

BAB II

TINJAUAN PUSTAKA

2.1 Teori Terkait

Menurut penelitian yang dilakukan oleh Dita Parwansa, et.al (2025) yang berjudul “Rancang Bangun *Smart Parking System* Berbasis *Internet of Things* Pada Tempat Parkir di Mall”. Menyatakan bahwa Penelitian ini menggunakan metode *Research and Development* (R&D) dengan tahapan sistematis. Tahapan pertama yaitu identifikasi masalah melalui analisis kebutuhan dan penentuan ruang lingkup penelitian. Data kemudian dikumpulkan melalui studi literatur terhadap berbagai referensi terkait IOT dan smart parking serta observasi lapangan untuk memantau pola parkir aktual meliputi kepadatan, durasi, dan preferensi tempat parkir [5]. Berdasarkan data yang terkumpul, dilakukan perancangan sistem yang mencakup aspek *hardware* berupa desain diagram blok, skematik, dan *prototype* 3D, serta aspek *software* meliputi pengembangan algoritma sensor, desain database menggunakan ERD, dan perancangan antarmuka website. Tahap implementasi melibatkan integrasi berbagai komponen IOT seperti sensor dan mikrokontroler dengan sistem *backend*. Sistem yang telah dibangun kemudian diuji secara menyeluruh untuk mengevaluasi fungsionalitas sekaligus menganalisis kinerja sistem berdasarkan desain yang telah ditetapkan. Validasi akhir dilakukan untuk memastikan keandalan sistem secara keseluruhan sebelum diterapkan [6].

Dalam penelitian yang dilakukan oleh Arif Wihandanto, et.al (2021) yang berjudul "Rancang Bangun Prototipe Sistem *Smart Parking* Berbasis IOT Menggunakan Node Mcu Esp8266". pentingnya memberikan informasi terkait ketersediaan slot parkir kepada pengguna sebelum mereka memasuki area parkir. Sebagai respons terhadap tuntutan tersebut, solusi yang tengah diupayakan melibatkan pemantauan intensif terhadap setiap slot parkir di area tersebut. Dalam konteks penelitian tersebut, konsep IOT terintegrasi dalam sistem parkir tidak hanya memungkinkan pemantauan slot parkir secara lebih efisien, tetapi juga memberikan informasi secara *real-time* tentang status ketersediaan slot parkir kepada pengguna. Pada penelitian tersebut, penggabungan IOT dan sensor untuk memantau dan mengelola slot parkir diharapkan dapat memberikan solusi yang responsif dan adaptif sesuai kebutuhan pengguna [7].

Adapun landasan teori pada “ Rancang Bangun Sistem Pendeteksi Ketersediaan Slot Parkir Berbasis IOT Menggunakan ESP 32” berpacuan dengan berbagai penelitian terdahulu di antaranya :

2.2.1 Internet Of Things

Internet of Things (IOT) adalah konsep jaringan yang menghubungkan perangkat fisik dengan internet sehingga perangkat tersebut dapat saling berkomunikasi dan bertukar data. Dalam sistem monitoring parkir, IOT memungkinkan sensor dan mikrokontroler untuk mengirimkan informasi ketersediaan slot parkir secara *real-time* ke *platform* monitoring [8]. Dengan adanya konektivitas ini, proses pemantauan menjadi lebih efisien karena data dapat diakses kapan saja

dan dari mana saja, Selain itu, teknologi IOT juga dapat membantu mengurangi kemacetan, meningkatkan kenyamanan pengguna, dan mengoptimalkan penggunaan lahan parkir melalui analisis data yang dikumpulkan secara berkelanjutan.

2.2.2 Smart Parking System

Smart Parking System adalah solusi parkir berbasis teknologi modern seperti sensor, IOT, dan perangkat lunak untuk mengoptimalkan penggunaan lahan parkir dan memudahkan dalam menemukan tempat parkir secara *real-time*. sistem ini memberikan kemudahan bagi pengendara dalam memarkir kendaraan, sementara menekankan fungsinya dalam monitoring dan keamanan akses parkir. menyatakan bahwa *smart parking* menguntungkan pengguna dengan menghemat waktu, biaya, dan tenaga [9].

Pada sistem *smart parking*, berbagai sensor digunakan untuk mendeteksi keberadaan kendaraan pada area parkir. Informasi yang diperoleh dari sensor kemudian diproses oleh mikrokontroler atau sistem komputer dan dapat ditampilkan melalui layar monitor, aplikasi, atau perangkat lainnya sehingga pengguna dapat mengetahui kondisi ketersediaan tempat parkir secara langsung.

2.2.3 ESP 32

ESP32 adalah modul mikrokontroler dengan fitur WiFi dan Bluetooth 2.4 GHz berdaya rendah, digunakan dalam proyek *Internet of Things*. Modul ini berfungsi sebagai “otak sistem”. Esp 32 juga

merupakan penerus dari ESP8266, ESP32 memiliki banyak fitur tambahan dan keunggulan dibandingkan generasi sebelumnya. Pada ESP32 terdapat inti CPU serta Wi-Fi yang lebih cepat, GPIO yang lebih banyak, dan dukungan terhadap *Bluetooth* 4.2, serta konsumsi daya yang rendah, sehingga sangat cocok untuk membuat beberapa proyek-proyek elektronika berbasis *Internet of Things*. penggunaan sensor inframerah dengan mikrokontroler berbasis IOT efektif dalam sistem monitoring parkir, mampu meningkatkan akurasi deteksi, dan memudahkan pengelolaan area parkir [10].



Gambar 2. 1 ESP 32

2.2.4 Sensor Inframerah

Sensor inframerah adalah perangkat elektronik yang mendeteksi radiasi inframerah (IR) atau panas untuk berbagai tujuan seperti mendeteksi gerakan, mengukur jarak. Ada dua jenis utama sensor aktif yang memancarkan dan menerima pantulan sinar IR, serta sensor pasif yang mendeteksi panas yang dipancarkan oleh objek lain. Sensor Inframerah (IR) bekerja dengan *LED*-photodiode untuk mendeteksi objek, panas, dan gerakan melalui radiasi inframerah. Jenis sensor ini hanya mengukur radiasi pancaran.

Karena kemampuannya dalam mendeteksi keberadaan objek tanpa kontak langsung, sensor inframerah banyak digunakan pada berbagai sistem elektronik seperti sistem keamanan, robotika, alat penghitung objek, *remote control*, serta sistem otomatisasi parkir (*smart parking*). Pada sistem *smart parking*, sensor inframerah dapat digunakan untuk mendeteksi keberadaan slot parkir kendaraan serta pada jalur masuk atau keluar sehingga sistem dapat mengontrol perangkat lain seperti motor servo atau palang pintu secara otomatis [11].



Gambar 2. 2 Sensor Inframerah

2.2.5 Motor Servo

Motor servo merupakan perangkat listrik berpresisi tinggi yang digunakan untuk mengontrol posisi sudut, kecepatan, dan akselerasi pada berbagai sistem otomatisasi. Motor servo umumnya terdiri dari motor DC yang dilengkapi dengan rangkaian kontrol internal sehingga mampu memberikan kontrol gerakan yang lebih akurat dibandingkan dengan motor DC biasa. Motor servo bekerja menggunakan sistem umpan balik tertutup (*closed-loop control system*) yang memungkinkan posisi poros motor dapat dikontrol secara presisi sesuai dengan perintah yang diberikan [12].

sehingga menghasilkan energi dalam bentuk cahaya. Selain berfungsi sebagai penanda kondisi sistem, *LED* juga sering dimanfaatkan untuk menguji keandalan rangkaian, memverifikasi keluaran logika, serta membantu peneliti mengamati respon perangkat secara *real time*. Sifatnya yang efisien, memiliki konsumsi daya rendah, dan mudah diintegrasikan menjadikan *LED* komponen penting dalam berbagai eksperimen dan pengembangan *prototype* elektronik [13].



Gambar 2. 4 LED

2.2.7 Terminal Blok

Terminal blok adalah komponen konektor listrik yang digunakan untuk menghubungkan dua atau lebih kabel dalam suatu rangkaian secara aman dan terorganisir. Komponen ini biasanya terdiri dari beberapa titik sambungan (terminal) yang dilengkapi dengan sekrup atau penjepit untuk mengunci kabel, sehingga koneksi menjadi lebih kuat dan tidak mudah lepas. Dalam perancangan sistem elektronika, terminal blok sering digunakan untuk distribusi tegangan seperti VCC dan GND dari adaptor ke berbagai komponen (sensor, *LCD*, servo, dll). Penggunaan terminal blok sangat membantu dalam membuat jalur

paralel yang menyerupai fungsi *breadboard* dan *pcb*, namun dengan koneksi yang lebih permanen dan kuat [14].

Terminal blok sering juga digunakan dalam instalasi listrik, rangkaian elektronika, maupun sistem berbasis mikrokontroler, karena memudahkan proses pemasangan, pelepasan, serta perawatan kabel tanpa harus melakukan penyolderan langsung.



Gambar 2. 5 Terminal Blok

2.2.8 LCD

LCD (Liquid Crystal Display) merupakan salah satu perangkat elektronika yang digunakan sebagai media tampilan untuk menampilkan informasi. *LCD* bekerja dengan memanfaatkan kristal cair yang ditempatkan di antara dua lapisan kaca atau plastik transparan sehingga mampu menghasilkan tampilan visual. Tampilan yang dihasilkan dapat berupa titik, garis, simbol, huruf, angka, maupun gambar tertentu sesuai dengan data yang diterima dari sistem pengendali. Berdasarkan bentuk tampilannya, *LCD* dibedakan menjadi dua jenis, yaitu *Text LCD* dan *Graphic LCD*. *Text LCD* digunakan untuk menampilkan karakter berupa huruf dan angka, sedangkan

Graphic LCD mampu menampilkan tampilan yang lebih kompleks seperti titik, garis, dan gambar.

Salah satu jenis *LCD* yang sering digunakan dalam sistem mikrokontroler adalah LCD 16x2, yaitu *LCD* yang memiliki dua baris tampilan dengan masing-masing baris terdiri dari 16 karakter. *LCD* jenis ini banyak digunakan pada berbagai proyek elektronika karena mudah dihubungkan dengan mikrokontroler seperti Arduino atau ESP32. Selain itu, *LCD* juga memiliki beberapa keunggulan seperti ukuran yang relatif kecil, penggunaan daya yang hemat, serta kemampuan menampilkan informasi secara jelas [15].



Gambar 2. 6 LCD

2.2.9 Buzzer

Buzzer merupakan salah satu komponen elektronika yang berfungsi untuk mengubah sinyal listrik menjadi getaran suara. Komponen ini banyak digunakan pada berbagai perangkat elektronik, seperti sistem alarm, permainan elektronik, kendaraan, serta berbagai alat elektronik lainnya sebagai penanda atau pemberi peringatan. *Buzzer* termasuk dalam jenis transduser, yaitu perangkat yang mampu mengubah suatu bentuk energi menjadi bentuk energi lainnya. Pada *Buzzer*, energi listrik

diubah menjadi energi suara. Secara sederhana, *Buzzer* memiliki dua kaki atau terminal, yaitu terminal positif dan terminal negatif yang digunakan untuk menghubungkannya dengan sumber listrik atau rangkaian elektronik.

Buzzer dapat menghasilkan berbagai jenis suara atau nada, seperti nada tunggal, nada ganda, maupun suara berkelanjutan, tergantung pada jenis dan rangkaian yang digunakan. Beberapa jenis *Buzzer* juga dilengkapi dengan pengaturan volume atau frekuensi yang dapat disesuaikan dengan kebutuhan pengguna. Selain itu, *Buzzer* dapat dioperasikan menggunakan berbagai sumber daya listrik, seperti baterai, sumber daya *DC*, maupun *AC* [16].



Gambar 2. 7 Buzzer

2.2.10 Kabel Jumper

Kabel jumper merupakan jenis kabel yang digunakan untuk menghubungkan dua titik atau komponen dalam suatu rangkaian elektronik, terutama pada proses perakitan dan pembuatan *prototype*. Pada bagian ujung kabel jumper biasanya terdapat konektor yang disesuaikan dengan jenis port atau pin yang akan dihubungkan sehingga memudahkan proses pemasangan pada perangkat elektronik. Fungsi

utama kabel jumper adalah sebagai media penghubung sinyal atau sumber daya listrik antara satu komponen dengan komponen lainnya.

Pada umumnya kabel jumper memiliki panjang yang relatif pendek karena digunakan untuk menghubungkan komponen yang posisinya berdekatan dalam suatu rangkaian. Meskipun demikian, kabel jumper juga tersedia dalam berbagai ukuran panjang sehingga dapat disesuaikan dengan kebutuhan koneksi pada rangkaian elektronik [17].



Gambar 2. 8 Kabel Jumper

2.2.11 Arduino IDE

Arduino IDE 2.3.6 (*Integrated Development Environment*) adalah perangkat lunak (*software*) yang berfungsi sebagai lingkungan pemrograman terpadu untuk menulis, mengedit, mengompilasi, dan mengunggah program ke papan mikrokontroler Arduino. Perangkat ini menyediakan antarmuka yang sederhana namun lengkap sehingga memudahkan pengguna, baik pemula maupun tingkat lanjut, dalam mengembangkan berbagai proyek elektronika. Arduino IDE menggunakan bahasa pemrograman berbasis C/C++, dilengkapi dengan fitur seperti editor kode, *compiler*, *library manager*, serta serial monitor untuk menampilkan data dari papan Arduino [18].

Arduino IDE juga mendukung berbagai jenis papan mikrokontroler seperti Arduino Uno, Arduino Mega, ESP8266, dan ESP32 melalui penambahan *board manager*. Fitur ini memungkinkan pengguna untuk mengembangkan berbagai sistem berbasis mikrokontroler dan *Internet of Things* dengan lebih fleksibel. Dengan integrasi fungsi tersebut, *Arduino IDE* menjadi alat utama yang menghubungkan perangkat keras Arduino dengan program yang dibuat, sehingga proses perancangan, pengujian, dan pengembangan sistem berbasis mikrokontroler dapat dilakukan dengan lebih mudah, cepat, dan efisien [19].



Gambar 2. 9 Arduino IDE

2.2.12 Adaptor

Adaptor adalah perangkat elektronika yang berfungsi untuk mengubah tegangan listrik dari sumber AC (arus bolak-balik) menjadi tegangan DC (arus searah) yang dibutuhkan oleh perangkat elektronik. Proses ini sangat penting karena sebagian besar perangkat *modern*, terutama sistem berbasis mikrokontroler, tidak dapat bekerja langsung menggunakan arus AC dari jaringan listrik (PLN), melainkan memerlukan arus DC dengan tegangan yang lebih rendah dan stabil.

Secara umum, adaptor bekerja melalui beberapa tahapan utama, yaitu penurunan tegangan, penyearahan, penyaringan, dan penstabilan. Pada tahap awal, tegangan AC yang tinggi (misalnya 220V) diturunkan menggunakan transformator menjadi tegangan yang lebih rendah. Selanjutnya, arus tersebut disearahkan menggunakan rangkaian dioda (*rectifier*) sehingga berubah menjadi arus DC [20].

Adaptor biasanya digunakan untuk menyuplai daya pada berbagai perangkat seperti mikrokontroler (misalnya ESP32), sensor, modul IOT, serta rangkaian elektronika lainnya. Selain itu, adaptor juga memiliki spesifikasi tertentu yang perlu diperhatikan, seperti tegangan output (*Volt*), arus (*Ampere*), daya (*Watt*), serta polaritas. Pemilihan adaptor harus disesuaikan dengan kebutuhan perangkat agar tidak terjadi kekurangan daya (*underpower*) atau kelebihan daya (*overvoltage*) yang dapat merusak sistem.

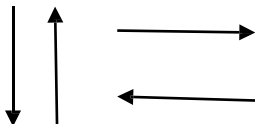
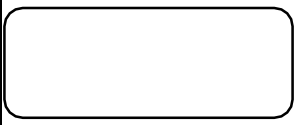
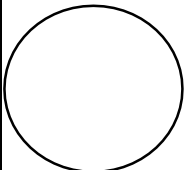
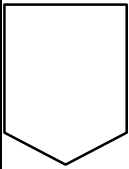
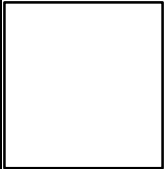


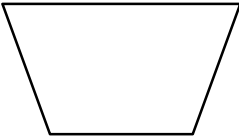
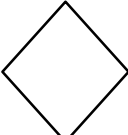
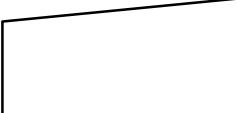
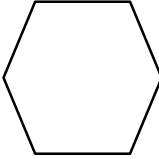
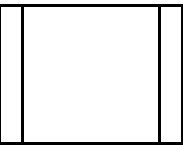
Gambar 2. 10 Adaptor

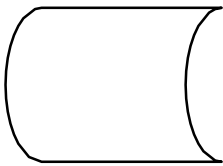
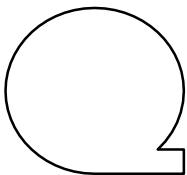
2.2.13 Flowchart

Flowchart merupakan diagram yang menggunakan simbol-simbol tertentu untuk menunjukkan urutan langkah secara rinci serta hubungan antarproses atau instruksi dalam suatu program. Dengan adanya *flowchart*, potensi masalah dalam suatu proses atau sistem dapat lebih mudah ditemukan, sekaligus membantu menentukan cara untuk meningkatkan efisiensi dan efektivitasnya. *Flowchart* juga sering dimanfaatkan sebagai dokumen pendukung untuk menggambarkan alur logis dari sistem yang akan dikembangkan sebelum diserahkan kepada programmer. Dalam pengembangan sistem, *flowchart* biasanya dibuat pada tahap perancangan (*design*) sebagai langkah awal sebelum penulisan kode program. *Flowchart* membantu dalam mengidentifikasi alur logika, kondisi percabangan, serta kemungkinan terjadinya pengulangan (*looping*) dalam suatu proses. Hal ini sangat penting untuk menghindari kesalahan logika yang dapat menyebabkan program tidak berjalan sesuai harapan. Dengan demikian, *flowchart* juga sebagai alat analisis dan perencanaan yang dapat meningkatkan kualitas sistem yang dikembangkan. Penggunaan *flowchart* yang baik akan menghasilkan sistem yang lebih terstruktur, mudah dipahami, serta lebih mudah dalam proses pengembangan dan pemeliharaan di masa mendatang. Dengan adanya *flowchart*, pengembang sistem dapat merencanakan logika program secara lebih terstruktur sebelum proses implementasi dilakukan [21].

Tabel 2. 1 *Flowchart*

No	Gambar	Keterangan
1.		<i>Flow Direction Symbol</i> Simbol yang digunakan untuk menghubungkan antara simbol satu dengan simbol yang lain.
2.		<i>Terminator Symbol</i> Simbol untuk permulaan (<i>start</i>) atau akhir (<i>stop</i>) dari suatu kegiatan.
3.		<i>Connector Symbol</i> Simbol untuk keluar-masuk atau penyambungan proses dalam lembar / halaman yang sama.
4.		<i>Connector Symbol</i> Simbol untuk keluar-masuk atau penyambungan proses dalam lembar / halaman yang berbeda.
5.		<i>Processing Symbol</i> Simbol yang menunjukkan pengelolaan yang dilakukan oleh komputer.

6.		Symbol Manual Operation Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh komputer.
7.		Symbol Decision Simbol pemilihan proses berdasarkan kondisi yang ada.
8.		Symbol Input-Output Simbol yang menyatakan proses <i>input</i> dan <i>output</i> tanpa tergantung dengan jenis peralatannya.
9.		Symbol Manual Input Simbol untuk memasukkan data secara manual <i>on-line keyboard</i> .
10.		Symbol Preparation Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam <i>storage</i> .
11.		Symbol Predifine Proces Simbol untuk pelaksanaan suatu bagian (<i>sub-program</i>) / <i>procedure</i> .
12.		Symbol Display Simbol yang menyatakan peralatan <i>output</i> yang digunakan yaitu layar, <i>plotter</i> , <i>printer</i> dan sebagainya.

13.		<i>Symbol Disk and On-line Storage</i> Simbol yang menyatakan input yang berasal dari <i>disk</i> atau disimpan ke <i>disk</i> .
14.		<i>Symbol Magnetic Tape Unit</i> Simbol yang menyatakan input berasal dari pita <i>magnetic</i> atau <i>output</i> disimpan ke pita <i>magnetic</i> .
15.		<i>Symbol Punch Card</i> Simbol yang menyatakan bahwa <i>input</i> berasal dari kartu atau <i>output</i> di tulis ke kartu.
16.		<i>Symbol Document</i> Simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau <i>output</i> dicetak ke kertas.