

BAB IV

PENGUJIAN DAN ANALISA

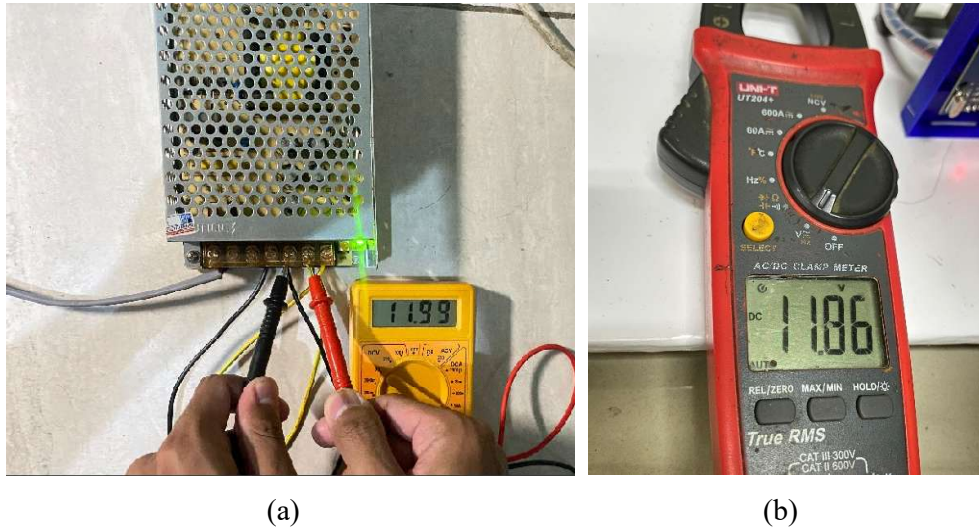
4.1 Pengujian Komponen

Pengujian komponen dilakukan untuk memastikan bahwa komponen yang digunakan dapat beroperasi secara baik dan akurat, sehingga dapat menghindari kesalahan komponen dalam sistem. Acuan yang digunakan dalam mengevaluasi hasil pengujian pada bab ini terbagi menjadi tiga kategori, disesuaikan dengan sifat parameter yang diuji. Pertama, untuk parameter elektris yang memiliki spesifikasi resmi dari pabrikan komponen, seperti tegangan keluaran modul *buck converter* LM2596 dan arus konsumsi modul RFID RC522, hasil pengujian dibandingkan langsung terhadap rentang nilai yang tercantum pada *datasheet* resmi. Kedua, untuk parameter yang tidak memiliki spesifikasi baku dari pabrikan, seperti karakteristik tegangan pada jalur komunikasi data (pin MISO) atau parameter latensi komunikasi jaringan publik, pengujian bersifat karakterisasi eksploratif yang dievaluasi berdasarkan konsistensi dan presisi statistik data itu sendiri, bukan terhadap acuan eksternal. Ketiga, untuk pengujian fungsional aktuator dan antarmuka (motor servo, LCD, OLED, serta logika pengambilan keputusan sistem), validasi dilakukan dengan membandingkan hasil aktual terhadap spesifikasi rancangan program (*expected output* berdasarkan kode sumber sistem), sebagai bentuk verifikasi kesesuaian implementasi terhadap desain, bukan validasi terhadap komponen fisik pihak ketiga.

4.1.1 Pengujian Power Supply 12V – 10A dan Buck Converter LM2596

Pengujian catu daya dilakukan dalam dua kondisi berbeda untuk mengevaluasi kestabilan regulasi tegangan, yaitu kondisi tanpa beban (sebelum terhubung ke rangkaian sistem utama) dan kondisi dengan beban (setelah terhubung penuh ke ESP32, motor servo, LCD, OLED, dan modul RFID). Pengujian turut dilakukan pada modul *buck converter* LM2596 untuk mengevaluasi keandalan regulasi penurunan tegangan dari 12V ke 5V. Setiap kondisi diukur sebanyak 5 kali

menggunakan multimeter digital pada titik keluaran masing-masing, sebagaimana ditunjukkan pada Gambar 4.1 dan Gambar 4.2.



Gambar 4. 1 Pengujian tegangan power supply (a) tanpa beban dan (b) dengan beban

Setelah dilakukan beberapa kali pengujian, didapatkan data pengujian dalam Tabel 4.1 di bawah.

Tabel 4.1 Pengujian power supply tanpa beban

Pengujian	V Input (VDC)	V Output (VDC)	Arus (A)	V Spesifikasi (VDC)	Error (%)
1	235.2	11,99	0.12	12	0,08
2	235.2	11,99	0.12	12	0,08
3	235.2	11,99	0.12	12	0,08
4	235.2	11,99	0.12	12	0,08
5	235.2	11,99	0.12	12	0,08
Rata-rata error (%)					0,08

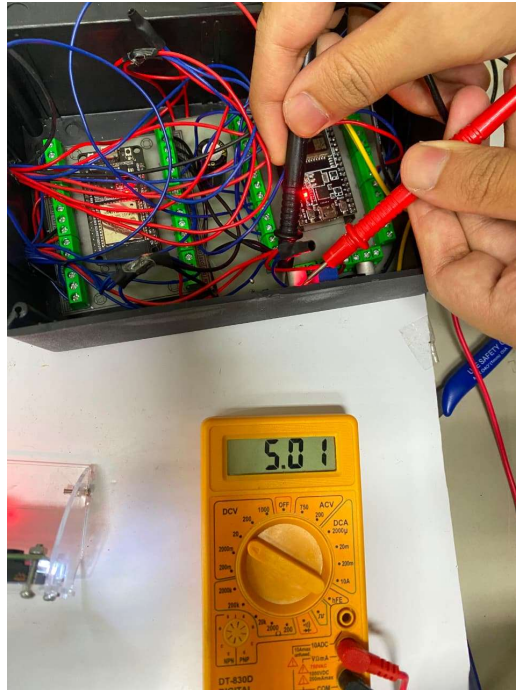
Berdasarkan Tabel 4.1, pengujian *power supply* pada kondisi tanpa beban menghasilkan tegangan keluaran yang sangat stabil sebesar 11,99 VDC pada seluruh 5 kali percobaan, dengan arus yang ditarik hanya sebesar 0,12 A. Persentase

error terhadap tegangan acuan 12 VDC tercatat sebesar 0,08% (2.1), jauh di bawah batas toleransi maksimal $\pm 5\%$. Konsistensi nilai yang identik pada seluruh percobaan menunjukkan regulasi tegangan *power supply* bekerja stabil pada kondisi tanpa beban listrik dari rangkaian sistem.

Tabel 4.2 Pengujian power supply dengan beban

Pengujian	V Input (VDC)	V Output (VDC)	Arus (A)	V Spesifikasi (VDC)	Error (%)
1	234.5	11.86	0.29	12	1,17
2	234.5	11.86	0.29	12	1,17
3	234.5	11.86	0.29	12	1,17
4	234.5	11.86	0.29	12	1,17
5	234.5	11.86	0.29	12	1,17
Rata-rata <i>error</i> (%)					1,17

Berdasarkan Tabel 4.2, pengujian pada kondisi dengan beban (seluruh sistem terhubung) menunjukkan penurunan tegangan keluaran menjadi 11,86 VDC, dengan arus yang ditarik meningkat menjadi 0,29 A akibat beban gabungan ESP32, motor servo, LCD, OLED, dan modul RFID yang beroperasi bersamaan. Persentase *error* terhadap tegangan acuan meningkat menjadi 1,17% (2.1), namun nilai ini masih berada jauh di bawah batas toleransi maksimal $\pm 5\%$ (11,4-12,6 VDC). Penurunan tegangan sebesar 0,13 VDC dari kondisi tanpa beban merupakan fenomena *voltage drop* yang wajar akibat peningkatan arus beban, dan besarnya masih tergolong sangat kecil sehingga tidak mengganggu kinerja komponen yang disuplai.



Gambar 4. 2 Pengujian tegangan modul LM2596

Setelah dilakukan beberapa kali pengujian, didapatkan data pengujian dalam Tabel 4.3 di bawah.

Tabel 4.3 Hasil tegangan modul LM2596

Pengujian	Vin (VDC)	Vout (VDC)	V Spesifikasi (VDC)	Error (%)
1	11.86	5,01	5	0,2
2	11.86	5,01	5	0,2
3	11.86	5,01	5	0,2
4	11.86	5,01	5	0,2
5	11.86	5,01	5	0,2
Rata-rata error (%)				0,2

Berdasarkan Tabel 4.3, modul *buck converter* LM2596 menerima tegangan masukan sebesar 11,86 VDC (sesuai hasil keluaran *power supply* pada kondisi berbeban) dan menghasilkan tegangan keluaran yang stabil sebesar 5,01 VDC pada

seluruh 5 kali percobaan. Persentase *error* terhadap tegangan acuan 5 VDC tercatat sebesar 0,2% (2.1), berada jauh di dalam batas toleransi maksimal $\pm 5\%$ (4,75-5,25 VDC). Konsistensi nilai keluaran ini membuktikan modul LM2596 mampu meregulasi tegangan secara andal meskipun tegangan masukannya telah mengalami sedikit penurunan akibat pembebanan sistem.

Secara keseluruhan, hasil pengujian pada ketiga kondisi menunjukkan bahwa rangkaian catu daya *prototype smart rotary parking* mampu mempertahankan regulasi tegangan dalam batas toleransi aman, baik pada kondisi tanpa beban maupun dengan beban penuh sistem. Peningkatan arus dari 0,12 A menjadi 0,29 A saat sistem beroperasi penuh hanya menyebabkan penurunan tegangan sebesar 0,13 VDC pada keluaran *power supply*, dan penurunan ini tidak berdampak signifikan terhadap kestabilan keluaran modul *buck converter* LM2596 yang tetap konsisten pada 5,01 VDC. Hal ini mengindikasikan bahwa kapasitas *power supply* memiliki *power margin* yang memadai untuk mensuplai seluruh komponen sistem secara bersamaan tanpa risiko *brownout* maupun gangguan operasional pada mikrokontroler ESP32 Master dan Slave.

4.1.2 Pengujian Modul RFID RC522

Pengujian tegangan pada pin MISO RC522 (jalur *output* data menuju ESP32) dilakukan untuk mengevaluasi karakteristik sinyal komunikasi SPI pada kondisi *idle* dan saat kartu di-*tap*, sekaligus memetakan jangkauan baca efektif modul terhadap variasi jarak. Pengujian dilakukan pada rentang jarak 0 hingga 4,5 cm dengan interval 0,5 cm, di mana jarak 4,5 cm merupakan batas maksimum modul masih mampu membaca kartu secara konsisten; pada jarak 5 cm, kartu sudah tidak lagi terdeteksi oleh pemindai. Probe negatif multimeter dihubungkan ke pin GND pada board RC522 (bukan GND ESP32) untuk meminimalkan pengaruh *ground drop* mengingat skala tegangan yang diukur cukup kecil.



(a)

(b)

Gambar 4. 3 Pengujian tegangan *output* (a) kondisi *idle* (b) kondisi tap
Setelah pengujian dilakukan, didapatkan Tabel 4.4 Data hasil pengujian tegangan *output* rfid.

Tabel 4.4 Pengujian tegangan *output* RFID

Pengujian	Jarak (cm)	Tegangan <i>Idle</i> (mV)	Tegangan Tap (mV)
1	0	79,0	69,2
2	0,5	79,5	69,3
3	1	79,9	68,8
4	1,5	79,3	69,1
5	2	79,0	69,0
6	2,5	79,4	68,7
7	3	79,3	68,5
8	3,5	79,1	69,8
9	4	79,6	68,8
10	4,5	79,2	68,9
Rata-rata		79,33	69,01

Berdasarkan Tabel 4.4, tegangan *idle* rata-rata tercatat sebesar 79,33 mV dan tegangan saat tap rata-rata sebesar 69,01 mV (2.2), dengan selisih konsisten sekitar 10,3 mV pada seluruh sepuluh percobaan (rentang 9,3-11,1 mV). Penurunan tegangan yang konsisten ini terjadi karena perbedaan aktivitas jalur *Chip Select* (SS) pada kedua kondisi. Pada kondisi *idle*, ESP32 tetap melakukan *polling* berkala terhadap register RC522 meskipun tidak ada kartu terdeteksi, namun dengan frekuensi transaksi SPI yang relatif rendah, sehingga pin MISO yang bersifat *high-*

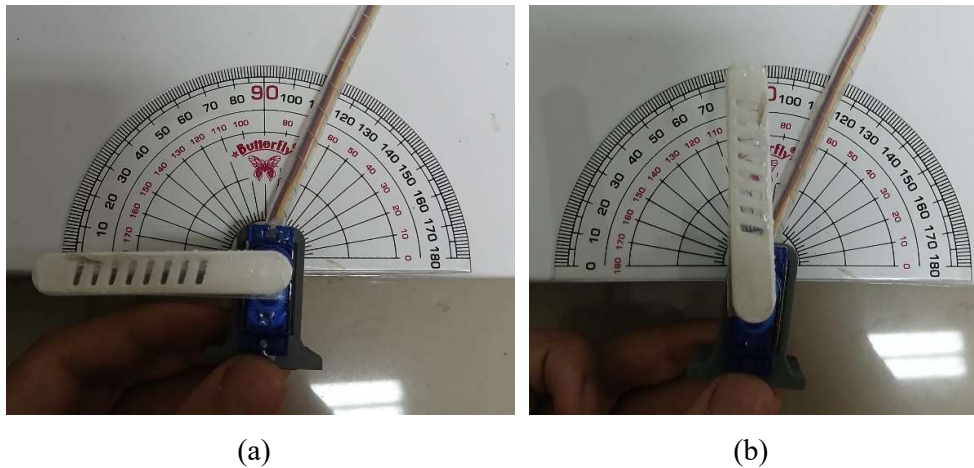
impedance (mengambang) di antara transaksi tidak didorong aktif ke level tegangan tertentu oleh RC522. Kondisi ini merupakan karakteristik umum pada arsitektur bus SPI, di mana jalur MISO dari perangkat *slave* yang tidak sedang dipilih akan berada dalam kondisi *tri-state*/mengambang, dengan level tegangan yang cenderung berada pada nilai kecil dan tidak stabil dibandingkan level logika penuh [35]. Sebaliknya, saat kartu terdeteksi, ESP32 melakukan rangkaian transaksi SPI yang jauh lebih intensif (proses *anti-collision*, seleksi UID, dan autentikasi), sehingga pin MISO lebih sering ditarik aktif ke level logika oleh RC522, menurunkan rata-rata tegangan yang terbaca oleh multimeter akibat laju *sampling* alat ukur yang terbatas terhadap sinyal yang berubah cepat.

Tingkat presisi pengukuran turut dievaluasi menggunakan standar deviasi, koefisien variasi, dan *relative range* sebagaimana dirumuskan pada Persamaan 2.3, 2.4, dan 2.5. Tegangan *idle* menghasilkan SD sebesar 0,283 mV dengan CV 0,36% dan RR 1,13%, sedangkan tegangan tap menghasilkan SD sebesar 0,367 mV dengan CV 0,53% dan RR 1,88%. Nilai CV yang berada jauh di bawah 5% pada kedua kondisi menunjukkan tingkat presisi pengukuran yang sangat tinggi, mengonfirmasi bahwa karakteristik tegangan pin MISO bersifat stabil dan bukan hasil fluktuasi acak alat ukur.

Hasil pengujian juga menunjukkan bahwa baik tegangan *idle* maupun tegangan tap tidak menunjukkan korelasi terhadap variasi jarak pembacaan, dengan fluktuasi nilai yang tetap berada pada rentang sempit (*idle*: 79,0-79,9 mV; tap: 68,5-69,8 mV) di seluruh titik jarak 0 hingga 4,5 cm. Hal ini mengindikasikan bahwa aktivitas elektrik modul RFID ditentukan oleh status komunikasi (*idle*/aktif), bukan oleh jarak fisik kartu terhadap antena, selama kartu masih berada dalam jangkauan baca efektif modul. Jangkauan baca efektif RC522 pada penelitian ini tercatat maksimum 4,5 cm, dengan kegagalan pembacaan mulai terjadi pada jarak 5 cm, konsisten dengan spesifikasi jangkauan baca RC522 pada rentang 3-5 cm menurut datasheet.

4.1.3 Pengujian Motor Servo

Pengujian motor servo dilakukan untuk mengevaluasi akurasi sudut putar aktual terhadap sudut yang diperintahkan oleh program, sebagai indikator presisi mekanis aktuator palang parkir. Pengukuran sudut aktual dilakukan menggunakan busur derajat cetak yang ditempatkan tepat pada titik pivot servo, dengan lengan servo diposisikan sejajar garis skala busur pada setiap kondisi pengujian. Setiap kondisi (tertutup dan terbuka) pada masing-masing servo diuji sebanyak 5 kali percobaan untuk memastikan konsistensi hasil, dengan dokumentasi berikut merupakan sampel representatif dari kelima percobaan tersebut.



Gambar 4. 4 Pengujian sudut motor servo *gate in* (a) posisi tertutup 0° dan (b) posisi terbuka 90°

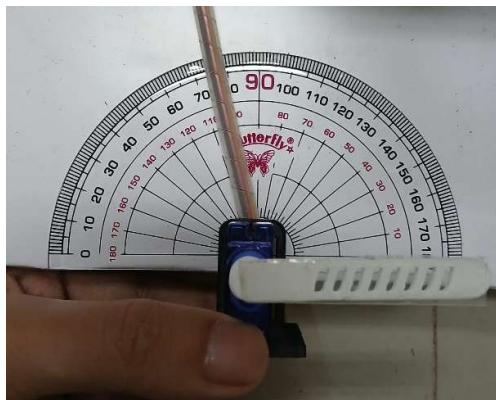
Setelah dilakukan beberapa kali pengujian, didapatkan data pengujian dalam Tabel 4.5 di bawah.

Tabel 4.5 Pengujian akurasi sudut motor servo *gate in*

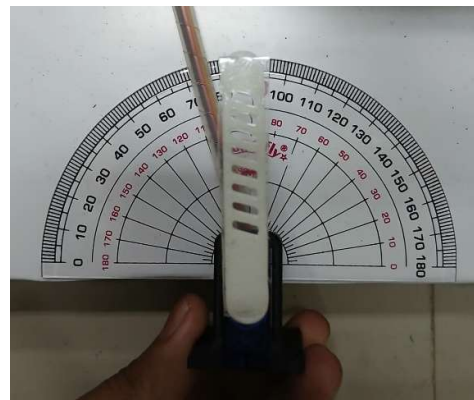
Pengujian	Sudut Perintah ($^\circ$)	Sudut Terukur ($^\circ$)
1	0°	0°
2	0°	0°
3	0°	0°

Pengujian	Sudut Perintah (°)	Sudut Terukur (°)
4	0°	0°
5	0°	0°
6	90°	90°
7	90°	90°
8	90°	90°
9	90°	90°
10	90°	90°

Berdasarkan hasil pengujian pada Tabel 4.5, motor servo *Gate In* secara konsisten menunjukkan posisi tertutup tepat pada sudut 0° dan posisi terbuka tepat pada sudut 90° di seluruh 5 kali percobaan, tanpa penyimpangan sudut sedikit pun antar-percobaan. Tidak ditemukan selisih antara sudut perintah dan sudut hasil pengukuran fisik pada kedua kondisi. Konsistensi hasil sebesar 5/5 mengindikasikan bahwa mekanisme Servo *Gate In* memiliki keterulangan (*repeatability*) mekanis yang baik, mampu kembali secara presisi ke posisi sudut yang sama pada setiap pengulangan perintah identik tanpa mengalami *drift* maupun pergeseran akibat gesekan pada engsel palang. Posisi tertutup yang presisi pada 0° memastikan palang benar-benar menghalangi jalur kendaraan masuk tanpa celah, sedangkan posisi terbuka pada 90° memastikan palang terangkat penuh sehingga tidak menghambat proses kendaraan masuk.



(a)



(b)

Gambar 4. 5 Pengujian sudut motor servo *gate out* (a) posisi tertutup 180° dan
(b) posisi terbuka 90°

Setelah dilakukan beberapa kali pengujian, didapatkan data pengujian dalam Tabel 4.6 di bawah.

Tabel 4.6 Pengujian akurasi sudut perintah motor servo *gate out*

Pengujian	Sudut Perintah (°)	Sudut Terukur (°)
1	180°	180°
2	180°	180°
3	180°	180°
4	180°	180°
5	180°	180°
6	90°	90°
7	90°	90°
8	90°	90°
9	90°	90°
10	90°	90°

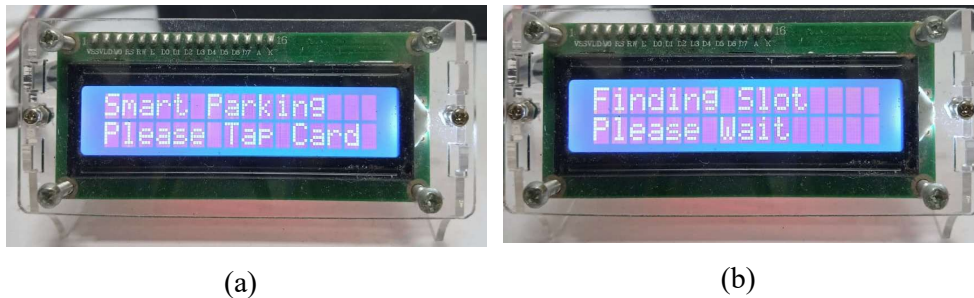
Berdasarkan hasil pengujian pada Tabel 4.6, motor servo *Gate Out* secara konsisten menunjukkan posisi tertutup tepat pada sudut 180° dan posisi terbuka tepat pada sudut 90° di seluruh 5 kali percobaan, tanpa penyimpangan sudut sedikit pun antar-percobaan. Tidak ditemukan selisih antara sudut perintah dan sudut hasil pengukuran fisik pada kedua kondisi. Konsistensi hasil sebesar 5/5 mengindikasikan bahwa mekanisme Servo *Gate Out* juga memiliki keterulangan mekanis yang baik dan setara dengan Servo *Gate In*. Posisi tertutup yang presisi pada 180° memastikan palang benar-benar menghalangi jalur kendaraan keluar, sedangkan posisi terbuka pada 90° memastikan palang terangkat penuh tanpa menghambat proses kendaraan keluar.

Perbedaan sudut posisi tertutup antara Servo *Gate In* (0°) dan Servo *Gate Out* (180°), meskipun keduanya memiliki posisi terbuka yang sama pada 90°, disebabkan oleh orientasi pemasangan mekanis yang saling berlawanan (*mirrored*) antara kedua aktuator. Kedua palang secara fungsional harus mengarah tegak lurus

ke atas (90°) saat posisi terbuka agar tidak menghalangi jalur kendaraan, namun posisi tertutupnya ditentukan oleh sisi pemasangan servo terhadap jalur gerbang masing-masing. Servo *Gate In* dipasang dengan lengan palang merebah ke arah 0° saat tertutup untuk menghalangi jalur masuk, sedangkan Servo *Gate Out* dipasang pada sisi berlawanan dengan lengan palang merebah ke arah 180° saat tertutup untuk menghalangi jalur keluar. Kedua konfigurasi sudut ini bersifat spesifik terhadap desain mekanis instalasi fisik masing-masing gerbang, bukan mencerminkan adanya inkonsistensi maupun kesalahan kalibrasi antar-aktuator.

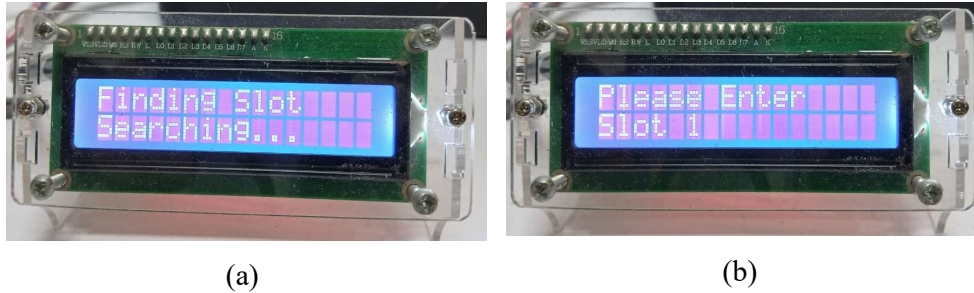
4.1.4 Pengujian LCD 16x2

Pengujian modul LCD 16x2 berbasis antarmuka I2C dilakukan untuk memastikan keandalan penyampaian instruksi visual secara *real-time* kepada pengemudi di area gerbang parkir. Pembaruan karakter pada layar diproses secara otomatis oleh fungsi `tampilLCD()` pada ESP32 Master, mengikuti enam skenario kondisi operasional sistem yang terjadi pada mekanisme *tap* tunggal.



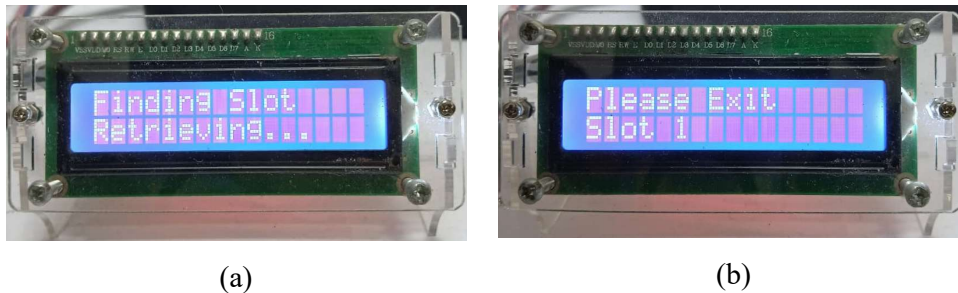
Gambar 4. 6 Pengujian kondisi LCD 16x2 (a) *idle*, (b) mencari slot kosong

Berdasarkan Gambar 4.6, pada kondisi awal (*idle*), LCD menampilkan teks identitas sistem "Smart Parking" pada baris pertama dan panduan "Please Tap Card" pada baris kedua. Begitu kartu terdeteksi dan dinyatakan belum tercatat di memori (Fase Masuk), tampilan langsung berubah menjadi "Finding Slot" disertai "Please Wait".



Gambar 4. 7 Pengujian kondisi LCD 16x2 (a) slot terisi lanjut mencari dan (b) berhasil masuk

Berdasarkan Gambar 4.7, ketika validasi visual ESP-CAM mengembalikan sinyal "T" (terisi), tampilan berubah menjadi "Searching..." pada baris kedua, menandakan mekanik *rotary* melanjutkan pencarian ke slot berikutnya. Setelah slot kosong ditemukan dan *servo gate in* terbuka, LCD menampilkan "Please Enter" disertai nomor slot tujuan sebagai informasi dinamis bagi pengemudi.



Gambar 4. 8 Pengujian kondisi LCD 16x2 (a) menjemput dan (b) berhasil keluar

Berdasarkan Gambar 4.8, pada Fase Keluar, yang kini dipicu langsung oleh satu kali *tap* pada kartu yang sudah tercatat di memori (tanpa memerlukan *tap* kedua), LCD menampilkan "Retrieving..." selama mekanik *rotary* mencari posisi slot asal kendaraan. Setelah slot ditemukan dan *servo gate out* terbuka, LCD menampilkan "Please Exit" disertai nomor slot asal.



Gambar 4. 9 Pengujian LCD 16x2 kondisi parkir penuh

Berdasarkan Gambar 4.9, saat kelima slot parkir terisi penuh, sistem langsung menampilkan notifikasi "Parking Full" disertai "Sorry :(" tanpa jeda tambahan, karena pemeriksaan kapasitas kini dieksekusi seketika begitu kartu baru terdeteksi.

Secara keseluruhan, hasil pengujian menunjukkan tingkat keberhasilan tampilan modul LCD 16x2 sebesar 100% pada tujuh skenario yang dieksekusi, dengan seluruh teks aktual yang tampil sesuai template yang dirancang pada program. Umpan balik visual kepada pengguna tersusun ringkas dan responsif, mencakup seluruh kondisi operasional mulai dari status menunggu kartu, proses pencarian slot, keberhasilan masuk maupun keluar, hingga kondisi penolakan akibat kapasitas penuh.

4.1.5 Pengujian OLED SSD1306

Pengujian modul *display* OLED SSD1306 berukuran 0.96 inci berbasis antarmuka I2C dilakukan untuk mengevaluasi keakuratan penayangan sisa kapasitas slot parkir secara *real-time* bagi pengguna gerbang. Pembacaan kondisi ketersediaan ruang parkir dieksekusi oleh sub-rutin `updateOLED()` pada ESP32 Master, yang melakukan pemindaian berulang terhadap seluruh elemen memori `array slotMemory[5]` dan merender hasilnya dalam bentuk karakter berukuran besar agar mudah terbaca dari jarak jauh.



(a)



(b)

Gambar 4. 10 Pengujian kondisi (a) 1 slot kosong dan (b) 2 slot kosong

Berdasarkan Gambar 4.10, pada kondisi tersisa 1 dan 2 slot kosong, layar OLED menampilkan angka tunggal sesuai jumlah slot yang belum terisi, dirender dengan ukuran teks besar (`setTextSize(7)`) sehingga karakter memenuhi hampir seluruh area layar dan mudah terbaca dari jarak pandang pengemudi di gerbang.



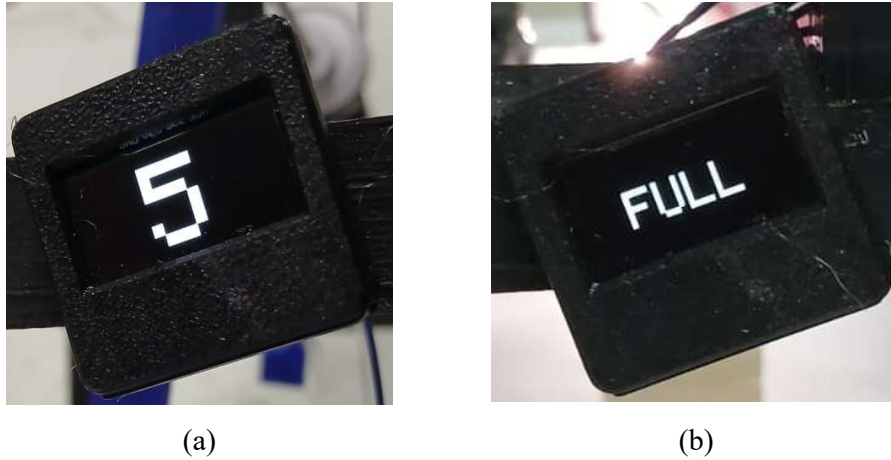
(a)



(b)

Gambar 4. 11 Pengujian kondisi (a) 3 slot kosong dan (b) 4 slot kosong

Berdasarkan Gambar 4.11, seiring bertambahnya jumlah kendaraan yang masuk melalui proses *tap-in*, jumlah slot kosong berkurang secara bertahap dari 4 menuju 3. Layar OLED merespons perubahan ini secara instan dengan memperbarui angka yang ditampilkan, tanpa jeda atau tampilan sisa digit sebelumnya (*ghosting*) yang teramati secara visual pada kedua kondisi.



Gambar 4. 12 Pengujian kondisi (a) 5 slot kosong dan (b) seluruh slot terisi

Berdasarkan Gambar 4.12, pada kondisi awal sebelum ada kendaraan masuk, OLED menampilkan angka "5" merepresentasikan kelima slot yang masih kosong. Sebaliknya, ketika seluruh slot parkir telah terisi (kosong = 0), logika program beralih dari penayangan karakter numerik ke penayangan notifikasi teks "FULL" dengan ukuran teks yang disesuaikan (`setTextSize(3)`) agar kata tersebut tetap tertampil proporsional dan terbaca jelas di tengah area layar berukuran 128×64 piksel, berfungsi sebagai penanda visual tegas bahwa kapasitas mekanik *rotary* telah maksimal.

Secara keseluruhan, pengujian modul OLED SSD1306 mencatatkan tingkat keberhasilan tampilan sebesar 100% pada seluruh enam skenario yang diujikan, dengan tampilan aktual yang identik terhadap nilai logis yang dihitung oleh sistem pada setiap kondisi. Penggunaan antarmuka I2C pada alamat *hardware* 0x3C, yang berbagi jalur *bus* fisik bersama LCD 16x2 (alamat 0x27) di GPIO 13 (SDA) dan GPIO 22 (SCL), tidak menimbulkan konflik komunikasi maupun gangguan pembaruan data pada kedua modul selama pengujian berlangsung.

4.1.6 Pengujian Komunikasi MQTT Latensi *End-to-End* dan UART Master (ESP32 V1) Slave (ESP32 V4)

Pengujian latensi dilakukan untuk mengevaluasi kecepatan dan responsivitas sistem dalam memproses serta menyampaikan informasi, baik secara

lokal pada perangkat keras maupun secara *real-time* melalui jaringan jarak jauh (*remote monitoring*). Parameter waktu tanggap (*latency*) ini menjadi tolok ukur krusial untuk memastikan bahwa setiap interaksi atau kejadian pada alat (seperti penempelan kartu RFID) dapat dieksekusi dengan cepat dan terpantau tanpa mengalami penundaan yang mengganggu kenyamanan pengguna. Dalam pengujian unjuk kerja ini, evaluasi latensi dibagi menjadi dua patokan utama, yaitu latensi lokal yang mengukur kecepatan respons langsung pada perangkat keras (layar LCD/OLED) serta latensi *end-to-end* yang mengukur durasi keseluruhan dari proses *tap* hingga data berhasil diterima, diekstrak, dan dirender ulang secara visual pada aplikasi Android melalui perantara jaringan MQTT. Pengujian dilakukan sebanyak lima kali percobaan untuk mendapatkan data kuantitatif yang objektif dan konsisten.



Gambar 4. 13 Pengujian latensi lokal dan aplikasi

Berikut pada Gambar 4.13 adalah hasil dari pengujian perbandingan latensi lokal dan aplikasi End-to-End, yang didapatkan pada Tabel 4.13 berikut

Tabel 4. 7 Pengujian latensi komunikasi mqtt

Pengujian	Latensi Lokal	Latensi <i>End-to-End</i>	Selisih Waktu
1	~35 ms	327,5 ms	292,5 ms

Pengujian	Latensi Lokal	Latensi <i>End-to-End</i>	Selisih Waktu
2	~32 ms	312,3 ms	280,3 ms
3	~38 ms	356,4 ms	318,4 ms
4	~34 ms	330,7 ms	296,7 ms
5	~36 ms	342,2 ms	306,2 ms
Rata-rata	~35 ms	333,82 ms	298,82 ms

Analisis Perbandingan Kinerja Latensi Lokal dan Aplikasi Berdasarkan Tabel 4.7 pengujian yang telah dikelompokkan menggunakan dua parameter patokan utama, terlihat adanya perbedaan karakteristik waktu tanggap yang jelas antara sistem lokal dan sistem jarak jauh. Latensi lokal yang diukur dari proses *tap* hingga tampil pada LCD/OLED mencatatkan rata-rata waktu yang sangat cepat, yakni sekitar 35 ms. Hal ini menunjukkan bahwa pemrosesan logika internal dan komunikasi perangkat keras (*hardwired*) berjalan secara seketika.

Sementara itu, ketika dijadikan patokan pada latensi *end-to-end* aplikasi Android, rata-rata waktu yang tercatat adalah 333,82 ms. Selisih waktu rata-rata sebesar ~298 ms tersebut murni merepresentasikan beban komunikasi jaringan nirkabel (pengiriman paket JSON ke *broker* MQTT publik, propagasi paket internet, hingga proses *parsing* serta *rendering* antarmuka di layar *smartphone*). Kendati demikian, dengan nilai latensi aplikasi yang tetap konsisten di bawah 0,4 detik dan memiliki koefisien variasi yang kecil, sistem terbukti memenuhi standar patokan komunikasi IoT yang andal dan responsif bagi pengguna

Pengujian latensi komunikasi serial UART antara ESP32 Master dan ESP32 Slave dilakukan dengan mengamati stempel waktu pada *Serial Monitor* Slave, mulai dari saat paket instruksi GATE_IN diterima melalui jalur *Serial2* hingga aktuator servo merespons, sebagaimana ditunjukkan pada Gambar 4.14.


```

15:15:07.243 -> TAP:43D64B1C
15:15:07.243 ->
15:15:07.243 -> [ DETEKSI ] UID 43D64B1C belum tercatat -> Fase MASUK.
15:15:07.243 ->
15:15:07.243 -> -> [ MASUK ] Lolos Validasi. Mengecek kondisi Slot SAAT INI (1)...
15:15:07.243 -> [ MQTT ] Publish: {"uid":"43D64B1C","akses":"ENTERING","gate":"CLOSED","slot_keluar":"-","durasi":"-","slot":[1,0,0,0]}
15:15:08.329 -> [ ESP-CAM ] TERISI: Slot 1 ada isinya. Lanjut putar motor mencari...
15:15:08.329 -> [ TRANSMIT ] Ke Slave: MOTOR_ON
15:15:08.371 -> [ MQTT ] Publish: {"uid":"43D64B1C","akses":"SEARCHING","gate":"CLOSED","slot_keluar":"-","durasi":"-","slot":[1,0,0,0]}
15:15:10.871 ->
15:15:10.871 -> [ SENSOR ] Keranjang fisik nomor 2 tiba.
15:15:10.871 -> [ TRANSMIT ] Ke Slave: MOTOR_OFF
15:15:10.921 -> [ MEKANIK ] Mengerem. Menunggu keranjang stabil 1.5 detik...
15:15:12.402 -> -> Membaca sinyal ESP-CAM untuk Slot 2 dalam kondisi statis...
15:15:12.531 -> [ ESP-CAM ] AMAN: Fisik Slot 2 KOSONG!
15:15:12.531 -> [ TRANSMIT ] Ke Slave: GATE_IN
15:15:12.531 -> -> Database menyimpan UID: 43D64B1C di Slot 2
15:15:12.531 -> [ MQTT ] Publish: {"uid":"43D64B1C","akses":"ENTERED","gate":"OPEN","slot_keluar":"-","durasi":"-","slot":[1,1,0,0]}
15:15:17.605 -> [ MQTT ] Publish: {"uid":"-","akses":"-","gate":"CLOSED","slot_keluar":"-","durasi":"-","slot":[1,1,0,0]}
15:15:17.605 -> [ SYSTEM ] Tampilan status di-reset kembali ke IDLE.

```

Gambar 4.15 Hasil rekaman Serial Monitor pada proses transmisi payload MQTT saat Check-In

Berdasarkan Gambar 4.15, *Serial Monitor* menampilkan log kronologis yang mengonfirmasi bahwa ESP32 Master berhasil mengeksekusi instruksi penyusunan dan pengiriman data JSON secara sinkron dengan pergerakan mekanik sistem. Pada tahap awal (waktu 15:15:07.243), saat kartu terdeteksi dan divalidasi sebagai Fase MASUK, sistem langsung menerbitkan *payload* MQTT pertama dengan parameter identitas uid terisi "43D64B1C", status akses bernilai "ENTERING", status gate "CLOSED", dan kondisi *array* slot [1,0,0,0]. Ketika sensor ESP-CAM mendeteksi Slot 1 terisi dan motor berputar mencari posisi baru (waktu 15:15:08.371), mikrokontroler secara dinamis memperbarui *payload* dan mem-*publish* status akses menjadi "SEARCHING" dengan parameter lainnya tetap sama.

Setelah keranjang nomor 2 tiba, divalidasi aman (kosong) oleh ESP-CAM, dan instruksi buka gerbang (GATE_IN) dikirim ke aktuator (waktu 15:15:12.531), ESP32 Master seketika menyimpan UID ke dalam *database* dan mem-*publish* paket data ketiga. Pada paket ini, status akses diperbarui menjadi "ENTERED", status gate menjadi "OPEN", dan matriks *array* slot menunjukkan penambahan beban pada Slot 2 menjadi [1,1,0,0]. Terakhir, saat memasuki fase penyegaran sistem (waktu 15:15:17.605), mikrokontroler mengeksekusi *reset idle* dengan menerbitkan *payload* keempat yang mengosongkan teks antarmuka menjadi "-" dan mengembalikan gate ke "CLOSED", namun berhasil mempertahankan rekaman ketersediaan ruang fisik pada matriks [1,1,0,0].

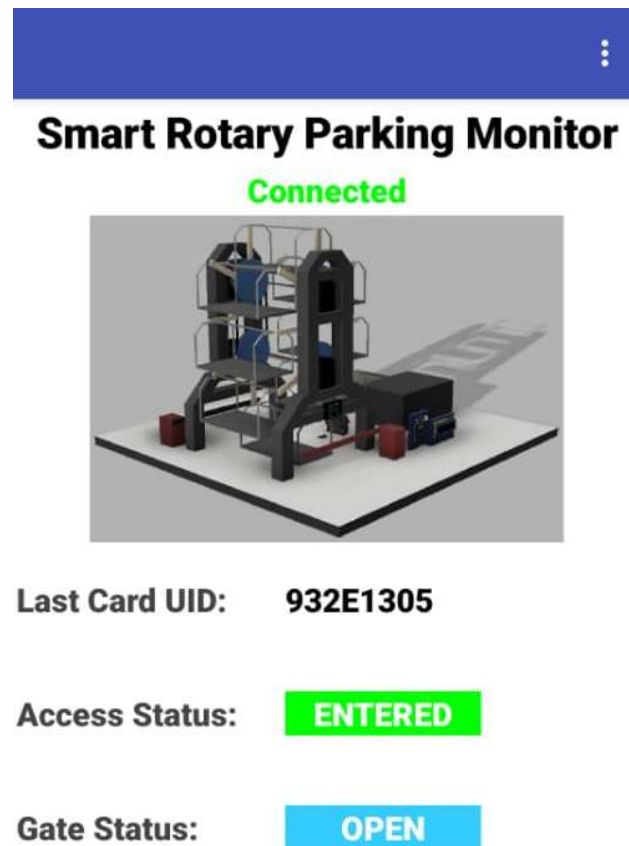
Hasil pengujian membuktikan tingkat responsivitas transmisi yang presisi, di mana empat tahapan pembaruan *payload* JSON terbentuk sempurna mengikuti transisi *state machine* alat dalam rentang waktu operasi 10,3 detik (15:15:07 hingga 15:15:17). Tidak ditemukan kegagalan pembentukan format (*parsing error*) maupun anomali parameter matriks selama pengiriman. Riwayat empiris ini memvalidasi bahwa algoritma telemetri IoT beroperasi secara stabil dan terintegrasi penuh dengan eksekusi perangkat keras, menjamin ketersediaan aliran data mutakhir di *broker* HiveMQ bagi aplikasi klien.

4.2. Pengujian Keseluruhan Sistem

Pengujian keseluruhan sistem dilakukan untuk mengamati kerja sistem secara terintegrasi, mulai dari pembacaan kartu, pergerakan gate, pencatatan slot, hingga penampilan data pada LCD, OLED, dan aplikasi *monitoring*. Pengujian dilakukan melalui beberapa skenario sebagai berikut.

4.2.1 Pengujian Proses Gate In dan Gate Out

Pengujian proses *check-in* dan *check-out* dilakukan sebanyak 5 kali masing-masing, dengan mencatat waktu proses mulai dari kartu ditempelkan hingga instruksi GATE IN/GATE OUT dikirimkan ke ESP32 Slave, mencakup keseluruhan waktu pencarian posisi slot oleh mekanik *rotary*. Pengujian dilakukan dengan membiarkan slot tujuan maupun slot asal bergeser secara alami mengikuti posisi mekanik *rotary* saat itu, tanpa direkayasa, sehingga setiap percobaan (kecuali percobaan pertama pada masing-masing skenario) merepresentasikan pergeseran sejauh 1 slot dari posisi sebelumnya.



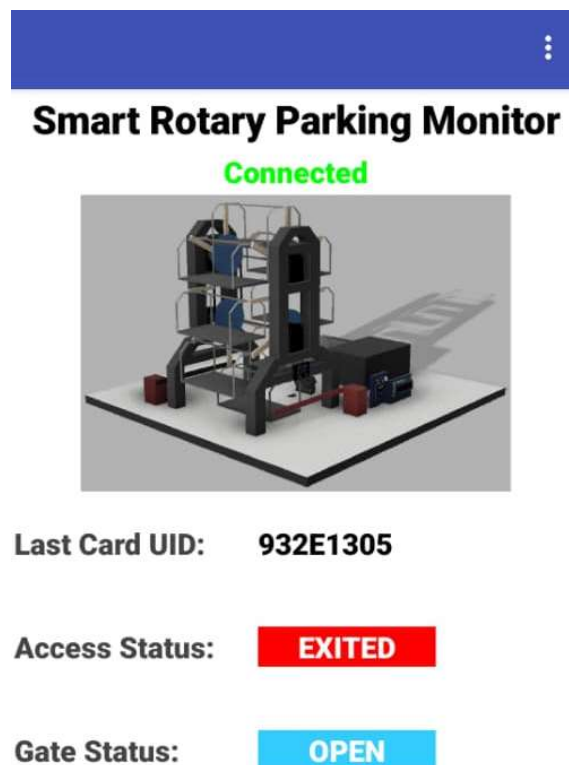
Gambar 4. 16 Hasil aplikasi pengujian gate in

Pada Gambar 4.16 adalah hasil dari hmi aplikasi yang menampilkan label UID “932E1305”, akses “ENTERED”, dan label gate “OPEN” yang artinya UID sudah tersimpan, dan mobil bisa masuk.

Tabel 4. 8 Pengujian skenario gate in

Pengujian	UID Kartu	Slot Tujuan	Waktu Proses (s)
1	0A3DD706	1	00:01,25
2	7D1EB5E3	2	00:05,39
3	B25CB4E3	3	00:05,19
4	932E1305	4	00:05,11
5	43D64B1C	5	00:05,24

Berdasarkan Tabel 4.8, waktu proses *check-in* pada percobaan pertama (Slot 1, tanpa pergeseran karena posisi *rotary* sudah berada tepat di slot tersebut) tercatat 1,25 detik. Keempat percobaan berikutnya, yang masing-masing memerlukan pergeseran sejauh 1 slot, menunjukkan waktu rata-rata 5,23 detik (P2.2) (rentang 5,11-5,39 detik). Waktu proses ini didominasi oleh tahapan validasi visual ESP-CAM pada tiap posisi kandidat slot yang dilalui mekanik *rotary*. Seluruh 5 percobaan tercatat berhasil, menunjukkan keandalan mekanisme validasi visual dalam mengidentifikasi slot kosong secara akurat.



Gambar 4. 17 Hasil pengujian aplikasi gate out

Pada Gambar 4.17 adalah hasil dari hmi aplikasi yang menampilkan label UID “932E1305”, akses “KELUAR”, dan label gate “TERBUKA” yang artinya UID sudah tersimpan, dan mobil bisa keluar

Tabel 4. 9 Pengujian skenario gate out

Pengujian	UID Kartu	Slot Asal	Waktu Proses (s)
1	43D64B1C	5	00:00,10
2	932E1305	1	00:02,65
3	B25CB4E3	2	00:02,23
4	7D1EB5E3	3	00:02,36
5	0A3DD706	4	00:02,35

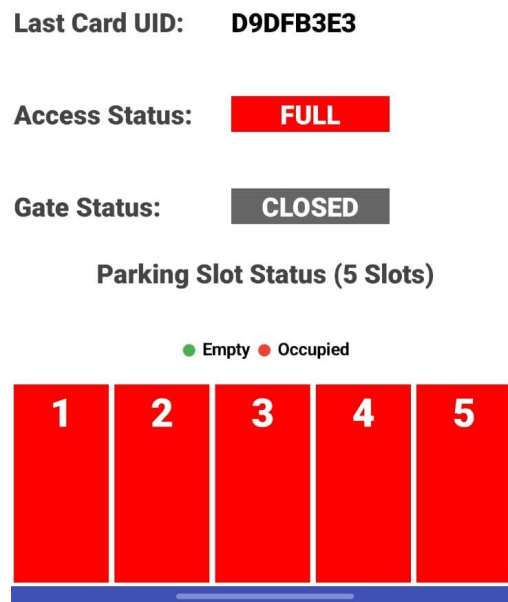
Berdasarkan Tabel 4.9, waktu proses *check-out* pada percobaan pertama (Slot 5, tanpa pergeseran karena posisi *rotary* sudah berada tepat di slot tersebut dari proses *check-in* terakhir) tercatat sangat singkat, yaitu 0,10 detik. Keempat percobaan berikutnya, yang masing-masing memerlukan pergeseran sejauh 1 slot, menunjukkan waktu rata-rata 2,40 detik (2.2) (rentang 2,23-2,65 detik) — kurang dari sepertiga waktu pergeseran setara pada proses *check-in*. Perbedaan signifikan ini disebabkan oleh mekanisme pencarian posisi yang berbeda: pada *check-out*, sistem langsung membandingkan pembacaan sensor Hall dengan nomor slot target yang telah diketahui secara pasti dari data yang tersimpan saat *check-in*, sehingga tidak memerlukan tahapan validasi visual tambahan seperti pada proses *check-in*. Seluruh 5 percobaan tercatat berhasil, menunjukkan keandalan sensor Hall dalam mendeteksi posisi slot secara presisi.

Secara keseluruhan, kesepuluh percobaan (5 *check-in* dan 5 *check-out*) menunjukkan tingkat keberhasilan sebesar 100%, membuktikan bahwa integrasi antara pembacaan RFID, validasi *database array*, mekanisme pencarian posisi *rotary* (baik berbasis visual ESP-CAM maupun sensor Hall), serta aktuasi servo palang berjalan sesuai rancangan sistem. Perbandingan waktu proses antara kedua skenario turut mengungkap bahwa metode validasi posisi berbasis citra pada proses *check-in* memerlukan waktu pemrosesan lebih dari 3 kali lipat dibandingkan metode berbasis sensor Hall pada proses *check-out*, sebuah temuan yang dapat menjadi pertimbangan untuk pengembangan sistem lebih lanjut apabila kecepatan proses *check-in* hendak dioptimalkan, misalnya dengan mempercepat siklus

pengambilan citra pada modul ESP-CAM atau menambahkan sensor posisi sebagai validasi sekunder yang lebih cepat.

4.2.2 Pengujian Kondisi Parkir Penuh

Pengujian skenario penolakan akses parkir penuh (kelima slot terisi) dilakukan menggunakan metode pengukuran otomatis berbasis skrip Python, mengikuti metode yang sama seperti pengujian latensi MQTT sebelumnya, dengan waktu diukur mulai dari kartu terdeteksi hingga *payload* status penolakan diterima oleh klien MQTT. Pengujian dilakukan sebanyak 10 kali menggunakan UID kartu yang berbeda-beda pada kondisi kelima slot dalam keadaan terisi penuh.



Gambar 4. 18 Hasil aplikasi pengujian skenario parkir penuh

Pada Gambar 4.18 adalah hasil dari hmi aplikasi yang menampilkan label UID “D9DFB3E3”, akses “FULL”, gate “CLOSED” dan untuk tampilan kondisi slot merah semua, yang artinya parkir full.

```

[TAP TERDETEKSI] UID=0A3DD706 waktu=1788703793.390555 -- menunggu MQTT...
[MQTT DITERIMA] payload={"uid":"0A3DD706","akses":"DITOLAK","gate":"TERTUTUP","slot_keluar":"-","durasi"...
[LATENSI #101] UID=0A3DD706 latensi = 246.4 ms

[TAP TERDETEKSI] UID=0A3DD706 waktu=1788703795.327045 -- menunggu MQTT...
[MQTT DITERIMA] payload={"uid":"0A3DD706","akses":"DITOLAK","gate":"TERTUTUP","slot_keluar":"-","durasi"...
[LATENSI #105] UID=0A3DD706 latensi = 250.7 ms

[TAP TERDETEKSI] UID=0A3DD706 waktu=1788703800.875664 -- menunggu MQTT...
[MQTT DITERIMA] payload={"uid":"0A3DD706","akses":"DITOLAK","gate":"TERTUTUP","slot_keluar":"-","durasi"...
[LATENSI #114] UID=0A3DD706 latensi = 279.6 ms

[TAP TERDETEKSI] UID=0A3DD706 waktu=1788703801.893943 -- menunggu MQTT...
[MQTT DITERIMA] payload={"uid":"0A3DD706","akses":"DITOLAK","gate":"TERTUTUP","slot_keluar":"-","durasi"...
[LATENSI #116] UID=0A3DD706 latensi = 218.0 ms

[TAP TERDETEKSI] UID=0A3DD706 waktu=1788703802.912372 -- menunggu MQTT...
[MQTT DITERIMA] payload={"uid":"0A3DD706","akses":"DITOLAK","gate":"TERTUTUP","slot_keluar":"-","durasi"...
[LATENSI #118] UID=0A3DD706 latensi = 241.0 ms

[TAP TERDETEKSI] UID=43D6481C waktu=1788704053.409136 -- menunggu MQTT...
[MQTT DITERIMA] payload={"uid":"43D6481C","akses":"DITOLAK","gate":"TERTUTUP","slot_keluar":"-","durasi"...
[LATENSI #120] UID=43D6481C latensi = 277.6 ms

[TAP TERDETEKSI] UID=43D6481C waktu=1788704130.582871 -- menunggu MQTT...
[MQTT DITERIMA] payload={"uid":"43D6481C","akses":"DITOLAK","gate":"TERTUTUP","slot_keluar":"-","durasi"...
[LATENSI #130] UID=43D6481C latensi = 296.3 ms

[TAP TERDETEKSI] UID=43D6481C waktu=1788704136.928069 -- menunggu MQTT...
[MQTT DITERIMA] payload={"uid":"43D6481C","akses":"DITOLAK","gate":"TERTUTUP","slot_keluar":"-","durasi"...
[LATENSI #136] UID=43D6481C latensi = 240.8 ms

[TAP TERDETEKSI] UID=43D6481C waktu=1788704148.439193 -- menunggu MQTT...
[MQTT DITERIMA] payload={"uid":"43D6481C","akses":"DITOLAK","gate":"TERTUTUP","slot_keluar":"-","durasi"...
[LATENSI #147] UID=43D6481C latensi = 223.5 ms

[TAP TERDETEKSI] UID=43D6481C waktu=1788704150.221978 -- menunggu MQTT...
[MQTT DITERIMA] payload={"uid":"43D6481C","akses":"DITOLAK","gate":"TERTUTUP","slot_keluar":"-","durasi"...
[LATENSI #149] UID=43D6481C latensi = 206.8 ms

```

Gambar 4. 19 Latensi skenario parkir penuh

Pada Gambar 4.19, seluruh *payload* yang diterima menunjukkan nilai akses bernilai "DITOLAK", membuktikan bahwa kedua UID kartu yang diujikan berhasil dikenali sistem dan menolak akses karna kondisi parkir penuh, dan hasil latensi yang ditunjukkan pada tabel 4.10 berikut:

Tabel 4. 10 Pengujian skenario parkir penuh

Pengujian	UID Kartu	Latensi (ms)
1	0A3DD706	246,4
2	0A3DD706	250,7
3	0A3DD706	279,6
4	0A3DD706	218,0
5	0A3DD706	241,0
6	43D64B1C	277,6
7	43D64B1C	296,3

Pengujian	UID Kartu	Latensi (ms)
8	43D64B1C	240,8
9	43D64B1C	223,5
10	43D64B1C	206,8
Rata-rata		248,07

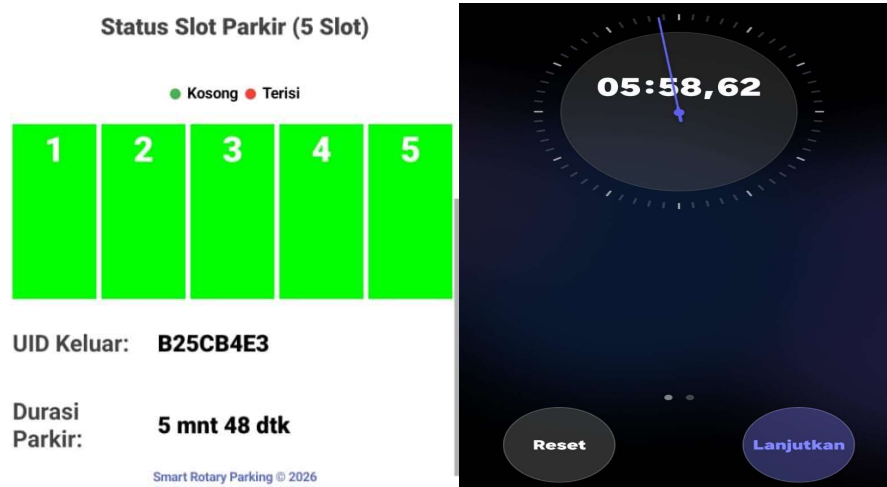
Berdasarkan Tabel 4.10, waktu respons penolakan akses pada kondisi parkir penuh tercatat rata-rata sebesar 248,07 ms (2.2), dengan nilai terendah 206,8 ms dan tertinggi 296,3 ms (rentang 89,5 ms) dari sepuluh kali percobaan. Latensi yang sangat singkat ini konsisten dengan sifat prosesnya yang murni berupa pemrosesan logis sederhana (pemeriksaan kondisi *array*) tanpa melibatkan pergerakan aktuator, diikuti satu kali transmisi *publish* status ke *broker*. Nilai latensi yang diperoleh pada skenario ini juga tercatat identik dengan latensi respons pada pengujian latensi MQTT umum (Tabel 4.7), yang wajar mengingat karakteristik proses keduanya setara.

Perhitungan menggunakan Persamaan 2.3-2.5 menghasilkan SD sebesar 28,90 ms, CV sebesar 11,65%, dan RR sebesar 36,08%. Tingkat variabilitas ini konsisten dengan karakteristik latensi berbasis komunikasi MQTT jaringan publik yang telah dibahas sebelumnya, bukan disebabkan oleh ketidakstabilan logika internal sistem. Seluruh 10 percobaan berhasil menampilkan notifikasi penolakan "Parking Full" secara tepat (*success rate* 100%), menunjukkan logika pemeriksaan kapasitas (kosong == 0) bekerja andal.

4.2.3 Pengujian Durasi Parkir

Pengujian durasi parkir dilakukan untuk mengevaluasi tingkat akurasi dan presisi pewaktuan (*timestamping*) internal ESP32 Master menggunakan fungsi `millis()` saat mencatat selisih waktu antara *check-in* (*tap-in*) dan *check-out* (*tap-out*). Hasil pewaktuan sistem yang dikirim via MQTT ke aplikasi HMI Android kemudian dibandingkan terhadap alat ukur acuan baku berupa *stopwatch* manual, menggunakan lima variasi durasi berbeda dari hitungan detik hingga menit.

Kuantifikasi deviasi pewaktuan dihitung menggunakan persentase *error* sebagaimana dirumuskan pada Persamaan 2.1.



Gambar 4. 20 Hasil pengujian komparasi durasi parkir

Berikut adalah hasil pengujian komparasi durasi parkir yang berada pada Gambar 4.20, yang mana pada hmi aplikasi menunjukkan total durasi parkir selama 5 menit 48 detik, dan untuk hasil pengujian menggunakan stopwatch dengan hasil 5 menit 48 detik 62 ms.

Tabel 4. 11 Komparasi durasi parkir dan stopwatch

Percobaan	Durasi Aplikasi	Durasi Stopwatch	Selisih (s)	Error (%)
1	13 detik (13,00 s)	00: 13,34 (13,34 s)	0,34	2,55
2	1 menit 5 detik (65,00 s)	01: 05,23 (65,23 s)	0,23	0,35
3	1 menit 58 detik (118,00 s)	01: 58,16 (118,16 s)	0,16	0,14
4	2 mnt 35 dtk (155,00 s)	02: 35,29 (155,29 s)	0,29	0,19

Percobaan	Durasi Aplikasi	Durasi Stopwatch	Selisih (s)	Error (%)
5	5 mnt 58 dtk (358,00 s)	05: 58,62 (358,62 s)	0,62	0,17
Rata-Rata			0,33	0,68

Berdasarkan Tabel 4.11, pengujian yang dilakukan sebanyak lima kali dengan variasi durasi antara 13 detik hingga 5 menit 58 detik menghasilkan rata-rata selisih waktu (Δt) sebesar 0,33 detik (2.2) dengan rata-rata persentase *error* yang sangat kecil, yaitu 0,68%.

Deviasi dalam hitungan desimal detik antara tampilan aplikasi dan *stopwatch* disebabkan oleh dua faktor utama. Pertama, terdapat perbedaan resolusi penayangan data, di mana aplikasi HMI Android sengaja didesain untuk menampilkan format pewaktuan yang *user-friendly* dalam satuan bulat detik dan menit (pembulatan terkecil 1 detik), sedangkan *stopwatch* mengukur hingga skala tingkat ketelitian 0,01 detik (10 ms). Presisi hingga satuan milidetik pada tampilan durasi parkir bagi pengguna akhir secara fungsional tidak diperlukan, mengingat konteks penggunaannya adalah estimasi lama waktu parkir kendaraan yang lazim disajikan dalam satuan detik atau menit pada aplikasi sejenis; presisi milidetik hanya relevan digunakan pada pengujian teknis performa sistem (seperti latensi MQTT dan UART) yang telah dibahas pada sub-bab sebelumnya, bukan pada antarmuka yang ditujukan langsung bagi pengguna. Kedua, selisih desimal tersebut dipengaruhi oleh faktor waktu reaksi manusia (*human reaction time*) saat penekanan tombol *start* dan *stop* pada *stopwatch* manual yang secara fisik memiliki jeda respons berkisar antara 0,15 hingga 0,30 detik bersamaan dengan penempelan kartu RFID.

Secara keseluruhan, hasil pengujian membuktikan bahwa selisih waktu yang terjadi murni berasal dari keterbatasan resolusi pembulatan tampilan aplikasi dan *delay* respons fisik penguji, bukan akibat kesalahan kalkulasi (*system error*) dari mikrokontroler. Dengan tingkat persentase keberhasilan pewaktuan mencapai 99,32% (rata-rata *error* 0,68%), mekanisme pencatatan durasi parkir berbasis

fungsi `millis()` pada ESP32 Master terbukti sangat stabil, presisi, dan dapat diandalkan sebagai basis perhitungan tarif parkir otomatis.

4.2.4 Pengujian Validasi Slot Parkir

Pengujian akurasi posisi slot parkir dilakukan dengan skenario memasukkan 5 kendaraan secara berurutan mulai dari Slot 1 hingga Slot 5, sehingga posisi mekanik *rotary* pada akhir proses *check-in* berada tepat di Slot 5. Selanjutnya, pengujian *check-out* dilakukan sebanyak 5 kali secara berkelanjutan menggunakan mekanisme *tap* tunggal tanpa mengatur ulang posisi *rotary* di antara tiap percobaan, sehingga posisi slot awal pada suatu pengujian mengikuti posisi slot akhir dari pengujian sebelumnya secara alami

```
15:59:26.613 -> TAP:0A3DD706
15:59:26.613 ->
15:59:26.613 -> [ DETEKSI ] UID 0A3DD706 sudah tercatat di memori -> Fase KELUAR.
15:59:26.613 ->
15:59:26.613 -> [ KELUAR ] Menjemput mobil di Slot: 1
15:59:26.613 -> [ MQTT ] Publish: {"uid":"0A3DD706","akses":"EXITING","gate":"CLOSED","slot_keluar":"-","durasi":"-","slot":[1,1,1,1,1]}
15:59:26.702 -> [ TRANSMIT ] Ke Slave: MOTOR_ON
15:59:29.133 ->
15:59:29.133 -> [ SENSOR ] Keranjang fisik nomor 1 tiba.
15:59:29.178 -> [ TRANSMIT ] Ke Slave: MOTOR_OFF
15:59:29.178 -> [ TRANSMIT ] Ke Slave: GATE_OUT
15:59:29.178 -> Mobil keluar. Slot 1 dikosongkan. Durasi: 2 minute 33 second
15:59:29.178 -> [ MQTT ] Publish: {"uid":"0A3DD706","akses":"EXITED","gate":"OPEN","slot_keluar":1,"durasi":"2 minute 33 second","slot":[0,1,1,1,1]}
15:59:34.214 -> [ MQTT ] Publish: {"uid":"-","akses":"-","gate":"CLOSED","slot_keluar":"-","durasi":"-","slot":[0,1,1,1,1]}
15:59:34.274 -> [ SYSTEM ] Tampilan status di-reset kembali ke IDLE.
```

Gambar 4. 21 Hasil pengujian akurasi posisi slot

Gambar 4.21 menyajikan hasil perekaman *log* data pada Serial Monitor ESP32 Master saat menguji keakuratan navigasi pergeseran posisi slot parkir. Pengujian diawali dengan pembacaan kartu RFID ber-UID 43D64B1C melalui pola penempelan 1 tap, yang secara otomatis diklasifikasikan oleh sistem sebagai instruksi Fase Keluar untuk menjemput kendaraan yang terdaftar pada Slot 1.

Tabel 4. 12 Pengujian akurasi posisi slot parkir

Pengujian	Posisi	Slot	Sensor	Slot	Slot
	Slot	Tujuan	Hall	Akhir	Akhir
	Awal		Mendeteksi	(Sistem)	(Fisik)
1	5	1	1	1	1
2	1	2	1	2	2
3	2	5	3	5	5

Pengujian	Posisi Slot Awal	Slot Tujuan	Sensor Hall Mendeteksi	Slot Akhir (Sistem)	Slot Akhir (Fisik)
4	5	4	4	4	4
5	4	3	4	3	3

Berdasarkan Tabel 4.12, posisi slot awal pada tiap pengujian mengikuti posisi akhir pengujian sebelumnya secara berkelanjutan (Pengujian 2 dimulai dari Slot 1, hasil akhir Pengujian 1; Pengujian 3 dimulai dari Slot 2, hasil akhir Pengujian 2; dan seterusnya), merepresentasikan kondisi operasional nyata di mana posisi *rotary* tidak pernah *direset* ke posisi tetap di antara transaksi. Jumlah deteksi sensor Hall pada tiap percobaan bervariasi mengikuti jarak tempuh aktual mekanik dari posisi awal ke posisi tujuan, dengan rentang 1 hingga 4 kali deteksi.

Pada seluruh 5 percobaan, slot akhir yang tercatat oleh sistem (variabel *currentSlot*) menunjukkan kesesuaian penuh dengan slot akhir fisik yang teramati secara visual, tanpa ditemukan satu pun kasus ketidaksesuaian posisi (*success rate* 100%). Konsistensi ini membuktikan bahwa mekanisme penghitungan posisi relatif berbasis sensor Hall tetap akurat meskipun diuji secara berkelanjutan pada berbagai posisi awal dan jarak pergeseran yang bervariasi, tidak terbatas pada skenario dengan posisi awal tetap.

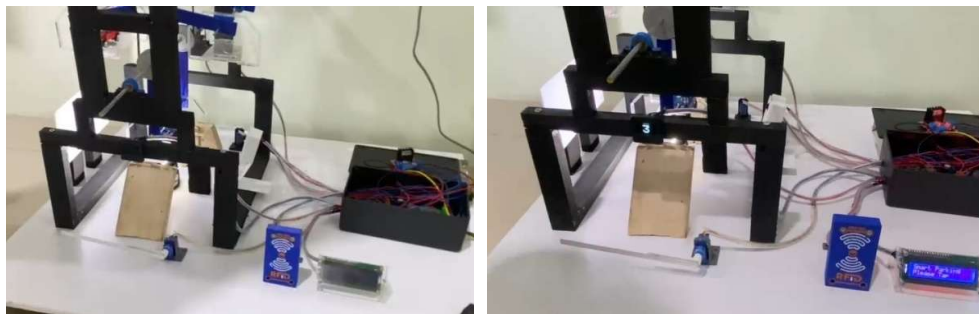
Hasil pengujian validasi posisi slot ini memperkuat keandalan mekanisme *dead-reckoning* berbasis sensor Hall yang menjadi tulang punggung sistem navigasi mekanik *rotary*. Tidak ditemukannya indikasi *drift* atau ketidaksesuaian pada seluruh variasi posisi awal dan jarak pergeseran yang diuji menunjukkan bahwa sistem penghitungan posisi cukup andal untuk mendukung operasional normal *prototype smart rotary parking* dalam rangkaian transaksi yang berkelanjutan.

4.2.5 Pengujian Pemulihan Data (NVS) terhadap Pemutusan Daya Mendadak

Pengujian ini bertujuan untuk memvalidasi keandalan fitur *Non-Volatile Storage* (NVS) dalam mencadangkan dan memulihkan konteks operasional sistem apabila terjadi pemadaman listrik secara tiba-tiba (*power outage*). Kestabilan

penyimpanan memori sangat krusial agar sistem tidak kehilangan data rekam jejak parkir dan koordinat mekanik motor saat sistem kehilangan daya secara tidak terduga.

Skenario pengujian dirancang dengan mengondisikan alat dalam keadaan beroperasi normal. Pada tahap awal, dilakukan proses penempelan (tap-in) dua buah kartu RFID yang berbeda secara berurutan, sehingga Slot 1 dan Slot 2 berstatus terisi dan layar OLED menampilkan sisa kapasitas 3 slot kosong. Kemudian, catu daya utama (adaptor 12V) dicabut secara paksa selama beberapa detik untuk menyimulasikan insiden pemutusan arus. Setelah kondisi *offline*, catu daya dihidupkan kembali (*reboot*) dan mikrokontroler dievaluasi apakah mampu memulihkan *database* array dan posisi mekanik alat tanpa intervensi manual. Setelah *booting* selesai, dilakukan penempelan kembali (tap-out) menggunakan kartu pertama untuk memverifikasi apakah sistem mengenali kartu tersebut sebagai proses keluar.



Gambar 4. 22 Pengujian Keandalan NVS Terhadap Pemutusan Daya

Berdasarkan Gambar 4.22, fitur `loadNVS()` yang dieksekusi pada fungsi `setup()` terbukti beroperasi secara sempurna dan berhasil menarik kembali variabel `array slotMemory` serta variabel `currentSlot` secara presisi dari memori *Flash*. Hal ini dibuktikan secara visual dari layar OLED yang langsung merender angka "3" (bukan "5") sesaat setelah mikrokontroler menyelesaikan *booting*.

Pengujian lebih lanjut dengan menempelkan kembali Kartu A membuktikan bahwa struktur *array memory* tidak kehilangan data UID (Unique Identifier) yang sebelumnya telah terparkir di Slot 1. Sistem mampu mengidentifikasi kartu tersebut

untuk proses *check-out* alih-alih menganggapnya sebagai mobil baru yang akan masuk. Tidak adanya pergerakan kalibrasi motor yang tidak semestinya pada saat dinyalakan juga mengonfirmasi bahwa posisi mekanik *rotary* (Slot 2) berhasil dipertahankan secara sinkron dengan NVS. Secara keseluruhan, pengujian ini mengonfirmasi bahwa arsitektur integrasi komplementer antara SRAM dan NVS yang dirancang telah sukses mencegah terjadinya kehilangan data (*data loss*) saat insiden kelistrikan ekstrem terjadi.