BAB I

PENDAHULUAN

A. Latar Belakang

Meningkatnya teknologi makin mempermudah orang dalam melakukan aktivitas dalam melaksanakan pembayaran, seperti kamu tahu bahwa semakin lama jumlah pengunjung akan bertambah terkhususnya di wisata Puncak Bila, agar memudahkan sistem pembayaran tersebut. Informasi dan pengetahuan semakin mudah diakses melalui berbagai platform (Tahir et al., 2025). Dalam era globalisasi yang pesat ini, perkembangan teknologi telah membawa banyak kemajuan dalam berbagai aspek kehidupan (Edwin et al., 2024). Terdapat teknologi yang dapat mengidentifikasi sesuatu secara otomatis atau sering di bilang *Automatic Identification (auto*-ID*)*. *auto*-ID yaitu metode pegambilan data dengan cara mengidentifikasi objek secara otomatis.

Dengan perkembangan Seiring berjalannya waktu, sistem teknologi terus mengalami kemajuan, sehingga informasi dapat lebih mudah tersebar ke berbagai kalangan masyarakat. Hal ini bertujuan untuk memudahkan aktifitas manusia dalam sehari-hari. Kemajuan teknologi dan kecepatan untuk memperoleh informasi tidak hanya bermanfaat dalam bidang pendidikan, tetapi juga memberikan dampak positif di berbagai aspek kehidupan kegiatan berekonomi.

Saat ini kita berada di zaman dunia yang serba canggih dan ditentukan kecepatan waktu, seperti dalam halnya kegiatan berekonomi yang mengunakan

kecanggihan teknologi informasi yang mempermudah masyarakat untuk melakukan teransaksi secara digital atau juga pembayaran untuk pembelian dan tagihan dengan menggunakan kartu.

RFID merupakan sebuah teknologi relatif baru yang memanfaatkan gelombang radio dengan frekuensi tertentu untuk mengidentifikasi suatu benda atau objek. Pemanfaatan RFID dapat memudahkan dalam melakukan pembayaran terutama pada wahana puncak bila sehingga dapat membantu petugas pada saat transaksi.

Radio Frequency Identification adalah salah satu teknologi yang mengunakan metode *auto*-ID. Dengan RFID penulis tertarik untuk membuat sebuah sistem pembayaran di wahana puncak bila yang memakai RFID. Dengan adanya sistem yang sedang berjalan saat ini yaitu secara manual dapat memakai banyak waktu, dengan memerlukan sistem yang akan memiliki *tag* RFID masing-masing dan kartu teg tersebut dapat diisi saldo yang nantinya ditukarkan dengan uang ditempat yang telah di sediakan. Sehingga pegunjung hanya perlu membawa kartu *teg* RFID yang telah terisi saldo pada saat memasuki sebuah wahana, setelah memasuki wahana maka saldo pegunjung akan berkuran sesuai dengan biaya masuk di setiap wahana yang di pilih.

Berdasarkan masalah diatas, maka penulis berencana melakukan penelitian mengenai pengunakan *Radio Frequency Identification* (RFID) dalam bentuk skripsi dengan judul "**APLIKASI PEMBAYARAN WAHANA BERBASIS RFID**"

B. Rumusan Masalah

Berdasarkan latar belakang diatas, maka dapat diidentifikasi beberapa masalah dalam merancang aplikasi Pembayaran Wahana Berbasis RFID yakni sebagai berikut:

1. Bagaimana cara merancang Aplikasi Pembayaran Wahana Berbasis RFID.
2. Bagaimana Menerapkan Aplikasi Pembayaran Wahana Berbasis RFID.

C. Tujuan Penelitian

Penelitian ini bertujuan untuk mengeimplementasikan sistem transaksi wahana menggunakan RFID sebagai alat pembayaran yang bertujuan untuk memudahkan pengelolah dalam mengatur sistem lebih praktis dan cepat pada pengeloalah maupun pengguana dalam melakukan transaksi pada Pembayaran Wahana Menggunakan RFID.

D. Batasan Masalah

Adapun batasan masalah sebagai berikut:

1. Aplikasi ini menggunakan sistem RFID
2. Aplikasi ini hanya dapat digunakan di web
3. Aplikasi ini digunakan di sekitar wisata puncak bila

E. Mannfaat Penelitian

Diharapkan dapat membantu pengerjaan pengelolah dan mempermudah pegunjung agar lebih cepat dalam melakukan transaksi wahana yang lebih praktis dan efektif serta mengurangi antrian panjang.

BAB II

TINJAUAN PUSTAKA

A. Kajian Hasil Penelitian Terdahulu

1. Asriani (2014) "penelitian yang berjudul **Prototype Palang Pintu Otomatis Menggunakan RFID (*Radio Frequncy Indentification*)". Dalam penelitian ini merancang sebuah sistem yang dapat mengetahui keluar dan masuknya kendaraan dengan menggunakan RFID sedangkan penelitian yang saya lakukan adalah sebuah sistem yang mengatur masalah pembayaran wahana.
2. Febry Eka Purwiantono[1], Muhammad Sofyan Romli[2], addin aditya[3] (2019) "pada penelitian yang berjudul **Pemanfaatan RFID (Radio Frequncy Identification) Sebagai Alternatif Absensi Siswa (Studi Kasus : SMK Ar-Rahmah Sukabumi Jawa Barat)"**. Dalam penelitian ini dirancang sebuah sistem yang mampu dalam mengelola data absensi siswa pada suatu pendidikan sedangkan penelitian yang saya lakukan adalah mengatur masalah pembayaran pada wisata wahana.
3. Jimy Satria Simanjuntak[1] , Jaka Prayudha[2] ,Jufri Halim[3] (2022) "pada penelitian yang berjudul **Sistem Pembayaran Non Tunai Dengan Memannfaatkan E-KTP Mengunakan Teknik Simplex Berbasis Arduino"**. Dalam penelitian ini dirancang sebuah sistem pembayaran tidak tunai menggunakan kartu E-KTP sebagai *input*-an proses

pembayaran.Sedangkan Penelitian yang saya lakukan menggunakan RFID dalam melakukan pembayaran wahana.

B. Aplikasi

Aplikasi merupakan sebuah program yang dikembangkan para pengguna untuk penyelesaian tugas tertentu (Kadir, 2003). Menurut Kadir (2008:3), "program aplikasi adalah perangkat lunak siap pakai yang dirancang untuk menjalankan fungsi tertentu bagi pengguna maupun aplikasi lainnya. Secara umum, aplikasi dapat di artikan sebagai penerapan dari suatu konsep yang menjadi fokus pembahasan, atau sebagai perangkat lunak yang diciptakan untuk membantu orang menyelesaikan pekerjaan tertentu. Aplikasi perangkat lunak ini dirancang untuk kebutuhan pengguna tertentu dan terbagi jadi dua klasifikasi, yaitu: perangkat lunak spesialis dengan dokumentasi program terintegrasi yang dibuat untuk menjalankan tugas khusus, serta perangkat lunak paket dengan dokumentasi program terintegrasi yang dibuat untuk menyelesaikan jenis masalah tertentu".

C. Pembayaran Wahana

Sistem pembayaran merupakan komponen penting yang tidak dapat terlepas dari sistem keuangan. Sistem ini mencakup seperangkat pengaturan, Kesepakatan, sarana operasional, dan prosedur teknis yang digunakan untuk mengirim, memberikan otorisasi, serta menerima instruksi terkait pembayaran. Secara umum, sistem pembayaran diklasifikasikan menjadi dua jenis utama, yaitu sistem pembayaran ritel (Retail Payment System) (Untoro, 2014). Perjanjian pemesanan tersebut dapat berupa perjanjian atas pembayara wahana dan lainnya pada waktu

tertentu dan disertai kemudahan akses (Henriani & Ade Hastuty, 2024). Pembayaran wahana sendiri merujuk pada sistem yang mengelola transaksi antara pengelola wahana dan para pengunjung.

D. RFID (Radio Frequency Identification)

Menurut Rusdi, 2013, “Radio Frequency Identification ialah teknologi komunikasi yang memanfaatkan gelombang radio untuk memungkinkan pertukaran data antara RFID Reader (penerima) dan RFID Tag (pengirim) dengan cara meletakan karu RFID diatas RFID reader. Dua komponen yang utama dalam penerapan teknologi ini adalah RFID Reader dan RFID Tag”. RFID memiliki beberapa jenis yang diklasifikasikan berdasarkan frekuensi operasionalnya.

RFID tag pasif tidak dilengkapi dengan sumber daya internal seperti baterai, sehingga antenanya berfungsi sebagai penyedia energi dengan memanfaatkan medan elektromagnetik dari reader/writer dan memodulasikannya untuk mengirimkan data yang tersimpan dalam tag. Data tersebut kemudian dibaca oleh reader/writer dan diteruskan ke sistem basis data pada komputer. Reader/writer memancarkan gelombang elektromagnetik yang ditangkap oleh antena pada RFID tag, dan RFID tag akan membalas dengan mengirimkan data — biasanya berupa nomor identifikasi — melalui gelombang radio kembali ke reader/writer. Proses komunikasi antara reader/writer dan RFID tag dilakukan melalui sinyal radio.

Pada sistem RFID pasif, reader/writer menghasilkan energi gelombang radio yang digunakan untuk mengaktifkan RFID tag sehingga dapat beroperasi. Sebaliknya, pada sistem RFID aktif, tag dilengkapi dengan baterai internal yang memungkinkan jangkauan komunikasi lebih luas serta mendukung fitur tambahan seperti pengukuran suhu. Informasi yang diperoleh dari RFID tag selanjutnya dikirimkan melalui jaringan komunikasi, baik kabel maupun nirkabel, ke sistem komputer untuk diproses lebih lanjut.

1. RFID *Tag*

RFID Tag merupakan perangkat yang terdiri dari rangkaian elektronik dan antena yang terintegrasi dalam satu kesatuan. Dalam penelitian ini, RFID Tag dimanfaatkan sebagai pengganti e-KTP. Komponen elektronik pada RFID Tag umumnya dilengkapi dengan memori, sehingga perangkat ini mampu menyimpan data. Memori pada tag terbagi ke dalam beberapa sel, di mana sebagian sel berisi data yang hanya dapat dibaca (Read Only), seperti nomor seri unik yang sudah ditanamkan sejak proses produksi (Prasetya, Suhartono, & Sholekan, 2014). Selain itu, terdapat juga jenis memori pada RFID yang memungkinkan untuk ditulis dan dibaca berkali-kali. Contoh bentuk RFID Tag dapat dilihat pada Gambar 2.1.



Gambar 2. 1 RFID *Tag*

Berdasarkan sumber dayanya, RFID tag terbagi menjadi dua jenis, yaitu RFID aktif dan RFID pasif. RFID aktif dilengkapi dengan baterai internal (on-board battery), sehingga tag ini secara terus-menerus dapat memancarkan sinyal layaknya suar. Sebaliknya, RFID pasif tidak memiliki sumber daya sendiri. Tag ini hanya mengandalkan induksi listrik yang dihasilkan oleh antena sebagai respons terhadap gelombang radio dari proses pemindaian, yang cukup untuk mengaktifkan chip di dalamnya agar dapat mengirimkan sinyal balasan ke RFID reader/writer.

2. RFID Reader/Writer

Dalam penelitian ini, digunakan RFID tag pasif tipe Mifare Classic 1K dengan frekuensi operasi sebesar 13,56 MHz. Oleh karena itu, dibutuhkan modul RFID reader/writer yang kompatibel dengan frekuensi tersebut dan mampu membaca tag jenis Mifare Classic 1K. Dengan mempertimbangkan kemudahan penggunaan serta harga yang ekonomis, peneliti memilih modul RFID reader/writer RC522 buatan NXP Semiconductors. Wujud fisik dari modul ini dapat dilihat pada Gambar 2.2



Gambar 2. 2 RFID Reader

E. Esp32

ESP32 adalah sebuah System-on-a-Chip (SoC) mikrokontroler yang dirancang khusus untuk aplikasi IoT (Internet of Things). Diproduksi oleh Espressif Systems, ESP32 merupakan penerus dari ESP8266 yang telah populer sebelumnya. Apa yang membedakan ESP32? Salah satu keunggulan utama ESP32 adalah integrasi modul Wi-Fi dan Bluetooth di dalamnya, sehingga memungkinkan perangkat yang menggunakan ESP32 untuk terhubung ke internet dan berkomunikasi dengan perangkat lain secara nirkabel. Selain itu, ESP32 juga ada berbagai peripheral lain seperti ADC, DAC, sensor, dan antarmuka komunikasi serial yang membuatnya sangat fleksibel dalam berbagai aplikasi.



Gambar 2. 3 Esp32

Keunggulan ESP32 tidak hanya terletak pada konektivitasnya yang mumpuni, tetapi juga pada performa dan efisiensi energinya. Dengan prosesor dual-core dan dukungan untuk berbagai mode tidur, ESP32 dapat beroperasi dengan daya yang sangat rendah, menjadikannya ideal untuk perangkat IoT yang membutuhkan daya tahan baterai yang lama. Potensi aplikasi ESP32 sangat luas, mulai dari perangkat smart home sederhana seperti lampu pintar dan sensor suhu hingga sistem kontrol industri yang kompleks.

F. Rdm 6300

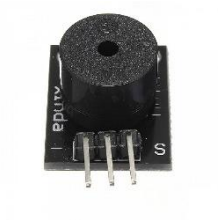
Pada kesempatan kali ini saya akan menjelaskan mengenai bagaimana cara mengakses RFID Reader RDM6300 / RDM630 dengan menggunakan Arduino. modul RFID ini sangatlah mudah sekali diakses dan juga harga yang relatif murah dari pada produk RFID reader lainnya. prinsip kerja dari alat ini sangatlah sederhana yaitu menampilkan kode Tag Card dan mencocokkan ID-nya, jika sama maka akan terdapat tulisan yang telah ditentukan, jika berbeda maka ada tulisan yang telah ditentukan pula. alat ini sangat cocok buat buka-tutup pintu atau mesin absensi karyawan.



Gambar 2. 4 Rdm6300

G. Buzzer

Buzzer Elektronika adalah sebuah komponen elektronika yang dapat menghasilkan getaran suara berupa gelombang bunyi. Buzzer elektronika akan menghasilkan getaran suara ketika diberikan sejumlah tegangan listrik dengan taraf tertentu sesuai dengan spesifikasi bentuk dan ukuran buzzer elektronika itu sendiri. Pada umumnya, buzzer elektronika ini sering digunakan sebagai alarm karena penggunaannya yang cukup mudah yaitu dengan memberikan tegangan input maka buzzer elektronika akan menghasilkan getaran suara berupa gelombang bunyi yang dapat didengar manusia.



Gambar 2. 5 Buzzer

Pada dasarnya, setiap buzzer elektronika memerlukan input berupa tegangan listrik yang kemudian diubah menjadi getaran suara atau gelombang bunyi yang memiliki frekuensi berkisar antara 1 - 5 KHz. Jenis buzzer elektronika yang sering digunakan dan ditemukan dalam rangkaian adalah buzzer yang berjenis Piezoelectric (Piezoelectric Buzzer). Hal itu karena Piezoelectric Buzzer memiliki berbagai kelebihan diantaranya yaitu lebih murah, relatif lebih ringan dan lebih mudah penggunaannya ketika diaplikasikan dalam rangkaian elektronika.

H. Kabel jumper

Dalam dunia teknologi dan elektronika, kabel jumper memiliki peran yang sangat penting. Sebagai penghubung antara komponen-komponen elektronik, kabel jumper memungkinkan terciptanya sambungan yang efisien dan efektif dalam sebuah rangkaian elektronik. Penggunaan kabel jumper tidak hanya terbatas pada perangkat komputer atau elektronik konsumen, namun juga luas digunakan dalam sistem-sistem industri dan proyek-proyek DIY (Do It Yourself). Pada artikel ini, Sonde.id akan membahas secara mendalam tentang pengertian kabel jumper, fungsi utamanya dalam dunia elektronika, cara kerja yang sederhana namun efektif, jenis-jenis kabel jumper yang umum digunakan, serta spesifikasi yang perlu diperhatikan saat memilih kabel jumper. Dengan pemahaman yang baik tentang kabel jumper, diharapkan pembaca dapat mengoptimalkan penggunaannya dalam berbagai proyek elektronika, baik itu untuk keperluan pemrograman, pengujian, perbaikan, atau modifikasi perangkat elektronik.



Gambar 2. 6 Kabel jumper

I. Kabel usb

Kabel USB merupakan kebutuhan yang tidak termasuk primer namun kehadirannya sangat penting dalam kehidupan sehari-hari. Anda yang menggunakan smartphone pasti sangat kenal dengan USB cable, karena digunakan untuk kebutuhan pengisian daya baterai.

Namun tidak hanya itu saja, USB cable juga berfungsi pada komputer dan jenis gadget lainnya karena digunakan sebagai konektor atau penghubung antar perangkat elektronik. USB memiliki banyak keistimewaan lainnya, sehingga Anda wajib mengetahuinya dari halaman ini.

USB atau Universal Serial Bus merupakan media transmisi yang bertujuan untuk memudahkan antara dua alat elektronik agar bisa saling terhubung dan terkoneksi satu dengan lainnya.dengan adanya kabel jenis ini, Anda bisa membuat dua perangkat elektronik saling terhubun bahkan juga melakukan berbagai fungsi lainnya.

J. Database

"Database merupakan himpunan data yang saling terintegrasi dan tersusun secara sistematis sehingga memungkinkan proses manipulasi, pencarian, dan pengambilan data dilakukan secara efisien", (Raharjo, 2015:2). Selain memuat data, sebuah database juga mengandung meta data, yaitu informasi yang mendeskripsikan struktur dari data itu sendiri (Raharjo, 2015:2). Berdasarkan uraian tersebut, penulis menyimpulkan bahwa database adalah media penyimpanan

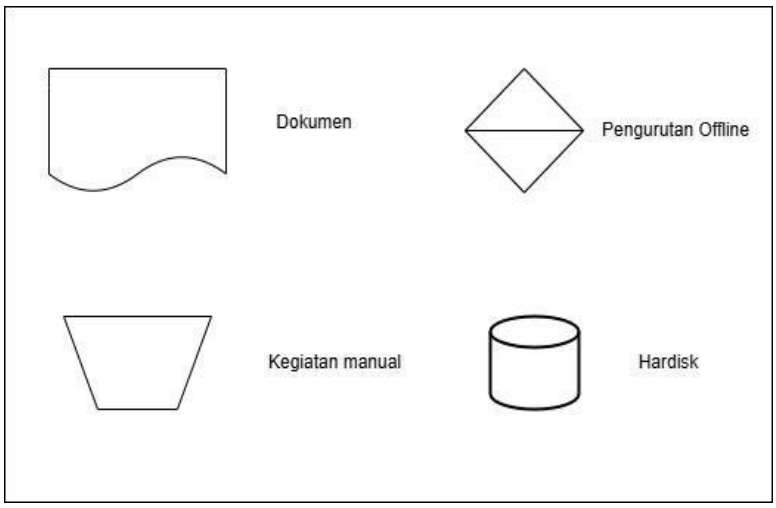
dan pengolahan data yang memungkinkan integrasi dengan program terkait, seperti VB.NET 2010. Dalam database, data dan metadata tersimpan secara terstruktur. Data yang terdapat di dalamnya dikategorikan menurut jenisnya dan disimpan dalam unit-unit terpisah yang disebut tabel.

K. Flowcart

Flowchart merupakan bentuk visual yang menggambarkan urutan langkah-langkah untuk menyelesaikan dalam suatu permasalahan, disusun menggunakan serangkaian simbol. Setiap simbol dalam *flowchart* merepresentasikan aktivitas atau proses tertentu. Secara umum, *flowchart* dimulai dari tahap penerimaan masukan (input), dilanjutkan dengan proses pengolahan data, dan akan diakhiri dengan penyajian keluaran (output). Adapun kata lain, *flowchart* adalah diagram yang menggambarkan alur kerja yang mencakup pembacaan data, pemroses informasi, pengambilan keputusan, serta persiapan hasil dari proses tersebut.

a. *Flowchart* sistem

Merupakan diagram yang menggunakan simbol-simbol khusus untuk menampilkan urutan proses pemindahan data dari satu file pada media ke file lainnya dalam media yang berbeda, dalam konteks sistem pengolahan data. Terdapat beberapa contoh *flowchart* sistem antara lain:



Gambar 2. 7 Flowchart sistem

b. *Flowchart* program

Flowchart adalah diagram yang menggunakan banyak imbol terkait untuk menggambarkan secara tepat urutan proses serta hubungan antara proses dalam sebuah program.

1. Kaidah-kaidah umum pembuatan *flowchart* program

Dalam menyusun *flowchart* program, tidak terdapat aturan baku atau rumus mutlak yang harus diikuti. Hal ini dikarenakan *flowchart* merupakan representasi hasil pemikiran untuk menganalisis suatu permasalahan menggunakan komputer. Oleh karena itu, bentuk *flowchart* yang dihasilkan bisa berbeda-beda bentuk antara satu program dengan program lainnya. Tetapi, secara umum, setiap proses pengolahan data terdiri dari tiga bagian utama, yaitu: Input, Pengolahan, dan Output.



Dalam pengolahan data menggunakan komputer, terdapat urain dasar dalam menyelesaikan satu permasalahan, yaitu sebagai berikut:

- START: Menunjukkan pernyataan awal yang berfungsi untuk mempersiapkan perangkat atau peralatan yang dibutuhkan sebelum proses penyelesaian masalah dimulai.
- READ: Menyatakan aktivitas membaca data dari perangkat input yang tersedia.
- PROCESS: Menjelaskan serangkaian tindakan yang dilakukan untuk memecahkan masalah berdasarkan data yang telah dibaca.
- WRITE: Merupakan pernyataan untuk menyimpan atau merekam hasil pemrosesan ke dalam perangkat output.
- END: Menandai berakhirnya proses pengolahan data.

Berikut ini ialah beberapa contoh simbol *flowchart* yang secara umum dalam dunia pemrograman

a. Simbol Input



Gambar 2. 8 simbol imput

Simbol input dalam *flowchart* dilambangkan dengan bentuk jajar genjang. Simbol ini berfungsi untuk merepresentasikan aktivitas penerimaan data atau masukan. Pada simbol ini, data input yang diperlukan dapat dituliskan satu demi satu atau secara keseluruhan. Namun, umumnya seluruh input yang dibutuhkan pada suatu waktu dicantumkan sekaligus dalam satu simbol untuk menghemat ruang dalam penyusunan diagram.

b. Simbol Proses



Gambar 2. 9 Simbol Proses

Simbol proses dalam *flowchart* digambarkan dengan bentuk persegi panjang. Simbol ini berfungsi untuk merepresentasikan aktivitas pengolahan atau pemrosesan data masukan (input). pada simbol ini, dapat dituliskan berbagai operasi yang diterapkan terhadap input, maupun operasi lainnya yang relevan. Seperti halnya dari simbol input, penulisan operasi bisa dilakukan secara terperinci satu per satu atau secara keseluruhan, tergantung pada kebutuhan dan efisiensi ruang dalam diagram.

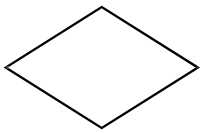
c. Simbol Output



Gambar 2. 10 Simbol Output

Simbol ini dipakai untuk merepresentasikan aktivitas penampilan keluaran (output). Pada simbol ini, semua informasi yang harus diperlihatkan oleh program kepada pengguna dapat dicantumkan. Seperti yang telah terlihat pada dua simbol sebelumnya, penulisan output dapat dilakukan secara terpisah Satu demi satu ataupun dalam bentuk keseluruhan, sesuai dengan kebutuhan penyajian dalam diagram.

d. Simbol Percabangan

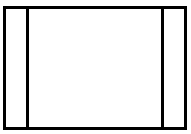


Gambar 2. 11 Simbol Percabangan

Simbol percabangan dalam *flowchart* Ditampilkan dengan bentuk belah ketupat. Simbol ini dipakai untuk menyatakan proses pengambilan keputusan atau percabangan, Merupakan proses memeriksa suatu keadaan atau pernyataan logis. Di dalam simbol ini dituliskan kondisi yang harus dievaluasi. Hasil dari evaluasi tersebut Menyebabkan munculnya dua kemungkinan, yaitu “YES” (benar) atau “NO” (salah). Jika kondisi terpenuhi atau bernilai benar, maka alur proses akan mengikuti jalur yang ditandai dengan label "YES".

Sebaliknya, jika kondisi tidak terpenuhi atau bernilai salah, maka alur akan mengikuti jalur dengan label "NO". Berbeda dengan tiga simbol sebelumnya, penulisan kondisi pada simbol percabangan dilakukan secara spesifik dan satu per satu untuk menjaga kejelasan logika proses.

e. Simbol Prosedur



Gambar 2. 12 Simbol Prosedur

Simbol ini berfungsi sebagai komponen dasar dalam penyusunan suatu program. Setiap prosedur memiliki *flowchart* utamanya sendiri. Oleh karena itu, pada simbol ini cukup dicantumkan nama prosedurnya saja, sebagaimana saat Menjalankan pemanggilan prosedur di dalam program utama (main program). Sama seperti pada simbol percabangan, penulisan nama prosedur dilakukan secara spesifik dan satu demi satu untuk menjaga kejelasan struktur program.

f. Simbol garis alir



Gambar 2. 13 Simbol garis alir

Simbol garis alir atau *flow lines* direpresentasikan dengan tanda panah. Simbol ini berfungsi sebagai penghubung antar langkah dalam *flowchart* serta

menunjukkan arah aliran proses dalam diagram. Arah panah sebaiknya mengikuti pola dari kiri ke kanan atau dari atas ke bawah. Tanda panah ini juga dapat diberi text, terutama apabila diluar dari simbol percabangan untuk memperjelas jalur keputusan yang diambil.

g. Simbol Terminator



Gambar 2. 14 Simbol terminator

Simbol terminator digambarkan dengan bentuk oval, berfungsi untuk menunjukkan titik awal dan akhir dari sebuah *flowchart*. Simbol ini umumnya diberi label "START" untuk menandai permulaan proses dan "STOP" untuk menunjukkan akhir dari alur proses. Dengan demikian, dalam setiap *flowchart* minimal terdapat dua simbol terminator, yaitu sebagai penanda awal (start) dan akhir (stop) dari proses yang digambarkan.

h. Simbol Konektor



Gambar 2. 15 Simbol Konektor

Simbol konektor ini digunakan sebagai penghubung antara satu langkah dengan langkah berikutnya dengan bagian lain dari *flowchart*, baik

dalam satu halaman (*on page*) maupun antar halaman yang berbeda (*off page*). "Konektor *on page* menghubungkan langkah-langkah dalam *flowchart* yang berada pada halaman yang sama, Adapun konektor *off-page* berfungsi mengaitkan langkah-langkah yang terletak pada halaman yang berbeda. Simbol konektor ini biasanya digunakan ketika ruang pada media yang digunakan untuk menggambar *flowchart* tidak cukup untuk menampilkan seluruh diagram secara utuh, sehingga diagram perlu dibagi menjadi beberapa bagian. Setiap pasangan konektor biasanya diberi label identik agar mudah dikenali dan dipasangkan". (Hastuty Hasyim, 2021).

L. White Box

White Box Testing ialah metode pengujian aplikasi atau perangkat lunak dengan cara memeriksa modul-modul secara langsung melalui analisis kode sumber untuk memastikan apakah terdapat kesalahan atau tidak. Jika output dari modul tidak sesuai dengan spesifikasi, maka kode akan dikompilasi ulang dan diuji kembali hingga hasilnya memenuhi harapan. Secara singkat, *White Box* Testing menguji aplikasi dengan melihat kode program secara langsung tanpa memperhatikan antarmuka pengguna.

Dalam melakukan *White Box* Testing, penguji harus memiliki kemampuan pemrograman agar dapat memahami kode sumber yang diuji serta memahami secara menyeluruh cara kerja aplikasi dan komponennya. Berbeda dengan *Black*

Box Testing yang berfokus pada perilaku aplikasi dari sudut pandang pengguna, *White Box* Testing dilakukan oleh penguji dengan peran sebagai pengembang.

Dengan metode ini, penguji dapat mengetahui baris kode mana saja yang dijalankan oleh setiap fungsi, memungkinkan pengujian aliran data, penanganan pengecualian, serta identifikasi kesalahan. Selain itu, penguji juga dapat memeriksa ketergantungan sumber daya, logika, dan validitas kode internal. Oleh karena itu, *White Box* Testing sangat bermanfaat selama tahap pengembangan aplikasi, meskipun dapat diterapkan pada berbagai fase proyek. Metode *White Box Testing* umumnya digunakan dalam pengujian unit, pengujian integrasi, dan pengujian sistem.

M. Bleck Box

Black box testing ialah tahap pengujian yang dilakukan untuk memastikan kelancaran dan keandalan program yang telah dikembangkan.Pengujian ini diperlukan agar alur program yang telah dibuat bebas dari kesalahan.Tes ini penting untuk memastikan tidak ada kesalahan dalam jalannya program yang dibuat.

Menurut Al-Kamal, 2020. "*Black box testing* merupakan metode pengujian yang berfokus pada verifikasi bahwa sistem berfungsi dengan benar dari sudut pandang pengguna. Pengujian ini tidak memeriksa proses internal sistem, melainkan hanya melihat hasil aktual yang terlihat oleh".

Metode *black box testing* penggunaannya mudah sebab hanya memerlukan batas minimum dan maksimum dari data yang diharapkan. Jumlah data uji bisa dihitung berdasarkan banyaknya field input yang akan diuji, aturan input yang harus dipenuhi, serta kasus batas bawah dan batas atas yang relevan. Dengan metode ini, bisa diketahui jika fungsi sistem masih menerima data yang tidak valid, sehingga data yang tersimpan menjadi kurang akurat.

Menurut Ni Made Dwi Febriyanti, dkk, 2021. “Pengujian adalah serangkaian aktivitas yang direncanakan dan sistematis untuk menilai apakah perangkat lunak memenuhi kriteria yang diinginkan. Dalam konteks *black box testing*, pengujian dilakukan berdasarkan spesifikasi fungsional diuji tanpa melihat desain atau kode program, dengan tujuan memastikan bahwa fungsi, input, dan output perangkat lunak sesuai dengan kebutuhan. Pengujian menggunakan metode black box bertujuan untuk menemukan kelemahan sistem sehingga data yang dihasilkan sesuai dengan data yang dimasukkan, serta untuk mencegah kekurangan dan kesalahan sebelum aplikasi digunakan oleh pengguna”.

N. PHP

Menurur Mundzir, 2014. “PHP adalah bahasa skrip utama di sisi server yang dimasukkan dalam kode HTML dan dilakukan pada server, serta dapat juga dipakai untuk mengembangkan aplikasi desktop”. PHP dapat dijalankan pada berbagai jenis web server seperti IIS (Internet Information Services), PWS (Personal Web Server), Apache, dan Xitami. Selain itu, PHP kompatibel dengan banyak sistem

operasi yang populer sekarang ini, termasuk Microsoft Windows (semua versi), Linux, MacOS, dan Solaris.

PHP dapat dijalankan sebagai modul pada web server Apache maupun sebagai binary yang beroperasi menggunakan CGI (Common Gateway Interface). PHP juga memiliki kemampuan untuk mengirim HTTP header, mengelola cookies, mengatur autentikasi, serta mengarahkan pengguna (redirect).

PHP memiliki keunggulan utama yaitu dapat berinteraksi dengan beragam sistem manajemen basis data (DBMS), sehingga memungkinkan pembuatan halaman web yang bersifat dinamis. PHP mendukung koneksi dalam beberapa DBMS populer seperti MySQL, Oracle, Sybase, Microsoft SQL Server, mSQL, Solid, PostgreSQL, Adabas, FilePro, Velocis, dBase, Unix dbm, serta semua database yang memiliki antarmuka ODBC.

PHP mempunyai berbagai keunggulan untuk menjadikannya populer dalam pengembangan web. Beberapa keunggulan tersebut antara lain adalah:

- Gratis dan open source, sehingga dapat digunakan tanpa biaya lisensi.
- Mudah dipelajari, terutama bagi pemula dalam dunia pemrograman.
- Kompatibel dengan berbagai sistem operasi dan server.
- Mendukung berbagai jenis database, terutama MySQL.
- Memiliki komunitas pengguna yang besar**,** sehingga mudah untuk mendapatkan dukungan dan sumber belajar.

- Integrasi mudah dengan HTML, CSS, dan JavaScript, memungkinkan pengembangan antarmuka yang dinamis.

Namun demikian, PHP juga memiliki beberapa kekurangan, seperti:

- Kurangnya struktur yang jelas dalam penulisan kode jika tidak menggunakan framework.
- Keamanan yang rentan jika tidak ditangani dengan benar oleh pengembang.
- Performa dapat menurun jika tidak dioptimalkan dengan baik dalam aplikasi berskala besar.

Variabel dalam PHP dimulai dengan simbol dolar ($) dan tidak memerlukan deklarasi tipe data. Tipe data akan dikenali secara otomatis oleh interpreter PHP. Contoh:

PHP juga mendukung berbagai macam operator seperti:

- Aritmatika: +, -, *, /, %
- Perbandingan: ==, !=, >, <, >=, <=
- Logika: &&, ||, !
- Penugasan**:** =, +=, -=, .=, dan lainnya

PHP menyediakan berbagai struktur pengaturan proses seperti:

- Percabangan**:** if, else, elseif, switch

- Perulangan**:** for, while, do-while, foreach

Keamanan menjadi aspek penting dalam pengembangan aplikasi PHP. Beberapa praktik terbaik untuk menjaga keamanan aplikasi antara lain:

- Melakukan validasi dan sanitasi input dari pengguna untuk mencegah serangan SQL Injection.
- Menggunakan prepared statement saat berinteraksi dengan database.
- Mengenkripsi data sensitif, seperti password, menggunakan fungsi password_hash() dan password_verify().
- Mengelola session dengan benar dan menghindari penyimpanan data sensitif di sisi klien.

Untuk mendukung pengembangan aplikasi yang lebih terstruktur dan efisien, banyak pengembang menggunakan framework PHP. Framework membantu dalam penerapan arsitektur MVC (Model-View-Controller) dan mempermudah penanganan routing, validasi data, autentikasi, dan lain sebagainya. Beberapa framework PHP yang populer di antaranya:

- Laravel: Framework modern dengan sintaks elegan dan fitur lengkap.
- CodeIgniter**:** Ringan dan mudah digunakan, cocok untuk pemula.
- Symfony**:** Framework enterprise dengan modularitas tinggi.
- Yii: Cepat dan efisien untuk pengembangan aplikasi skala menengah hingga besar

O. Javascript

JavaScript adalah bahasa pemrograman level tinggi dengan karakteristik dinamis**,** berbasis prototipe, dan interpreted (diterjemahkan baris per baris oleh browser). JavaScript pertama kali dikembangkan oleh Brendan Eich pada tahun 1995 saat bekerja di Netscape, dan awalnya dikenal dengan nama Mocha, lalu berubah menjadi LiveScript**,** sebelum akhirnya dinamai JavaScript**.**

JavaScript merupakan merupakan salah satu teknologi utama dalam pengembangan web, bersama dengan HTML dan CSS. Jika HTML berfungsi untuk struktur dan CSS untuk tampilan, maka JavaScript digunakan untuk menambahkan interaktivitas pada halaman web.

JavaScript memiliki berbagai fungsi dan kegunaan dalam pengembangan aplikasi web, di antaranya:

- Meningkatkan Interaktivitas: JavaScript memungkinkan pengembang untuk menambahkan fitur interaktif seperti validasi form, dropdown dinamis, slideshow, dan lain-lain.
- Manipulasi DOM (Document Object Model): JavaScript dapat mengakses dan memodifikasi struktur dokumen HTML secara dinamis tanpa harus me-reload halaman.
- Validasi Data: JavaScript dapat digunakan untuk memvalidasi inputan pengguna di sisi klien (client-side validation) sebelum data dikirim ke server.

- Pembuatan Aplikasi Web Dinamis: JavaScript memungkinkan penggunaan AJAX (Asynchronous JavaScript and XML) untuk membuat halaman web yang dinamis dan responsif tanpa harus memuat ulang halaman secara keseluruhan.
- Pengembangan Front-End dan Back-End: Dengan adanya *framework* dan library seperti Angular, React.js, Node.js (untuk back-end), serta Vue.js (untuk front-end), JavaScript kini menjadi bahasa yang digunakan di seluruh stack pengembangan aplikasi web.

JavaScript dijalankan oleh JavaScript Engine yang terdapat pada browser. Misalnya:

- Google Chrome menggunakan V8 Engine
- Mozilla Firefox menggunakan SpiderMonkey
- Safari menggunakan JavaScriptCore (Nitro)

Saat pengguna membuka halaman web, browser akan membaca dan menjalankan skrip JavaScript yang tertanam di dalam HTML. Proses ini terjadi di sisi klien (client-side), sehingga tidak perlu interaksi langsung dengan server untuk menjalankan perintah JavaScript dasar.

P. Xampp

Menurut Setyawan dan Pratiwi, “XAMPP merupakan perangkat lunak komputer yang namanya berasal dari gabungan huruf awal dari Apache, MySQL

atau MariaDB, Perl, dan PHP,. Sedangkan huruf 'X' mencerminkan istilah 'cross-platform' yang menunjukkan bahwa aplikasi ini bisa dijalankan pada berbagai sistem operasi seperti Linux, Windows, dan Mac OS."

Yudhanto & Prasetyo menyatakan bahwa "XAMPP adalah sekumpulan aplikasi perangkat lunak gratis yang sangat dikenal di kalangan pengembang atau programmer, digunakan dalam membangun situs web berbasis PHP dan MySQL".

Sementara itu, menurut Habibi dan rekan-rekannya, "XAMPP adalah perangkat lunak sumber terbuka yang kompatibel dengan berbagai sistem operasi dan merupakan gabungan beberapa aplikasi penting. XAMPP memadukan Apache, MySQL, PHP, dan phpMyAdmin dalam satu paket, sehingga mempermudah pengguna dalam proses instalasi dan konfigurasi ketiga aplikasi tersebut secara otomatis".

Berdasarkan pendapat-pendapat tersebut, dengan demikian, dapat dikatakan bahwa PHP ialah bahasa pemrograman yang dipakai untuk pengembangan situs web berbasis PHP dan MySQL, serta didukung oleh platform seperti XAMPP yang mempermudah pengelolaan dan pengembangan website.

Q. Mysql (My Structured Query Language)

Sistem manajemen basis data ini bersifat open source dan digunakan untuk menyimpan serta mengatur data. Menurut Anhar (2010:21), "MySQL (My Structured Query Language) ialah sistem manajemen basis data berbasis SQL

(Database Management System/DBMS), yang merupakan salah satu dari banyak jenis DBMS lainnya seperti Oracle, MS SQL, PostgreSQL, dan sebagainya". Kadir (2008:2) juga menyatakan bahwa "MySQL adalah perangkat lunak yang termasuk dalam kategori DBMS dan bersifat open source". Sementara itu, menurut Arief (2011:151), "MySQL merupakan suatu database server yang populer dan sering difungsikan dalam mengembangkan aplikasi berbasis web yang membutuhkan database untuk sumber dan pengelola datanya".

Dari pendapat-pendapat tersebut dapat diartikan bahwa MySQL adalah perangkat lunak open source yang sering dimanfaatkan dalam pengembangan aplikasi web dan digunakan dalam proses pengelolaan dan penyimpanan data.

MySQL banyak digunakan karena memiliki sejumlah keunggulan, di antaranya:

1. Bersifat Open source, sehingga dapat dipakai tanpa biaya dan dimodifikasi sesuai dengan kebutuhan pengguna.
2. Kinerja tinggi, terutama untuk aplikasi web dengan beban ringan hingga menengah.
3. Mendukung integrasi lintas platform, seperti Windows, Linux, Unix, dan macOS.
4. Kompatibel dengan berbagai bahasa pemrograman, seperti Java, PHP, C++, dan Python
5. Mendukung transaksi dan integritas data, terutama pada versi MySQL dengan mesin penyimpanan InnoDB.

6. Tersedia dokumentasi dan komunitas global yang aktif, memudahkan proses pembelajaran dan pemecahan masalah.

MySQL merupakan sistem basis data relasional**,** artinya data disimpan tersusun dalam struktur tabel yang saling berkaitan. Setiap tabel terdiri dari kolom (field) dan baris (record). Hubungan antar tabel dapat didefinisikan melalui kunci primer dan kunci asing.

Contoh hubungan antar tabel:

- Tabel mahasiswa memiliki id_mahasiswa sebagai primary key.
- Tabel nilai memiliki id_mahasiswa sebagai foreign key yang merujuk pada tabel mahasiswa.

Beberapa komponen dasar dalam struktur database MySQL antara lain:

- Database: kumpulan data yang saling berhubungan.
- Table: struktur utama penyimpanan data, terdiri atas baris dan kolom.
- Field: kolom dalam tabel, menyimpan jenis data tertentu.
- Record: baris dalam tabel, menyimpan satu unit data.

Operasi dasar yang umum dilakukan dalam MySQL meliputi:

- CREATE: membuat database atau tabel
- **I**NSERT: tambah data
- SELECT: tampil data

- UPDATE: memperbarui data
- DELETE: menghapus data

MySQL mendukung berbagai macam tipe data, di antaranya:

1. Numerik: INT, FLOAT, DOUBLE, DECIMAL
2. String: VARCHAR, TEXT, CHAR
3. Tanggal/Waktu: DATE, DATETIME, TIME, TIMESTAMP
4. Boolean: BOOLEAN atau TINYINT(1)

Dalam desain basis data relasional, proses normalisasi diperlukan untuk menghindari redundansi data dan memastikan integritas data. Normalisasi dilakukan melalui beberapa tahapan bentuk normal (normal form), antara lain:

- Bentuk Normal Pertama (1NF)
- Bentuk Normal Kedua (2NF)
- Bentuk Normal Ketiga (3NF)

Tujuan dari normalisasi adalah memastikan bahwa setiap tabel menyimpan data secara efisien dan meminimalkan pengulangan data yang tidak perlu.

Dalam penggunaan MySQL, aspek keamanan perlu diperhatikan untuk mencegah akses tidak sah dan serangan seperti SQL Injection. Beberapa langkah keamanan yang dapat diterapkan antara lain:

- Memberikan hak akses (privilege) secara selektif pada pengguna database.

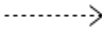
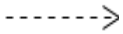
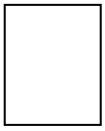
- Menggunakan prepared statement untuk mencegah SQL injection.
- Melakukan backup database secara berkala.
- Menggunakan password yang kuat untuk akun MySQL.

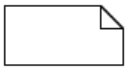
R. UML (Unified Modeling Language)

Menurut Nugroho (2009:4), "UML (Unified Modeling Language) merupakan hasil hubungan dari berbagai metode seperti Booch, OMT (Object Modeling Technique), OOSE (Object Oriented Software Engineering), dan terdapat beberapa pendekatan lainnya. UML adalah metodologi yang saat ini paling banyak digunakan untuk kegiatan analisis dan perancangan sistem berbasis pendekatan berorientasi objek, yang menyesuaikan dengan semakin luasnya penggunaan bahasa pemrograman Berdasarkan konsep objek (OOP)". Dalam UML terdapat sejumlah jenis diagram, antara lain:

a. *UseCaseDiagram*

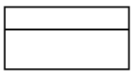
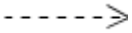
Tabel 2. 1 Simbol-simbol use case

NO	GAMBAR	NAMA	KETERANGAN
1		*Actor*	Identifikasi himpunan Peran yang dijalankan oleh user saat berinteraksi dengan use case
2		*Dependency*	Suatu relasi di mana perubahan pada elemen yang bersifat mandiri (*independen*) akan berdampak pada elemen lain yang bergantung padanya, yaitu elemen yang tidak mandiri (*dependen*).
3		*Generalization*	Relasi di mana objek turunan (d*escendent*) mewarisi perilaku dan struktur data dari objek induknya (*ancestor*)
4		*Include*	Mengspesifikasikan bahwa *use case* sumber secara *eksplisit*.
5		*Extend*	Mengspesifikasikan Use case target memperluas fungsi dari use case sumber pada titik tertentu yang telah ditentu
6		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		*System*	Identifikasi paket yang menampilkan sistem secara terbatas.
8		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor

NO	GAMBAR	NAMA	KETERANGAN
9		*Collaboration*	Kombinasi aturan dan elemen lain yang berfungsi bersama untuk memberikan hasil atau perilaku yang melampaui akumulasi bagian-bagiannya.
10		*Note*	Entitas fisik yang hadir selama proses eksekusi aplikasi dan menggambarkan suatu sumber daya komputasi

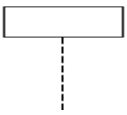
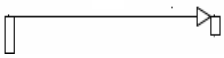
b. Class Diagram

Tabel 2. 2 Simbol-Simbol Class Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		*Generalization*	Relasi di mana objek turunan (*descendent*) mewarisi perilaku dan struktur data dari objek induk (*ancestor*)
2		*Nary Association*	Usaha menghindari hubungan dengan lebih dari dua objek
3		*Class*	Kumpulan objek yang memiliki kesamaan dalam atribut dan operasi
4		*Collaboration*	Deskripsi tentang langkah-langkah sistem yang menghasilkan output terukur bagi aktor tertentu.
5		*Realization*	Aktivitas yang benar-benar dilaksanakan oleh suatu objek
6		*Dependency*	Relasi di mana perubahan pada elemen yang bersifat mandiri akan memengaruhi elemen lain yang tidak mandiri dan bergantung padanya

c. *Sequence* Diagram

Tabel 2. 3 *Sequence Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*LifeLine*	Objek *entity*, antarmuka yang saling berinteraksi.
2		*Message*	Spesifikasi komunikasi antar objek yang berisi informasi mengenai aktivitas yang berlangsung
3		*Message*	Penjabaran komunikasi antar objek yang mencakup informasi tentang kegiatan yang sedang berlangsung.

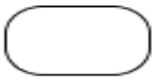
d. *State Chart* Diagram

Tabel 2. 4 State Chart Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		*State*	Nilai atribut dan nilai hubungan pada suatu waktu tertentu yang dimiliki oleh sebuah objek.
2		*Initial Pseudo State*	Cara objek dibuat atau dimulai
3		*Final State*	Proses pembentukan dan penghapusan objek
4		*Transition*	Suatu peristiwa yang menyebabkan perubahan state objek dengan memperbarui satu atau lebih nilai atributnya.
5		*Association*	Faktor yang menghubungkan objek satu dengan objek lainnya.
6		*Node*	Bagian fisik yang eksis saat aplikasi aktif dan mencerminkan suatu sumber daya komputasi.

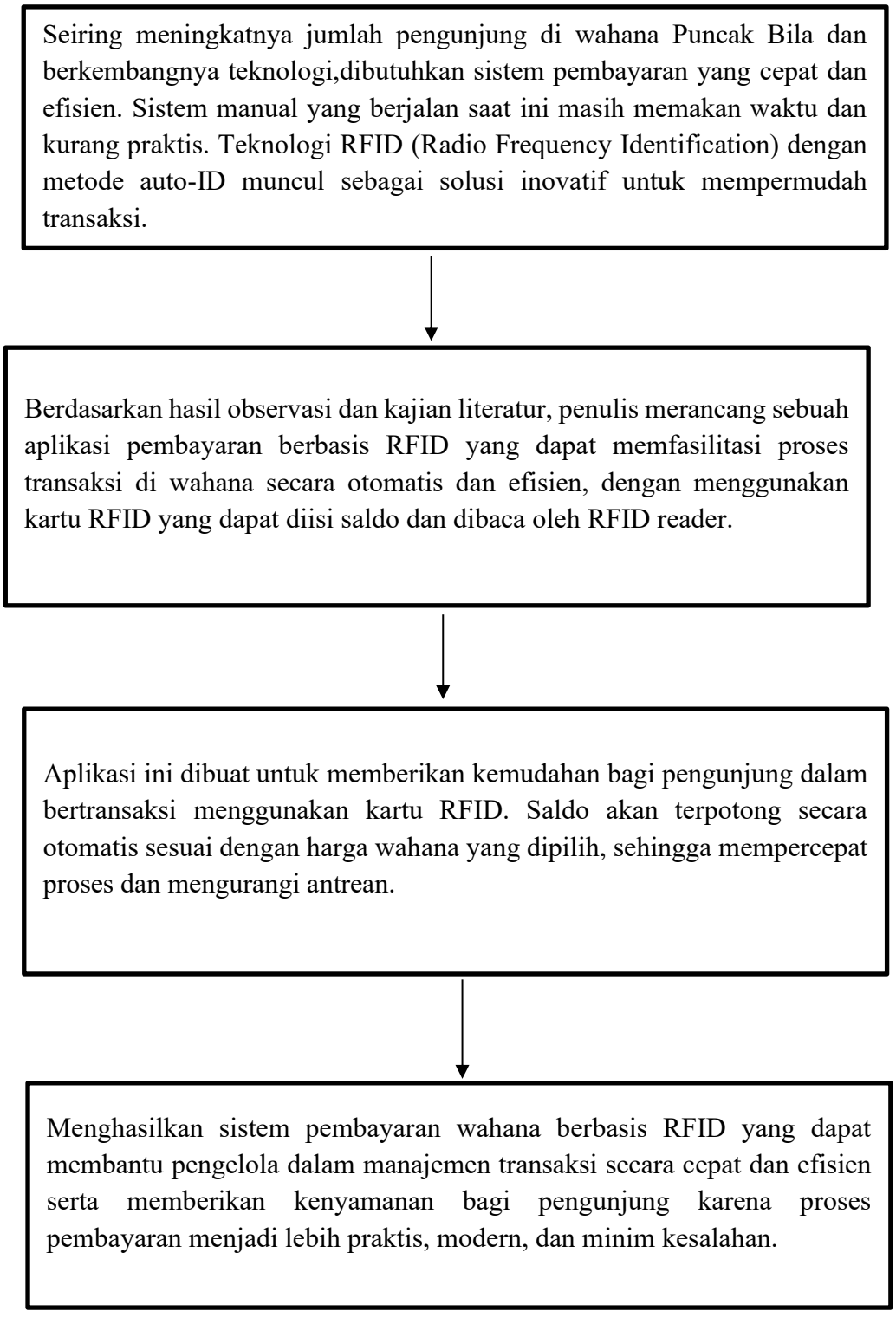
e. *Actifity Diagram*

Tabel 2. 5Simbol-Simbol *Activity Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*Activity*	Menggambarkan bagaimana kelas-kelas antarmuka saling berkomunikasi.
2		*Action*	Kondisi sistem yang menggambarkan pelaksanaan suatu aksi.
3		*Initial Node*	Proses permulaan objek.
4		*Activity Final Node*	Bagaimana objek diinisialisasi dan kemudian dihapus.
5		*Fork Node*	Aliran yang pada titik tertentu bercabang menjadi beberapa aliran.

S. Kerangka Pikir

Kerangka pikir dari penelitian ini yakni:



BAB III

METODE PENELITIAN

A. Jenis Penelitian

Jenis penelitian yang diterapkan ialah penelitian deskriptif, bertujuan menggambarkan fenomena sebagaimana adanya. Dalam penyusunan proposal ini, metode deskriptif digunakan untuk memaparkan fakta dan informasi secara sistematis, nyata, dan tepat.

B. Lokasi dan Waktu Penelitian

1. Lokasi

   Lokasi penelitian dilakukan di wisata punca bila kabupaten sidrap.

2. Waktu Penelitian

   Waktu yang digunakan dalam melakukan penelitian ±3 bulan

C. Metode Pegumpulan Data

1. *Observasi* merupakan teknik untuk memperoleh data dengan cara mengenali informasi serta situasi awal objek penelitian.
2. *Dokumentasi*, yakni pengumpulan data dalam bentuk catatan tertulis atau dokumen.
3. Wawancara, yaitu dengan melakukan wawancara terhadap tempat penelitian.

D. Alat dan Bahan

Adapun alat dan bahan sebagai berikut :

1. Perangkat Keras (*Hardware*)

   Hardware yang digunakan untuk membangun aplikasi ini adalah Laptop Asus dengan spesifikasi sebagai berikut:

   - Laptop : Aspire A515-45
   - Processor : AMD Ryzen 5 5500U with Radeon Graphics 2.10 GHz
   - RAM : 8GB
   - SSD : 512GB
   - RFID : Tag,Reader
   - Kabel : USB, Jumper
   - Rdm 6300
   - Esp 32
   - Buzzer

2. Perangkat lunak (*Sofware*)

   Sofware yang digunakan untuk membangun aplikasi ini adalahsebagai berikut:

   - Php
   - Javasript
   - Xampp
   - Windows 11

E. Objek Penelitian

Objek penelitian yang dilakukan adalah membuat sistem pembayaran non tunai dengan mengunakan RFID sebagai alat untuk melakukan teransaksi untuk memasuki suatu wahana permainan di wisata puncak bila.

F. Tahapan Penelitian

Tahapan metode yang diterapkan pada penelitian ini terdiri dari tahap tahap persiapan, pengumpulan data, analisis sistem, desain sistem, pengujian, serta penerapan. Penjelasan dari setiap tahapan tersebut yakni sebagai berikut:

1. Pegumpulan data

   Melakukan pengumpulan data dan informasi dalam rangka dijadikan acuan untuk membangun aplikasi yang di rancang
2. Analisa sistem

   Megidentifikasi dan menganali msalah yang ada, kemudian mencari solusi atau alternatif
3. Desain sistem

   Setelah masalah di tentukan dan analisis data sudah di lakukan maka perlu di lakukan pembuatan dan merancang sistem berdasarkan permasalahan yang telah diidentifikasi.
4. Pegujian program

   Setelah proses penulisan kode program sudah proses selesai, dilakukan pengujian sistem dengan metode black box.
5. Implementasi sistem

Tahapan ini diaplikasikan untuk menerapkan serta pegujian sistem kedalam kondisi sebenarnya agar dapat diketahui kekurangan dan kelebihannya.

G. Jenis Data

Jenis data yang digunakan untuk penelitian ini terdiri dari data sekunder dan data primer yang memiliki keterkaitan pada topik yang dikaji.

1. Data primer

   Data primer merupakan informasi yang memperoleh secara langsung dari sumber utama di lapangan, seperti hasil wawancara dengan pihak-pihak yang berkaitan langsung dengan penelitian ini.

2. Data Sekunder

   Data sekunder ialah informasi yang didapatkan secara tidak langsung dari objek penelitian. Data tersebut diperoleh melalui sumber yang telah tersedia sebelumnya dan dikumpulkan oleh pihak lain, baik melalui metode komersial maupun non-komersial.

H. Desain Sistem

Pembuatan sistem pembayaran non tunai disajikan menggunakan UML. Flowchart dan Use Case diagram yang digunakan untuk menggambarkan keadaan lingkungan yang ada didalam sistem.

1. Sistem yang berjalan

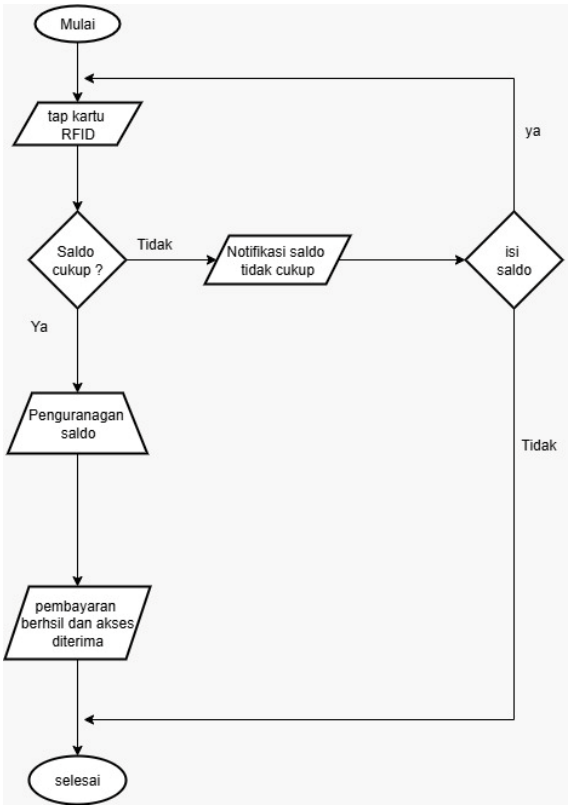
   Adapun sistem yang berjalan saat ini menggunakan use case yaitu :

   - Pergi kekasir untuk antri dalam melakukan pembelian tiket untuk wahana yang akan di masuki
   - Setelah mendapatkan tiket wahana pergi ke lokasi wahana berada
   - lalu berikan tiket kepada pengawas wahana
   - stelah tiket sudah di berikan mainkan wahananya

2. Sistem yang diusulkan

   a. Flowchart

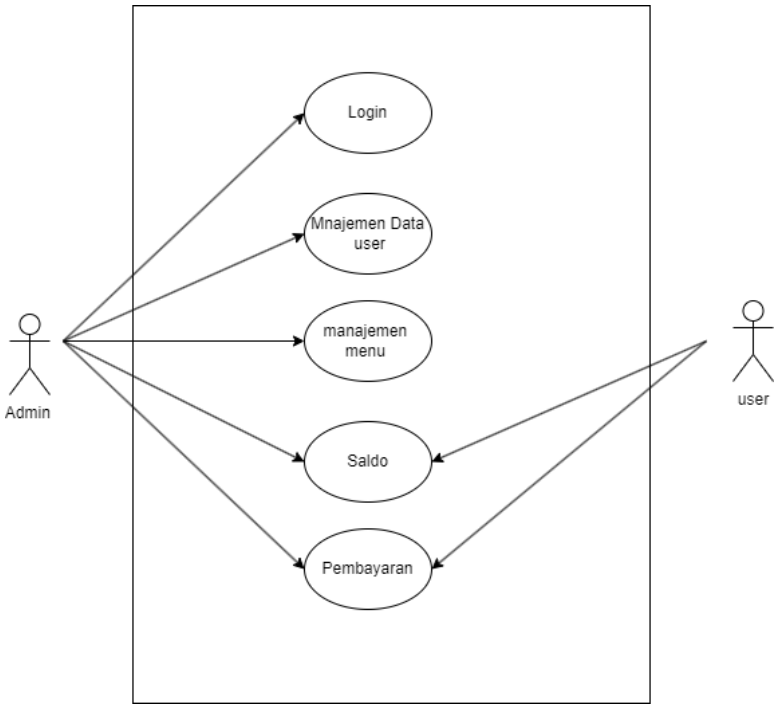
   Adapun sistem yang diusulkan mengunakan flowchart saat ini yaitu :



Gambar 3. 1flowchart sistem yang diusulkan

b. Use Case

Adapun sistem yang diusulkan mengunakan use case saat ini yaitu :



Gambar 3. 2Diagram *Use Case*

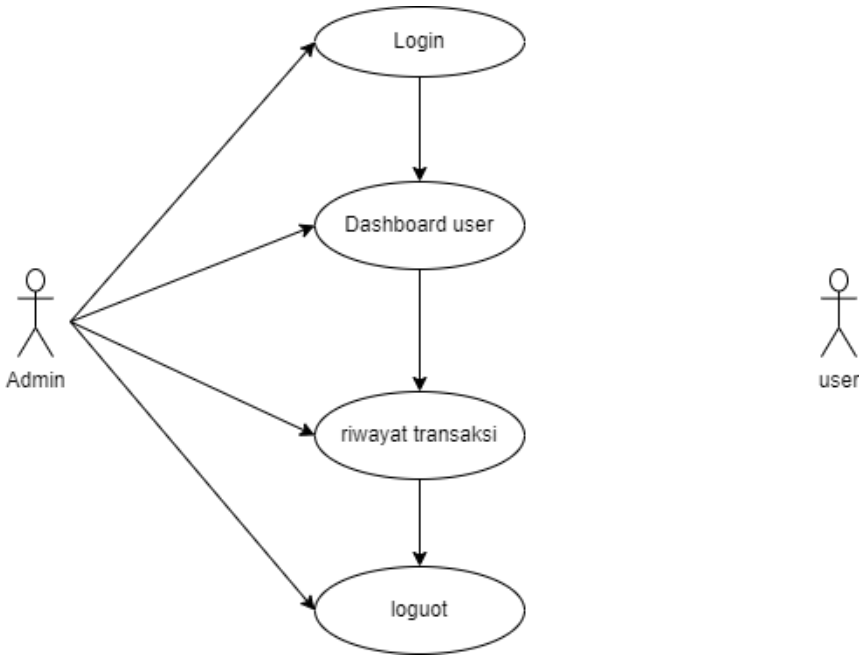
Ketika pengunjung ingin memasuki wahana terlebih dahulu melakukan pendaftaran kartu RFID di admin dan juga melakukan penambahan saldo, jika selesai maka pegunjung dapat memasuki wahana yang di ingingan dengan melakukan pembayaran menggunakan kartu RFID yang telah terdatar dan terdapat saldo yang cukup untuk memasuki suatu wahana dengan cara men tap kartu di RFID reader.

BAB IV

HASIL DAN PEMBAHASAN

A. Analisa aliran data

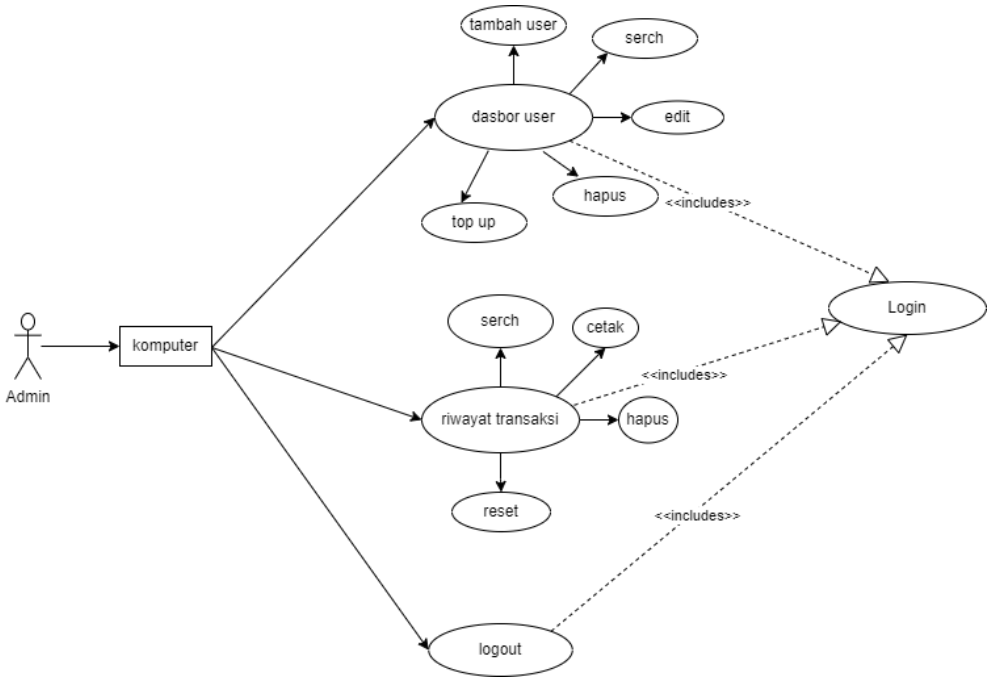
1. *Use case* diagram



Gambar 4. 1 Diagram *Use Case*

2. Use case implementasi desain jaringan

   a. Aktor Admin

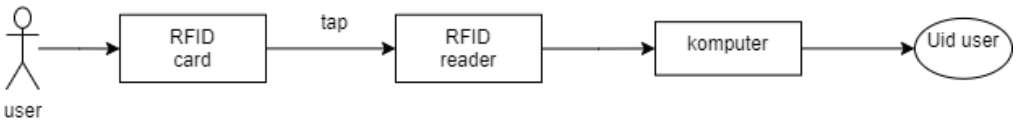


Gambar 4 .2 *use case* admin

Tabel 4. 1 Penjelasan *use case* admin

Nama *use case*	Deskrip *use case*
Login	Admin terlebih dahulu login ke aplikasi agar bisa di gunakan
Dasboard user	Admin bisa mengedit, menambah, dan menghapus data pegunjung dan juga bisa topup
Laporan transaksi	Admin dapat melihat pengunjung yang memasuki wahana dan bisa juga menghapus, mereset, mencari uid user, dan mencetak laporan transaksi
Logout	Jika admin sudah selesai menggunakan aplikasi bisa langsug keluar

b. Aktor pegunjung



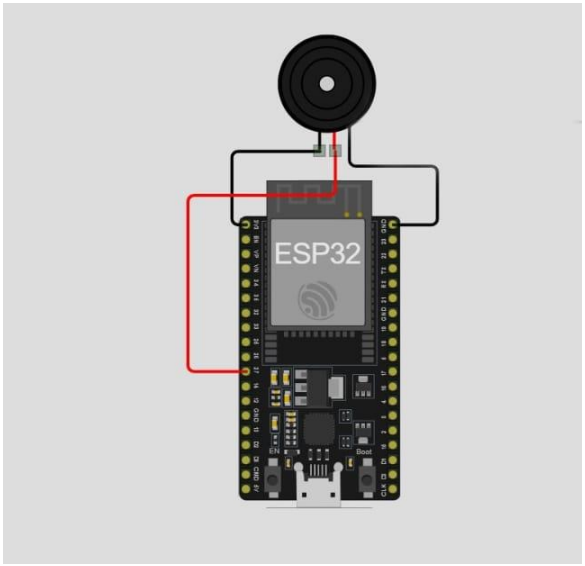
Gambar 4 3 *use case* pengunjung

Tabel 4. 2 penjelasan use case user

Nama *use case*	Deskrip *use case*
Login	User tidak dapat login ke aplikasi
Dasboard user	User tidak bisa mengedit, menambah, dan menghapus data user
Laporan transaksi	User tidak dapat melihat pengunjung yang telah memasuki wahana dan tidak bisa juga menghapus, mereset, mencari uid user, dan mencetak laporan transaksi
Logout	User tidak dapat melakuka logout

B. Rancangan perangkat keras (*Hardware*)

Pembuatan perangkat keras (hardware) merupakan terbentuknya suatu system atau rangkaian elektronika. Rancangan ini akan menjelaskan alat yang digunakan serta pemilihan komponen yang tepat. pada rancangan aplikasi pembayaran wahana berbasis RFID memiliki *hardware* yaitu; Rdm 6300, Esp 32,buzeer,RFID Reader.

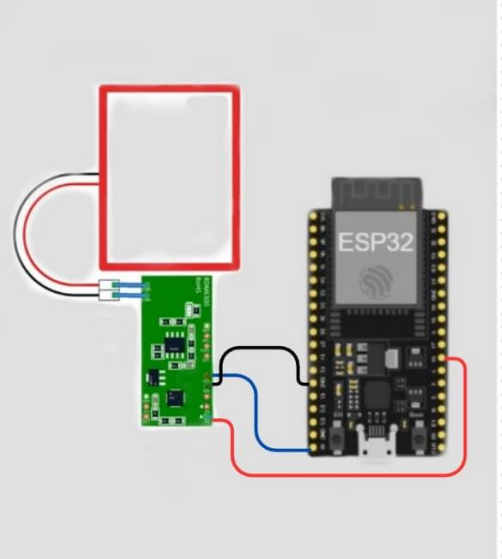


Gambar 4 4 rancangan buzzer ke esp 32

Pada gambar di atas menunjukkan rangkain alat yang digunakan di aplikasi pembayaran wahan berbasis RFID.

Esp 32	Buzzer
3v3	-
Gnd	Gnd
27	Pwm

Pada rangkaian gambar di atas menjelaskan alur *hardware* yang menghubungkan berbagai pin. Dapat dilihat dari esp 32 terhubung dengan buzzer dengan pin, 3v3 ke tengah(-), Gnd ke Gnd dan 27 ke pwm(s).



Gambar 4 5 rancangan rdm 6300 ke esp 32

Pada gambar di atas menunjukkan rangkain alat yang digunakan di aplikasi pembayaran wahan berbasis RFID.

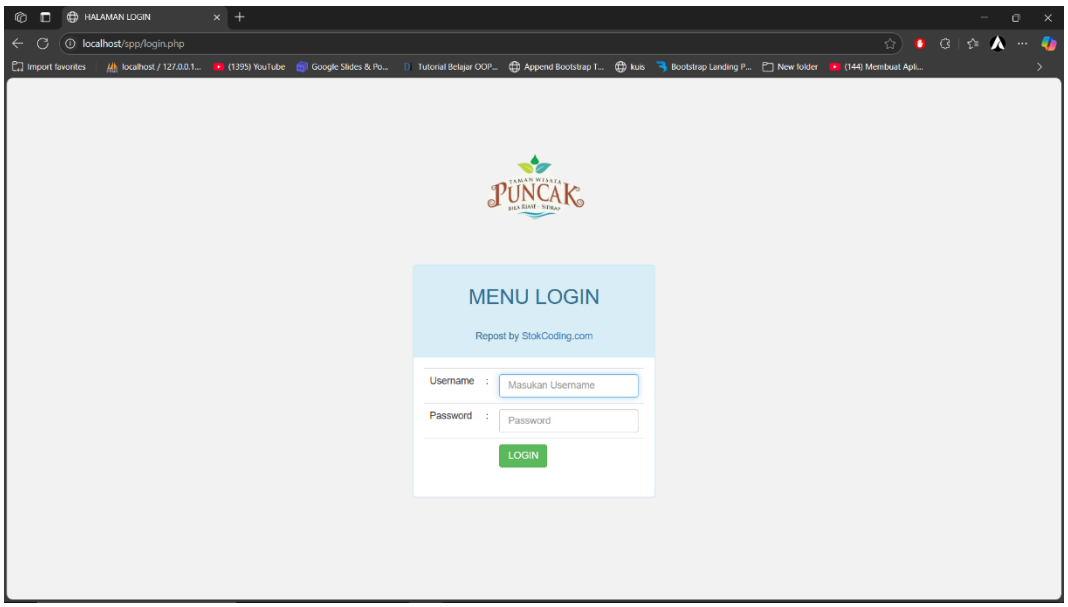
Esp 32	Rdm 6300
5v	Vcc (5v)
Gnd	Gnd
16	Tx

Pada rangkaian gambar di atas menjelaskan alur *hardware* yang menghubungkan berbagai pin. Dapat dilihat dari esp 32 terhubung dengan rdm 6300 dengan pin, 5v ke vcc (5v), Gnd ke Gnd dan 16 ke Tx.

C. Implementasi

1. Tampilan awal

Halaman menu Login merupakan tampilan awal pada saat admin ingin masuk ke aplikasi pembyaran wahana berbasis RFID.



Gambar 4. 6 Tampilan Menu login

Halaman ini menampilkan form login dengan logo di atas, judul "Menu Login", kolom username dan password, serta tombol hijau "Login" untuk masuk ke sistem aplikasi pembayaran wahana.

a. Inisialisasi dan Cek Login

```
session_start();
if (isset ($_SESSION['login']) ) {
    header ('Location: index.php');
}
```

- session_start() digunakan untuk memulai sesi PHP agar dapat menggunakan variabel $_SESSION.
- if (isset($_SESSION['login'])) akan mengecek apakah pengguna sudah login. Jika ya, maka pengguna langsung diarahkan ke halaman index.php

b. Koneksi database

```
include 'koneksi.php';
```

- File koneksi.php berisi konfigurasi koneksi ke database (biasanya menyimpan host, username, password, dan nama database).
- Ini memungkinkan file login untuk berkomunikasi dengan database

c. Logika login saat form di kirim

```
if ($_SERVER['REQUEST_METHOD'] == 'POST' ) {

    $user  = $_POST['username'];
    $pass  = $_POST['password'];
    $p     = hash('sha1', $pass);
```

- Mengecek apakah form dikirim menggunakan metode POST (yaitu saat pengguna menekan tombol LOGIN)
- Mengambil input username dan password dari form.
- Password di-*hash* menggunakan algoritma SHA-1 sebelum dibandingkan dengan data di database

d. Validasi imput kosong

```
if ( $user == "" || $p == ""){
    $error = true;
}
```

- Jika username atau password kosong, maka $error akan bernilai true dan ditampilkan pesan peringatan

e. Pemeriksaan data ke data base

```
$data = $konek -> query ("SELECT * FROM admin WHERE username ='".$user."' AND password = '".$p."'");
$dt = mysqli_num_rows($data);
$dta = mysqli_fetch_Assoc($data);
```

- Query ke tabel admin untuk mencari pengguna dengan username dan password yang sesuai.
- $dt adalah jumlah baris hasil pencarian (harus lebih dari 0 jika ditemukan).
- $dta berisi data admin yang ditemukan.

f. Autentikasi berhasil

```
if ($dt > 0) {
    session_start();
    $_SESSION['login']    = TRUE;
    $_SESSION['username'] = $dta['username'];
    $_SESSION['id']       = $dta['idadmin'];
    header('Location: index.php');
}
```

- Jika pengguna ditemukan, maka sesi login dibuat dan informasi penting seperti username dan id disimpan di $_SESSION.
- Setelah login berhasil, pengguna diarahkan ke index.php.

g. Autentikasi gagal

```
else{
    echo "
    <script>
    alert ('username atau password anda salah');
    document.location.href = 'login.php';
```

```
    </script>
    ";
}
```

- Jika data tidak ditemukan di database, ditampilkan *alert* menggunakan JavaScript dan diarahkan kembali ke halaman login

h. Bagian HTML-tampilan login

Bagian ini mencakup struktur HTML dan CSS untuk menampilkan form login:

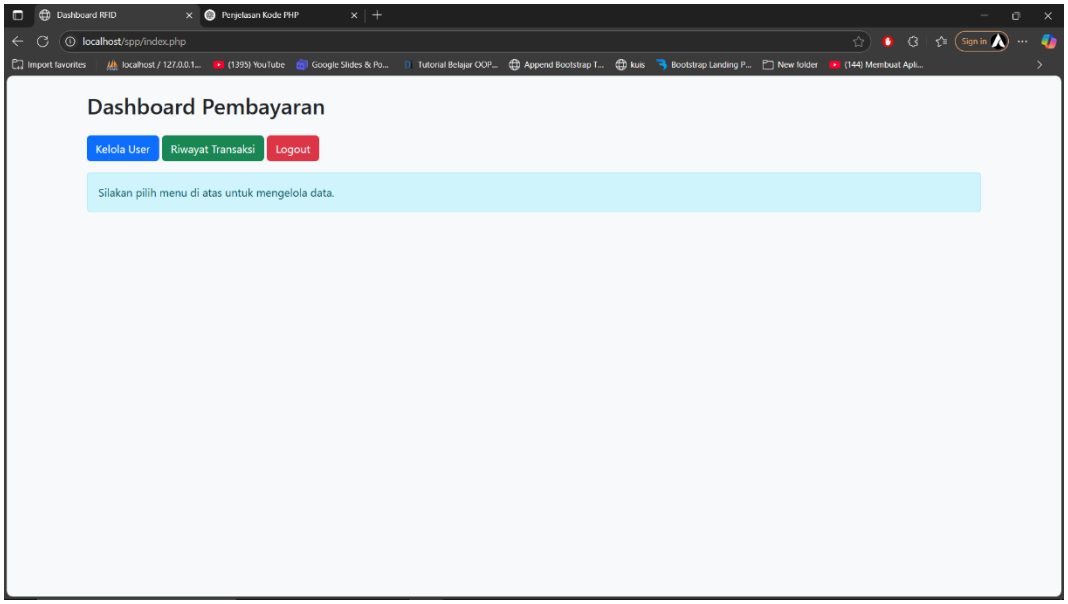
- Masukkan username dan password
- Tombol login
- Tampilan responsif menggunakan Bootstrap
- Pesan peringatan jika ada kesalahan input (menggunakan `$error`)

```
<?php if (isset($error) ) :  ?>
<div class="alert alert-warning">
<span><b>Peringatan!!</b></span>
<h class="text-center">masukan data dengan benar</h>
</div>
<?php endif;  ?>
```

- Blok ini menampilkan pesan jika $error diatur sebelumnya karena input kosong

2. Halama Home

Halaman Home merupakan tampilan awal saat sudah login di aplikasi pembayaran wahana berbasis RFID. Yang terhubung dengan halaman-halaman lain seperti : kelola user dan riwayat transaksi.



Gambar 4 7 Tampilan Menu *Home*

Halaman ini adalah Dashboard Pembayaran setelah *login*. Di bagian atas ada tiga tombol utama: Kelola *User* untuk mengatur data pengguna**,** Riwayat Transaksi untuk melihat catatan transaksi, dan *Logout* untuk keluar dari sistem. Di bawahnya terdapat kotak informasi berwarna biru muda yang berisi pesan singkat agar pengguna memilih menu di atas untuk mengelola data. Tampilan sederhana ini dibuat supaya pengguna langsung fokus pada pilihan menu utama.

a. Bagian php

```
<?php
session_start();
if (!isset($_SESSION['username'])) {
    header("Location: login.php");
    exit;
```

```
}
include 'koneksi.php';
?>
```

ession_start();

Memulai session untuk mengecek apakah user sudah login.

if (!isset($_SESSION['username'])) { ... }

Jika session 'username' belum ada (belum login)**,** maka:

- Redirect user ke login.php menggunakan header().
- Menghentikan script dengan exit; agar kode selanjutnya tidak dijalankan.

include 'koneksi.php';

Menghubungkan file koneksi.php yang biasanya berisi koneksi ke database

b. Bagian html

```
<!DOCTYPE html>
<html lang="id">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Dashboard RFID</title>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.mi
n.css">
</head>
<body class="bg-light">
    ...
</body>
</html>
```

- <!DOCTYPE html> ... </html>

  Struktur standar HTML untuk halaman web.

<meta charset="UTF-8">

Mendukung karakter UTF-8 (supaya teks Indonesia tampil benar).

<meta name="viewport" content="width=device-width, initialscale=1.0">

Membuat tampilan responsif di berbagai perangkat.

<title>Dashboard RFID</title>

Judul halaman browser: *Dashboard RFID*.

Bootstrap CSS digunakan agar tampilan rapi dan modern tanpa harus membuat CSS manual.

c. Baagian body

```
<body class="bg-light">
        <div class="container py-4">
                <h2 class="mb-4">Dashboard Pembayaran</h2>

                <div class="mb-3">
                        <a href="user/list.php" class="btn btn-primary">Kelola User</a>
                        <a href="transaksi/list.php" class="btn btn-success">Riwayat Transaksi</a>
                        <a href="logout.php" class="btn btn-danger">Logout</a>
                </div>

                <div class="alert alert-info">
                        Silakan pilih menu di atas untuk mengelola data.
                </div>
```

```
        </div>
</body>
```

<body class="bg-light">

Memberi latar belakang abu-abu terang.

Bootstrap container dengan padding atas dan bawah.

<h2> Dashboard Pembayaran </h2>

Judul halaman di dalam dashboard.

Tiga tombol menu:

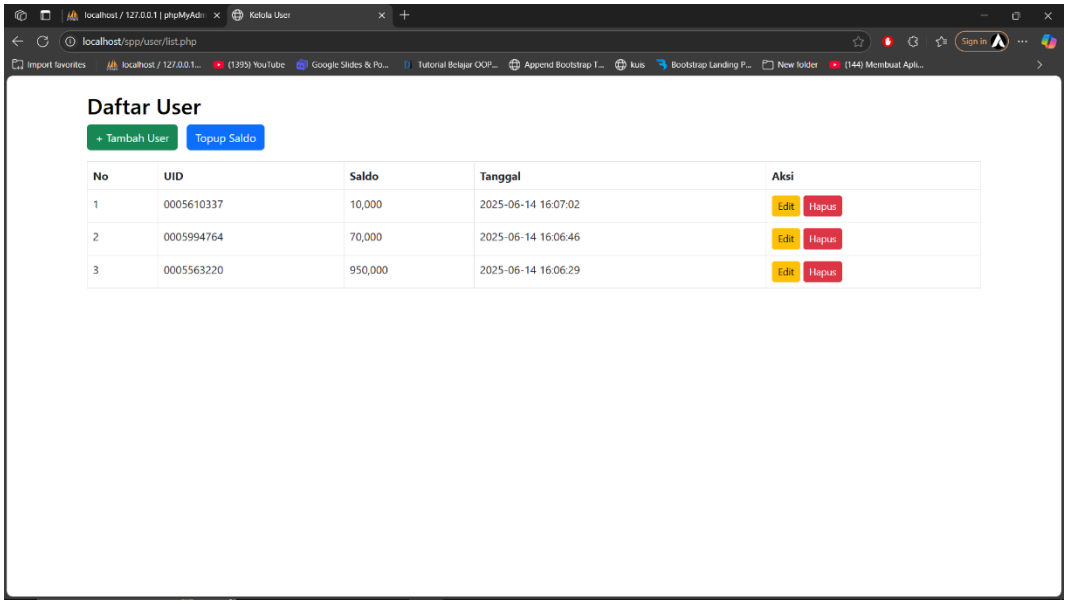
- Kelola User (menuju user/list.php)
- Riwayat Transaksi (menuju transaksi/list.php)
- Logout (menuju logout.php)

Semua tombol menggunakan Bootstrap button agar rapi:

- btn-primary = biru
- btn-success = hijau
- btn-danger = merah

3. Daftar user

Pada halaman daftar user terdapat data pegunjung yang sudah terdaftar seperti; Uid, saldo dan aksi, pada halam ini juga bisa digunakan untuk tambah pegunjung dan tambah saldo.



Gambar 4. 8 Tampilan data pegunjung

Halaman ini adalah Daftar *User*. Di bagian atas ada dua tombol, Tambah *User* untuk mendaftarkan pengguna baru dan *Topup* Saldo untuk menambah saldo kartu. Tabel di bawahnya menampilkan data *user* yang sudah terdaftar, meliputi nomor urut, UID kartu, saldo, tanggal pendaftaran atau *update*, serta kolom Aksi dengan tombol *Edit* untuk mengubah data dan Hapus untuk menghapus *user*. Tampilan sederhana ini memudahkan admin dalam mengelola data pengguna RFID

a. Session dan akses login

```
session_start();
if (!isset($_SESSION['username'])) {
    header("Location: ../login.php");
    exit;
}
```

- Memastikan hanya user yang sudah login yang dapat mengakses halaman ini.
- Jika belum login, akan redirect ke login.php.

b. Koneksi ke database

```
require_once '../koneksi.php';
```

- Menghubungkan ke database menggunakan file koneksi.php (lokasi di folder induk).

c. pagination

```
$limit = 10; // data per halaman
$page = isset($_GET['page']) ? (int)$_GET['page'] : 1;
$page = max($page, 1);
$offset = ($page - 1) * $limit;
```

- Pagination membatasi tampilan 10 data per halaman**.**
- Jika URL memiliki parameter ?page=2, maka akan menampilkan halaman kedua.
- offset digunakan untuk SQL agar data tidak tampil semua sekaligus.

d. Pencarian data uid

```
$cari = $_GET['cari'] ?? '';
```

- Mengambil input pencarian UID dari URL (?cari=xxx), default kosong jika tidak ada

e. Hitung total data untuk pagination

```
$sqlCount = "SELECT COUNT(*) AS total FROM users WHERE uid LIKE
'%$cari%'";
$total = mysqli_fetch_assoc(mysqli_query($konek, $sqlCount))['total'];
$totalPages = ceil($total / $limit);
```

- Menghitung jumlah total data user yang cocok dengan pencarian.
- Menghitung berapa total halaman (totalPages) berdasarkan jumlah data.

f. Ambil data user

```
$sql = "SELECT * FROM users WHERE uid LIKE '%$cari%' ORDER BY iduser DESC
LIMIT $limit OFFSET $offset";
$result = mysqli_query($konek, $sql);
```

Mengambil data user berdasarkan:

- Filter UID jika ada pencarian.
- Diurutkan berdasarkan iduser terbaru.
- Hanya mengambil sejumlah $limit data dengan OFFSET sesuai halaman aktif.

g. Header dan menu

```
<a href="../index.php">Daftar User</a>
<a href="add.php" class="btn btn-success">+ Tambah User</a>
<a href="topup.php" class="btn btn-primary">Topup Saldo</a>
<a href="../logout.php" class="btn btn-danger">Logout</a>
```

- Navigasi untuk kembali ke dashboard, tambah user, top-up saldo, dan logout.

h. From pencarian

```
<form class="d-flex" method="get">
  <input type="text" name="cari" placeholder="Cari UID..." value="<?=
htmlspecialchars($cari) ?>">
  <button type="submit">Cari</button>
</form>
```

- Untuk mencari UID dengan metode GET (URL).
- Menampilkan kembali keyword pencarian yang terakhir dicari

i. Tabel data useer

```
<table class="table table-bordered">
  <thead>
    <tr>
      <th>No</th>
      <th>UID</th>
      <th>Saldo</th>
      <th>Tanggal</th>
      <th>Aksi</th>
    </tr>
  </thead>
  <tbody>
    <!-- Loop data user -->
  </tbody>
</table>
```

Menampilkan No, UID, Saldo, Tanggal, Aksi Edit/Hapus untuk setiap user.

Menggunakan:

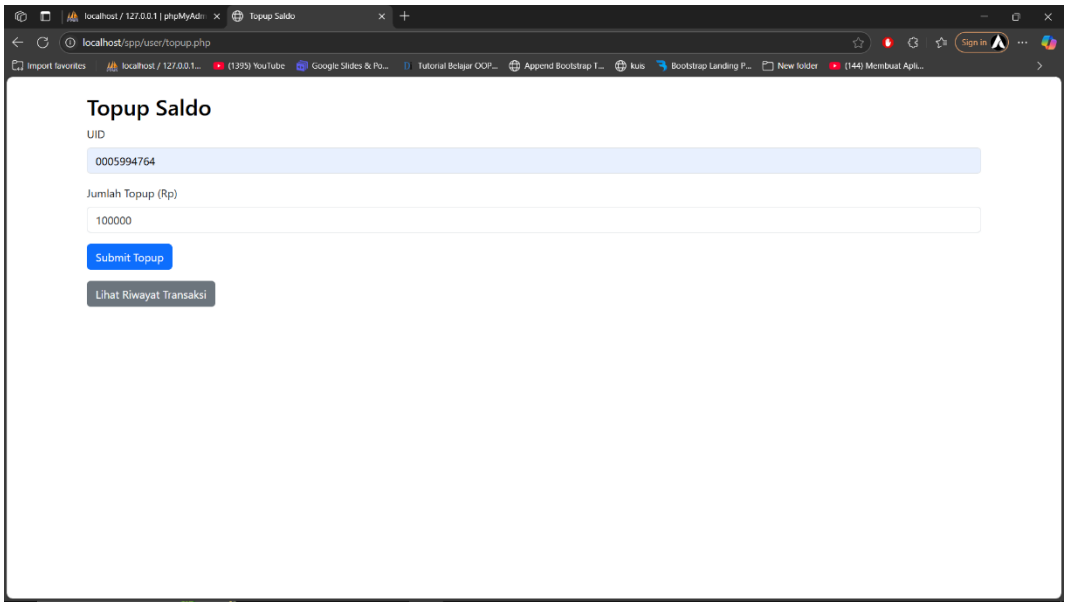
- <?= $no++ ?> untuk nomor urut sesuai pagination.
- <?= $row['uid'] ?>, <?= number_format($row['saldo']) ?>, dan <?= $row['tgl'] ?> untuk data user.
- Tombol Edit menuju edit.php?id=uid.
- Tombol Hapus menuju delete.php?id=iduser dengan konfirmasi sebelum hapus

j. Pagination navigasi

```
<?php if ($totalPages > 1): ?>
<nav>
  <ul class="pagination">
    <?php for ($i = 1; $i <= $totalPages; $i++): ?>
      <li class="page-item <?= ($i == $page) ? 'active' : '' ?>">
        <a class="page-link" href="?cari=<?= urlencode($cari)
?>&page=<?= $i ?>"><?= $i ?></a>
      </li>
    <?php endfor; ?>
  </ul>
</nav>
<?php endif; ?>
```

- Menampilkan navigasi halaman jika data lebih dari satu halaman.
- Halaman aktif akan memiliki class active.
- Link akan membawa parameter cari agar pencarian tetap aktif saat berpindah halaman.

4. Halam Topup



Gambar 4. 9 Tampilan *Top Up*

Halaman ini adalah *Topup Saldo*. Di bagian atas terdapat kolom *UID* yang menampilkan kode kartu pengguna. Di bawahnya ada kolom Jumlah *Topup* **(**Rp**)** tempat admin memasukkan nominal saldo yang ingin ditambahkan. Setelah itu, tombol *Submit Topup* digunakan untuk menyimpan transaksi top-up ke sistem. Tersedia juga tombol Lihat Riwayat Transaksi agar admin bisa langsung mengecek catatan transaksi yang sudah dilakukan. Tampilan dibuat sederhana agar proses pengisian saldo cepat dan mudah.

a. Koneksi dan timeson

```
require_once '../koneksi.php';
date_default_timezone_set('Asia/Makassar');
```

- Menghubungkan ke database menggunakan koneksi.php.
- Mengatur timezone ke WITA (Asia/Makassar) agar waktu transaksi sesuai lokasi server/user.

b. Cek from sumbit

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    ...
}
```

- Mengecek apakah form dikirim menggunakan metode POST.

c. Ambil dan validasi input

```
$uid = trim($_POST['uid'] ?? '');
$nominal = (int)($_POST['nominal'] ?? 0);
$pin = trim($_POST['pin'] ?? '');
$pin_admin = "135790";
```

Mengambil:

- uid: UID user yang akan di-topup.
- nominal: jumlah topup.
- pin: PIN admin untuk verifikasi.

$pin_admin adalah PIN admin hardcoded (bisa dipindah ke database untuk keamanan lebih baik).

d. Validasi input

```
if ($uid === '' || !preg_match('/^\d+$/', $uid)) { ... }
elseif ($nominal <= 0) { ... }
elseif ($pin === '') { ... }
elseif ($pin !== $pin_admin) { ... }
```

Mengecek:

- UID tidak kosong & hanya angka.
- Nominal harus lebih dari 0.
- PIN harus diisi dan harus cocok dengan $pin_admin.

Jika gagal, akan menampilkan pesan error sesuai kondisi.

e. Cek uid

```
$sqluser = "SELECT * FROM users WHERE uid = ?";
$stmtuser = mysqli_prepare($konek, $sqluser);
mysqli_stmt_bind_param($stmtuser, 's', $uid);
mysqli_stmt_execute($stmtuser);
$result = mysqli_stmt_get_result($stmtuser);
$user = mysqli_fetch_assoc($result);
mysqli_stmt_close($stmtuser);
```

- Menggunakan prepared statement untuk mencegah SQL Injection.
- Mencari user dengan UID yang diinputkan.

f. Jika user tidak di temukan

```
if (!$user) {
    $pesan = "User dengan UID tersebut tidak ditemukan.";
}
```

- Menampilkan error jika UID tidak terdaftar.

g. Update data user

```
$saldo_baru = $user['saldo'] + $nominal;
$sql_update = "UPDATE users SET saldo = ? WHERE uid = ?";
$stmt_update = mysqli_prepare($konek, $sql_update);
mysqli_stmt_bind_param($stmt_update, 'is', $saldo_baru, $uid);
$update_success = mysqli_stmt_execute($stmt_update);
mysqli_stmt_close($stmt_update);
```

- Menambahkan nominal topup ke saldo lama user dan update saldo di database.

h. Simpan riwayat transaksi

```
if ($update_success) {
    $waktu = date('Y-m-d H:i:s');
    $stmt = $konek->prepare("INSERT INTO transaksi2 (uid, keterangan, nominal, waktu) VALUES (?, 'topup', ?, ?)");
    $stmt->bind_param("sis", $uid, $nominal, $waktu);
    ...
}
```

Jika update saldo berhasil:

Menyimpan ke tabel transaksi2 dengan detail:

- uid

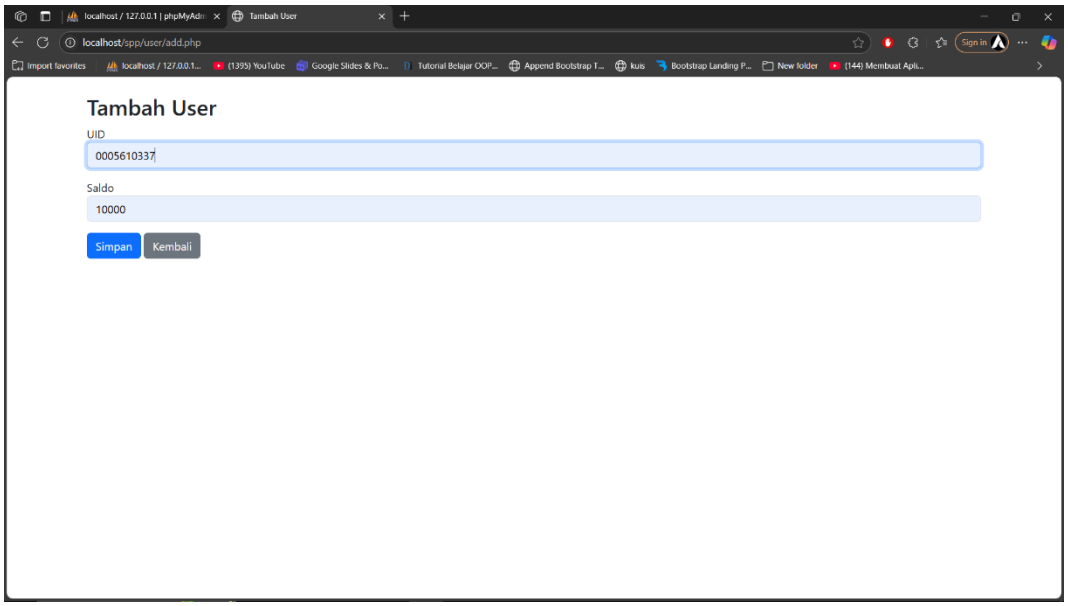
- keterangan (topup)
- nominal
- waktu saat topup

i. tampilkan pesan keberhasilan

```
$pesan = "Topup berhasil! Saldo baru: Rp " . number_format($saldo_baru, 0, ',', '.');
```

- Memberikan pesan sukses dengan saldo user terbaru dalam format Rupiah.

5. Tambah user



Gambar 4. 10 Tampilan Tambah *User*

Halaman ini adalah Tambah *User*. Tersedia kolom UID untuk memasukkan kode kartu RFID pengguna baru, serta kolom Saldo untuk menentukan saldo awal. Setelah data diisi, tombol Simpan digunakan untuk menyimpan user ke sistem, sementara tombol Kembali membawa admin kembali ke halaman daftar user. Tampilan ini sederhana agar proses pendaftaran pengguna baru cepat dilakukan.

a. Koneksi zona waktu

```
require_once '../koneksi.php';
date_default_timezone_set('Asia/Makassar');
```

- require_once '../koneksi.php' : Menghubungkan ke database MySQL.
- date_default_timezone_set('Asia/Makassar') : Mengatur waktu server menjadi WITA agar waktu data tgl sesuai lokasi.

b. Variabel pesan eror

```
$pesan_error = '';
```

- Digunakan untuk menyimpan pesan error apabila terjadi kesalahan saat proses tambah user, seperti UID sudah terdaftar.

c. Proses sumbit from

```
if ($_SERVER['REQUEST_METHOD'] === 'POST') { ... }
```

- Memastikan kode hanya dieksekusi jika form dikirim menggunakan metode POST.

d. Mengambil data dari from

```
$uid = trim($_POST['uid']);
$nama = trim($_POST['nama'] ?? '');
$alamat = trim($_POST['alamat'] ?? '');
$saldo = (float)($_POST['saldo']);
$tgl = date('Y-m-d H:i:s');
```

- uid: UID RFID user yang akan didaftarkan.

- nama: Nama user (sementara dikomentari pada form).
- alamat: Alamat user (sementara dikomentari pada form).
- saldo: Saldo awal user.
- tgl: Tanggal dan waktu saat user didaftarkan.

e. Menyimpan data ke database

```
$sql = "INSERT INTO users (uid, nama, alamat, saldo, tgl) VALUES (?, ?, ?, ?, ?)";
$stmt = mysqli_prepare($konek, $sql);
mysqli_stmt_bind_param($stmt, 'sssds', $uid, $nama, $alamat, $saldo, $tgl);
mysqli_stmt_execute($stmt);
mysqli_stmt_close($stmt);
```

- Menggunakan prepared statement agar aman dari SQL Injection.
- Menyimpan data uid, nama, alamat, saldo, dan tgl ke tabel users.

f. Pesan eror

```
<?php if ($pesan_error): ?>
    <div class="alert alert-danger"><?= $pesan_error ?></div>
<?php endif; ?>
```

- Menampilkan pesan error jika UID sudah terdaftar saat pendaftaran user baru.

g. From tambah user

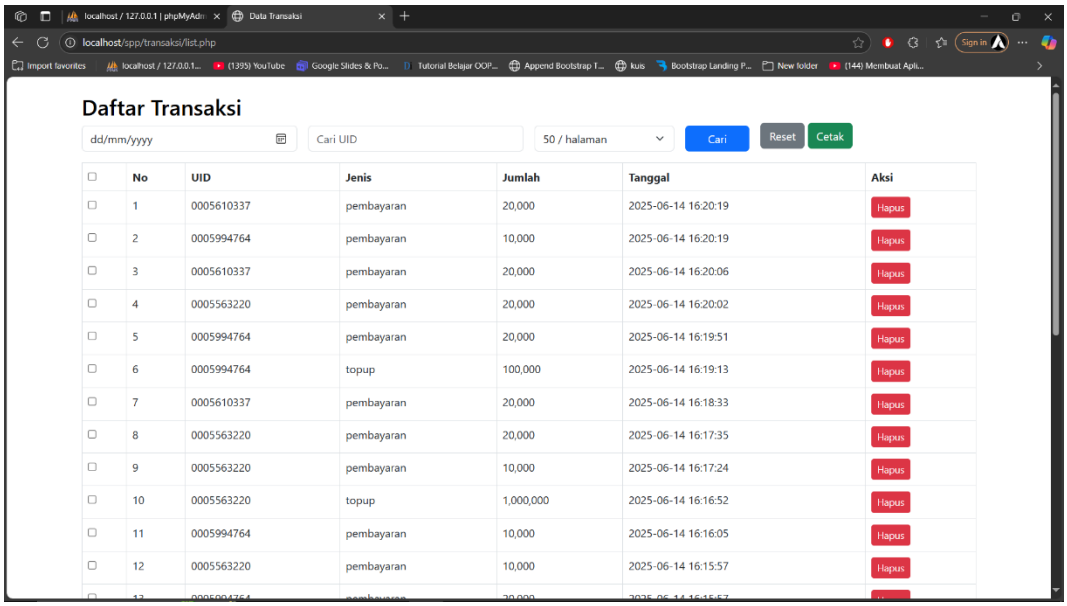
```
<form method="post">
    <div class="mb-3">
        <label>UID</label>
        <input type="text" name="uid" class="form-control" required>
    </div>
    <!-- Nama dan Alamat dikomentari -->
    <div class="mb-3">
```

```
            <label>Saldo</label>
            <input type="number" name="saldo" class="form-control" required>
        </div>
        <button type="submit" class="btn btn-primary">Simpan</button>
        <a href="list.php" class="btn btn-secondary">Kembali</a>
    </form>
```

- Form input:
- UID (wajib diisi) → UID RFID user.
- Saldo (wajib diisi) → saldo awal user.
- Kolom Nama dan Alamat dikomentari, sehingga saat ini tidak digunakan tetapi siap jika ingin diaktifkan kembali di masa mendatang.

6. Halaman Laporan Transaksi

Pada halaman laporan transaksi terdapat data laporan yang telah memasuki wahana seperti; Uid, jenis, total biaya, dan tanggal transaksi.



Gambar 4 11 Tampilan laporan transaksi

Halaman ini adalah Daftar Transaksi. Terdapat kolom pencarian berdasarkan tanggal dan UID untuk memudahkan filter data, pilihan jumlah data per halaman, serta tombol Cari, *Reset*, dan Cetak. Tabel di bawahnya menampilkan daftar transaksi dengan informasi nomor urut, UID kartu, jenis transaksi (pembayaran atau topup), jumlah uang, tanggal, serta tombol Hapus pada kolom aksi. Halaman ini berfungsi sebagai pusat monitoring seluruh aktivitas transaksi dalam sistem

a. Keamanan session

```
session_start();
if (!isset($_SESSION['username'])) {
    header("Location: ../login.php");
    exit;
}
```

- Memastikan hanya user yang sudah login (terautentikasi) yang dapat mengakses halaman transaksi ini.
- Jika tidak login, akan diarahkan ke login.php.

b. Koneksi database

```
require_once '../koneksi.php';
```

- Menghubungkan ke database MySQL untuk pengolahan data transaksi.

c. Pengambilan Parameter Filter dan Pagination

```
$tanggal = $_GET['tanggal'] ?? '';
$uid = $_GET['uid'] ?? '';
$limit = (int)($_GET['limit'] ?? 50);
$page = (int)($_GET['page'] ?? 1);
$page = max($page, 1);
$offset = ($page - 1) * $limit;
```

Mengambil filter:

- tanggal: untuk filter transaksi berdasarkan tanggal tertentu.
- uid: untuk filter transaksi berdasarkan UID RFID tertentu.

Mengambil pagination:

- limit: jumlah data per halaman (default 50).
- page: halaman aktif.

offset: untuk menentukan dari data ke berapa data akan ditampilkan pada query SQL.

d. Query Hitung Total Data

```
$count_sql = "SELECT COUNT(*) as total FROM transaksi2 t LEFT JOIN users u ON t.uid = u.uid WHERE 1=1";
if (!empty($tanggal)) $count_sql .= " AND DATE(t.waktu) = '$tanggal'";
if (!empty($uid)) $count_sql .= " AND t.uid LIKE '%$uid%'";
$total_rows = mysqli_fetch_assoc(mysqli_query($konek, $count_sql))['total'];
$total_pages = ceil($total_rows / $limit);
```

Query untuk menghitung total data transaksi sesuai filter:

- Jika tanggal terisi → filter tanggal transaksi.
- Jika UID terisi → filter UID.

Menghitung total halaman (total_pages) untuk pagination.

e. Tabel transaksi

```
<form method="post" action="hapus_transaksi.php" id="formHapus">
   <table class="table table-bordered">...</table>
</form>
```

Tabel menampilkan:

- Checkbox untuk memilih data.
- No.

- UID.
- Jenis transaksi (topup/deduct).
- Jumlah transaksi.
- Tanggal transaksi.
- Tombol aksi (jika diperlukan).

Terdapat tombol:

- Hapus Semua data transaksi dengan konfirmasi.

Bagian <tbody id="dataTransaksi"> kosong saat awal karena akan diisi otomatis menggunakan AJAX agar data realtime.

f. Pagination

```
<nav>
  <ul class="pagination justify-content-center">...</ul>
</nav>
```

Jika total halaman lebih dari 1, akan muncul navigasi halaman:

- Tombol Sebelumnya.
- Nomor halaman.
- Tombol Berikutnya.

Pagination menjaga efisiensi loading halaman saat data transaksi sangat banyak.

g. Checkbox Select All

```
document.getElementById("checkAll").onclick = function() {
```

```
    document.querySelectorAll('input[name="uid[]"]').forEach(cb =>
cb.checked = this.checked);
};
```

- Jika checkbox di header tabel dicentang, semua checkbox transaksi akan ikut dicentang secara otomatis.

h. AJAX Memuat Data Realtime

```
function loadData() {
    const params = new URLSearchParams(window.location.search);
    fetch('transaksi_data.php?' + params.toString())
        .then(res => res.text())
        .then(html => {
            document.getElementById('dataTransaksi').innerHTML = html;
        });
}

// Jalankan pertama kali + interval 3 detik
loadData();
setInterval(loadData, 3000);
```

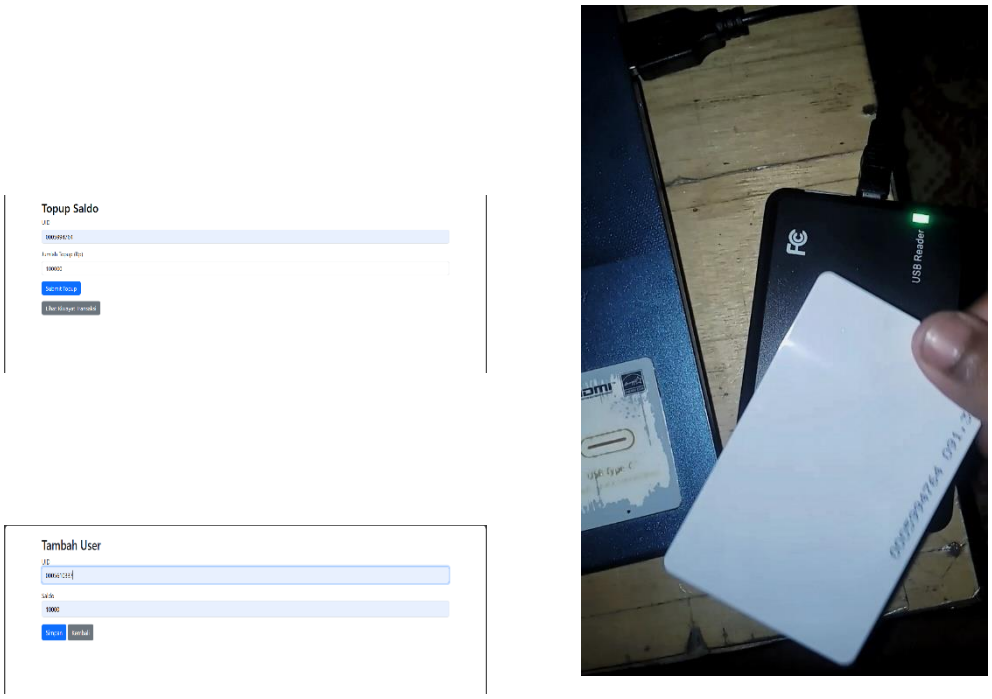
- Memanggil transaksi_data.php untuk mengambil data transaksi sesuai filter tanpa reload halaman**.**
- Data akan diperbarui secara otomatis setiap 3 detik (3000 ms) agar admin/operator dapat melihat transaksi terbaru secara realtime.

D. Pengujian Sistem

1. Pengujian Alat

Ada beberapa hasil percobaan atau hasil pengujian alat agar memastikan perangkat berjalan dengan baik,bisa dilihat dari gambar berikut;

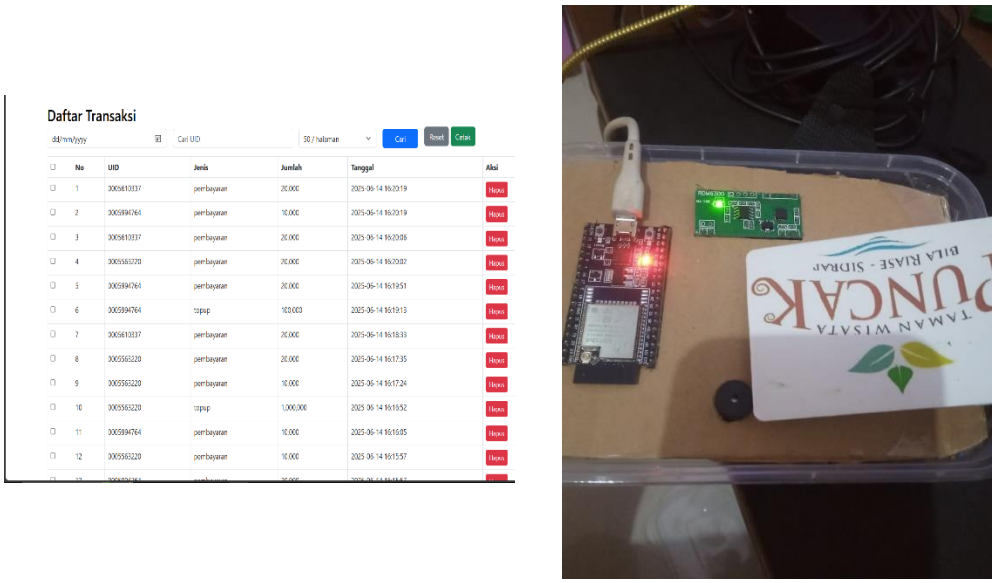
a. Pengujian daftar dan mengisi saldo pengunjung



Gambar 4. 12 pengujian daftar user dan top up

Pada gambar di atas telah menujukkan bahwa alat dapat berfungsi dengan baik pada aplikasi pembayaran wahana berbasis RFID, dengan cara *teg* kartu RFID di RFID reader untuk mendaftark Uid dan masukkan nominal saldo lalu simpan tidak dapat melakukan transaksi melalui alat ini.

b. Pengujian transaksi

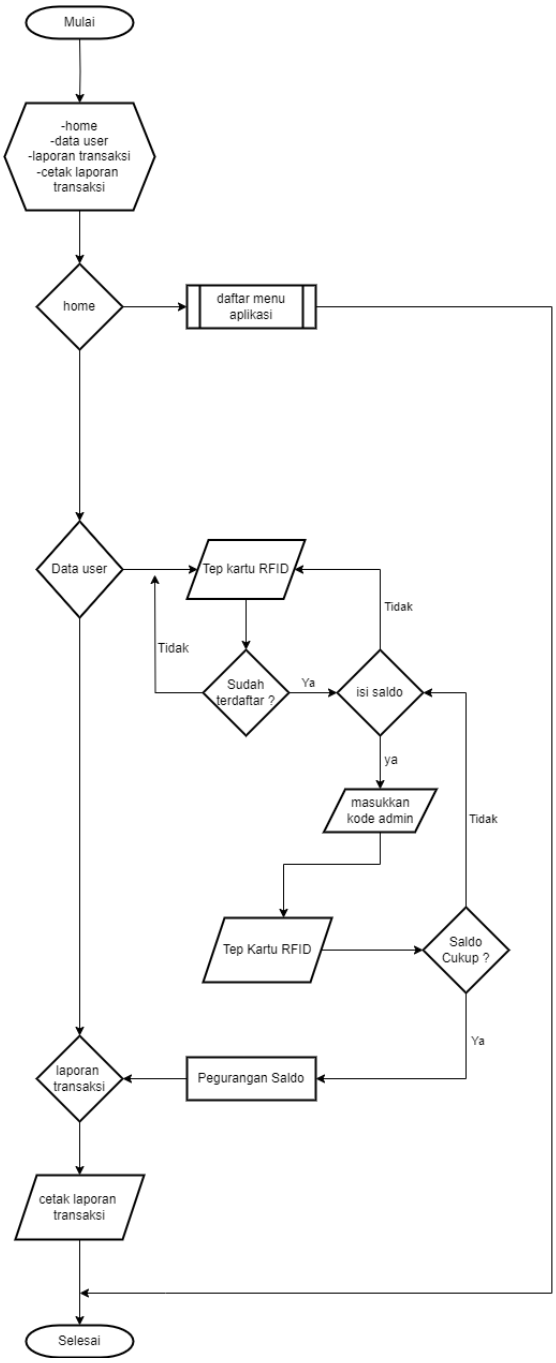


Gambar 4. 13 pengujian transaksi

Pada gambar di atas telah menujukkan bahwa alat dapat berfungsi dengan baik pada aplikasi pembayaran wahana berbasis RFID, dengan cara *teg* kartu RFID pada RFID reader, lalu saldo akan terkurang secara otomatis sesuai harga wahana yang di masuki dan tidak dapat melakun daftar uid dan topup melui alat ini.

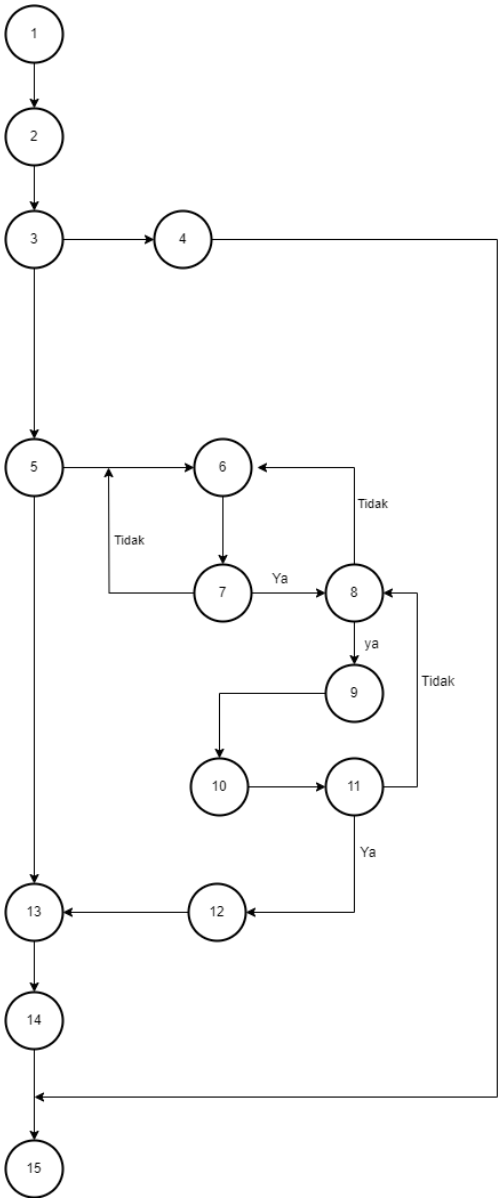
2. Pegujian White Box

a. Flowchard



Gambar 4. 14 Flowchard

b. Flowgrafh



Gambar 4. 15 Flowgrafh

Dari flowgrafh menu halaman katalog di atas dapat dilakukan proses perhitungan sebagai berikut :

a. Menghitung *cyclomatic complexity* V(G) dari *edge* dan *node* dengan rumus : V (G) = E – N + 2

   N (node) = 15

   E (edge) = 19

   Penyelesaian :

   V(G) = E – N + 2

   = 19 – 15 +2

   = 6

   Predikat (P) = P + 1

   = 5 + 1

   = 6

b. Berdasarkan perhitungan *cyclomatic complexity* dari flowgrafh di atas memiliki regional =

c. Independen path pada flowgrafh di atas adalah :

   Path 1 = 1,2,3,5,13,14,15

   Path 2 =1,2,3,4,15

   Path 3= 1,2,3,5,6,7,8,9,10,11,12,13,14,15

Tabel 4. 3 Grafik Matriks Aplikasi

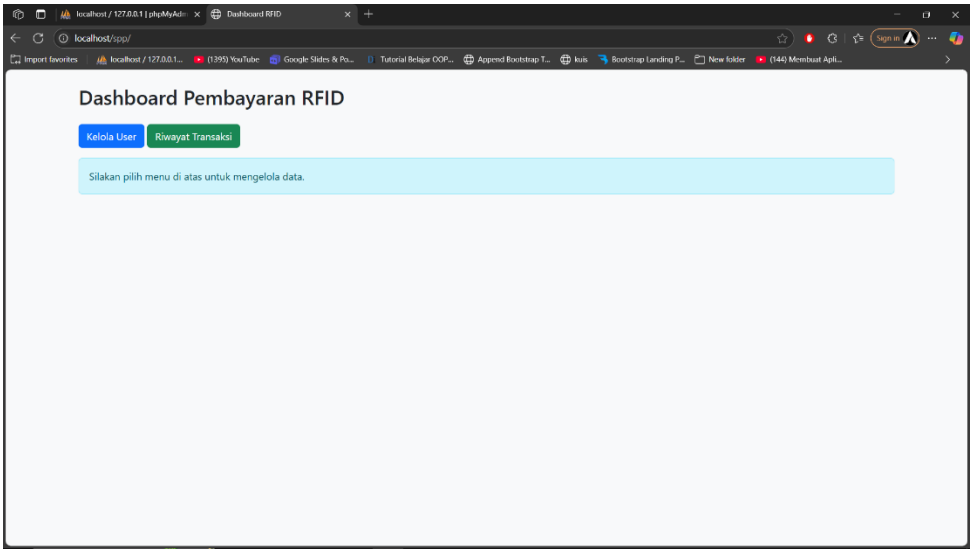
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		E – 1
1		1															1-1=0
2			1														1-1=0
3				1	2												3-1=2
4													1		1		2-1=1
5						1											1-1=0
6							1										1-1=0
7								1									1-1=0
8									1								1-1=0
9										1							1-1=0
10											1						1-1=0
11												1					1-1=0
12													1				1-1=0
13														2			2-1=1
14															2		2-1=1
15																	
		Sum (E + 1)															5+1=6

3. Pengujian *Bleck Box*

   a. Pengujian *Bleck Box* Menu Home

Tabel 4. 4 *Bleck Box* Menu Home

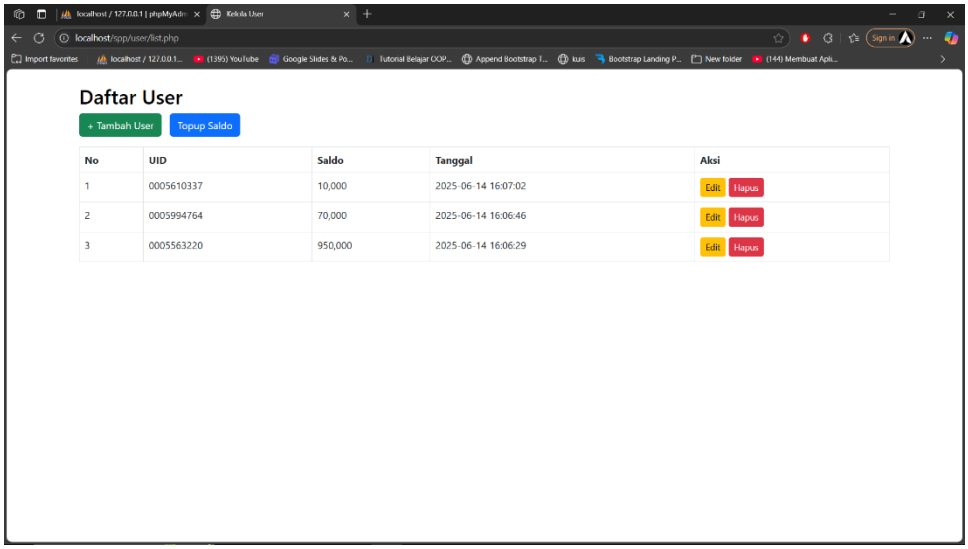
Tes Faktor	Hasil	Keterangan
Menu Home	✓	Sukses pada saat sudah login di aplikasi langsung ke menu Home



b. Pengujian *Beleck Box* Data user

Tabel 4. 5 *Bleck Box* Data user

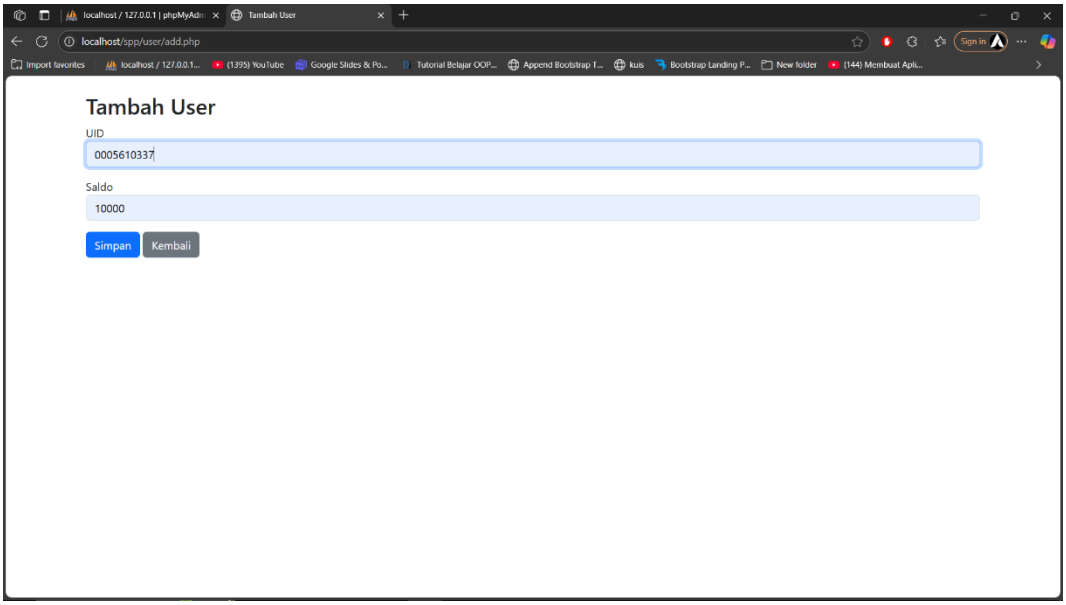
Tes Faktor	Hasil	Keterangan
Data user	✓	Sukses pada saat menampilkan data user.



c. Pegujian *Bleck Box* Tambah user

Tabel 4. 6 *Bleck Box* Tambah user

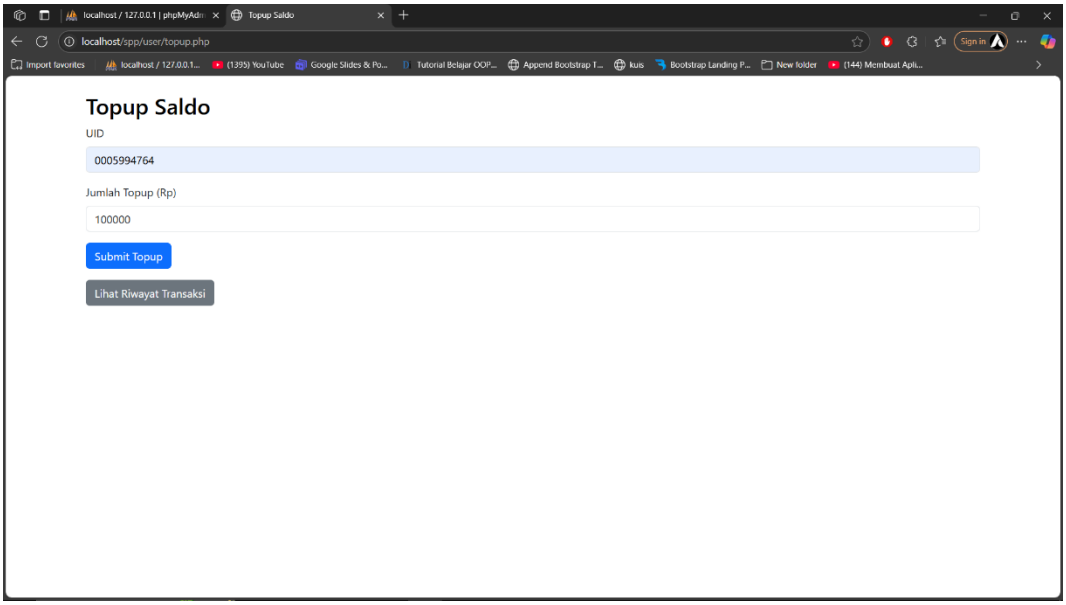
Tes Faktor	Hasil	Keterangan
Kelola user	✓	Sukses menampilkan tambah user



d. Pengujian bleck box topup saldo

Tabel 4. 7 *Bleck Box* top up saldo

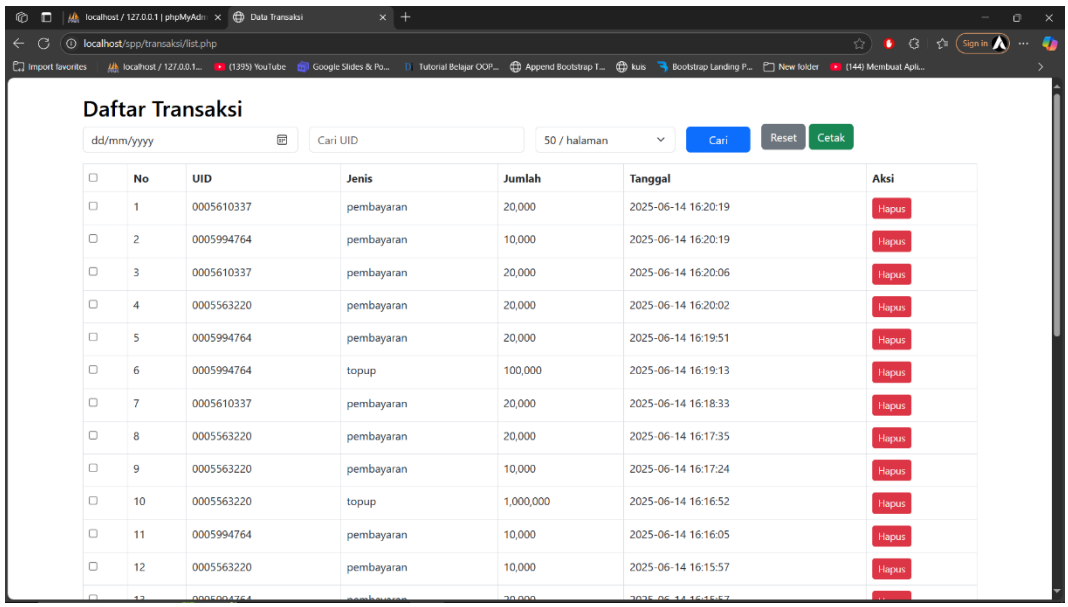
Tes Faktor	Hasil	Keterangan
Topup Saldo	✓	Sukses menampilkan topup saldo dan menambahkan saldo.



e. Pegujian *Bleck Box* DaftarTransaksi

Tabel 4. 8 *Bleck Box* Daftar Transaksi

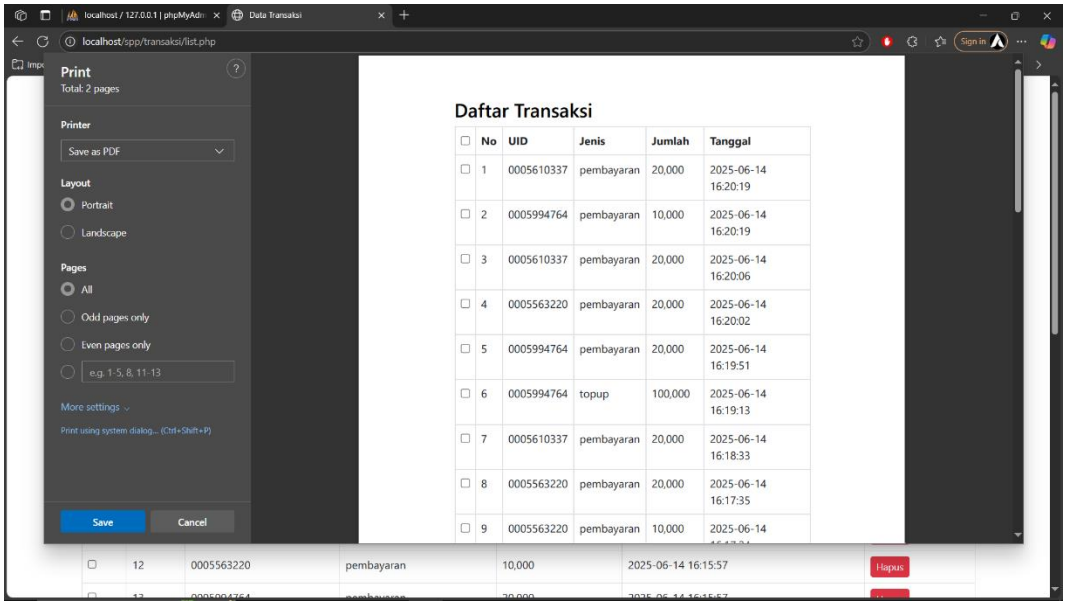
Tes Faktor	Hasil	Keterangan
Daftar Transaksi	✓	Sukses membaca data user pada saat melakukan topup dan transaksi di wahan yang di pilih



f. Pegujian *Bleck Box* Cetak Daftar Transaksi

Tabel 4. 9 *Bleck Box* Cetak Daftar Transaksi

Tes Faktor	Hasil	Keterangan
Cetak daftar transaksi	✓	Sukses pada saat melakukan percetakan.



BAB V

PENUTUP

A. KESIMPULAN

Berdasarkan hasil penelitian yang dijabarkan pada bab-bab sebelumnya, maka di simpulkan sebagai berikut:

1. Setelah aplikasi ini selesai maka dapat diketahui bahwah sistem pembayaran wahana berbasis RFID dapat menjadi sebuah alat untuk melakukan pembayaran wahana dimana kita menggunakan alat yaitu; kartu RFID dan RFID reader.
2. Sistem pembayaran ini menggunakan kartu RFID yang termasuk salah satu alat yang berfungsi untuk melakukan transaksi ataupun sebagai identitas pemilik kartu, dalam kartu terdapat uid yang membedakan antara kartu yang satu dengan yang lain. Dan adapun alat yang lain di gunakan yaitu RFID reader sebagai alat untuk pembaca uid kartu RFID.

B. SARAN

Berdarkan hasil di atas penulis memberikan saran sebagai berikut:

1. Aplikasi pembayaran wahana dapat di tamabahkan notifikasi ke smarphone untuk mengetahui isi saldo kartu RFID.
2. Aplikasi pembayaran wahana dapat di tambahkan big coin jika sudah menaiki wahan, jika sudah mencapai 4 kali naik maka bisa memilih salah satu wahana untuk di mainkan.

DAFTAR PUSTAKA

Ade Hastuty (2021), Dasar Pemrograman, Cetakan 2 (revisi), Makassar: CV Bangun Bumitama, hal 18-23.

Adi Nugroho. 2009. *Rekayasa Perangkat Lunak Menggunakan UML Dan Java* Yogyakarta : Andi

Asriani. 2014. *Prototype Palang Pintu otomatis menggunakan RFID (Radio Frekuensi Identification)*. Makassar : Universitas Islam Negri Alauddin

Budiharto, W. dan , D., 2014, *ARTIFICIAL INTELLLIGENCE Konsep dan Penerapanya*, Andi Offset, Yogyakarta..

Dermawan, R., & Rusdi. (2013). *Keperawatan Jiwa: Konsep dan Kerangka Kerja Asuhan Keperawatan Jiwa.* Yogyakarta : Gosyen Publishing.

Edwin, M., Hastuty, A., & Wafiah, A. (2024). APLIKASI PENGAMAN BERBASIS ANDROID [Universitas Muhammadiyah Parepare]. In Jurnal Fakultas Teknik Universitas Muhamadiyah Parepare: Vol. xx No. xx. https://repository.umpar.ac.id/id/eprint/1006/

Henriani, & Ade Hastuty. (2024). E – Booking Bus Pariwisata Berbasis Web. Jurnal Sintaks Logika, 4(3), 2775–412. DOI: https://doi.org/10.31850/jsilog.v4i3.3326

Istiyanto,jazi Eko.2014.*Pengantar Elektronika dan Istrumentasi Pendekatan Project Arduino dan Android*.Yogyakarta : Penerbit Andi

Purwiantono, Febry Eka, Muhammad Sofwan Romli, dan Addin Aditya.2019.*Pemanfaatan RFID (Radio Frequncy Identification) Sebagai Alternatif Absensi Siswa (Studi Kasus : SMK Ar-Rahmah Sukabumi Jawa Barat)*

Raharjo, B. (2015). *Belajar Otodidak Framework Codeigniter*. Bandung: Informatika Bandung.

Simanjuntak, JimmySatria, Jaka Prayudha, Jufri Halim. 2022. *Sistem Pembayaran Non Tunai Dengan Memanfaatkan E-KTP Menggunakan Teknik Simplex Berbasis Arduino*

Tahir, F., Hastuty, A., & Yunus, M. (2025). *Sistem Informasi Pengelolaan Data Pada Perpustakaan Umum Parepare.* Komputek, 9(1). DOI: https://doi.org/10.24269/jkt.v9i1.3227.