

BAB II

DASAR TEORI

2.1 Tinjauan Pustaka

Setelah melihat berbagai referensi, baik dari jurnal maupun penelitian sebelumnya, ditemukan beberapa penelitian yang berkaitan dengan sistem *smart rotary parking*. Ringkasan dari penelitian sebelumnya yang digunakan sebagai dasar dan acuan untuk tugas akhir ini ditampilkan dalam Tabel 2.1.

Tabel 2. 1 Tinjauan Pustaka

Peneliti & Judul	Keterangan	Keterbatasan & Perbaikan
Bintang Rajapasha [3]	Menerapkan sistem <i>rotary parking</i> berbasis mikrokontroler untuk efisiensi pemanfaatan ruang parkir dengan antarmuka masukan kode berupa <i>keypad</i> [3].	Pada penelitian ini penggunaan <i>keypad</i> memerlukan entri kode secara manual, rawan kesalahan pengetikan (<i>human error</i>), serta berisiko terhadap kebocoran kode akses.

Inovasi yang dibawakan mengganti *keypad* dengan otentikasi *tap-in/tap-out* RFID otomatis berbasis ESP32, serta menambahkan fitur *monitoring* HMI berbasis IoT (*real-time*) untuk meningkatkan keamanan dan efisiensi akses.

Peneliti & Judul	Keterangan	Keterbatasan & Perbaikan
Muchtar <i>et al.</i> [4] <i>Real-Time Navigation of Mecanum Wheel-Based Mobile Robot in a Dynamic Environment</i>	Mengembangkan sistem <i>barrier gate</i> berbasis ESP32 dan RFID dengan performa respon pergerakan palang pintu yang stabil dan responsive [4].	Pada penelitian ini kontrol palang pintu masih bersifat umum dan berdiri sendiri, belum terintegrasi ke dalam ekosistem parkir mekanis secara menyeluruh. Inovasi yang dibawaikan yaitu mengintegrasikan kontrol <i>gate</i> RFID berbasis ESP32 langsung ke dalam ekosistem <i>smart rotary parking</i> , lengkap dengan indikator tampilan fisik dan <i>monitoring dashboard</i> IoT.
Ashari <i>et al.</i> [5] <i>Prototype Smart Parking at STMIK AKBA Based on Microcontroller Using Radio Frequency Identification (RFID) Technology</i>	Berhasil mengimplementasikan sistem <i>smart parking</i> berbasis teknologi RFID untuk otomatisasi identifikasi dan pemindaian akses kendaraan [5].	Pada Penelitian ini masih menggunakan arsitektur mikrokontroler tunggal sederhana, sehingga pemrosesan data rawan <i>delay</i> dan belum siap dikembangkan ke skala sistem terintegrasi. Inovasi yang dibawaikan yaitu menggunakan

Peneliti & Judul	Keterangan	Keterbatasan & Perbaikan
		mikrokontroler ESP32 berperforma tinggi dengan pembagian beban pemrosesan (<i>Master-Slave</i> via UART) serta konektivitas IoT untuk menjamin kecepatan data dan keandalan sistem.

2.2 Rotary Parking

Perkembangan teknologi *Internet of Things* (IoT) telah memberikan kontribusi signifikan dalam pengembangan sistem transportasi cerdas, salah satunya pada sistem manajemen parkir. Permasalahan keterbatasan lahan parkir, lamanya waktu pencarian slot parkir, serta rendahnya tingkat keamanan kendaraan menjadi tantangan yang sering ditemui pada sistem parkir konvensional. Sistem parkir tradisional umumnya masih mengandalkan pengawasan manual sehingga sering menimbulkan antrian kendaraan, ketidakaturan parkir, serta kesulitan dalam memonitor ketersediaan slot parkir secara *real-time*. Oleh karena itu, pengembangan sistem smart parking berbasis teknologi IoT menjadi solusi yang banyak dikembangkan untuk meningkatkan efisiensi, keamanan, dan kemudahan pengelolaan area parkir [7], [8]. Sistem ini memungkinkan integrasi antara sensor, mikrokontroler, serta sistem *monitoring* berbasis jaringan sehingga informasi ketersediaan parkir dapat diperoleh secara cepat dan akurat.



Gambar 2. 1 *Rotary Parking* [35]

Salah satu teknologi yang banyak digunakan dalam sistem parkir cerdas adalah *Radio Frequency Identification* (RFID) yang berfungsi sebagai sistem identifikasi otomatis pengguna. RFID bekerja dengan memanfaatkan gelombang radio untuk membaca data yang tersimpan pada tag atau kartu RFID yang memiliki identitas unik. Ketika kartu RFID ditempelkan pada modul pembaca RFID, sistem akan membaca *Unique ID* (UID) dari kartu tersebut dan mengirimkannya ke mikrokontroler untuk dilakukan proses identifikasi data. Implementasi teknologi RFID pada sistem parkir terbukti mampu meningkatkan keamanan serta mempercepat proses identifikasi pengguna dibandingkan metode manual, karena proses identifikasi dapat dilakukan secara otomatis dan memiliki tingkat akurasi yang tinggi [6], [9]. Selain itu, penggunaan RFID juga memungkinkan integrasi dengan sistem *database* sehingga data akses kendaraan dapat tercatat secara otomatis pada sistem.

Dalam sistem yang dirancang pada penelitian ini, ESP32 digunakan sebagai pengendali utama sistem yang bertugas mengolah data dari berbagai sensor dan modul yang terhubung. Mikrokontroler ESP32 dipilih karena memiliki kemampuan pemrosesan yang lebih tinggi dibandingkan mikrokontroler konvensional serta dilengkapi dengan konektivitas WiFi dan Bluetooth yang memungkinkan integrasi dengan sistem *monitoring* berbasis jaringan. Dengan kemampuan tersebut, ESP32 dapat mengontrol berbagai perangkat seperti modul RFID, motor servo, *driver*

motor, sensor pendeteksi kendaraan, serta modul komunikasi data secara terintegrasi dalam satu sistem kontrol [10], [11]. Ketika kartu RFID berhasil dibaca, ESP32 akan memberikan perintah untuk mengaktifkan motor servo yang berfungsi membuka palang pintu masuk sehingga kendaraan dapat memasuki area parkir.

Setelah kendaraan memasuki area parkir, sistem akan mengarahkan kendaraan menuju mekanisme *rotary parking* yang berfungsi untuk mengoptimalkan pemanfaatan ruang parkir secara vertikal. Mekanisme *rotary parking* bekerja dengan menggerakkan struktur parkir menggunakan motor DC yang dikendalikan melalui *driver* motor. Dengan memanfaatkan sistem parkir vertikal, kapasitas parkir dapat ditingkatkan tanpa memerlukan perluasan lahan parkir secara horizontal. Penggunaan konsep parkir bertingkat atau modular ini telah banyak diterapkan dalam sistem smart parking modern untuk mengatasi keterbatasan ruang parkir di wilayah dengan kepadatan kendaraan yang tinggi [8]. Sensor infrared digunakan untuk mendeteksi keberadaan kendaraan pada slot parkir serta membantu proses perhitungan jumlah kendaraan yang masuk ke dalam sistem.

Selain sistem kontrol dan pendeteksian slot parkir, sistem yang dirancang juga dilengkapi dengan *Human Machine Interface* (HMI) berbasis aplikasi yang berfungsi sebagai media *monitoring* sistem parkir. Informasi yang sebelumnya ditampilkan pada layar LCD seperti status slot parkir, jumlah kendaraan, serta informasi akses pengguna dapat ditampilkan kembali pada aplikasi *smartphone* melalui koneksi jaringan yang terhubung dengan ESP32. Implementasi HMI berbasis aplikasi memungkinkan operator maupun pengguna untuk memonitor kondisi sistem parkir secara *real-time* tanpa harus berada di lokasi perangkat. Sistem *monitoring* berbasis aplikasi ini juga banyak diterapkan pada sistem *smart parking* modern untuk meningkatkan kemudahan akses informasi serta mendukung konsep pengelolaan parkir berbasis IoT [9].

2.3 Metode Logika Kondisional (Rule-Based System)

Metode penelitian yang diterapkan dalam Tugas Akhir ini adalah metode rekayasa sistem (*engineering method*) dengan pendekatan sistem berbasis logika (*rule-based system*). Metode ini mencakup alur perancangan, integrasi perangkat

keras dan perangkat lunak, hingga pengujian alat untuk membangun sistem *barrier gate* parkir otomatis berbasis RFID yang mampu melakukan kontrol akses dan manajemen slot secara *real-time*.

Proses pengambilan keputusan pada sistem diatur menggunakan algoritma logika kondisional (*if-else*) berdasarkan pola penempelan (*tagging*) kartu RFID dan manajemen status memori sementara (*array storage*) pada mikrokontroler. Setiap kali kartu RFID ditempelkan, sistem terlebih dahulu memeriksa keberadaan *Unique Identifier* (UID) kartu tersebut di dalam *array* memori. Apabila UID belum tercatat, sistem memeriksa ketersediaan slot parkir; jika slot masih tersedia, UID kartu dicatat ke dalam *array* memori, status slot berkurang, dan *gate* terbuka untuk proses masuk, sedangkan jika seluruh slot telah terisi, akses masuk ditolak. Sebaliknya, apabila UID kartu ditemukan sudah tercatat di dalam *array* memori, sistem secara otomatis mengklasifikasikannya sebagai proses keluar; data UID dihapus dari *array*, status slot bertambah, dan *gate* terbuka untuk keluar kendaraan. Dengan demikian, logika kontrol tidak membandingkan UID terhadap *database* pengguna statis maupun bergantung pada pola jumlah penempelan kartu, melainkan mengevaluasi ketersediaan slot dan presensi UID secara dinamis pada memori *array* melalui satu kali penempelan kartu.

Selain menerapkan logika kontrol lokal, sistem mengintegrasikan ekosistem *Internet of Things* (IoT) dengan pembagian beban kerja berbasis arsitektur *dual microcontroller*. ESP32 Master bertugas memproses pemindaian RFID RC522, memperbarui tampilan lokal pada LCD 16x2 I2C serta OLED Display, dan mengeksekusi logika *rule-based*, kemudian mengirimkan instruksi kendali ke ESP32 Slave melalui komunikasi serial UART untuk menggerakkan motor *servo*. Secara bersamaan, ESP32 Master mentransmisikan data status ketersediaan slot parkir secara *real-time* menggunakan protokol MQTT melalui *broker* HiveMQ menuju aplikasi Android berbasis MIT App Inventor sebagai media *dashboard monitoring* jarak jauh. Pendekatan ini dipilih karena memiliki kompleksitas komputasi yang rendah, penggunaan memori RAM yang efisien, serta latensi

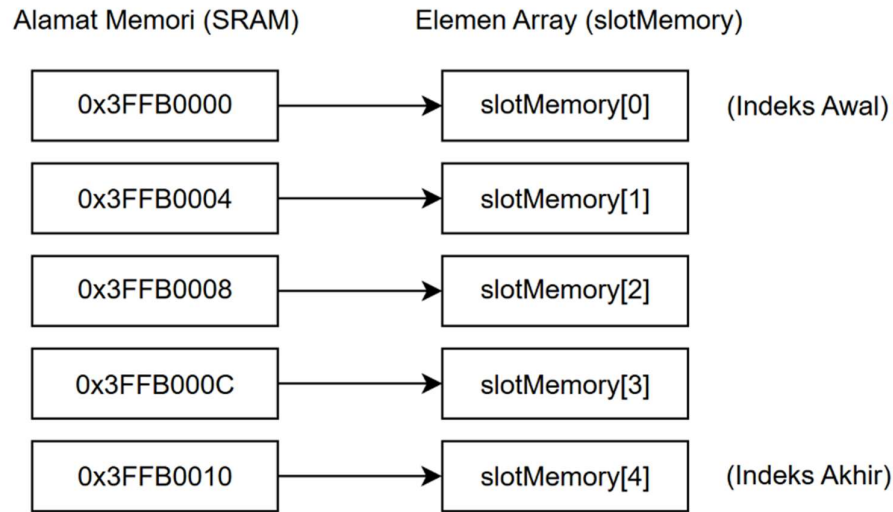
eksekusi yang sangat cepat untuk menjamin keandalan sistem kontrol akses kendaraan.

2.4 Struktur Data *Memory Array* (SRAM)

Struktur data array (larik) merupakan metode alokasi data linier yang menyimpan sekumpulan variabel ber-tipe data sama pada lokasi memori fisik yang berurutan (*contiguous memory locations*) [30]. Setiap elemen di dalam array diidentifikasi menggunakan indeks numerik berbasis nol (*zero-based indexing*) yang memetakan posisi data secara langsung pada memori RAM (*Static Random-Access Memory* / SRAM) mikrokontroler [30], [31]. Pada sistem tertanam (*embedded system*), array dimanfaatkan sebagai media penyimpanan data sementara (*buffer*) berbasis memori volatil untuk mengeliminasi ketergantungan terhadap *database* eksternal maupun penulisan berulang pada memori *non-volatile* seperti EEPROM atau Flash memori yang memiliki keterbatasan siklus penulisan [31].

Dari sudut pandang efisiensi komputasi, struktur data array menawarkan karakteristik performa yang terukur untuk pemrosesan data secara *real-time* [30]. Pengaksesan elemen berdasarkan posisi indeks (*direct access*) memiliki kompleksitas waktu konstan, di mana durasi pengambilan data tidak dipengaruhi oleh total kapasitas slot yang dialokasikan [30]. Sementara itu, proses pencarian data (*linear search*) untuk memverifikasi presensi *Unique Identifier* (UID) kartu memiliki kompleksitas waktu, dengan sistem memindai elemen dari indeks awal hingga indeks terakhir [30]. Dari segi alokasi memori (*space complexity*), konsumsi

memori berbanding lurus terhadap jumlah slot yang dideklarasikan sejak awal pembentukan program [30], [31].



Gambar 2. 2 Visualisasi Alokasi Contiguous Memory pada Array slotMemory[5] di SRAM ESP32

Berdasarkan Gambar 2.2, alokasi memori fisik untuk *array* slotMemory[5] pada SRAM ESP32 bekerja secara linier berurutan (*contiguous memory allocation*). Struktur ini memetakan lima elemen array dari indeks 0 hingga indeks 4 ke dalam blok alamat memori fisik yang berdampingan (contoh: 0x3FFB0000 hingga 0x3FFB0010).

Penerapan *array storage* pada memori SRAM ESP32 Master memberikan keunggulan teknis yang signifikan dalam mendukung operasional sistem *smart parking* [31]. Pertama, sistem mampu melakukan penanganan latensi nol (*zero latency validation*), di mana proses verifikasi keberadaan UID kartu pada memori array untuk menentukan proses masuk atau keluar diproses secara lokal tanpa terkendala *delay* jaringan internet yang umum terjadi pada pemanggilan *database cloud* [31]. Kedua, alokasi memori SRAM untuk kapasitas 5 slot parkir tergolong sangat efisien, sehingga mampu mencegah terjadinya kebocoran memori (*memory leak*) atau *stack overflow* selama alat beroperasi secara berkelanjutan [30], [31]. Selain itu, pemetaan data status UID lokal di dalam array dapat secara langsung

dikonversi menjadi susunan biner (1 untuk terisi dan 0 untuk kosong) saat pembentukan *payload* JSON untuk ditransmisikan via *broker* MQTT [31].

Meskipun alokasi pada SRAM menawarkan efisiensi komputasi yang tinggi, memori jenis ini memiliki karakteristik *volatile* [31]. Artinya, keseluruhan data yang tersimpan di dalam struktur *array* akan hilang dan ter-*reset* setiap kali mikrokontroler kehilangan daya (*power outage*) atau mengalami siklus *restart*. Oleh karena itu, operasional *array* di RAM memerlukan mekanisme integrasi komplementer dengan memori penyimpanan permanen (*non-volatile*) agar data rekam jejak parkir dapat dicadangkan dan tidak hilang saat terjadi gangguan listrik.

2.5 Memori Non-Volatile Storage (NVS) dan Pustaka Preferences

Pada arsitektur mikrokontroler ESP32, penyimpanan data yang bersifat persisten (permanen) dikelola melalui partisi *Non-Volatile Storage* (NVS) yang terintegrasi di dalam ruang memori *Flash* internal. Berbeda dengan memori SRAM yang bersifat *volatile*, data *state* komputasi yang dituliskan ke dalam NVS akan tetap tersimpan dengan aman meskipun mikrokontroler terputus secara mendadak dari sumber daya listrik (*abrupt power interruptions*) [32].

Dalam lingkungan pengembangan *framework* Arduino untuk ESP32, manipulasi data pada partisi NVS tidak lagi disarankan menggunakan pustaka EEPROM konvensional, melainkan dialihkan menggunakan pustaka Preferences.h. Pustaka *Preferences* menawarkan manajemen alokasi memori yang lebih mutakhir melalui mekanisme *wear-leveling*. Mekanisme ini secara otomatis mendistribusikan siklus operasi baca/tulis secara merata pada berbagai sektor *Flash* memori, yang secara krusial memperpanjang usia pakai mikrokontroler dan mencegah kerusakan prematur blok memori akibat penulisan data yang terkonsentrasi pada sektor yang sama (*write fatigue*).

Secara struktural, pustaka *Preferences* tidak menggunakan pendekatan pemetaan alamat memori fisik (*byte addressing*) secara manual, melainkan mengadopsi sistem penyimpanan berbasis pasangan kunci-nilai (*Key-Value Pair*). Metode ini memanfaatkan label *string* (kunci) sebagai penanda identitas unik untuk

mengeksekusi operasi data. Pustaka ini mendukung pengelolaan berbagai tipe variabel, mulai dari tipe primitif seperti bilangan bulat (`putInt()`, `getInt()`) untuk menyimpan parameter posisi slot, hingga struktur data majemuk berupa susunan *byte* memori menggunakan fungsi `putBytes()` dan `getBytes()`.

Pada implementasi kendali sistem *smart parking*, memori NVS dan SRAM dirancang untuk beroperasi secara komplementer. Struktur *array* pada memori SRAM berfungsi sebagai area kerja utama (*working space*) untuk mengeksekusi logika program dan validasi kartu secara *real-time* dengan latensi komputasi minimal. Sementara itu, partisi NVS yang diakses melalui pustaka *Preferences* difungsikan secara eksklusif sebagai repositori pencadangan (*persistent flash storage*). Setiap kali alat mendeteksi perubahan status ruang parkir (kendaraan masuk atau keluar), sistem akan menyalin keseluruhan nilai dari *memory array* di SRAM dan merekamnya seketika ke dalam NVS. Melalui arsitektur ini, mikrokontroler menjamin terjadinya pemulihan konteks data secara menyeluruh (*complete context restoration*), di mana posisi mekanik motor dan rekam jejak slot parkir dapat dimuat ulang dari *Flash* ke RAM segera setelah sistem pulih dari insiden mati listrik [32].

2.6 Formulasi Matematika Pengujian dan Analisis Statistik

Formulasi matematika dan analisis statistik digunakan sebagai landasan teoritis untuk menguji serta mengevaluasi kinerja sistem, mencakup akurasi pembacaan sensor RFID RC522, kepastian sudut pergerakan motor servo SG90, hingga latensi transmisi data pada komunikasi serial UART dan protokol MQTT.

2.6.1 Persentase Kesalahan (Percentage Error)

Persentase *error* digunakan untuk mengukur tingkat penyimpangan atau kesalahan antara nilai hasil pengujian fisik alat terhadap nilai acuan baku.

$$\text{Error (\%)} = \left(\frac{|x_{\text{acuan}} - x_i|}{x_{\text{acuan}}} \right) \times 100\% \quad (2.1)$$

2.6.2 Nilai Rata-rata (Mean)

Nilai rata-rata ($\bar{x}$) digunakan untuk menentukan nilai tengah dari sekumpulan data sampel pengujian.

$$\bar{x} = \frac{\sum_{i=1}^N x_i}{N} \quad (2.2)$$

2.6.3 Standar Deviasi (Standar Deviation – SD)

Standar deviasi (SD) mengukur tingkat sebaran data pengujian terhadap nilai rata-ratanya. Semakin kecil nilai SD , semakin tinggi tingkat konsistensi dan presisi pengukuran sistem.

$$SD = \sqrt{\frac{\sum_{i=1}^N (x_i - \bar{x})^2}{N - 1}} \quad (2.3)$$

2.6.5 Koefisien Variasi (Coefficient of Variation – CV)

Koefisien variasi (CV) digunakan untuk mengukur tingkat presisi relatif dari data pengujian yang dinyatakan dalam bentuk persentase.

$$CV = \left(\frac{SD}{\bar{x}} \right) \times 100\% \quad (2.4)$$

2.6.6 Rentan Relatif (Relative Range – RR)

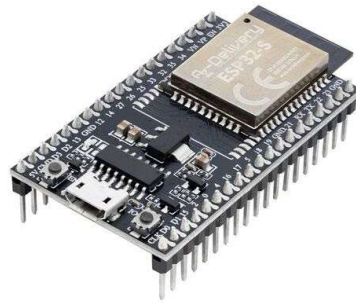
Relative Range (RR) digunakan untuk mengevaluasi rentang sebaran antara data maksimum dan minimum relatif terhadap nilai rata-rata pengujian.

$$RR = \left(\frac{x_{\max} - x_{\min}}{\bar{x}} \right) \times 100\% \quad (2.5)$$

2.7 Mikrokontroler ESP32

ESP32 merupakan mikrokontroler berbasis *System on Chip* (*SoC*) yang dikembangkan oleh *Espressif Systems* dan banyak digunakan dalam pengembangan sistem embedded serta *Internet of Things* (IoT). Modul ini dilengkapi dengan prosesor dual-core Tensilica Xtensa LX6 yang mampu bekerja hingga frekuensi 240

MHz, sehingga memiliki kemampuan pemrosesan data yang relatif tinggi untuk kelas mikrokontroler. Selain itu, ESP32 telah terintegrasi dengan modul WiFi dan Bluetooth, memori SRAM internal, serta berbagai antarmuka komunikasi dalam satu chip, sehingga memudahkan pengembangan sistem yang membutuhkan konektivitas dan pengolahan data secara simultan [12].



Gambar 2. 3 Mikrokontroller ESP32 [36]

Adapun spesifikasi dari ESP32 yang digunakan adalah sebagai berikut:

Tabel 2. 2 Spesifikasi ESP32

Parameter	Spesifikasi
Tipe Modul	ESP32 DevKit V1 & DevKit V4 (ESP32-WROOM-32)
Prosesor	Dual-core Tensilica LX6
Clock Speed	Hingga 240 MHz
Tegangan Operasi	3.3V
Tegangan <i>Input</i> (VIN)	5V–12V
Jumlah GPIO	±30 Pin
Koneksi Nirkabel	WiFi 802.11 b/g/n, Bluetooth 4.2
ADC	12-bit
Fitur Tambahan	PWM, SPI, I2C, UART

Dibandingkan dengan mikrokontroler generasi sebelumnya seperti Arduino Uno, ESP32 menawarkan keunggulan dalam hal jumlah GPIO, kecepatan pemrosesan, serta dukungan multitasking. ESP32 mendukung berbagai protokol komunikasi seperti UART, SPI, I2C, PWM, ADC, dan DAC yang memungkinkan integrasi dengan berbagai sensor maupun aktuator. Fleksibilitas ini menjadikan ESP32 banyak digunakan dalam penelitian dan pengembangan sistem kendali berbasis mikrokontroler, termasuk sistem otomasi dan *monitoring* berbasis IoT. Beberapa penelitian juga menunjukkan bahwa ESP32 memiliki performa yang stabil dan efisien dalam pengendalian perangkat keras secara *real-time* dengan konsumsi daya yang relatif rendah [13].

Pada tugas akhir ini, sistem menggunakan dua unit ESP32 yang bekerja dengan konfigurasi *master-slave*. ESP32 yang berperan sebagai *master* bertugas membaca data dari modul RFID RC522 yang digunakan sebagai media identifikasi pengguna, kemudian melakukan proses identifikasi dan pencatatan data UID (*Unique Identifier*) pada memori sistem untuk menentukan proses masuk atau keluar kendaraan. Selain itu, *master* juga menampilkan informasi sistem melalui LCD dan OLED sebagai media tampilan lokal, serta mengirimkan data ke aplikasi berbasis MIT App Inventor melalui protokol komunikasi MQTT sebagai bagian dari sistem *Human Machine Interface* (HMI). Adapun pengendalian aktuator *gate* berupa motor servo dilakukan oleh ESP32 yang berperan sebagai *slave*. *Master* dan *slave* saling terhubung melalui komunikasi serial (UART), sehingga ketika proses identifikasi selesai, *master* akan mengirimkan perintah kepada *slave* untuk menggerakkan motor servo dalam membuka atau menutup palang parkir secara otomatis. Dengan kemampuan pemrosesan yang cepat serta dukungan konektivitas jaringan, ESP32 mampu menjalankan logika sistem secara terstruktur dan *real-time* sehingga proses identifikasi, kontrol *gate*, dan *monitoring* sistem dapat berjalan secara sinkron dan efisien. Adapun perincian pembagian tugas serta konfigurasi pin dari kedua ESP32 dijelaskan lebih lanjut pada Bab III.

2.8 Radio Frequency Identification (RFID)

RFID (*Radio Frequency Identification*) merupakan teknologi identifikasi otomatis yang menggunakan gelombang radio untuk membaca dan mengidentifikasi data yang tersimpan pada sebuah tag atau kartu RFID tanpa kontak fisik secara langsung. Pada sistem *Smart Rotary Parking* yang dirancang, RFID digunakan sebagai media identifikasi pengguna untuk mengakses sistem parkir. Setiap pengguna memiliki kartu RFID dengan kode identitas unik (UID) yang akan dibaca oleh modul RFID *reader*. Apabila kartu terdaftar pada sistem, maka mikrokontroler ESP32 akan memberikan perintah untuk mengaktifkan mekanisme sistem parkir seperti membuka akses atau mengoperasikan mekanisme *rotary parking*. Penggunaan RFID dalam sistem parkir dapat meningkatkan efisiensi, keamanan, serta mempercepat proses identifikasi pengguna dibandingkan dengan sistem manual [14].

Secara umum, sebuah sistem RFID terdiri atas dua komponen utama, yaitu pembaca (*reader*) dan penanda (*tag* atau *transponder*). Jika *reader* berfungsi memancarkan gelombang radio dan membaca data, maka *tag* merupakan bagian yang menyimpan data identitas dan dibawa oleh pengguna, umumnya berbentuk kartu atau gantungan kunci. Sebuah *tag* RFID pasif tersusun atas dua bagian utama, yaitu sebuah keping *microchip* (IC) yang menyimpan kode identitas unik (UID) dan sebuah antena berupa kumparan. *Tag* jenis ini tidak memiliki sumber daya sendiri, melainkan memperoleh energi dari medan elektromagnetik yang dipancarkan *reader* melalui prinsip induksi. Ketika *tag* berada dalam jangkauan medan tersebut, rangkaian di dalamnya akan aktif dan mengirimkan kembali data UID yang tersimpan menuju *reader*.

Berdasarkan sumber dayanya, *tag* RFID dibedakan menjadi tiga jenis, yaitu *tag* pasif (tanpa baterai), *tag* aktif (memiliki baterai internal), serta *tag* semi-pasif. Pada sistem *Smart Rotary Parking* yang dirancang, digunakan *tag* pasif berjenis kartu *MIFARE Classic 1K* yang bekerja pada frekuensi 13.56 MHz. Kartu ini memiliki UID yang bersifat permanen dan unik untuk setiap kartu, sehingga dapat digunakan sebagai penanda identitas masing-masing pengguna. Karena bersifat

pasif, kartu tidak memerlukan baterai, berukuran ringkas, tahan lama, serta berbiaya rendah, sehingga sesuai untuk diterapkan pada sistem kontrol akses parkir.

Pada penelitian ini digunakan modul RFID RC522 (MFRC522) yang umum digunakan pada sistem embedded dan *Internet of Things* (IoT). Modul ini bekerja pada frekuensi 13.56 MHz dan dapat berkomunikasi dengan mikrokontroler menggunakan protokol SPI (Serial Peripheral Interface).



Gambar 2. 4 RFID RC522 [37]

Adapun spesifikasi dari RFID RC522 yang digunakan adalah sebagai berikut:

Tabel 2. 3 Spesifikasi RFID RC522

Parameter	Spesifikasi
Tipe Modul	MFRC522
Frekuensi Kerja	13.56 MHz
Tegangan Operasi	3.3 V
Antarmuka Komunikasi	SPI
Jarak Pembacaan	$\pm 3 - 5$ cm
Arus Operasi	$\pm 13 - 26$ mA
Jenis Tag	MIFARE 1K / RFID Card

Modul RFID RC522 banyak digunakan pada sistem kontrol akses karena memiliki ukuran yang kecil, konsumsi daya rendah, serta mampu membaca UID kartu dengan cepat dan akurat [15].

Cara kerja RFID pada sistem *Smart Rotary Parking* dimulai ketika pengguna mendekatkan kartu RFID ke modul *reader*. *Reader* kemudian memancarkan gelombang radio untuk mengaktifkan tag RFID yang berada pada kartu. Setelah aktif, tag akan mengirimkan data identitas unik (UID) kembali ke *reader*. Data UID tersebut kemudian dikirimkan ke mikrokontroler ESP32 melalui komunikasi SPI. ESP32 akan memproses data tersebut dan membandingkannya dengan *database* kartu yang telah terdaftar pada sistem. Jika kartu valid, maka ESP32 akan memberikan perintah untuk mengaktifkan sistem parkir seperti menggerakkan motor atau membuka akses parkir. Sebaliknya, jika kartu tidak terdaftar, maka sistem akan menolak akses dan tidak mengaktifkan mekanisme parkir. Proses identifikasi ini berlangsung sangat cepat sehingga dapat meningkatkan efisiensi sistem parkir otomatis berbasis IoT [16].

2.9 Motor Servo

Motor servo merupakan aktuator elektromekanik yang digunakan untuk menghasilkan pergerakan sudut yang presisi dan dapat dikontrol secara akurat melalui sinyal kendali dari mikrokontroler. Motor ini banyak digunakan dalam sistem otomasi, robotika, serta berbagai aplikasi kontrol posisi yang membutuhkan ketepatan gerak. Pada sistem *Smart Rotary Parking* yang dirancang, motor servo digunakan sebagai aktuator untuk menggerakkan mekanisme tertentu pada sistem parkir, seperti membuka atau menutup pengunci mekanik ataupun mekanisme akses pada bagian sistem parkir otomatis.

Penggunaan motor servo pada sistem ini dipilih karena memiliki kemampuan pengaturan sudut yang presisi, respon yang cepat, serta mudah dikontrol menggunakan sinyal *Pulse Width Modulation* (PWM) dari mikrokontroler ESP32. Dengan adanya motor servo, sistem parkir dapat bekerja secara otomatis ketika pengguna yang teridentifikasi melalui RFID melakukan akses terhadap

sistem. Implementasi motor servo dalam sistem otomatis terbukti mampu meningkatkan akurasi kontrol posisi serta efisiensi kerja sistem berbasis mikrokontroler [17].

Pada penelitian ini digunakan motor servo tipe SG90, yang merupakan salah satu jenis motor servo mikro yang umum digunakan pada sistem embedded dan proyek berbasis mikrokontroler.



Gambar 2. 5 Motor Servo SG90 [38]

Adapun spesifikasi dari Motor servo SG90 yang digunakan adalah sebagai berikut:

Tabel 2. 4 Spesifikasi Motor Servo SG90

Parameter	Spesifikasi
Tipe Modul	SG90
Tegangan Operasi	4.8 – 6 V DC
Torsi Maksimum	± 1.8 kg/cm (pada 4.8 V)
Kecepatan Operasi	± 0.1 s/60°
Sudut Rotasi	0° – 180°
Jenia Kontrol	PWM (<i>Pulse Width Modulation</i>)

Motor servo SG90 dipilih karena memiliki ukuran yang kecil, konsumsi daya rendah, serta mampu memberikan kontrol sudut yang cukup presisi untuk berbagai sistem otomasi berbasis mikrokontroler [16].

Pengendalian motor servo dilakukan menggunakan sinyal *Pulse Width Modulation* (PWM). PWM merupakan sebuah teknik pengaturan sinyal dengan cara mengubah lebar pulsa (*pulse width*) pada suatu gelombang kotak yang memiliki frekuensi tetap. Melalui teknik ini, sinyal digital yang hanya bernilai tinggi (*high*) dan rendah (*low*) dapat dimanfaatkan untuk menghasilkan keluaran yang seolah-olah bersifat analog. Parameter utama pada PWM adalah *duty cycle*, yaitu perbandingan antara lamanya sinyal berada pada kondisi *high* (*tON*) terhadap satu periode penuh gelombang (*ttotal*), yang biasanya dinyatakan dalam persen. Semakin besar *duty cycle*, semakin besar pula nilai rata-rata keluaran yang dihasilkan.

Pada pengendalian motor servo, sinyal PWM yang digunakan memiliki periode tetap, umumnya sebesar 20 ms (setara frekuensi 50 Hz). Nilai sudut motor servo ditentukan oleh lebar pulsa *high* dalam setiap periode tersebut, bukan oleh besarnya tegangan. Dengan demikian, mikrokontroler ESP32 cukup mengubah lebar pulsa PWM yang dihasilkan pada pin keluaran untuk menggerakkan motor servo ke sudut tertentu secara presisi. Kemampuan ESP32 dalam menghasilkan sinyal PWM secara digital inilah yang dimanfaatkan untuk mengontrol aktuator *gate* pada sistem parkir.

Cara kerja motor servo pada sistem *Smart Rotary Parking* dimulai ketika mikrokontroler ESP32 menerima data identitas dari modul RFID. Setelah proses identifikasi selesai, ESP32 akan mengirimkan sinyal PWM (*Pulse Width Modulation*) ke motor servo melalui pin kontrol. Lebar pulsa PWM yang diberikan oleh mikrokontroler akan menentukan sudut pergerakan motor servo. Sebagai contoh, pulsa sekitar 1 ms akan menghasilkan sudut sekitar 0°, pulsa 1.5 ms menghasilkan sudut sekitar 90°, dan pulsa sekitar 2 ms menghasilkan sudut sekitar 180°. Berdasarkan sinyal tersebut, motor servo akan bergerak ke posisi sudut tertentu untuk mengaktifkan mekanisme sistem parkir, seperti membuka akses atau menggerakkan komponen mekanik pada sistem.

Setelah proses akses selesai, mikrokontroler dapat kembali mengirimkan sinyal PWM untuk mengembalikan posisi motor servo ke posisi awal. Dengan

mekanisme tersebut, motor servo dapat berfungsi sebagai aktuator yang mengontrol pergerakan sistem secara otomatis dan terintegrasi dengan sistem identifikasi RFID serta kontrol utama dari ESP32 [17].

2.10 Power Supply

Sistem catu daya merupakan bagian penting dalam perancangan sistem *smart rotary parking* karena berfungsi sebagai penyedia energi listrik untuk seluruh komponen, mulai dari mikrokontroler, sensor, modul kamera, hingga aktuator motor DC. Stabilitas tegangan dan arus yang diberikan sangat mempengaruhi kinerja serta keandalan sistem secara keseluruhan. Gangguan pada catu daya dapat menyebabkan sistem *reset*, kesalahan pembacaan sensor, atau bahkan kerusakan komponen elektronik [18].



Gambar 2. 6 Power Supply 12V 10V [39]

Adapun spesifikasi dari Power supply 12V 10A yang digunakan adalah sebagai berikut

Tabel 2. 5 Spesifikasi Power Supply

Parameter	Spesifikasi
Tegangan <i>Output</i>	12V DC
Arus <i>Output</i>	10A
Regulator Tambahan	Step-down 5V
Jenis	Switching Power Supply

Pada sistem ini digunakan sumber tegangan utama yang disesuaikan dengan kebutuhan motor DC, umumnya berada pada rentang 9–12V. Tegangan tersebut kemudian diregulasi menggunakan modul *step-down* (buck converter) untuk menghasilkan tegangan 5V dan 3.3V yang dibutuhkan oleh ESP32, Motor Servo, serta LCD I2C. Regulasi tegangan diperlukan karena ESP32 bekerja pada logika 3.3V dan tidak toleran terhadap tegangan 5V secara langsung pada pin *input/output*. Oleh karena itu, pemisahan jalur catu daya antara rangkaian logika dan rangkaian motor menjadi aspek penting untuk mencegah gangguan noise akibat beban induktif motor.

2.11 Liquid Crystal Display (LCD)

Liquid Crystal Display (LCD) merupakan perangkat display elektronik yang digunakan untuk menampilkan informasi dalam bentuk teks maupun karakter. LCD banyak digunakan pada sistem berbasis mikrokontroler sebagai media untuk menampilkan data hasil pemrosesan sistem secara langsung kepada pengguna. Keunggulan LCD antara lain memiliki konsumsi daya yang rendah, ukuran yang relatif kecil, serta mudah diintegrasikan dengan berbagai jenis mikrokontroler [19].

Pada sistem *Smart Rotary Parking* yang dirancang, LCD digunakan sebagai media tampilan utama untuk menampilkan informasi operasional sistem. Informasi yang ditampilkan meliputi status pembacaan kartu RFID, proses akses kendaraan, serta kondisi sistem parkir. Dengan adanya LCD, pengguna dapat mengetahui kondisi sistem secara langsung ketika berinteraksi dengan perangkat parkir.

Penggunaan LCD pada sistem berbasis mikrokontroler seperti ESP32 memungkinkan informasi sistem dapat ditampilkan secara *real-time* sehingga memudahkan proses *monitoring* dan interaksi antara pengguna dengan perangkat otomasi.

Pada penelitian ini digunakan LCD 16×2 yang merupakan salah satu jenis display karakter yang umum digunakan pada sistem embedded. LCD jenis ini mampu menampilkan 16 karakter pada setiap baris dan memiliki 2 baris tampilan.



Gambar 2. 7 Liquid Crystal Display I2C [40]

Adapun spesifikasi dari *Liquid Crystal Display I2C* yang digunakan adalah sebagai berikut

Tabel 2.6 Spesifikasi LCD 16x2 I2C

Parameter	Spesifikasi
Jenis Display	LCD
Jumlah Baris	2
Jumlah Karakter	16
Tegangan Operasi	5 V DC
Interface Komunikasi	I2C
Jalur Komunikasi	SDA dan SCL

Penggunaan modul I2C pada LCD bertujuan untuk mengurangi jumlah pin yang digunakan pada mikrokontroler. Dengan menggunakan modul ini, komunikasi antara LCD dan mikrokontroler hanya memerlukan dua jalur komunikasi yaitu Serial Data (SDA) dan Serial Clock (SCL) sehingga lebih efisien dalam integrasi sistem [20].

Cara kerja LCD pada sistem *Smart Rotary Parking* dimulai ketika mikrokontroler ESP32 menerima data dari berbagai perangkat *input* seperti modul RFID, sensor inframerah, maupun tombol kontrol. Data tersebut kemudian diproses

oleh ESP32 untuk menentukan kondisi operasional sistem parkir. Setelah proses pengolahan data selesai, ESP32 akan mengirimkan informasi status sistem ke LCD melalui komunikasi I2C. Data yang dikirimkan berupa karakter teks yang kemudian ditampilkan pada layar LCD. Informasi yang ditampilkan dapat berupa pesan seperti proses pembacaan kartu RFID, status akses pengguna, serta informasi operasional sistem parkir. Selain ditampilkan pada LCD, informasi tersebut juga dapat dikirimkan ke aplikasi *monitoring* berbasis MIT App Inventor sehingga tampilan pada aplikasi dapat berfungsi sebagai *mirror display* dari informasi yang ditampilkan pada LCD. Dengan mekanisme ini, pengguna dapat memantau kondisi sistem baik secara langsung melalui perangkat LCD maupun melalui aplikasi pada *smartphone*.

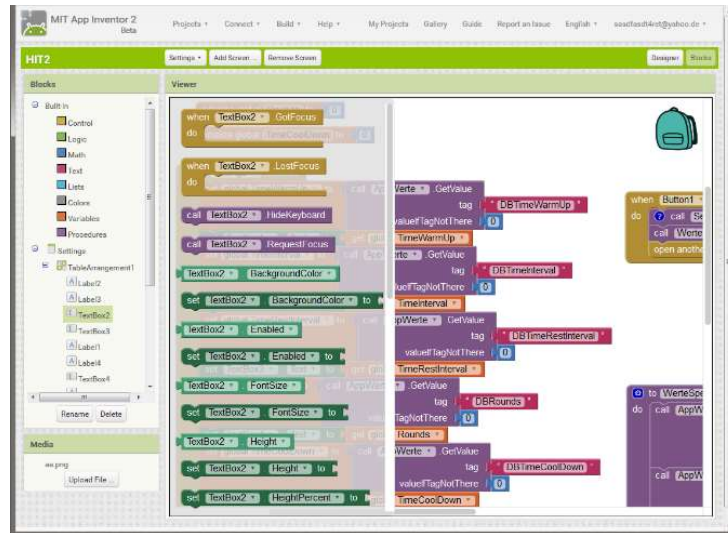
2.12 MIT App Inventor

MIT App Inventor merupakan platform pengembangan aplikasi berbasis visual yang memungkinkan pengguna untuk membuat aplikasi Android menggunakan metode pemrograman berbasis blok (*block programming*). Platform ini dikembangkan untuk mempermudah pembuatan aplikasi mobile tanpa memerlukan penulisan kode program yang kompleks. MIT App Inventor banyak digunakan dalam pengembangan sistem berbasis *Internet of Things* (IoT) sebagai media antarmuka antara pengguna dengan perangkat keras [21].

Pada sistem *Smart Rotary Parking* yang dirancang, aplikasi berbasis MIT App Inventor digunakan sebagai *Human Machine Interface* (HMI) untuk melakukan *monitoring* sistem parkir secara jarak jauh. Aplikasi ini berfungsi untuk menampilkan informasi yang sama dengan yang ditampilkan pada LCD, seperti status pembacaan RFID, kondisi sistem, serta informasi slot parkir. Dengan demikian, aplikasi ini berperan sebagai *mirror display* dari sistem utama sehingga pengguna dapat memantau kondisi sistem melalui *smartphone*.

Penggunaan aplikasi berbasis mobile dalam sistem IoT memberikan keunggulan dalam hal fleksibilitas dan kemudahan akses, karena pengguna dapat

memonitor sistem kapan saja dan di mana saja selama terhubung dengan jaringan [22].



Gambar 2. 8 Program MIT App Inventor

MIT App Inventor memiliki beberapa karakteristik yang mendukung penggunaannya sebagai HMI pada sistem berbasis IoT, antara lain:

Tabel 2.7 Karakteristik mit app inventor

No	Karakteristik
1	Platform berbasis visual (<i>block programming</i>)
2	Mendukung pembuatan aplikasi Android
3	Dapat terhubung dengan mikrokontroler melalui jaringan (WiFi/Internet)
4	Mendukung komponen antarmuka seperti tombol, label, dan indikator
5	Mudah digunakan untuk integrasi dengan sistem IoT seperti ESP32

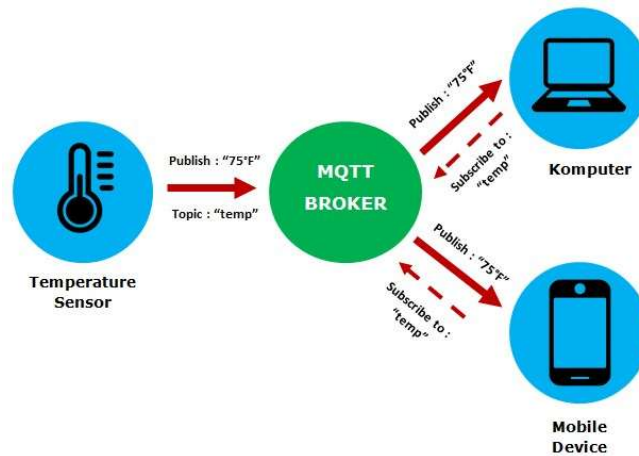
Dalam penelitian ini, aplikasi yang dibuat menggunakan MIT App Inventor berfungsi sebagai media *monitoring* sistem yang menampilkan data dari ESP32 secara *real-time*. Komunikasi antara aplikasi dan ESP32 tidak dilakukan secara langsung, melainkan melalui sebuah perantara berupa *broker* MQTT (Message Queuing Telemetry Transport). ESP32 dan aplikasi sama-sama terhubung ke *broker* melalui jaringan WiFi, sehingga pertukaran data sistem berlangsung melalui mekanisme *publish* dan *subscribe* pada *broker* tersebut.

Cara kerja aplikasi HMI berbasis MIT App Inventor pada sistem *Smart Rotary Parking* dimulai ketika mikrokontroler ESP32 menerima dan memproses data dari berbagai *input* seperti RFID, sensor, dan tombol kontrol. Data hasil pemrosesan tersebut kemudian tidak hanya ditampilkan pada LCD, tetapi juga dikirimkan melalui jaringan WiFi. ESP32 berperan sebagai *publisher* yang menerbitkan (*publish*) informasi sistem ke sebuah topik pada *broker* MQTT. Aplikasi yang dibuat menggunakan MIT App Inventor berperan sebagai *subscriber* yang berlangganan (*subscribe*) pada topik yang sama, sehingga setiap data yang diterbitkan ESP32 akan diteruskan oleh *broker* ke aplikasi dan ditampilkan dalam bentuk teks atau indikator pada layar *smartphone*.

Dengan mekanisme ini, aplikasi mobile akan menampilkan informasi yang sama dengan LCD sehingga berfungsi sebagai *mirror display* dari sistem. Pengguna dapat memantau status sistem parkir seperti akses RFID, proses pencarian slot, maupun kondisi sistem lainnya secara *real-time* melalui *smartphone*. Integrasi aplikasi mobile dengan sistem berbasis mikrokontroler seperti ESP32 merupakan salah satu implementasi konsep IoT yang memungkinkan *monitoring* sistem dilakukan secara lebih efektif dan fleksibel [23].

2.13 Protokol Komunikasi Message Queuing Telemetry Transport (MQTT)

Message Queuing Telemetry Transport (MQTT) merupakan protokol komunikasi ringan yang dirancang khusus untuk pertukaran data pada sistem *Internet of Things* (IoT) dengan kebutuhan *bandwidth* yang rendah dan sumber daya yang terbatas. MQTT bekerja menggunakan model komunikasi *publish-subscribe*, yang berbeda dengan model *request-response* pada komunikasi konvensional. Model ini memungkinkan pertukaran data berlangsung secara efisien dan *real-time* antara banyak perangkat sekaligus [25].



Gambar 2. 9 Arsitektur komunikasi MQTT [41]

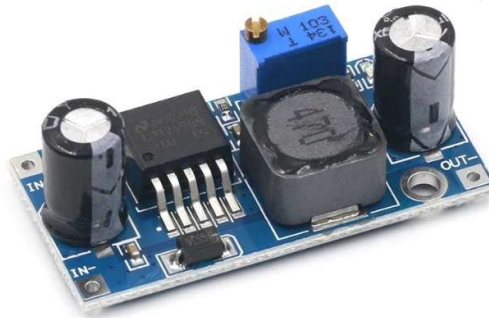
Dalam arsitektur MQTT terdapat tiga komponen utama, yaitu *publisher*, *subscriber*, dan *broker*. *Publisher* adalah perangkat yang menerbitkan data ke sebuah *topic* tertentu, sedangkan *subscriber* adalah perangkat yang berlangganan pada *topic* tersebut untuk menerima data. Adapun *broker* berperan sebagai perantara (server pusat) yang menerima seluruh pesan dari *publisher* dan meneruskannya kepada seluruh *subscriber* yang berlangganan pada *topic* yang sesuai. Dengan mekanisme ini, *publisher* dan *subscriber* tidak perlu saling mengetahui alamat satu sama lain, sehingga sistem menjadi lebih fleksibel dan mudah dikembangkan [26].

Pada sistem *Smart Rotary Parking* yang dirancang, mikrokontroler ESP32 berperan sebagai *publisher* yang menerbitkan data status sistem, seperti hasil pembacaan RFID, status akses, serta jumlah slot parkir, ke sebuah *topic* pada *broker* MQTT. Aplikasi *Human Machine Interface* (HMI) berbasis MIT App Inventor berperan sebagai *subscriber* yang berlangganan pada *topic* yang sama, sehingga setiap perubahan data yang diterbitkan ESP32 akan langsung diterima dan ditampilkan oleh aplikasi secara *real-time*. *Broker* yang digunakan pada penelitian ini adalah *broker* MQTT publik HiveMQ yang dapat diakses melalui jaringan internet. Penggunaan protokol MQTT dipilih karena bersifat ringan, efisien, serta andal dalam mendukung komunikasi data dua arah pada sistem *monitoring* berbasis IoT [25].

2.14 Step-down Converter LM2596

Step-down converter LM2596 merupakan modul DC-DC buck converter yang berfungsi untuk menurunkan tegangan *input* menjadi tegangan *output* yang lebih rendah sesuai kebutuhan sistem. Modul ini banyak digunakan pada sistem berbasis mikrokontroler untuk menyesuaikan tegangan sumber daya agar sesuai dengan spesifikasi komponen yang digunakan.

Pada sistem *Smart Rotary Parking*, modul LM2596 digunakan untuk menurunkan tegangan dari power supply menjadi tegangan yang stabil sebesar 5V yang digunakan untuk mensuplai ESP32, LCD, dan motor servo. Penggunaan step-down converter ini bertujuan untuk menjaga kestabilan tegangan serta mencegah kerusakan komponen akibat tegangan yang tidak sesuai.



Gambar 2. 10 Step-Down Converter LM2596 [42]

Adapun spesifikasi umum dari modul LM2596 yang digunakan adalah sebagai berikut.

Tabel 2.8 Spesifikasi LM2596

Parameter	Spesifikasi
Tipe Modul	DC-DC Step-Down (Buck Converter)
Tegangan <i>Input</i>	4V – 40V DC
Tegangan <i>Output</i>	1.25V – 35V DC
Arus Maksimum	3A
Frekuensi Switching	150 kHz

Modul ini memiliki keunggulan dalam hal efisiensi yang tinggi serta kemampuan menghasilkan tegangan *output* yang stabil sehingga sangat cocok digunakan pada sistem embedded dan IoT [24].

Cara kerja modul LM2596 didasarkan pada prinsip buck converter, yaitu dengan menurunkan tegangan *input* melalui proses switching menggunakan komponen internal seperti transistor, dioda, induktor, dan kapasitor. Tegangan *output* diatur dengan memanfaatkan potensiometer yang terdapat pada modul.

Pada sistem yang dirancang, tegangan dari power supply terlebih dahulu masuk ke modul LM2596. Selanjutnya, tegangan tersebut diturunkan menjadi 5V dan disalurkan ke ESP32, LCD, serta motor servo. Dengan adanya modul ini, sistem mendapatkan suplai tegangan yang stabil sehingga kinerja komponen menjadi lebih optimal.

2.15 Organic Light Emitting Diode (OLED) SSD1306

Organic Light Emitting Diode (OLED) merupakan jenis layar penampil yang memanfaatkan bahan organik yang dapat memancarkan cahaya ketika dialiri arus listrik. Berbeda dengan LCD yang membutuhkan lampu latar (*backlight*), setiap piksel pada OLED menghasilkan cahayanya sendiri, sehingga layar OLED memiliki tingkat kontras yang tinggi, konsumsi daya yang rendah, serta ukuran yang ringkas. Pada penelitian ini digunakan modul OLED berukuran 0.96 inci dengan pengendali (*driver*) SSD1306 yang berkomunikasi dengan mikrokontroler melalui antarmuka I2C [27].

Pada sistem *Smart Rotary Parking* yang dirancang, OLED SSD1306 digunakan sebagai penampil khusus yang menunjukkan jumlah slot parkir yang tersedia dalam bentuk satu angka berukuran besar. Ketika seluruh slot telah terisi, layar akan menampilkan tulisan “FULL” sebagai penanda bahwa kapasitas parkir telah penuh. Modul OLED ini bersifat monokrom (satu warna), sehingga informasi ditampilkan dalam warna tunggal sesuai jenis modul yang digunakan. OLED terhubung pada jalur I2C yang sama dengan LCD, namun menggunakan alamat

yang berbeda sehingga keduanya dapat bekerja secara bersamaan tanpa saling mengganggu.

Penggunaan OLED sebagai penampil jumlah slot dipilih karena tampilannya yang jelas dan mudah dibaca dari jarak tertentu, sehingga melengkapi fungsi LCD yang menampilkan instruksi dan status akses.



Gambar 2. 11 OLED SSD1306 [43]

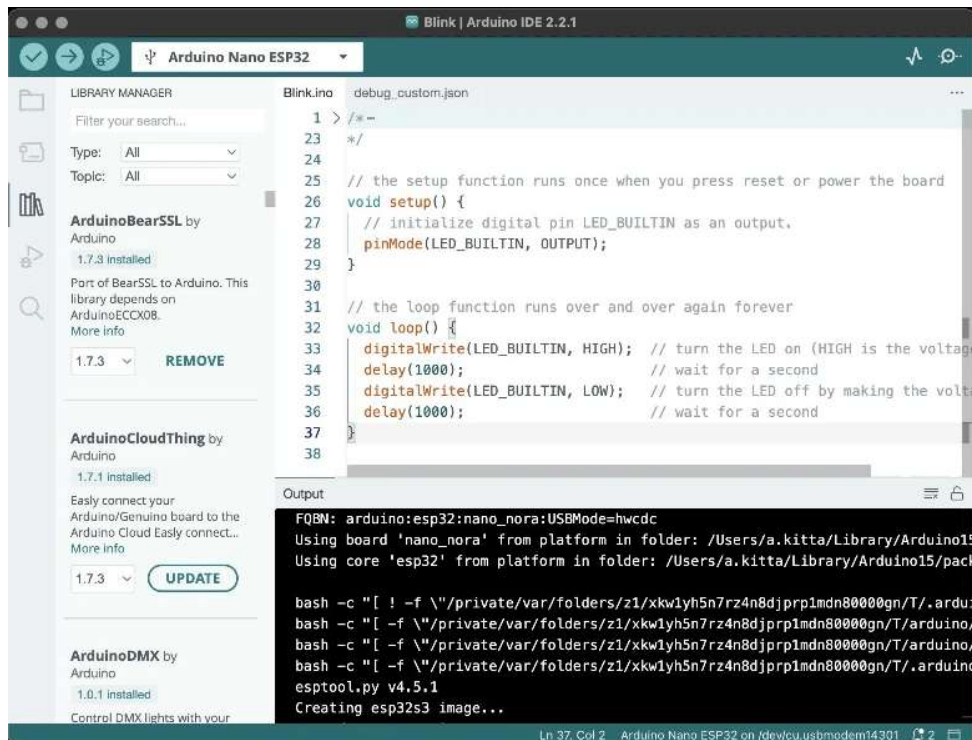
Adapun spesifikasi modul OLED SSD1306 yang digunakan ditunjukkan pada tabel 2.9 berikut:

Tabel 2.9 Spesifikasi OLED SSD1306

Parameter	Spesifikasi
Tipe Modul	SSD1306
Ukuran Layar	0.96 inci
Resolusi	128 x 64 piksel
Tegangan Operasi	5 V DC
Interface Komunikasi	I2C
Alamat I2C	0x3C

2.16 Arduino IDE

Arduino IDE merupakan perangkat lunak yang digunakan untuk membuat sketch pemrograman, atau dengan kata lain berfungsi sebagai media untuk menulis dan memuat program ke dalam board Arduino. Software ini memungkinkan pengguna untuk membuat, mengedit, menulis kode, serta mengunggah program ke papan pengendali yang ditentukan. Arduino IDE dikembangkan menggunakan bahasa pemrograman JAVA dan dilengkapi dengan pustaka C/C++ (wiring) yang mempermudah pengoperasian *input* dan *output*. Dengan menggunakan Arduino IDE, pengguna dapat memodelkan karakteristik dari parameter rangkaian analog maupun digital. Kemampuan yang dimiliki oleh Arduino IDE mencakup perancangan berbagai jenis rangkaian, pengujian rangkaian dengan kombinasi komponen yang berbeda, serta analisis karakteristik keseluruhan rangkaian, baik dalam bentuk analisis AC, DC, maupun transien [28].

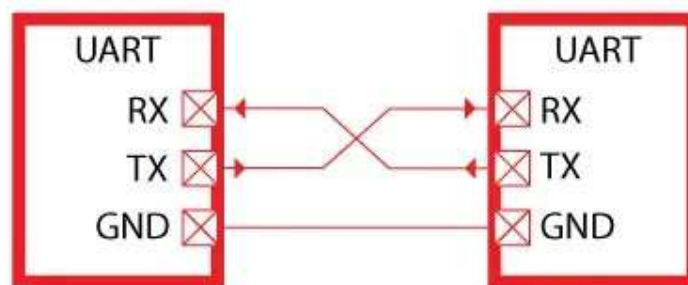


Gambar 2. 12 Dashboard Arduino IDE

2.17 Protokol Komunikasi UART

Universal Asynchronous Receiver-Transmitter atau UART merupakan protokol sekaligus modul perangkat keras komunikasi serial yang bertugas mengalirkan data biner bit demi bit secara berurutan antara dua perangkat elektronik. Berbeda dengan komunikasi paralel yang menggunakan banyak jalur data sekaligus, UART menyederhanakan koneksi fisik hingga hanya membutuhkan tiga jalur utama: TX (Transmit) untuk mengirim data, RX (Receive) untuk menerima data, dan GND (Ground) sebagai acuan tegangan bersama. Dalam praktiknya, jalur ini dihubungkan secara silang, di mana pin TX perangkat pertama tersambung ke pin RX perangkat kedua, dan sebaliknya.

Sifat asinkron pada UART menandakan bahwa transmisi data dilakukan tanpa mengandalkan sinyal *clock* bersama untuk menyelaraskan waktu pengiriman. Sebagai gantinya, pengirim dan penerima harus menyepakati kecepatan transfer data yang sama melalui pengaturan Baud Rate, seperti 9600 atau 115200 bit per detik. Agar perangkat penerima dapat mendeteksi awal dan akhir informasi tanpa sinyal *clock*, UART mengemas data ke dalam sebuah paket khusus (*frame*). Transmisi diawali oleh Start Bit yang menurunkan tegangan sinyal dari *HIGH* ke *LOW*, dilanjutkan dengan pengiriman 5 hingga 9 Data Bits, lalu Parity Bit opsional untuk mendeteksi kesalahan data, dan diakhiri oleh Stop Bit yang mengembalikan tegangan jalur ke kondisi *HIGH* [29].



Gambar 2. 13 Arsitektur komunikasi UART [44]