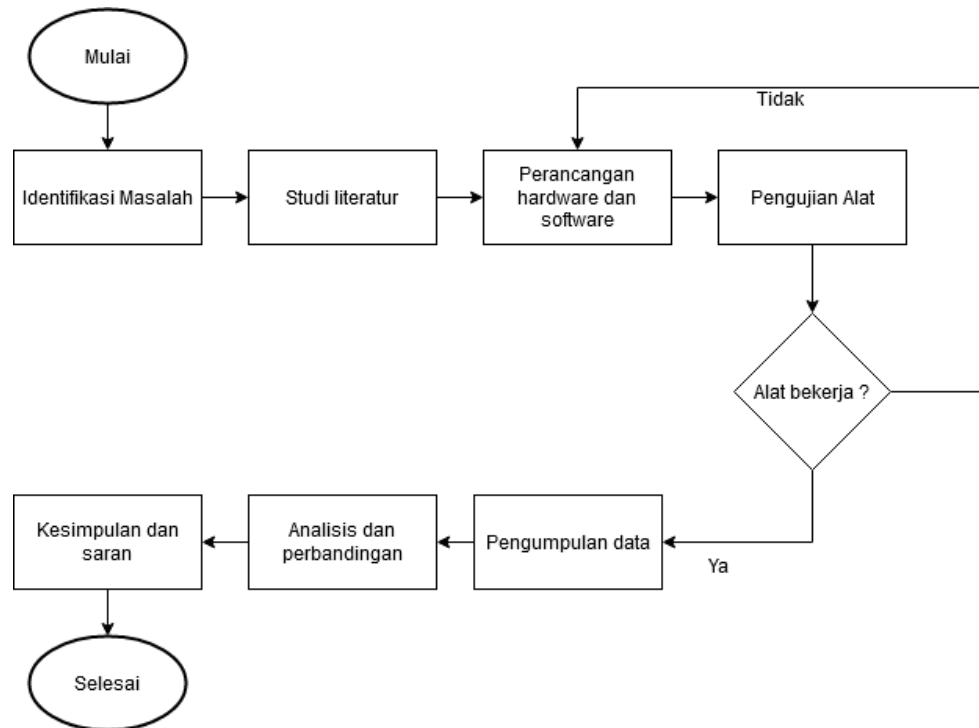


## BAB 3

### METODE PENELITIAN

#### 3.1 Tahapan Penelitian

Penelitian ini dilakukan dengan penulis menyusun tahapan-tahapan yang akan dilakukan penulis guna mencapai tujuan awal penelitian. Penyusunan tahapan penelitian dapat dilihat dalam bentuk *flowchart* sebagai berikut:



Gambar 3. 1 Flowchart Tahapan Penelitian

Sumber : Dokumen Pribadi

##### 3.1.1 Identifikasi Masalah

Pada tahap ini, identifikasi masalah digunakan untuk menentukan permasalahan yang dikaji pada penelitian penulis. Topik yang digunakan dalam penelitian ini mengenai uji performansi *core* pada mikrokontroler ESP32 menggunakan metode *Real-Time Operating System* (RTOS). Permasalahan yang ditentukan akan dicari solusinya sehingga menjadi tujuan dari penelitian ini.

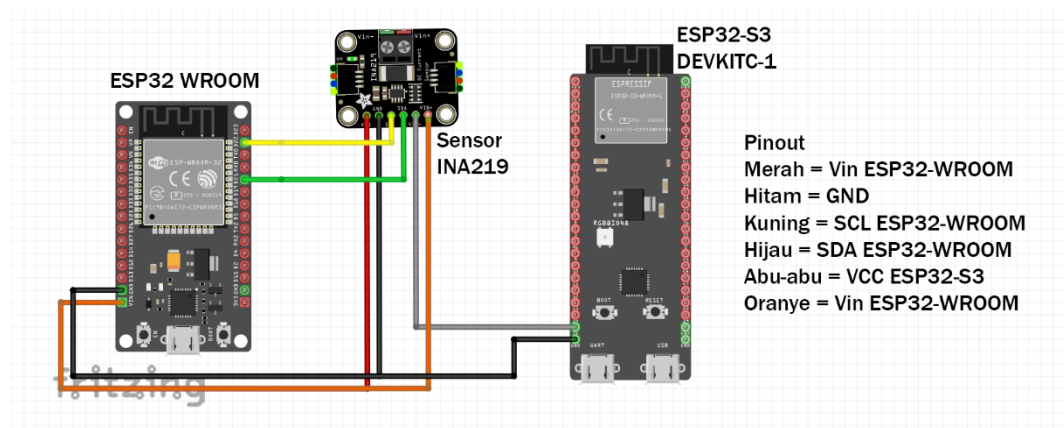
### 3.1.2 Studi Literatur

Studi literatur dilakukan oleh penulis untuk mempelajari dan melakukan pemahaman mengenai masalah uji performansi *core*. Melalui studi literatur, penulis dapat memperoleh informasi yang berguna dan mendalam tentang topik penelitian yang dilakukan.

### 3.1.3 Perancangan *Hardware* dan *Software*

#### 3.1.3.1 Perancangan *Hardware*

Pada Gambar 3.2, dilakukan perancangan *hardware* untuk mendukung penelitian yang terdiri dari ESP32-WROOM, Sensor INA219, ESP32-S3 DEVKITC-1, *Printed Circuit Board* (PCB).



Gambar 3. 2 Skema Perancangan Hardware

Sumber : Dokumen Pribadi

Tujuan dibentuknya perancangan ini adalah untuk memudahkan mendapatkan data dan menganalisis uji performansi *core* terhadap mikrokontroler yang dipakai sesuai dengan tujuan penelitian. Data yang diambil dari rancangan *hardware* ini meliputi kecepatan prosesor, respon waktu eksekusi, suhu operasional, dan konsumsi daya.

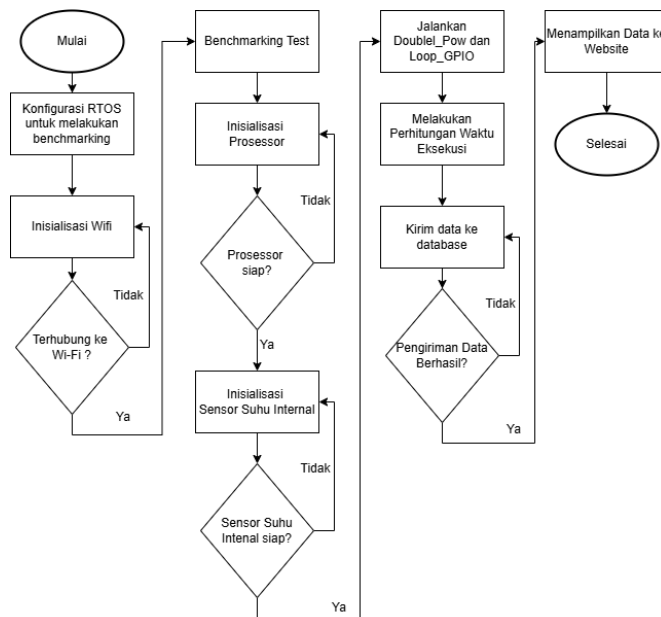
#### 3.1.3.2 Perancangan *Software*

Perancangan *software* difokuskan untuk mendukung pengoperasian perangkat untuk mengelola tugas yang akan diberikan pada mikrokontroler ESP32, mengirimkan kedalam

*database*, dan menampilkan data dalam *website*. Berikut adalah aplikasi yang digunakan :

### 3.1.3.2.1 Arduino IDE

Arduino IDE digunakan sebagai platform utama untuk merancang dan mengembangkan pemrograman mikrokontroler. Dalam konteks ini, IDE tersebut mendukung pemrograman untuk dua jenis mikrokontroler yang berbeda, termasuk ESP32.



Gambar 3. 3 Alur Pemrograman

Sumber : Dokumen Pribadi

Tahapan perancangan *software* dimulai dengan konfigurasi RTOS untuk mendukung pengujian *benchmarking* pada ESP32, terutama dalam pemanfaatan *dual-core*. Setelah itu, sistem menginisialisasi koneksi *Wi-Fi* dan memastikan koneksi berhasil sebelum melanjutkan ke tahap berikutnya. Pengujian dilakukan melalui inisialisasi prosesor dan sensor suhu internal, dengan memastikan semua komponen siap untuk mendukung proses pengukuran performansi dan suhu operasional. Setelah kode dijalankan pada mikrokontroler, hasil

pengujian performansi dikirim ke *database* dan serial monitor, yang digunakan untuk memantau data secara *real-time*, memudahkan analisis dan evaluasi performansi ESP32 secara efektif.

#### **3.1.3.2.2 Database MySQL**

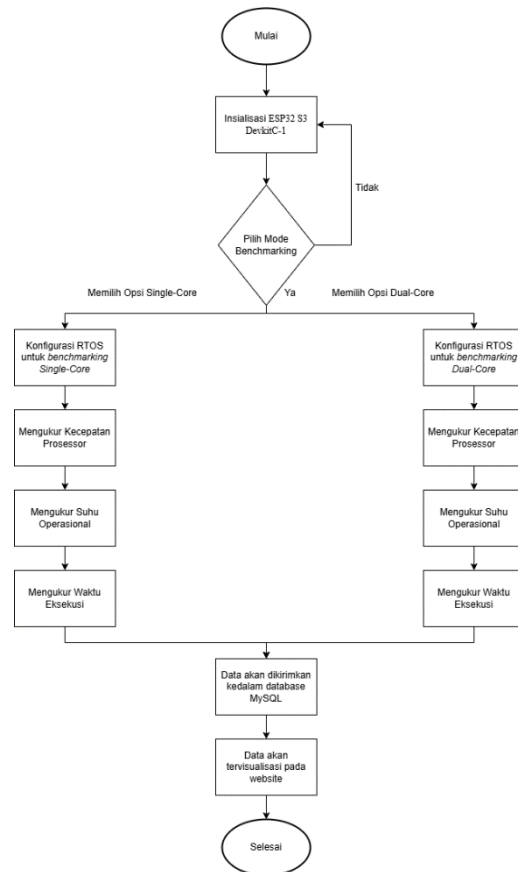
*Database* MySQL digunakan untuk menyimpan dan mengelola data yang dikirimkan dari hasil pengumpulan data. *Database* ini berfungsi sebagai media penyimpanan data terpusat yang mana nantinya akan dihubungkan kedalam *website*, sehingga data dapat ditampilkan dengan baik.

#### **3.1.3.2.3 Website**

*Website* yang efektif dan efisien sangat penting untuk memberikan pengguna akses cepat dan mudah terhadap informasi penting. *Website* berfungsi sebagai penampil data yang relevan secara *real-time* dari *database*, memungkinkan pengguna untuk membuat keputusan yang tepat berdasarkan data yang ada. Dalam proyek ini, kami akan merancang halaman *Website* menggunakan *database* MySQL untuk menyimpan dan mengelola data yang akan ditampilkan.

#### **3.1.4 Pengujian Alat**

Pengujian dilakukan untuk memeriksa rancangan *software* dan *hardware* sudah sesuai dengan yang diinginkan atau tidak. Pengujian pada penelitian ini ditinjau berdasarkan rancangan perangkat keras yang sudah sesuai dan tidak mengalami malfungsi secara fisik dengan melihat kondisi mikrokontroler dan sensor, respon sensor yang tertanam ditinjau dengan melihat apakah terjadi *error code* atau tidak dan juga melihat akses pengiriman data dari mikrokontroler ke jaringan lokal, pada pengiriman ke *website* di *web server*, serta tampilan *website*.



Gambar 3. 4 Alur Pengujian *Benchmarking* ESP32

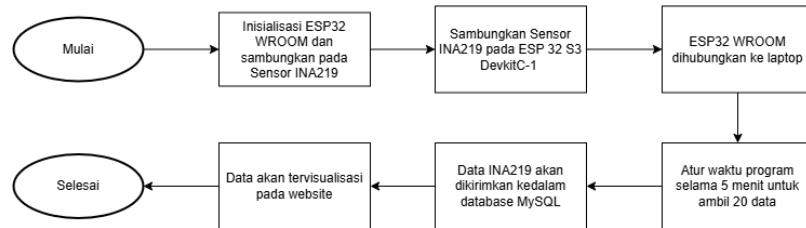
Sumber : Dokumen Pribadi

Berikut merupakan tahapan dalam melakukan *benchmarking* pada mikrokontroler ESP32-S3 DevkitC-1 :

1. Perangkat laptop sudah dinyalakan dan telah mengaktifkan *localhost* pada laptop.
2. Melakukan inisialisasi pada mikrokontroler ESP32-S3 DevkitC-1 dan memastikan sumbernya berasal dari sensor INA219.
3. Sistem menawarkan pilihan mode *benchmarking* :
  - a. *Single-Core*
  - b. *Dual-Core*
4. Mengatur konfigurasi RTOS untuk menjalankan *single-core* ataupun *dual-core*.
5. Melakukan *benchmarking* untuk mengambil data kecepatan prosesor (MHz), respon waktu eksekusi (s), suhu operasional

(°C) baik dalam mode *benchmarking single-core* ataupun *dual-core*.

6. Data yang telah diperoleh dari hasil *benchmarking single-core* ataupun *dual-core* akan dikirimkan ke *database MySQL*.
7. Data yang telah disimpan akan tervisualisasikan ke dalam *website*.



Gambar 3. 5 Alur Pengukuran Konsumsi Daya

Sumber : Dokumen Pribadi

Adapun tahapan dalam mengukur konsumsi daya menggunakan sensor INA219 sebagai berikut :

1. Perangkat laptop sudah dinyalakan dan telah mengaktifkan *localhost* pada laptop.
2. Menyambungkan sensor INA219 ke ESP32 WROOM dan sambungkan juga sensor INA219 ke ESP32-S3 DevkitC-1 sebagai sumber masukan.
3. Melakukan inisialisasi pada mikrokontroler ESP32 WROOM dan hubungkan ke laptop.
4. Melakukan pengukuran konsumsi daya (Watt) pada saat *benchmarking* berlangsung.
5. Data yang diperoleh dari hasil pembacaan sensor INA219 akan dikirimkan ke *database MySQL*.
6. Data yang telah disimpan akan tervisualisasikan ke dalam *website*.

### 3.1.5 Pengumpulan Data

Setelah alat atau sistem diuji dan dinyatakan berfungsi dengan baik, tahap selanjutnya adalah melakukan pengumpulan data. Pengumpulan data dilakukan sebanyak 100 kali setiap interval 5 menit. Pengumpulan

data meliputi variabel kecepatan prosesor dengan satuan MegaHertz (MHz), respon waktu eksekusi dengan satuan detik (s), Suhu operasional *board* (°C), dan Konsumsi Daya dengan satuan watt (W). Lalu hasil baca sensor tersebut akan dikirimkan ke jaringan lokal dan akan ditampilkan pada halaman *website*.

### 3.1.6 Pengolahan dan Analisis Data

Pengelolaan data hasil pengumpulan akan dimasukkan ke dalam aplikasi pengelolaan numerik. Data kecepatan prosesor, respon waktu eksekusi, dan suhu operasional *board* akan dikirim langsung melalui jaringan lokal ke database MySQL. Data arus yang awalnya dalam satuan *milliampere* (mA) akan dikonversi ke *ampere* (A), data tegangan dengan satuan *volt* (V), dan data konsumsi daya akan dikonversi menjadi *watt* (W). Selanjutnya, data konsumsi daya akan dianalisis untuk membandingkan dan menganalisis performansi *core* pada ESP32.

#### a. Verifikasi Data

Analisis standar deviasi dan standar kesalahan dilakukan untuk menilai sebaran data dari nilai rata-rata serta mengukur tingkat akurasi rata-rata tersebut dalam mewakili hasil pengukuran. Nilai standar deviasi yang kecil menunjukkan bahwa data lebih konsisten dan stabil, sedangkan nilai standar kesalahan yang kecil menunjukkan bahwa rata-rata data lebih akurat[21]. Dengan demikian, data yang memiliki standar deviasi dan standar kesalahan rendah dapat dipercaya untuk validasi atau perbandingan performansi sistem. Rumus yang digunakan untuk mengukur standar deviasi dapat dilihat pada persamaan 3.1, 3.2, 3.3. [22].

$$\text{Standar Deviasi} = \sqrt{\frac{\sum (x - \bar{x})^2}{n-1}} \quad (3.1)$$

$$\text{Standar Kesalahan} = \frac{\text{Standar Deviasi}}{\sqrt{n}} \quad (3.2)$$

$$\text{Presisi} = (1 - \text{Standar Kesalahan}) \times 100\% \quad (3.3)$$

\*Keterangan :

x : Variabel pengujian

$\bar{x}$  : Rata-rata dari variabel pengujian

n : Jumlah data

#### b. Persentase Perbandingan

Persentase perubahan atau perbandingan untuk setiap variabel (misalnya, konsumsi daya, respon waktu eksekusi, suhu operasional) dihitung menggunakan persamaan 3.4 [23].

$$\text{Persentase} = \frac{(\bar{X} \text{ Nilai } dualcore - \bar{X} \text{ Nilai } singlecore)}{\bar{X} \text{ Nilai } singlecore} \times 100\% \quad (3.4)$$

\*Keterangan :

Persentase = Hasil perbandingan (%)

$\bar{X}$  Nilai *dualcore* = Rata-rata nilai *dual-core*

$\bar{X}$  Nilai *singlecore* = Rata-rata nilai *single-core*

Hasil analisis ini bertujuan untuk mengungkap perbedaan performansi yang signifikan antara *single-core* dan *dual-core* pada ESP32, serta menganalisis efektivitas sistem dan akurasi pengukuran dengan sensor INA219 dalam mengukur arus dan tegangan. Fokus utama adalah untuk menunjukkan dampak nyata dari penggunaan *dual-core* dalam meningkatkan kinerja,



serta memastikan bahwa pengukuran arus dan tegangan dapat diandalkan untuk keperluan analisis daya yang lebih tepat.

### 3.1.7 Kesimpulan dan Saran

Kesimpulan dan saran adalah proses akhir dalam penelitian ini. Kesimpulan berisi hasil-hasil penelitian yang sesuai dengan tujuan penelitian, untuk mengetahui performansi *core* pada ESP. Saran berisi perbaikan baik dalam pelaksanaan penelitian, maupun pada kendala-kendala yang terjadi pada penelitian.

## 3.2 Jenis Penelitian

Jenis penelitian yang akan dilakukan adalah penelitian kuantitatif dengan metode eksperimental. Peneliti memilih metode ini karena alat yang dirancang akan melihat respon antara empat variabel. Eksperimen yang akan dilakukan melibatkan analisis kecepatan prosesor, respon waktu eksekusi, suhu operasional, dan konsumsi daya. Eksperimen ini menggunakan alat ukur berupa sensor INA219 yang dapat mengukur konsumsi daya. Tujuannya adalah untuk melihat bagaimana performansi pada mikrokontroler ESP32 ketika dijalankan *single-core*nya dan *dual-core*nya.

## 3.3 Variabel Penelitian

### 3.3.1 Variabel Independent

Variabel independen, sering disebut juga sebagai variabel bebas, variabel yang mempengaruhi. Variabel bebas juga dapat diartikan sebagai suatu kondisi atau nilai yang jika muncul maka akan memunculkan (mengubah) kondisi atau nilai yang lain[24]. Variabel *independent* dalam penelitian ini adalah konfigurasi *core*, yang mencakup dua kondisi, yaitu *single-core* dan *dual-core*. Perubahan pada konfigurasi *core* ini memungkinkan peneliti untuk menganalisis bagaimana perubahan tersebut mempengaruhi kinerja ESP32 dalam berbagai aspek yang diukur.

### 3.3.2 Variabel Dependent

Variabel dependen (variabel terikat) adalah variabel yang secara struktur berpikir keilmuan menjadi variabel yang disebabkan oleh adanya perubahan variabel lainnya[24]. Variabel *dependent* dalam

penelitian ini adalah respon waktu eksekusi, suhu operasional *board*, dan konsumsi daya. Respon waktu eksekusi mengacu pada waktu yang dibutuhkan untuk menyelesaikan tugas-tugas paralel yang diberikan dalam lingkungan RTOS, dan pengukurannya dilakukan untuk konfigurasi *single-core* dan *dual-core*. Suhu operasional *board* mengukur suhu kerja ESP32 selama eksekusi tugas, untuk memahami bagaimana konfigurasi *core* mempengaruhi panas yang dihasilkan oleh mikrokontroler. Konsumsi daya mengukur energi yang digunakan oleh ESP32 selama menjalankan tugas, untuk mengetahui efisiensi energi dari kedua konfigurasi *core*. Untuk memastikan bahwa hasil penelitian hanya dipengaruhi oleh perubahan pada konfigurasi *core*, sejumlah variabel dikendalikan agar tetap konstan.

### 3.3.3 Variabel Terkontrol

Variabel kontrol (*control variable*) merupakan variabel yang mengontrol pengaruh variabel bebas terhadap variabel tak bebas. Variabel ini, kualitas dan kuantitasnya bisa dikendalikan oleh peneliti sesuai dengan waktu dan tempat yang dikehendaki[24]. Variabel-variabel terkontrol dalam penelitian ini meliputi tugas yang dijalankan, perangkat keras dan perangkat lunak, serta catu daya. Tugas atau beban kerja yang diberikan pada ESP32 harus sama baik untuk *single-core* maupun *dual-core*, memastikan perbandingan kinerja yang adil dan setara. Versi perangkat keras ESP32, sensor INA219, serta versi perangkat lunak RTOS dan kode yang dijalankan harus sama untuk kedua konfigurasi, menghindari variabilitas yang tidak diinginkan dari versi yang berbeda. Tegangan dan sumber daya yang digunakan dijaga konstan untuk kedua konfigurasi *core*, penting untuk memastikan bahwa perubahan konsumsi daya benar-benar disebabkan oleh konfigurasi *core* dan bukan oleh variasi catu daya.

### 3.4 Instrument Penelitian

Adapun perangkat keras yang digunakan dalam penelitian ini adalah :

#### a. Laptop

- 1) Model sistem : ASUS PRO P1440FA
- 2) Prosesor : Intel Core i3 Gen 6
- 3) Memori : 8GB
- 4) Storage : 1.25TB

#### b. Daftar Komponen

Pembuatan alat penelitian ini melibatkan beberapa komponen utama yang mendukung proses pengembangan. Spesifikasi dari masing-masing komponen yang digunakan disajikan pada tabel berikut.

No.	Nama Komponen	Qty	Spesifikasi
1	ESP32 WROOM	1	<ul style="list-style-type: none"><li>• Daya input : 3.3VDC – 5VDC</li><li>• Arus Operasional : 80mA – 240mA</li><li>• Dimensi : 18mm x 25.5mm x 3.10 mm</li></ul>
2	ESP32-S3 DevkitC-1	1	<ul style="list-style-type: none"><li>• Daya input : 3.3VDC – 5VDC</li><li>• Arus Operasional : 80mA – 240mA</li><li>• Dimensi : 54mm x 27mm</li></ul>
3	Sensor INA219	1	<ul style="list-style-type: none"><li>• Daya input : 3VDC – 5.5VDC</li><li>• Pengukuran tegangan : 0-26VDC</li><li>• Pengukuran arus : 0 – 3.2A</li></ul>

			<ul style="list-style-type: none"> <li>• Dimensi : 0.8inch x 0.9inch</li> </ul>
--	--	--	---

### 3.5 Tempat Penelitian

Tempat pelaksanaan penelitian ini adalah di lingkungan Fakultas Teknik UPN Veteran Jakarta Kampus Limo, khususnya di ruang laboratorium elektronika dan kendali. Penulis memilih lingkungan tersebut karena terdapat fasilitas pendukung penelitian.

### 3.6 Jadwal Penelitian

Berikut ini jadwal penelitian kegiatan pelaksanaan dalam bentuk tabel *timeline* dibawah ini:

Tabel 3. 1 Jadwal Penelitian

Waktu Kegiatan	Bulan Ke 1				Bulan Ke 2				Bulan Ke 3				Bulan Ke 4				Bulan Ke 5			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Identifikasi Masalah																				
Studi Literatur																				
Prancangan Hardware dan Software																				
Pengujian Alat																				
Pengumpulan Data																				
Hasil dan Pembahasan																				
Kesimpulan																				