

BAB IV PERANCANGAN

4.1. Analisa dan Perancangan Prototipe

Pada bab ini menjelaskan dari segi proses pengerjaan sistem ini dimulai dari penyusunan pernakat keras yang dibutuhkan mulai dari membangun, perancangan, perakitan, pengkodean, hingga pengujian sistem guna memastikan kesesuaiannya dengan tujuan penelitian.

4.1.1. Analisa Kebutuhan Perangkat Keras

Membangun sistem *self-checkout* dengan *barcode scanner*, sensor RFID berbasis ESP32 tentu memerlukan perangkat keras yang dibutuhkan untuk memnagun sistem kendali secara fisik. Perangkat keras yang dibutuhkan untuk membangun sistem kendali disajikan pada tabel di bawah.

Tabel 4.1 Kebutuhan Perangkat Keras

No.	Nama Perangkat	Jumlah	Kebutuhan
1.	ESP32	1	Pusat pengolahan perintah.
2.	ESP32 <i>Expansion Board</i>	1	<i>Board</i> tambahan yang dirancang untuk memperluas fungsionalitas dan memudahkan penggunaan modul ESP32.
3.	GM67 <i>Barcode Scanner</i>	1	Pemindai kode <i>barcode</i> produk.
4.	RFID RC522	1	Pengenalan tanpa kabel melalui frekuensi radio yang dapat mengambil data tanpa harus bersentuhan.

4.1.2. Analisa Kebutuhan Perangkat Lunak

Pengembangan sistem ini tidak lepas dari dibutuhkannya sebuah perangkat lunak. Perangkat lunak di sistem ini sangat penting karena bukan hanya sebagai sistem pendukung untuk perangkat keras, tetapi merupakan sistem yang

menangani tampilan antarmuka dan pemrosesan logika. Berikut merupakan kebutuhan perangkat lunak yang dibutuhkan.

Tabel 4.2 Kebutuhan Perangkat Lunak

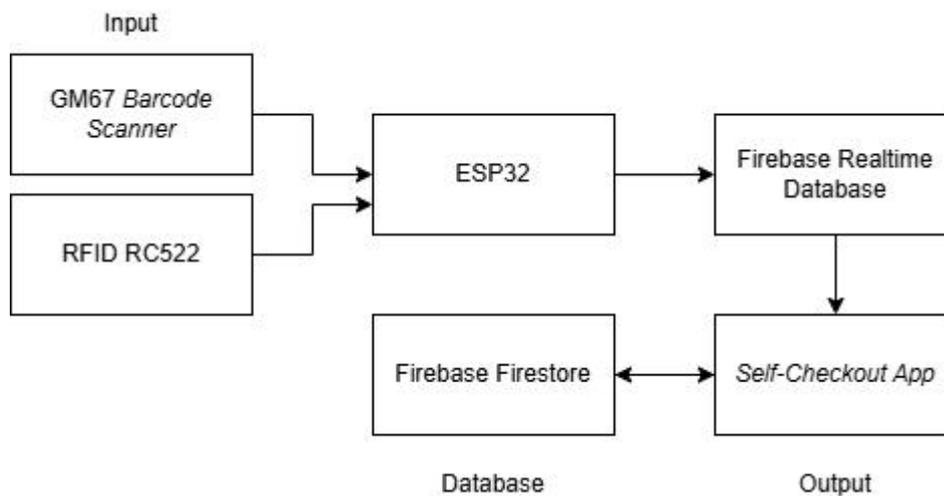
No.	Nama Perangkat	Kebutuhan
1	Windows 10 64 bit	Sistem operasi yang digunakan.
2	Arduino IDE	Perangkat lunak untuk membuat program logika dari sistem ESP32.
3	Intelij IDEA	Perangkat lunak untuk membuat program logika dari sistem <i>self-checkout</i> .
4	Flutter	<i>Framework</i> yang digunakan dalam pembuatan perangkat lunak <i>self-checkout</i> .
5	Firebase	Sebuah layanan dari Google yang digunakan sebagai <i>database</i> dan <i>console</i> pada sistem <i>self-checkout</i> .

4.2. Perancangan Prototipe

Tahapan perancangan prototipe akan menjelaskan bagaimana prototipe ini dirancang. Model prototyping adalah cara cepat untuk mendapatkan informasi tentang kebutuhan pengguna tertentu yang berfokus pada menyajikan elemen perangkat lunak. dengan metode prototype, pengembang dapat menguji dan memvalidasi fitur dan fungsionalitas sistem secara awal. (Ichwani et al., 2024).

4.2.1. Prinsip Kerja Sistem

Prinsip kerja sistem *self-checkout* ini menggambarkan bagaimana sistem bekerja dengan berhubungan dengan masing-masing sistem.

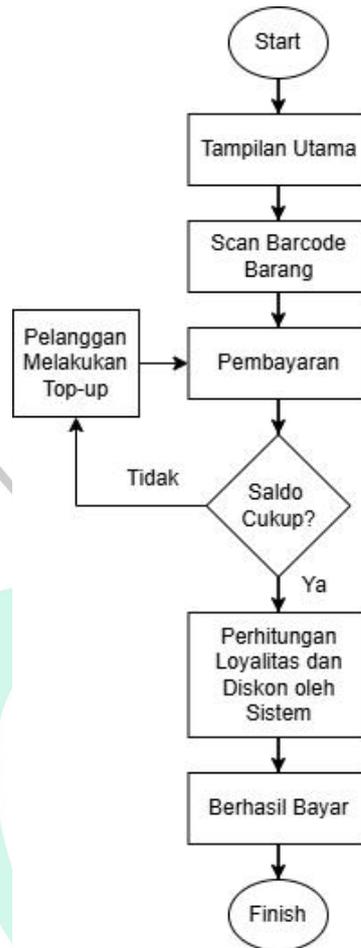


Gambar 4.1 Diagram Blok Prinsip Kerja Sistem

Gambar 4.1 merupakan prinsip kerja dari Prototipe Sistem *Self-Checkout*. Pada prototipe ini, sistem akan menerima *input* dari 2 perangkat yaitu GM67 *Barcode Scanner* dan RFID RC522. Pertama *Barcode Scanner* akan memindai *barcode* dari produk yang dibeli *customer*, kemudian hasil pemindaian akan diproses oleh ESP32 yang kemudian dikirim ke *Firestore* sebelum masuk ke aplikasi *Self-Checkout*. Aplikasi *Self-Checkout* akan mengambil data hasil pemindaian *barcode* dari *Firestore* lalu mencari data barang dari *Firestore*. Setelah *customer* selesai dengan pemindaian, *customer* akan melakukan pembayaran dengan kartu pembayaran yang di-*scan* dengan RFID RC522. Kartu yang dipindai oleh RFID akan dikirim datanya ke *Firestore* oleh ESP32, lalu dibaca oleh aplikasi *Self-Checkout*. Aplikasi kemudian akan memproses data tersebut dan terjadilah transaksi.

4.2.2. Flowchart

Flowchart di sini akan menjelaskan bagaimana logika dari sistem *self-checkout*.



Gambar 4.2 Flowchart Logika Sistem

Gambar 4.2 menjelaskan alur proses transaksi *customer*. Proses dimulai dengan *customer* melakukan *scan* barang belanjaan. Setelah selesai, *customer* akan melakukan pembayaran dan sistem akan memeriksa apakah saldo *customer* mencukupi untuk melakukan transaksi. Jika saldo tidak mencukupi, maka proses dihentikan dan sistem akan menampilkan pesan “Transaksi Gagal”. Jika saldo cukup, sistem melanjutkan proses dengan melakukan perhitungan loyalitas dan diskon. Kemudian sistem akan memotong saldo *customer* dan transaksi selesai.

4.2.3. Rancangan Sistem Loyalitas dan Diskon

Sistem loyalitas pelanggan ini menggunakan pendekatan algoritma RFM (*Recency, Frequency, Monetary*) untuk mengidentifikasi dan memberi penghargaan kepada pelanggan yang paling loyal, berdasarkan kebiasaan belanja mereka. RFM dipilih karena metodenya yang sederhana, mudah diterapkan, dan tidak memerlukan data finansial yang rumit seperti margin keuntungan atau

prediksi nilai pelanggan di masa depan, seperti yang dibutuhkan dalam pendekatan *Customer Lifetime Value* (CLV). Menurut studi terbaru dari Zheng et al. (2024), algoritma seperti SeqRFM terbukti tetap efisien dan akurat dalam mengklasifikasikan pelanggan, bahkan pada skala besar. Meskipun CLV menawarkan wawasan lebih dalam terhadap nilai jangka panjang pelanggan, perhitungannya jauh lebih kompleks dan membutuhkan data yang lebih lengkap. Sebaliknya, RFM tetap menjadi pilihan yang andal dan banyak digunakan terutama dalam tahap awal implementasi sistem loyalitas karena mampu memberikan segmentasi berbasis perilaku nyata, yang langsung bisa dimanfaatkan untuk pengambilan keputusan pemasaran (Vianna Filho et al., 2025).

Proses dimulai dengan mengambil seluruh data transaksi pelanggan dari Firestore, terutama informasi seperti ID pengguna, jumlah pembelian, dan tanggal transaksi. Data ini kemudian dikelompokkan berdasarkan *userId* agar setiap pelanggan bisa dianalisis secara individu. Setelah pengelompokan, sistem menghitung tiga komponen utama:

1. *Recency*: seberapa baru pelanggan terakhir kali melakukan transaksi,
2. *Frequency*: seberapa sering mereka berbelanja dalam periode tertentu,
3. *Monetary*: total nilai belanja pelanggan selama periode tersebut.

Ketiga nilai ini kemudian dikonversi ke dalam skala skor 1–5. Misalnya, pelanggan yang sering belanja akan mendapat skor tinggi di aspek *Frequency*. Setelah ketiga skor dihitung, sistem akan menggabungkannya menjadi RFM Score yang digunakan untuk menentukan tier loyalitas pelanggan. Pelanggan dengan skor tertinggi akan masuk ke tier tertinggi dan mendapatkan diskon yang lebih besar. Seluruh informasi skor dan tier disimpan di Firestore dalam sebuah *collection* baru, yang nantinya digunakan saat proses checkout untuk memberikan diskon otomatis sesuai level loyalitas pelanggan.

Dengan pendekatan ini, pelanggan yang sering dan konsisten berbelanja akan diberi penghargaan yang sesuai, sementara pelanggan yang hanya belanja besar sekali-sekali tidak serta-merta mendapatkan keuntungan besar. Sistem ini tidak hanya menjaga loyalitas pelanggan, tetapi juga menjaga margin keuntungan toko karena diskon diberikan secara adil dan berdasarkan data nyata. Pendekatan berbasis data seperti ini juga sejalan dengan tren terbaru yang menunjukkan

bahwa RFM merupakan solusi segmentasi yang efektif, terutama bagi bisnis kecil hingga menengah yang membutuhkan sistem yang cepat, mudah dipahami, dan dapat dijelaskan dengan jelas kepada pemilik usaha maupun pihak terkait lainnya.

Tabel 4.3 Rentang Skor dan Diskon.

Rentang Skor RFM	Kriteria Umum	Diskon yang Diberikan
$RFM \geq 4.5$	<ol style="list-style-type: none"> 1. Recency score = 5 (terakhir belanja ≤ 14 hari) 2. Frequency score = 5 (≥ 9 transaksi dalam 90 hari) 3. Monetary score = 5 (total belanja tertinggi di antara pelanggan) 	15%
$3.5 \leq RFM < 4.5$	<ol style="list-style-type: none"> 1. Recency score = 4 (belanja dalam 15–30 hari) 2. Frequency score = 4 (6 transaksi dalam 90 hari) 3. Monetary score = 4 (total belanja menengah ke atas) 	10%
$2.5 \leq RFM < 3.5$	<ol style="list-style-type: none"> 1. Recency score = 3 (belanja dalam 31–45 hari) 2. Frequency score = 3 (2–3 transaksi dalam 90 hari) 3. Monetary score = 3 	5%

	(total belanja sedang)	
RFM < 2.5	-	-

Tabel 4.4 Perhitungan Skor RFM

Komponen	Rumus Perhitungan
Recency	Hitung selisih hari antara tanggal hari ini dan tanggal transaksi terakhir pelanggan. Skor 1–5 ditentukan dengan pengelompokan (misal: ≤ 14 hari = skor 5, > 90 hari = skor 1).
Frequency	Hitung jumlah total transaksi dalam rentang 90 hari terakhir, lalu kelompokkan dalam kuantil atau rentang (misal: ≥ 6 kali = skor 5).
Monetary	Hitung total jumlah uang yang dibelanjakan pelanggan dalam 90 hari terakhir, lalu kelompokkan dari tertinggi ke terendah dan beri skor 1–5.
Nilai akhir RFM Score = $(R + F + M) / 3$, dan digunakan untuk menentukan skor pelanggan.	

4.2.4. Perancangan Pin

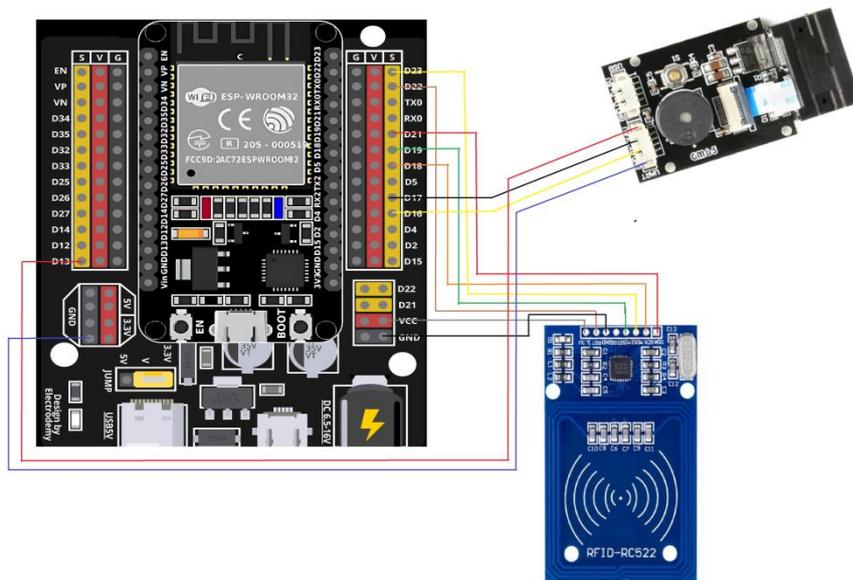
Tabel 4.5 Perancangan Pin

Pin ESP32 Expasion Board	Kebutuhan
D23	Pin pada sensor RFID MOSI
D22	Pin pada sensor RFID RST
D21	Pin pada sensor RFID SDA
D19	Pin pada sensor RFID MISO
D18	Pin pada sensor RFID SCK
D17	Pin pada <i>Barcode Reader</i> TY
D16	Pin pada <i>Barcode Reader</i> DY

D13	Pin pada <i>Barcode Reader</i> 5V
GND	Pin untuk GND pada sensor RFID dan <i>Barcode Reader</i>
VCC	Pin untuk 3.3V pada sensor RFID

4.2.4. Perancangan Rangkaian Elektronika

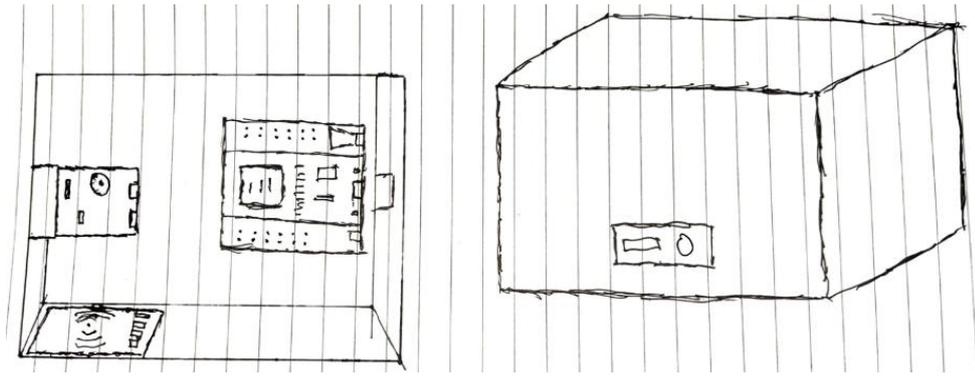
Pada Gambar di bawah ini menunjukkan skema rancangan elektronik sistem *self-checkout* yang menghubungkan antara 4 komponen utama yaitu PC, ESP32, Sensor RFID dan *Barcode Scanner*.



Gambar 4.3 Perancangan Rangkaian Elektronika

4.2.5. Perancangan Fisik Prototipe

Perancangan fisik pada sistem *self checkout* dilakukan sedemikian rupa, guna menempatkan semua komponen sistem sebagai komponen-komponen utama pembangun sistem ini seperti pada gambar berikut.



Gambar 4.4 Sketsa Rancangan Awal

4.2.6. Rancangan Pengujian *Black-Box*

Pengujian dilakukan untuk mengukur tingkat keberhasilan pemrograman yang telah dikembangkan sebagai inti dari sistem *Self-Checkout*, berdasarkan algoritma yang telah dirancang sebelumnya. Metode pengujian yang digunakan adalah *black box testing*, yaitu metode yang berfokus pada pengujian fungsionalitas sistem tanpa memperhatikan struktur internal program. Pemilihan metode ini didasarkan pada karakteristik sistem yang melibatkan pengendalian perangkat keras, sehingga diperlukan pengujian yang dapat mengamati respons sistem terhadap instruksi yang diberikan melalui program (Pressman, 2010). Untuk mendukung proses pengujian ini, Tabel 8 berikut menyajikan rancangan pengujian sistem yang akan digunakan sebagai acuan dalam pelaksanaannya.

Tabel 4.6 Rancangan Pengujian *Black-Box*

No.	Skenario Pengujian, Hasil yang Diharapkan dan Hasil Pengamatan	
1	<i>Barcode Scanner</i> dapat memindai <i>barcode</i> dari produk.	Hasil yang dikehendaki: <i>Barcode</i> terpindai dengan benar dan muncul di Realtime Database
2.	Sensor RFID dapat memindai kartu RFID.	Hasil yang dikehendaki: Kode RFID terpindai dengan benar dan muncul di Realtime Database
3.	Barang berhasil di tambahkan ke <i>cart</i> setelah <i>barcode</i> dipindai.	Hasil yang dikehendaki: Barang muncul di sebelah kanan layar tempat kumpulan barang yang

		dibeli
4.	Harga barang terakumulasi dengan benar.	Hasil yang dikehendaki: Harga barang terakumulasi sesuai dengan total barang yang dibeli
5.	Saldo RFID terpotong dengan nominal yang benar setelah pembayaran.	Hasil yang dikehendaki: Akumulasi saldo ditampilkan dengan baik
6.	Pelanggan melakukan 9 transaksi dalam 3 bulan terakhir dengan total belanja Rp5 juta	Hasil yang dikehendaki: Sistem memberikan diskon 15%
7.	Pelanggan belanja 6 kali dengan total Rp2 juta, transaksi terakhir 20 hari lalu.	Hasil yang dikehendaki: Sistem memberikan diskon 10%
8.	Pelanggan baru belanja 3 kali dalam 3 bulan dengan nominal Rp800.000	Hasil yang dikehendaki: Sistem memberikan diskon 5%
9.	Pelanggan hanya belanja 1 kali dalam 3 bulan dengan nominal Rp200.000	Hasil yang dikehendaki: Sistem tidak memberikan diskon