Testing : Tools and Challenges," pp. 2257–2261, 2018.

- [19] T. A. Ghaffur, “Analisis Kualitas Sistem Informasi Kegiatan Sekolah Berbasis Mobile Web Di Smk Negeri 2 Yogyakarta,” *Elinvo (Electronics, Informatics, Vocat. Educ.*, vol. 2, no. 1, pp. 94–101, 2017, doi: 10.21831/elinvo.v2i1.16426.
- [20] B. W. Putra, A. Saputra, R. Sanjaya, and D. Kurniawan, “Implementasi Framework CodeIgniter dan Restful API pada Sistem Informasi Manajemen Tugas Akhir,” *Annu. Res. Semin.*, vol. 5, no. 1, pp. 307–313, 2020.
- [21] Y. Chandra, T. Putra, T. Adi, P. Sidi, and J. E. Samodra, “Implementasi Arsitektur Microservice pada Aplikasi Web Pengajaran Agama Islam Home Pesantren,” vol. 1, no. November, pp. 88–97, 2020.
- [22] X. J. Hong, H. Sik Yang, and Y. H. Kim, “Performance Analysis of RESTful API and RabbitMQ for Microservice Web Application,” in *9th International Conference on Information and Communication Technology Convergence: ICT Convergence Powered by Smart Intelligence, ICTC 2018*, 2018, pp. 257–259, doi: 10.1109/ICTC.2018.8539409.
- [23] A. DENNIS, B. H. WIKOM, and R. M. ROTH, *Systems Analysis and Design, 5th Edition*, 5th ed. Don Fowley, 2012.