

**LEAF: PENGEMBANGAN KERANGKA KERJA  
AGENTIK MINIM KODE YANG DAPAT  
DIPERLUAS UNTUK OBSERVABILITAS  
KOMPUTASI AWAN MODERN**

**PROPOSAL TUGAS AKHIR**

Ditulis sebagai Syarat untuk Pengajuan Tugas Akhir pada Program Studi  
Teknologi Informasi Fakultas Teknik Universitas Negeri Yogyakarta



Oleh:  
**MUHAMMAD NAUFAL KHOIRUL IMAMILHAQ ALHIFDI**  
**NIM 22051130001**

**PROGRAM STUDI TEKNOLOGI INFORMASI  
FAKULTAS TEKNIK  
UNIVERSITAS NEGERI YOGYAKARTA  
2026**

# **LEAF: PENGEMBANGAN KERANGKA KERJA AGENTIK MINIM KODE YANG DAPAT DIPERLUAS UNTUK OBSERVABILITAS KOMPUTASI AWAN MODERN**

## **PROPOSAL TUGAS AKHIR**

Ditulis sebagai Syarat untuk Pengajuan Tugas Akhir pada Program Studi  
Teknologi Informasi Fakultas Teknik Universitas Negeri Yogyakarta



Oleh:  
**MUHAMMAD NAUFAL KHOIRUL IMAMILHAQ ALHIFDI**  
**NIM 22051130001**

**PROGRAM STUDI TEKNOLOGI INFORMASI  
FAKULTAS TEKNIK  
UNIVERSITAS NEGERI YOGYAKARTA  
2026**

# **LEMBAR PERSETUJUAN**

Proposal Tugas Akhir dengan Judul

## **LEAF: PENGEMBANGAN KERANGKA KERJA AGENTIK MINIM KODE YANG DAPAT DIPERLUAS UNTUK OBSERVABILITAS KOMPUTASI AWAN MODERN**

Disusun oleh:

**Muhammad Naufal Khoirul Imamilhaq Alhifdi**  
**NIM 22051130001**

telah memenuhi syarat dan disetujui oleh Dosen Pembimbing untuk dilaksanakan  
Seminar Proposal Tugas Akhir bagi yang bersangkutan.

Yogyakarta, 30 Maret 2026

Mengetahui,  
Koordinator Program Studi,

Disetujui,  
Dosen Pembimbing TA,

Nurkhamid, S.Si., M.Kom., Ph.D.  
NIP. 19680707 199702 1 001

Muslikhin, S.Pd., M.Pd., Ph.D.  
NIP. 19850101 201404 1 001

## DAFTAR ISI

<b>HALAMAN SAMPUL . . . . .</b>	<b>i</b>
<b>LEMBAR PERSETUJUAN PROPOSAL . . . . .</b>	<b>ii</b>
<b>DAFTAR ISI . . . . .</b>	<b>iii</b>
<b>DAFTAR SINGKATAN . . . . .</b>	<b>v</b>
<b>DAFTAR GAMBAR . . . . .</b>	<b>vi</b>
<b>DAFTAR TABEL . . . . .</b>	<b>vii</b>
<b>BAB 1 PENDAHULUAN . . . . .</b>	<b>1</b>
A. Latar Belakang . . . . .	1
B. Rumusan Masalah . . . . .	5
C. Tujuan Proyek . . . . .	5
D. Manfaat Proyek . . . . .	6
E. Batasan Proyek . . . . .	6
F. Keaslian Gagasan . . . . .	6
G. Sistematika Penulisan . . . . .	7
<b>BAB 2 TINJAUAN PUSTAKA . . . . .</b>	<b>8</b>
A. Teori Dasar Komponen Elektronika . . . . .	8
1. Jenis dan Karakteristik Komponen Pasif . . . . .	8
2. Jenis dan Karakteristik Komponen Aktif . . . . .	8
3. Peran dan Fungsi Modul dalam Sistem Elektronika . . . . .	8
4. Analisis Daya dan Efisiensi Komponen . . . . .	8
B. Sistem dan Teknik Rangkaian Elektronika . . . . .	9
1. Konsep Dasar Rangkaian Analog . . . . .	9
2. Konsep Dasar Rangkaian Digital . . . . .	9
3. Teknik Pengolahan Sinyal pada Sistem Elektronika . . . . .	9
4. Pengkabelan dan Pengaturan Sirkuit untuk Keandalan Sistem . . . . .	9
5. Pengendalian dan Penggerak (Motor Driver, Relay, dsb.) . . . . .	9
C. Teknologi yang Digunakan . . . . .	10
1. Mikrokontroler dan Mikroprosesor . . . . .	10
2. Sensor dan Aktuator . . . . .	10
3. Teknologi Nirkabel . . . . .	10
D. Metode Kontrol dan Kecerdasan Buatan . . . . .	10
1. Pengendalian PID (Proportional-Integral-Derivative) . . . . .	10
2. Fuzzy Logic Control . . . . .	11
3. Deep Learning . . . . .	11
4. Perbandingan dan Pemilihan Metode yang Sesuai . . . . .	11
E. Konsep Engineering Design Process . . . . .	11
1. Pengertian dan Langkah-langkah Engineering Design Process . . . . .	11
2. Aplikasi Engineering Design Process pada Proyek Elektronika . . . . .	11
3. Studi Kasus Implementasi Engineering Design Process dalam Desain Elektronika . . . . .	12
4. Teknik Evaluasi dan Optimasi Desain . . . . .	12

F. Penelitian Terdahulu yang Relevan . . . . .	12
1. Tinjauan Penelitian Terdahulu tentang Proyek Serupa . . . . .	12
2. Analisis Kekurangan dan Kelebihan Metode pada Penelitian Terdahulu . . . . .	12
3. Inovasi dan Kontribusi yang Dibawa dalam Penelitian Ini . . . . .	12
<b>BAB 3 KONSEP RANCANGAN ALAT DAN PENGUJIAN . . . . .</b>	<b>13</b>
A. Metode Pengerjaan Project Berbasis Engineering Design Process . . . . .	13
1. Identifikasi Masalah . . . . .	13
2. Definisi Kebutuhan . . . . .	13
3. Generasi Ide dan Solusi . . . . .	13
4. Perencanaan dan Desain Awal . . . . .	13
5. Pembuatan Prototipe . . . . .	14
6. Pengujian dan Evaluasi . . . . .	14
7. Perbaikan dan Penyempurnaan . . . . .	14
B. Perancangan Sistem Elektronika . . . . .	14
1. Blok Diagram Sistem . . . . .	14
2. Pemilihan dan Spesifikasi Komponen Elektronika . . . . .	14
3. Perancangan Rangkaian Elektronika . . . . .	15
C. Perancangan Mekanik . . . . .	15
1. Spesifikasi Desain Mekanik . . . . .	15
2. Pemilihan Bahan dan Komponen Mekanik . . . . .	15
3. Desain Struktur dan Konstruksi . . . . .	15
D. Perancangan Perangkat Lunak . . . . .	15
1. Flowchart atau Diagram Alir Perangkat Lunak . . . . .	15
2. Pemrograman dan Pengembangan Kode . . . . .	16
3. Pengujian Kode Perangkat Lunak . . . . .	16
E. Perancangan Integrasi Sistem . . . . .	16
1. Integrasi Komponen Elektronika, Mekanik, dan Perangkat Lunak . . . . .	16
2. Pengujian Awal dan Penyempurnaan Integrasi . . . . .	16
F. Rencana Pengujian . . . . .	16
1. Metode Pengujian Sistem . . . . .	16
2. Prosedur Pengujian . . . . .	17
3. Kriteria Keberhasilan Pengujian . . . . .	17
<b>DAFTAR PUSTAKA . . . . .</b>	<b>18</b>
<b>LAMPIRAN A KODE PROGRAM . . . . .</b>	<b>21</b>
Lampiran A.1. Program Pembacaan Sensor Ultrasonic . . . . .	21
Lampiran A.2. Program Keseluruhan Proyek Akhir . . . . .	21
<b>LAMPIRAN B GAMBAR-GAMBAR . . . . .</b>	<b>22</b>
Lampiran B.1. Foto Aktivitas Kegiatan Proyek Akhir . . . . .	22
Lampiran B.2. Foto Produk Proyek Akhir . . . . .	22

## DAFTAR SINGKATAN

FWHM	:	<i>Full width half maximum</i>
rms	:	<i>root mean square</i>
RFS	:	<i>Rotary forcespinning</i>
PVP	:	Polivinil pirolidon
SI	:	Satuan Internasional
SI	:	Satuan Internasional
SI	:	Satuan Internasional
SI	:	Satuan Internasional
SI	:	Satuan Internasional
SI	:	Satuan Internasional
SI	:	Satuan Internasional
SI	:	Satuan Internasional
SI	:	Satuan Internasional

## **DAFTAR GAMBAR**

## **DAFTAR TABEL**



# BAB I

## PENDAHULUAN

### A. Latar Belakang

Perkembangan teknologi *cloud computing* dan arsitektur *microservices* telah mengubah secara fundamental cara organisasi membangun dan mengoperasikan sistem perangkat lunak Burns et al. (2016). Sistem modern yang berjalan di platform seperti Kubernetes dapat terdiri dari ratusan hingga ribuan *container* yang saling berinteraksi, menciptakan kompleksitas operasional yang belum pernah terjadi sebelumnya Cloud Native Computing Foundation (2024). Dalam konteks ini, praktik *Site Reliability Engineering* (SRE) yang diperkenalkan oleh Google menjadi pendekatan standar untuk memastikan keandalan sistem berskala besar Beyer et al. (2016).

Untuk memahami dan mengelola sistem terdistribusi yang kompleks, konsep *observability* telah berevolusi dari pemantauan tradisional menjadi pendekatan yang lebih komprehensif Sridharan (2018). *Observability* modern dibangun di atas tiga pilar utama: *metrics* (data numerik tentang kondisi sistem), *logs* (catatan kejadian terstruktur), dan *traces* (pelacakan alur permintaan antar layanan) Picoreti et al. (2018). Namun, ketiga pilar ini memiliki keterbatasan ketika digunakan secara terpisah—*metrics* tidak menjelaskan penyebab masalah, *logs* sulit dikorelasikan lintas layanan, dan *traces* memerlukan instrumentasi yang kompleks Soldani et al. (2022).

Arsitektur Kubernetes dan *microservices* memperparah tantangan ini karena sifatnya yang dinamis dan terdistribusi. Sebuah permintaan pengguna dapat melewati puluhan layanan sebelum menghasilkan respons, dan kegagalan di satu komponen dapat berdampak kaskade ke seluruh sistem Burns et al. (2018). Hal ini menciptakan kebutuhan mendesak akan *unified observability*—kemampuan untuk mengkorelasikan data dari ketiga pilar secara otomatis untuk memahami kondisi sistem secara holistik.

Untuk mengatasi kompleksitas ini, industri telah mengembangkan solusi *Artificial Intelligence for IT Operations* (AIOps) yang menerapkan teknik *machine*

*learning* untuk mengotomatisasi tugas-tugas operasional Notaro et al. (2021). Platform komersial seperti Datadog, Dynatrace, dan Splunk AIOps menawarkan kemampuan deteksi anomali, korelasi kejadian, dan prediksi kegagalan berbasis AI Dang et al. (2019). Namun, solusi-solusi ini memiliki keterbatasan fundamental yang menghambat adopsi dan efektivitasnya.

Pertama, arsitektur AIOps saat ini bersifat *monolithic*—menggabungkan *data ingestion*, analisis, dan aksi dalam satu platform *vendor* tunggal Lyu et al. (2021). Organisasi tidak dapat dengan mudah mengganti atau menggabungkan komponen dari *vendor* berbeda untuk mendapatkan solusi terbaik. Kedua, masalah *vendor lock-in* membuat organisasi bergantung pada ekosistem tertutup, mengurangi fleksibilitas dan meningkatkan biaya jangka panjang.

Menanggapi kebutuhan akan interoperabilitas, standar komunikasi baru untuk agen AI telah muncul. *Model Context Protocol* (MCP) yang dikembangkan oleh Anthropic menyediakan antarmuka standar bagi agen untuk mengakses konteks lingkungan—seperti data dari Kubernetes, Prometheus, atau sistem *logging*—tanpa perlu implementasi API spesifik untuk setiap sumber data Anthropic (2024). Sementara itu, protokol *Agent-to-Agent* (A2A) dari Google memungkinkan pertukaran tugas terstruktur antar agen heterogen, terlepas dari *framework* atau bahasa pemrograman yang digunakan Google (2024).

Kemunculan protokol-protokol ini menandai pergeseran paradigma dari sistem AI yang terisolasi menuju ekosistem agen yang dapat berinteroperasi Xi et al. (2023). Beberapa *framework* pengembangan agen telah mengadopsi standar ini: Google Agent Development Kit (ADK) mendukung A2A secara native Google Cloud (2024), LangChain menyediakan integrasi dengan MCP LangChain Inc. (2024), dan CrewAI memungkinkan orkestrasi multi-agen dengan dukungan protokol standar CrewAI Inc. (2024).

Meskipun standar komunikasi telah tersedia dan diadopsi oleh *framework* modern, terdapat kesenjangan signifikan dalam aksesibilitas. Semua *framework* agen yang ada saat ini—termasuk Google ADK, LangChain, dan CrewAI—memerlukan keahlian pemrograman yang substansial untuk digunakan. Pengembang harus memahami pola-pola SDK, menulis kelas agen dalam Python

atau JavaScript, mengimplementasikan *binding* untuk *tools*, dan mengelola logika koordinasi yang kompleks—semuanya dalam kode CrewAI Inc. (2024); LangChain Inc. (2024).

Situasi ini menciptakan hambatan masuk (*entry barrier*) yang tinggi ke era agentik. Pakar domain seperti praktisi SRE, analis data, atau tim operasional yang memahami alur kerja mereka dengan baik tidak dapat membangun aplikasi agentik tanpa terlebih dahulu menjadi pengembang perangkat lunak. Terdapat dilema pilihan yang semu (*false choice*): menerima keterbatasan alat *no-code* yang sederhana, atau berkomitmen untuk menguasai pemrograman secara penuh. Tidak ada *framework* yang menawarkan kesederhanaan deklaratif dengan ekstensibilitas berbasis standar.

Dari perspektif teoritis, sistem multi-agen (*Multi-Agent Systems/MAS*) telah dipelajari secara ekstensif dalam literatur kecerdasan buatan. Wooldridge (2009) mendefinisikan agen sebagai entitas yang otonom, reaktif terhadap lingkungan, proaktif dalam mencapai tujuan, dan mampu berinteraksi sosial dengan agen lain. Jennings et al. (1998) memetakan berbagai paradigma koordinasi agen, termasuk sistem *blackboard*, *contract nets*, dan dekomposisi tugas hierarkis.

Paradigma *blackboard* khususnya relevan untuk koordinasi agen dalam konteks *observability*. Seperti dijelaskan oleh Corkill (1991), sistem *blackboard* menyediakan ruang kerja bersama di mana agen-agen spesialis dapat berkontribusi secara inkremental untuk memecahkan masalah kompleks. Pendekatan ini mengatasi masalah *context bloat*—di mana pengiriman data mentah antar agen menyebabkan *overflow* token dan inefisiensi—dengan memungkinkan agen berbagi *pointer* ke *state* daripada data itu sendiri.

Berdasarkan analisis kesenjangan di atas, penelitian ini mengusulkan **LEAF** (*Low-code Extensible Agentic Framework*)—sebuah *framework* agentik *low-code* yang dirancang untuk mendemokratisasi pengembangan aplikasi multi-agen. LEAF menggunakan pendekatan deklaratif berbasis manifes YAML/JSON, analog dengan cara Kubernetes mengabstraksi manajemen infrastruktur melalui konfigurasi deklaratif Artac et al. (2017); Burns et al. (2018); Morris (2020).

Kontribusi utama LEAF adalah menyediakan model kompleksitas progresif (*progressive complexity model*) dengan tiga tingkat:

1. **Level 1 (Deklaratif Murni):** Pengguna mendefinisikan agen sepenuhnya dalam YAML, mereferensikan *tools* MCP dari komunitas, tanpa perlu menulis kode sama sekali.
2. **Level 2 (Deklaratif + Referensi):** Manifes YAML dapat mereferensikan agen A2A eksternal atau server MCP kustom yang dikembangkan secara terpisah.
3. **Level 3 (Implementasi Kustom):** Pengembang yang memerlukan kontrol penuh dapat menulis agen kustom menggunakan LEAF SDK, yang tetap dapat berinteroperasi melalui standar A2A/MCP.

Dengan demikian, standar MCP/A2A berfungsi sebagai “*escape hatch*”—memungkinkan kesederhanaan *low-code* tanpa menjadi pembatas yang mengekang. Pakar domain dapat memulai dengan konfigurasi deklaratif untuk kasus penggunaan umum (Level 1), kemudian berkembang sesuai kebutuhan tanpa harus memulai dari awal.

Arsitektur LEAF dibangun di atas konsep “*Opinionated Grid*” yang memisahkan penalaran AI (*reasoning*) dari pipa infrastruktur (*plumbing*) melalui protokol komunikasi terstandarisasi. Grid ini mendefinisikan tiga arah komunikasi: *Southbound* (menggunakan MCP untuk query konteks lingkungan), *Northbound* (menggunakan A2A untuk komunikasi dengan pengguna), dan *East-West* (menggunakan A2A untuk koordinasi antar agen) Anthropic (2024); Google (2024).

LEAF mengorganisasi agen dalam taksonomi empat lapis: *Orchestrator* (perencanaan dan koordinasi), *Sensor* (pengumpulan data), *Processor* (analisis dan inferensi), dan *Executor* (implementasi aksi). Untuk koordinasi, LEAF mengadopsi pola *Blackboard* yang memungkinkan agen berbagi *state pointer* (URI) ke penyimpanan versi terpusat, menghindari *context bloat* Corkill (1991); Dorri et al. (2018).

Sebagai prinsip desain inti, LEAF menekankan pengawasan manusia (*human-in-the-loop*) untuk semua aksi yang mengubah sistem. Berdasarkan

pedoman interaksi manusia-AI Amershi et al. (2019); Shneiderman (2020), semua aksi *Executor* memerlukan persetujuan eksplisit dari manusia sebelum dieksekusi. Selain itu, pola *Reflector agent* menyediakan verifikasi independen terhadap temuan sebelum aksi diambil.

Kontribusi penelitian ini adalah sebagai *framework* agentik *low-code* pertama yang menggabungkan desain berbasis manifes deklaratif dengan ekstensibilitas berbasis standar (MCP/A2A). LEAF memungkinkan pakar domain untuk membangun sistem multi-agen tanpa pemrograman, sambil mempertahankan "*escape hatch*" untuk kustomisasi lanjutan oleh pengembang. Penelitian ini mencakup: (1) cetak biru arsitektur, (2) spesifikasi manifes deklaratif (JSON Schema), dan (3) implementasi referensi.

**Ruang Lingkup Tesis:** Meskipun LEAF dirancang sebagai *framework* tujuan umum yang dapat diterapkan pada domain apa pun yang memerlukan koordinasi multi-agen (misalnya pengembangan perangkat lunak, *pipeline* data, otomatisasi bisnis), tesis ini secara spesifik berfokus pada ***observability* Kubernetes dan operasi SRE** sebagai studi kasus implementasi untuk memvalidasi pola-pola inti *framework*.

## **B. Rumusan Masalah**

Rumusan masalah adalah langkah penting yang merangkum permasalahan spesifik atau kebutuhan yang menjadi dasar pengembangan proyek. Pada bagian ini, dijelaskan permasalahan utama yang dihadapi, seperti keterbatasan pada sistem atau perangkat yang sudah ada, atau kebutuhan baru yang belum terpenuhi oleh teknologi saat ini. Identifikasi masalah juga mencakup tantangan teknis, fungsional, atau ekonomi yang menjadi penghambat dan bagaimana proyek ini diharapkan dapat menjawab permasalahan tersebut. Fokusnya adalah memberikan pemahaman yang jelas mengenai alasan pentingnya mengembangkan proyek ini sebagai solusi yang dibutuhkan.

## **C. Tujuan Proyek**

Tujuan proyek menyatakan secara spesifik hasil atau capaian yang diinginkan dari pengembangan proyek ini. Bagian ini dirancang untuk memastikan

bahwa proyek memiliki sasaran yang jelas dan terukur. Tujuan tersebut dirumuskan berdasarkan masalah yang telah diidentifikasi dan mencakup pencapaian tertentu, seperti peningkatan kinerja sistem, efisiensi, atau kemudahan penggunaan yang diharapkan. Selain itu, tujuan proyek dapat berupa pengembangan prototipe, penerapan teknologi tertentu, atau pencapaian fungsionalitas baru yang belum ada. Penjabaran tujuan yang jelas membantu menjaga fokus proyek dan memberikan arah yang tepat dalam setiap tahapan pengembangan.

#### **D. Manfaat Proyek**

Manfaat proyek menguraikan dampak positif yang diharapkan dari hasil proyek ini bagi pengguna, industri, atau masyarakat secara umum. Manfaat ini mencakup berbagai aspek, seperti kontribusi terhadap peningkatan produktivitas, pengurangan biaya, peningkatan kualitas, atau kemudahan dalam penggunaan teknologi. Selain manfaat langsung, proyek ini juga diharapkan memiliki dampak jangka panjang yang bermanfaat, seperti mendorong inovasi di bidang terkait atau membuka peluang baru untuk pengembangan lebih lanjut. Dengan menjelaskan manfaat proyek, pembaca dapat memahami nilai tambah yang dihadirkan oleh proyek ini.

#### **E. Batasan Proyek**

Batasan proyek mengidentifikasi cakupan dan batasan ruang lingkup pengembangan sistem atau perangkat yang dirancang. Bagian ini mencakup aspek-aspek yang akan menjadi fokus utama dalam pengembangan serta aspek yang akan dikecualikan dari lingkup proyek. Penjelasan batasan ini penting agar proyek tetap terarah dan tidak meluas ke aspek-aspek yang berada di luar tujuan awal. Batasan proyek juga mencakup keterbatasan teknis, waktu, atau sumber daya yang mempengaruhi desain dan implementasi sistem. Dengan menetapkan batasan, proyek ini dapat lebih terfokus dan efisien dalam pencapaiannya.

#### **F. Keaslian Gagasan**

Keaslian gagasan bertujuan untuk menekankan inovasi atau kontribusi unik yang ditawarkan oleh proyek ini. Bagian ini menjelaskan bagaimana proyek ini

menawarkan pendekatan yang berbeda atau peningkatan dibandingkan dengan metode atau perangkat yang sudah ada. Keaslian gagasan dapat diperlihatkan melalui perbandingan dengan proyek atau produk serupa, menunjukkan perbedaan signifikan atau keunggulan yang dihadirkan oleh solusi yang diusulkan. Misalnya, peningkatan kinerja, efisiensi, atau kemudahan penggunaan yang dihasilkan dari metode atau pendekatan baru. Selain itu, bagian ini juga bisa mencakup penggunaan teknologi atau desain yang belum banyak diterapkan dalam konteks yang sama. Penekanan pada keaslian gagasan membantu menunjukkan bahwa proyek ini tidak hanya mengikuti pola yang sudah ada, tetapi juga menghadirkan sesuatu yang baru dan relevan.

#### **G. Sistematika Penulisan**

Sistematika penulisan memberikan panduan mengenai struktur dari keseluruhan laporan proyek ini, sehingga memudahkan pembaca dalam memahami alur isi laporan dari setiap bab. Bagian ini menjelaskan isi dari setiap bab secara singkat, mulai dari latar belakang hingga kesimpulan dan rekomendasi. Misalnya, BAB I membahas pendahuluan dan dasar pengembangan proyek, BAB II menguraikan tinjauan pustaka dan landasan teori, dan seterusnya. Dengan memberikan sistematika penulisan, pembaca dapat memahami bagaimana laporan ini disusun secara keseluruhan dan bagaimana setiap bab saling berkaitan dalam mencapai tujuan akhir proyek.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **A. Teori Dasar Komponen Elektronika**

Bagian ini membahas teori dasar mengenai komponen elektronika yang digunakan dalam perancangan sistem. Setiap komponen memiliki karakteristik khusus yang mempengaruhi kinerja keseluruhan sistem. Pemahaman tentang karakteristik komponen-komponen ini sangat penting dalam mengoptimalkan fungsi dan stabilitas rangkaian.

##### **1. Jenis dan Karakteristik Komponen Pasif**

Komponen pasif seperti resistor, kapasitor, dan induktor memiliki fungsi dasar dalam pengaturan arus dan tegangan dalam sirkuit. Bagian ini menguraikan jenis-jenis komponen pasif serta karakteristik utama yang mempengaruhi performa dan fungsi komponen tersebut dalam sirkuit elektronika.

##### **2. Jenis dan Karakteristik Komponen Aktif**

Komponen aktif, seperti transistor, dioda, dan IC, memainkan peran penting dalam penguatan dan pengaturan sinyal. Bagian ini menjelaskan berbagai jenis komponen aktif yang digunakan dalam proyek serta karakteristik utamanya, yang menentukan efektivitas dan efisiensi sistem elektronika.

##### **3. Peran dan Fungsi Modul dalam Sistem Elektronika**

Modul-modul elektronika memberikan fungsionalitas tambahan yang membantu dalam memperkuat performa sistem. Bagian ini mengulas modul-modul yang sering digunakan, seperti modul daya atau komunikasi, serta peran masing-masing dalam mendukung integrasi sistem yang lebih efisien.

##### **4. Analisis Daya dan Efisiensi Komponen**

Analisis daya dan efisiensi komponen adalah aspek penting dalam desain sistem yang hemat energi. Bagian ini membahas cara-cara mengevaluasi dan mengoptimalkan daya yang dikonsumsi oleh komponen, yang berperan dalam meningkatkan efisiensi energi dari sistem secara keseluruhan.



## **B. Sistem dan Teknik Rangkaian Elektronika**

Bagian ini membahas berbagai sistem dan teknik yang digunakan dalam perancangan rangkaian elektronika, baik analog maupun digital. Setiap teknik ini memungkinkan sistem berfungsi dengan lebih efektif sesuai dengan kebutuhan aplikasi.

### **1. Konsep Dasar Rangkaian Analog**

Rangkaian analog digunakan untuk memproses sinyal kontinu dan memainkan peran penting dalam berbagai aplikasi. Bagian ini menjelaskan prinsip-prinsip dasar yang digunakan dalam rangkaian analog, termasuk elemen-elemen utamanya dan penggunaannya.

### **2. Konsep Dasar Rangkaian Digital**

Rangkaian digital beroperasi dengan sinyal diskrit, cocok untuk pemrosesan informasi digital. Bagian ini menguraikan prinsip dasar rangkaian digital serta komponen-komponen utama yang mendukung fungsi-fungsi digital dalam proyek ini.

### **3. Teknik Pengolahan Sinyal pada Sistem Elektronika**

Pengolahan sinyal adalah proses penting untuk interpretasi informasi dari lingkungan. Bagian ini membahas metode umum dalam pengolahan sinyal yang diterapkan pada sistem elektronika, termasuk teknik yang digunakan dalam pemfilteran atau pemrosesan data.

### **4. Pengkabelan dan Pengaturan Sirkuit untuk Keandalan Sistem**

Pengkabelan dan tata letak yang baik meningkatkan keandalan sistem secara keseluruhan. Bagian ini menguraikan teknik pengkabelan dan pengaturan sirkuit yang efektif, serta bagaimana hal ini dapat mempengaruhi performa sistem.

### **5. Pengendalian dan Penggerak (Motor Driver, Relay, dsb.)**

Bagian ini menjelaskan penggunaan penggerak seperti motor driver dan relay untuk menggerakkan komponen mekanis. Diperlukan teknik pengendalian khusus untuk memastikan bahwa setiap penggerak bekerja sesuai dengan tujuan sistem.

### **C. Teknologi yang Digunakan**

Bagian ini mengulas teknologi yang umum digunakan dalam proyek berbasis elektronika, seperti mikrokontroler, sensor, aktor, dan teknologi komunikasi nirkabel. Teknologi ini memungkinkan sistem untuk merespons lingkungan dan berinteraksi dengan pengguna.

#### **1. Mikrokontroler dan Mikroprosesor**

Mikrokontroler dan mikroprosesor berfungsi sebagai unit pemrosesan utama dalam sistem elektronika. Bagian ini menguraikan arsitektur dasar, bahasa pemrograman yang relevan, serta protokol komunikasi yang sering digunakan dalam proyek ini.

#### **2. Sensor dan Aktuator**

Sensor dan aktuator memungkinkan interaksi sistem dengan lingkungannya. Bagian ini membahas jenis-jenis sensor yang digunakan, cara kerja, dan integrasinya ke dalam sistem agar sistem dapat mengumpulkan data dan merespons secara aktif.

#### **3. Teknologi Nirkabel**

Teknologi nirkabel seperti Bluetooth dan Wi-Fi memungkinkan komunikasi jarak jauh dalam sistem IoT. Bagian ini menguraikan jenis teknologi nirkabel yang relevan, termasuk protokol komunikasi dan aspek keamanan yang perlu dipertimbangkan.

### **D. Metode Kontrol dan Kecerdasan Buatan**

Bagian ini membahas metode kontrol dan kecerdasan buatan yang diterapkan dalam sistem elektronika untuk mencapai pengendalian yang lebih cerdas dan otomatis, seperti kontrol PID, logika fuzzy, dan deep learning.

#### **1. Pengendalian PID (Proportional-Integral-Derivative)**

PID adalah metode kontrol yang efektif dalam mengatur respons sistem. Bagian ini menjelaskan prinsip dasar PID, aplikasinya dalam pengaturan sistem elektronika, serta teknik tuning yang dapat meningkatkan stabilitas dan respons sistem.

## **2. Fuzzy Logic Control**

Logika fuzzy adalah metode kontrol fleksibel yang sering digunakan dalam sistem nonlinear. Bagian ini menjelaskan konsep dasar logika fuzzy, serta cara implementasi dan manfaatnya dalam pengendalian sistem yang kompleks.

## **3. Deep Learning**

Deep learning memungkinkan sistem untuk belajar dari data, yang sangat berguna dalam aplikasi otomatisasi. Bagian ini menguraikan algoritma dasar dalam deep learning, seperti CNN dan RNN, serta penerapannya dalam pengembangan sistem IoT.

## **4. Perbandingan dan Pemilihan Metode yang Sesuai**

Bagian ini membahas perbandingan antara berbagai metode kontrol yang tersedia, menjelaskan kelebihan dan kekurangannya masing-masing, serta bagaimana memilih metode yang paling sesuai untuk aplikasi proyek ini.

## **E. Konsep Engineering Design Process**

Engineering Design Process adalah metodologi sistematis yang digunakan untuk merancang sistem secara efektif. Bagian ini menguraikan prinsip utama dan langkah-langkah dari Engineering Design Process dalam konteks pengembangan sistem elektronika.

### **1. Pengertian dan Langkah-langkah Engineering Design Process**

Engineering Design Process adalah proses desain iteratif yang mencakup beberapa tahapan untuk mencapai desain yang optimal. Bagian ini menjelaskan langkah-langkah dasar yang terlibat dan bagaimana proses ini diadaptasi dalam proyek ini.

### **2. Aplikasi Engineering Design Process pada Proyek Elektronika**

Bagian ini membahas penerapan Engineering Design Process dalam proyek elektronika untuk mencapai hasil desain yang optimal. Diuraikan langkah-langkah praktis dalam menerapkan metodologi ini.

### **3. Studi Kasus Implementasi Engineering Design Process dalam Desain Elektronika**

Studi kasus ini menunjukkan contoh penerapan Engineering Design Process dalam desain sistem yang relevan dengan proyek. Dengan studi kasus ini, pembaca dapat memahami implementasi proses desain secara nyata.

### **4. Teknik Evaluasi dan Optimasi Desain**

Teknik evaluasi dan optimasi desain merupakan langkah penting dalam proses desain yang berkelanjutan. Bagian ini menjelaskan metode yang digunakan untuk mengevaluasi dan menyempurnakan desain agar mencapai hasil terbaik.

## **F. Penelitian Terdahulu yang Relevan**

Bagian ini berisi ulasan terhadap penelitian terdahulu yang relevan dengan proyek ini. Tujuannya adalah untuk melihat pendekatan yang telah digunakan, menemukan kelebihan dan kekurangannya, serta mengidentifikasi inovasi yang dapat dikembangkan.

### **1. Tinjauan Penelitian Terdahulu tentang Proyek Serupa**

Bagian ini membahas penelitian terdahulu yang serupa dengan proyek ini. Tinjauan ini bertujuan untuk memahami bagaimana proyek ini dapat memberikan kontribusi yang berbeda atau lebih baik.

### **2. Analisis Kekurangan dan Kelebihan Metode pada Penelitian Terdahulu**

Setiap metode yang digunakan dalam penelitian terdahulu memiliki kelebihan dan kekurangan. Bagian ini mengidentifikasi aspek yang perlu diperbaiki atau dikembangkan lebih lanjut berdasarkan analisis metode-metode tersebut.

### **3. Inovasi dan Kontribusi yang Dibawa dalam Penelitian Ini**

Penelitian ini membawa inovasi tertentu yang berkontribusi dalam memperkaya hasil penelitian sebelumnya. Bagian ini menjelaskan kontribusi utama proyek ini terhadap bidang elektronika, serta perbedaan yang ditawarkan.

## **BAB III**

### **KONSEP RANCANGAN ALAT DAN PENGUJIAN**

#### **A. Metode Pengerjaan Project Berbasis Engineering Design Process**

Bagian ini menjelaskan metode pengerjaan proyek yang mengadopsi pendekatan Engineering Design Process. Pendekatan ini digunakan untuk memastikan perancangan dan pengembangan proyek dilakukan secara sistematis, mulai dari identifikasi masalah hingga evaluasi akhir. Langkah-langkah dalam metode ini membantu dalam mencapai solusi yang optimal dan terukur.

##### **1. Identifikasi Masalah**

Tahap identifikasi masalah bertujuan untuk menguraikan permasalahan utama yang dihadapi dan memerlukan solusi. Pada tahap ini, dilakukan analisis untuk memahami aspek-aspek penting dari masalah dan menentukan faktor-faktor yang perlu diatasi melalui proyek ini.

##### **2. Definisi Kebutuhan**

Berdasarkan masalah yang telah diidentifikasi, tahap ini berfokus pada definisi kebutuhan proyek secara jelas dan terstruktur. Kebutuhan ini meliputi spesifikasi teknis, fungsi yang diinginkan, serta kriteria-kriteria lain yang harus dipenuhi agar solusi dapat berfungsi dengan baik.

##### **3. Generasi Ide dan Solusi**

Pada tahap ini, berbagai ide dan solusi alternatif dikembangkan dan dievaluasi. Bagian ini menjelaskan proses brainstorming untuk menghasilkan ide yang inovatif, termasuk analisis terhadap kelebihan dan kekurangan dari setiap alternatif solusi yang diusulkan.

##### **4. Perencanaan dan Desain Awal**

Desain awal sistem dikembangkan berdasarkan solusi yang dipilih, dengan mempertimbangkan aspek teknis dan kebutuhan yang telah didefinisikan. Bagian ini menyajikan perencanaan mengenai struktur sistem, alat, dan bahan yang akan digunakan, serta jadwal kerja proyek secara keseluruhan.

## **5. Pembuatan Prototipe**

Prototipe dibuat untuk menguji konsep dan desain awal dari sistem. Bagian ini menguraikan langkah-langkah dalam proses pembuatan prototipe, termasuk alat dan bahan yang diperlukan, serta tantangan yang mungkin dihadapi selama proses.

## **6. Pengujian dan Evaluasi**

Prototipe yang telah dibuat kemudian diuji untuk menilai kinerjanya terhadap kebutuhan dan spesifikasi yang telah ditetapkan. Bagian ini menjelaskan prosedur pengujian yang diterapkan, metode pengumpulan data, serta analisis terhadap hasil yang diperoleh.

## **7. Perbaikan dan Penyempurnaan**

Berdasarkan hasil pengujian dan evaluasi, dilakukan perbaikan untuk menyempurnakan sistem. Bagian ini menjelaskan penyesuaian yang dilakukan untuk meningkatkan kinerja sistem, serta proses iterasi yang dilakukan hingga mencapai hasil yang optimal.

## **B. Perancangan Sistem Elektronika**

Bagian ini menguraikan perancangan dari sisi elektronika yang menjadi inti dari sistem. Perancangan ini mencakup blok diagram, pemilihan komponen, dan perancangan rangkaian.

### **1. Blok Diagram Sistem**

Blok diagram sistem memberikan gambaran umum mengenai arsitektur sistem secara keseluruhan. Bagian ini menyajikan diagram beserta penjelasan fungsi setiap blok yang terdapat di dalam sistem, termasuk bagaimana setiap blok berinteraksi.

### **2. Pemilihan dan Spesifikasi Komponen Elektronika**

Pemilihan komponen elektronika dilakukan berdasarkan kebutuhan dari desain sistem. Bagian ini menjelaskan spesifikasi teknis dari setiap komponen yang digunakan, seperti mikrokontroler, sensor, aktuator, dan komponen pendukung lainnya.

### **3. Perancangan Rangkaian Elektronika**

Perancangan rangkaian elektronika bertujuan untuk mencapai fungsionalitas yang diinginkan dari sistem. Bagian ini menguraikan skema rangkaian, penjelasan aliran arus dan tegangan, serta hubungan antar komponen dalam sistem.

### **C. Perancangan Mekanik**

Bagian ini membahas perancangan mekanik dari sistem yang mendukung komponen elektronik secara fisik, termasuk struktur dan pemilihan bahan.

#### **1. Spesifikasi Desain Mekanik**

Desain mekanik disusun berdasarkan kebutuhan fisik sistem untuk memastikan bahwa komponen elektronika terlindungi dan dapat berfungsi dengan baik. Bagian ini menjelaskan spesifikasi teknis desain mekanik yang digunakan.

#### **2. Pemilihan Bahan dan Komponen Mekanik**

Pemilihan bahan didasarkan pada kriteria seperti kekuatan, ketahanan, dan biaya. Bagian ini menguraikan bahan dan komponen mekanik yang digunakan untuk konstruksi sistem, serta alasan pemilihan bahan tersebut.

#### **3. Desain Struktur dan Konstruksi**

Struktur dan konstruksi sistem dirancang untuk memberikan dukungan fisik yang stabil. Bagian ini menjelaskan proses desain dan konstruksi dari struktur mekanik sistem, serta tantangan yang mungkin dihadapi.

### **D. Perancangan Perangkat Lunak**

Bagian ini mencakup perancangan perangkat lunak yang mengendalikan sistem, meliputi diagram alir, pengembangan kode, dan pengujian perangkat lunak.

#### **1. Flowchart atau Diagram Alir Perangkat Lunak**

Diagram alir menggambarkan alur kerja dari perangkat lunak yang mengontrol sistem. Bagian ini menyajikan flowchart lengkap yang menunjukkan logika dan struktur kontrol dari perangkat lunak.

## **2. Pemrograman dan Pengembangan Kode**

Kode perangkat lunak dikembangkan untuk mendukung fungsionalitas sistem. Bagian ini menguraikan struktur dan logika dari program yang dibuat, bahasa pemrograman yang digunakan, serta strategi pengembangan kode.

## **3. Pengujian Kode Perangkat Lunak**

Setelah kode perangkat lunak dikembangkan, dilakukan pengujian untuk memastikan bahwa perangkat lunak berfungsi sesuai yang diharapkan. Bagian ini menjelaskan metode pengujian, jenis uji (misalnya, uji unit dan uji integrasi), serta hasil yang diperoleh.

## **E. Perancangan Integrasi Sistem**

Bagian ini menjelaskan proses integrasi antara komponen elektronik, mekanik, dan perangkat lunak agar sistem dapat bekerja sebagai satu kesatuan.

### **1. Integrasi Komponen Elektronika, Mekanik, dan Perangkat Lunak**

Proses integrasi bertujuan untuk menyatukan seluruh komponen sistem agar berfungsi sebagai satu kesatuan. Bagian ini menguraikan langkah-langkah dalam proses integrasi dan memastikan semua komponen bekerja secara sinkron.

### **2. Pengujian Awal dan Penyempurnaan Integrasi**

Pengujian awal dilakukan setelah integrasi untuk menilai performa sistem secara keseluruhan. Bagian ini menjelaskan hasil pengujian integrasi dan modifikasi yang diperlukan untuk meningkatkan kinerja sistem.

## **F. Rencana Pengujian**

Bagian ini merencanakan pengujian yang komprehensif terhadap sistem untuk memastikan bahwa semua komponen dan fungsi bekerja dengan baik.

### **1. Metode Pengujian Sistem**

Metode pengujian dipilih berdasarkan tujuan dan kebutuhan proyek. Bagian ini menjelaskan berbagai metodologi pengujian yang dirancang, seperti uji kinerja, uji stabilitas, dan uji kompatibilitas.



## **2. Prosedur Pengujian**

Prosedur pengujian disusun untuk menguji sistem secara menyeluruh, mulai dari pengaturan awal hingga pelaksanaan uji. Bagian ini menjelaskan langkah-langkah pengujian secara detail untuk menjamin konsistensi dan keandalan hasil.

## **3. Kriteria Keberhasilan Pengujian**

Kriteria keberhasilan ditentukan untuk mengevaluasi kinerja sistem berdasarkan parameter-parameter tertentu. Bagian ini menguraikan kriteria-kriteria keberhasilan yang digunakan, seperti ketepatan, keandalan, dan efisiensi sistem.

Pada bagian ini akan dijelaskan beberapa tahap utama dalam proses pengembangan proyek dan beberapa rekomendasi peningkatan yang dapat dilakukan. Daftar tahapan pengembangan proyek akan disajikan dalam bentuk bernomor untuk menunjukkan urutan logis dari proses, sementara rekomendasi peningkatan akan disajikan dalam bentuk daftar berpoin untuk mempermudah identifikasi setiap item secara mandiri.

## DAFTAR PUSTAKA

- Amershi, S., Weld, D., Vorvoreanu, M., Fournay, A., Nushi, B., Collisson, P., et al. (2019). Guidelines for human-ai interaction. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–13.
- Anthropic (2024). Model context protocol specification. <https://modelcontextprotocol.io/specification>. Accessed: 2025-01-15.
- Arifin, F., Nasuha, A., and Hermawan, H. D. (2015). Lip reading based on background subtraction and image projection. In *2015 International Conference on Information Technology Systems and Innovation (ICITSI)*, pages 1–3.
- Artac, M., Borovssak, T., Di Nitto, E., Guerriero, M., and Tamburri, D. A. (2017). Devops: Introducing infrastructure-as-code. In *2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*, pages 497–498. IEEE.
- Beyer, B., Jones, C., Petoff, J., and Murphy, N. R. (2016). *Site Reliability Engineering: How Google Runs Production Systems*. O’Reilly Media, Sebastopol, CA.
- Burns, B., Beda, J., and Hightower, K. (2018). *Kubernetes: Up and Running*. O’Reilly Media, Sebastopol, CA, 2nd edition.
- Burns, B., Grant, B., Oppenheimer, D., Brewer, E., and Wilkes, J. (2016). Borg, omega, and kubernetes. volume 14, pages 70–93. ACM.
- Cloud Native Computing Foundation (2024). Kubernetes documentation. <https://kubernetes.io/docs/>. Accessed: 2025-01-15.
- Corkill, D. D. (1991). Blackboard systems. *AI Expert*, 6(9):40–47.
- CrewAI Inc. (2024). Crewai: Framework for orchestrating role-playing ai agents. <https://docs.crewai.com/>. Accessed: 2025-01-15.
- Dang, Y., Lin, Q., and Huang, P. (2019). Aiops: Real-world challenges and research innovations. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, pages 4–5. IEEE.
- Dhewa, O. A., Dharmawan, A., and Priyambodo, T. K. (2017). Model of linear quadratic regulator (lqr) control method in hovering state of quadrotor. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 9(3):135–143.
- Dorri, A., Kanhere, S. S., and Jurdak, R. (2018). Multi-agent systems: A survey. *IEEE Access*, 6:28573–28593.
- Google (2024). Agent-to-agent (a2a) protocol specification. <https://github.io/A2A/>. Accessed: 2025-01-15.

- Google Cloud (2024). Google agent development kit (adk) documentation. <https://cloud.google.com/vertex-ai/docs/generative-ai/agents/adk>. Accessed: 2025-01-15.
- Gunawan, S. A., Pratama, G. N. P., Cahyadi, A. I., Winduratna, B., Yuwono, Y. C. H., and Wahyunggoro, O. (2019). Smoothed a-star algorithm for nonholonomic mobile robot path planning. In *2019 International Conference on Information and Communications Technology (ICOIACT)*, pages 654–658.
- Jennings, N. R., Sycara, K., and Wooldridge, M. (1998). A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems*, 1(1):7–38.
- LangChain Inc. (2024). Langchain: Building applications with llms through composability. <https://python.langchain.com/docs/>. Accessed: 2025-01-15.
- Lyu, Y., Li, H., Sheng, Q. Z., et al. (2021). Towards a unified framework for aiops. *IEEE Access*, 9:98288–98303.
- Mansur, S., Habib, M., Pratama, G. N. P., Cahyadi, A. I., and Ardiyanto, I. (2017). Real time monocular visual odometry using optical flow: Study on navigation of quadrotors uav. In *2017 3rd International Conference on Science and Technology - Computer (ICST)*, pages 122–126.
- Marpanaji, E., Yuwono, K. T., Mahali, M. I., Aji, P. T., and Nugraha, N. A. B. (2019). Experimental study of measuring radiation patterns for vhf and uhf antennas. *Journal of Physics: Conference Series*, 1413(1):012013.
- Morris, K. (2020). Infrastructure as code: Dynamic systems for the cloud age.
- Nasuha, A., Arifin, F., Sardjono, T., Takahashi, H., and Purnomo, M. (2017). Automatic lip reading for daily indonesian words based on frame difference and horizontal-vertical image projection. *Journal of Theoretical and Applied Information Technology*, 95(2):393–402.
- Notaro, P., Cardoso, J., and Gerndt, M. (2021). A survey of aiops methods for failure management. *ACM Transactions on Intelligent Systems and Technology*, 12(6):1–45.
- Ogata, K. (1987). *Discrete-Time Control Systems*. Prentice Hall, Australia, Sydney.
- Picoreti, R., do Carmo, A. B., de Queiroz, F. M., Garcia, A. S. A. C., Vassallo, R. F., and Simeonidou, D. (2018). Multilevel observability in cloud orchestration. In *2018 IEEE 16th International Conference on Dependable, Autonomic and Secure Computing*, pages 776–784. IEEE.
- Priambodo, A. S., Arifin, F., Nasuha, A., and Winursito, A. (2021). Face tracking for flying robot quadcopter based on haar cascade classifier and pid controller. *Journal of Physics: Conference Series*, 2111(1):012046.

- Priambodo, A. S. and Nugroho, A. P. (2021). Design & implementation of solar powered automatic weather station based on esp32 and gprs module. *Journal of Physics: Conference Series*, 1737(1):012009.
- Priyambodo, T. K., Dhewa, O. A., and Susanto, T. (2020). Model of linear quadratic regulator (lqr) control system in waypoint flight mission of flying wing uav. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 12(4):43–49.
- Shneiderman, B. (2020). Human-centered artificial intelligence: Reliable, safe & trustworthy. *International Journal of Human-Computer Interaction*, 36(6):495–504.
- Soldani, J., Tamburri, D. A., and Van Den Heuvel, W.-J. (2022). The pains and gains of microservices: A systematic grey literature review. *Journal of Systems and Software*, 146:215–232.
- Sridharan, C. (2018). *Distributed Systems Observability: A Guide to Building Robust Systems*. O’Reilly Media, Sebastopol, CA.
- Wooldridge, M. (2009). *An Introduction to MultiAgent Systems*. John Wiley & Sons, Chichester, UK, 2nd edition.
- Xi, Z., Chen, W., Guo, X., He, W., Ding, Y., Hong, B., et al. (2023). The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864*.

## **LAMPIRAN A**

### **KODE PROGRAM**

**Lampiran A.1. Program Pembacaan Sensor Ultrasonic**

**Lampiran A.2. Program Keseluruhan Proyek Akhir**

**LAMPIRAN B**  
**GAMBAR-GAMBAR**

**Lampiran B.1. Foto Aktivitas Kegiatan Proyek Akhir**



**Lampiran B.2. Foto Produk Proyek Akhir**

