ANALISA PERFORMA PENJADWALAN *TASK* FREERTOS PADA SISTEM PEMROSESAN DATA MPU6050 BERBASIS ESP32

DAVID SATIVA CHRISTIAN ASACANNO
H1051191048

SKRIPSI

PROGRAM STUDI REKAYASA SISTEM KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS TANJUNGPURA
2026

ANALISA PERFORMA PENJADWALAN *TASK* FREERTOS PADA SISTEM PEMROSESAN DATA MPU6050 BERBASIS ESP32

DAVID SATIVA CHRISTIAN ASACANNO

H1051191048

Skripsi
Sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer Pada Program Studi Rekayasa Sistem Komputer



PROGRAM STUDI REKAYASA SISTEM KOMPUTER

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS TANJUNGPURA

2026

LEMBAR PENGESAHAN

Judul Tugas Akhir	: Analisa Performa Penjadwalan *Task FreeRTOS* Pada Sistem Pemrosesan Data MPU6050 Berbasis ESP32
Nama Mahasiswa	: David Sativa Christian Asacanno
NIM	: H1051191048
Jurusan/Program Studi	: Rekayasa Sistem Komputer
Tanggal Lulus	: 29 Juli 2026
SK Pembimbing	: No.1111/DST/UN22.8/DT.00.99/2026/Tanggal 16 Maret 2026
SK Penguji	: No.4904/DST/UN22.8/DT.00.09/2026/Tanggal 28 2026

Dosen Pembimbing

Pembimbing I	Pembimbing II
Irma Nirmala, S.T., M.T.	Rahmi Hidayati, S.Kom., M.Cs.
NIP 198404052019032015	NIP 198607202015042001

Dosen Penguji

Ketua Penguji	Anggota Penguji
Drs. Cucu Suhery, M.A.	Tedy Rismawan, S.Kom., M.Cs
NIP 196108291989031002	NIP 198609222014041002

Pimpinan Sidang (merangkap anggota penguji)	Sekretaris Sidang (merangkap anggota penguji)
Irma Nirmala, S.T., M.T.	Rahmi Hidayati, S.Kom., M.Cs.
NIP 198404052019032015	NIP 198607202015042001

Mengesahkan
Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Tanjungpura

Prof. Dr. Gusrizal, S.Si., M.Si.
NIP. 197108022000031001

PERNYATAAN INTEGRITAS AKADEMIK

Saya yang bertanda tangan di bawah ini:

Nama	:	David Sativa Christian Asacanno
NIM	:	H1051191048
Program Studi/ Jurusan	:	Rekayasa Sistem Komputer
Fakultas	:	Matematika dan Ilmu Pengetahuan Alam

dengan ini menyatakan bahwa dokumen ilmiah Tugas Akhir yang disajikan ini tidak mengandung unsur pelanggaran integritas akademik sesuai Peraturan Menteri Pendidikan, Kebudayaan, Riset, dan Teknologi Republik Indonesia Nomor 39 Tahun 2021. Apabila di kemudian hari dokumen ilmiah Tugas Akhir ini mengandung unsur pelanggaran integritas akademik sesuai ketentuan perundangan tersebut, maka saya bersedia menerima sanksi akademik dan/atau sanksi hukum yang berlaku.

Demikian pernyatan ini untuk digunakan sebagaimana mestinya.

Pontianak, 27 Juli 2026



David Sativa Christian Asacanno

NIM H1051191048

Analisa Performa Penjadwalan *Task* FreeRTOS Pada Sistem Pemrosesan Data MPU6050 Berbasis ESP32

Abstrak

Perkembangan sistem tertanam modern yang menggunakan sensor *Inertial Measurement Unit* (IMU) menuntut kemampuan pemrosesan data multi*task*ing yang kompleks dan *real-time*. Sayangnya, mayoritas penelitian sebelumnya yang menggunakan mikrokontroler ESP32 lebih terfokus pada hasil akhir aplikasi, seperti tingkat akurasi gerakan, dan kurang menganalisis secara mendalam performa sistem waktu nyata seperti waktu respon, latensi, *jitter*, dan tingkat pemenuhan tenggat waktu (*deadline miss*) saat menghadapi variasi beban kerja. Oleh karena itu, penelitian ini bertujuan untuk mengevaluasi pengaruh konfigurasi penjadwalan *task*, prioritas, dan beban sistem terhadap parameter performa *real-time* pada sistem operasi *FreeRTOS* berbasis mikrokontroler ESP32 dan pemrosesan data sensor MPU6050. Metode penelitian yang digunakan adalah eksperimental terapan melalui perancangan sistem perangkat keras dan perangkat lunak multi*task*ing. Pengumpulan data dilakukan melalui MPU6050 melalui empat skenario yang melibatkan jeda *task*, jumlah *task* aktif, dan prioritas *task*. Pada pengujian sistem, peningkatan jeda *task* terbukti mampu menurunkan waktu respon dan penggunaan CPU, meskipun berdampak pada peningkatan *jitter*. Selain itu, modifikasi dengan menurunkan prioritas pada *task* akuisisi data sensor menyebabkan terhambatnya pemrosesan sistem secara keseluruhan, yang memicu tingginya *deadline miss* dan lonjakan penggunaan CPU. Pada kondisi beban komputasi maksimal, performa sistem mengalami penurunan secara drastis hingga mencapai 3547 kejadian *deadline miss* dengan tingkat penggunaan CPU sebesar 152,56%. Dapat disimpulkan bahwa arsitektur *FreeRTOS* pada ESP32 sangat efektif dalam mengelola penjadwalan multi*task*ing berbasis prioritas, namun performanya berbanding terbalik dengan intensitas beban komputasi yang diberikan.

Kata Kunci: ESP32, FreeRTOS, Penjadwalan *Task*, Sistem *Real-time*, MPU6050.

Analysis of FreeRTOS *Task Scheduling Performance in an* ESP32*-Based* MPU6050 *Data Processing System*

Abstract

The development of modern embedded systems utilizing Inertial Measurement Unit (IMU) sensors requires the capability to process complex multitasking operations in real time. However, most previous studies employing the ESP32 microcontroller have primarily focused on application outcomes, such as motion measurement accuracy, while providing limited analysis of real-time performance metrics, including response time, latency, jitter, and deadline miss under varying computational workloads. Therefore, this study aims to evaluate the effects of task scheduling configuration, task priority, and system workload on the real-time performance of a FreeRTOS-based ESP32 system for MPU6050 sensor data processing. An applied experimental method was employed through the design and implementation of both hardware and multitasking software systems. Data were collected from the MPU6050 sensor under four experimental scenarios involving variations in task delay, the number of active tasks, and task priority. The experimental results indicate that increasing task delay effectively reduced response time and CPU utilization, although it also resulted in higher jitter. Furthermore, lowering the priority of the sensor data acquisition task significantly delayed overall system processing, leading to an increase in deadline misses and CPU utilization. Under maximum computational workload conditions, the system performance degraded substantially, reaching 3,547 deadline miss events with CPU utilization of 152.56%. It can be concluded that the FreeRTOS architecture on the ESP32 is highly effective in managing priority-based multitasking scheduling; however, its performance is inversely proportional to the computational workload imposed on the system.

Keywords: ESP32, *FreeRTOS, Task Scheduling, Real-time System,* MPU6050.

PRAKATA

Puji dan syukur kepada Tuhan Yang Maha Esa, atas limpahan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul "Analisa Performa Penjadwalan *Task FreeRTOS* Pada Sistem Pemrosesan Data MPU6050 Berbasis ESP32" sebagai syarat mutlak bagi penulis untuk dapat memperoleh gelar Sarjana Komputer (S.Kom.) di Jurusan Rekayasa Sistem Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Tanjungpura, Pontianak. Sebagai ungkapan penghargaan, perkenankan penulis mengucapkan terima kasih kepada:

1. Bapak Kristian Taedo dan Ibu Utit yang merupakan kedua orang tua penulis, yang senantiasa memberikan doa, semangat, motivasi, dukungan moil maupun materil kepada penulis hingga akhirnya penulis dapat menyelesaikan tugas akhir skripsi ini.
2. Bapak Prof. Dr. Gusrizal, S.Si., M.Si., selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Tanjungpura.
3. Ibu Irma Nirmala, S.T., M.T., selaku Dosen Pembimbing Skripsi pertama, yang telah membimbing, memberikan segenap masukan serta dukungan positif kepada penulis selama proses penyusunan skripsi.
4. Ibu Rahmi Hidayati, S.Kom., M.Cs., selaku Ketua Jurusan Rekayasa Sistem Komputer sekaligus Dosen Pembimbing Skripsi kedua, yang telah banyak memberikan bimbingan, masukan, serta dukungan kepada penulis.
5. Bapak Ikhwan Ruslianto, S.Kom., M.Cs., selaku Dosen Pembimbing Akademik, yang telah banyak membantu dan memberikan dukungan kepada penulis sejak awal perkuliahan hingga penyelesaian skripsi.
6. Bapak Drs. Cucu Suhery, M.A., selaku Dosen Penguji pertama yang telah banyak memberikan masukan dan saran kepada penulis dalam tahap penyelesaian penulisan skripsi.
7. Bapak Tedy Rismawan, S.Kom., M.Cs., selaku Dosen Penguji kedua yang telah

   banyak memberikan masukan dan saran kepada penulis dalam tahap penyelesaian penulisan skripsi.
8.

9. Bapak dan Ibu Dosen, serta segenap staf Jurusan Rekayasa Sistem Komputer FMIPA UNTAN.

Penulis menyadari akan masih banyaknya kekurangan dalam tugas akhir ini. Oleh sebab itu, penulis sangat mengharapkan kritik dan saran yang membangun dari pembaca. Semoga, melalui karya tulis ini, penulis dapat memberikan manfaat bagi kemajuan ilmu pengetahuan, khususnya dalam bidang ilmu komputer.

Pontianak, 27 Juli 2026



David Sativa Christian Asacanno

NIM H1051191048

DAFTAR ISI

LEMBAR PENGESAHAN ii
PERNYATAAN INTEGRITAS AKADEMIK iii
Abstrak iv
Abstract v
PRAKATA vi
DAFTAR ISI viii
DAFTAR GAMBAR x
DAFTAR TABEL xi
DAFTAR KODE PROGRAM xii
BAB 1 PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 3
1.3 Batasan Masalah 3
1.4 Tujuan Penelitian 3
1.5 Manfaat Penelitian 4
1.6 Sistematika Penulisan 4
BAB 2 LANDASAN TEORI 6
2.1 Perbandingan Dengan Penelitian Terdahulu 6
2.2 Sistem Tertanam 7
2.3 ESP32 8
2.4 MPU6050 8
2.5 RTOS 9
2.6 FreeRTOS 10
2.7 Arduino IDE 10
2.8 Galat Relatif dan Akurasi 11
2.9 *Response Time* 11
2.10 *Latency* 12
2.11 *Jitter* 12
2.12 *Deadline miss* 12
2.13 *CPU Utilization* 12
BAB 3 METODE PENELITIAN 13
3.1 Studi Literatur 13
3.2 Metode Pengumpulan Data 14
3.3 Analisis Kebutuhan 14
3.3.1 Kebutuhan Perangkat Keras 14
3.3.2 Kebutuhan Perangkat Lunak 15

3.4 Perancangan Sistem....15
3.5 Implementasi....16
3.6 Pengujian....16
3.7 Kesimpulan dan Saran....17
BAB 4 PERANCANGAN....18
4.1 Deskripsi Sistem....18
4.2 Perancangan Arsitektur Sistem....18
4.3 Perancangan Perangkat Keras....20
4.3.1 Perancangan Koneksi....20
4.4 Perancangan Perangkat Lunak....20
BAB 5 IMPLEMENTASI PENGUJIAN DAN PEMBAHASAN....22
5.1 Implementasi Sistem....22
5.1.1 Implementasi Perangkat Keras....22
5.2 Kode Program....23
5.2.1 Kode Program Akselerometer....23
5.2.2 Kode Program *Task* Giroskop....24
5.2.3 Kode Program *Task* ProsesData....24
5.2.4 Implementasi *Task* Background Load 1....26
5.2.5 Implementasi Background Load *Task* 2....27
5.3 Pengujian....28
5.3.1 Pengujian Sensor MPU6050....28
5.3.2 Pengujian Performa Sistem....63
5.4 Pembahasan....65
BAB 6 PENUTUP....67
6.1 Kesimpulan....67
6.2 Saran....67
DAFTAR PUSTAKA....69

DAFTAR GAMBAR

Gambar 2. 1 ESP 32 8
Gambar 2. 2 MPU6050 9
Gambar 2. 3 Ilustrasi preemptive priority-based scheduling. 10
Gambar 2. 5 Halaman Utama Arduino IDE 11
Gambar 3. 1 Diagram Alir Penelitian 13
Gambar 4. 1 Diagram Alir Arsitektur Sistem 19
Gambar 4. 2 Gambar Perancangan Perangkat Keras 20
Gambar 5. 1 ESP32 dan MPU6050 yang sudah dirakit. 22
Gambar 5. 2 Menunjukkan posisi sensor dalam keadaan normal 29
Gambar 5. 3 Posisi Sensor Dibalik 180° 35
Gambar 5. 4 Gambar Sensor Dimiringkan 90° Ke Arah Kanan 40
Gambar 5. 5 Sensor Dimiringkan ke Kiri 90° 46
Gambar 5. 6 Sensor Di Miringkan 90° ke Arah Depan 52
Gambar 5. 7 Sensor di Miringkan ke Arah Belakang 90° 57

DAFTAR TABEL

Tabel 2. 1 Perbandingan dengan Penelitian Terdahulu.....6
Tabel 4. 1 Konfigurasi Pin ESP32 dan MPU6050 20
Tabel 5. 1 Tabel Data Sensor Posisi Normal 29
Tabel 5. 2 Perhitungan Error dan Akurasi Sensor Posisi Normal 32
Tabel 5. 3 Data Sensor Posisi Dibalik 180° 35
Tabel 5. 4 Perhitungan Error dan Akurasi Sensor Posisi Dibalik 180° 38
Tabel 5. 5 Data Sensor Dimiringkan Ke Arah Kanan 90° 41
Tabel 5. 6 Perhitungan Error dan Akurasi Sensor Posisi Dimiringkan ke Kanan 90° 43
Tabel 5. 7 Data Sensor Dimiringkan Ke Kiri 90° 46
Tabel 5. 8 Pengukuran Error dan Akurasi Sensor Dimiringkan 90° ke Arah Kiri .49
Tabel 5. 9 Data Sensor di Miringkan 90° ke Arah Depan 52
Tabel 5. 10 Pengukuran Error dan Akurasi Sensor di Miringkan 90° ke Arah Depan 54
Tabel 5. 11 Data Sensor di Miringkan ke Arah Belakang 90° 58
Tabel 5. 12 Pengukuran Error dan Akurasi Sensor di Miringkan ke Arah Belakang 90° 60
Tabel 5. 13 Hasil Pengujian Skenario 1 63
Tabel 5. 14 Hasil Skenario 2.....64
Tabel 5. 15. Hasil Pengujian Skenario 3 64
Tabel 5. 16 Hasil Pengujian Skenario 4 65

DAFTAR KODE PROGRAM

Kode Program 5. 1 Kode Task Akselerometer.....23
Kode Program 5. 2 Kode Program Task Giroskop 24
Kode Program 5. 3 Kode Program Task Monitoring.....26
Kode Program 5. 4 Kode Program Task Background Load 1 27
Kode Program 5. 5 Kode Program Task Background Load 2.....28

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Perkembangan sistem tertanam (*embedded system*) saat ini semakin pesat dan banyak digunakan pada berbagai bidang, seperti Internet of Things (IoT), sistem kesehatan, robotika, hingga sistem pemantauan aktivitas manusia. Salah satu komponen penting dalam sistem tertanam modern adalah penggunaan sensor *Inertial Measurement Unit* (IMU) yang terdiri dari *accelerometer* dan *gyroscope* untuk mendeteksi pergerakan, orientasi, serta perubahan posisi suatu objek atau pengguna. Sensor IMU banyak diaplikasikan pada sistem deteksi aktivitas, postur tubuh, navigasi, dan pemantauan gerak secara *real-time*(Sultan dkk, 2022).

Seiring dengan meningkatnya kompleksitas aplikasi berbasis sensor IMU, sistem operasi biasa sering kali tidak cukup untuk skenario dengan banyak tugas bersamaan dan prioritas berbeda, sehingga muncul kebutuhan akan RTOS yang menyediakan mekanisme *task* scheduling, manajemen prioritas, dan ketepatan waktu eksekusi. Pada kondisi ini, penggunaan *Real-time Operating System* (RTOS) menjadi sangat penting. RTOS memungkinkan pembagian tugas (*task*) secara terstruktur, pengaturan prioritas, serta penjadwalan eksekusi yang terprediksi sehingga sistem dapat memenuhi batasan waktu (*deadline*) yang telah ditentukan. Salah satu RTOS yang banyak digunakan pada sistem tertanam adalah *FreeRTOS* karena bersifat ringan, bersifat *open-source*, dan mendukung berbagai arsitektur mikrokontroler, termasuk ESP32.

ESP32 merupakan mikrokontroler yang banyak digunakan dalam pengembangan sistem tertanam karena memiliki kemampuan pemrosesan yang cukup tinggi, dukungan komunikasi yang lengkap, serta kompatibilitas yang baik dengan *FreeRTOS*. Dengan kemampuan *multicore* dan dukungan RTOS, ESP32 sangat cocok digunakan untuk aplikasi pemrosesan data sensor secara *real-time*. Namun demikian, implementasi *FreeRTOS* pada ESP32 tidak hanya bergantung pada fungsionalitas sistem, tetapi juga pada bagaimana penjadwalan *task* diatur agar sistem dapat bekerja secara optimal.

Dalam praktiknya, banyak penelitian dan implementasi sistem berbasis ESP32 dan IMU lebih menitikberatkan pada hasil akhir aplikasi, seperti tingkat

akurasi deteksi gerakan atau keberhasilan sistem dalam mengenali aktivitas tertentu. Sementara itu, aspek performansi sistem *real-time*, seperti waktu respon (*response time*), *latency*, *jitter*, dan tingkat kegagalan pemenuhan deadline (*deadline miss*), sering kali kurang dianalisis secara mendalam. Padahal, aspek-aspek tersebut sangat penting dalam sistem *real-time*, khususnya ketika sistem harus bekerja pada kondisi beban yang bervariasi.

Penelitian terkait sebelumnya adalah "Analisa Sistem Operasi *Real-time* pada Pengendalian Robot Industri: Evaluasi Performa dan Stabilitas" (Veliza & Saputra, 2025). Penelitian ini membandingkan performa *FreeRTOS* dan VxWorks dalam pengendalian robot industri, penelitian diukur dengan melihat waktu respon. Hasil dari pengujian adalah *FreeRTOS* bekerja dengan baik pada robot kecil, sedangkan VxWorks bekerja lebih baik pada robot besar.

Penelitian selanjutnya adalah "Analisis Penggunaan *FreeRTOS* Pada Konsep Multi*task*ing Berbasis Single Core" (Akbar dkk, 2024), penelitian yang dilakukan adalah membandingkan performa mikrokontroler Arduino Uno saat mengeksekusi 3 thread berbeda dengan *FreeRTOS* dan tanpa *FreeRTOS*. Hasil pengujian adalah dengan adanya *FreeRTOS* dalam 1 menit mampu mengeksekusi 437 *task*, sedangkan tanpa *FreeRTOS* hanya mampu mengeksekusi 122 *task*.

Penelitian selanjutnya berjudul "Peningkatan Kestabilan Sistem Kontrol UGV melalui Optimasi Manajemen Core dan *FreeRTOS* pada ESP32" (Azhar dkk, 2023). Penelitian ini menguji seberapa efisien pemanfaatan dual core dan *FreeRTOS* pada UGV(*Unmanned Grounded Vehicle*). Hasil menunjukkan waktu rata-rata sampling adalah 75,3 ms saat menggunakan *FreeRTOS* dan dual core, dan 88,9 ms saat tidak menggunakan *FreeRTOS* dan dual core.

Berdasarkan penelitian sebelumnya, penelitian ini difokuskan pada analisis performansi penjadwalan *task FreeRTOS* pada sistem pemrosesan data IMU berbasis ESP32 menggunakan lingkungan simulasi. Penelitian ini bertujuan untuk mengevaluasi bagaimana konfigurasi penjadwalan *task*, prioritas, dan beban sistem memengaruhi karakteristik sistem *real-time*. Hasil dari penelitian ini diharapkan dapat memberikan pemahaman yang lebih mendalam mengenai perilaku *FreeRTOS* pada ESP32 serta menjadi referensi dalam perancangan sistem tertanam berbasis sensor MPU6050 yang andal.

1.2 Rumusan Masalah

Berdasarkan permasalahan diatas, adapun permasalahan yang diangkat dalam penelitian ini adalah sebagai berikut.

1. Bagaimana implementasi FreeRTOS pada sistem pemrosesan data MPU6050 berbasis ESP32 dalam menjalankan beberapa task?
2. Bagaimana pengaruh konfigurasi prioritas dan periode *task* terhadap *response time*, *jitter*, dan *latency* sistem?
3. Bagaimana pengaruh penambahan beban sistem (*background load*) terhadap tingkat pemenuhan *deadline* pada *task* pemrosesan data IMU?

1.3 Batasan Masalah

Batasan masalah dibuat agar maksud dan tujuan dari penelitian dapat terfokus sesuai dengan tujuan. Adapun Batasan masalah dalam penelitian ini antara lain sebagai berikut.

1. Penelitian ini menggunakan mikrokontroler ESP32 sebagai platform sistem tertanam dengan sistem operasi *FreeRTOS*.
2. Sensor yang digunakan dibatasi pada sensor *Inertial Measurement Unit* (IMU), yaitu *accelerometer* dan *gyroscope*.
3. Analisis performansi difokuskan pada aspek sistem *real-time*, meliputi *response time*, *latency*, *jitter*.
4. Penjadwalan *task* yang dianalisis terbatas pada mekanisme penjadwalan yang disediakan oleh *FreeRTOS*, tanpa melakukan modifikasi terhadap kernel *FreeRTOS*.
5. Algoritma pemrosesan data IMU yang digunakan bersifat sederhana dan tidak berfokus pada peningkatan akurasi deteksi gerakan, melainkan pada pengaruhnya terhadap performansi sistem *real-time*.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut.

1. Mengimplementasikan FreeRTOS pada sistem pemrosesan data MPU6050 berbasis ESP32 sehingga mampu menjalankan beberapa task secara paralel.
2. Mengevaluasi pengaruh pengaturan prioritas dan periode *task* terhadap

parameter *real-time* seperti *response time*, *latency*, dan *jitter*.

3. Mengetahui dampak beban sistem terhadap kemampuan *FreeRTOS* dalam memenuhi batasan waktu (*deadline*) pada sistem pemrosesan data IMU.

1.5 Manfaat Penelitian

Manfaat dari adanya penelitian ini adalah agar menjadi acuan dalam merancang sistem tertanam berbasis ESP32 dan sensor IMU yang memiliki kebutuhan waktu nyata (*real-time constraint*). Analisis yang dilakukan dapat membantu pengembang dalam menentukan konfigurasi penjadwalan *task*, prioritas, dan beban sistem yang sesuai agar sistem dapat bekerja secara deterministik dan andal.

1.6 Sistematika Penulisan

Dalam rangka penulisan skripsi ini, disusun suatu sistematika penulisan dengan membaginya menjadi beberapa BAB. Susunan sistematika tersebut adalah:

BAB I. PENDAHULUAN

Menjelaskan tentang latar belakang permasalahan, tujuan dilakukannya penelitian, manfaat yang dapat diberikan dari penelitian, perumusan masalah, batasan masalah, hipotesis, dan sistematika penulisan.

BAB II. LANDASAN TEORI

Bagian ini berisi tabel perbandingan dengan penelitian terdahulu, teori-teori sistem mikrokontroler ESP32, sensor MPU6050, *FreeRTOS* yang digunakan dalam menyelesaikan penelitian ini.

BAB III. METODE PENELITIAN

Bab ini menjelaskan prosedur dan tahapan penelitian yang dilakukan agar proses penyusunan tugas akhir berjalan secara sistematis dan terstruktur. Pembahasan diawali dengan pemaparan diagram alir penelitian yang mencakup studi literatur, pengumpulan data, hingga analisis kebutuhan perangkat keras maupun perangkat lunak. Tahapan dilanjutkan dengan menguraikan proses perancangan, implementasi, dan pengujian sistem. Kemudian, ditutup dengan tahap pembahasan, serta penarikan kesimpulan dan saran dari keseluruhan penelitian.

BAB IV. PERANCANGAN

Bagian ini berisi tentang perancangan alat serta pemograman di dalamnya.

BAB V. IMPLEMENTASI, PENGUJIAN, DAN PEMBAHASAN

Bagian ini tentang hasil dari alat yang sudah dibuat, pengujiannya, serta menganalisa hasil dari pengujian.

BAB VI. PENUTUP

Bab ini akan menyimpulkan semua kegiatan dan hasil-hasil yang diperoleh selama proses perancangan dan pembuatan alat. Diberikan juga saran-saran yang perlu dipertimbangkan dalam upaya pengembangan lebih lanjut.