

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Dispenser *Earplug* Otomatis Berbasis *Radio Frequency Identification* (RFID) dan ESP32 merupakan teknologi inovatif yang menjanjikan untuk melakukan efisiensi penggunaan alat pelindung diri di sektor industri. Sistem ini menawarkan solusi yang lebih efisien untuk melakukan pemantauan dan pemberian jatah alat pelindung diri khususnya *earplug*. Sebelum mengembangkan lebih jauh sistem ini, penting untuk memahami konteks dan landasan teori yang telah dibangun oleh peneliti-peneliti sebelumnya. Hal ini tidak hanya membantu dalam mengidentifikasi celah dalam pengetahuan yang ada, tapi juga dalam menentukan arah yang harus diambil oleh penelitian ini.

- 1) Berdasarkan penelitian yang dilakukan oleh Muhammad Wisnu, Abdul Harris, dan Hendri pada Jurnal Manajemen Teknologi dan Sistem Informasi (JMS) dengan judul “Perancangan Sistem Informasi Absensi Guru Menggunakan RFID Pada SMP N 9 Kota Jambi Berbasis Web” dijelaskan bahwa pembacaan RFID dapat disimpan pada *localhost* yang dengan menggunakan *mysql*. Untuk mengakses *localhost* diperlukan pemrograman menggunakan bahasa PHP. Bahasa PHP digunakan untuk membaca data pada RFID dan menyimpan serta sinkronisasi dengan data yang ada di server *localhost*. *Dashboard* yang digunakan untuk memperbarui data diprogram menggunakan bahasa HTML yang kemudian dilakukan pemanggilan dengan bahasa PHP agar data yang berada di server *localhost* dapat ditampilkan atau diubah pada *dashboard*. Namun pada penelitian ini belum ditemukan adanya fungsi untuk *export* data dari *database*, sehingga untuk melakukan pengolahan data lebih lanjut diperlukan proses tambahan. [2]
- 2) Berdasarkan penelitian yang dilakukan oleh Suliswaningsih, Nugie Dwitama, dan Anugerah Bagus Wijaya pada Jurnal Infotekmesin dengan judul “Perancangan Sistem Presensi Siswa dengan RFID Berbasis IoT Menggunakan NodeMCU ESP8266” dijelaskan bahwa perancangan sistem presensi dengan

metode *waterfall* bisa dilakukan. Metode ini menerapkan tahapan untuk mengembangkan sebuah sistem dimana tahapan akan diproses setelah tahapan sebelumnya selesai diproses. Selain itu dalam penelitian ini dijelaskan juga diagram aktivitas sistem presensi secara rinci. Penelitian ini memberikan referensi NodeMCU ESP8266 untuk dijadikan mikrokontroler utama dalam mengolah data yang dibaca oleh RFID. Untuk sistem server, penelitian ini menggunakan server *localhost* dengan *basic* bahasa PHP. Salah satu kelebihan dalam jurnal ini adalah penulis menyertakan wiring diagram alat yang dibuat sehingga dapat dijadikan referensi penulis. Namun dalam penelitian ini masih menggunakan *database localhost* yang mengharuskan laptop sebagai server terus menyala. [3]

- 3) Berdasarkan penelitian yang dilakukan oleh Rizky Fenaldo Maulana, Muhammad Akbar Ramadhan, Wiranti Maharani, dan Muhammad Iqbal Maulana pada jurnal *Indonesian Journal of Multidisciplinary on Social and Technology* dengan judul “Rancang Bangun Sistem Monitoring Suhu dan Kelembapan Berbasis IoT Studi Kasus Ruang Server IT Telkom Surabaya” dijelaskan bahwa pengiriman data IoT dapat dilakukan menggunakan web Thingier.io. Web tersebut dapat mengolah data sensor yang dikirimkan tanpa menggunakan *database* dan dapat menampilkannya pada *dashboard* secara langsung. Namun kelemahannya adalah pilihan tampilan *dashboard* terbatas karena *user* tidak dapat melakukan kustomisasi via *coding* seperti yang dilakukan pada *visual studio code*. Selain itu jurnal ini menjelaskan juga koneksi antara telegram dan *dashboard* yang sudah dibuat. Sehingga monitoring dapat dilakukan via bot telegram. [4]
- 4) Berdasarkan penelitian yang dilakukan oleh Arief Hediansah, Desi Nurnaningsih, dan Hengki Rusdianto pada jurnal Teknoinfo dengan judul “Pemanfaatan Flutter Pada Pengembangan Aplikasi *Mobile Ebisnis* Penyediaan Bahan Baku Bisnis Katering” dijelaskan bahwa Flutter merupakan sebuah *toolkit User Interface* (UI) yang dikembangkan oleh Google dan dapat digunakan untuk membangun prototipe sistem dengan tampilan menarik. Flutter memungkinkan pengembangan *interface* aplikasi pada perangkat

smartphone maupun berbasis *web* hanya dari satu basis kode (*single codebase*). Pada penelitian yang dilakukan, penulis telah menghasilkan sebuah perancangan dan pengembangan prototipe ebisnis pengelolaan bahan baku bisnis katering dan telah dilakukan sebuah pengujian dengan metode *black box testing*. Metode tersebut menguji fungsionalitas perangkat lunak tanpa memperhatikan struktur internal atau kodenya. Namun dalam penelitian tersebut hanya melakukan pengujian untuk akses *login*, *register*, dan pengubahan data, sehingga belum bisa terhubung dengan fitur-fitur yang lain seperti lokasi toko. [5]

Berdasarkan kajian terhadap penelitian-penelitian sebelumnya, masih terdapat beberapa keterbatasan yang ditemukan. Beberapa di antaranya meliputi ketiadaan fitur *export data* yang mempermudah pengelolaan informasi, penggunaan *database* lokal (*localhost*) yang mengharuskan perangkat tetap menyala untuk menjalankan sistem, keterbatasan fleksibilitas dalam kustomisasi tampilan UI pada platform *thinger.io*, serta pengembangan sistem yang belum optimal sehingga hanya berfungsi sebagai prototipe. Untuk mengatasi kekurangan-kekurangan tersebut, penulis berinisiatif untuk menambahkan fitur *export data*, memanfaatkan layanan *hosting* untuk pengelolaan *database*, menggunakan *framework* Flutter dalam pengembangan aplikasi, serta membangun sistem yang siap digunakan secara langsung dan tidak lagi sebatas prototipe.

2.2 Dasar Teori

2.2.1 Earplug

Kebisingan di lingkungan perusahaan dapat diartikan sebagai suara yang tidak diinginkan dan bersumber dari aktivitas internal, seperti penggunaan mesin kerja, alat-alat yang bergerak, peralatan yang bersentuhan dengan logam berat, maupun kompresor dan perlengkapan lainnya [6]. Lingkungan industri memiliki resiko kebisingan tinggi hingga melebihi batas aman (80 dB) yang dapat menimbulkan gangguan pendengaran permanen apabila terpapar secara terus menerus. Paparan kebisingan sebesar 85-92 dBA selama lebih dari 8 jam per hari di tempat kerja berpotensi menyebabkan gangguan pendengaran [1]. Oleh karena

itu setiap orang yang bekerja di industri wajib menggunakan alat pelindung diri untuk menghindarkan dari kebisingan tinggi. *Earplug* merupakan alat pelindung diri yang termasuk kedalam jenis *Hearing Protection Device* (HPD).

Earplug dirancang untuk melindungi telinga dari suara bising, air, dan benda asing lainnya. Alat ini dapat mengurangi tingkat kebisingan sehingga menghindarkan pengguna dari kerusakan pendengaran. UU Nomor 13 tahun 2003 tentang Ketenagakerjaan pada pasal 86 ayat 1 menyebutkan bahwa setiap pekerja/buruh berhak atas perlindungan Keselamatan dan Kesehatan Kerja (K3), dalam hal ini termasuk perlindungan terhadap resiko kebisingan. Oleh karena itu setiap perusahaan wajib menyediakan alat pelindung diri (APD) bagi pekerja.



Gambar 2.1 Earplug Yang Umum Digunakan Pada Industri

Earplug umumnya terbuat dari bahan busa poliuretan, silikon, dan lilin. Pemilihan bahan yang tepat bergantung pada kebutuhan spesifik pengguna, seperti tingkat kenyamanan, durasi penggunaan, dan tingkat perlindungan yang diinginkan. Gambar 2.1 menunjukkan *earplug* yang umum digunakan pada industri. Salah satu produsen alat pelindung pendengaran yang dikenal secara global adalah perusahaan 3M. Dalam penelitian ini, digunakan salah satu jenis

earplug dari 3M yaitu jenis 3M™ E-A-R™ Classic™. Pemilihan *earplug* jenis ini didasarkan pada tingkat penggunaan yang tinggi di dunia industri serta pertimbangan ketersediaan dan efisiensi biaya.

Dispenser *earplug* merupakan perangkat yang menyediakan *earplug* secara higienis dan efisien di area kerja. Penggunaan dispenser *earplug* membantu memastikan ketersediaan pelindung pendengaran yang bersih dan mudah diakses oleh pekerja serta meningkatkan kepatuhan terhadap protokol keselamatan kerja. Pada Gambar 2.2 menunjukkan dispenser *earplug* yang biasa digunakan di lingkungan industri. Gambar 2.2 juga menjadi referensi utama dalam membuat desain dan mekanisme dispenser *earplug*.



Gambar 2.2 Dispenser Earplug Di Lingkungan Industri

2.2.2 Otomatisasi

Otomatisasi merupakan proses pengendalian suatu sistem atau alat yang bekerja secara mandiri, dengan sedikit atau tanpa campur tangan manusia, untuk meningkatkan efisiensi, efektivitas, dan keakuratan dalam suatu proses kerja [7]. Dalam dunia industri dan keselamatan kerja, otomatisasi memiliki peran penting dalam mendukung penerapan kebijakan Keselamatan dan Kesehatan Kerja (K3).

Salah satu contoh penerapannya adalah dalam pengelolaan pemakaian Alat Pelindung Diri (APD) seperti *earplug*. Permasalahan umum di lapangan adalah penggunaan APD yang tidak terpantau atau tidak terkontrol, sehingga dapat mengurangi efektivitas perlindungan bagi pekerja. Dengan otomatisasi, proses distribusi APD dapat dilakukan secara lebih tertib, terukur, dan terdokumentasi.

Penelitian ini merancang sistem dispenser *earplug* otomatis berbasis RFID dan ESP32, yang memungkinkan proses pengambilan *earplug* dilakukan secara otomatis dan dibatasi maksimal dua kali per hari per karyawan. Sistem ini bekerja dengan mendeteksi identitas karyawan menggunakan kartu RFID, memverifikasi hak akses, lalu mengaktifkan mekanisme dispenser jika syarat terpenuhi. Informasi pengambilan *earplug* juga dicatat dan ditampilkan melalui sistem monitoring berbasis web dan aplikasi mobile.

Mikrokontroler ESP32 dipilih sebagai otak pengendali karena mendukung konektivitas Wi-Fi dan Bluetooth, serta memiliki performa yang baik untuk aplikasi IoT (*Internet of Things*) dengan konsumsi daya rendah. Dengan memanfaatkan ESP32, sistem otomatisasi ini mampu terhubung ke *server* atau *database online* sehingga data penggunaan APD dapat dipantau secara *real-time*.

Otomatisasi dalam sistem ini tidak hanya memberikan kemudahan operasional, tetapi juga integrasi antara sistem fisik dan digital untuk menciptakan proses kerja yang cerdas dan terhubung. Melalui sistem otomatis ini, perusahaan dapat menerapkan pengawasan penggunaan APD secara digital, mencegah penyalahgunaan, serta meningkatkan disiplin dan kesadaran pekerja terhadap keselamatan kerja.

Dengan demikian, otomatisasi tidak hanya berkaitan dengan mekanisme kerja otomatis, tetapi juga menyangkut integrasi data dan sistem pengambilan keputusan yang lebih cepat dan akurat berbasis teknologi.

2.2.3 Mikrokontroler

Mikrokontroler merupakan sebuah komputer kecil (*special purpose computers*) di dalam satu IC yang berisi CPU, memori, timer, saluran komunikasi serial dan paralel, *port input/output*, dan ADC [8]. Perangkat ini terdiri dari

beberapa komponen utama yang membentuk suatu sistem. Komponen tersebut terdiri dari CPU, RAM, ROM, *Serial Port*, *Timer*, *Counter*, *Port Input/Output*, *Special Functioning Block*, *Interrupt Control*, *Analog to Digital Converter*, *Digital to Analog Converter*. Mikrokontroler bekerja dengan menerima, memproses, dan menghasilkan data atau sinyal yang digunakan untuk mengendalikan perangkat lain.

ESP32 merupakan salah satu jenis mikrokontroler pengembangan dari ESP8266 yang memiliki fitur lebih lengkap dan kinerja yang lebih tinggi. ESP32 termasuk mikrokontroler berbasis *system on chip* (SoC) yang dikembangkan oleh *Espressif Systems*. Mikrokontroler ini dirancang untuk mendukung aplikasi *Internet of Things* (IoT) dengan menyediakan konektivitas WiFi dan Bluetooth dalam satu modul. ESP32 bekerja sebagai pengendali utama yang dapat membaca dan memproses data dari perangkat eksternal seperti sensor dan aktuator yang kemudian mengirimkan data tersebut ke jaringan menggunakan WiFi atau Bluetooth. ESP 32 memiliki beberapa mode seperti pada Tabel 2.3, salah satunya adalah operasi hemat daya yang dirancang untuk memperpanjang masa pakai baterai. Tabel 2.4 menunjukkan spesifikasi dan Gambar 2.5 menunjukkan deskripsi pin dan pinout pada ESP32.

Tabel 2-1 Mode Operasi ESP32

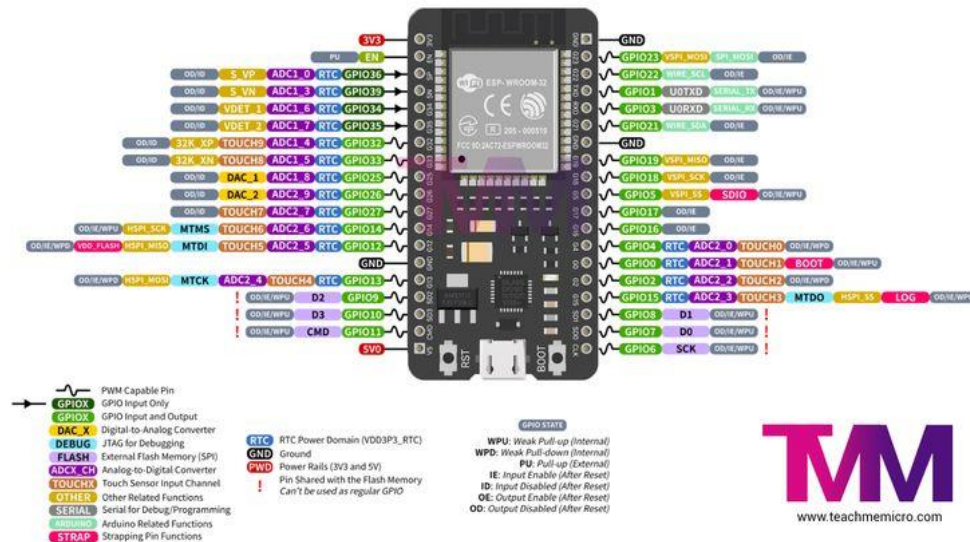
Mode	Deskripsi
Active Mode	Prosesor berjalan pada kecepatan penuh dan semua fitur aktif
Modem Sleep	WiFi atau Bluetooth dinonaktifkan sementara prosesor tetap aktif untuk tugas-tugas tertentu
Light Sleep	Prosesor dimatikan sementara RAM tetap aktif
Deep Sleep	Hampir semua fungsi dimatikan kecuali RTC (<i>Real Time Clock</i>)

Tabel 2-2 Spesifikasi ESP32 WROOM 32D [9]

Kategori	Spesifikasi
Prosesor	Dual-core Xtensa® 32-bit LX6, hingga 240 MHz per core
Memori	520 KB SRAM, hingga 16 MB Flash
Konektivitas	Wi-Fi: IEEE 802.11 b/g/n (2.4 GHz)
	Bluetooth 4.2 (Classic & BLE)
GPIO	34 Pin
ADC	12-bit ADC, hingga 18 saluran
DAC	8-bit DAC (2 saluran)
PWM	Hingga 16 saluran PWM
I2C, SPI, UART	Yes
Tegangan input	5V
Tegangan Kerja	3.3V (operasi stabil antara 2.2V hingga 3.6V)
Konsumsi Daya	Active Mode: ~160 mA, Modem Sleep: ~30 mA, Light Sleep: ~0.8 mA, Deep Sleep: <10 μ A
Ukuran Modul	18 x 25.5 mm
Jumlah Pin	38 pin
WiFi Range	Hingga 200 meter (tergantung lingkungan)
Bluetooth Range	Hingga 100 meter untuk BLE (tergantung daya transmisi)
Lingkungan Kerja	Suhu: -40°C hingga +125°C, Kelembapan: 5% hingga 95% (non-kondensasi)
Sistem Operasi	Mendukung FreeRTOS, Arduino IDE, Espressif IDF, PlatformIO
Platform Pengembangan	Arduino IDE, Espressif IoT Development Framework (IDF), PlatformIO

ESP32 PINOUT DIAGRAM

ESP32-WROOM-32



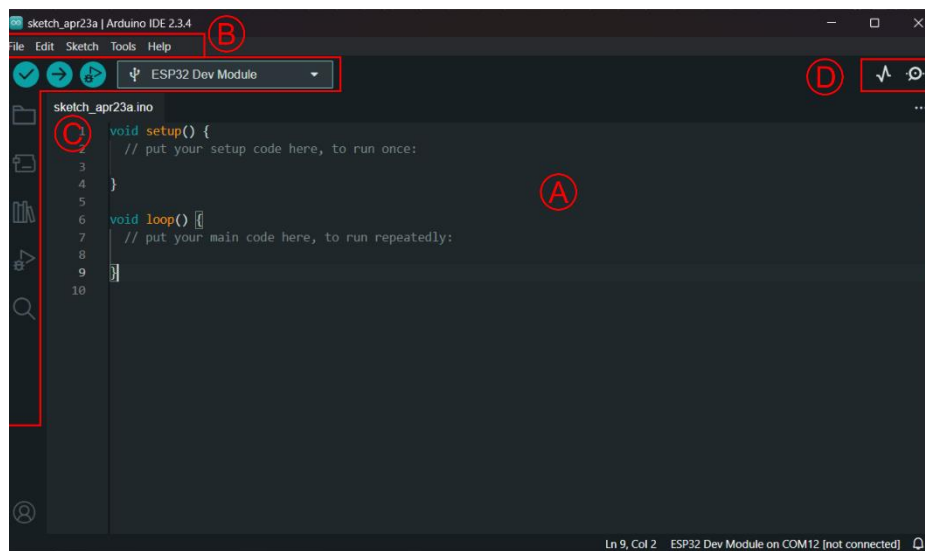
Gambar 2.3 Pin Out ESP32 WROOM 32D

Untuk dapat melakukan pemrosesan data sesuai dengan fungsi yang diinginkan, mikrokontroler ESP32 memerlukan deklarasi logika dalam bentuk instruksi atau program. Instruksi ini disusun menggunakan bahasa pemrograman tertentu yang kemudian dikompilasi dan diunggah ke dalam mikrokontroler. Salah satu perangkat lunak yang umum digunakan untuk menulis dan mengunggah program ke ESP32 adalah Arduino IDE.

2.2.3.1 Arduino IDE

Arduino IDE (*Integrated Development Environment*) merupakan perangkat lunak yang dirancang untuk memudahkan proses pemrograman pada mikrokontroler berbasis platform Arduino. Perangkat lunak ini memiliki antarmuka yang sederhana dan intuitif, sehingga sangat mendukung pembelajaran dan pengembangan sistem tertanam, baik bagi pemula maupun kalangan profesional. Melalui Arduino IDE, pengguna dapat menulis, mengedit, dan mengunggah program atau *sketch* ke papan mikrokontroler dengan cepat dan efisien. Bahasa pemrograman yang digunakan dalam Arduino IDE berbasis pada C/C++ yang telah

disederhanakan, sehingga memudahkan dalam proses penulisan kode. Selain mendukung berbagai jenis papan pengendali seperti Arduino Uno, Arduino Mega, dan Arduino Nano, Arduino IDE juga kompatibel dengan mikrokontroler lain seperti ESP32. Dukungan ini memungkinkan pengguna untuk memprogram ESP32 secara langsung melalui Arduino IDE, dengan terlebih dahulu menambahkan pustaka dan pengaturan papan (*board manager*) yang sesuai. Dengan demikian, Arduino IDE tidak hanya terbatas pada papan Arduino standar, tetapi juga dapat digunakan sebagai alat utama dalam pengembangan aplikasi berbasis ESP32.



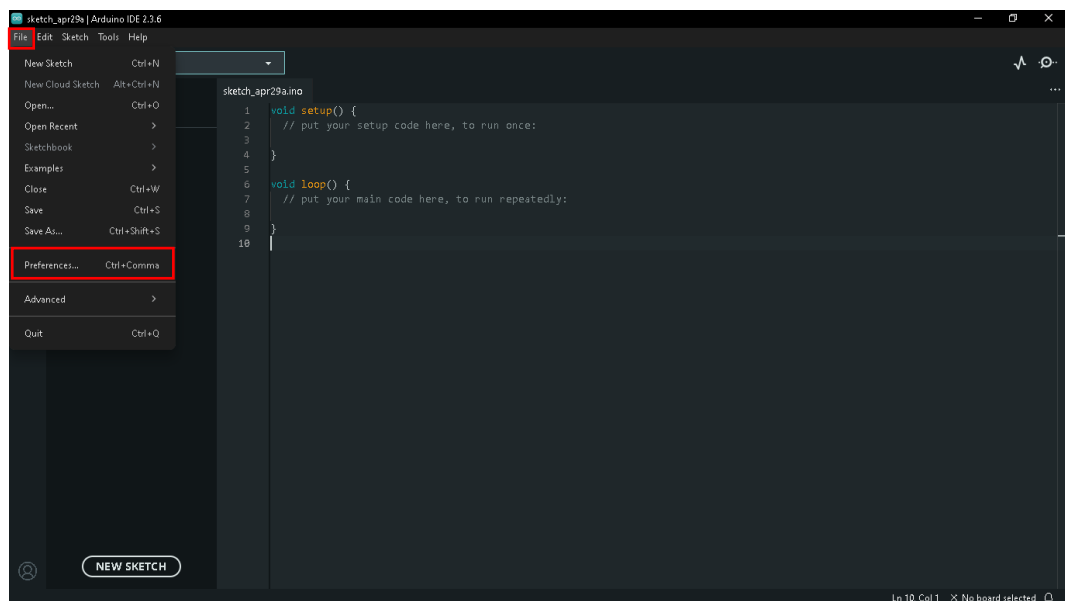
Gambar 2.4 Bagian-Bagian Arduino IDE

Gambar 2.4 menunjukkan tampilan awal ketika aplikasi Arduino IDE dibuka. Secara umum Arduino IDE memiliki 4 bagian utama. Kode A merupakan *workspace* yang menjadi tempat untuk menuliskan kode yang akan digunakan untuk memprogram mikrokontroler. Kode B merupakan *menu bar* yang terdiri dari *file*, *edit*, *sketch*, *tools*, dan *help*. Menu *file* berisi fungsi dasar untuk mengelola file *sketch* (program). Menu *edit* berfungsi untuk mengedit kode seperti menyalin, memformat, atau memberi komentar. Menu *sketch* digunakan untuk mengompilasi, mengunggah program, dan menambahkan pustaka ke dalam *sketch*. Menu *tools* berisi pengaturan teknis seperti pemilihan *board* dan *port* serta alat pemantauan serial. Menu *help* menyediakan bantuan, dokumentasi, dan informasi tentang

Arduino IDE. Kode C merupakan *tool bar* yang digunakan untuk melakukan komunikasi dengan mikrokontroler. Pada *tool bar* terdapat *tool verify* yang berfungsi untuk melakukan cek pada *sketch* sebelum proses *upload* pada mikrokontroler, *tool upload* yang berfungsi untuk mengirimkan *sketch* ke *board*, *tool start debugging* yang berfungsi untuk memulai *debugging*, dan informasi *board* dan *port* yang memberikan informasi jenis *board* dan *port*. Pada kode D terdapat dua tombol yaitu *serial plotter* dan *serial monitor*. *Serial plotter* berfungsi untuk memvisualisasikan data numerik yang dikirim oleh *board* dalam bentuk grafik. Sedangkan *serial monitor* berfungsi untuk membuka *interface* komunikasi serial yang memungkinkan pengguna mengetahui hasil pembacaan atau proses yang dilakukan oleh *board*. Untuk menggunakan Arduino IDE dapat melakukan langkah-langkah sebagai berikut.

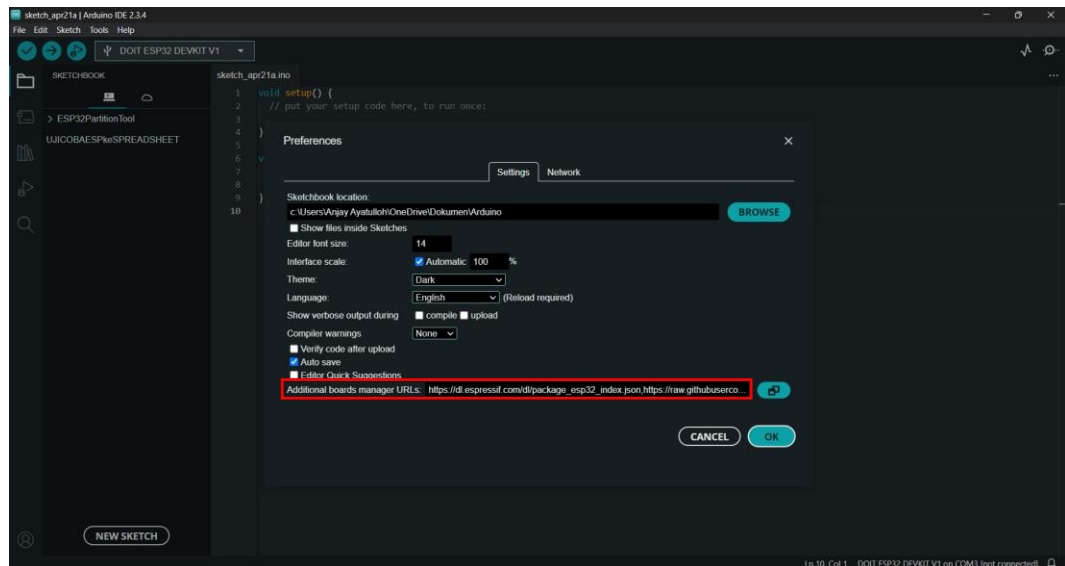
1. Membuka aplikasi Arduino IDE hingga muncul seperti tampilan pada Gambar 2.4.
2. Menambahkan *board* ke Arduino IDE

Untuk menambahkan jenis papan mikrokontroler yang tidak tersedia secara bawaan di Arduino IDE, pengguna dapat membuka menu *File > Preferences* seperti pada Gambar 2.5.



Gambar 2.5 Tampilan Menu File

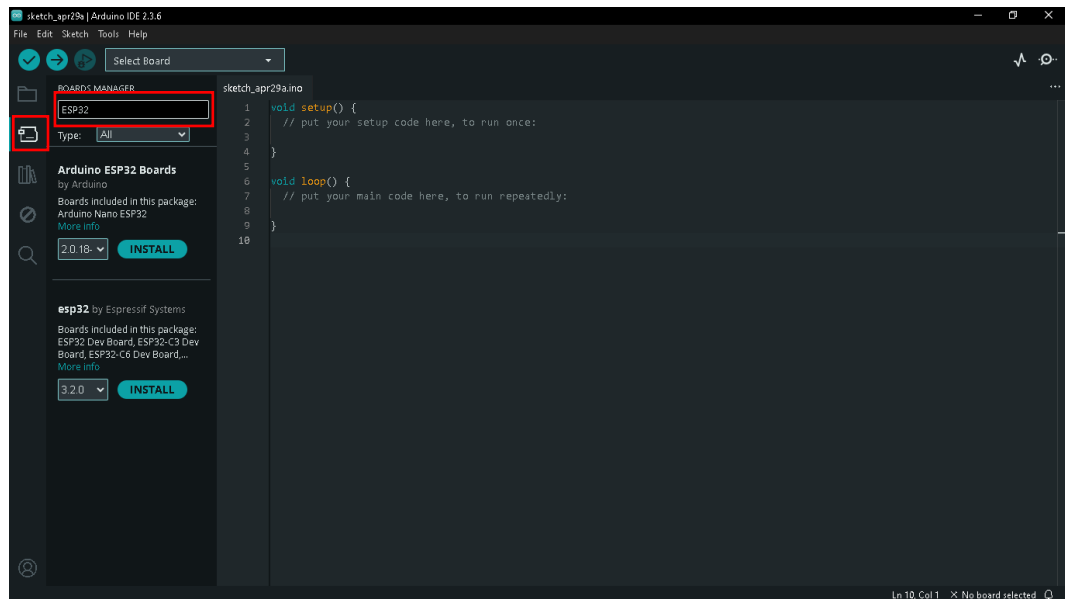
Pada jendela yang muncul, pengguna dapat memasukkan tautan URL repositori board (seperti pada Gambar 2.6) pada kolom yang telah disediakan. Langkah ini bertujuan untuk memungkinkan Arduino IDE mengenali dan mendukung board tambahan seperti ESP32.



Gambar 2.6 URL Box untuk menambahkan board

3. Menginstall *board* melalui *board manager*

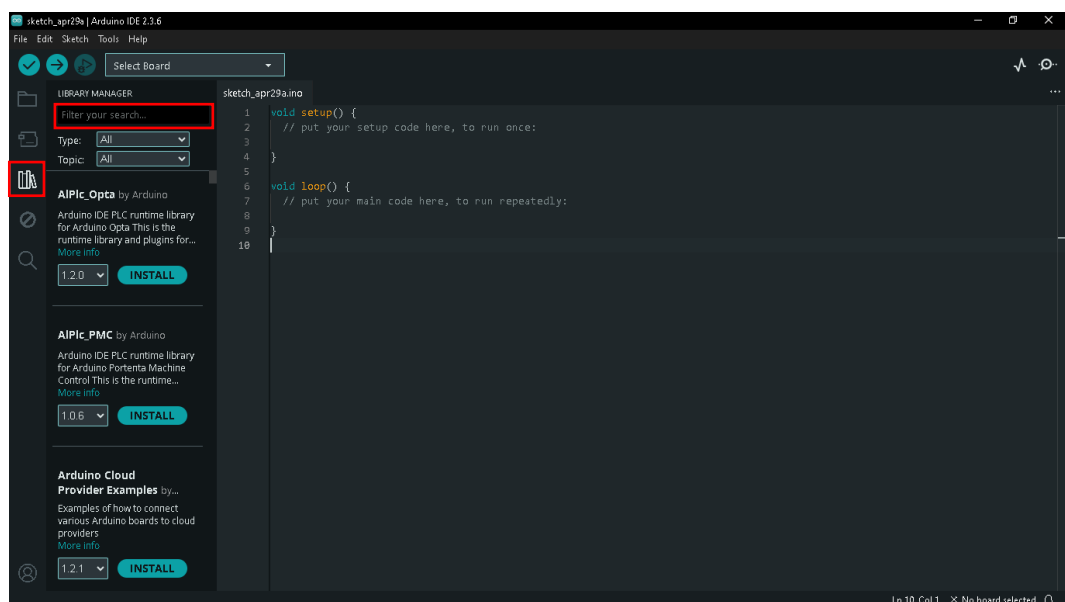
Setelah menambahkan URL repositori, pengguna perlu mengunduh dan menginstal jenis papan yang diinginkan melalui fitur *Board Manager* seperti pada Gambar 2.7. Akses dapat dilakukan melalui menu *Tools > Board > Board Manager*. Pada jendela yang tersedia, pengguna dapat mencari nama papan, misalnya “ESP32 by Espressif Systems,” kemudian menekan tombol *Install* untuk mengunduh dan menambahkan board tersebut ke dalam Arduino IDE.



Gambar 2.7 Tampilan Board Manager

4. Menginstall *Library*

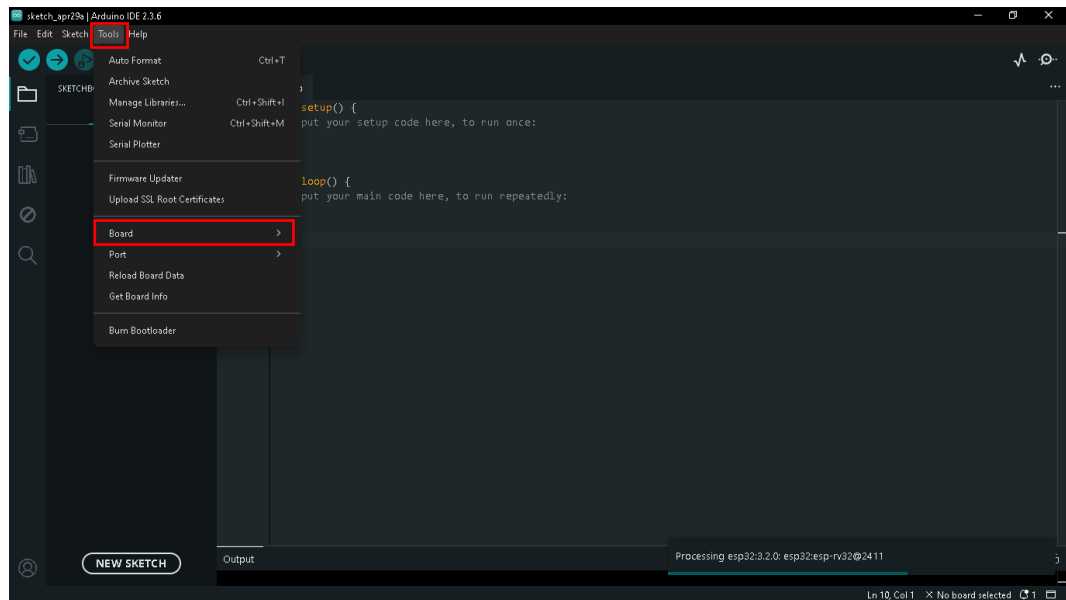
Untuk mengaktifkan berbagai fitur tambahan pada ESP32, pengguna juga perlu menambahkan pustaka pendukung. Proses ini dapat dilakukan melalui menu *Tools > Manage Libraries* atau *Sketch > Include Library > Manage Libraries* seperti pada Gambar 2.8. Melalui jendela *Library Manager*, pengguna dapat mencari dan menginstal pustaka yang sesuai dengan kebutuhan pemrograman.



Gambar 2.8 Tampilan Manage Library

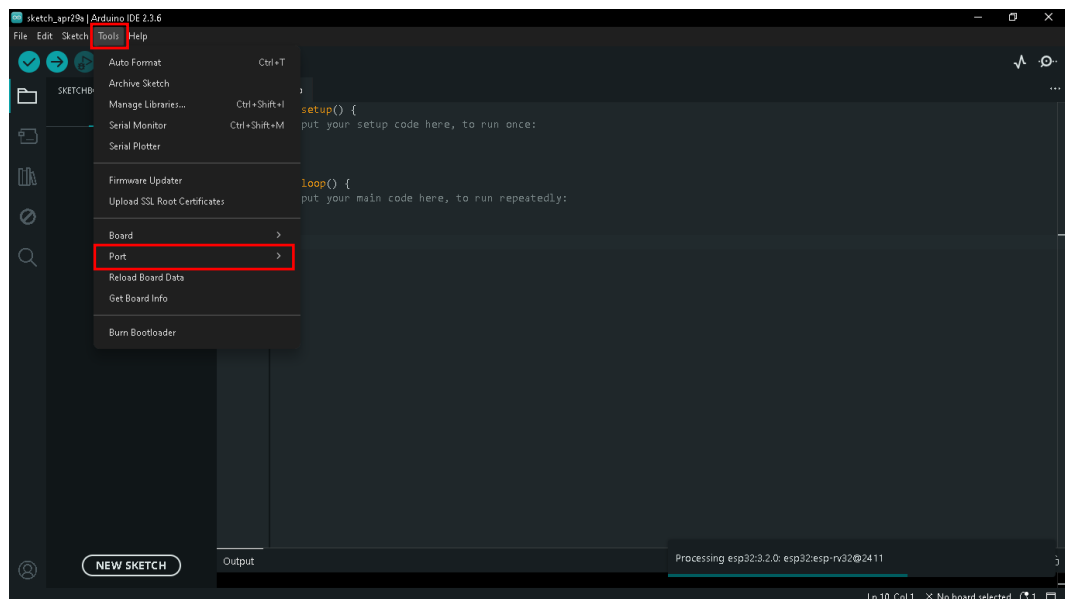
5. Memilih jenis *board* dan *port* yang digunakan

Sebelum memulai proses pemrograman, pengguna harus menentukan jenis board dan port komunikasi serial yang akan digunakan. Pemilihan jenis board dilakukan melalui menu *Tools > Board* seperti pada Gambar 2.9.



Gambar 2.9 Tampilan Menu Tools untuk memilih Board

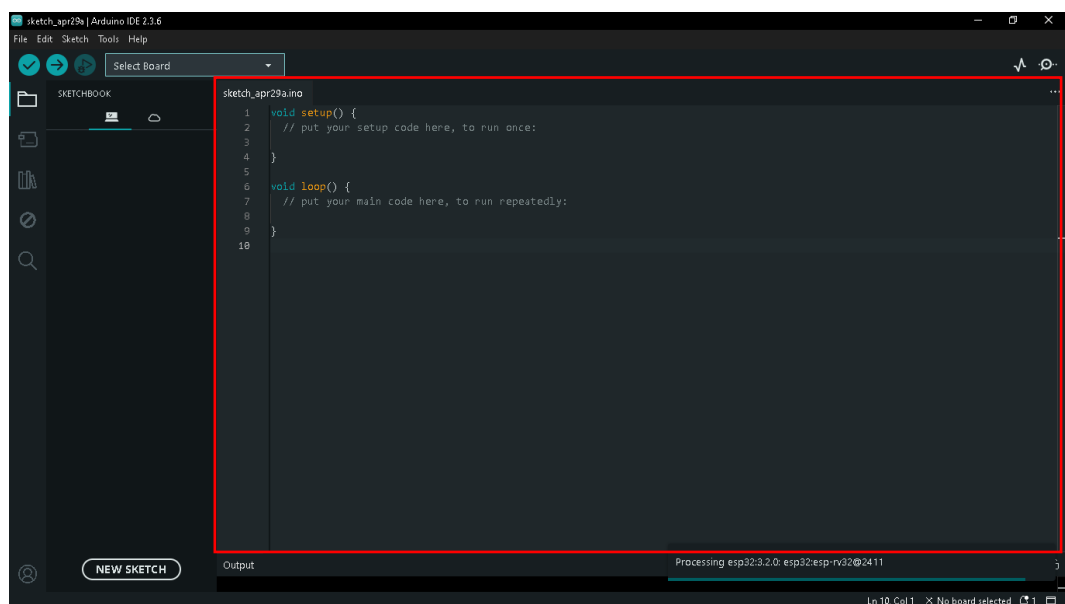
Sedangkan port komunikasi dapat dipilih melalui *Tools > Port* seperti pada Gambar 2.10. Pengaturan ini penting agar Arduino IDE dapat terhubung secara tepat dengan perangkat keras yang digunakan.



Gambar 2.10 Tampilan Menu Tools untuk memilih Port

6. Menulis program

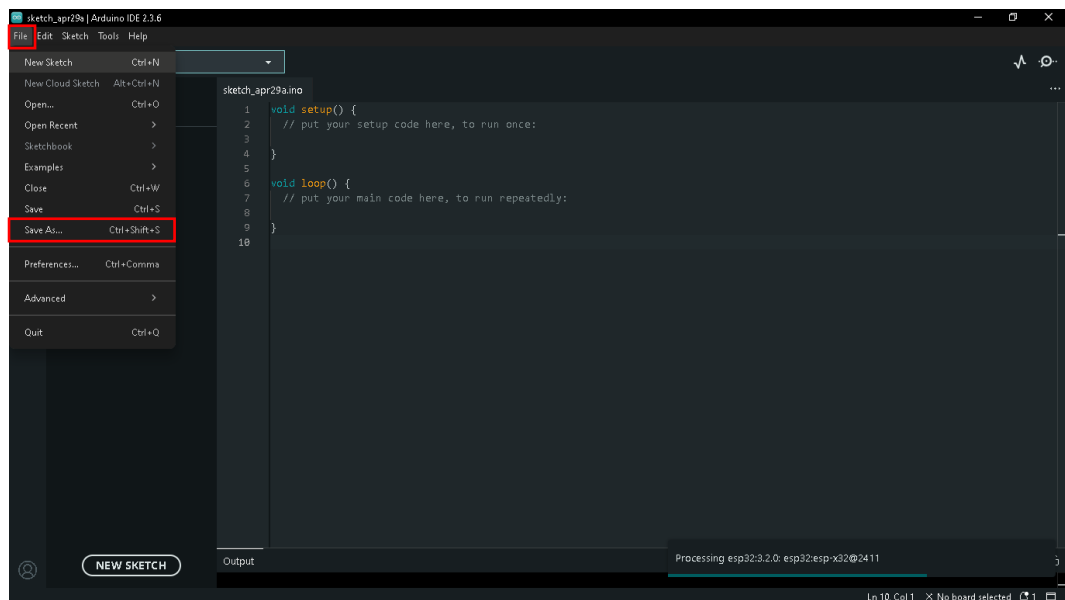
Setelah seluruh konfigurasi selesai, pengguna dapat mulai menulis program atau *sketch* pada area editor yang tersedia di Arduino IDE. Program ini berisi instruksi logika yang akan dijalankan oleh mikrokontroler. Gambar 2.11 menunjukkan area *sketch* pada arduino IDE.



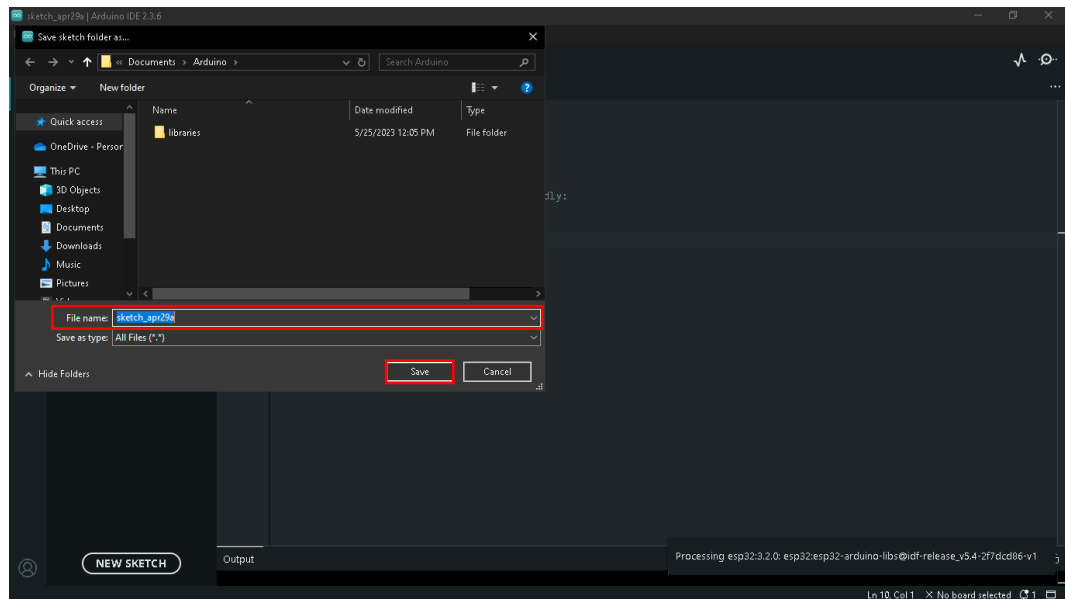
Gambar 2.11 Area Sketch Pada Arduino IDE

7. Menyimpan program

Setelah program selesai ditulis, langkah selanjutnya adalah menyimpan file program ke dalam folder pada perangkat komputer. Proses penyimpanan dilakukan melalui menu *File > Save As*, di mana pengguna dapat menentukan lokasi dan nama file sesuai keinginan. Menu *Save As* dapat dilihat pada Gambar 2.12. Program yang telah tersimpan dapat digunakan untuk proses verifikasi (*compiling*) dan selanjutnya diunggah ke papan mikrokontroler agar dapat dijalankan. Untuk menentukan tempat penyimpanan file dapat dilihat pada Gambar 2.13.



Gambar 2.12 Tampilan Menu File untuk Save Project



Gambar 2.13 Tampilan Menu untuk menyimpan project di komputer

2.2.4 RFID

RFID atau *Radio Frequency Identification* adalah teknologi komunikasi yang memanfaatkan gelombang radio untuk memungkinkan pertukaran data antara RFID *Reader* dan RFID *Tag* ketika keduanya berada dalam jarak tertentu. Sistem ini terdiri dari dua komponen utama, yaitu RFID *Reader* dan RFID *Tag*. Teknologi ini mampu mengidentifikasi berbagai objek secara akurat tanpa memerlukan kontak langsung, dengan memanfaatkan transmisi frekuensi radio tertentu seperti 125 kHz, 13,65 MHz, atau 800–900 MHz. RFID menjadi salah satu bentuk teknologi pengumpulan data otomatis yang berkembang dengan sangat cepat dan efisien dalam berbagai bidang [10]. Saat ini banyak industri yang sudah memanfaatkan RFID untuk pemantauan absen karyawan atau *access control*. Biasanya kartu karyawan yang diberikan oleh perusahaan sudah berupa *RFID Card* dimana karyawan cukup melakukan *scan* pada modul yang tersedia untuk absen dan keperluan lain misalnya pengambilan barang. RFID secara umum dapat dibagi menjadi tiga jenis yaitu catu daya, memori, dan frekuensi radio.

RFID dengan catu daya memiliki dua tipe yaitu *tag* aktif dan *tag* pasif. RFID dengan tipe catu daya tag aktif memiliki informasi yang dapat dibaca dan ditulis oleh *scanner* RFID pada jarak jauh bahkan hingga ratusan meter (tergantung

spesifikasi). Sementara RFID tipe tag pasif hanya mengandalkan teknologi elektromagnetik untuk sumber *power* sehingga jarak bacanya lebih pendek. RFID jenis memori memiliki dua tipe yaitu *read and write* serta *read only*. Tag tipe *read and write* memiliki arti bahwa informasi yang disediakan dapat dibaca dan ditulis secara berulang atau dinamis. Sedangkan tag tipe *read only* hanya menyimpan informasi saat tag dibuat dan informasinya tidak dapat diubah (statis). Kemudian untuk RFID frekuensi radio dibagi menjadi tiga jenis, yaitu *low frequency* (LF), *high frequency* (HF), dan *ultrahigh frequency* (UHF). Secara umum jenis RFID ini memiliki makna semakin kecil frekuensi maka semakin pendek jarak baca dan semakin lemah juga tingkat keterbacaannya. Gambar 2.14 menunjukkan cara kerja RFID dari Tag, Reader, hingga komputer.



Gambar 2.14 Cara Kerja RFID

Modul RFID RC522 merupakan salah satu modul RFID yang termasuk jenis *read and write* dan bekerja pada frekuensi 13.56 MHz. Modul ini terdiri dari beberapa bagian penting yang memungkinkan untuk berkomunikasi dengan tag RFID seperti Chip MFRC522 yang berperan sebagai inti komunikasi RFID. Modul RFID RC522 bekerja dengan memancarkan gelombang elektromagnetik dengan frekuensi 13.56 MHz melalui antena. Gelombang ini digunakan untuk mengaktifkan RFID pasif di sekitarnya. Ketika tag RFID berada dalam jangkauan gelombang elektromagnetik, antena pada tag akan menangkap gelombang tersebut dan mengaktifkan sirkuit internalnya. Setelah aktif, tag akan mengirimkan data yang tersimpan dalam memorinya melalui modulasi sinyal balik. Modul RC522 kemudian mendemodulasi sinyal tersebut dan meneruskan data yang diterima ke

mikrokontroler melalui salah satu protokol komunikasi seperti SPI, I2C, atau UART untuk diproses lebih lanjut.

Agar proses pembacaan dapat berlangsung, tag RFID harus bekerja pada frekuensi yang sama dengan reader, yaitu 13,56 MHz. Jika frekuensinya tidak sama, maka reader tidak dapat membaca data dari tag. Selain frekuensi 13,56 MHz yang umum digunakan pada sistem RFID berbasis *proximity card* seperti yang digunakan dalam tugas akhir ini, ada juga frekuensi RFID lain yang sering ditemukan, misalnya RFID berfrekuensi 125 kHz (sering digunakan untuk kontrol akses sederhana) dan RFID berfrekuensi 860–960 MHz (UHF RFID, digunakan untuk pelacakan logistik dan inventarisasi, seperti pada e-KTP di Indonesia).

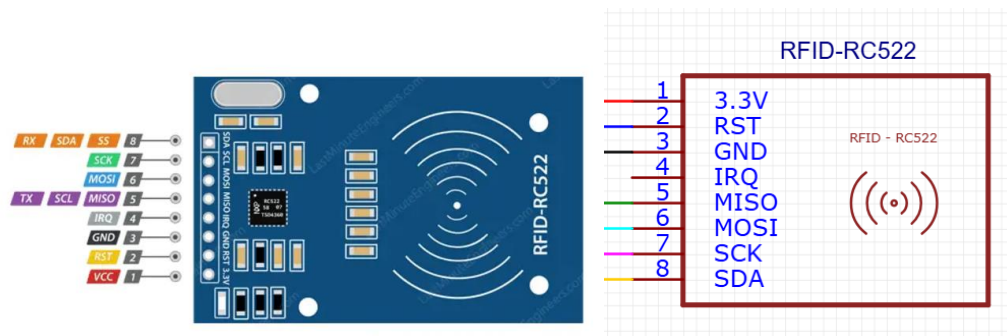
Dalam proses pengambilan data serta rancang bangun sistem, akan digunakan modul RFID RC522 sebagai perangkat pembaca (RFID reader). Modul ini memiliki beberapa pin untuk mendukung proses komunikasi dan integrasi dengan mikrokontroler, yang dideskripsikan pada Tabel 2.3 dan Gambar 2.15. Spesifikasi teknis dari modul RFID RC522 ditunjukkan pada Tabel 2.4.

Tabel 2-3 Deskripsi Pin Modul RFID RC-522 [11]

Pin	Deskripsi
VCC	Tegangan daya (3.3V)
GND	Ground
RST	Pin Reset, digunakan untuk mereset modul
IRQ	Pin Interupsi, opsional untuk pengendalian tambahan
MISO	Data Keluar (<i>Master Input Slave Output</i>) pada SPI
MOSI	Data Masuk (<i>Master Output Slave Input</i>) pada SPI
SCK	Clock SPI, digunakan untuk sinkronisasi data
SDA/SS	Slave Select, mengaktifkan modul saat berkomunikasi

Tabel 2-4 Spesifikasi Modul RFID RC-522

Kategori	Spesifikasi
Frekuensi Operasi	13.56 MHz
Protokol yang Didukung	ISO/IEC 14443A (MIFARE)
Antarmuka Komunikasi	SPI, UART, I2C
Kecepatan SPI	Up to 10Mbps
Tegangan Operasi	2.5V – 3.3V
Arus Operasi	~13 – 26mA (saat membaca tag)
Dimensi modul	4 cm × 6 cm



Gambar 2.15 Pin Output Modul RFID RC-522

Sebagai media identifikasi, digunakan empat buah kartu RFID (RFID tag) yang masing-masing memiliki kode unik, yaitu 9928392, 4299D5, D4C4C5, dan 4B214C5. Keempat tag ini akan digunakan untuk mensimulasikan proses autentikasi atau identifikasi objek pada sistem yang dikembangkan, sehingga memungkinkan pengujian fungsionalitas sistem secara menyeluruh dalam berbagai skenario. Pada penelitian dan rancang bangun ini, tag RFID yang digunakan beroperasi pada frekuensi 13,56 MHz agar kompatibel dengan modul RC522, sehingga memastikan sistem dapat berjalan dengan optimal.

Untuk melakukan pembacaan kartu RFID, digunakan beberapa fungsi penting dari library MFRC522. Pertama, fungsi `!mfrc522.PICC_IsNewCardPresent()` digunakan untuk mendeteksi keberadaan kartu baru di area pembacaan. Fungsi ini mengembalikan nilai `true` jika ada kartu

RFID baru yang terdeteksi, dan *false* jika tidak ada. Operator negasi (!) digunakan untuk menghentikan proses lebih lanjut apabila tidak ada kartu baru yang berada dalam jangkauan, sehingga dapat menghemat sumber daya mikrokontroler dan mencegah pembacaan berulang pada kartu yang sama.

Setelah keberadaan kartu terdeteksi, fungsi `!mfrc522.PICC_ReadCardSerial()` dipanggil untuk membaca data serial unik (UID) dari kartu tersebut. Fungsi ini bertanggung jawab untuk mengakses informasi identifikasi dari kartu RFID. Jika pembacaan gagal, operator negasi (!) menyebabkan program keluar dari proses pembacaan untuk menjaga akurasi data. Data UID yang berhasil dibaca kemudian ditampilkan melalui monitor serial menggunakan perintah berikut.

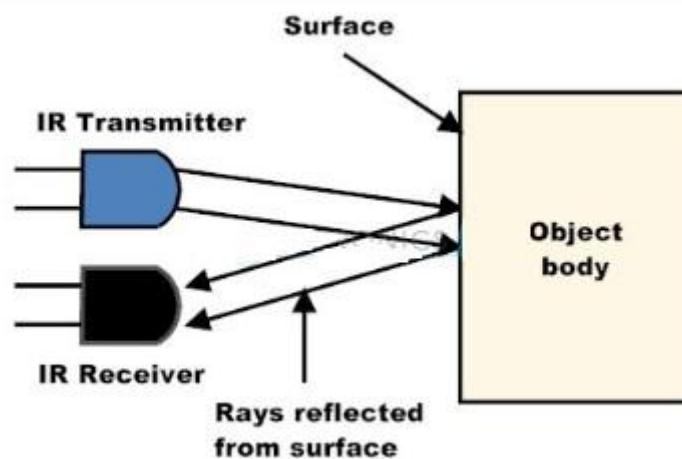
```
Serial.print("UID tag : ");
for (byte i = 0; i < mfrc522.uid.size; i++) {
  Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
  Serial.print(mfrc522.uid.uidByte[i], HEX);
}
Serial.println();
```

Potongan kode tersebut diawali dengan menampilkan teks "UID tag :" untuk memberikan informasi awal kepada pengguna. Selanjutnya, dilakukan perulangan sebanyak jumlah byte UID dengan menggunakan perintah *for*. Di dalam perulangan, setiap *byte* UID diperiksa: apabila nilainya kurang dari 16 (0x10 dalam bilangan heksadesimal), maka akan ditambahkan angka nol di depan untuk menjaga konsistensi format dua digit heksadesimal. Setiap *byte* kemudian dicetak ke *monitor serial* dalam format heksadesimal menggunakan `Serial.print` dengan parameter HEX. Setelah seluruh UID dicetak, `Serial.println()` digunakan untuk mengakhiri proses dengan pindah ke baris baru, agar hasil pembacaan berikutnya tidak tercampur. Dengan tahapan tersebut, proses identifikasi *tag* RFID menjadi lebih terstruktur, efisien, dan memudahkan validasi data pada sistem yang dirancang.

2.2.5 Sensor Infrared

Sensor *infrared* (IR) merupakan perangkat elektronik yang memanfaatkan gelombang elektromagnetik pada spektrum *infrared*, khususnya pada panjang gelombang sekitar 700 nm hingga 1 mm, yang berada di luar jangkauan penglihatan manusia. Sensor ini umum digunakan dalam berbagai aplikasi, seperti sistem navigasi robot, penghindar halangan (*obstacle avoidance*), pendeteksi garis (*line follower*), serta aplikasi keamanan dan otomasi.

Secara umum, sensor IR terdiri atas dua komponen utama, yaitu pemancar (*emitter*) dan penerima (*receiver*). Pemancar IR biasanya berupa dioda pemancar cahaya (*infrared LED*) yang memancarkan cahaya tak kasatmata, sementara penerima IR dapat berupa fotodioda atau fototransistor yang mendeteksi pantulan sinyal IR dari suatu objek. Ketika cahaya *infrared* yang dipancarkan mengenai suatu objek, sebagian cahaya akan dipantulkan kembali dan diterima oleh fototransistor. Intensitas cahaya yang diterima akan diubah menjadi sinyal listrik yang dapat diproses lebih lanjut untuk menentukan keberadaan atau jarak objek [12].



Gambar 2.16 Cara Kerja Modul TCRT5000

Modul sensor TCRT5000 bekerja berdasarkan prinsip pantulan cahaya inframerah yang dipancarkan oleh *infrared light-emitting diode* (IR LED) dan diterima kembali oleh *phototransistor* dalam satu paket sensor seperti ilustrasi pada

Gambar 2.16. Ketika sensor diberikan tegangan melalui pin VCC dan GND, IR LED akan memancarkan cahaya tak kasatmata ke arah objek. Jika terdapat objek di depan sensor, terutama dengan permukaan cerah atau reflektif, maka cahaya inframerah akan dipantulkan dan diterima oleh *phototransistor*, yang selanjutnya memengaruhi sinyal keluaran. Modul ini memiliki dua jenis keluaran, yaitu *analog output* (A0) dan *digital output* (D0). Sinyal dari pin A0 berupa tegangan analog yang nilainya berbanding terbalik dengan intensitas cahaya yang diterima, sehingga dapat dibaca oleh mikrokontroler melalui fitur *analog-to-digital converter* (ADC) untuk mendapatkan informasi tentang keberadaan dan jarak objek secara lebih detail. Sementara itu, sinyal dari pin D0 berupa logika digital HIGH atau LOW yang ditentukan oleh perbandingan antara sinyal pantulan dengan ambang batas yang disetel melalui potensiometer pada modul. Jika intensitas pantulan cahaya melebihi nilai ambang, maka output D0 akan bernilai LOW menandakan adanya objek, dan sebaliknya akan bernilai HIGH jika tidak ada pantulan yang cukup. Dengan konfigurasi ini, modul TCRT5000 dapat digunakan dalam berbagai aplikasi, baik yang membutuhkan pembacaan analog secara presisi maupun deteksi biner secara cepat dan sederhana.

Terdapat beberapa jenis sensor IR yang dikategorikan berdasarkan fungsinya, antara lain sensor IR pasif (PIR) dan sensor IR aktif. Sensor IR pasif (Passive Infrared Sensor) mendeteksi radiasi *infrared* alami yang dipancarkan oleh objek, terutama oleh tubuh manusia, sehingga sering digunakan dalam sistem keamanan dan deteksi gerakan. Sebaliknya, sensor IR aktif terdiri atas pasangan pemancar dan penerima yang bekerja bersama dalam satu sistem. Jenis ini lebih umum digunakan dalam aplikasi kontrol otomatis dan penghindaran objek karena kemampuannya dalam mendeteksi pantulan cahaya dari permukaan objek.

Dalam penelitian ini, digunakan sensor *infrared* jenis TCRT5000. Sensor ini termasuk sensor IR aktif yang dirancang untuk mendeteksi objek pada jarak dekat. TCRT5000 terdiri dari satu LED *infrared* sebagai pemancar dan satu fototransistor sebagai penerima dalam satu modul kecil. Cara kerjanya adalah dengan memancarkan cahaya ke arah objek. Jika ada benda di depan sensor, cahaya akan dipantulkan dan diterima kembali oleh fototransistor, menghasilkan sinyal output.

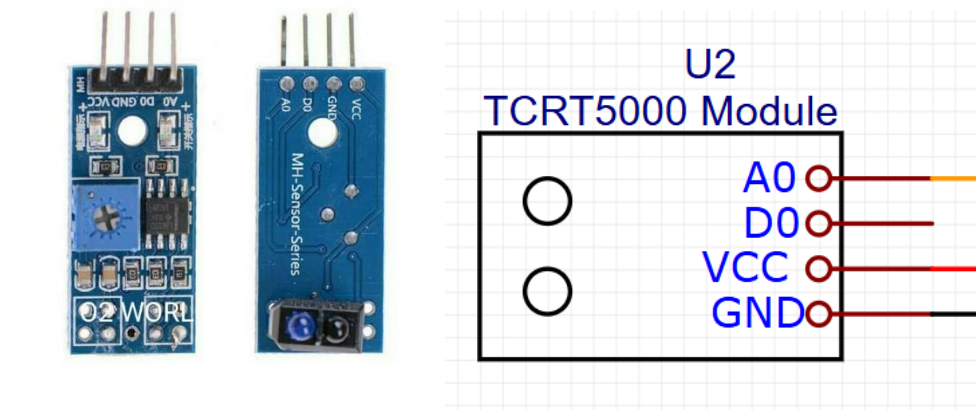
Kemampuan ini membuat TCRT5000 efektif untuk mendeteksi ada atau tidaknya benda. Pada sistem dispenser *earplug*, sensor ini digunakan untuk memastikan bahwa *earplug* benar-benar keluar, dengan mendeteksi setiap *earplug* yang jatuh ke tangan pengguna [13]. Pada tabel 2-5 dituliskan deskripsi pin pada modul TCRT5000 dan pada tabel 2-6 dituliskan spesifikasi dari modul TCRT5000 yang digunakan oleh sistem. Adapun pada Gambar 2.17 ditampilkan bentuk fisik sensor yang digunakan beserta simbolnya pada *wiring diagram*.

Tabel 2- 5 Deskripsi Pin Modul TCRT5000

Pin	Deskripsi
VCC	Tegangan daya
GND	Ground
A0	Analog Output, memberikan nilai tegangan proporsional terhadap intensitas pantulan cahaya
D0	Digital Output, memberikan sinyal HIGH atau LOW berdasarkan ambang (threshold) tertentu

Tabel 2-6 Spesifikasi Modul TCRT5000 [14]

Parameter	Spesifikasi
Tegangan Operasi	3.3 V – 5V
Jarak Deteksi Optimal	0,2 mm – 15 mm
Panjang Gelombang IR	± 950 nm
Tipe Pemancar	<i>Infrared LED</i>
Tipe Penerima	<i>Phototransistor</i>
Arus LED Maksimum	60 mA
Waktu Respon	< 1 ms
Dimensi	10 mm $\times$ 15 mm



Gambar 2.17 Bentuk Fisik dan Simbol Modul TCRT5000

2.2.6 Servo

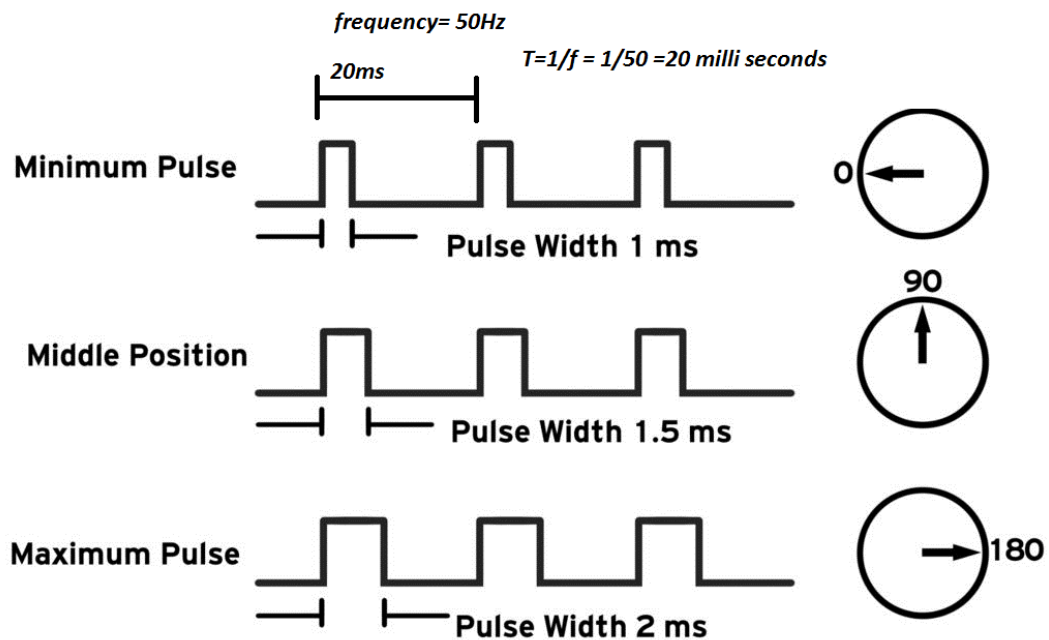
Servo motor merupakan salah satu jenis aktuator elektromekanis yang digunakan untuk mengontrol gerakan sudut atau posisi dengan tingkat presisi yang tinggi. Servo motor bekerja berdasarkan prinsip sistem kendali umpan balik tertutup (*closed-loop feedback control system*), di mana terdapat mekanisme untuk membandingkan posisi keluaran aktual dengan posisi yang diinginkan, kemudian mengoreksi perbedaan tersebut secara otomatis [15]. Teknologi ini memungkinkan servo motor untuk mempertahankan posisi poros secara stabil meskipun terdapat gangguan luar, sehingga cocok digunakan pada berbagai aplikasi yang membutuhkan kendali presisi, seperti robotika, otomasi industri, dan sistem penggerak presisi.

Pada dasarnya, servo motor dikendalikan menggunakan sinyal PWM (*Pulse Width Modulation*) yang diberikan melalui kabel kontrol. Sinyal PWM ini menentukan posisi sudut dari poros motor dengan cara mengubah lebar pulsa (*duty cycle*). Misalnya, pulsa berdurasi 1,5 ms biasanya digunakan untuk posisi tengah (90°), sedangkan pulsa berdurasi 1 ms atau 2 ms digunakan untuk mengarahkan poros ke posisi ekstrem kiri (0°) atau kanan (180°). Agar servo motor dapat mempertahankan posisinya, sinyal PWM harus dikirimkan secara berulang, biasanya setiap 20 ms.

Secara umum, terdapat beberapa jenis servo motor yang diklasifikasikan berdasarkan arus listrik yang digunakan, jenis gerakan, serta mode operasinya.

Berdasarkan jenis arusnya, servo motor dibagi menjadi dua, yaitu servo motor DC (arus searah) dan servo motor AC (arus bolak-balik). Servo DC biasanya digunakan untuk aplikasi presisi kecil seperti robotika, sedangkan servo AC digunakan untuk aplikasi industri berskala besar. Berdasarkan jenis gerakannya, terdapat servo motor rotari (putar) dan servo motor linier. Sementara itu, berdasarkan mode operasi, terdapat dua jenis utama, yaitu servo posisi dan servo rotasi kontinu. Servo posisi memiliki sudut rotasi terbatas, umumnya 0° hingga 180° , dan digunakan dalam aplikasi yang membutuhkan kontrol posisi tetap. Adapun servo rotasi kontinu mampu berputar secara terus-menerus hingga 360° , bahkan lebih, sehingga cocok untuk aplikasi yang membutuhkan gerak rotasi berkelanjutan seperti penggerak roda robot.

Pada sistem yang digunakan dalam penelitian ini, jenis servo motor yang dipilih adalah MG996R. Servo MG996R merupakan jenis servo rotasi kontinu 360° dengan torsi tinggi dan gear berbahan logam, yang menjadikannya tahan lama dan mampu bekerja dalam kondisi beban menengah hingga berat. Motor ini dapat dioperasikan pada tegangan 4,8 V hingga 7,2 V, dengan torsi maksimum sebesar 11 kg·cm pada tegangan 6 V. Kecepatan rotasinya cukup responsif, yaitu sekitar 0,13 detik untuk rotasi 60° pada tegangan 6 V. Selain itu, servo ini kompatibel dengan berbagai mikrokontroler, seperti Arduino dan ESP32, sehingga memudahkan integrasi dalam sistem otomasi atau robotika.



Gambar 2.18 Cara Kerja Servo Berdasarkan PWM

Pada Gambar 2.18 dijelaskan bahwa servo motor bekerja berdasarkan sinyal *Pulse Width Modulation (PWM)*, yaitu teknik modulasi pulsa yang digunakan untuk mengendalikan arah dan kecepatan rotasi poros motor. Sinyal ini dikirimkan secara berkala, umumnya setiap 20 milidetik (50 Hz), dari mikrokontroler ke pin kendali servo. Pada jenis servo rotasi kontinu seperti MG996R, lebar pulsa menentukan arah dan kecepatan rotasi, bukan posisi tetap sebagaimana pada servo posisi. Pulsa dengan lebar sekitar 1,5 milidetik akan menghentikan rotasi (diam), sedangkan pulsa yang lebih pendek dari 1,5 milidetik menyebabkan poros berputar ke arah kiri (*counterclockwise*) dan pulsa yang lebih panjang dari 1,5 milidetik menyebabkan poros berputar ke arah kanan (*clockwise*), dengan kecepatan yang meningkat seiring dengan semakin jauhnya lebar pulsa dari nilai tengah. Oleh karena itu, pengendalian *PWM* yang presisi memungkinkan pengguna untuk mengatur gerakan servo secara halus dan sesuai dengan kebutuhan aplikasi.

Dengan mempertimbangkan spesifikasi seperti pada tabel 2-8, prinsip kerja, serta fleksibilitas dalam penggunaannya, servo MG996R menjadi pilihan yang tepat untuk sistem yang memerlukan penggerak rotasi kontinyu dengan tingkat kendali

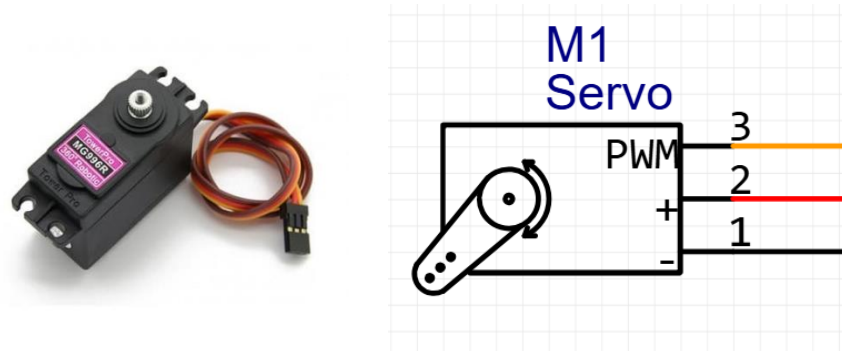
yang baik. Penggunaan servo ini juga dinilai efisien dari segi konsumsi daya dan kinerja mekanis, serta telah terbukti banyak digunakan dalam berbagai prototipe dan sistem robotika modern. Pada Gambar 2.19 ditunjukkan bentuk fisik servo yang digunakan dan simbol yang ada pada *wiring diagram*.

Tabel 2-7 Deskripsi Pin Servo

Pin	Deskripsi
VCC	Terhubung dengan <i>power supply</i>
GND	Terhubung dengan <i>ground</i> sistem
<i>Signal</i> (PWM)	Terhubung dengan pin digital PWM mikrokontroler

Tabel 2-8 Spesifikasi Servo MG966R

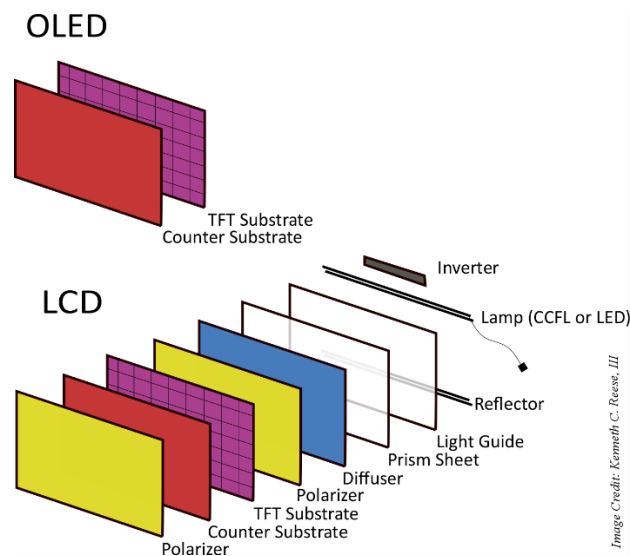
Parameter	Spesifikasi
Tipe	Servo motor <i>continous rotation</i> (360°)
Model	MG996R
Tegangan Operasi	4,8 V – 7,2 V DC
Torsi Maksimum	9,4 kg.cm (pada 4,8 V) 11 kg.cm (pada 6V)
Kecepatan Rotasi	0,19 detik/60° (pada 4,8 V) 0,13 detik/60° (pada 6 V)
Tipe Gear	<i>Metal Gear</i>
Sudut Rotasi	360° (<i>continous</i>)
Sinyal Kontrol	PWM (<i>Pulse Width Modulation</i>)
Berat	±55 gram
Dimensi	40,7 mm × 19,7 mm × 42,9 mm
Konektor Kabel	3 Kabel (VCC, GND, <i>Signal</i>)



Gambar 2.19 Bentuk Fisik dan Simbol Servo MG996R

2.2.7 Organic Light Emitting Diode

Organic Light-Emitting Diode (OLED) adalah teknologi tampilan yang memanfaatkan lapisan tipis bahan organik yang dapat memancarkan cahaya saat diberikan arus listrik. Berbeda dengan *Liquid Crystal Display (LCD)* yang memerlukan sumber cahaya latar (*backlight*), setiap piksel pada *OLED* dapat menghasilkan cahaya sendiri, memungkinkan tampilan dengan kontras tinggi dan warna hitam yang lebih pekat. Keunggulan utama *OLED* dibandingkan *LCD* meliputi kontras yang lebih tinggi, sudut pandang yang lebih lebar, serta desain yang lebih tipis dan fleksibel karena tidak memerlukan *backlight*. Namun, *OLED* juga memiliki beberapa keterbatasan, seperti potensi terjadinya *burn-in* dan umur pakai yang lebih pendek dibandingkan *LCD*.



Gambar 2.20 Perbedaan Cara Kerja OLED dan LCD

Berbeda dengan *Liquid Crystal Display* (LCD) yang membutuhkan sumber cahaya latar (*backlight*) untuk menampilkan gambar, teknologi *Organic Light-Emitting Diode* (OLED) memungkinkan setiap piksel memancarkan cahaya secara mandiri ketika diberi tegangan listrik. Hal ini dimungkinkan oleh lapisan bahan organik yang tersusun di antara dua elektroda, yaitu anoda dan katoda, di mana proses rekombinasi antara elektron dan *hole* pada lapisan emisi menghasilkan foton melalui mekanisme elektroluminesensi. Dengan tidak bergantung pada *backlight*, OLED dapat menghasilkan tampilan yang lebih kontras dengan warna hitam yang benar-benar gelap, karena piksel dapat sepenuhnya dimatikan. Selain itu, kemampuan setiap piksel untuk menyala dan mati secara individual menjadikan konsumsi daya lebih efisien, khususnya pada tampilan dengan dominasi warna gelap. Prinsip kerja ini juga memungkinkan desain tampilan yang lebih tipis, fleksibel, dan responsif dibandingkan LCD, sehingga OLED menjadi pilihan unggul dalam aplikasi perangkat portabel dan sistem tampilan modern yang mengutamakan kualitas visual dan efisiensi energi. Terdapat berbagai jenis tampilan *OLED* yang dikembangkan untuk memenuhi kebutuhan aplikasi yang beragam, antara lain:

1. Passive-Matrix OLED (PMOLED): Menggunakan struktur matriks pasif untuk mengontrol piksel, cocok untuk tampilan kecil dengan konsumsi daya rendah.
2. Active-Matrix OLED (AMOLED): Menggunakan transistor film tipis (*Thin-Film Transistor*) untuk mengontrol setiap piksel secara individual, memungkinkan resolusi tinggi dan respons cepat, umum digunakan pada perangkat mobile.
3. Transparent OLED (TOLED): Memiliki lapisan transparan yang memungkinkan cahaya melewati tampilan, cocok untuk aplikasi seperti *head-up display* dan jendela pintar.
4. Top-Emitting OLED: Dirancang untuk aplikasi dengan kepadatan piksel tinggi, seperti kamera digital dan perangkat portabel.
5. Foldable OLED: Memiliki substrat fleksibel yang memungkinkan tampilan dapat dilipat, digunakan pada perangkat dengan desain inovatif.

6. White OLED (WOLED): Menghasilkan cahaya putih yang kemudian difilter untuk menghasilkan warna, digunakan pada televisi besar untuk efisiensi produksi.

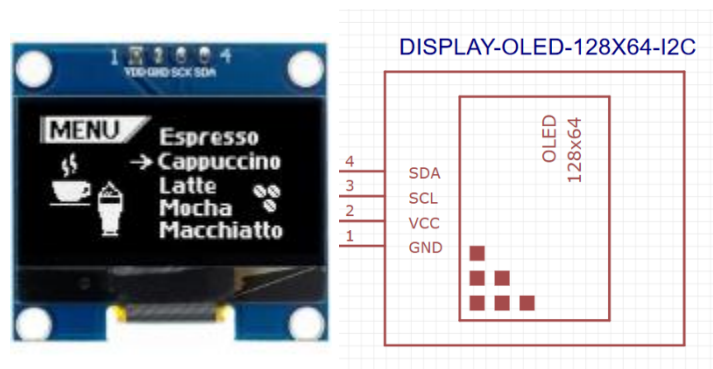
Dalam sistem yang dirancang, digunakan modul tampilan *OLED* dengan resolusi 128×64 piksel. Modul ini menggunakan antarmuka I2C (*Inter-Integrated Circuit*) untuk komunikasi data, yang kompatibel dengan mikrokontroler ESP32. Pada tabel 2-9 dijelaskan pin yang digunakan OLED 128x64 dan pada tabel 2-10 dituliskan spesifikasi dari OLED 128x64 yang digunakan oleh sistem. Adapun pada Gambar 2.21 ditampilkan bentuk fisik OLED yang digunakan beserta simbolnya pada *wiring diagram*.

Tabel 2-9 Deskripsi Pin OLED

Pin	Deskripsi
VCC	Terhubung dengan <i>power supply</i>
GND	Terhubung dengan <i>ground</i> sistem
SDA	Jalur data komunikasi I2C dari mikrokontroler ke OLED
SCL	Jalur clock I2C untuk sinkronisasi data

Tabel 2-10 Spesifikasi OLED 128x64 [16]

Parameter	Spesifikasi
Ukuran	1,3 inch
Resolusi	128 x 64
Viewing Angle	>160°
Driver Chip	SH11106
Komunikasai	I2C
Tegangan kerja	3,3 – 5V



Gambar 2.21 Bentuk Fisik dan Simbol OLED Display 128 x 64

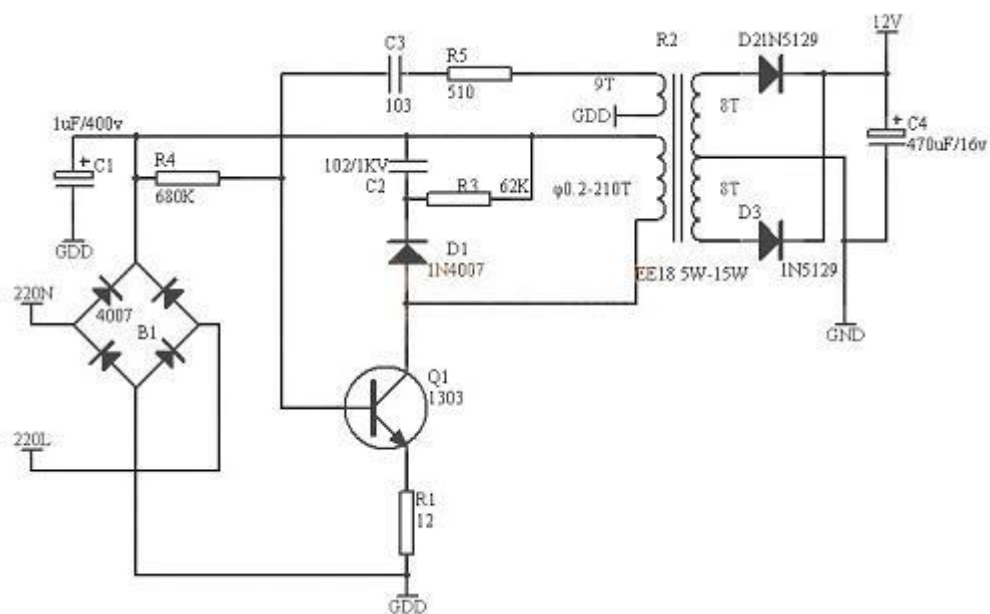
2.2.8 Power Supply

Power supply atau catu daya adalah perangkat yang menyediakan energi listrik kepada sistem elektronik dengan mengubah sumber daya listrik menjadi bentuk, tegangan, dan arus yang sesuai dengan kebutuhan beban. Dalam konteks sistem elektronik, *power supply* berperan penting dalam memastikan kinerja yang stabil dan efisien dari seluruh komponen yang terhubung.

Secara umum, *power supply* dapat diklasifikasikan menjadi beberapa jenis berdasarkan karakteristik dan metode konversinya:

1. Catu Daya Linier (*Linear Power Supply*): Menggunakan transformator untuk menurunkan tegangan AC, diikuti oleh penyearah dan regulator linier untuk menghasilkan tegangan DC yang stabil. Meskipun memiliki keunggulan dalam hal kebisingan rendah, efisiensinya cenderung rendah dan ukurannya relatif besar.
2. Catu Daya Mode-Saklar (*Switch-Mode Power Supply* - SMPS): Menggunakan teknik pensaklaran frekuensi tinggi untuk mengubah dan mengatur tegangan. SMPS lebih efisien dan kompak dibandingkan catu daya linier, namun dapat menghasilkan lebih banyak kebisingan elektromagnetik.
3. Catu Daya Teregulasi dan Tidak Teregulasi: Catu daya teregulasi mempertahankan tegangan output yang konstan meskipun terjadi perubahan pada beban atau tegangan input, sedangkan catu daya tidak teregulasi tidak memiliki mekanisme pengaturan tersebut.

Prinsip kerja *power supply* melibatkan serangkaian proses konversi energi listrik yang bertujuan untuk menghasilkan tegangan DC yang stabil dan sesuai dengan kebutuhan beban. Proses ini diawali oleh transformator (*transformer*) yang berfungsi mengubah tegangan AC dari sumber listrik ke tingkat tegangan yang diinginkan, baik dengan menaikkan (*step-up*) maupun menurunkan (*step-down*) tegangan. Tegangan AC yang telah disesuaikan kemudian diarahkan ke tahap penyearah (*rectifier*) yang menggunakan rangkaian dioda untuk mengonversi arus bolak-balik (AC) menjadi arus searah (DC), dengan konfigurasi penyearah setengah gelombang atau gelombang penuh. Tegangan DC hasil penyearahan tersebut masih mengandung riak (*ripple*), sehingga diperlukan penyaring (*filter*) berupa kapasitor untuk meredam fluktuasi tegangan agar lebih halus. Tahap terakhir adalah regulator tegangan (*voltage regulator*), yang memastikan bahwa tegangan keluaran tetap konstan dan stabil, meskipun terjadi fluktuasi pada tegangan input atau perubahan beban sistem [17].



Gambar 2.22 Rangkaian Power Supply Switching

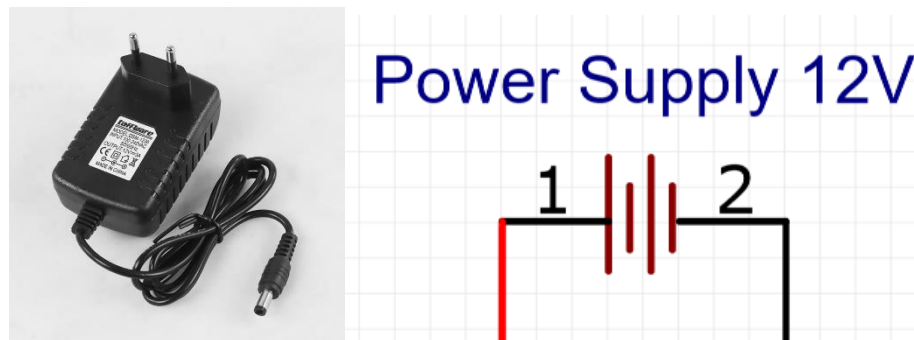
Pada Gambar 2.22 ditunjukkan rangkaian *power supply switching* dengan topologi *flyback*, di mana prinsip kerjanya didasarkan pada teknik pensaklaran berfrekuensi tinggi untuk mengubah dan mengatur tegangan listrik secara efisien.

Tegangan bolak-balik (AC) dari sumber listrik utama terlebih dahulu disearahkan menjadi tegangan searah (DC) oleh rangkaian penyearah. Tegangan DC ini kemudian diatur menggunakan sebuah transistor sebagai saklar yang bekerja secara osilatif, tanpa menggunakan pengontrol *Pulse Width Modulation (PWM)* khusus, melainkan dengan memanfaatkan rangkaian umpan balik (*feedback*) sederhana. Arus pulsa dari transistor disalurkan ke transformator frekuensi tinggi untuk disesuaikan tingkat tegangannya. Setelah itu, tegangan hasil transformasi disearahkan kembali dan difilter menggunakan kapasitor untuk menghasilkan keluaran DC yang stabil. Pendekatan ini memungkinkan efisiensi daya yang tinggi, bentuk fisik rangkaian yang ringkas, serta panas yang lebih rendah dibandingkan dengan catu daya linier.

Pada sistem yang dirancang, digunakan adaptor DC 12V 3A sebagai sumber utama catu daya. Adaptor ini termasuk jenis *Switch Mode Power Supply (SMPS)* yang memiliki efisiensi tinggi serta bentuk yang *compact*, sehingga sangat sesuai untuk aplikasi portabel. Tegangan 12V yang dihasilkan oleh adaptor tersebut kemudian diatur lebih lanjut menggunakan modul regulator tegangan tipe LM2596 untuk memenuhi kebutuhan tegangan spesifik pada setiap komponen sistem. Spesifikasi lengkap dari catu daya yang digunakan ditampilkan pada Tabel 2-11, sedangkan Gambar 2.23 memperlihatkan bentuk fisik dari adaptor serta simbol yang digunakan dalam wiring diagram.

Tabel 2-11 Spesifikasi Power Supply Yang Digunakan

Parameter	Spesifikasi
Tegangan Output	12V DC
Arus Maksimum	3A
Daya Maksimum	36W
Tipe <i>Power Supply</i>	<i>Switch Mode Power Supply</i>
Tegangan Input	100V-240V AC, 50/60Hz
Konektor Output	<i>DC Barrel Jack (male)</i>



Gambar 2.23 Bentuk Fisik dan Simbol Power Supply

2.2.9 Website Monitoring

Website merupakan kumpulan halaman dalam jaringan yang saling terhubung dan dapat diakses melalui internet menggunakan protokol HTTP atau HTTPS. Dalam konteks sistem monitoring, website berfungsi sebagai antarmuka pengguna (*user interface*) yang memungkinkan interaksi antara pengguna dengan data yang disajikan secara real-time. Dalam pengembangan sistem *monitoring* berbasis *website*, digunakan kombinasi beberapa bahasa pemrograman, yaitu *HTML*, *CSS*, *JavaScript*, dan *PHP*. Keempat bahasa tersebut bekerja secara sinergis untuk membentuk tampilan antarmuka yang menarik, fungsional, dan mampu menangani proses interaksi data antara pengguna dan peladen (*server*) secara *real-time*. Proses pengembangan dilakukan menggunakan *Visual Studio Code (VS Code)*, yang menyediakan berbagai fitur serta ekstensi pendukung dalam kegiatan penulisan kode (*coding*), pengujian (*testing*), hingga pelacakan dan perbaikan kesalahan (*debugging*).

HTML merupakan bahasa markah standar yang digunakan untuk menyusun struktur dasar sebuah halaman *web*. Bahasa ini tidak bersifat logis seperti bahasa pemrograman lainnya, namun berperan penting dalam membentuk kerangka halaman, mulai dari bagian kepala (*header*), paragraf, formulir (*form*), hingga elemen-elemen penting lainnya, seperti tabel dan tombol. Contoh implementasi *HTML* dalam sistem *monitoring* ini dapat ditemukan pada halaman formulir registrasi pengguna baru, yang menggunakan elemen `<form>`, `<input>`, dan `<button>` untuk menerima masukan dari pengguna. Selain itu, *HTML* juga

digunakan untuk menampilkan data *monitoring* dalam bentuk tabel dengan memanfaatkan elemen `<table>`, `<tr>`, dan `<td>`. Setiap elemen tersebut memiliki atribut tertentu yang menentukan perilaku dan tampilan dasar sebelum diubah melalui *CSS*.

CSS (Cascading Style Sheets) digunakan untuk mengatur tampilan visual dari elemen-elemen *HTML* agar lebih menarik, intuitif, dan mudah digunakan. Dalam proyek ini, *CSS* dimanfaatkan untuk merancang antarmuka sistem *monitoring*, seperti *dashboard* yang menampilkan informasi *RFID*, penggunaan pelindung telinga (*earplug*), dan data pengisian ulang (*refill*). Elemen-elemen seperti warna latar, ukuran huruf, *margin*, *padding*, posisi elemen, hingga animasi ringan diatur dengan menggunakan *CSS*. Salah satu teknik penting yang digunakan adalah *media query*, yang memungkinkan tampilan halaman menyesuaikan diri dengan berbagai ukuran layar (*responsive design*). Dengan pendekatan ini, halaman *monitoring* tetap nyaman diakses baik melalui perangkat *desktop* maupun perangkat *seluler*. Selain itu, penggunaan *class* dan *ID* pada elemen *HTML* dihubungkan dengan *CSS* melalui *selector*, sehingga tiap elemen dapat memiliki gaya visual yang konsisten dan mudah dimodifikasi jika diperlukan.

JavaScript merupakan bahasa pemrograman yang berjalan di sisi klien (*client-side*) dan digunakan untuk menambahkan interaktivitas pada halaman *web*. Dalam sistem *monitoring* ini, *JavaScript* memiliki beberapa fungsi utama, antara lain:

1. *JavaScript* memeriksa apakah seluruh masukan dari pengguna telah diisi dengan benar sebelum data dikirimkan ke peladen.
2. Pembaruan data secara *real time* dengan menggunakan teknik *AJAX (Asynchronous JavaScript and XML)*, data *monitoring* yang disimpan di peladen dapat diperbarui pada halaman *web* tanpa perlu memuat ulang seluruh halaman.
3. Manipulasi *DOM (Document Object Model)*.

PHP (Hypertext Preprocessor) adalah bahasa pemrograman sisi *server* yang berfungsi untuk memproses logika program, menangani permintaan pengguna, serta menghubungkan sistem dengan basis data *MySQL*. Dalam implementasinya, *PHP* memiliki peran penting dalam menangani berbagai proses logika di sisi *server*.

Pertama, *PHP* digunakan untuk menangani masukan yang dikirimkan melalui formulir *HTML*, yang selanjutnya akan diproses untuk disimpan ke dalam basis data atau digunakan dalam proses autentikasi pengguna. Selain itu, *PHP* juga berfungsi untuk mengakses data dari basis data, dalam hal ini *MySQL*, dengan menggunakan perintah *SELECT*, kemudian menampilkannya dalam format *HTML* pada halaman *web*. Tidak hanya itu, *PHP* juga mengatur logika *backend* lainnya, termasuk pembatasan jumlah pemindaian *RFID* harian serta pengelolaan akun pengguna agar sistem dapat berjalan secara terstruktur dan aman. Kode *PHP* ditulis dalam berkas dengan ekstensi *.php* dan sering kali dikombinasikan dengan kode *HTML* untuk menghasilkan halaman dinamis. Sebagai contoh, pada halaman *dashboard*, *PHP* digunakan untuk mengambil dan menampilkan data terkini dari basis data.

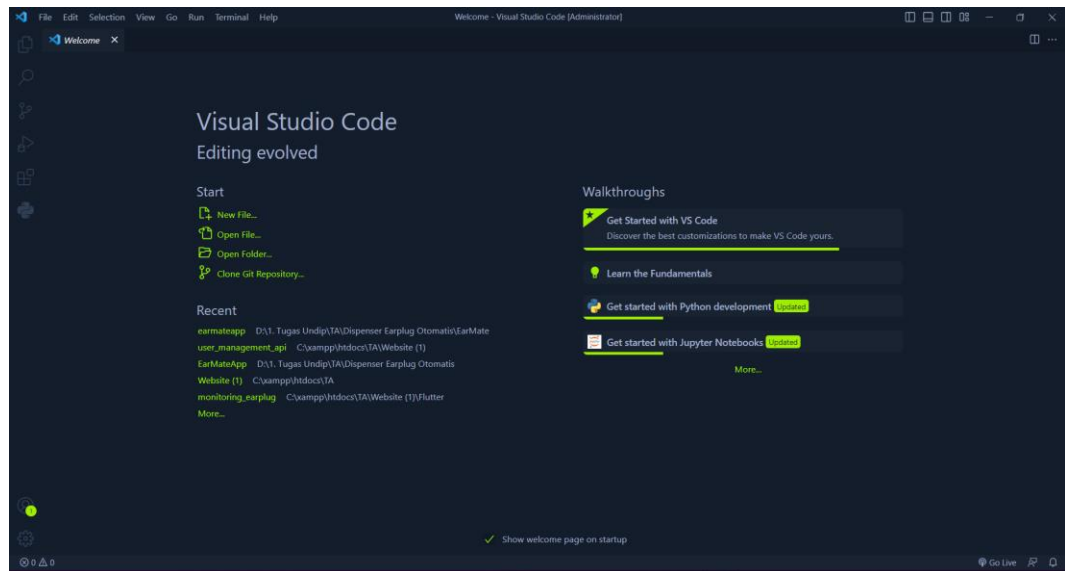
Dalam proses pengembangan *website*, diperlukan perangkat lunak pendukung berupa *Visual Studio Code (VS Code)*, yaitu sebuah *text editor* modern yang dirancang untuk mendukung aktivitas *pemrograman* dalam berbagai bahasa. Salah satu aspek penting dalam proses *coding* menggunakan *VS Code* adalah penamaan dan penyimpanan berkas yang harus disesuaikan dengan format ekstensi dari masing-masing bahasa pemrograman yang digunakan. Hal ini bertujuan agar *editor* dapat mengenali sintaksis dengan benar dan memberikan fitur bantu seperti *highlighting*, *auto-complete*, serta *debugging* secara optimal. Sebagaimana ditunjukkan pada Gambar 2.24, berkas-berkas dalam proyek sistem monitoring ini disimpan menggunakan ekstensi yang sesuai, antara lain *dashboard.html* untuk struktur halaman *web* berbasis *HTML*, *style_dashboard.css* untuk pengaturan gaya visual menggunakan *CSS*, *proses.js* untuk logika interaktif dengan *JavaScript*, dan *export_data.php* untuk pemrosesan data di sisi *server* menggunakan *PHP*. Penyesuaian ekstensi ini menjadi bagian krusial dalam menjaga keteraturan struktur proyek serta memudahkan proses pengembangan dan pemeliharaan sistem secara keseluruhan.



Gambar 2.24 Format Penyimpanan File Kode Sesuai Bahasa Yang Digunakan

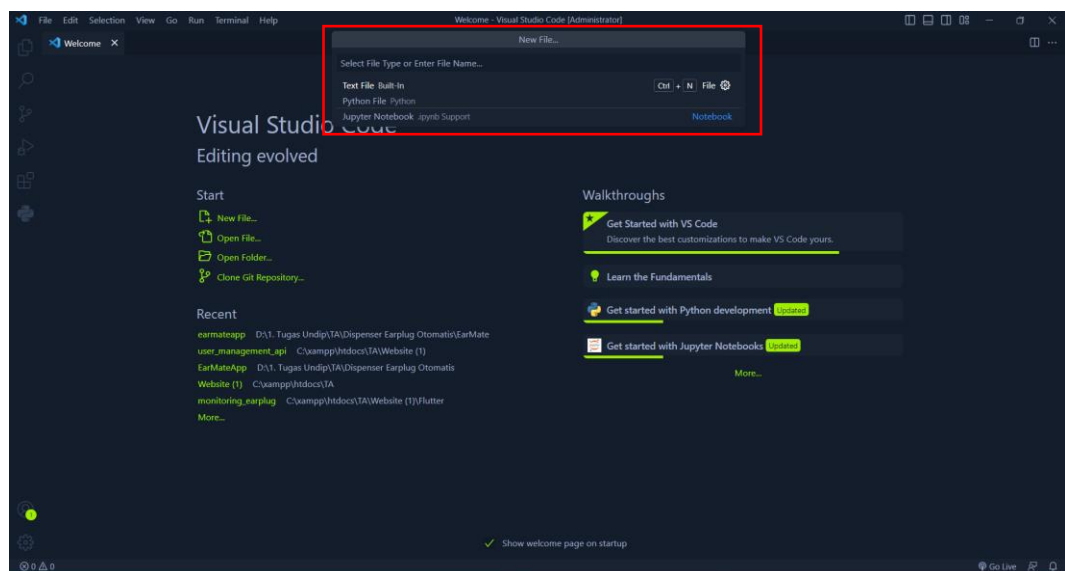
Untuk mulai membuat website menggunakan VS Code dapat melakukan langkah-langkah sebagai berikut.

1. Buka *Visual Studio Code* sehingga akan menunjukkan tampilan awal seperti pada Gambar 2.25.



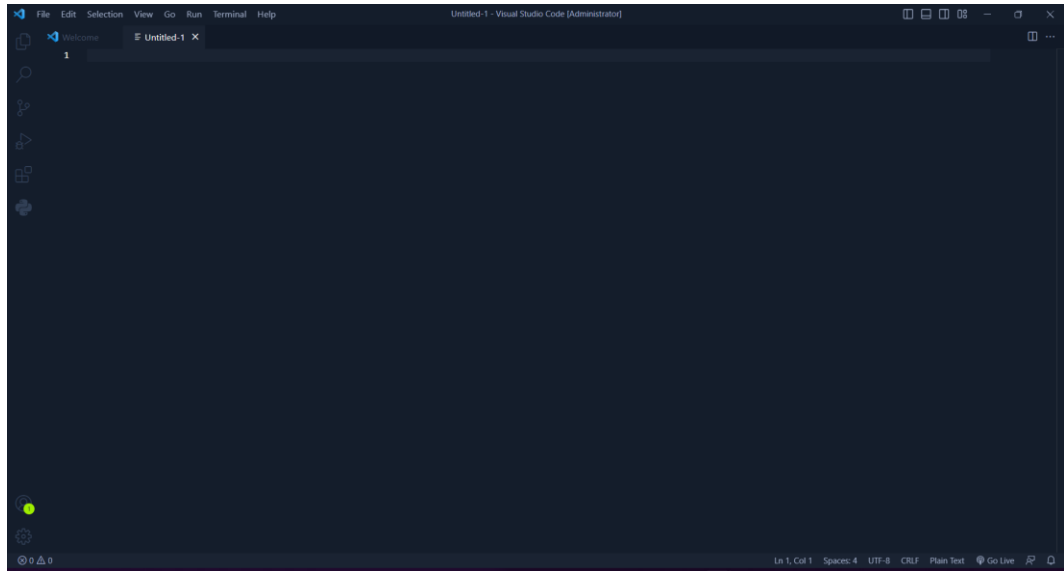
Gambar 2.25 Tampilan Awal Visual Studio Code

2. Pilih menu *New File* dan pilih *Text File* seperti pada Gambar 2.26.



Gambar 2.26 Menu Text File Untuk Memulai Pemrograman

3. Setelah memilih menu *Text File*, pengguna akan diarahkan ke *workspace* utama seperti pada Gambar 2.27 yang siap digunakan untuk pemrograman.



Gambar 2.27 Workspace Utama Visual Studio Code

2.2.10 Aplikasi Mobile

Aplikasi mobile merupakan perangkat lunak yang dirancang untuk dijalankan pada perangkat bergerak seperti ponsel pintar (*smartphone*) atau komputer tablet. Aplikasi ini memungkinkan pengguna untuk berinteraksi secara langsung dengan sistem guna menjalankan fungsi tertentu. Berdasarkan platform dan metode pengembangannya, aplikasi mobile terbagi menjadi beberapa jenis, yaitu aplikasi *native*, *hybrid*, dan *web*. Aplikasi *native* dikembangkan secara khusus untuk satu sistem operasi, seperti Android atau iOS, menggunakan bahasa pemrograman dan pustaka (*library*) yang sesuai dengan sistem operasi tersebut. Aplikasi jenis ini cenderung memiliki performa yang lebih optimal karena mampu mengakses seluruh fitur perangkat keras dan sistem operasi secara langsung.

Dalam pengembangan sistem ini, aplikasi mobile yang dibuat termasuk dalam kategori aplikasi *native* untuk platform Android. Proses pengembangannya dilakukan menggunakan Flutter, sebuah *Software Development Kit* (SDK) yang bersifat *open-source* dan dikembangkan oleh Google. Flutter memungkinkan pengembangan aplikasi dengan satu basis kode yang dapat dikompilasi secara

langsung ke dalam kode asli (*native code*) untuk berbagai platform, termasuk Android dan iOS. Meskipun mendukung pengembangan lintas platform, Flutter tetap menghasilkan performa mendekati aplikasi *native* karena tidak menggunakan lapisan interpretasi seperti pada kerangka kerja *hybrid*.

Flutter adalah kerangka kerja *open-source* atau SDK yang dikembangkan Google untuk membuat antarmuka pengguna aplikasi (*Customer Interface/UI*) yang berkinerja tinggi dan dapat dipublikasikan untuk platform Android dan iOS menggunakan basis kode tunggal. Fitur *hot reload* yang disediakan oleh Flutter memungkinkan setiap perubahan kode ditampilkan secara langsung tanpa perlu kompilasi ulang, sehingga mempercepat proses pengembangan. Flutter menggunakan bahasa pemrograman Dart, yang mirip dengan Java dan JavaScript. Dart adalah bahasa komputer yang dikembangkan Google untuk pemrograman tujuan umum (*general-purpose programming language*), yang dapat digunakan untuk membangun aplikasi Android, *web front-end*, IoT, *backend* (CLI), hingga *game*. Dart merupakan bahasa pemrograman bertipe dinamis yang disediakan dalam paket Dart. Bahasa ini mudah digunakan, memiliki implementasi kinerja tinggi, dan tidak memerlukan kompilasi terdahulu [18].

Selain itu, Dart dirancang untuk pengembangan aplikasi modern dengan pendekatan *object-oriented* dan *strong typing*, sehingga cocok untuk membangun antarmuka pengguna yang kompleks dan interaktif. Bahasa ini mendukung paradigma *asynchronous programming* secara bawaan, yang memudahkan pengembang dalam menangani proses *real-time* atau komunikasi data yang berlangsung secara terus-menerus, seperti pemantauan sensor atau pembacaan basis data secara langsung. Salah satu keunggulan utama Flutter adalah pendekatan pengembangan berbasis *widget*, di mana hampir semua elemen *interface* dari struktur tata letak hingga kontrol interaktif dibangun menggunakan komponen *widget* yang dapat dikustomisasi dan disusun sesuai kebutuhan.

Dalam penelitian ini, aplikasi mobile dirancang untuk memberikan informasi secara *real-time* mengenai penggunaan earplug oleh pengguna serta status stok yang tersedia. Selain fungsi monitoring, aplikasi juga dibekali fitur untuk mengubah batas maksimum pemindaian RFID harian, sehingga dapat disesuaikan

berdasarkan kebutuhan operasional. Semua data diperoleh dari basis data yang dikelola menggunakan phpMyAdmin pada server lokal. Proses pengembangan dilakukan menggunakan Visual Studio Code (VS Code), sebuah lingkungan pengembangan terintegrasi (*Integrated Development Environment/IDE*) yang populer karena ringan dan mendukung berbagai ekstensi untuk pengembangan Flutter dan Dart. Logo Flutter yang merupakan *framework* dalam pengembangan aplikasi *mobile* ditunjukkan pada Gambar 2.28.

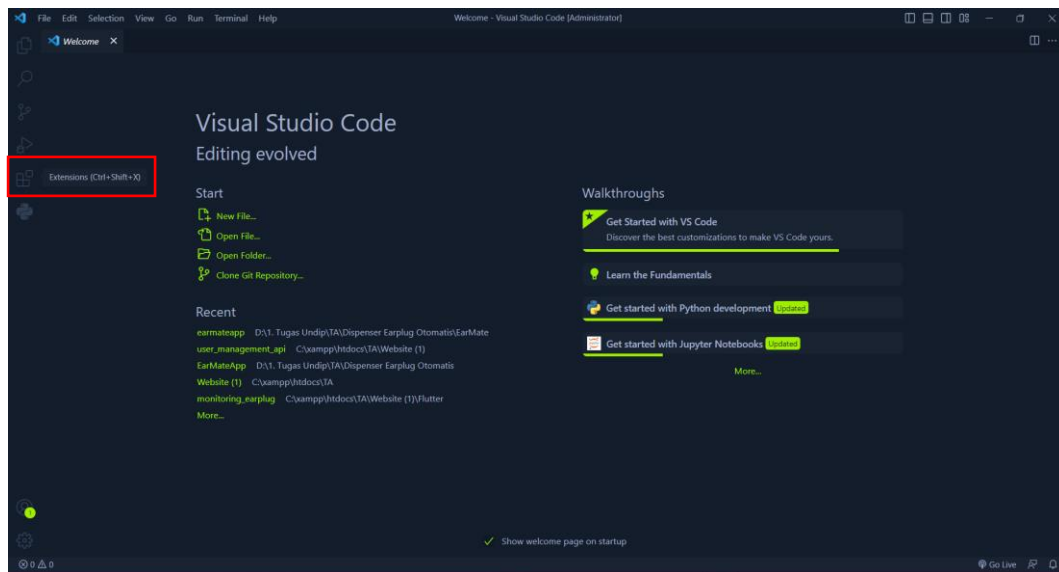


Gambar 2.28 Logo Flutter

Untuk dapat membangun aplikasi mobile menggunakan Flutter, langkah awal yang harus dilakukan adalah menyiapkan lingkungan pengembangan yang sesuai. Langkah-langkah yang harus dilakukan untuk melakukan *install framework* Flutter pada VS Code adalah sebagai berikut.

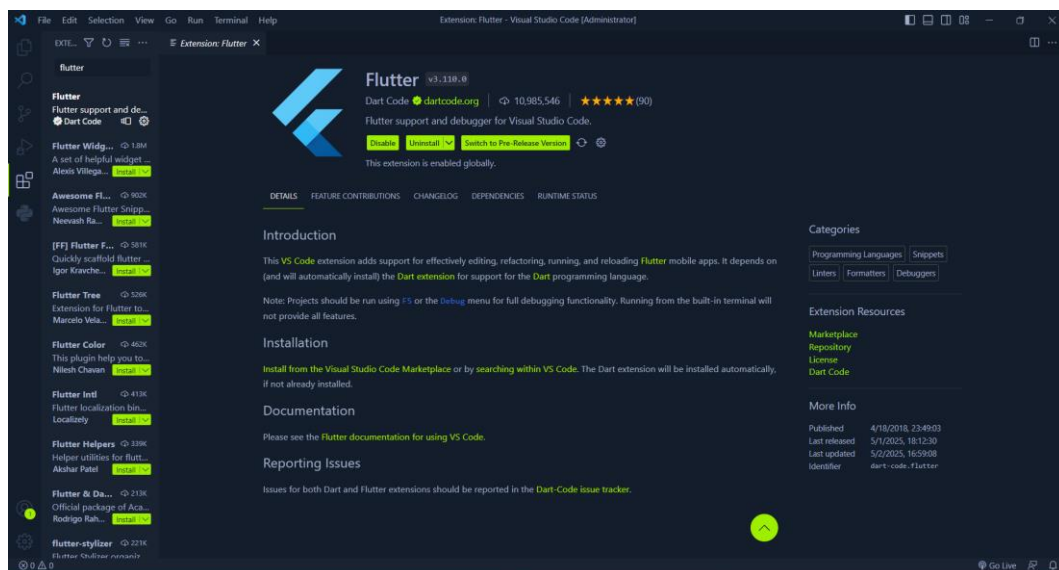
1. Pastikan Flutter telah diunduh dari web <https://flutter.dev/>.
2. Buka aplikasi Visual Studio Code hingga muncul seperti pada Gambar 2.25.

3. Klik menu *extension* yang berada pada *sidebar* seperti pada Gambar 2.29.



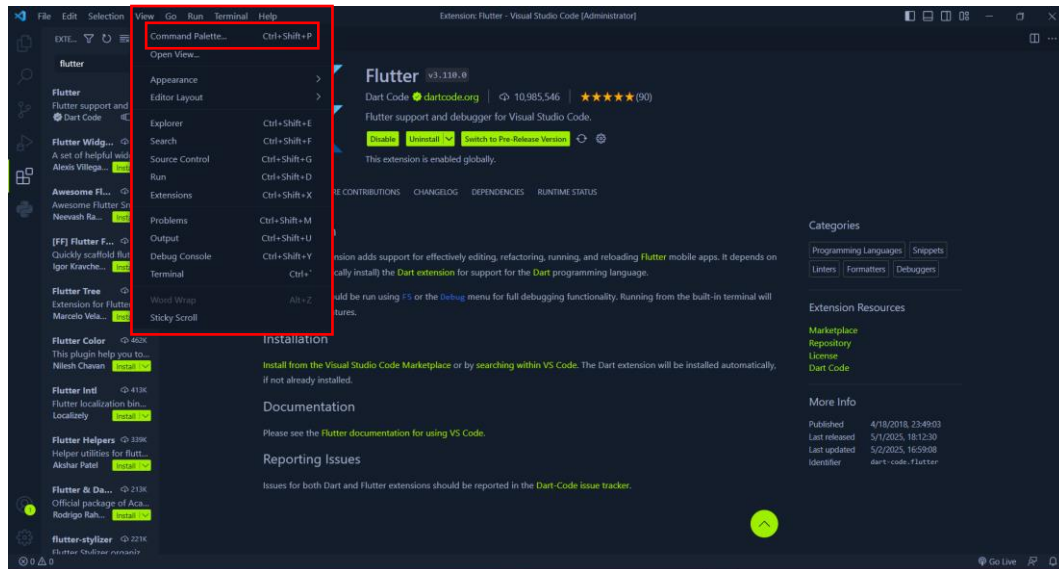
Gambar 2.29 Letak Menu Extension Pada Sidebar

4. Ketikkan kata Flutter dan lakukan instalasi seperti pada Gambar 2.30.



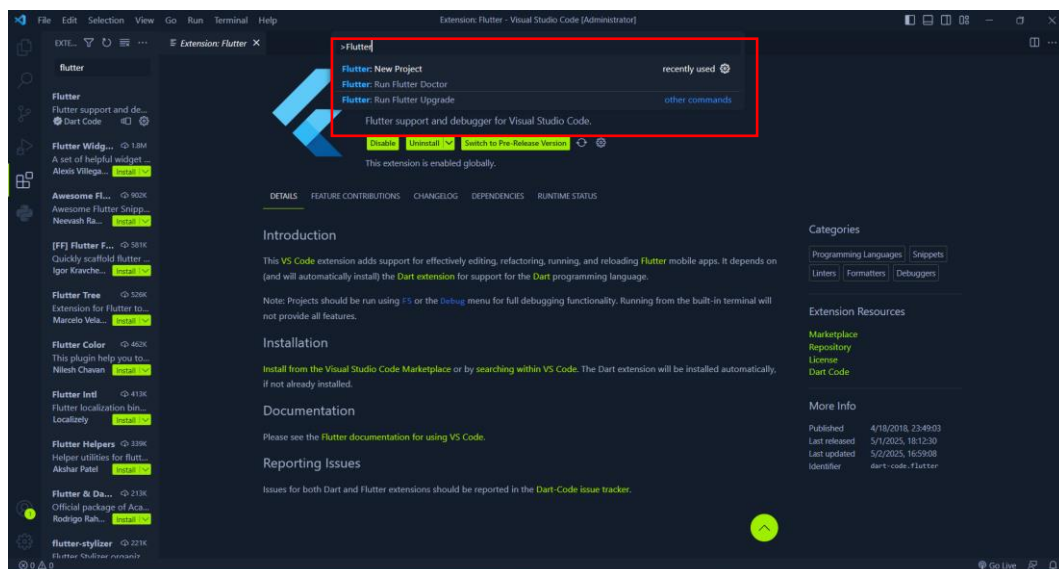
Gambar 2.30 Instalasi Flutter Pada Menu Extension

5. Apabila *extension* Flutter telah terpasang, pilih menu *View* dan pilih *Command Palette* seperti pada Gambar 2.31.



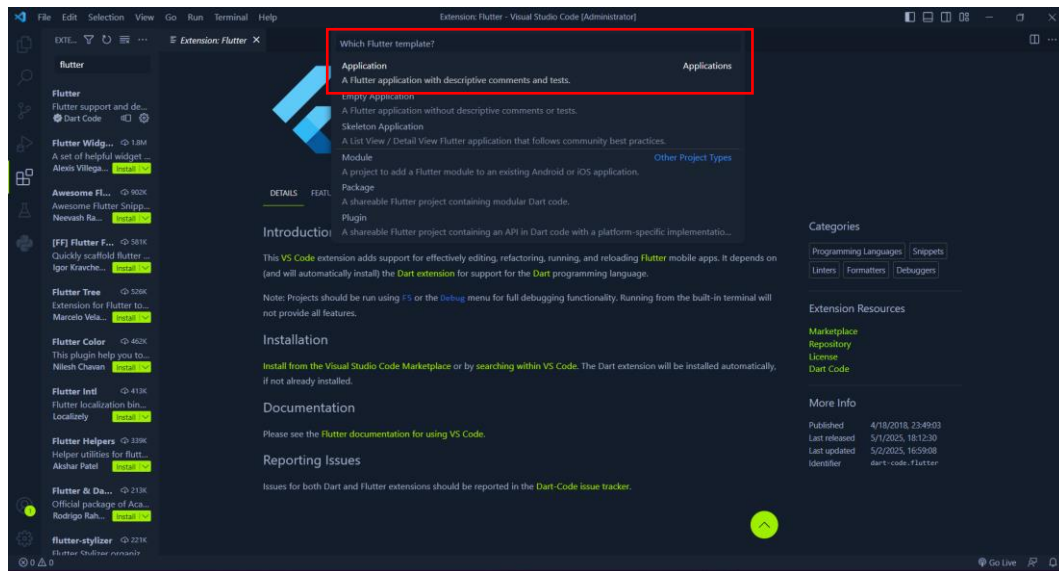
Gambar 2.31 Submenu Command Palette Pada Menu View

6. Ketikkan kata Flutter kemudian pilih *Flutter::New Project* seperti pada Gambar 2.32.



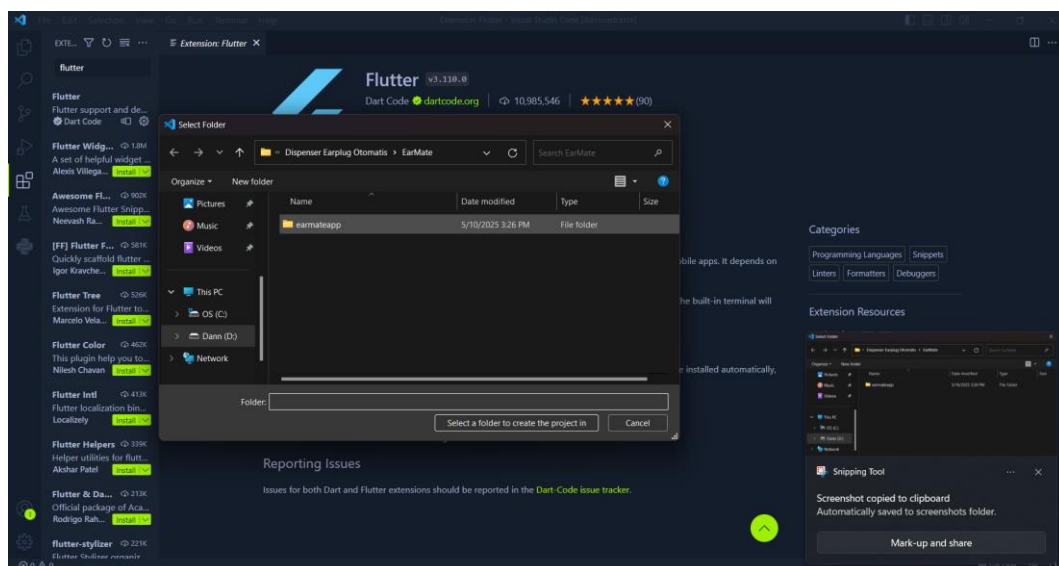
Gambar 2.32 Menu Flutter::New Project

7. Pilih menu lanjutan *Application* agar mendapat *template* yang berisi dengan deskripsi dari folder yang akan dilakukan pengkodean seperti pada Gambar 2.33.



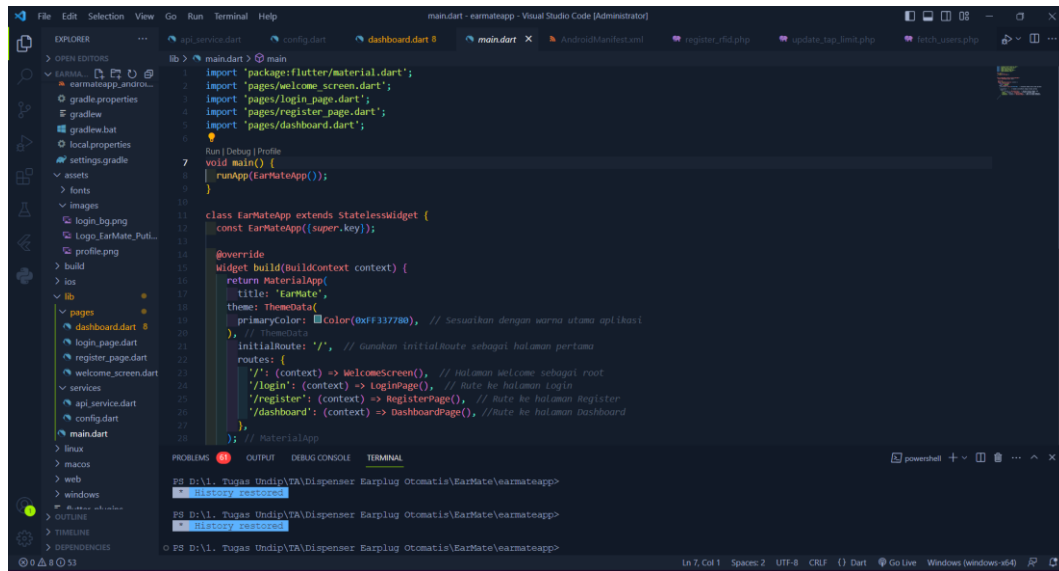
Gambar 2.33 Menu Lanjutan Application

8. Tentukan nama dan lokasi *project* yang akan dibuat seperti pada Gambar 2.34.



Gambar 2.34 Lokasi Penyimpanan Pada Project Flutter

9. Apabila aplikasi berhasil disimpan maka *user* akan diarahkan ke *main.dart* untuk memulai kode seperti pada Gambar 2.35.



Gambar 2.35 Main.dart Sebagai File Pertama Yang Dibuka

2.2.11 Visual Studio Code

Visual Studio Code (VS Code) merupakan sebuah *source-code editor* yang dikembangkan oleh Microsoft dan tersedia secara gratis untuk berbagai sistem operasi seperti Windows, macOS, dan Linux. VS Code menjadi salah satu editor yang populer digunakan oleh para pengembang perangkat lunak karena ringan, fleksibel, serta mendukung berbagai bahasa pemrograman dan framework modern.

Dalam perancangan sistem pada proyek ini, VS Code digunakan sebagai lingkungan pengembangan utama (*development environment*) untuk menulis kode baik pada sisi *frontend* maupun *backend*. Bahasa pemrograman yang digunakan antara lain HTML, CSS, JavaScript, dan PHP untuk pengembangan tampilan dan fungsionalitas pada website monitoring. Selain itu, VS Code juga digunakan untuk membangun aplikasi mobile berbasis Flutter yang ditulis dalam bahasa Dart.

Salah satu keunggulan VS Code adalah kemampuannya dalam mendukung berbagai ekstensi yang dapat disesuaikan dengan kebutuhan pengembangan. Dalam proyek ini, beberapa ekstensi seperti Flutter, Dart, dan PHP IntelliSense sangat membantu dalam mempercepat proses penulisan kode serta mendeteksi kesalahan

secara dini. Penggunaan VS Code juga memungkinkan integrasi yang baik dengan *version control* seperti Git, serta menyediakan fitur *live preview* yang memungkinkan pengujian antarmuka web secara langsung melalui browser. Hal ini memudahkan dalam proses pengembangan dan pengujian antarmuka monitoring selama tahap perancangan sistem.

Dengan semua kelebihan tersebut, VS Code dipilih sebagai alat utama dalam proses pengembangan karena mampu memberikan efisiensi dan kenyamanan dalam menulis serta mengelola kode program.

2.2.12 Database

Database merupakan komponen penting dalam suatu sistem informasi yang berfungsi untuk menyimpan, mengelola, dan mengakses data secara sistematis dan terstruktur. Dalam perancangan sistem pada proyek ini, *database* digunakan sebagai pusat penyimpanan data yang berkaitan dengan pengguna dan aktivitas pada sistem dispenser *earplug* otomatis berbasis RFID.

Jenis *database* yang digunakan adalah MySQL, yang diakses melalui *phpMyAdmin* secara lokal (*localhost*). *PhpMyAdmin* merupakan antarmuka berbasis web yang mempermudah pengguna dalam mengelola basis data MySQL tanpa perlu menggunakan antarmuka baris perintah (*command line*). Pemilihan *phpMyAdmin* didasarkan pada kemudahannya dalam melakukan pembuatan tabel, pengolahan data, serta integrasi dengan *backend* berbasis PHP.

Struktur *database* dirancang untuk menyimpan berbagai informasi penting, antara lain:

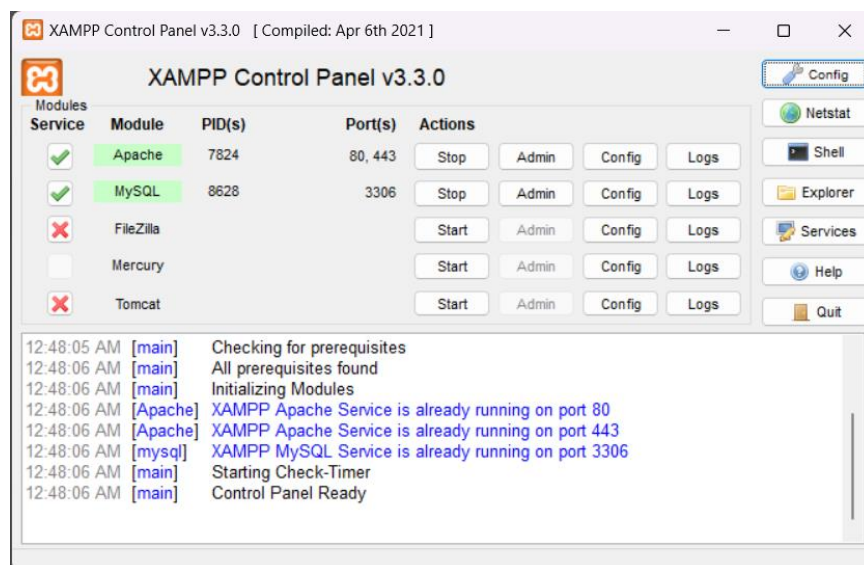
1. Nama pengguna
2. Nomor induk
3. Departemen
4. Tanggal penggunaan
5. Waktu penggunaan
6. Tag RFID
7. Status penggunaan
8. Jumlah penggunaan *earplug*

9. ID *refill*
10. Jumlah *refill*
11. Batas jumlah *tap* (*tap limit*)

Seluruh data tersebut dikumpulkan dan diolah oleh sistem secara otomatis guna mendukung proses pemantauan penggunaan *earplug*. Informasi yang tersimpan akan ditampilkan melalui antarmuka berbasis web dan aplikasi *mobile* yang dapat diakses oleh administrator. Dalam tahap pengembangan ini, *database* dioperasikan pada lingkungan *local server* untuk keperluan internal. Akses terhadap *database* dibatasi agar hanya dapat diakses oleh pihak yang berwenang melalui jaringan lokal. Hal ini dilakukan untuk menjaga keamanan data serta memastikan efisiensi dalam proses pengujian sistem. Dengan adanya sistem penyimpanan data berbasis *database*, seluruh proses pencatatan dan pemantauan penggunaan *earplug* menjadi lebih terstruktur, otomatis, dan terintegrasi secara menyeluruh dengan komponen perangkat keras maupun perangkat lunak pada sistem.

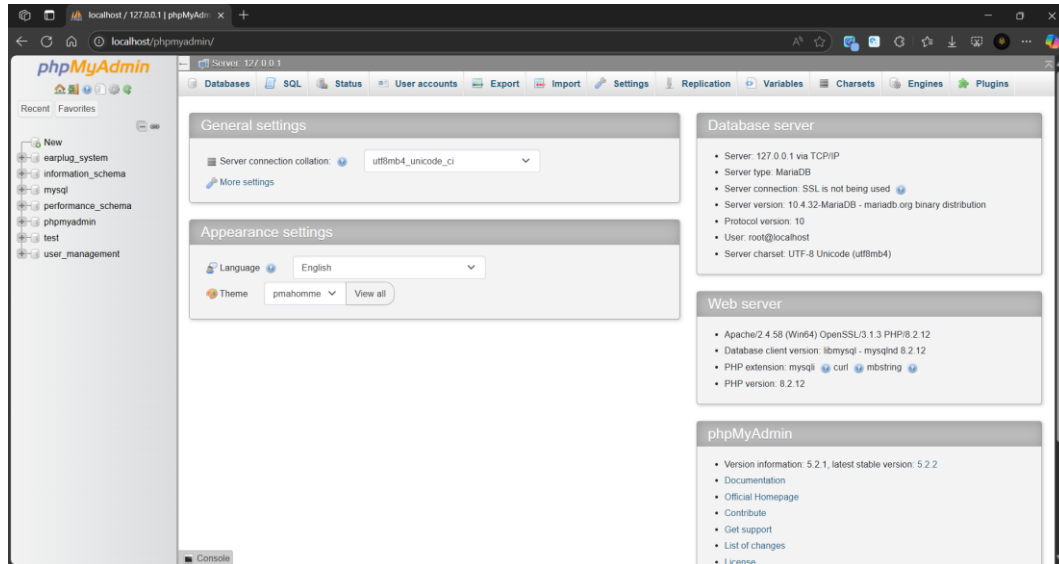
Untuk membuat *database* menggunakan *localhost phpmyadmin* dapat mengikuti langkah-langkah berikut.

1. Install XAMPP sebagai *local server environment* pada server lokal dan buka XAMPP Control Panel seperti pada Gambar 2.36. Apabila *Apache* dan *MySQL* masih belum menyala, klik tombol start.



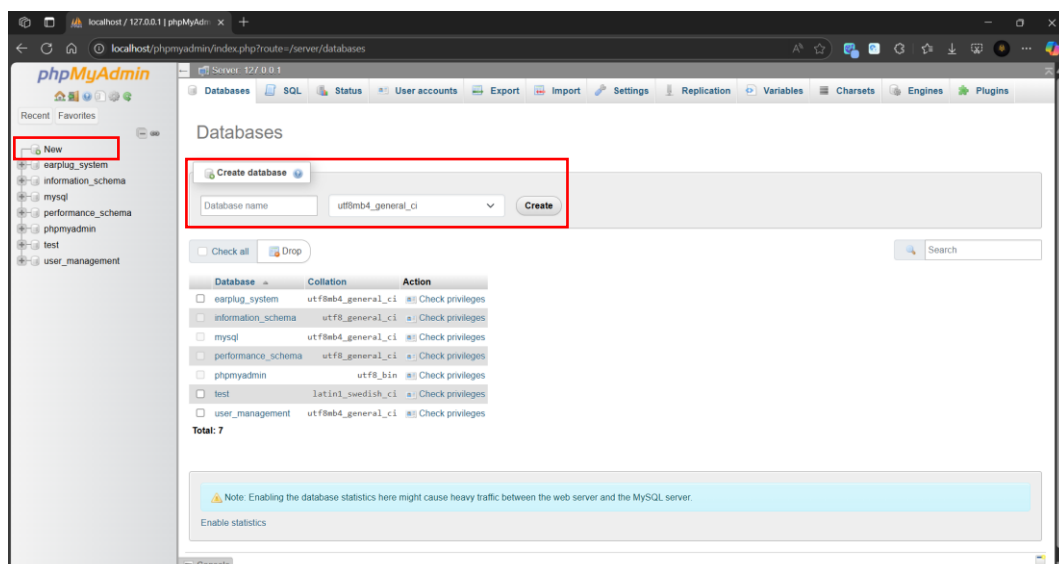
Gambar 2.36 XAMPP Control Panel

2. Buka *browser* dan ketik <http://localhost/phpmyadmin/> untuk mengakses database *localhost* seperti pada Gambar 2.37.



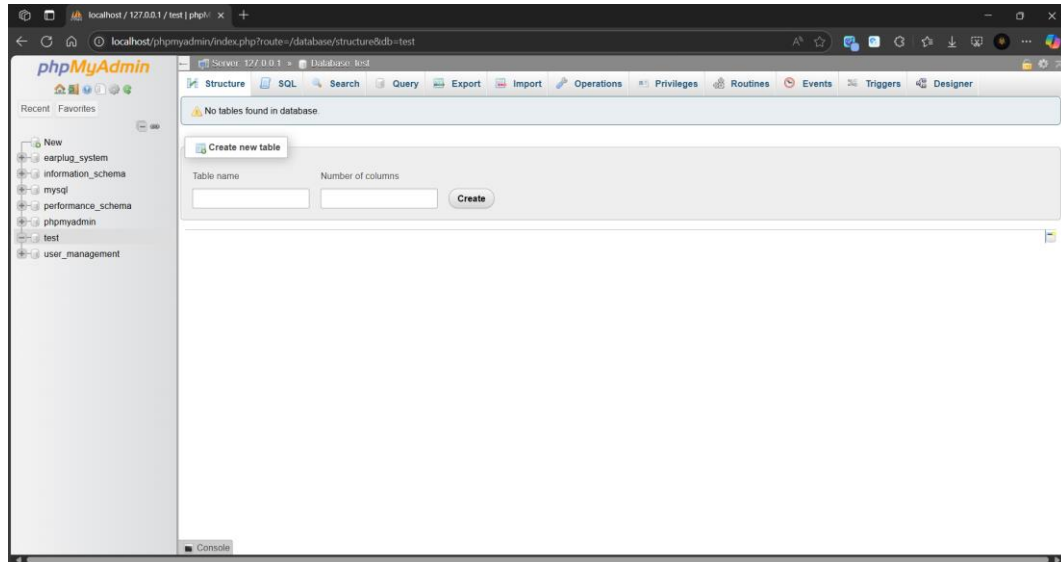
Gambar 2.37 Tampilan Localhost PHPMyAdmin

3. Klik menu *New* dan isi nama dari *database* yang akan dibuat seperti pada Gambar 2.38.



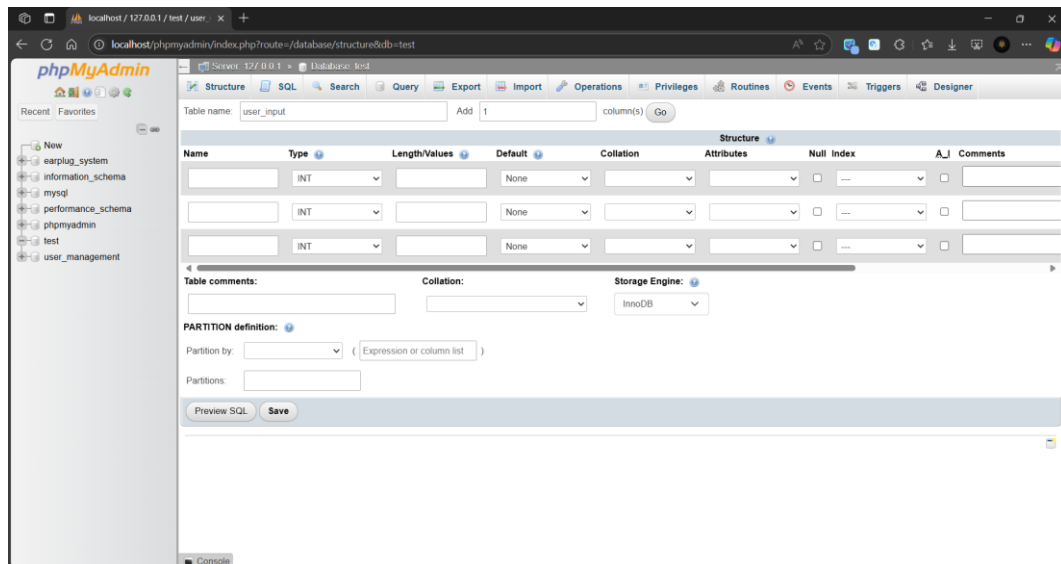
Gambar 2.38 Pengisian Nama Database Pada PHPMyAdmin

4. Tentukan nama tabel dan jumlah kolom yang akan dibuat seperti pada Gambar 2.39.



Gambar 2.39 Pengisian Nama Tabel dan Jumlah Kolom Pada PHPMYAdmin

5. Isi nama kolom dan jenis data sesuai dengan kebutuhan seperti pada Gambar 2.40.



Gambar 2.40 Pengisian Nama Tabel dan Jenis Data