BAB I

PENDAHULUAN

A. Latar Belakang

Kemajuan teknologi yang pesat didorong oleh keinginan masyarakat untuk mempermudah aktivitas sehari-hari. Banyak teknologi, seperti internet, dikembangkan untuk memberikan kenyamanan tersebut. Saat ini, tidak bisa dipungkiri bahwa internet tersedia untuk hampir semua kebutuhan. Dengan perkembangan teknologi internet, orang dapat memperoleh informasi dan melakukan transaksi dengan leluasa tanpa batasan ruang dan waktu. Salah satu bentuk kemajuan teknologi yang memudahkan kegiatan ini adalah *E-tiket* (Masuara dkk., 2015).

E-tiket, singkatan dari Elektronik Tiket, adalah tiket dalam bentuk digital. Menurut Akbar & Sembiring (2023), E-tiket sering digunakan dalam acara-acara besar karena mudah diakses. Wisatawan atau pembeli tidak perlu datang langsung ke lokasi untuk membeli tiket mereka cukup memesan melalui internet atau aplikasi.

Salah satu penerapan umum dari *E-tiket* adalah menggunakan Quick Response Code *(QR Code)*. Teknologi *QR Code* ini merupakan pengembangan dari *Barcode*, yang awalnya hanya berupa kode satu dimensi dan kini menjadi kode dua dimensi dengan kapasitas penyimpanan data yang lebih besar. Melalui *QR Code,* data yang tersimpan dapat berupa angka, huruf, atau biner. Teknologi ini telah diterapkan dalam berbagai sektor, termasuk di bidang pariwisata.

Penggunaan *QR Code* pada *e-tiket* memiliki banyak keunggulan, salah satunya adalah kemampuan *QR Code* dalam menyimpan lebih banyak data dibandingkan dengan *barcode* (Hadiana, 2016). Selain itu, menurut Brown (2019), *QR* Code juga lebih sulit untuk dipalsukan jika dibandingkan dengan tiket fisik. *E-tiket* yang menggunakan *QR Code* memungkinkan proses *check-in* menjadi lebih cepat, karena dapat dipindai langsung dari *smartphone* pengguna.

E-tiket menjadi sangat penting dalam konteks pariwisata karena memungkinkan wisatawan untuk merencanakan perjalanan mereka dengan lebih mudah fleksibel dan efisien. Salah satu tempat pariwisata yang belum menerapkan penerapan teknologi *E-tiket* ini yaitu Diana *waterpark* Barru, yang masih melakukan pemesanan tiket secara manual di kasir. Wisata Diana *Waterpark* Barru adalah salah satu destinasi pariwisata di Sulawesi Selatan, yang sering menghadapi antrian di loket tiket baik itu di tiket masuk akibat tingginya jumlah pengunjung.

Upaya untuk mengatasi masalah ini, pemanfaatan teknologi melalui aplikasi pemesanan tiket masuk dan tiket wahana di wisata diana waterpark Barru menggunakan *QR Code* menjadi solusi yang efektif. Aplikasi ini dirancang untuk meningkatkan efisiensi pemesanan tiket, mengurangi waktu tunggu, yang tujuan nya adalah mempercepat proses pemesanan, meningkatkan kepuasan pengunjung, dan mengelola data pengunjung serta pendapatan yang lebih baik. Dengan demikian, aplikasi ini diharapkan dapat meningkatkan kualitas layanan di Diana *Waterpark* Barru.

B. Rumusan Masalah

Bagaimana merancang aplikasi pemesanan tiket masuk dan tiket wahana di Wisata Diana *Waterpark* Barru Menggunakan *QR Code*?

C. Tujuan Penelitian

Penelitian ini bertujuan untuk merancang Aplikasi Pemesanan Tiket Masuk dan Tiket Wahana Wisata Diana *Waterpark* Barru menggunakan *QR Code*

D. Batasan Masalah

1. Penelitian ini fokus pada perancangan aplikasi pemesanan tiket masuk dan tiket wahana di dalam Diana *Waterpark*, tidak mencakup tiket untuk acara khusus atau paket wisata lain yang mungkin ditawarkan oleh tempat tersebut.
2. Aplikasi ini hanya berfokus pada objek Wisata Diana *Waterpark.*

E. Manfaat Penelitian

Adapun Manfaat Penelitian ini yaitu sebagai berikut:

1. Segi pendidikan

   Penelitian ini memberikan wawasan tentang penerapan teknologi dalam industri pariwisata. Mahasiswa dapat memahami bagaimana teknologi *QR code* dapat digunakan untuk meningkatkan efisiensi dan kenyamanan dalam proses pemesanan tiket. Ini membuka peluang untuk pengembangan lebih lanjut dalam penggunaan teknologi digital di berbagai sektor lainnya. Selain itu, penelitian ini juga memberikan contoh kasus nyata tentang bagaimana integrasi teknologi dapat mengatasi masalah-masalah operasional dan logistik di dunia nyata, yang

sangat bermanfaat untuk studi-studi di bidang manajemen dan teknologi informasi.

2. Bagi pengunjung atau pengguna

Manfaat penelitian ini terlihat dalam peningkatan pengalaman mereka saat berkunjung ke Diana *Waterpark*. Aplikasi pemesanan tiket yang menggunakan *QR code* menawarkan kemudahan dan efisiensi, mengurangi waktu antrian dan penggunaan teknologi ini juga meningkatkan keamanan transaksi karena mengurangi risiko tiket palsu. Selain itu, aplikasi ini dapat memberikan informasi tambahan yang relevan bagi pengunjung, meningkatkan kepuasan dan keinginan mereka untuk kembali berkunjung.

BAB II

TINJAUAN PUSTAKA

A. Kajian Teori

1. Pemesanan

Pemesana adalah suatu aktifitas yang dilakukan oleh konsumen sebelum membeli. Untuk mewujudkan kepuasan konsumen maka suatu perusahaan harus memiliki sebuah sistem pemesanan yang baik. Santoso & Rahman (2015)

Menurut Edwin dan Chris, pemesanan secara umum adalah kesepakatan atau kontrak antara dua pihak atau lebih untuk memesan suatu layanan atau fasilitas tertentu. Bentuk pemesanan ini dapat meliputi tiket, reservasi barang, ruangan, kamar, tempat duduk, dan lainnya pada waktu yang telah ditentukan, termasuk produk jasa yang menyertainya.

Berdasarkan definisi pemesanan yang telah dijelaskan sebelumnya, dapat disimpulkan bahwa pemesanan adalah kesepakatan antara dua pihak atau lebih, yaitu penyedia dan pengguna layanan atau barang, untuk memenuhi kebutuhan tertentu. Kesepakatan ini dapat mencakup reservasi barang, ruangan, tempat duduk, kamar, dan lainnya pada periode waktu yang ditentukan.

2. Tiket

Tiket adalah sebuah tanda masuk atau syarat masuk yang digunakan untuk mendatangi atau mengahadiri sebuah tempat/acara tertentu. Supriyono & Nugroho (2019). Tiket adalah dokumen atau karcis yang digunakan sebagai bukti pembayaran untuk mendapatkan akses ke suatu acara, tempat, atau layanan. Secara

umum, tiket berfungsi untuk mengatur dan memverifikasi akses pengunjung, baik dalam konteks acara hiburan, transportasi, maupun tempat wisata.

Tiket berfungsi sebagai media yang memberikan hak akses bagi seseorang untuk memasuki suatu tempat wisata atau fasilitas tertentu. Saat ini, pengelola wisata menggunakan tiket sebagai bukti bahwa seseorang telah menyelesaikan pembayaran dan memperoleh akses masuk ke area wisata tersebut.

Tiket menjadi elemen penting dalam sektor pariwisata, termasuk tiket pesawat, wisata, hingga konser. Pemesanan tiket wisata dan wahana kini dapat dilakukan melalui sistem yang memanfaatkan fitur *QR Code*, yang juga efektif mengurangi potensi kecurangan, seperti pemalsuan tiket. Hirawan & Safitri (2019). Dengan menggunakan sistem pemesanan ini, pengunjung dapat menghemat waktu karena tidak perlu mengantri di loket tiket di berbagai tempat wisata.

Pemesanan tiket merupakan salah satu proses dimana seseorang memesan tempat atau kursi untuk sebuah acara, perjalanan, atau layanan tertentu. Salah satunya adalah tiket masuk dan wahana, pemesanan tiket sering dilakukan dengan cara memesan langsung ke loket, namun proses pemesanan tersebut kurang efektif baik dari segi waktu. Untuk itulah perlu adanya proses pemesanan yang lebih mengefektifkan baik dari segi waktu itu sendiri serta lebih memudahkan, lebih praktis dan lebih cepat dalam melakukan pemesanan tiket adalah dengan menggunakan smartphone dan internet, karena dengan fasilitas ini segala bentuk pemesanan dapat dilakukan kapan dan dimana saja sehingga lebih memudahkan masyarakat yang akan melakukan pemesanan tiket.

Terdapat beberapa macam jenis tiket dalam objek wisata diantaranya yaitu:

- Tiket Masuk

  Tiket masuk adalah syarat utama seseorang dapat diberikan izin akses untuk masuk kedalam objek wisata yang ingin dituju, tampa ada nya bukti tiket seseorang tidak dapat memasuki tempat wisata tersebut

- Tiket Wahana

  Tiket wahana adalah syarat seseorang untuk dapat diberikan akses menggunakan atau memainkan wahana yang tersedia pada objek wisata tersebut, tampa adanya tiket wahana seseorang tidak dapat diberikan izin atau akses untuk menggunakan atau memainkan wahana yang ingin di mainkan.

3. Android



Gambar 2. 1 Android
Sumber: https://www.google.com

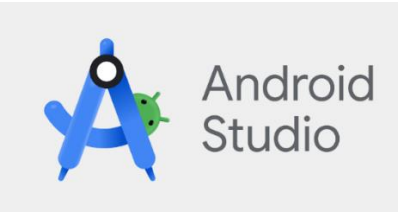
Menurut Santoso, S. dkk. (2015) Android merupakan sebuah *Operating System* berbasis *linux* yang dipakai oleh perangkat seluler seperti *Smartphone* dan tablet. Menurut Hermawan Yunus & Sardiwan (2018) mengemukakan bahwa Android merupakan suatu *Operating System* atau OS yang sampai saat ini masih dalam tahap perkembangan, OS ini seperti OS lainnya seperti, Symbian, IOS di IPhone, dan lain sebagainya

Android adalah sistem operasi berbasis *Linux* yang dirancang untuk perangkat seluler, memberikan platform bagi pengembang untuk menciptakan aplikasi.

Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Pada saat ini banyak vendor-vendor yang telah memproduksi smartphone berbasis Android karena Android merupakan system operasi yang *open source* sehingga bebas dipakai oleh vendor manapun. Untuk meningkatkan kualitas pemesanan yang sebelumnya dilakukan secara manual, sehingga menjadi lebih efisien dalam mencari informasi terkait pemesanan tiket masuk dan wahana di Diana *Waterpark*, aplikasi ini dikembangkan guna memudahkan pelanggan menghemat waktu dan biaya.

Pengembang memiliki beberapa opsi ketika membuat aplikasi android. Ada beberapa pengembang yang menggunakan Eclipse, sebuah IDE yang tersedia secara gratis, untuk merancang dan mengembangkan aplikasi Android. Eclipse populer untuk pengembangan Android karena memiliki plugin khusus yang memudahkan proses tersebut. Selain itu, Eclipse sebagai IDE untuk pengembangan aplikasi Android didukung langsung oleh Google, yang terlihat dari adanya plugin tambahan untuk pembuatan proyek.

4. Android Studio



Gambar 2. 2 Android Studio
Sumber: https://share.google/images

Android Studio adalah IDE untuk pengembangan Android yang diperkenalkan oleh Google pada acara Google I/O 2013. Berbasis pada IDE Java populer, IntelliJ IDEA, Android Studio dikembangkan dari Eclipse IDE. (Thamrin, dkk 2019)

Kini, Android Studio menjadi IDE resmi untuk pembuatan aplikasi Android dan menawarkan banyak fitur baru yang tidak terdapat pada Eclipse IDE. Beberapa fitur unggulan yang dimiliki oleh Android Studio meliputi integrasi yang lebih baik dengan Gradle, dukungan untuk berbagai template dan layout, serta alat analisis performa aplikasi yang canggih. Selain itu, Android Studio juga menyediakan emulator yang lebih cepat dan lebih responsif, sehingga memudahkan pengembang untuk menguji aplikasi mereka di berbagai perangkat dan konfigurasi. pembaruan yang terus menerus dan dukungan dari komunitas yang luas, Android Studio terus berkembang menjadi platform yang sangat andal dan efisien bagi para pengembang aplikasi Android di seluruh dunia. Integrasi tools yang digunakan dalam membuat aplikasi android seperti:

a. IDE (*Integrated Development Environment),* digunakan untuk menulis kode, mendesain antarmuka pengguna dan menguji aplikasi
b. Layout XML, digunakan untuk mendesain antarmuka pengguna (UI) untuk mendefinisikan elemen UI seperti tombol,teks dan gambar
c. PhpMyadmin, alat untuk mengelola database Mysql digunakan untuk membuat tabel, untuk menyimpan data pengguna dan informasi lainnya
d. API (*Apllication Programming Interface*) artinya Antarmuka pemrograman aplikasi yang berfungsi sebagai penghubung antar aplikasi

android dan database.API ini yang akan menangani permintaan dari aplikasi android.

e. JSON (*Java script object Notation*) merupakan format data yang umum yang digunakan untuk komunikasi antara aplikasi dan server, data harus dikembalikan dalam format JSON oleh API

f. *Library QR Qode* untuk membuat *QR Code* yang berisi informasi tiket, gunakan library seperti ZXing atau QRGen. *QR Code* ini dapat dipindai untuk validasi tiket saat Anda masuk ke lokasi wisata.

g. Integrasi *QR Code*, setelah pemesanan tiket berhasil pastikan aplikasi dapat menampilkan QR.

h. Retrofit/Volley, untuk melakukan permintaan HTTP ke API server, gunakan *library* seperti Retrofit atau Volley. Ini akan membuat pengelolaan permintaan dan respons server lebih mudah.

i. AsyncTask, Jika tidak menggunakan *library*, Anda bisa menggunakan AsyncTask untuk melakukan operasi jaringan di background thread.

j. APK Generation, Setelah aplikasi selesai, buat file APK untuk diinstal di perangkat Android.

5. PhpMyAdmin



Gambar 2. 3 PHP MyAdmin
Sumber: https://share.google/images

PhpMyAdmin adalah aplikasi berbasis *open-source* yang tersedia gratis dan dapat digunakan untuk pemrograman serta administrasi database MySQL. Dibangun menggunakan bahasa PHP, PhpMyAdmin mendukung berbagai operasi MySQL, termasuk mengelola basis data, tabel, kolom, relasi, indeks, pengguna, izin akses, dan lainnya. (Mukhaiyar & Ramadhan 2020)

PhpMyAdmin adalah perangkat lunak *open-source* yang dikembangkan menggunakan bahasa pemrograman PHP dan berfungsi untuk mengelola administrasi database MySQL melalui jaringan lokal atau internet. Perbedaan utama antara phpMyAdmin dan MySQL terletak pada fungsinya. phpMyAdmin adalah alat yang mempermudah pengelolaan database MySQL, sementara MySQL sendiri merupakan sistem database tempat data disimpan. phpMyAdmin berfungsi sebagai alat bantu untuk mengelola dan mengatur data yang tersimpan dalam MySQL. (Hartiwati 2022)

6. Java



Gambar 2. 4 Java
Sumber: https://share.google/images

Java adalah bahasa pemrograman yang dapat digunakan di telepon genggam dan berbagai komputer lainnya. Java Virtual Machine (JVM) adalah mesin interpreter yang digunakan untuk menjalankan Java. Ini adalah representasi langsung program dengan bahasa mesin (JVM) yang dapat membaca bytecode dalam file *class program*. Oleh karena itu, bahasa Java dianggap sebagai bahasa pemrograman portabel karena dapat dijalankan pada setiap sistem operasi yang memiliki JVM. (Setiawan, dkk 2020)

Bahasa Java adalah bahasa berorientasi objek yang sederhana dengan banyak kesamaan komponen dengan bahasa C dan C++. Kelebihan bahasa ini termasuk kemampuan untuk beroperasi di berbagai *platform*, kemampuan untuk menerapkan prinsip threading, dan kemampuan untuk meningkatkan kinerja program aplikasi yang dibuat. (Firdaus, 2022).

7. PHP



Gambar 2. 5 *PHP*
Sumber: https://share.google/images

PHP, singkatan dari PHP *Hypertext Preprocessor*, adalah *software open source* yang didistribusikan dan dilisensikan secara gratis dan dapat didownload secara bebas. Digunakan sebagai bahasa *script server-side* dalam pengembangan web, PHP disisipkan pada dokumen HTML. Sahi (2020). PHP, juga dikenal sebagai preprocessor hiperteks, adalah suatu bahasa pemrograman yang digunakan untuk

menterjemahkan basis kode program menjadi kode mesin yang dapat dimengerti oleh komputer di sisi server dan dimasukkan ke dalam HTML.

Aplikasi android yang menggunakan java untuk menghubungkan ke database Mysql melalui PHP (yang di hosting di server) dan PhpMyadmin. Berikut adalah contoh Script PHP yang digunakan untuk menghubungkan database:

config > konek.php > ...

```php
<?php
$database_host = "localhost";
$database_user = "root";
$database_password = "";
$database_nama = "dianawaterpark";

$konek = mysqli_connect($database_host, $database_user, $database_password, $database_nama);

if (!$konek) {
    die("Konek Gagal: " . mysqli_connect_error());
}

mysqli_set_charset($konek, "utf8");
```

Gambar 2. 6 Script *PHP*

8. UML

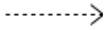
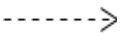
Unified Modeling Language (UML) adalah bahasa pemodelan yang distandarisasi untuk membuat blueprint perangkat lunak. UML dapat digunakan untuk memvisualisasikan, menentukan spesifikasi, membangun, dan mendokumentasikan berbagai komponen dari sistem yang terdapat dalam perangkat lunak (Pressman). (Cahyo, dkk 2021)

Unified Modeling Language (UML) adalah metode pemodelan visual yang digunakan sebagai alat untuk merancang sistem berorientasi objek. UML juga dapat didefinisikan sebagai bahasa standar yang digunakan untuk visualisasi, perancangan, dan pendokumentasian sistem. (Syahputra & Affandi 2018)

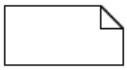
Pada UML memiliki 14 jenis diagram yang digunakan dalam pemodelan perangkat lunak, tetapi dalam Implementasinya terdapat 4 diagram yang paling sering dipakai, yaitu *Use Case Diagram, Class Diagram, Activity Diagram*, dan *Sequence Diagram*.

Berikut ini daftar simbol UML yang digunakan yaitu:

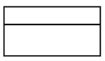
Tabel 2. 1 Simbol *use case diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*Actor*	Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.
2		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*independent*) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*).
3		*Generalization*	Menspesifikasikan bahwa *use case* sumber secara eksplisit.
4		*Include*	Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan.
5		*Extend*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6		*Association*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
7		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor

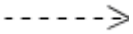
Lanjutan Tabel 2.1

NO	GAMBAR	NAMA	KETERANGAN
8		*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
9		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

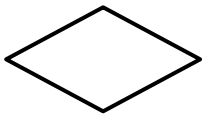
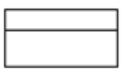
Tabel 2.2 Simbol *Class Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*Generalization*	Hubungan dimana objek anak *(descendent)* berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
2		*Nary Association*	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		*Class*	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		*Collaboration*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu *actor*
5		*Realization*	Operasi yang benar-benar dilakukan oleh suatu objek.

Lanjutan Tabel 2.2

NO	GAMBAR	NAMA	KETERANGAN
6		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
7		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.

Tabel 2.3 Simbol *Activity Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		Status Awal	Sebuah Diagram Aktivitas Memiliki sebuah status awal
2		Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
3		Percabangan/Decision	Percabangan dimana ada pipihan aktivitas yang lebih dari satu
4		Penggabungan/join	Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu
5		Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
6		Swimlane	Swimlane Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Tabel 2.4 Simbol *Sequence Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*LifeLine*	Objek *entity*, antarmuka yang saling berinteraksi.
2		*Message*	Spesifikasidari komunikasi antar objek yang memuat in*form*asi-in*form*asi tentang aktivitas yang terjadi
3		*Message*	Spesifikasi dari komunikasi antar objek yang memuat in*form*asi-in*form*asi tentang aktivitas yang terjadi.

9. Teknologi *QR code*

Quick Response Code atau yang biasa disebut dengan *QR Code* merupakan sebuah *barcode* dua dimensi yang diperkenalkan oleh Perusahaan Jepang Denso Wave pada tahun 1994. *QR Code* adalah image dua dimensi yang merepresentasikan suatu data, terutama data berbentuk teks (Sugiana & Dedi Muhtadi, 2019). Pada dasarnya bahwa *QR Code* dikembangkan sebagai suatu kode yang memungkinkan isinya untuk dapat diterjemahkan dengan kecepatan tinggi (Rouillard, 2008). Keunggulan dari *QR Code* adalah mampu menyimpan informasi secara horizontal dan vertikal. Oleh karena itu, *QR Code* dapat menampung informasi yang lebih banyak dibandingkan dengan barcode satu dimensi. Saat ini, untuk penggunaan *QR Code* telah banyak diimplementasikan dalam bentuk aplikasi *QR Code Reader* dan *QR Code Generator*, sehingga seseorang akan sangat mudah untuk membuat informasi dalam bentuk *QR Code* dan mendapatkan informasi yang ingin diketahuinya, hanya dengan melakukan proses scanning dan pemindaian data melalui media dari kamera handphone.



Gambar 2. 7 *QR Code*
Sumber: https://share.google/images

Ada delapan struktur *Quick Response Code (QR Code)*, yaitu:

a. Pola Pencarian adalah pola yang digunakan untuk mengidentifikasi posisi *QR Code.*

b. Pola timing adalah pola yang digunakan untuk mengidentifikasi koordinat pusat kode QR, yang dibuat dalam format modul hitam-putih bergantian.

c. Informasi versi *QR Code* adalah versinya. Versi terkecil adalah 1 (21 x 21) modul dan versi terbesar adalah 40 (177 x 177) modul.

d. *Quiet Zone* adalah area kosong di bagian terluar Kode QR yang memudahkan sensor CCD untuk mengidentifikasi pengenal QR.

e. Versi *QR Code* adalah versi *QR Code*. Pada gambar ini, versi 3 (29 x 29 modul) digunakan.

f. Data adalah lokasi di mana data disimpan atau dikodekan.

g. Pola aliran digunakan untuk memperbaiki penyimpangan *QR Code*.

h. Format data mencakup informasi tentang tingkat koreksi kesalahan dan pola masking.

QR Code memiliki kemampuan untuk menyimpan banyak data. Secara teoritis, dapat menyimpan hingga 7089 karakter numerik, memiliki kerapatan tinggi

(100 kali lebih tinggi dari kode simbol linier), dan dapat membaca kode dengan cepat. (Clivan, dkk 2019)

QR Code adalah sebuah gambar matriks dua dimensi yang memiliki kemampuan untuk menyimpan data di dalamnya (wijaya & gunawan 2016). *QR Code* ini merupakan hasil evolusi dari kode batang atau *barcode*. *Barcode* sendiri adalah sebuah simbol yang digunakan untuk menandai objek nyata, yang terdiri dari pola batang-batang berwarna hitam dan putih, sehingga memudahkan komputer untuk mengenali dan memproses informasi yang terkandung di dalamnya. perkembangan teknologi dengan *QR Code* telah diperkenalkan sebagai versi yang lebih canggih dan serbaguna dari barcode tradisional, memungkinkan penyimpanan data yang lebih besar dan berbagai aplikasi yang lebih luas Penggunaan teknologi *QR Code* dalam aplikasi pemesanan tiket masuk memberikan efisiensi dan kemudahan bagi pengunjung dan pengelola, Dengan memanfaatkan *QR Code,* aplikasi pemesanan tiket dapat mengurangi kebutuhan akan tiket fisik, mempercepat proses masuk, dan meningkatkan akurasi dalam pengecekan tiket.

QR Code memungkinkan data pemesanan dan validasi tiket tersimpan dalam format yang mudah diakses dan dipindai, baik oleh pengunjung maupun petugas di lapangan. Implementasi ini tidak hanya meningkatkan efisiensi operasional tetapi juga memberikan pengalaman yang lebih baik bagi pengunjung, mengurangi antrian, dan meminimalkan risiko kesalahan manusia dalam proses verifikasi tiket. Penggunaan *QR Code* dalam aplikasi berbasis android yang dapat membantu memudahkan dalam proses pemesanan tiket (Nurmanto & Gunawan, 2020;

Puspaningrum et al., 2020). ini memperlihatkan potensi signifikan untuk modernisasi sistem pemesanan tiket dan manajemen.

B. Penelitian Terdahulu

Penelitian terdahulu merupakan sumber yang penting sebagai referensi, pertimbangan, serta data pendukung dalam pelaksanaan penelitian adapun penelitian terdahulu yang pernah dilakukan yaitu sebagai berikut:

1. Rivaldi Masuara, Rizal Sengkey, Virginia Tulenan (2015) UNSRAT FT Jurusan Teknik Elektro Manado dalam jurnal E-journal Teknik Elektro dan Komputer yang berjudul **"Rancang Bangun e-Ticketing Bioskop Studio 21 Manado Berbasis Multiplatform"** Penelitian ini bertujuan untuk membangun sebuah sistem pemesanan dan pembelian tiket bioskop yang dapat diakses dari berbagai platform (multiplatform). Dengan adanya penelitian ini, dapat dibuat suatu perangkat lunak untuk memudahkan costumer dalam melakukan suatu proses bisnis pemesanan dan pembelian tiket bioskop dengan menggunakan perangkat yang telah didukung dengan fasilitas internet. Sehingga dapat memberikan kemudahan dan efektifitas waktu bagi setiap pelaku bisnis.
2. Asep Id Hadiana (2016) UNJANI FMIPA Jurusan Teknik Informatika dalam jurnal dengan judul **"Pemanfaatan Teknologi QR Code Untuk Verifikasi Akta Notaris (PPAT)"** penelitian ini bertujuan untuk merancang sistem verifikasi keaslian akta Notaris/PPAT menggunakan *QR Code* sebagai penanda, sehingga pemalsuan dokumen oleh pihak yang tidak bertanggung jawab dapat diminimalisir. Selain itu, penelitian ini juga bertujuan untuk meningkatkan keamanan dan efisiensi dalam pengecekan keaslian dokumen legal melalui

teknologi *QR Code*, yang memungkinkan penyimpanan informasi lebih banyak dan dapat diakses secara cepat oleh pihak yang berwenang.

BAB III

METODE PENELITIAN

A. Jenis Penelitian

Jenis yang digunakan pada penelitian ini adalah Jenis Penelitian kualitatif yang mana penelitian ini digunakan untuk meneliti kondisi objek yang alamiah, di mana peneliti adalah sebagai instrumen kunci, teknik pengumpulan data dilakukan secara triangulasi, analisis data bersifat induktif.

B. Lokasi dan Waktu

Lokasi penelitian dilaksanakan di wilayah Desa Balusu, kecamatan Balusu, Kabupaten Barru, dan waktu penelitian dilakukan selama ± 1 (satu) bulan.

C. Alat dan Bahan

Untuk melakukan proses penelitian dalam pembuatan aplikasi, maka diperlukan perangkat keras dan perangkat lunak guna mendukung kegiatan penelitian tersebut. Berikut ini merupakan penjelasan dari *hardware* dan *software* yang digunakan.

1. Perangkat keras

Perangkat keras yang digunakan untuk membuat aplikasi tersebut yaitu dapat dilihat pada tabel dibawah ini:

Tabel 3. 1 Spesifikasi *Hardware*

Spesifikasi	
Merk Laptop	HP
Processor Laptop	Athlon Silver
RAM Laptop	4 GB

2. Perangkat lunak

Perangkat lunak yang digunakan untuk membuat aplikasi tersebut yaitu dapat dilihat pada tabel dibawah ini:

Tabel 3. 2 Spesifikasi *Software*

Spesifikasi	
Sistem Operasi	Windows 11
Tool Pemrograman	*Android Studio*
Software basis data	*PhpMyadmin*
Bahasa Pemrograman	*Java, PHP*

D. Metode Pengumpulan Data

Pada tahap ini dilakukan dengan mempelajari teori dasar yang mendukung penelitian, pencarian dan pengumpulan data-data yang dibutuhkan. Untuk mengumpulkan data yang dibutuhkan, maka penulis menggunakan beberapa teknik.

1. Pengamatan langsung (*Observation)* Penelitian melakukan pengamatan langsung pada objek wisata diana waterpark untuk mendapat data yang berkaitan dengan penelitian.
2. Wawancara (*Interview*) Peneliti secara langsung bertatap muka dengan pengelola untuk mendapatkan data-data yang lebih lengkap dan jelas melalui wawancara yaitu pengenalan objek wisata.

B. Metode Pengujian

Pengujian sistem dilakukan dengan menguji kode program serta mengimplementasikan hasilnya melalui *test-case* di lingkungan penelitian. Jika struktur program sudah baik, pengujian unit menjadi sangat penting. Pengujian unit bertujuan untuk memastikan setiap modul berfungsi dengan benar. Terdapat dua metode dalam melakukan pengujian unit, yaitu:

1. *White box testing*

Pengujian *whitebox* dikenal sebagai pengujian struktural di mana perangkat lunak yang diuji transparan bagi penguji. *white box testing* adalah pengujian perangkat lunak pada tingkat alur kode program, apakah masukan dan keluaran yang sesuai dengan spesifikasi yang dibutuhkan. (Cholifah, Yulianingsih, & Sagita, 2018). Berbagai tipe kesalahan bisa ditemukan, yang bisa menyebabkan asumsi

yang salah dalam program. Pengujian *white box* diterapkan di semua tingkatan perancangan perangkat lunak. Penguji harus meninjau kode sumber dan mengidentifikasi bagian mana yang tidak berfungsi dengan baik. *White box* testing memerlukan akses ke kode sumber dan dilakukan selama pengembangan perangkat lunak. Pengujian ini juga dikenal sebagai pengujian *clear box*, *glass box*, atau *open box* Desikan dkk (2008).

2. *Black box testing*

Pengujian *blackbox* dikenal sebagai pengujian perilaku, di mana struktur internal dan logika perangkat lunak yang diuji tidak diketahui oleh penguji. Pengujian ini didasarkan pada spesifikasi kebutuhan tanpa perlu menganalisis kode. Black box testing adalah metode pengujian perangkat lunak yang tes fungsionalitas dari aplikasi yang bertentangan dengan struktur internal atau kerja (Puspitasari dkk, 2020). *Black box testing* memiliki kelebihan dan kekurangan dalam implementasinya. Salah satu kelebihannya adalah membantu menemukan aspek-aspek yang tidak terpenuhi dari spesifikasi kebutuhan yang diberikan dalam pengembangan perangkat lunak. Namun, kekurangannya adalah pengujian tidak dapat dilakukan secara menyeluruh karena pengetahuan penguji tentang perangkat lunak yang diuji terbatas.

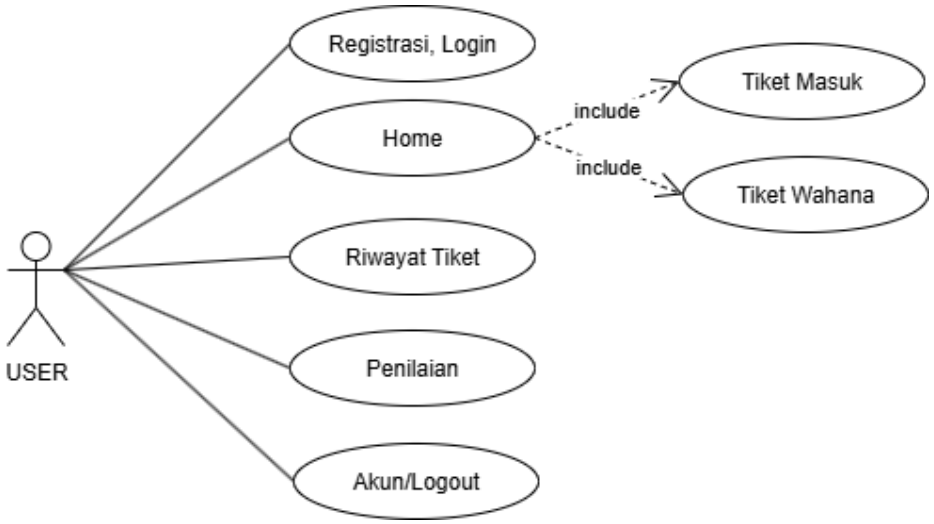
BAB IV

HASIL DAN PEMBAHASAN

A. Analisis Aliran Data UML

1. *Use Case Diagram*

Use case diagram adalah salah satu jenis diagram dalam Unified Modeling Language (UML) yang digunakan untuk menggambarkan interaksi antara pengguna atau aktor dengan sistem yang akan dibangun.



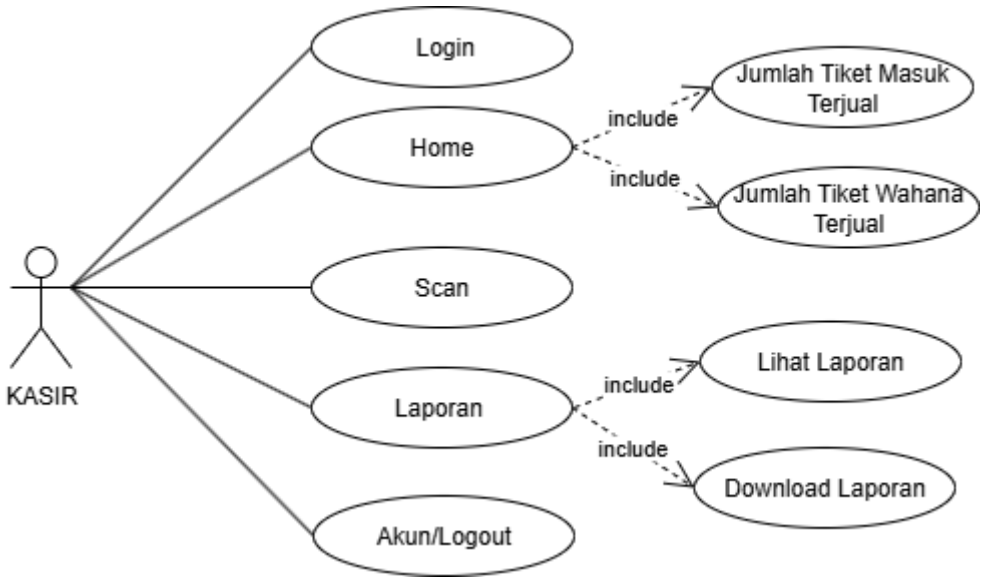
Gambar 4.1 *Use case diagram user*

Tabel 4.1 *Use case* skenario *user*

Nama Use Case	Deskripsi
Menu Register, Login	*Use case* ini menjelaskan proses dimana pengguna dapat melakukan register terlebih dahulu. Dimana ketika pengguna telah melakukan registrasi selanjutnya akan masuk ke menu login. Tujuan dari proses ini adalah untuk memastikan bahwa hanya pengguna terdaftar yang dapat mengakses sistem.

Lanjutan Tabel 4.1

Nama Use Case	Deskripsi
Menu Home	*Use case ini* menjelaskan proses dimana pengguna dapat memilih jenis tiket sesuai dengan yang dibutuhkan apakah itu tiket masuk maupun tiket wahana. Tujuan dari proses ini adalah untuk memudahkan pengguna dalam menentukan dan memesan tiket yang diinginkan dengan tepat.
Menu Riwayat Tiket	*Use Case* ini menjelaskan proses dimana pengguna dapat melihat Riwayat tiket dalam bentuk QR Code.Tujuan dari proses ini adalah sebagai bukti bahwa pengguna telah melakukan pembelian tiket sebagai syarat izin untuk masuk kedalam wisata tersebut maupun wahana yang ada di dalam wisata tersebut.
Penilaian	*Use Case* ini menjelaskan proses dimana pengguna dapat memberikan penilaian/rating berupa bintang dan juga komentar. Tujuan dari proses ini adalah untuk memberikan evaluasi kepada pihak dianawaterpark mengenai kepuasan ataupun keluhan pengunjung.
Menu Akun/Logout	*Use case* ini menjelaskan proses di mana pengguna dapat melihat informasi akun pribadi dan melakukan logout dari sistem. Tujuan dari proses ini adalah untuk memberikan kemudahan dalam keamanan data pengguna setelah selesai menggunakan aplikasi.



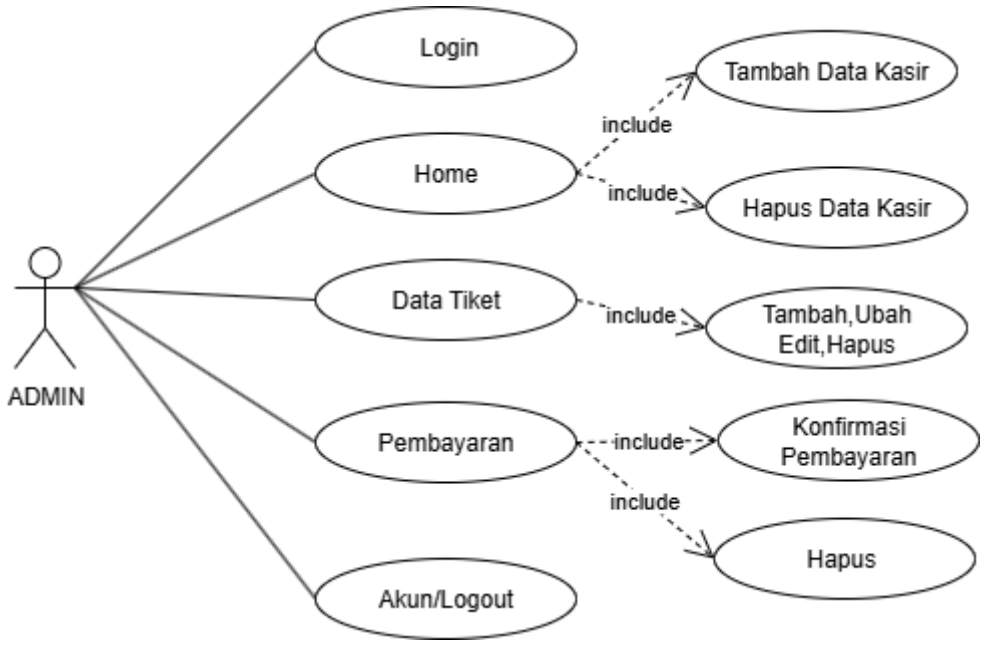
Gambar 4.2 *Use case diagram* kasir

Tabel 4.2 *Use Case* skenario kasir

Nama Use Case	Deskripsi
Menu Login	*Use case* ini menjelaskan proses dimana kasir harus melakukan login terlebih dahulu. Setelah itu akan diarahkan ke menu tampilan kasir . Tujuan dari proses ini adalah Memastikan bahwa hanya kasir yang memiliki akun terverifikasi yang dapat mengakses sistem kasir.
Menu Home	*Use case ini* menjelaskan proses dimana kasir dapat melihat jumlah serta jenis tiket yang sudah terjual apakah itu tiket masuk maupun tiket wahana.Tujuan dari proses ini adalah untuk memudahkan kasir dalam menentukan dan mengetahui jumlah tiket yang terjual apakah itu tiket masuk maupun tiket wahana.

Lanjutan Tabel 4.2

Nama Use Case	Deskripsi
Menu Scan	*Use case ini* menjelaskan proses dimana kasir melakukan pemindaian *QR Code* menggunakan kamera bawaan handphone . Tujuan dari proses ini adalah untuk memudahkan kasir dalam memverifikasi serta menentukan dan mengetahui jenis tiket yang telah di beli.
Laporan	*Use case* ini menjelaskan proses dimana kasir dapat mengakses melihat dan mendownload laporan transaksi,jumlah pengunjung dan jenis tiket. Laporan dapat ditampilkan berdasarkan rentang waktu tertentu, seperti per hari, per bulan, atau per tahun. Kasir dapat memilih jenis laporan yang diinginkan dengan menginput tanggal, bulan, dan tahun sesuai kebutuhan.Tujuan dari proses ini adalah memberikan kemudahan dalam memantau transaksi berdasarkan periode waktu
Menu Akun/LogOut	*Use case* ini menjelaskan proses di mana kasir dapat melihat informasi akun pribadi dan melakukan logout dari sistem. Tujuan dari proses ini adalah untuk memberikan kemudahan dalam keamanan data pengguna setelah selesai menggunakan aplikasi.



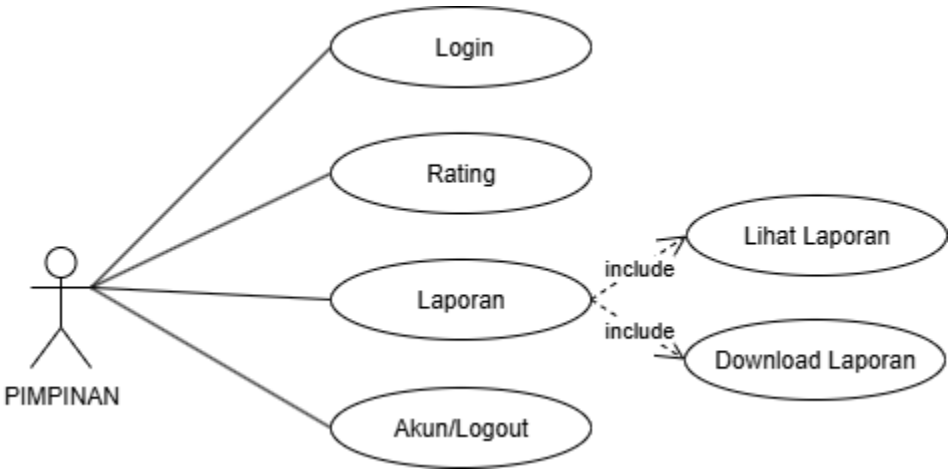
Gambar 4.3 *Use case diagram* admin

Tabel 4.3 *Use case* skenario admin

Nama Use Case	Deskripsi
Menu Login	*Use case* ini menjelaskan proses dimana admin harus melakukan login terlebih dahulu. Setelah itu akan diarahkan ke menu tampilan admin . Tujuan dari proses ini adalah Memastikan bahwa hanya admin yang memiliki akun terverifikasi yang dapat mengakses sistem admin.
Menu Home	*Use case* ini menjelaskan proses di mana admin dapat mengelola data kasir dan pengguna. Admin dapat melihat jumlah kasir yang terdaftar, menambah kasir baru, serta menghapus data kasir yang sudah ada. Selain itu, admin juga dapat melihat data pengguna atau user. Tujuan dari proses ini adalah untuk memudahkan admin dalam mengelola data kasir dan dapat melihat data pengguna/user secara efektif.

Lanjutan Tabel 4.3

Nama Use Case	Deskripsi
Menu Data Tiket	*Use case* ini menjelaskan proses di mana admin dapat mengelola data tiket masuk maupun tiket wahana seperti dapat menambahkan tiket wahana, mengedit tiket wahana yang sudah ada serta dapat menghapus wahana yang di inginkan jika sudah tidak terpakai serta dibagian tiket masuk dapat mengubah harga tiket jika suatu waktu ada perubahan harga.Tujuan dari proses ini adalah untuk memastikan data tiket selalu akurat dan dapat disesuaikan dengan kebutuhan operasional atau perubahan kebijakan.
Pembayaran	*Use case* ini menjelaskan proses diaman admin dapat menekan button konfirmasi sebagai bukti bahwa user telah berhasil melakukan transaksi pembelian tiket serta admin dapat menghapus pembelian jika user membatalkan pembelian tiket.
Menu Akun/Logout	*Use case* ini menjelaskan proses di mana admin dapat melihat informasi akun pribadi dan melakukan logout dari sistem. Tujuan dari proses ini adalah untuk memberikan kemudahan dalam keamanan data pengguna setelah selesai menggunakan aplikasi.



Gambar 4.4 *Use case diagram* pimpinan

Tabel 4.4 *Use case diagram* pimpinan

Nama Use Case	Deskripsi
Menu Login	*Use case* ini menjelaskan proses dimana pimpinan harus melakukan login terlebih dahulu. Setelah itu akan diarahkan ke menu tampilan pimpinan .Tujuan dari proses ini adalah Memastikan bahwa hanya pimpinan yang memiliki akun terverifikasi yang dapat mengakses sistem pimpinan.
Rating	*Use case* ini menjelaskan proses di mana pimpinan dapat melihat penilaian atau rating yang diberikan oleh pengguna, baik yang bersifat positif maupun negatif. Tujuan dari proses ini adalah untuk membantu pengelola Diana Waterpark dalam mengidentifikasi kelebihan dan kekurangan yang ada, sehingga dapat dilakukan evaluasi secara menyeluruh terhadap sistem maupun layanan yang masih kurang optimal. Dengan demikian, perbaikan dapat dilakukan guna meningkatkan kualitas pelayanan di masa mendatang.

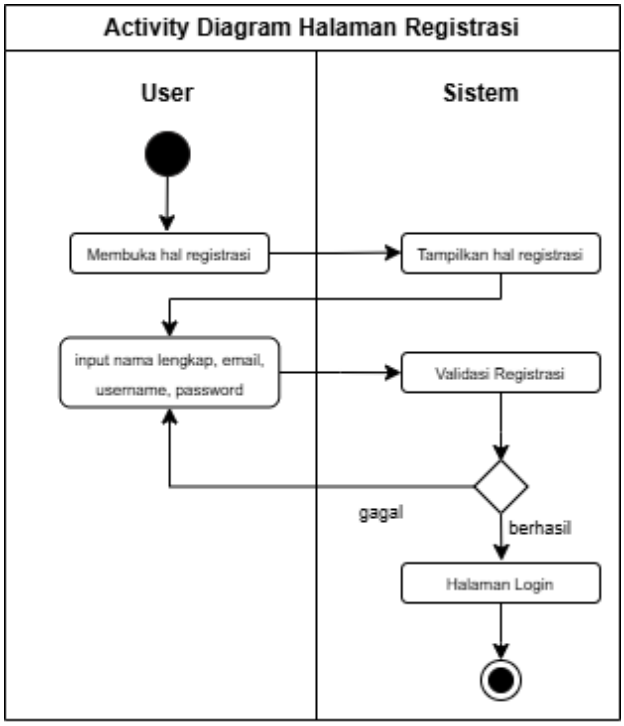
Lanjutan Tabel 4.4

Nama Use Case	Deskripsi
Menu Laporan	*Use case* ini menjelaskan proses dimana pimpinan dapat mengakses melihat dan mendownload laporan transaksi,jumlah pengunjung dan jenis tiket. Laporan dapat ditampilkan berdasarkan rentang waktu tertentu, seperti per hari, per bulan, atau per tahun. pimpinan dapat memilih jenis laporan yang diinginkan dengan menginput tanggal, bulan, dan tahun sesuai kebutuhan.Tujuan dari proses ini adalah memberikan kemudahan dalam memantau transaksi berdasarkan periode waktu.
Menu Akun/Logout	*Use case* ini menjelaskan proses di mana pimpinan dapat melihat informasi akun pribadi dan melakukan logout dari sistem. Tujuan dari proses ini adalah untuk memberikan kemudahan dalam keamanan data pengguna setelah selesai menggunakan aplikasi.

2. *Activity Diagram*

Activity Diagram adalah diagram yang digunakan untuk memodelkan berbagai proses yang berlangsun dalam suatu sistem.

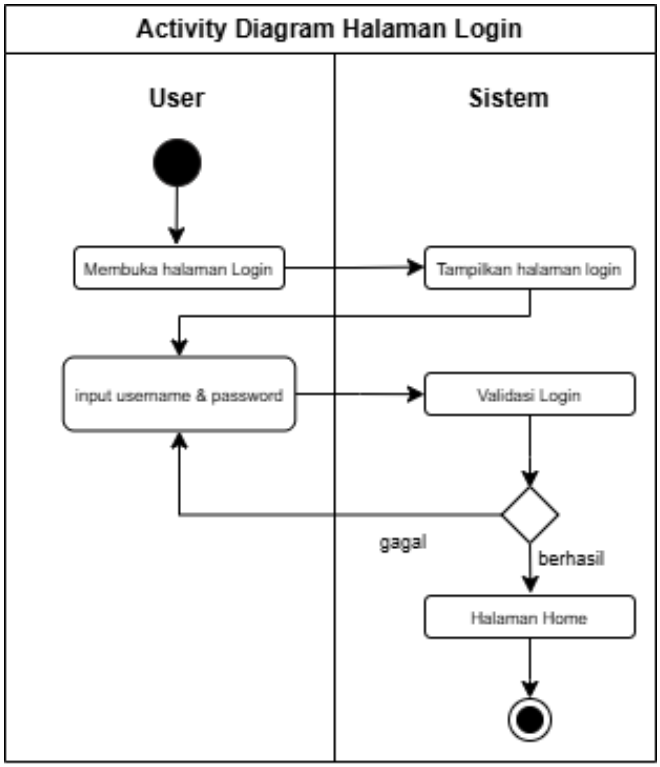
a. *Activity Diagram* Halaman *Registrasi User*



Gambar 4.5 *Activity diagram* halaman *registrasi user*

Gambar 4.5 *Activity diagram* di atas menunjukkan alur registrasi oleh admin, yang terbagi menjadi dua peran: Admin dan Sistem. Proses dimulai saat admin membuka halaman registrasi, kemudian sistem menampilkannya. Admin mengisi data seperti nama lengkap, *email, username,* dan *password,* lalu sistem memvalidasi data tersebut. Jika validasi gagal, admin diminta mengulang pengisian data. Jika berhasil, sistem mengarahkan ke halaman login sebagai tanda registrasi sukses.

b. *Activity Diagram* Halaman *Login User*

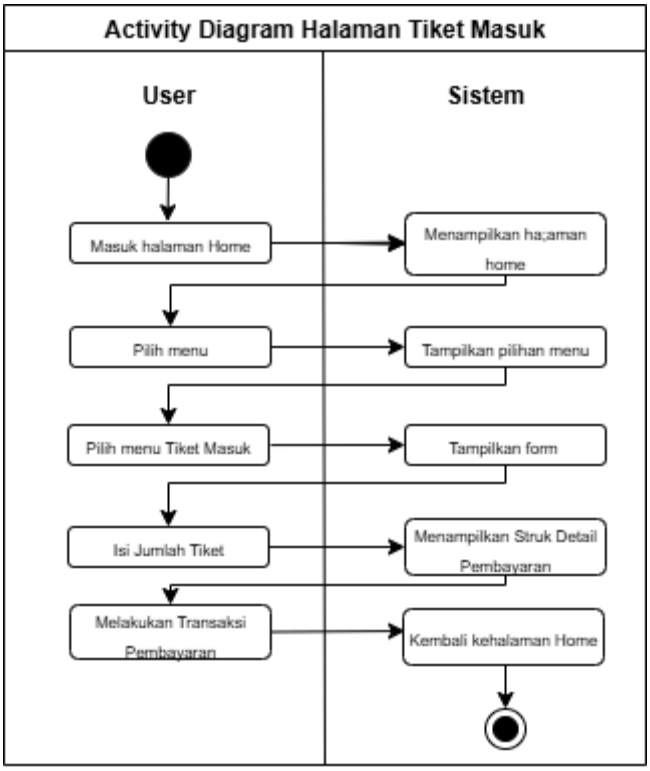


Gambar 4.6 *Activity diagram* halaman *login user*

Gambar 4.6 Proses dimulai ketika user membuka halaman *login*. Setelah itu, sistem merespon dengan menampilkan halaman login tersebut. User kemudian mengisi username dan password di halaman yang telah ditampilkan, lalu mengirimkan data tersebut ke sistem untuk dilakukan proses validasi login.

Sistem akan memeriksa apakah *username* dan *password* yang dimasukkan sesuai dengan data yang ada di database. Jika proses validasi gagal, user akan diarahkan kembali untuk menginput ulang data login yang benar. Namun, jika validasi berhasil, sistem akan mengarahkan *user* ke halaman *home* sebagai tanda bahwa proses *login* telah berhasil.

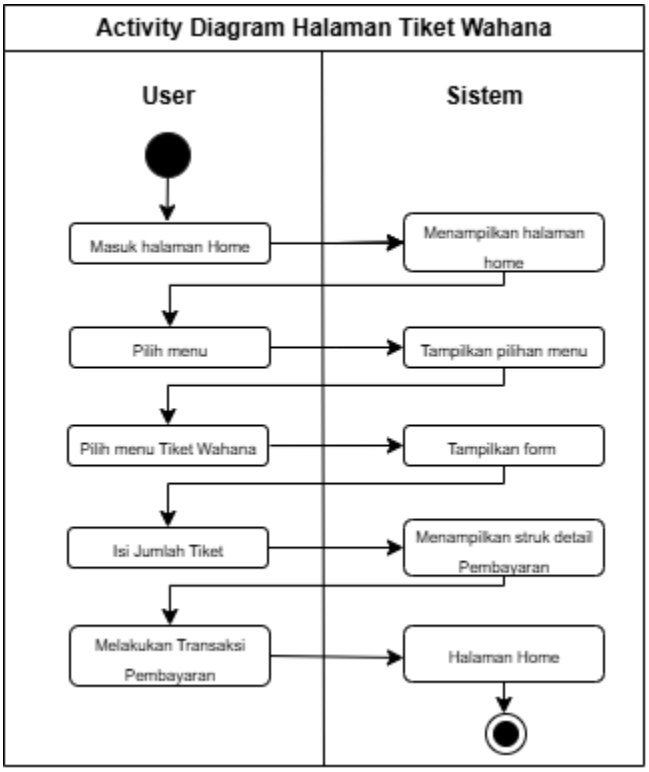
c. *Activity Diagram* Halaman Tiket Masuk *User*



Gambar 4.7 *Activity diagram* halaman tiket masuk *user*

Gambar 4.7 Proses dimulai ketika *user* masuk ke halaman *home*. Sistem kemudian menampilkan halaman home sebagai tampilan awal. Setelah itu, user memilih menu yang tersedia, dan sistem merespons dengan menampilkan pilihan menu. Selanjutnya, ketika user memilih menu “Tiket Masuk”, lalu sistem menampilkan form yang berkaitan dengan pemesanan tiket masuk. Setelah form ditampilkan, pengguna mengisi jumlah tiket yang ingin dipesan. Sistem kemudian menampilkan struk detail pembayaran sebagai langkah selanjutnya *user* akan melakukan transaksi pembayaran yang sesuai setelah *user* melakukan transaksi pembayaran yang tertera selanjutnya sistem akan memproses pembayaran nya dan setelah berhasil maka selanjutnya akan di arahkan kembali kemenu home

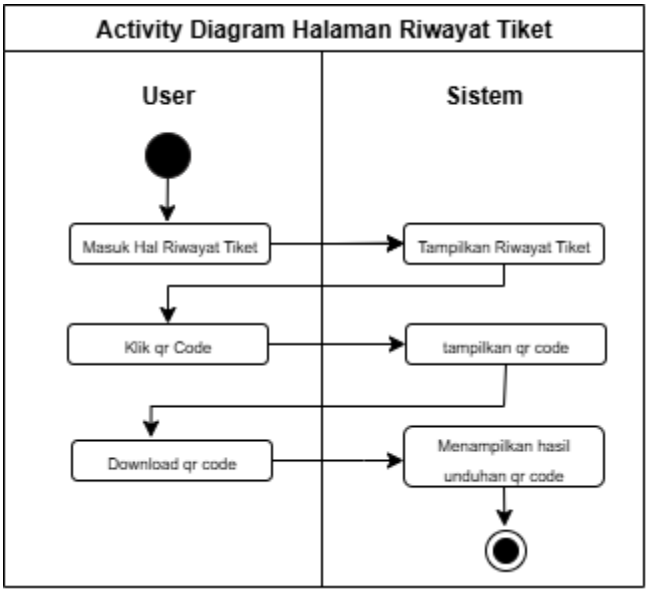
d. *Activity Diagram* Halaman Tiket Wahana *User*



Gambar 4.8 *Activity diagram* halaman tiket wahana *user*

Gambar 4.8 Proses dimulai ketika user masuk ke halaman home. Sistem kemudian menampilkan halaman home sebagai tampilan awal. Setelah itu, user memilih menu yang tersedia, dan sistem merespons dengan menampilkan pilihan menu. Selanjutnya, ketika user memilih menu “Tiket Wahana”, lalu sistem menampilkan form yang berkaitan dengan pemesanan tiket wahana. Setelah form ditampilkan, pengguna memilih jenis tiket wahana dan mengisi jumlah tiket yang ingin dipesan. Sistem kemudian menampilkan struk detail pembayaran sebagai langkah selanjutnya user akan melakukan transaksi pembayaran yang sesuai setelah user melakukan transaksi pembayaran yang tertera selanjutnya sistem akan memproses pembayaran nya dan setelah berhasil maka selanjutnya akan di arahkan kembali kemenu home

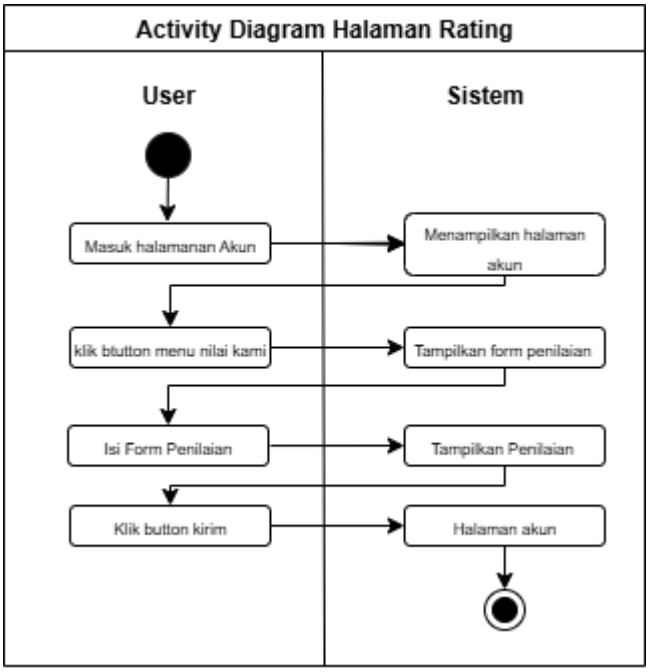
e. *Actvity Diagram* Halaman Riwayat Tiket *User*



Gambar 4.9 *Activity diagram* halaman riwayat tiket *user*

Gambar 4.9 Proses dimulai ketika user masuk ke halaman riwayat tiket. Setelah itu, sistem merespons dengan menampilkan data riwayat tiket yang dimiliki oleh user tersebut selanjutnya user dapat melihat tiket terbaru dan tiket yang sudah digunakan. Setelah itu user dapat mengklik gambar *QR code* tersebut. Selanjutnya sistem akan menampilkan *QR code* yang dipilih dan akan muncul button download gambar kemudian user memilih download *QR code* sistem akan melakukan proses pengunduhan dalam bentuk gambar untuk mengunduh *QR code* sesuai permintaan *user.*

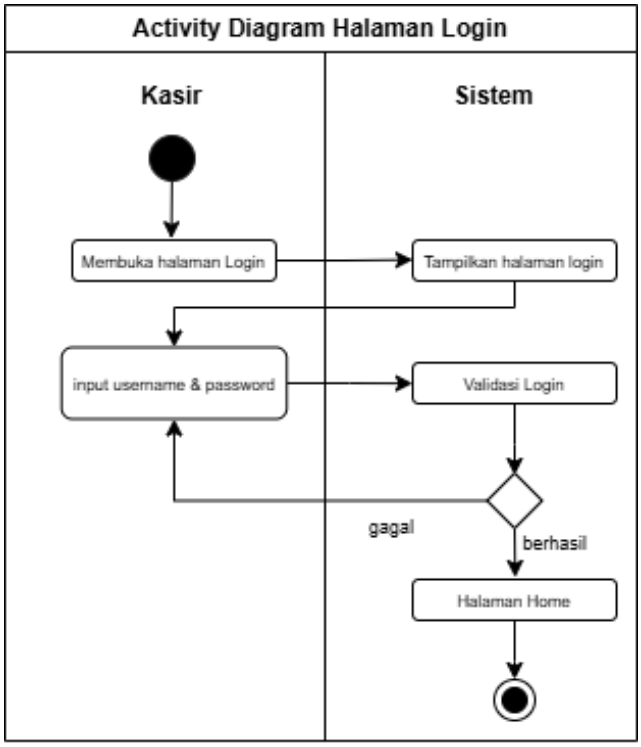
f. *Activity Diagram* Halaman *Rating*



Gambar 4.10 *Activity diagram* halaman *rating*

Gambar 4.10 Proses dimulai ketika *user* masuk ke halaman akun, kemudian sistem merespons dengan menampilkan halaman akun yang berisi menu logout dan nilai kami yang tersedia. Setelah itu, user mengklik button menu "nilai kami" untuk memberikan penilaian terhadap layanan. Sistem kemudian akan menampilkan form penilaian yang dapat diisi oleh user. Selanjutnya, user mengisi form penilaian sesuai dengan pengalaman mereka, lalu sistem akan menampilkan penilaian yang telah diisi tersebut sebagai pratinjau sebelum dikirimkan. Setelah mengecek, user mengklik *button* kirim, dan sistem akan memproses data penilaian dan mengembalikan *user* ke halaman akun sebagai akhir dari proses.

g. *Activity Diagram* Halaman *Login* Kasir

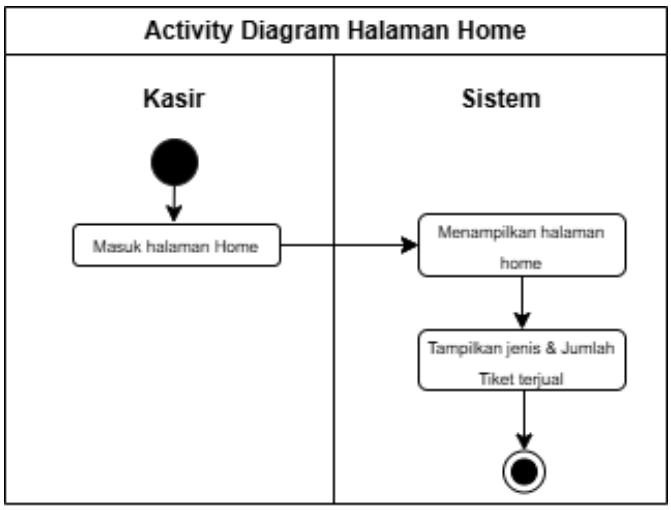


Gambar 4.11 *Activiy diagram* halaman *login* kasir

Gambar 4.11 Proses dimulai ketika kasir membuka halaman *login*. Setelah itu, sistem merespon dengan menampilkan halaman *login* tersebut. kasir kemudian mengisi username dan password di halaman yang telah ditampilkan, lalu mengirimkan data tersebut ke sistem untuk dilakukan proses validasi *login.*

Sistem akan memeriksa apakah *username* dan *password* yang dimasukkan sesuai dengan data yang ada di database. Jika proses validasi gagal, kasir akan diarahkan kembali untuk menginput ulang data login yang benar. Namun, jika validasi berhasil, sistem akan mengarahkan kasir ke halaman *home* sebagai tanda bahwa proses *login* telah berhasil.

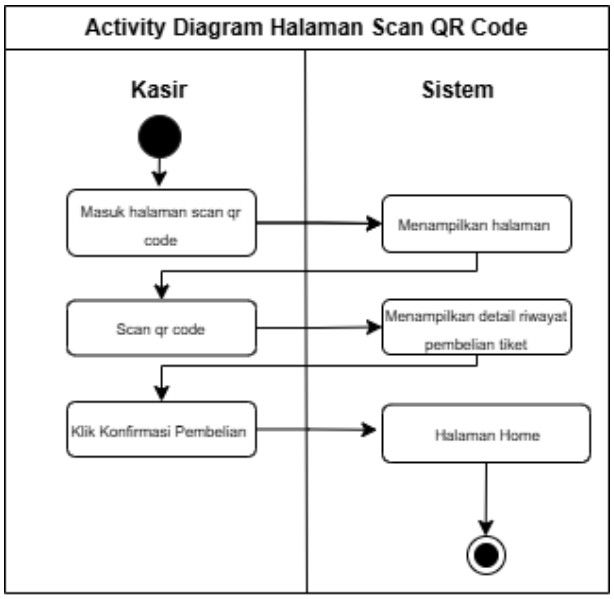
h. *Activity Diagram* halaman *Home* Kasir



Gambar 4.12 *Activity diagram* halaman *home* kasir

Gambar 4.12 Proses dimulai ketika kasir membuka halaman *home.* Setelah itu, sistem merespon dengan menampilkan halaman home tersebut dan akan tampil jenis dan jumlah total tiket yang terjual baik itu total tiket masuk yang terjual maupun total tiket wahana yang terjual.

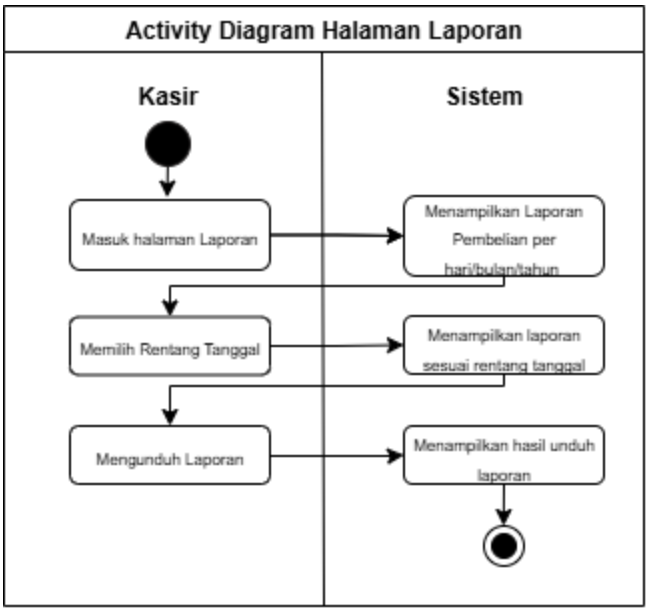
i. *Activity Diagram* Halaman *Scan* Kasir



Gambar 4.13 Activity *diagram* halaman *scan* kasir

Gambar 4.13 Proses dimulai ketika kasir mengakses halaman *scan QR code*. Setelah kasir masuk ke halaman *scan QR code*, sistem secara otomatis merespons dengan menampilkan halaman tersebut. Selanjutnya, kasir melakukan proses pemindaian *QR code*. Setelah *QR code* dipindai, sistem akan menampilkan detail riwayat pembelian tiket berdasarkan informasi yang didapat dari *QR code* tersebut. Kemudian, kasir melakukan klik pada tombol "Konfirmasi Pembelian". lalu sistem mengarahkan atau mengembalikan kasir ke halaman utama (*Home*).

j. *Activity Diagram* Halaman Laporan Kasir

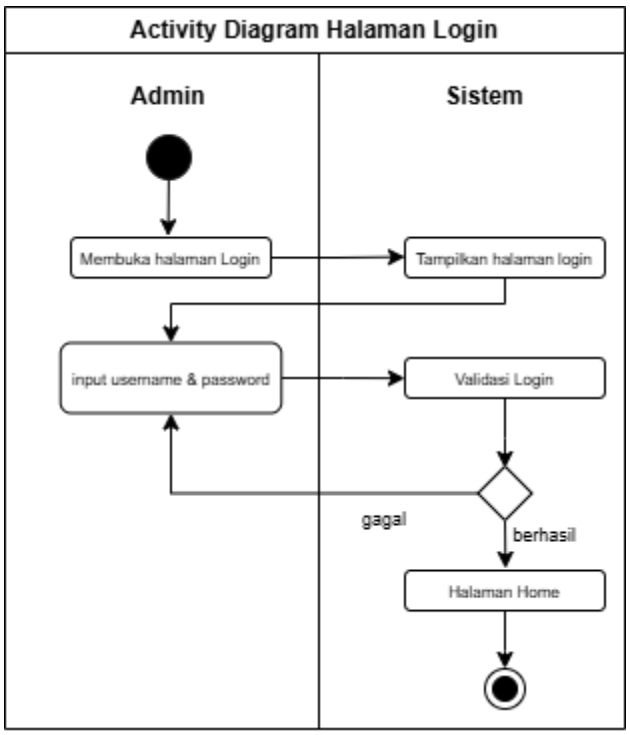


Gambar 4. 14 *Activity diagram* halaman laporan

Gambar 4.14 Proses dimulai ketika kasir mengakses halaman laporan. Setelah berhasil masuk ke halaman tersebut, sistem secara otomatis merespons dengan menampilkan laporan pembelian yang tersedia dalam pilihan periode harian, bulanan, atau tahunan. Apabila kasir ingin melihat laporan yang lebih spesifik, ia dapat menentukan rentang tanggal sesuai dengan kebutuhan. Rentang tanggal yang dipilih ini kemudian dikirimkan ke sistem. Selanjutnya, sistem akan memproses

permintaan tersebut dan menampilkan laporan pembelian yang sudah difilter berdasarkan rentang tanggal yang telah dipilih oleh kasir. Setelah laporan ditampilkan, kasir dapat mengunduh laporan tersebut sesuai kebutuhan. Setelah itu sistem akan menampilkan hasil unduhan laporan tersebut.

k. *Activity Diagram* Halaman *Login* Admin



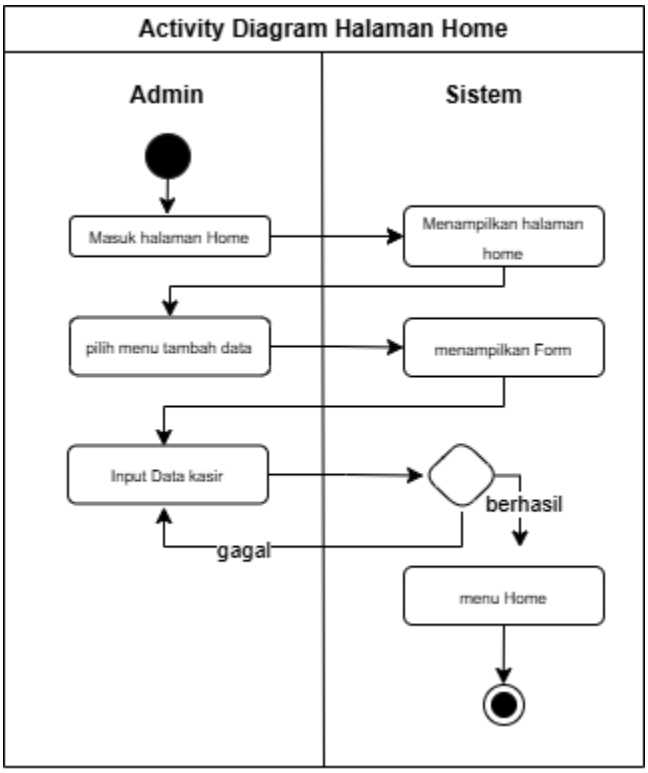
Gambar 4.15 *Activity diagram* halaman *login* admin

Gambar 4.15 Proses dimulai ketika admin membuka halaman *login*. Setelah itu, sistem merespon dengan menampilkan halaman login tersebut. admin kemudian mengisi *username* dan *password* di halaman yang telah ditampilkan, lalu mengirimkan data tersebut ke sistem untuk dilakukan proses validasi *login*.

Sistem akan memeriksa apakah *username* dan *password* yang dimasukkan sesuai dengan data yang ada di database. Jika proses validasi gagal, admin akan diarahkan kembali untuk menginput ulang data *login* yang benar. Namun, jika validasi

berhasil, sistem akan mengarahkan *admin* ke halaman *home* sebagai tanda bahwa proses *login* telah berhasil.

l. *Activity Diagram* Halaman *Home Admin*

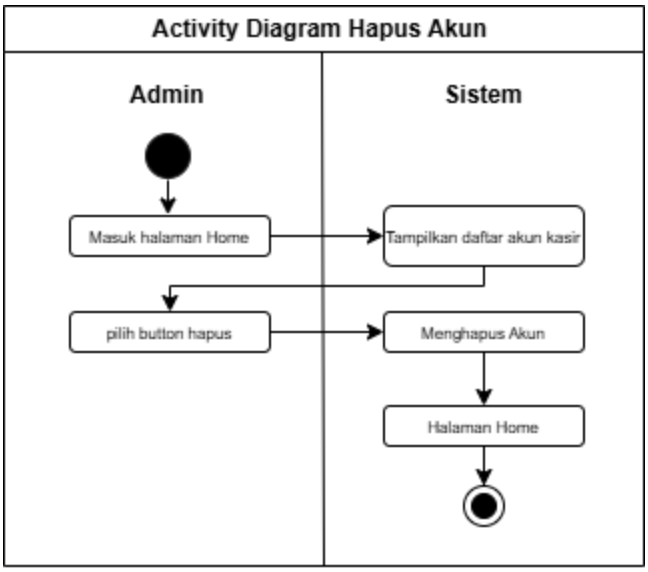


Gambar 4.16 *Activity diagram* halaman *home* tambah data kasir

Gambar 4.16 Proses dimulai dengan *Admin* yang masuk ke halaman *Home*. Setelah itu, sistem merespons dengan menampilkan halaman *Home*. Langkah berikutnya, Admin memilih menu "tambah data" yang tersedia di halaman. Selanjutnya, sistem menampilkan form yang akan digunakan untuk menginput data kasir. Selanjutnya, *Admin* mengisi *form* dengan data kasir yang ingin ditambahkan. Setelah data diinput, sistem akan melakukan validasi atau pengecekan terhadap data yang dimasukkan. Apabila data yang dimasukkan sama dengan data sebelumnya, tidak valid atau proses gagal, maka sistem tidak melanjutkan proses ke halaman

berikutnya dan pengguna tetap berada pada proses input data. Namun jika proses input data berhasil, maka sistem akan mengarahkan kembali ke menu *Home.*

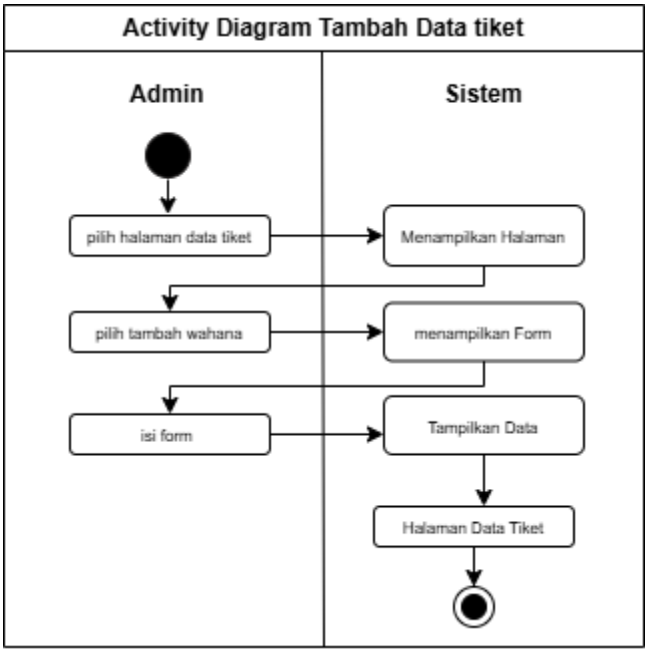
m. *Activity Diagram* Hapus Akun



Gambar 4.17 *Activity diagram* hapus akun

Gambar 4.17 proses dimulai saat Admin masuk ke halaman *Home*. Setelah masuk, sistem menampilkan daftar akun kasir yang tersedia untuk dikelola. Selanjutnya, *Admin* memilih tombol hapus yang tersedia pada salah satu akun dalam daftar tersebut. Ketika tombol hapus dipilih, sistem akan menjalankan proses penghapusan akun kasir yang dipilih. Setelah akun berhasil dihapus, sistem akan mengarahkan kembali ke halaman *Home.*

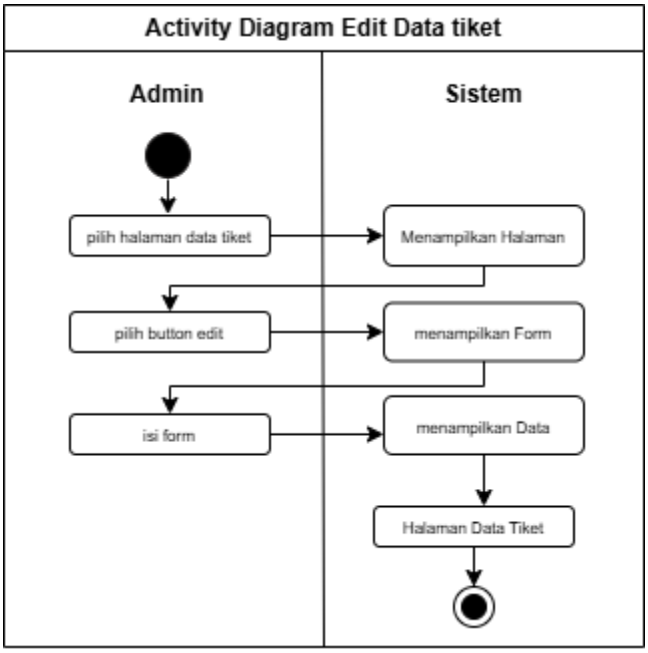
n. *Activity Diagram* Data Tiket Wahana



Gambar 4.18 *Activity diagram* tambah data tiket

Gambar 4.18 Proses dimulai ketika *Admin* memilih halaman data tiket. Setelah itu, sistem merespons dengan menampilkan halaman data tiket. Setelah halaman ditampilkan, *Admin* kemudian memilih button tambah wahana untuk menambahkan wahana baru. Sistem merespon dengan menampilkan *form input* yang digunakan untuk mengisi data tiket wahana yang akan ditambahkan. *Admin* lalu mengisi sesuai form tersebut. Setelah data diisi, sistem akan menampilkan data yang telah dimasukkan ke dalam sistem dan mengarahkan kembali ke halaman data tiket, menandakan bahwa proses penambahan tiket telah selesai dan berhasil dijalankan.

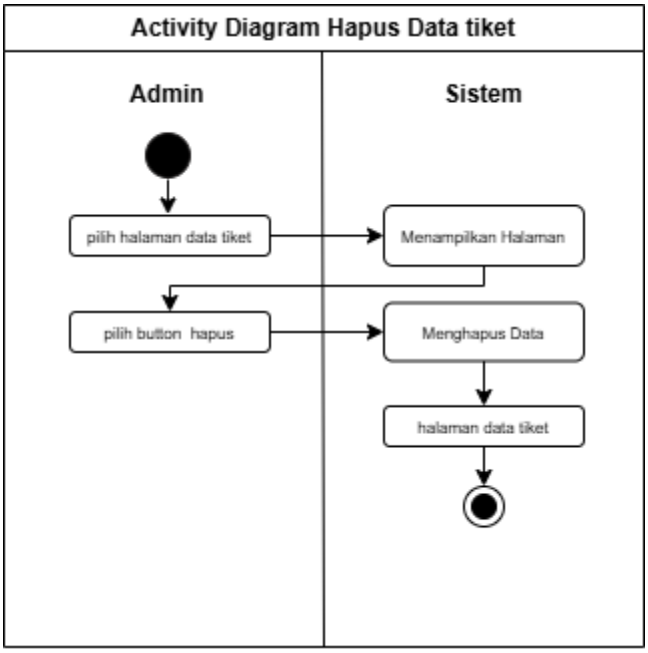
o. *Activity Diagram* Edit Data Tiket Wahana



Gambar 4.19 *Activity diagram* edit data tiket wahana

Gambar 4.19 Proses dimulai dengan *Admin* memilih halaman data tiket. Setelah itu, sistem merespons dengan menampilkan halaman data tiket. Setelah halaman tampil, *admin* kemudian menekan tombol edit pada salah satu data tiket yang ingin diubah. Sistem kemudian menampilkan form untuk pengisian atau pengeditan data. *Admin* lalu mengisi form tersebut sesuai dengan data yang ingin diperbarui. Setelah *form* diisi, sistem menampilkan data yang telah diperbarui sebagai konfirmasi bahwa perubahan telah berhasil dilakukan. Terakhir, sistem mengarahkan kembali ke halaman data tiket.

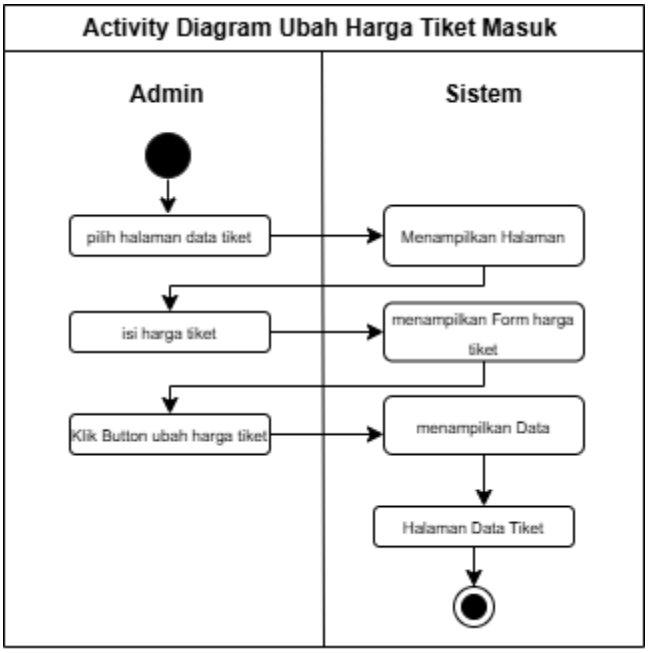
p. *Activity Diagram* Hapus Data Tiket Wahana



Gambar 4.20 *Activity diagram* hapus data tiket wahana

Gambar 4.20 proses dimulai saat Admin masuk ke halaman Data Tiket. Setelah masuk, sistem menampilkan daftar data tiket yang tersedia untuk dikelola. Selanjutnya, *Admin* memilih tombol hapus yang tersedia pada salah satu data tiket wahana dalam daftar tersebut. Ketika tombol hapus dipilih, sistem akan menjalankan proses penghapusan wahana yang dipilih. Setelah wahana berhasil dihapus, sistem akan mengarahkan kembali ke halaman data tiket.

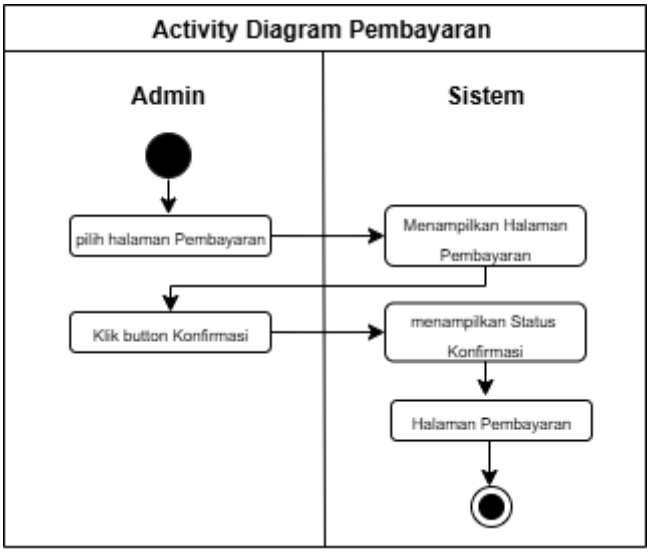
q. *Activity Diagram* Ubah Harga Tiket Masuk



Gambar 4.21 *Activity diagram* ubah harga tiket masuk

Gambar 4.20 Proses dimulai saat *Admin* masuk ke halaman Data Tiket. Setelah itu, sistem menampilkan halaman yang berisi data tiket yang tersedia. Kemudian, *Admin* mengisi harga tiket yang ingin diubah pada *form* harga tiket masuk yang disediakan. Setelah harga tiket diisi, *Admin* menekan tombol "ubah harga tiket" untuk menyimpan perubahan. Selanjutnya, sistem akan memproses dan menampilkan data tiket yang telah diperbarui. Setelah proses selesai, sistem akan kembali mengarahkan ke halaman data tiket.

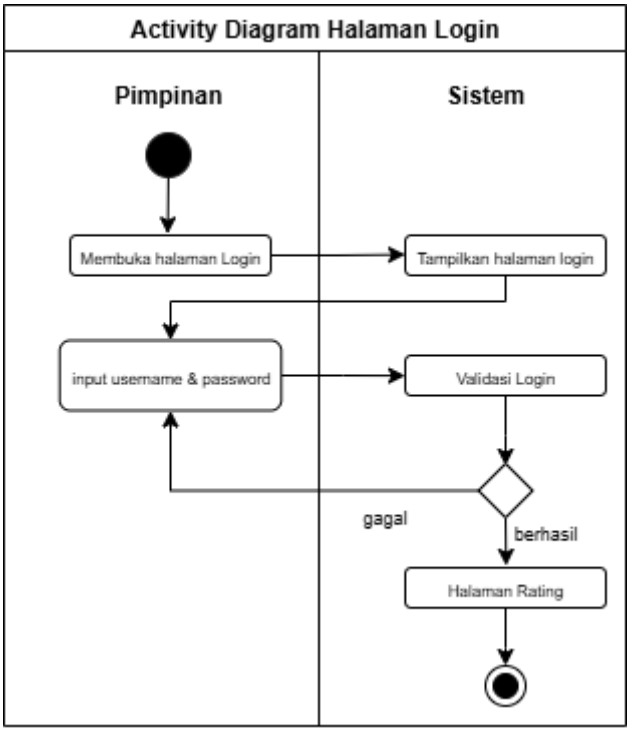
r. *Activity diagram* pembayaran



Gambar 4.22 *Activity diagram* pembayaran

Gambar 4.22 Proses dimulai ketika *admin* memilih untuk membuka halaman pembayaran. Setelah itu proses selanjutnya, sistem akan menampilkan halaman pembayaran yang berisi informasi terkait proses pembayaran yang sedang berlangsung. Selanjutnya, *admin* melakukan konfirmasi pembayaran dengan menekan tombol konfirmasi. proses ini dikirim ke sistem, yang kemudian memproses permintaan tersebut dan menampilkan status konfirmasi sebagai bentuk respon atas tindakan admin. Setelah status konfirmasi ditampilkan, sistem kembali mengarahkan atau memperbarui tampilan pada halaman pembayaran, menandai bahwa proses konfirmasi telah berhasil diselesaikan dan proses pembayaran selesai dilakukan.

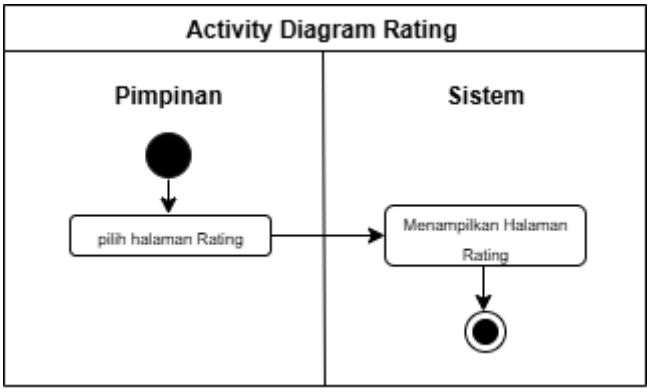
s. *Activity Diagram Login* Pimpinan



Gambar 4.23 *Actvity diagram login* pimpinan

Gambar 4.23 Proses dimulai ketika Pimpinan membuka halaman *login.* Setelah itu, sistem merespon dengan menampilkan halaman *login* tersebut. pimpinan kemudian mengisi *username* dan *password* di halaman yang telah ditampilkan, lalu mengirimkan data tersebut ke sistem untuk dilakukan proses validasi *login*. Sistem akan memeriksa apakah *username* dan *password* yang dimasukkan sesuai dengan data yang ada di database. Jika proses validasi gagal, pimpinan akan diarahkan kembali untuk menginput ulang data login yang benar. Namun, jika validasi berhasil, sistem akan mengarahkan pimpinan ke halaman rating sebagai tanda bahwa proses *login* telah berhasil.

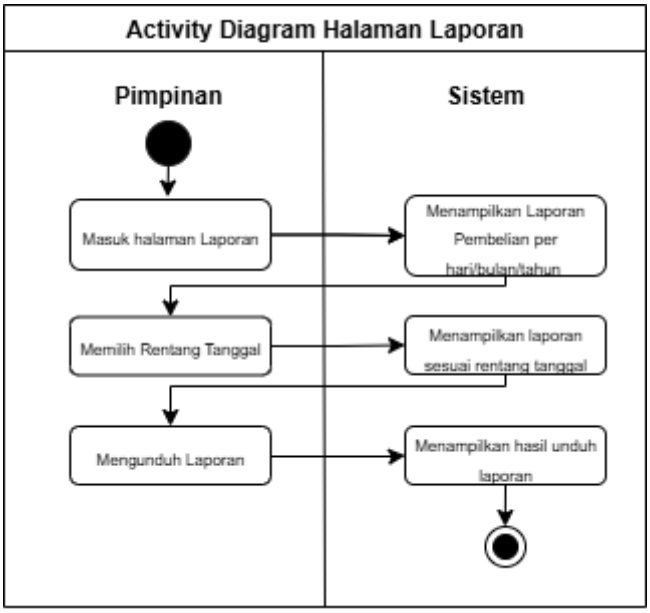
t. *Activity diagram rating*



Gambar 4.24 *Activity diagram rating*

Gambar 4.24 Proses dimulai ketika pimpinan telah berhasil *login* dan otomatis diarahkan ke halaman *rating*. Setelah itu sistem merespons dengan menampilkan halaman rating sesuai permintaan.

u. *Activity diagram* halaman laporan



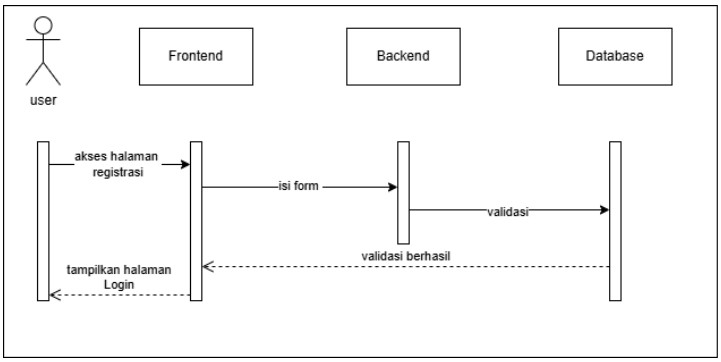
Gambar 4.25 *Activity diagram* halaman laporan

Gambar 4.25 Proses dimulai ketika pimpinan mengakses halaman laporan. Setelah berhasil masuk ke halaman tersebut, sistem secara otomatis merespons dengan menampilkan laporan pembelian yang tersedia dalam pilihan periode harian, bulanan, atau tahunan. Apabila pimpinan ingin melihat laporan yang lebih

spesifik, ia dapat menentukan rentang tanggal sesuai dengan kebutuhan. Rentang tanggal yang dipilih ini kemudian dikirimkan ke sistem. Selanjutnya, sistem akan memproses permintaan tersebut dan menampilkan laporan pembelian yang sudah difilter berdasarkan rentang tanggal yang telah dipilih oleh kasir. Setelah laporan ditampilkan, pimpinan dapat mengunduh laporan tersebut sesuai kebutuhan. Setelah itu sistem akan menampilkan hasil unduhan laporan tersebut.

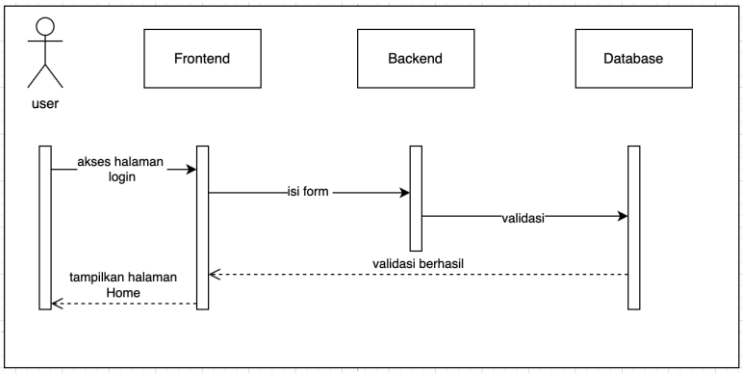
3. *Sequence Diagram*

a. *Sequence diagram* Halaman *Registrasi user*



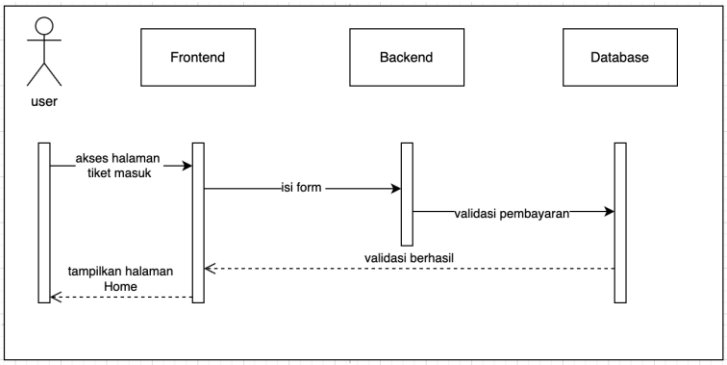
Gambar 4.26 *Sequence diagram registrasi user*

b. *Sequence diagram* Halaman *login user*



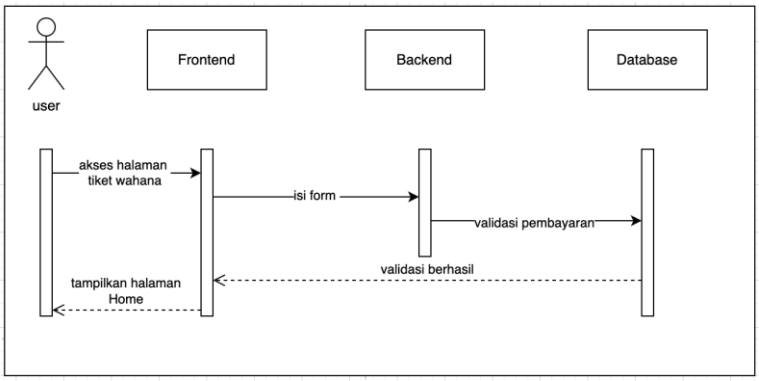
Gambar 4.27 *Sequence diagram login user*

c. *Sequence diagram* Halaman tiket masuk *user*



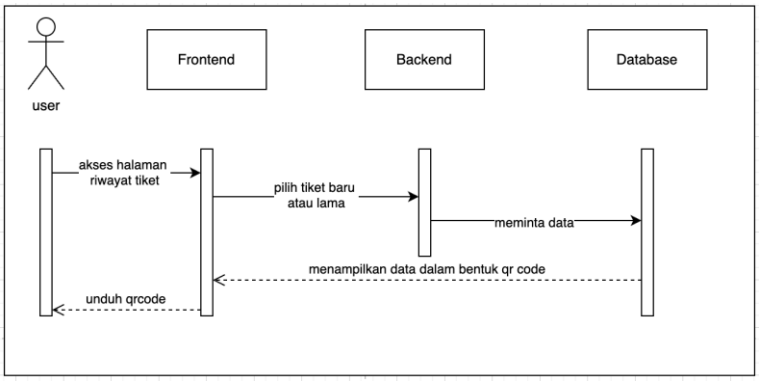
Gambar 4.28 *Sequence diagram* halaman tiket masuk *user*

d. *Sequence diagram* Halaman tiket wahana *user*



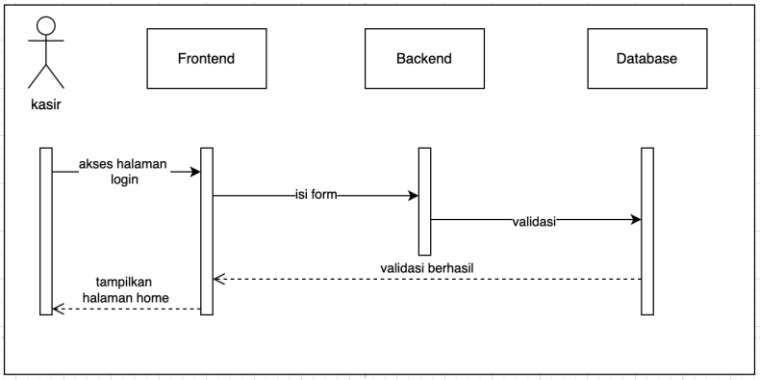
Gambar 4.29 *Sequence diagram* halaman tiket wahana *user*

e. *Sequence diagram* Halaman riwayat tiket



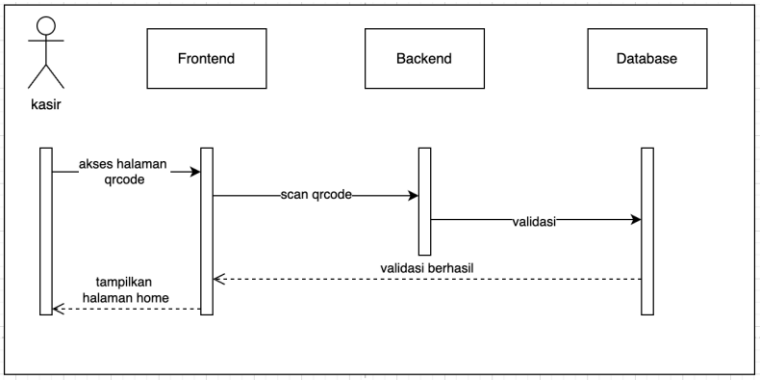
Gambar 4.30 *Sequence diagram* halaman riwayat tiket

f. *Sequence diagram* Halaman *login* kasir



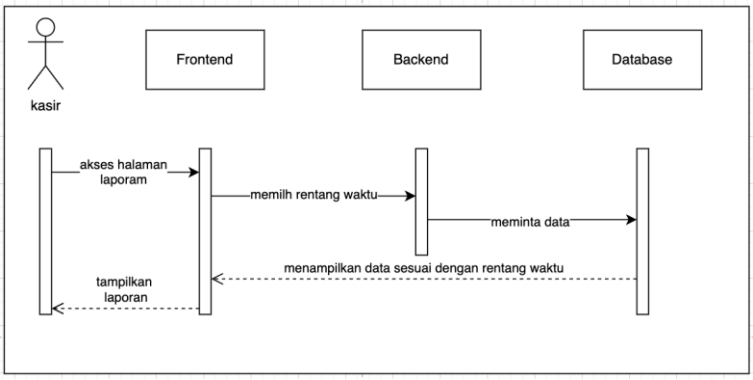
Gambar 4.31 *Sequence diagram* halaman *login* kasir

g. *Sequence diagram* Halaman *Scan* Kasir



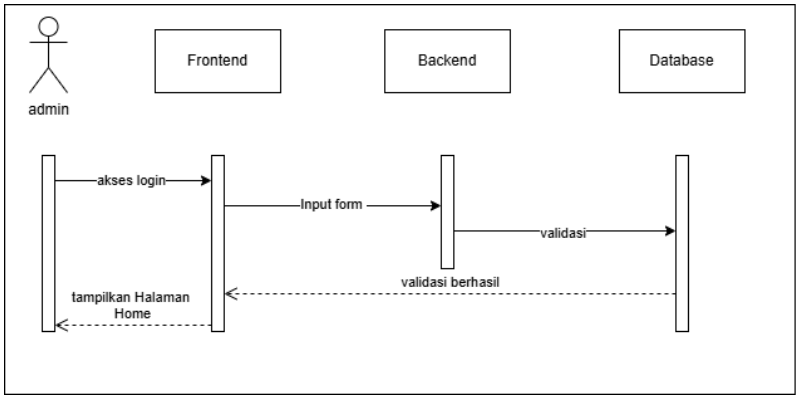
Gambar 4.32 *Sequence diagram* halaman *scan* kasir

h. *Sequence diagram* halaman laporan



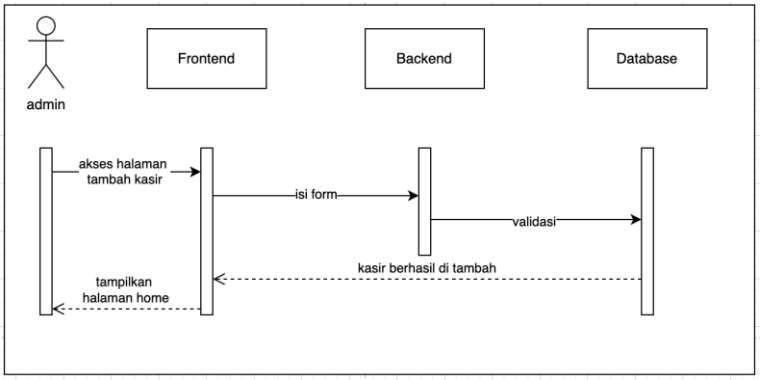
Gambar 4.33 *Sequence diagram* halaman laporan

i. *Sequence diagram* halaman *login admin*



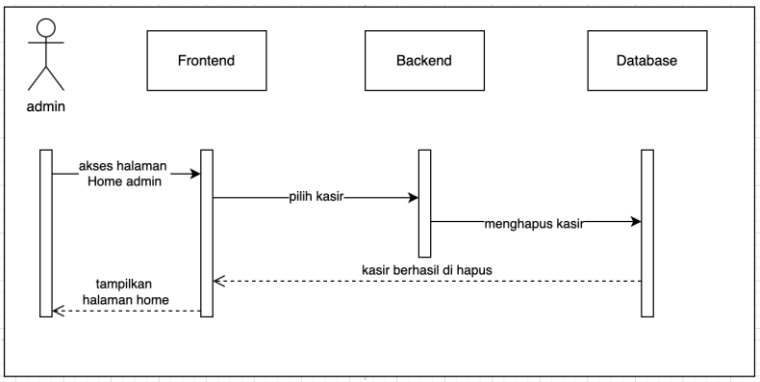
Gambar 4.34 *Sequence diagram* halaman *login admin*

j. *Sequence diagram* tambah kasir



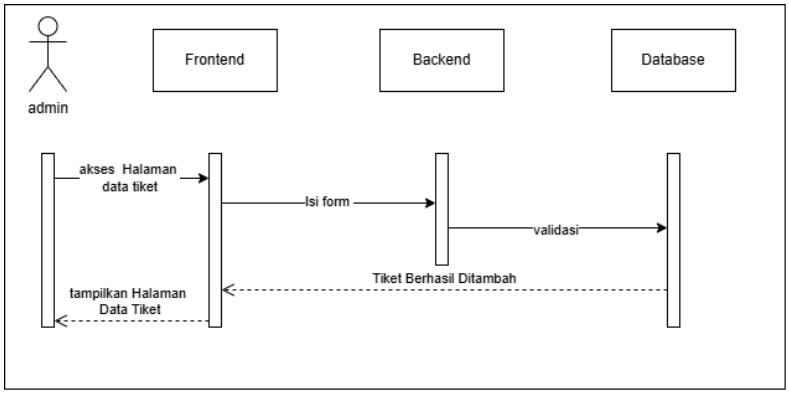
Gambar 4.35 *Sequence diagram* halaman tambah kasir

k. *Sequence diagram* hapus kasir



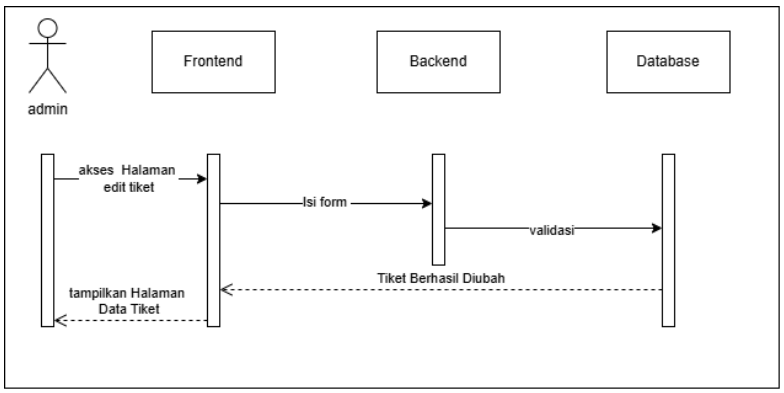
Gambar 4.36 *Sequence diagram* hapus kasir

l. *Sequence diagram* tambah tiket wahana



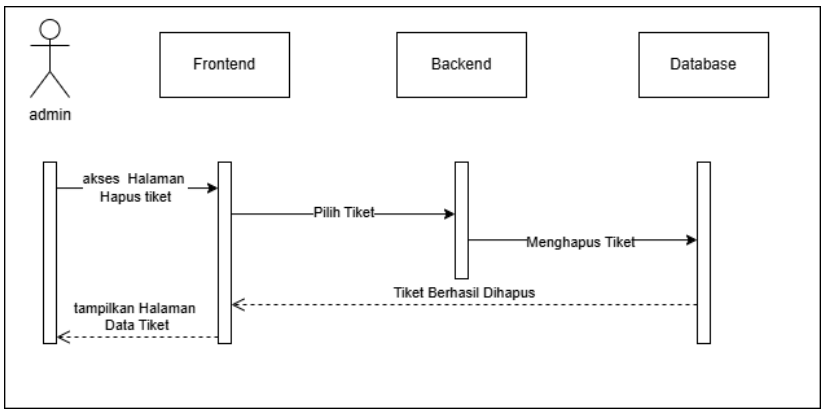
Gambar 4.37 *Sequence diagram* tambah tiket wahana

m. *Sequence diagram* edit tiket wahana



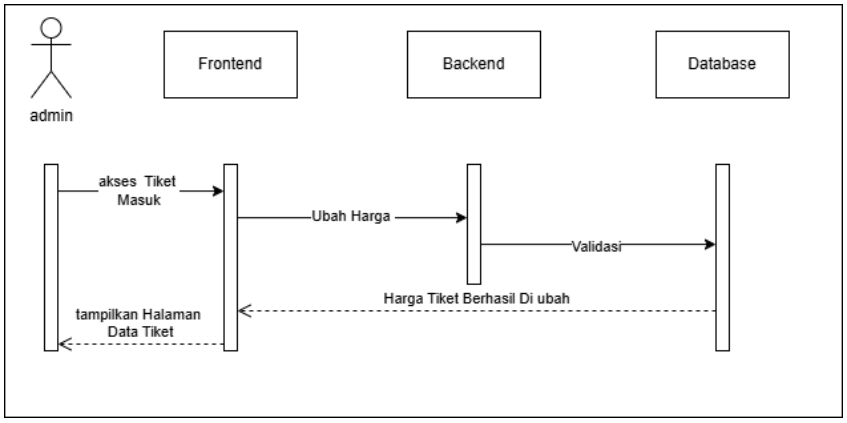
Gambar 4.38 *Sequence diagram* edit tiket wahana

n. *Sequence diagram* hapus tiket wahana



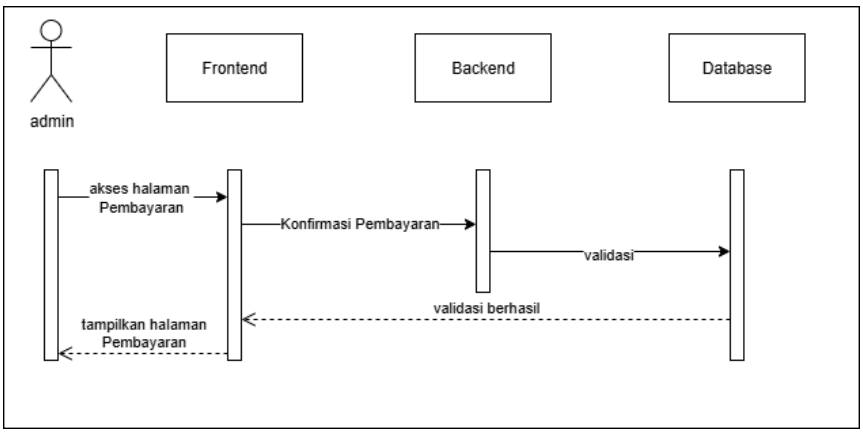
Gambar 4.39 *Sequence diagram* hapus tiket

o. *Sequence diagram* ubah harga tiket masuk



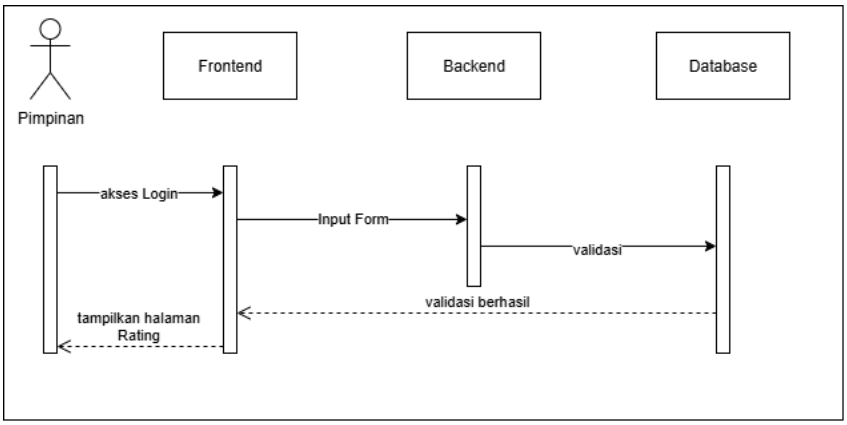
Gambar 4.40 *Sequence diagram* ubah harga tiket masuk

p. *Sequence diagram* Pembayaran



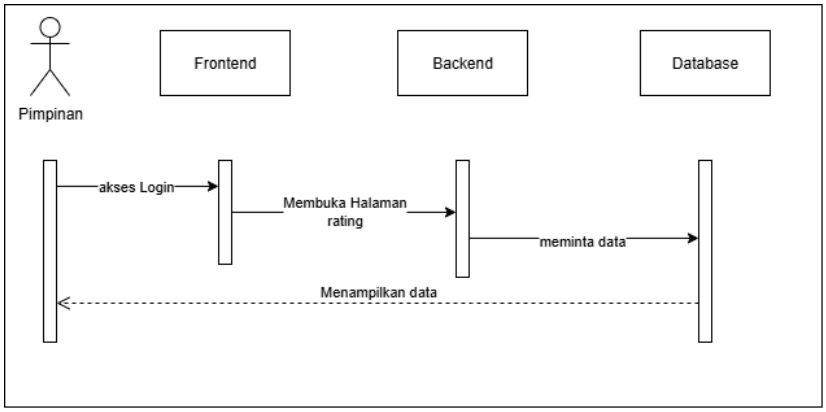
Gambar 4.41 *Sequence diagram* pembayaran

q. *Sequence diagram login* pimpinan



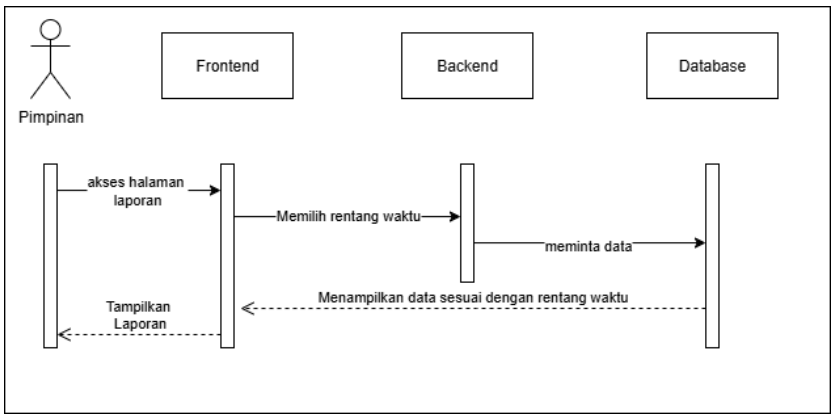
Gambar 4.42 *Sequence diagram login* pimpinan

r. *Sequence diagram rating*



Gambar 4.43 *Sequence diagram rating*

s. *Sequence diagram* laporan



Gambar 4.44 *Sequence diagram* laporan

B. Desain Sistem

1. User

a. Tampilan *splash*

Tampilan ini merupakan layar awal yang muncul ketika pengguna pertama kali membuka aplikasi, menampilkan logo sebagai identitas utama. Tujuan dari *splash screen* ini adalah memberikan jeda sejenak bagi pengguna untuk mengenali aplikasi sebelum diarahkan ke halaman menu utama**.**



Gambar 4.45 Tampilan *splash*

Berikut merupakan *sourcode* dari pemanggilan *splash*:

```
public class SplashActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        if (getSupportActionBar() != null) {
            getSupportActionBar().hide();
        }
        setContentView(R.layout.activity_splash);
        // Delay 5 detik sebelum pindah ke MainActivity
        new Handler().postDelayed(new Runnable() { // Membuat handler untuk menjalankan
delay
            @Override
            public void run() { // Metode run akan dijalankan setelah delay selesai
                // Pindah ke MainActivity
                Intent intent = new Intent(SplashActivity.this, MainActivity.class);
                // Membuat intent untuk berpindah dari SplashActivity ke MainActivity
                startActivity(intent); // Menjalankan MainActivity
                finish(); // Tutup SplashActivity agar tidak kembali saat tombol Back ditekan
            }
        }, 3000); // 3000 ms = 3 detik
    }
}
```

b. Tampilan Halaman *Login*

Halaman login merupakan hal yang penting untuk menjaga keamanan sistem dan memastikan hanya pengguna yang berwenang yang dapat mengakses nya serta dapat masuk dengan memasukkan data yang valid, seperti *username* dan *password.*



Gambar 4. 46 Halaman *Login*

Berikut merupakan *sourcode* dari halaman *login*:

```
//Data
    $username = $_POST['username'] ?? '';
    $password = $_POST['password'] ?? '';

    $response = []; //Data Response

    //Cek username didalam databse
    $usersQuery = $koneksi->prepare("SELECT * FROM users
where username = ?");
    $usersQuery->execute(array($username));
    $query = $usersQuery->fetch();

    if ($usersQuery->rowCount() == 0) {
        $response['status'] = false;
        $response['message'] = "Username Tidak
Terdaftar";
    } else {
        // Ambil password di db
```

```
    $passwordDB = $query['password'];

        if (strcmp($password, $passwordDB) === 0) {
            $response['status'] = true;
            $response['message'] = "Login Berhasil";
            $response['data'] = [
                'id' => $query['id'],
                'username' => $query['username'],
                'password' => $query['password'],
                'name' => $query['name'],
                'email' => $query['email'],
                'role' => $query['role']

            ];
        } else {
            $response['status'] = false;
            $response['message'] = "Password Anda Salah";
        }
```

c. Tampilan halaman registrasi

Halaman registrasi berfungsi untuk mendaftarkan pengguna baru ke dalam sistem. Pada halaman ini, pengguna dapat membuat akun dengan mengisi formulir yang tersedia, seperti nama lengkap, alamat email, *username* dan *password.*



Gambar 4. 47 Halaman Registrasi

Berikut merupakan *sourcode* dari halaman registrasi:

```
<?php
include '../config/koneksi.php';
if ($_POST) {
    //POST DATA
    $username = filter_input(INPUT_POST, 'username',
FILTER_SANITIZE_STRING);
    $password = filter_input(INPUT_POST, 'password',
FILTER_SANITIZE_STRING);
    $name = filter_input(INPUT_POST, 'name',
FILTER_SANITIZE_STRING);
    $email = filter_input(INPUT_POST, 'email',
FILTER_SANITIZE_STRING);
    $role = filter_input(INPUT_POST, 'role',
FILTER_SANITIZE_STRING);
    $response = [];
    //Cek email didalam databse
    $emailQuery = $koneksi->prepare("SELECT * FROM users where
email = ?");
    $emailQuery->execute(array($email));

    //Cek username didalam databse
    $userQuery = $koneksi->prepare("SELECT * FROM users where
username = ?");
    $userQuery->execute(array($username));

   // Cek email dan username apakah ada tau tidak
    if ($emailQuery->rowCount() != 0) {
        // Beri Response
        $response['status'] = false;
        $response['message'] = 'Email sudah digunakan';
    } else if ($userQuery->rowCount() != 0) {
        // Beri Response
        $response['status'] = false;
    $response['message'] = 'Username sudah digunakan';
    } else {
        // Jika tidak ada yang digunakan, maka insert data ke
database

        $insertAccount = 'INSERT INTO users (username,
password, name, email, role) values (:username, :password,
:name, :email, :role)';
        $statement = $koneksi->prepare($insertAccount);
try {
            //Eksekusi statement db
            $statement->execute([
                ':username' => $username,
                ':password' => $password,
                ':name' => $name,
                ':email' => $email,
                ':role' => $role
            ]);
```

```
                //Beri response
            $response['status'] = true;
            $response['message'] = 'Akun berhasil didaftar';
            $response['data'] = [
                'username' => $username,
                'password' => $password,
                'name' => $name,
                'email' => $email,
                'role' => $role
        ];
        } catch (Exception $e) {
            die($e->getMessage());
        }
    }
  //Jadikan data JSON
    $json = json_encode($response, JSON_PRETTY_PRINT);
    echo $json;
}
```

d. Halaman Home User

Halaman home berfungsi untuk menampilkan pilihan menu yang tersedia di home untuk user akan diperlihatkan pilihan tiket yang ingin dibeli apakah itu tiket masuk ataupun tiket wahana



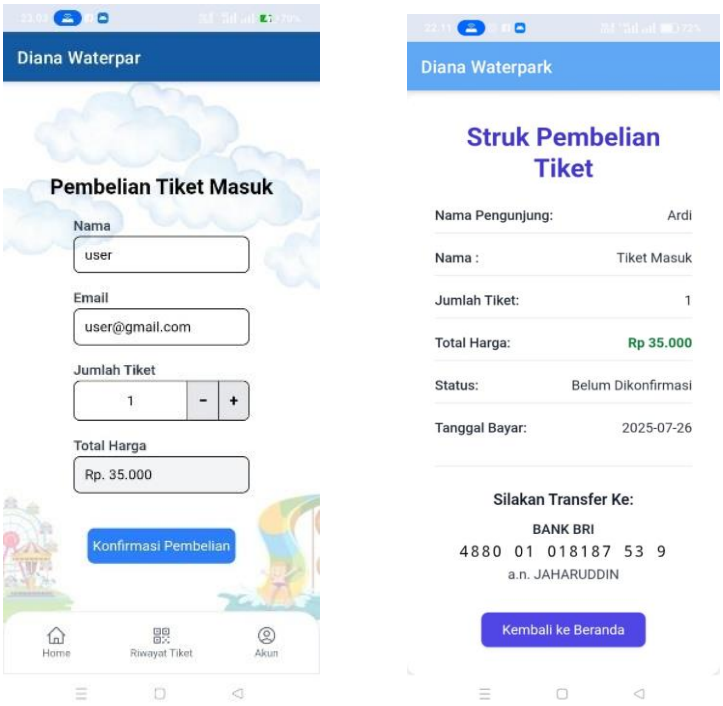
Gambar 4. 48 Halaman *Home*

Berikut merupakan *sourcode* dari halaman *home*:

```
<?php
$midtrans = isset($_COOKIE['id_tmasuk']) &&
isset($_COOKIE['name']) && isset($_COOKIE['email']) &&
isset($_COOKIE['jumlah_tmasuk']) &&
isset($_COOKIE['total_harga']) && isset($_COOKIE['order_id']);
?>
<div class="flex items-center justify-center pt-[230px]">
    <div class="flex flex-col space-y-7">
        <a href="tiketmasuk.php?id=<?= $_GET['id'] ?>"
class="px-[60px] py-4 font-bold bg-blue-400 text-white rounded-
full shadow-md hover:bg-blue-600 transition">
            TIKET MASUK
        </a>
        <a href="tiketwahana.php?id=<?= $_GET['id'] ?>"
class="px-[60px] py-4 font-bold bg-blue-400 text-white rounded-
full shadow-md hover:bg-blue-600 transition">
            TIKET WAHANA
        </a>
    </div>
</div>
```

e. Tiket masuk

Tiket masuk berfungsi jika seseorang masuk ke halaman *home* dan memilih tiket masuk maka user akan di perintahkan untuk mengisi jumlah tiket yang dibutuhkan atau ingin dibeli selanjutnya akan diarahkan untuk menekan button konfirmasi pembelian dan akan muncul struk pembayaran yang tujuan nya agar pengguna dapat melakukan transaksi pebayaran sesuai dengan yang tertera.



Gambar 4. 49 Tiket Masuk

Berikut merupakan *source*code dari pembelian tiket masuk:

```
<div class="flex items-center justify-center pt-[80px]">
    <div class="w-96 p-6 rounded-xl bg-transparent">
        <h2 class="text-2xl font-bold text-center mb-
4">Pembelian Tiket Masuk</h2>
        <form action="proses_tmasuk.php" class="bg-transparent
px-14" method="POST">
            <input type="hidden" name="action" value="CREATE">
            <input type="hidden" name="id_user" value="<?=
$_GET['id'] ?>">
            <!-- Nama -->
            <div class="mb-4">
                <label class="block text-gray-700 font-
semibold">Nama</label>
                <input type="text" name="nama" value="<?php
echo htmlspecialchars($row['name']); ?>" class="w-full px-3 py-
2 border rounded-lg focus:outline-none focus:ring focus:border-
blue-300" readonly>
            </div>
            <div class="mb-4">
                <label class="block text-gray-700 font-
semibold">Email</label>
                <input type="email" name="email" value="<?php
echo htmlspecialchars($row['email']); ?>" class="w-full px-3
py-2 border rounded-lg focus:outline-none focus:ring
focus:border-blue-300" readonly>
            </div>
```

```
            <div class="mb-4">
                <label class="block text-gray-700 font-
semibold">Jumlah Tiket</label>
                <div class="flex items-center border rounded-lg
overflow-hidden">
                    <input type="number" name="jumlah_tmasuk"
id="jumlah_tmasuk" value="0" min="0" required
                        class="w-full text-center px-3 py-2
focus:outline-none focus:ring-none focus:border-blue-300"
                        oninput="hitungTotal()">
                    <button type="button" onclick="ubahJumlah(-
1)" class="bg-gray-200 px-3 py-2 text-lg font-bold hover:bg-
gray-300 border-x">-</button>

                    <button type="button"
onclick="ubahJumlah(1)" class="bg-gray-200 px-3 py-2 text-lg
font-bold hover:bg-gray-300">+</button>
                </div>
      </div>
<!-- Harga Tiket (Otomatis) -->
            <div class="mb-4">
                <label class="block text-gray-700 font-
semibold">Total Harga</label>
                <input type="text" id="total_harga"
name="total_harga" readonly
                    class="w-full px-3 py-2 border rounded-lg
bg-gray-100">
            </div>
            <div class="px-4 mt-10">
                <!-- Tombol Submit -->
                <button type="submit" class="w-full bg-blue-500
text-white py-2 rounded-lg hover:bg-blue-600 transition">
          Konfirmasi Pembelian
                </button>
            </div>

        </form>
    </div>
</div>
           <!DOCTYPE html>
<html lang="id">
<head>
    <meta charset="UTF-8" />
    <title>Struk Pembelian</title>
    <script src="https://cdn.tailwindcss.com"></script>
</head>
<body class="bg-gray-50">

<div class="max-w-lg mx-auto mt-12 p-8 bg-white rounded-xl
shadow-md">
    <h2 class="text-3xl font-bold mb-6 text-center text-indigo-
700">
```

```
        <?= $is_wahana ? 'Struk Pembelian Tiket' : 'Struk
Pembelian Tiket Masuk' ?>
    </h2>
    <div class="space-y-4 text-gray-800">
        <div class="flex justify-between border-b pb-2">
            <span class="font-medium">Nama Pengunjung:</span>
            <span><?= htmlspecialchars($is_wahana ?
$row['nama_pengunjung'] : $row['nama']) ?></span>
        </div>
        <?php if (!$is_wahana): ?>
        <div class="flex justify-between border-b pb-2">
            <span class="font-medium">Email:</span>
            <span><?= htmlspecialchars($row['email']) ?></span>
        </div>
        <?php endif; ?>
        <?php if ($is_wahana): ?>
        <div class="flex justify-between border-b pb-2">
            <span class="font-medium">Nama :</span>
            <span class="text-right"><?=
nl2br(htmlspecialchars($row['nama_wahana'])) ?></span>
        </div>
        <div class="flex justify-between border-b pb-2">
            <span class="font-medium">Jumlah Tiket:</span>
            <span><?= htmlspecialchars($row['jumlah_twahana'])
?></span>
        </div>
        <?php else: ?>
        <div class="flex justify-between border-b pb-2">
            <span class="font-medium">Jumlah Tiket
Masuk:</span>
            <span><?= htmlspecialchars($row['jumlah_tmasuk'])
?></span>
        </div>
        <?php endif; ?>
<div class="flex justify-between border-b pb-2">
            <span class="font-medium">Total Harga:</span>
            <span class="text-green-700 font-bold">Rp <?=
number_format($row['total_harga'], 0, ',', '.') ?></span>
        </div>
<div class="flex justify-between border-b pb-2">
            <span class="font-medium">Status:</span>
            <span><?= htmlspecialchars($row['pembayaran'] ??
$row['status']) ?></span>
        </div>
        <?php if ($is_wahana): ?>
        <div class="flex justify-between">
            <span class="font-medium">Tanggal Bayar:</span>
            <span><?= htmlspecialchars($row['tanggal_bayar'])
```

```
?></span>
        </div>
        <?php endif; ?>

</div>

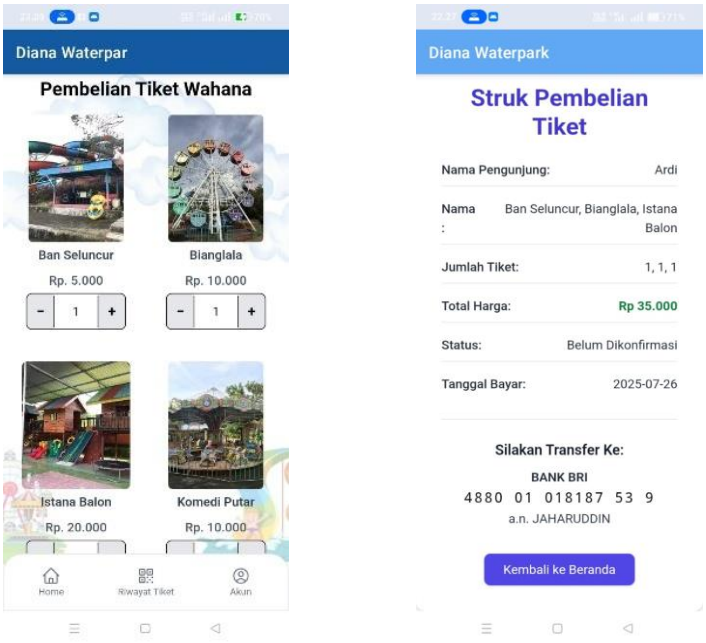
    <div class="mt-8 border-t pt-6 text-center">
        <h3 class="text-lg font-semibold text-gray-900 mb-
2">Silakan Transfer Ke:</h3>
        <p class="font-semibold text-gray-800">BANK BRI</p>
        <p class="font-mono text-lg tracking-widest">4880 01
018187 53 9</p>
        <p class="text-gray-700">a.n. JAHARUDDIN</p>
    </div>

     <div class="mt-8 flex justify-center">
        <a href="index.php?id=<?= urlencode($id_user) ?>"
           class="px-6 py-2 bg-indigo-600 text-white rounded-lg
shadow hover:bg-indigo-700 transition">
            Kembali ke Beranda
        </a>
    </div>
</div>

</body>
</html>
```

f. Tiket Wahana

Tiket wahana berfungsi jika seseorang masuk ke halaman *home* dan memilih tiket wahana maka *user* akan ditampilkan beberapa pilihan wahana serta di perintahkan untuk mengisi jumlah tiket yang dibutuhkan atau ingin dibeli selanjutnya akan diarahkan untuk menekan *button* konfirmasi pembelian dan akan muncul pilihan menu pembayaran



Gambar 4. 50 Tiket Wahana

Berikut merupakan *sourcode* dari tampilan tiket wahana:

```
<div class="flex items-center justify-center pt-[20px] mb-30">
    <div class="w-96 p-6 rounded-xl bg-transparent">
        <h2 class="text-2xl font-bold text-center mb-
4">Pembelian Tiket Wahana</h2>
        <form action="proses_twahana.php" class="bg-
transparent" method="POST" onsubmit="return gabungkanData()">
            <input type="hidden" name="action" value="CREATE">
            <input type="hidden" name="id_user" value="<?=
$_GET['id'] ?>">

            <input type="hidden" value="" id="nama_wahana"
name="nama_wahana">
            <input type="hidden" value="" id="jumlah_twahana"
name="jumlah_twahana">

            <div class="flex flex-col items-center">

                <div class="grid grid-cols-2 gap-10">
                    <?php foreach ($wahana as $item): ?>
                        <div class="flex flex-col items-center
rounded-lg">
                            <img loading="lazy" class="h-40 w-
auto object-cover rounded-md" src="./img/<?=
htmlspecialchars($item['gambar']) ?>" alt="<?=
htmlspecialchars($item['nama']) ?>">
```

```
<h2 class="text-md font-semibold text-gray-700 mt-1"><?=
htmlspecialchars($item['nama']) ?></h2>

                            <label class="block text-gray-600
font-medium mt-2">Rp. <?= number_format($item['harga'], 0, ',',
'.') ?></label>
                            <div class="flex items-center
border rounded-lg overflow-hidden mt-1">
                                <button type="button"
onclick="ubahJumlah('jumlah_<?= $item['id_wahana'] ?>', -1)"
                                    class="bg-gray-200 px-3 py-
2 text-lg font-bold hover:bg-gray-300">-</button>
<input type="number" id="jumlah_<?= $item['id_wahana'] ?>"
value="0" min="0" required
                                    class="w-14 text-center px-
3 py-2 focus:outline-none focus:ring-2 focus:ring-blue-300
border-x">

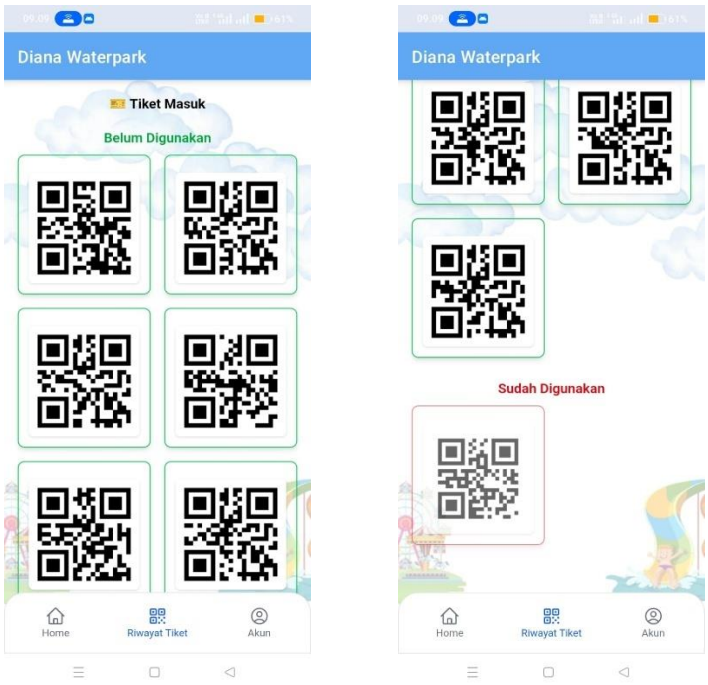
                                <button type="button"
onclick="ubahJumlah('jumlah_<?= $item['id_wahana'] ?>', 1)"
                                    class="bg-gray-200 px-3 py-
2 text-lg font-bold hover:bg-gray-300">+</button>
                            </div>
                        </div>
                    <?php endforeach; ?>
                </div>
            </div>
            <div class="px-14 pt-8">
                <!-- Nama -->
                <div class="mb-4">
                    <label class="block text-gray-700 font-
semibold">Nama</label>
                    <input type="text" name="nama" value="<?php
echo htmlspecialchars($row['name']); ?>" class="w-full px-3 py-
2 border rounded-lg focus:outline-none focus:ring focus:border-
blue-300" readonly>
                </div>

                <div class="mb-4">
                    <label class="block text-gray-700 font-
semibold">Email</label>
                    <input type="email" name="email"
```

g. Menu Riwayat Tiket

Menu riwayat tiket berfungsi untuk menampilkan *Qr Code* atau sebagai bukti bahwa *user* telah berhasil membeli suatu tiket apakah itu tiket masuk maupun tiket wahana dan diriwayat tiket juga memperlihatkan tiket yang belum

di gunakan maupun tiket yang sudah digunakan user serta memiliki fitur download *Qr Code* untuk mempermudah user ketika ingin mencetak tiket tersebut secara fisik



Gambar 4. 51 Tampilan Riwayat Tiket

Berikut merupakan *sourcode* dari tampilan riwayat tiket:

```
<div class="mb-[100px]">

    <!-- Tiket Masuk -->
    <div class="flex justify-center items-center">
        <div class="overflow-x-auto max-w-full p-4">
            <h2 class="text-center font-bold mb-4">🎟️ Tiket
Masuk</h2>

            <!-- Belum Digunakan -->
            <h3 class="text-center font-semibold mb-2 text-
green-600">Belum Digunakan</h3>
            <?php
            $belumMasuk = array_filter($dataTiketMasuk,
fn($row) => $row['validasi'] !== 'Sudah Digunakan');
            ?>
            <?php if (count($belumMasuk) > 0): ?>
                <div class="grid grid-cols-2 md:grid-cols-4
gap-4 mb-6">
                    <?php foreach ($belumMasuk as $row): ?>
                        <div class="border border-green-500 p-3
rounded-lg shadow-md text-center">
```

```
                            <!-- Tampilkan Nama Wahana -->
                            <h4 class="font-semibold mb-2"><?=
htmlspecialchars($row['nama_wahana']) ?></h4>
                            <img src="<?=
htmlspecialchars($row['qrcode']) ?>"
                                alt="QR Code"
                                class="mx-auto max-w-[150px]
w-full h-auto rounded-md shadow-sm cursor-pointer"

onclick="showImageModal(this.src)">
                        </div>
                    <?php endforeach; ?>
                </div>
            <?php else: ?>
                <p class="text-center text-gray-500 mb-6">Tidak
ada tiket masuk yang belum digunakan.</p>
            <?php endif; ?>
<!-- Sudah Digunakan -->
            <h3 class="text-center font-semibold mb-2 text-red-
600">Sudah Digunakan</h3>
            <?php
            $sudahMasuk = array_filter($dataTiketMasuk,
fn($row) => $row['validasi'] === 'Sudah Digunakan');
            ?>
            <?php if (count($sudahMasuk) > 0): ?>
                <div class="grid grid-cols-2 md:grid-cols-4
gap-4">
                    <?php foreach ($sudahMasuk as $row): ?>
                        <div class="border border-red-500 p-3
rounded-lg shadow-md text-center opacity-60">
                            <h4 class="font-semibold mb-2"><?=
htmlspecialchars($row['nama_wahana']) ?></h4>
                            <img src="<?=
htmlspecialchars($row['qrcode']) ?>"
                                alt="QR Code"
                                class="mx-auto max-w-[150px]
w-full h-auto rounded-md shadow-sm cursor-pointer"
onclick="showImageModal(this.src)">
                        </div>
                    <?php endforeach; ?>
                </div>
            <?php else: ?>
                <p class="text-center text-gray-500">Tidak ada
tiket masuk yang sudah digunakan.</p>
            <?php endif; ?>
        </div>
    </div>
<!-- Tiket Wahana -->
    <div class="flex justify-center items-center mt-8">
        <div class="overflow-x-auto max-w-full p-4">
            <h2 class="text-center font-bold mb-4">🎟 Tiket
Wahana</h2>
            <!-- Belum Digunakan -->
            <h3 class="text-center font-semibold mb-2 text-
green-600">Belum Digunakan</h3>
            <?php
            $belumWahana = array_filter($dataTiketWahana,
fn($row) => $row['validasi'] !== 'Sudah Digunakan');
            ?>
```

```
            <?php if (count($belumWahana) > 0): ?>
                <div class="grid grid-cols-2 md:grid-cols-4
gap-4 mb-6">
                    <?php foreach ($belumWahana as $row): ?>
                        <div class="border border-green-500 p-3
rounded-lg shadow-md text-center">
                            <!-- Tampilkan Nama Wahana -->
                            <h4 class="font-semibold mb-2"><?=
htmlspecialchars($row['nama_wahana']) ?></h4>
                            <!-- QR Code -->
                            <img src="<?=
htmlspecialchars($row['qrcode']) ?>"
                                 alt="QR Code"
                                 class="mx-auto max-w-[150px]
w-full h-auto rounded-md shadow-sm cursor-pointer"

onclick="showImageModal(this.src)">
                        </div>
                    <?php endforeach; ?>
                </div>
            <?php else: ?>
                <p class="text-center text-gray-500 mb-6">Tidak
ada tiket wahana yang belum digunakan.</p>
            <?php endif; ?>

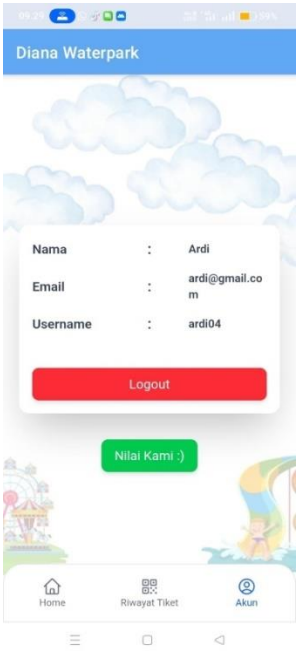
            <!-- Sudah Digunakan -->
            <h3 class="text-center font-semibold mb-2 text-red-
600">Sudah Digunakan</h3>
            <?php
            $sudahWahana = array_filter($dataTiketWahana,
fn($row) => $row['validasi'] === 'Sudah Digunakan');
            ?>
            <?php if (count($sudahWahana) > 0): ?>
                <div class="grid grid-cols-2 md:grid-cols-4
gap-4">
                    <?php foreach ($sudahWahana as $row): ?>
                        <div class="border border-red-500 p-3
rounded-lg shadow-md text-center opacity-60">
                            <!-- Tampilkan Nama Wahana -->
                            <h4 class="font-semibold mb-2"><?=
htmlspecialchars($row['nama_wahana']) ?></h4>
                            <!-- QR Code -->
                            <img src="<?=
htmlspecialchars($row['qrcode']) ?>"
                                 alt="QR Code"
                                 class="mx-auto max-w-[150px]
w-full h-auto rounded-md shadow-sm cursor-pointer"

onclick="showImageModal(this.src)">
                        </div>
                    <?php endforeach; ?>
                </div>
            <?php else: ?>
        <p class="text-center text-gray-500">Tidak ada tiket
wahana yang sudah digunakan.</p>
            <?php endif; ?>
        </div>
    </div>
</div>
```

```
            <!-- Modal Fullscreen -->
<div id="imageModal"
     class="fixed inset-0 bg-black bg-opacity-80 flex flex-col
items-center justify-center z-50 hidden">
    <img id="modalImage" src="" alt="QR Fullscreen"
         class="w-[200px] h-[200px] object-contain rounded-lg
shadow-lg mb-4">
    <a id="downloadButton" href="#" download="qr-code.png"
       class="bg-white text-black px-4 py-2 rounded-lg shadow
hover:bg-gray-200 transition">
        Download
    </a>
</div>
```

h. Menu *Logout & rating*

Menu ini berfungsi untuk mengakhiri sesi pengguna pada aplikasi. *Logout* digunakan untuk menjaga keamanan dan privasi akun, serta memungkinkan pengguna untuk keluar dari akun saat ini dan masuk menggunakan akun lain jika diperlukan serta dapat memberikan penilaian atau *rating* dengan menekan *button* berikami nilai .



gambar 4. 52 *Logout & rating*

Berikut merupakan *sourcode* dari akun *Logout*:

```
<div class="min-h-screen flex flex-col items-center justify-
center">

    <div class="w-80 bg-white rounded-xl shadow-2xl p-4">
        <!--  Profil -->
        <div class="flex flex-col space-y-3 font-semibold">
            <div class="grid grid-cols-3 items-center">
                <h2 class="text-md text-gray-700">Nama</h2>
                <h2 class="text-lg text-gray-700 text-
center">:</h2>
                <h2 class="text-sm text-gray-700 break-
words"><?php echo htmlspecialchars($row['name']); ?> </h2>
            </div>
            <div class="grid grid-cols-3 items-center">
                <h2 class="text-gray-700 text-md">Email</h2>
                <h2 class="text-gray-700 text-lg text-
center">:</h2>
                <h2 class="text-gray-700 text-sm break-
words"><?php echo htmlspecialchars($row['email']); ?></h2>
            </div>
            <div class="grid grid-cols-3 items-center">
                <h2 class="text-gray-700 text-md">Username</h2>
                <h2 class="text-gray-700 text-lg text-
center">:</h2>
                <h2 class="text-gray-700 text-sm break-
words"><?php echo htmlspecialchars($row['username']); ?></h2>
            </div>
        </div>

        <!-- Tombol Aksi -->
        <div class="mt-10 space-y-3">
            <button onclick="logout()" class="w-full px-4 py-2
bg-red-500 text-white rounded-lg shadow-md hover:bg-red-600
transition">
                Logout
            </button>
        </div>
    </div>
    <?php if ($role === 'user') { ?>
        <div class="mt-8">
            <button onclick="toggleModal()" class="w-full px-4
py-2 bg-green-500 text-white rounded-lg shadow-md hover:bg-
green-600 transition">
                Nilai Kami :)
            </button>
        </div>
    <?php } ?>
</div>
```

2. Kasir

a. Menu *home*

Menu *Home* Kasir berfungsi untuk menampilkan total jumlah tiket masuk dan tiket wahana yang telah terjual secara keseluruhan. Informasi ini disajikan secara ringkas dan real-time agar memudahkan kasir dalam memantau jumlah penjualan tanpa perlu melakukan perhitungan manual



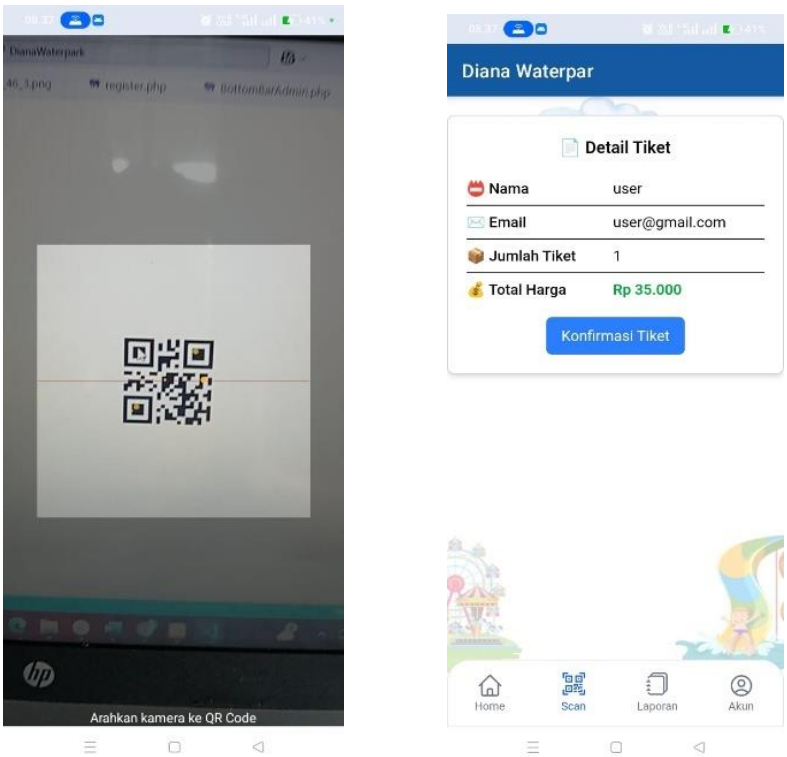
Gambar 4. 53 *Home*

Berikut merupakan *sourcode* dari *Home* Kasir:

```
<div class="flex items-center justify-center pt-[230px]">
    <div class="flex flex-col space-y-7">
        <div class="text-center px-4 py-4 font-bold bg-blue-400
text-white rounded-full shadow-md transition">
            JUMLAH TIKET MASUK TERJUAL : <?php echo
$dataMasuk['total_masuk'] ?? 0; ?>
        </div>
        <div class=" text-center px-4 py-4 font-bold bg-blue-
400 text-white rounded-full shadow-md transition">
            JUMLAH TIKET WAHANA TERJUAL : <?php echo
$totalWahana ?>
        </div>
    </div>
</div>
```

b. Fitur Scan

Menu ini berfungsi untuk memindai qr code dari user agar kasir dapat melihat detail dari tiket yang dipesan selanjutnya kasir memvalidasi dengan mengklik tombol button “konfirmasi tiket” sebagai bukti bahwa tiket tersebut sudah digunakan.



Gambar 4. 54 Fitur *Scan*

Berikut merupakan fitur *scan* kamera :

```
<uses-permission android:name="android.permission.CAMERA" />
```

Berikut merupakan sourcode dari tampilan detail tiket:

```
<!-- Container hasil scan tiket masuk-->
<div id="scan-result" class="mt-5 p-5 bg-white border border-
gray-300 rounded-lg shadow-md max-w-md mx-auto"
    style="display: <?= $ticketData ? 'block' : 'none' ?>;">

    <h3 class="text-lg font-semibold text-center mb-3">📄
Detail Tiket</h3>
    <table class="w-full text-left border-collapse">
        <tr class="border-b">
            <td class="font-medium py-2">👄 Nama</td>
            <td><?= htmlspecialchars($ticketData['nama'] ?? '')
?></td>
                                    </tr>
        <tr class="border-b">
            <td class="font-medium py-2">✉ Email</td>
            <td><?= htmlspecialchars($ticketWahanaData['email']
?? '') ?></td>
        </tr>
        <tr class="border-b align-top">
            <td class="font-medium py-2">👍 Wahana</td>
            <td>
                <ul class="space-y-1">
                    <?php foreach ($wahanaList as $item): ?>
                        <li class="flex justify-between">
                            <span class="text-left"><?=
htmlspecialchars($item['nama']) ?></span>
                            <span class="text-right font-
semibold text-gray-700"><?= htmlspecialchars($item['jumlah'])
?></span>
                        </li>
                    <?php endforeach; ?>
                </ul>
            </td>
        </tr>
        <tr>
            <td class="font-medium py-2">💰 Total Harga</td>
            <td class="text-green-600 font-semibold">
                Rp <?=
number_format($ticketWahanaData['total_harga'] ?? 0, 0, ',',
'.') ?>
            </td>
        </tr>
    </table>
<tr class="border-b">
            <td class="font-medium py-2">✉ Email</td>
            <td><?= htmlspecialchars($ticketData['email'] ??
'') ?></td>
        </tr>
        <tr class="border-b">
            <td class="font-medium py-2">📦 Jumlah Tiket</td>
            <td><?=
```

```
htmlspecialchars($ticketData['jumlah_tmasuk'] ?? '') ?></td>
        </tr>
        <tr>
            <td class="font-medium py-2">💰 Total Harga</td>
            <td class="text-green-600 font-semibold">
                Rp <?= number_format($ticketData['total_harga']
?? 0, 0, ',', '.') ?>
            </td>
        </tr>
    </table>

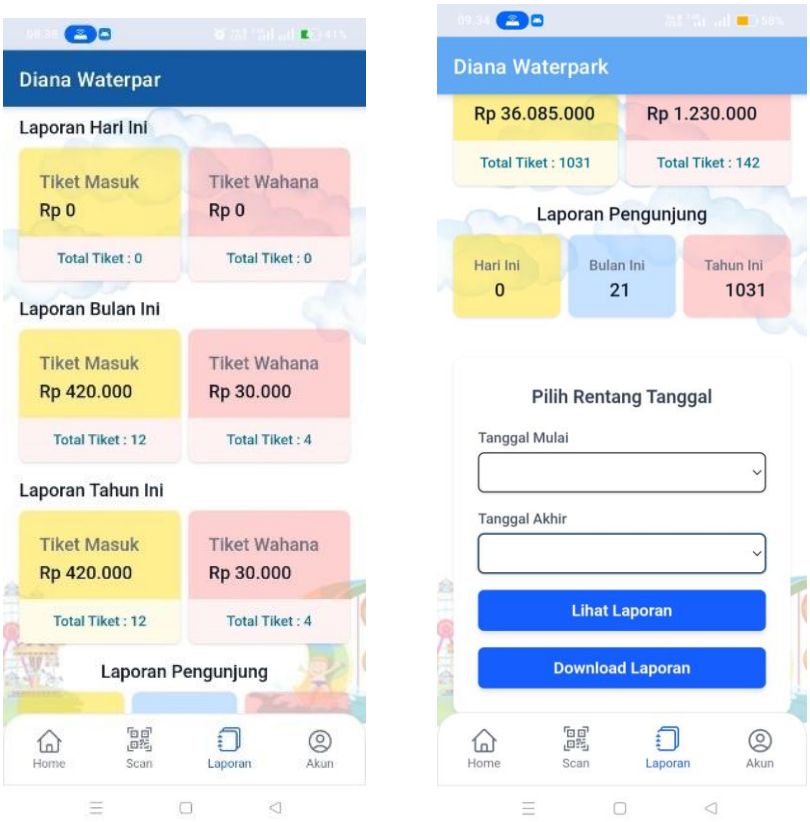
    <?php if ($ticketData && $ticketData['validasi'] !== "Sudah
Digunakan") : ?>
        <form action="proses_validasi.php" method="POST">
            <input type="hidden" name="action" value="UPDATE">
            <input type="hidden" name="id_tmasuk" value="<?=
htmlspecialchars($ticketData['id_tmasuk']) ?>">
            <input type="hidden" name="id_user" value="<?=
$_GET['id'] ?>">
            <input type="hidden" value="Sudah Digunakan"
name="validasi">

            <div class="text-center">
                <button type="submit" class="bg-blue-500 text-
white px-4 py-2 rounded-md mt-3">
                    Konfirmasi Tiket
        </button>
            </div>
        </form>
    <?php else: ?>
        <p class="text-red-600 text-center font-semibold mt-
3">❌ Tiket masuk sudah digunakan!</p>
    <?php endif; ?>
</div>

<!-- Container hasil scan tiket wahana-->
<div id="scan-result-wahana" class="mt-5 p-5 bg-white border
border-gray-300 rounded-lg shadow-md max-w-md mx-auto"
     style="display: <?= $ticketWahanaData ? 'block' : 'none'
                            ?>;">
<h3 class="text-lg font-semibold text-center mb-3">📄 Detail
Tiket</h3>
    <table class="w-full text-left border-collapse">
        <tr class="border-b">
            <td class="font-medium py-2">👄 Nama</td>
            <td><?= htmlspecialchars($ticketWahanaData['nama']
?? '') ?></td>
                                </tr>
```

c. Menu Laporan

Menu ini berfungsi untuk menampilkan laporan dari penjualan tiket yang dapat dilihat perhari,perbulan dan juga pertahun dan juga terdapat menu pilihan laporan tertentu dengan memasukkan rentang tanggal yang di inginkan setelah itu laporan akan muncul sesuai dengan rentang tanggal yang di isi serta dapat di download dan tersimpan dipenyimpanan *handphone.*



Gambar 4. 55 Laporan

Berikut merupakan *sourcode* dari tampilan laporan:

```
 <!-- // untuk memamnggil tanggal atau rentang tanggal nya -->
    <?php
    $totalJumlahMasuk = 0;
    $totalHargaMasuk = 0;
    $totalJumlahWahana = 0;
    $totalHargaWahana = 0;

    // Lihat Laporan
    if ($_SERVER["REQUEST_METHOD"] === "POST" &&
isset($_POST['start_date'], $_POST['end_date'],
$_POST['lihat_laporan'])) {
        $start = $_POST['start_date'];
        $end = $_POST['end_date'];
```

```
         // Tiket Masuk
        $stmt = $konek->prepare("SELECT SUM(jumlah_tmasuk) AS
total_jumlah, SUM(total_harga) AS total_harga FROM tiketmasuk
WHERE DATE(created_at) BETWEEN ? AND ?");
        $stmt->bind_param("ss", $start, $end);
        $stmt->execute();
        $result = $stmt->get_result();
                    $row = $result->fetch_assoc();

$totalJumlahMasuk = $row['total_jumlah'] ?? 0;
        $totalHargaMasuk = $row['total_harga'] ?? 0;

        // Tiket Wahana
        $stmt = $konek->prepare("SELECT jumlah_twahana,
total_harga FROM tiketwahana WHERE DATE(created_at) BETWEEN ?
AND ?");
        $stmt->bind_param("ss", $start, $end);
        $stmt->execute();
        $result = $stmt->get_result();

        while ($row = $result->fetch_assoc()) {
            // Memecah jumlah_twahana yang dipisahkan koma
            $jumlahArrayWahana = explode(',',
$row['jumlah_twahana']);

            // Menghitung total jumlah tiket wahana
            foreach ($jumlahArrayWahana as $jumlahWahana) {
                if (is_numeric($jumlahWahana)) {
                    $totalJumlahWahana += (int) $jumlahWahana;
                }
            }

            // Menambahkan total harga tiket wahana
            $totalHargaWahana += $row['total_harga'] ?? 0;
        }
    }
    // Download Laporan
    if (
        $_SERVER["REQUEST_METHOD"] === "POST"
        && isset($_POST['start_date'], $_POST['end_date'],
$_POST['download_laporan'])
    ) {
        $_SESSION['laporan_didownload'] = true; // Set session
        $start = $_POST['start_date'];
        $end = $_POST['end_date'];
        $sqltiket = "SELECT * FROM tiketmasuk
                 WHERE pembayaran = 'Lunas'
                 AND DATE(created_at) BETWEEN ? AND ?
                 ORDER BY created_at ASC";
        $stmttiket = $konek->prepare($sqltiket);
        $stmttiket->bind_param("ss", $start, $end);
        $stmttiket->execute();
        $resulttiket = $stmttiket->get_result();
        $sqlwahana = "SELECT * FROM tiketwahana
```

```
                    WHERE pembayaran = 'Lunas'
                   AND DATE(created_at) BETWEEN ? AND ?
                   ORDER BY created_at ASC";
        $stmtwahana = $konek->prepare($sqlwahana);
        $stmtwahana->bind_param("ss", $start, $end);
        $stmtwahana->execute();
        $resultwahana = $stmtwahana->get_result();
    }
    ?>
<div class="overflow-hidden rounded-lg bg-white shadow mt-10">
    <div class="hadow-lg rounded-lg p-6 w-full ">

<h2 class="text-lg font-semibold text-gray-700 mb-4 text-
center">Pilih Rentang Tanggal</h2>
            <form action="" method="post" class="space-y-4">
                <div>
                    <label for="start_date" class="block text-
sm font-medium text-gray-600">Tanggal Mulai</label>
                    <input value="2025-02-01" type="date"
name="start_date" required
                        class="mt-1 block w-full px-4 py-2
border rounded-md shadow-sm focus:ring focus:ring-blue-300
focus:outline-none">
                </div>
                <div>
                    <label for="end_date" class="block text-sm
font-medium text-gray-600">Tanggal Akhir</label>
                    <input value="2025-02-01" type="date"
name="end_date" required
                        class="mt-1 block w-full px-4 py-2
border rounded-md shadow-sm focus:ring focus:ring-blue-300
focus:outline-none">
                </div>
                <button type="submit" name="lihat_laporan"
                    class="w-full bg-blue-600 hover:bg-blue-700
text-white font-semibold py-2 rounded-md shadow-md transition
duration-300">
                    Lihat Laporan
                </button>
                <button type="submit" name="download_laporan"
                    class="w-full bg-blue-600 hover:bg-blue-700
text-white font-semibold py-2 rounded-md shadow-md transition
duration-300">
                    Download Laporan
                </button>
            </form>
        </div>
    </div>
  <?php if ($_SERVER["REQUEST_METHOD"] === "POST" &&
isset($_POST['lihat_laporan'])): ?>
        <div class="mt-6 grid grid-cols-2 gap-2 md:gap-5
sm:grid-cols-2 lg:grid-cols-2">
            <!-- Card -->
            <div class="overflow-hidden rounded-lg bg-yellow-
300/50 shadow">

?></div>
```

```
 <div class="px-5 pt-5 md:p-5">
                    <div class="flex items-center">
                        <div class="flex-shrink-0">
                        </div>
                        <div class="w-0 flex-1">
                            <dl>
                <dt class="truncate text-lg md:text-sm font-
medium text-gray-500">Tiket Masuk</dt>
                                <dd class="hidden sm:block">
                                    <div class="text-lg font-
medium text-gray-900"><?= formatRupiah($totalHargaMasuk)

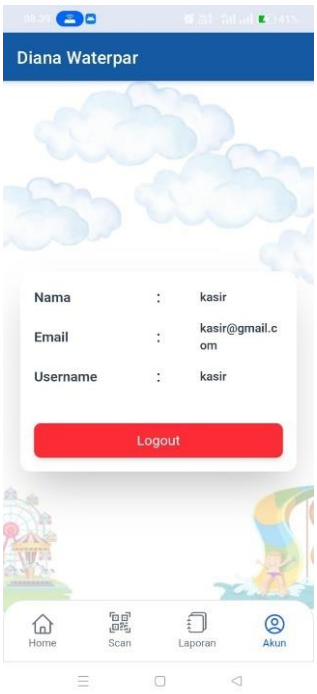
                                </dd>
                            </dl>
                        </div>
                    </div>
                </div>

        <div class="px-5 pb-3 block sm:hidden">
                    <div class="text-lg font-medium text-gray-
900"><?= formatRupiah($totalHargaMasuk) ?></div>
                </div>
                <div class="bg-red-50 px-5 py-3">
<div class="text-sm flex justify-center">
                        <div class="font-medium text-cyan-700
hover:text-cyan-900">Total Tiket : <?= $totalJumlahMasuk
?></div>
                    </div>
                </div>
            </div>
            <div class="overflow-hidden rounded-lg bg-red-
300/50 shadow">
                <div class="px-5 pt-5 md:p-5">
                    <div class="flex items-center">
                        <div class="flex-shrink-0">
                        </div>
                        <div class="w-0 flex-1">
                            <dl>
                                <dt class="truncate text-lg
md:text-sm font-medium text-gray-500">Tiket Wahana</dt>
                                <dd class="hidden sm:block">
                                    <div class="text-lg font-
medium text-gray-900"><?= formatRupiah($totalHargaWahana)
?></div>
                                </dd>
                            </dl>
                        </div>
```

d. Menu *Logout*

Menu ini berfungsi untuk mengakhiri sesi pengguna pada aplikasi. *Logout* digunakan untuk menjaga keamanan dan privasi akun, serta memungkinkan

pengguna untuk keluar dari akun saat ini dan masuk menggunakan akun lain jika diperlukan.



Gambar 4. 56 *Logout*

Berikut merupakan *sourcode* dari tampilan *logout:*

```
<div class="min-h-screen flex items-center justify-center">

    <div class="w-80 bg-white rounded-xl shadow-2xl p-4">
        <!--  Profil -->
        <div class="flex flex-col space-y-3 font-semibold">
            <div class="grid grid-cols-3 items-center">
                <h2 class="text-md text-gray-700">Nama</h2>
                <h2 class="text-lg text-gray-700 text-
center">:</h2>
                <h2 class="text-sm text-gray-700 break-
words"><?php echo htmlspecialchars($row['name']); ?> </h2>
                                    </div>
            <div class="grid grid-cols-3 items-center">
                <h2 class="text-gray-700 text-md">Email</h2>
                <h2 class="text-gray-700 text-lg text-
center">:</h2>
                <h2 class="text-gray-700 text-sm break-
words"><?php echo htmlspecialchars($row['email']); ?></h2>
                                    </div>
```

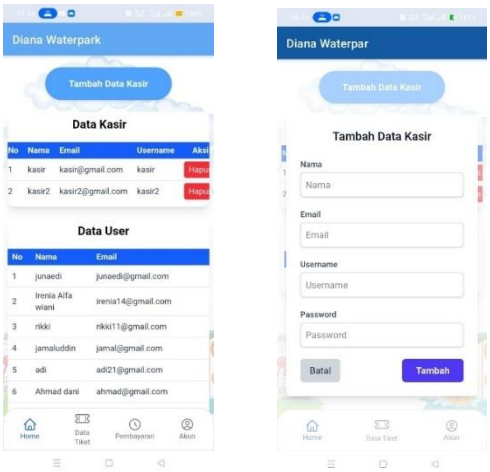
```
<div class="grid grid-cols-3 items-center">
                <h2 class="text-gray-700 text-md">Username</h2>
                <h2 class="text-gray-700 text-lg text-
center">:</h2>
                <h2 class="text-gray-700 text-sm break-
words"><?php echo htmlspecialchars($row['username']); ?></h2>
            </div>
        </div>

        <!-- Tombol Aksi -->
        <div class="mt-10 space-y-3">
            <button onclick="logout()" class="w-full px-4 py-2
bg-red-500 text-white rounded-lg shadow-md hover:bg-red-600
transition">
                Logout
            </button>
        </div>
    </div>
                            </div>
```

3. Admin

a. Menu *home*

Menu Home berfungsi untuk menampilkan data kasir dan data *user* yang telah terdaftar dalam sistem. Selain itu, melalui menu ini admin dapat menambahkan data kasir baru, menghapus data kasir yang tidak lagi aktif, serta mengelola data user secara efisien. Menu ini menjadi pusat pengelolaan informasi terkait pengguna sistem agar tetap terorganisir dan terkini.



Gambar 4. 57 *Home*

Berikut merupakan tampilan dari *home* kasir dan tambah data kasir :

```
<div class="flex items-center justify-center pt-[20px]">
    <div class="flex">
        <button type="button" onclick="openModal()" class="px-
10 py-4 font-bold bg-blue-400 text-white rounded-full shadow-md
hover:bg-blue-600 transition">
            Tambah Data Kasir
        </button>
    </div>
</div>
<div class="flex items-center justify-center pt-[20px] mx-2">
    <div class="w-full p-2 rounded-xl bg-white shadow-lg
overflow-x-auto">
        <h2 class="text-xl font-bold text-center mb-4">Data
Kasir</h2>
        <table class="min-w-full divide-y divide-gray-200 text-
sm text-left">
            <thead class="bg-blue-600 text-white">
                <tr>
                    <th class="px-2 py-1">No</th>
                    <th class="px-2 py-1">Nama</th>
                    <th class="px-2 py-1">Email</th>
                    <th class="px-2 py-1">Username</th>
                    <th class="px-2 py-1 text-center">Aksi</th>
                </tr>
            </thead>
            <tbody class="bg-white divide-y divide-gray-200
text-gray-800">
                <?php while ($row = $result->fetch_assoc()): ?>
                    <tr>
                        <td class="px-2 py-1"><?= $no++ ?></td>
                        <td class="px-2 py-1"><?=
htmlspecialchars($row['name']) ?></td>
                        <td class="px-2 py-1"><?=
htmlspecialchars($row['email']) ?></td>
<td class="px-2 py-1"><?= htmlspecialchars($row['username'])
?></td>
                        <td class="px-2 py-1 text-center">
                            <form action="proses_user.php"
method="POST" class="inline">
                                <input type="hidden"
name="action" value="DELETE">
                                <input type="hidden" name="id"
value="<?= $row['id'] ?>">
                                <input type="hidden"
name="id_user" value="<?= $_GET['id'] ?>">
                                <button type="submit"
class="bg-red-500 text-white px-2 py-1 rounded hover:bg-red-
600">Hapus</button>
            </form>
```

```
</td>
                    </tr>
                <?php endwhile; ?>
            </tbody>
        </table>
    </div>
</div>

<div class="flex items-center justify-center pt-[20px] mx-2">
    <div class="w-full p-2 rounded-xl bg-white shadow-lg
overflow-x-auto">
        <h2 class="text-xl font-bold text-center mb-4">Data
User</h2>
        <table class="min-w-full divide-y divide-gray-200 text-
sm text-left">
            <thead class="bg-blue-600 text-white">
                <tr>
                    <th class="px-2 py-1">No</th>
                    <th class="px-2 py-1">Nama</th>
                    <th class="px-2 py-1">Username</th>
                    <th class="px-2 py-1">Email</th>
                    <th class="px-2 py-1 text-center">Aksi</th>
                </tr>
            </thead>
            <tbody class="bg-white divide-y divide-gray-200
text-gray-800">
                <?php while ($row = $resultUser-
>fetch_assoc()): ?>
                    <tr>
                        <td class="px-2 py-1"><?= $noUser++
?></td>
                        <td class="px-2 py-1"><?=
htmlspecialchars($row['name']) ?></td>
                        <td class="px-2 py-1"><?=
htmlspecialchars($row['email']) ?></td>
                        <td class="px-2 py-1"><?=
htmlspecialchars($row['username']) ?></td>
                        <td class="px-2 py-1 text-center">
                            <form action="proses_user.php"
method="POST" class="inline">
                                <input type="hidden"
name="action" value="DELETE">
                                <input type="hidden" name="id"
value="<?= $row['id'] ?>">
<input type="hidden" name="id_user" value="<?= $_GET['id'] ?>">
                                <button type="submit"
class="bg-red-500 text-white px-2 py-1 rounded hover:bg-red-
600">Hapus</button>
                            </form>
                        </td>
                    </tr>
                <?php endwhile; ?>
            </tbody>
        </table>
    </div>
</div>
```

```
<div id="tambahDataKasir" class="fixed inset-0 z-50 hidden flex
items-center justify-center bg-white/50 bg-opacity-50 p-4 sm:p-
0">
    <div class="bg-white rounded-lg shadow-lg w-full max-w-md
sm:max-w-lg p-6 sm:p-8 overflow-auto max-h-full">
        <h2 class="text-xl font-semibold text-gray-900 mb-6
text-center">Tambah Data Kasir</h2>

<form method="POST" action="proses_user.php"
enctype="multipart/form-data">
            <input type="hidden" name="action" value="CREATE">
            <input type="hidden" name="id_user" value="<?=
$_GET['id'] ?>">
            <input type="hidden" name="role">

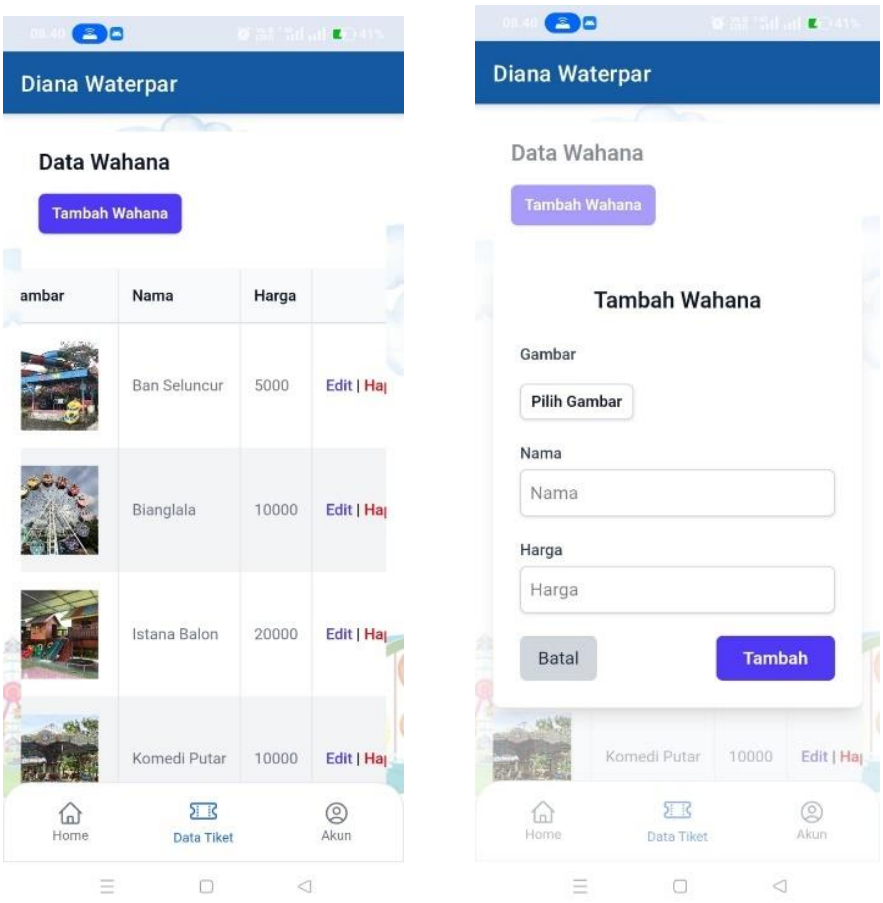
            <div class="mb-5">
                <label for="name" class="block text-sm font-
medium text-gray-700">Nama</label>
                <input type="text" id="name" name="name"
autocomplete="0" required class="mt-1 p-2 block w-full border
border-gray-300 rounded-md shadow-sm" placeholder="Nama">
            </div>
            <div class="mb-5">
                <label for="email" class="block text-sm font-
medium text-gray-700">Email</label>
                <input type="E" id="email" name="email"
required class="mt-1 p-2 block w-full border border-gray-300
rounded-md shadow-sm" placeholder="Email">
            </div>

            <div class="mb-5">
                <label for="username" class="block text-sm
font-medium text-gray-700">Username</label>
                <input type="text" id="username"
name="username" required class="mt-1 p-2 block w-full border
border-gray-300 rounded-md shadow-sm" placeholder="Username">
            </div>

            <div class="mb-5">
                <label for="password" class="block text-sm
font-medium text-gray-700">Password</label>
                <input type="password" id="password"
name="password" required class="mt-1 p-2 block w-full border
border-gray-300 rounded-md shadow-sm" placeholder="Password">
            </div>
            <div class="flex justify-between items-center">
                <button type="button" class="px-4 py-2 bg-gray-
300 text-gray-800 rounded-md"
onclick="closeModal()">Batal</button>
                <button type="submit" class="px-6 py-2 bg-
indigo-600 text-white font-semibold rounded-md">Tambah</button>
            </div>
        </form>
    </div>
</div>
```

b. Menu Data tiket

Menu ini berfungsi untuk menampilkan data tiket masuk maupun data tiket wahana yang tersedia dalam sistem. Selain itu, melalui menu ini admin dapat menambahkan wahana baru, mengedit informasi wahana yang sudah ada, serta menghapus wahana yang sudah tidak digunakan. Menu ini memudahkan pengelolaan data tiket dan wahana agar selalu terupdate dan sesuai dengan kondisi operasional.



Gambar 4. 58 Data Tiket

Berikut merupakan *sourcode* dari tampilan halaman data tiket wahan, tambah wahana edit serta hapus :

```
<div class="px-4 sm:px-6 lg:px-8 bg-white m-4 rounded-lg">
    <div class="sm:flex sm:items-center pt-4">
        <div class="sm:flex-auto">
            <h1 class="text-xl font-semibold leading-6 text-
gray-900">Data Wahana</h1>
        </div>
        <div class="mt-4 sm:ml-16 sm:mt-0 sm:flex-none">
            <button onclick="openModal()" type="button"
class="block rounded-md bg-indigo-600 px-3 py-2 text-center
text-sm font-semibold text-white shadow-sm hover:bg-indigo-
500">Tambah Wahana</button>
        </div>
    </div>
    <div class="mt-8 flow-root pb-4">
        <div class="-mx-4 -my-2 overflow-x-auto sm:-mx-6 lg:-
mx-8">
            <div class="inline-block min-w-full py-2 align-
middle sm:px-6 lg:px-8">
                <div class="overflow-hidden shadow sm:rounded-
lg">
                    <table class="min-w-full divide-y divide-
gray-300">
                        <thead class="bg-gray-50">
                            <tr class="divide-x divide-gray-
200">
                                <th scope="col" class="min-w-
[110px] w-[110px] px-3 py-3.5 text-left text-sm font-semibold
text-gray-900">Gambar</th>
                                <th scope="col" class="px-3 py-
3.5 text-left text-sm font-semibold text-gray-900">Nama</th>
                                <th scope="col" class="px-3 py-
3.5 text-left text-sm font-semibold text-gray-900">Harga</th>
                                <th scope="col" class="relative
py-3.5 pl-3 pr-4 sm:pr-6 w-28">
                                    <span class="sr-
only">Aksi</span>
                                </th>
                            </tr>
                        </thead>

<tbody class="bg-white">
                            <?php if ($resultWahana->num_rows >
0): ?>
                                <?php while ($row =
$resultWahana->fetch_assoc()): ?>
                                    <tr class="divide-x divide-
gray-200 even:bg-gray-100">
                                        <td class="whitespace-
nowrap px-3 py-4 text-sm text-gray-500"><img loading="lazy"
class="w-20 h-20 object-cover" src="./img/<?php echo
htmlspecialchars($row['gambar']); ?>" alt=""></td>
```

```
                                        <td class="relative
whitespace-nowrap py-4 pl-3 pr-4 text-right text-sm font-medium
sm:pr-6">
                                            <button
onclick="openEditModal('<?php echo $row['id_wahana']; ?>',
'<?php echo $row['nama']; ?>', '<?php echo $row['harga']; ?>',
'<?php echo $row['gambar']; ?>')" class="text-indigo-600
hover:text-indigo-900">Edit</button> |
                                            <form
id="deleteForm" method="POST" action="proses_datatiket.php"
style="display:inline;">
                                                <input
type="hidden" name="action" value="DELETE">
                                                <input
type="hidden" name="id_user" value="<?= $_GET['id'] ?>">
                                                <input
type="hidden" name="id_wahana" value="<?php echo
$row['id_wahana']; ?>">
                                                <button
type="button" class="text-red-600 hover:text-red-900"
onclick="openDeleteModal()">Hapus</button>
                                            </form>
                                        </td>
                                    </tr>
                                <?php endwhile; ?>
                            <?php else: ?>
                                <tr>
                                    <td colspan="4"
class="whitespace-nowrap px-3 py-4 text-sm text-gray-500 text-
center">Tidak Ada Data Wahana, Silahkan Tambahkan Wahana.</td>
                                </tr>
                            <?php endif; ?>
                        </tbody>
                    </table>
</div>
            </div>
        </div>
    </div>
</div>

<div id="tambahModal" class="fixed inset-0 z-50 hidden flex
items-center justify-center bg-white/50 bg-opacity-50 p-4 sm:p-
0">
    <div class="bg-white rounded-lg shadow-lg w-full max-w-md
sm:max-w-lg p-6 sm:p-8 overflow-auto max-h-full">
        <h2 class="text-xl font-semibold text-gray-900 mb-6
text-center">Tambah Wahana</h2>
<form method="POST" action="proses_datatiket.php"
enctype="multipart/form-data">
            <input type="hidden" name="action" value="CREATE">
            <input type="hidden" name="id_user" value="<?=
$_GET['id'] ?>">
            <div class="mb-5">
```

```
                <input type="file" id="update_image"
name="image" class="hidden mt-1 p-2 block w-full border border-
gray-300 rounded-md shadow-sm" placeholder="Gambar"
onchange="previewUpdateImage(event)">
                <img id="updateImagePreview" class="mt-4 w-28
h-28 hidden object-cover" alt="Image Preview">
                <label for="update_image" class="mt-4 inline-
block cursor-pointer rounded-md bg-white px-2.5 py-1.5 text-sm
font-semibold text-gray-900 shadow-sm ring-1 ring-inset ring-
gray-300 hover:bg-gray-50">
                    Pilih Gambar
                </label>
            </div>

            <div class="mb-5">
                <label for="update_name" class="block text-sm
font-medium text-gray-700">Nama</label>
                <input type="text" id="update_name" name="nama"
autocomplete="off" required class="mt-1 p-2 block w-full border
border-gray-300 rounded-md shadow-sm" placeholder="Nama">
            </div>
            <div class="mb-5">
                <label for="update_harga" class="block text-sm
font-medium text-gray-700">Harga</label>
                <input type="number" id="update_harga"
name="harga" required class="mt-1 p-2 block w-full border
border-gray-300 rounded-md shadow-sm" placeholder="Harga">
            </div>
            <div class="flex justify-between items-center">
                <button type="button" class="px-4 py-2 bg-gray-
300 text-gray-800 rounded-md"
onclick="closeEditModal()">Batal</button>
                <button type="submit" class="px-6 py-2 bg-
indigo-600 text-white font-semibold rounded-md">Update</button>
            </div>
        </form>
    </div>
</div>
<div id="DeleteModal" class="fixed inset-0 z-50 hidden flex
items-center justify-center bg-white/50 bg-opacity-50 p-4 sm:p-
0">
        <p class="text-gray-700 text-center">Apakah Anda Yakin
Ingin Menghapus Wahana Ini?</p>
        <div class="flex justify-between items-center mt-6">
            <button type="button" onclick="closeDeleteModal()"
class="px-4 py-2 bg-gray-300 text-gray-800 rounded-md shadow
hover:bg-gray-400 focus:outline-none focus:ring-2 focus:ring-
gray-500 focus:ring-offset-2">Batal</button>
      <button id="confirmDeleteButton" class="px-4 py-2 bg-red-
600 text-white rounded-md shadow hover:bg-red-500
focus:outline-none focus:ring-2 focus:ring-red-600 focus:ring-
offset-2">Hapus</button>
        </div>
    </div>
</div>
```

```
if (imageFileName) {
            const imageUrl = `img/${imageFileName}`;
            imagePreview.src = imageUrl;
            imagePreview.classList.remove('hidden');
        } else {
            imagePreview.classList.add('hidden');
        }
        document.getElementById('updateModal').classList.remove
('hidden');
    }
    function closeEditModal() {
    document.getElementById('updateModal').classList.add('hidde
n');
    }
    let deleteForm;
function openDeleteModal() {
        deleteForm = event.target.closest('form');
        document.getElementById('DeleteModal').classList.remove
('hidden');
    }
    function closeDeleteModal() {
        document.getElementById('DeleteModal').classList.add('h
idden');
    }
    document.getElementById('confirmDeleteButton').addEventList
ener('click', function() {
        if (deleteForm) {
            deleteForm.submit();
        }
    });
</script>
```

c. Ubah Harga Tiket Masuk

Fitur ini berfungsi untuk mengubah harga tiket masuk sesuai kebutuhan operasional atau kebijakan manajemen. Dengan adanya fitur ini, admin dapat memperbarui harga tiket secara langsung di sistem sehingga informasi harga yang ditampilkan kepada pengguna selalu akurat dan terkini.



Gambar 4. 59 Ubah Harga Tiket Masuk

berikut merupakan *sourcode* dari ubah harga tiket masuk:

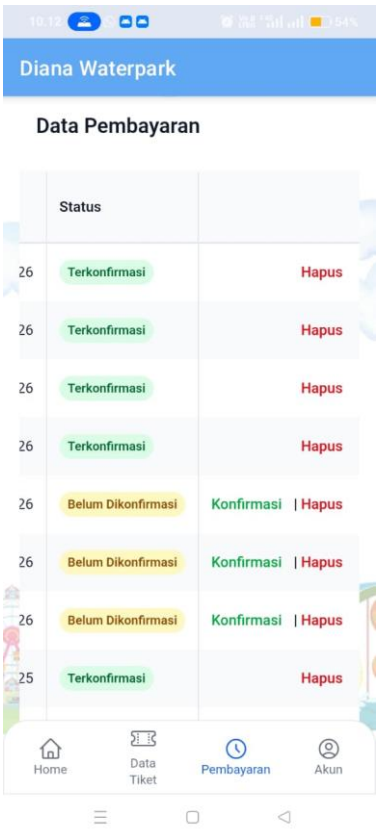
```
<script>
    function previewUpdateImage(event) {
        const file = event.target.files[0];
        const preview =
document.getElementById('updateImagePreview');

        if (file) {
            const reader = new FileReader();
            reader.onload = function(e) {
                preview.src = e.target.result;
                preview.classList.remove('hidden');
            };
            reader.readAsDataURL(file);
        }
    }
    function previewImage(event) {
        const image = document.getElementById('image');
```

```
<div class="px-4 sm:px-6 lg:px-8 bg-white m-4 rounded-lg mb-
[100px] mt-10">
    <div class="sm:flex sm:items-center pt-4">
        <div class="sm:flex-auto">
            <h1 class="text-xl font-semibold leading-6 text-
gray-900">Data Tiket Masuk</h1>
        </div>
    </div>
<div class="mt-8 flow-root pb-4">
        <div class="-mx-4 -my-2 overflow-x-auto sm:-mx-6 lg:-
mx-8">
            <div class="inline-block min-w-full py-2 align-
middle sm:px-6 lg:px-8">
                <div class="overflow-hidden shadow sm:rounded-
lg">
          <div class="flex items-center justify-center pt-
[20px]">
             <div class="flex p-4">
<form method="POST" action="proses_datatiket.php">
                                    <input type="hidden"
name="action" value="UPDATETIKETMASUK">
                                    <input type="hidden"
name="id_user" value="<?= $_GET['id'] ?>">
                                    <input type="hidden"
name="id_data" value="<?= $id_data ?>">
                                    <div class="mb-5">
                                        <label for="harga"
class="block text-sm font-medium text-gray-700">Harga Tiket
Masuk</label>
                                        <input type="number"
value="<?= $harga ?>" id="harga" name="harga"
autocomplete="off" required class="mt-1 p-2 block w-full border
border-gray-300 rounded-md shadow-sm" placeholder="Harga Tiket
Masuk">
                                    </div>
                                    <div class="flex justify-
between items-center">
                                        <button type="submit"
class="px-6 py-2 bg-indigo-600 text-white font-semibold
rounded-md">Ubah Harga Tiket Masuk</button>
                                    </div>
                                </form>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
```

d. Menu Pembayaran

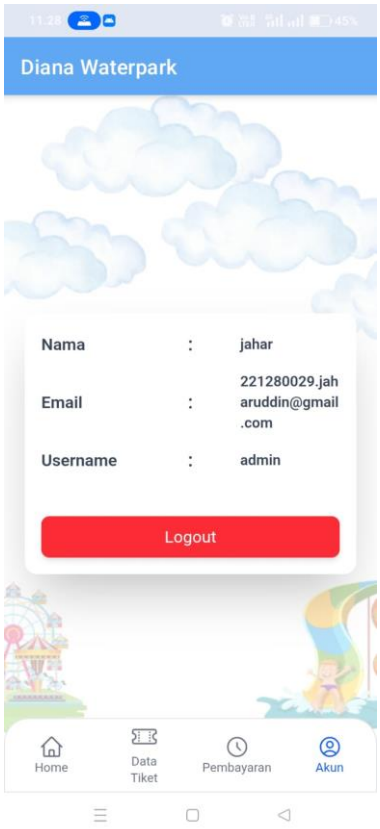
Menu ini berfungsi untuk menampilkan data pembayaran dari user.pembayaran digunakan untuk mengonfirmasi serta menghapus pembayaran dari user jika pembayaran user telah berhasil maka admin akan menekan button konfirmasi sebagai bukti bahwa user telah membeli tiket dan otomatis *qr code* pada user akan muncul.



Gambar 4. 60 Menu Pembayaran

e. Menu *logout*

Menu ini berfungsi untuk mengakhiri sesi pengguna pada aplikasi. *Logout* digunakan untuk menjaga keamanan dan privasi akun, serta memungkinkan pengguna untuk keluar dari akun saat ini dan masuk menggunakan akun lain jika diperlukan.



Gambar 4. 61 *Logout*

Berikut merupakan *sourcode* dari menu *logout:*

```
<div class="min-h-screen flex items-center justify-center">

    <div class="w-80 bg-white rounded-xl shadow-2xl p-4">
        <!--  Profil -->
        <div class="flex flex-col space-y-3 font-semibold">
            <div class="grid grid-cols-3 items-center">
                <h2 class="text-md text-gray-700">Nama</h2>
                <h2 class="text-lg text-gray-700 text-
center">:</h2>
                <h2 class="text-sm text-gray-700 break-
words"><?php echo htmlspecialchars($row['name']); ?> </h2>
</div>
            <div class="grid grid-cols-3 items-center">
                <h2 class="text-gray-700 text-md">Email</h2>
                <h2 class="text-gray-700 text-lg text-
center">:</h2>
                <h2 class="text-gray-700 text-sm break-
words"><?php echo htmlspecialchars($row['email']); ?></h2>
            </div>
      <div class="grid grid-cols-3 items-center">
                <h2 class="text-gray-700 text-md">Username</h2>
                <h2 class="text-gray-700 text-lg text-
center">:</h2>
```

```
                <h2 class="text-gray-700 text-sm break-
words"><?php echo htmlspecialchars($row['username']); ?></h2>
            </div>
        </div>

        <!-- Tombol Aksi -->
        <div class="mt-10 space-y-3">
            <button onclick="logout()" class="w-full px-4 py-2
bg-red-500 text-white rounded-lg shadow-md hover:bg-red-600
transition">
                Logout
            </button>
        </div>
    </div>
</div>
```

4. Pimpinan

f. Menu *rating*

Menu ini berfungsi untuk menampilkan penilaian dari user baik itu *rating* maupun komentar. Menu *rating* digunakan untuk mengetahui penilaian mengenai layanan yang tersedia pada wisata diana *waterpark* pimpinan dapat melihat rating positif maupun negatif dari *user,* sehingga pimpinan dapat mengevaluasi lebih baik lagi layanan yang dianggap kurang bagi *user.*



Gambar 4. 62 Menu *Rating*

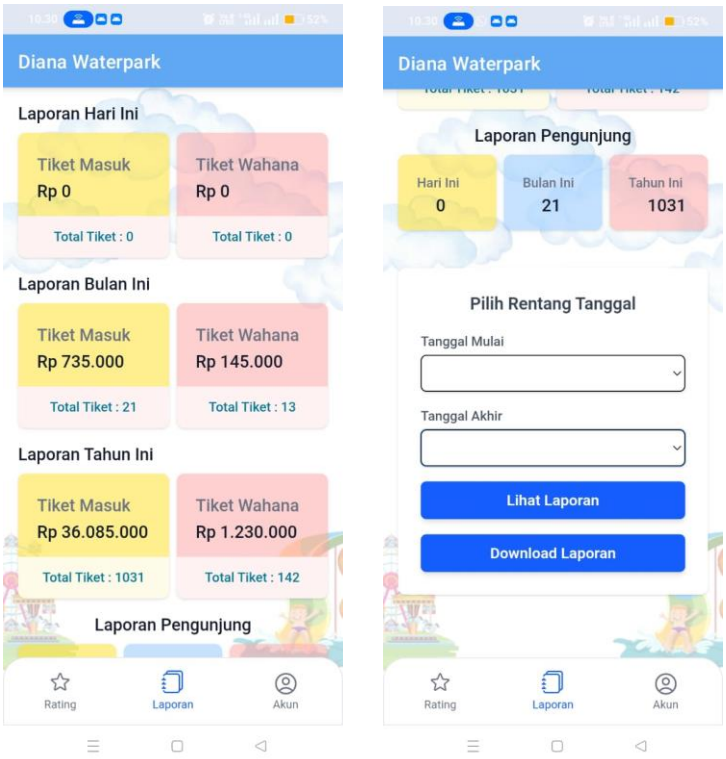
Berikut merupakan *sourcode* dari menu *rating*:

```
<div class="mx-6 mt-6 mb-[180px]">
    <h1 class="text-2xl font-bold text-gray-800 mb-
6">Selamat Datang Pimpinan</h1>
    <p class="text-gray-600 mb-6">Anda bisa melihat rating
pengguna.</p>
    <div class="mb-6 bg-white p-5 rounded-xl shadow border
border-gray-200">
        <h2 class="text-lg font-semibold text-gray-800 mb-
4">Ringkasan Rating</h2>
        <?php for ($i = 5; $i >= 1; $i--):
            $count = $rating_counts[$i];
            $percent = $total_rating > 0 ? round(($count /
$total_rating) * 100) : 0;
        ?>
            <div class="flex items-center mb-2">
                <div class="w-12 text-sm font-medium text-
gray-700"><?php echo $i; ?> ★</div>
                <div class="flex-1 h-3 bg-gray-200 rounded
overflow-hidden mx-2">
                    <div class="h-full bg-yellow-400"
style="width: <?php echo $percent; ?>%"></div>
                </div>
                <div class="w-10 text-sm text-gray-
600"><?php echo $percent; ?>%</div>
            </div>
        <?php endfor; ?>
    </div>
    <div class="space-y-2">
        <?php if ($resultrating->num_rows > 0): ?>
            <?php while ($datarating = $resultrating-
>fetch_assoc()): ?>
                <div class="bg-white p-5 rounded-xl shadow
border border-gray-200">
<div class="flex items-start gap-4">
                        <!-- Avatar -->
                        <div class="h-11 w-11 rounded-full
bg-gray-300 flex items-center justify-center text-white
font-bold text-lg">
                            <?php echo
strtoupper(substr($datarating['name'], 0, 1)); ?>
                        </div>
                        <!-- Konten -->
                        <div class="flex-1">
                            <div class="flex justify-
between items-start">
                                <div class="font-semibold
text-gray-900 text-sm">
                                    <?php echo
htmlspecialchars($datarating['name']); ?>
                                </div>
                                <div class="text-xs text-
gray-400">
                                    <?php echo date('d M
Y', strtotime($datarating['tanggal'])); ?>
                </div>
                            </div>
```

```
<!-- Bintang -->
                                <div class="flex items-center
mt-1">
                                    <?php
                                    $rating =
(int)$datarating['rating'];
                                    for ($i = 1; $i <= 5;
$i++) {
                                        $color = $i <= $rating
? 'text-yellow-400' : 'text-gray-300';
                                        echo "<svg class='h-5
w-5 $color' fill='currentColor' viewBox='0 0 20 20'>
                                            <path d='M10 15l-
5.878 3.09 1.122-6.545L.488 6.91l6.562-.955L10 0l2.95
5.955 6.562.955-4.756 4.635 1.122 6.545z'/>
                                          </svg>";
                                    }
                                    ?>
                                </div>
                                <!-- Komentar -->
                                <p class="mt-2 text-sm text-
gray-700 leading-relaxed">"<?php echo
nl2br(htmlspecialchars($datarating['komentar'])); ?>"
                                </p>
                            </div>
                        </div>
                    </div>
                <?php endwhile; ?>
            <?php else: ?>
                <div class="text-center text-gray-500">Tidak
ada feedback yang tersedia.</div>
            <?php endif; ?>
        </div>
                            </div>
```

g. Menu laporan

Menu ini berfungsi untuk menampilkan laporan dari penjualan tiket yang dapat dilihat perhari,perbulan dan juga pertahun dan juga terdapat menu pilihan laporan tertentu dengan memasukkan rentang tanggal yang di inginkan setelah itu laporan akan muncul sesuai dengan rentang tanggal yang di isi serta dapat di download dan tersimpan dipenyimpanan *handphone.*



Gambar 4. 63 Laporan

Berikut merupakan *sourcode* dari menu laporan:

```
<!-- // untuk memamnggil tanggal atau rentang tanggal nya
-->
    <?php
    $totalJumlahMasuk = 0;
    $totalHargaMasuk = 0;
    $totalJumlahWahana = 0;
    $totalHargaWahana = 0;
    // Lihat Laporan
    if ($_SERVER["REQUEST_METHOD"] === "POST" &&
isset($_POST['start_date'], $_POST['end_date'],
$_POST['lihat_laporan'])) {
        $start = $_POST['start_date'];
        $end = $_POST['end_date'];
// Tiket Masuk
        $stmt = $konek->prepare("SELECT SUM(jumlah_tmasuk)
AS total_jumlah, SUM(total_harga) AS total_harga FROM
tiketmasuk WHERE DATE(created_at) BETWEEN ? AND ?");
        $stmt->bind_param("ss", $start, $end);
        $stmt->execute();
        $result = $stmt->get_result();
        $row = $result->fetch_assoc();
        $totalJumlahMasuk = $row['total_jumlah'] ?? 0;
        $totalHargaMasuk = $row['total_harga'] ?? 0;
```

```
            // Tiket Wahana
            $stmt = $konek->prepare("SELECT jumlah_twahana,
    total_harga FROM tiketwahana WHERE DATE(created_at)
    BETWEEN ? AND ?");
            $stmt->bind_param("ss", $start, $end);
            $stmt->execute();
            $result = $stmt->get_result();
            while ($row = $result->fetch_assoc()) {
                // Memecah jumlah_twahana yang dipisahkan koma
                $jumlahArrayWahana = explode(',',
    $row['jumlah_twahana']);

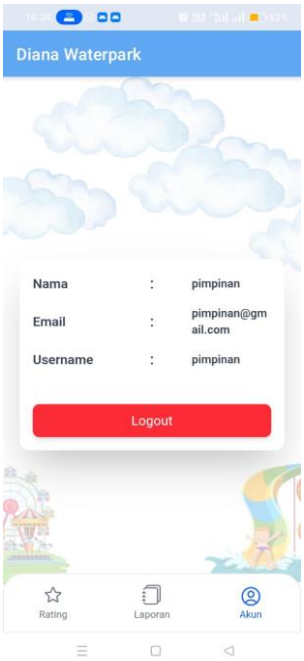
                // Menghitung total jumlah tiket wahana
                foreach ($jumlahArrayWahana as $jumlahWahana)
    {
                    if (is_numeric($jumlahWahana)) {
                        $totalJumlahWahana += (int)
    $jumlahWahana;
                    }
                }
                // Menambahkan total harga tiket wahana
                $totalHargaWahana += $row['total_harga'] ?? 0;
            }
        }
        // Download Laporan
        if (
            $_SERVER["REQUEST_METHOD"] === "POST"
            && isset($_POST['start_date'], $_POST['end_date'],
    $_POST['download_laporan'])
        ) {
            $_SESSION['laporan_didownload'] = true; // Set
    session
            $start = $_POST['start_date'];
            $end = $_POST['end_date'];
            $sqltiket = "SELECT * FROM tiketmasuk
    WHERE pembayaran = 'Lunas'
                    AND DATE(created_at) BETWEEN ? AND ?
                    ORDER BY created_at ASC";
            $stmttiket = $konek->prepare($sqltiket);
            $stmttiket->bind_param("ss", $start, $end);
            $stmttiket->execute();
            $resulttiket = $stmttiket->get_result();
            $sqlwahana = "SELECT * FROM tiketwahana
                    WHERE pembayaran = 'Lunas'
                    AND DATE(created_at) BETWEEN ? AND ?
                    ORDER BY created_at ASC";
            $stmtwahana = $konek->prepare($sqlwahana);
          $stmtwahana->bind_param("ss", $start, $end);
            $stmtwahana->execute();
            $resultwahana = $stmtwahana->get_result();
        }
        ?>
    <div class="overflow-hidden rounded-lg bg-white shadow mt-
    10">
            <div class="hadow-lg rounded-lg p-6 w-full ">
```

```
<form action="" method="post" class="space-y-4">
                <div>
                    <label for="start_date" class="block
text-sm font-medium text-gray-600">Tanggal Mulai</label>
                    <input value="2025-02-01" type="date"
name="start_date" required
                        class="mt-1 block w-full px-4 py-2
border rounded-md shadow-sm focus:ring focus:ring-blue-300
focus:outline-none">
                </div>
                <div>
                    <label for="end_date" class="block
text-sm font-medium text-gray-600">Tanggal Akhir</label>
                    <input value="2025-02-01" type="date"
name="end_date" required
                        class="mt-1 block w-full px-4 py-2
border rounded-md shadow-sm focus:ring focus:ring-blue-300
focus:outline-none">
                </div>
                <button type="submit" name="lihat_laporan"
                    class="w-full bg-blue-600 hover:bg-
blue-700 text-white font-semibold py-2 rounded-md shadow-
md transition duration-300">
                    Lihat Laporan
                </button>
                <button type="submit"
name="download_laporan"
                    class="w-full bg-blue-600 hover:bg-
blue-700 text-white font-semibold py-2 rounded-md shadow-
md transition duration-300">
                    Download Laporan
                </button>
</form>
        </div>
    </div>
    <?php if ($_SERVER["REQUEST_METHOD"] === "POST" &&
isset($_POST['lihat_laporan'])): ?>
        <div class="mt-6 grid grid-cols-2 gap-2 md:gap-5
sm:grid-cols-2 lg:grid-cols-2">
            <!-- Card -->
            <div class="overflow-hidden rounded-lg bg-
yellow-300/50 shadow">
                <div class="px-5 pt-5 md:p-5">
                    <div class="flex items-center">
                        <div class="flex-shrink-0">
            </div>
                        <div class="w-0 flex-1">
                            <dl>
                                <dt class="truncate text-
lg md:text-sm font-medium text-gray-500">Tiket Masuk</dt>
                                <dd class="hidden
sm:block">
```

```
<div class="text-lg font-medium text-gray-900"><?=
formatRupiah($totalHargaMasuk) ?></div>
                                </dd>
                            </dl>
                        </div>
                    </div>
                </div>
<div class="px-5 pb-3 block sm:hidden">
                    <div class="text-lg font-medium text-
gray-900"><?= formatRupiah($totalHargaMasuk) ?></div>
                </div>
                <div class="bg-red-50 px-5 py-3">
                    <div class="text-sm flex justify-
center">
                        <div class="font-medium text-cyan-
700 hover:text-cyan-900">Total Tiket : <?=
$totalJumlahMasuk ?></div>
                    </div>
                </div>
            </div>
            <div class="overflow-hidden rounded-lg bg-red-
300/50 shadow">
                <div class="px-5 pt-5 md:p-5">
                    <div class="flex items-center">
                        <div class="flex-shrink-0">
                        </div>
                        <div class="w-0 flex-1">
                            <dl>
                                <dt class="truncate text-
lg md:text-sm font-medium text-gray-500">Tiket Wahana</dt>
                                <dd class="hidden
sm:block">
                                    <div class="text-lg
font-medium text-gray-900"><?=
formatRupiah($totalHargaWahana) ?></div>
                                </dd>
                            </dl>
                        </div>
          </div>
                </div>
                <div class="px-5 pb-3 block sm:hidden">
                    <div class="text-lg font-medium text-
gray-900"><?= formatRupiah($totalHargaWahana) ?></div>
                </div>
                <div class="bg-red-50 px-5 py-3">
                    <div class="text-sm flex justify-
center">
                        <div class="font-medium text-cyan-
700 hover:text-cyan-900">Total Tiket : <?=
$totalJumlahWahana ?></div>
          </div>
                </div>
            </div>
        </div>
  <?php endif; ?>
```

h. Menu *logout*

Menu ini berfungsi untuk mengakhiri sesi pengguna pada aplikasi. *Logout* digunakan untuk menjaga keamanan dan privasi akun, serta memungkinkan pengguna untuk keluar dari akun saat ini dan masuk menggunakan akun lain jika diperlukan.



Gambar 4. 64 *Logout*

Berikut merupakan *sourcode* dari tampilan *logout*:

```
<div class="min-h-screen flex flex-col items-center justify-
center">

    <div class="w-80 bg-white rounded-xl shadow-2xl p-4">
        <!--  Profil -->
        <div class="flex flex-col space-y-3 font-semibold">
            <div class="grid grid-cols-3 items-center">
                <h2 class="text-md text-gray-700">Nama</h2>
                <h2 class="text-lg text-gray-700 text-
center">:</h2>
                <h2 class="text-sm text-gray-700 break-
words"><?php echo htmlspecialchars($row['name']); ?> </h2>
                 </div>
<div class="grid grid-cols-3 items-center">
                <h2 class="text-gray-700 text-md">Email</h2>
                <h2 class="text-gray-700 text-lg text-
center">:</h2>
```

```
<h2 class="text-gray-700 text-sm break-words"><?php echo
htmlspecialchars($row['email']); ?></h2>
            </div>
            <div class="grid grid-cols-3 items-center">
                <h2 class="text-gray-700 text-md">Username</h2>
                <h2 class="text-gray-700 text-lg text-
center">:</h2>
                <h2 class="text-gray-700 text-sm break-
words"><?php echo htmlspecialchars($row['username']); ?></h2>
            </div>
        </div>

        <!-- Tombol Aksi -->
        <div class="mt-10 space-y-3">
            <button onclick="logout()" class="w-full px-4 py-2
bg-red-500 text-white rounded-lg shadow-md hover:bg-red-600
transition">
                Logout
            </button>
        </div>
    </div>
```

C. Pengujian Sistem

1. *Blackbox*

   a. *Blackbox testing login user*

Tabel 4. 5 *blackbox testing login user*

Test faktor	Hasil	Keterangan
Memasukkan username dan password yang sesuai	✔	Sistem akan menampilkan halaman login
Screenshot		
		

   b. *Blackbox testing* registrasi *user*

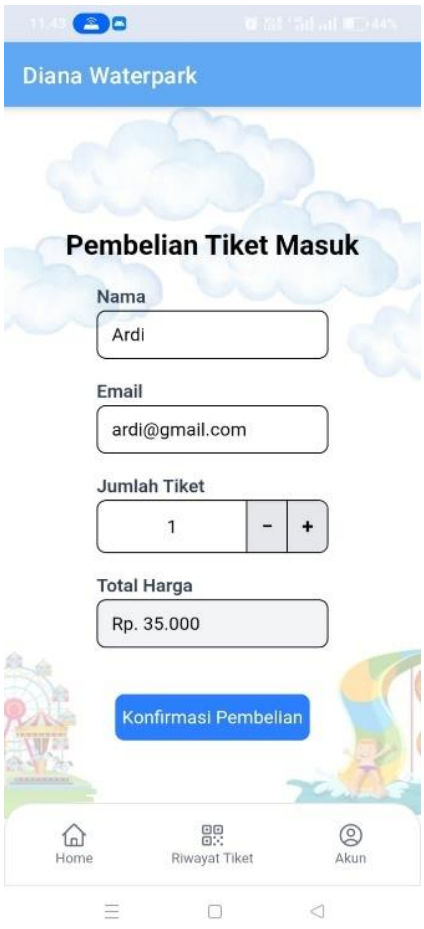
Tabel 4. 6 *blackbox testing* registrasi *user*

Test faktor	Hasil	Keterangan
Mengisi form yg sesuai	✔	Sistem akan menampilkan halaman registrasi
Screenshot		
		

   c. *Blackbox testing* tiket masuk

Tabel 4. 7 *Blackbox testing* tiket masuk

Test faktor	Hasil	Keterangan
Mengisi form yg sesuai	✔	Sistem akan menampilkan struk pembelian tiket dan menuju ke halaman home
Screenshot		

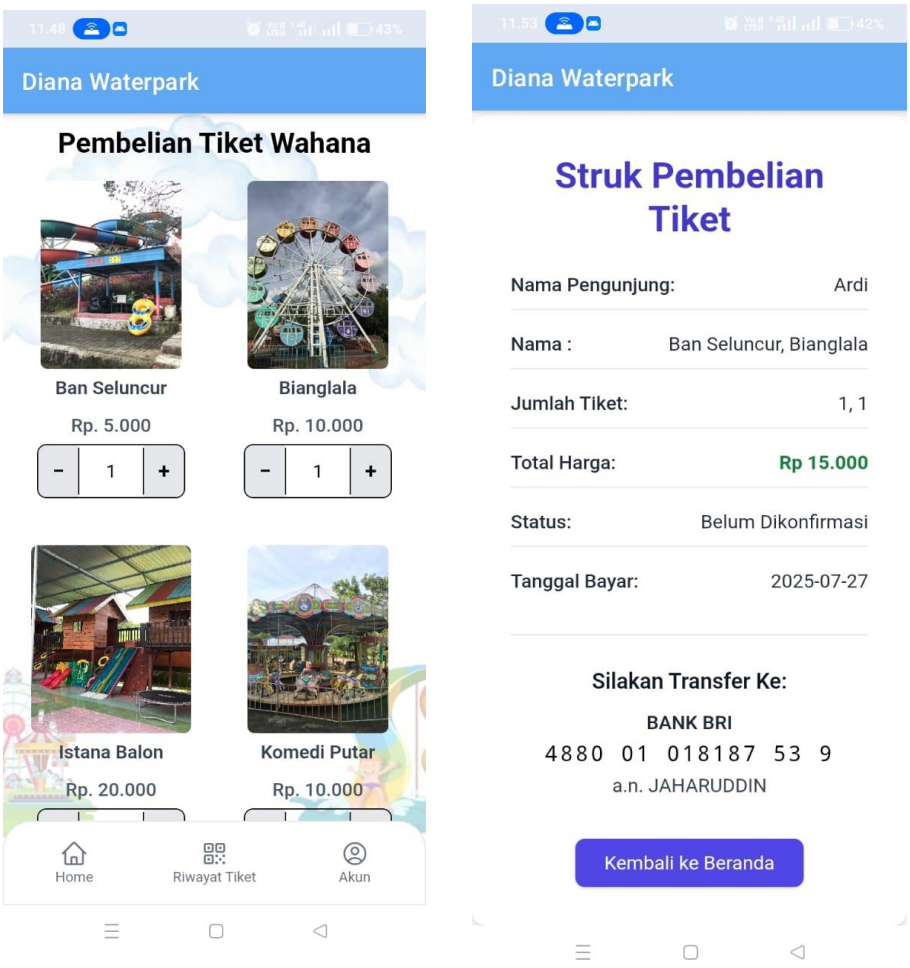




d. *Blackbox testing* tiket wahana

Tabel 4. 8 *Blackbox testing* tiket wahana

Test faktor	Hasil	Keterangan
Mengisi form yg sesuai	✔	Sistem akan menampilkan struk pembelian tiket dan menuju ke halaman home
Screenshot		



e. *Blackbox testing* riwayat tiket

Tabel 4. 9 *Blackbox testing* riwayat tiket

Test faktor	Hasil	Keterangan
Menampilkan tiket yang belum digunakan dan sudah digunakan	✔	Sistem akan menampilkan qr code dan tombol unduh untuk mendownload qrcode nya
Screenshot		
 		

f. *Blackbox testing login* kasir

Tabel 4. 10 *Blackbox testing login* kasir

Test faktor	Hasil	Keterangan
Memasukkan *username* dan *password* yang sesuai	✔	Sistem akan menampilkan halaman *login*
Screenshot		
		

g. *Blackbox testing* home kasir

Tabel 4. 11 *blackbox testing* home kasir

Test faktor	Hasil	Keterangan
Menampilkan jumlah pembelian tiket masuk & tiket wahana	✔	Sistem akan menampilkan halaman **home**
Screenshot		
		

h. *Blackbox testing scan* kasir

Tabel 4. 12 *Blackbox testing scan* kasir

Test faktor	Hasil	Keterangan
Menscan *qr code* yang tersedia	✔	Sistem akan menampilkan detail riwayat pembelian tiket
Screenshot		





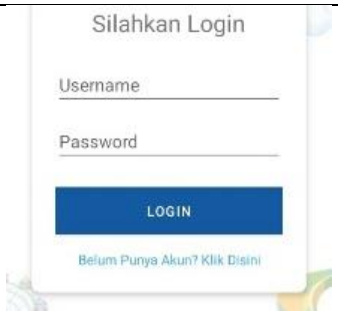
i. *Blackbox testing* laporan

Tabel 4. 13 *Blackbox testing* laporan

Test faktor	Hasil	Keterangan
Menampilkan laporan pembelian tiket masuk dan tiket wahana per hari ,per bulan,per tahun dan Memilih rentang waktu laporan serta dapat mendownload laporan	✔	Sistem akan menampilkan laporan pembelian per hari, per bulan serta per tahun dan akan menampilkan berdasarkan rentang waktu yang dipilih serta dapat mendownload laporan
Screenshot		
		

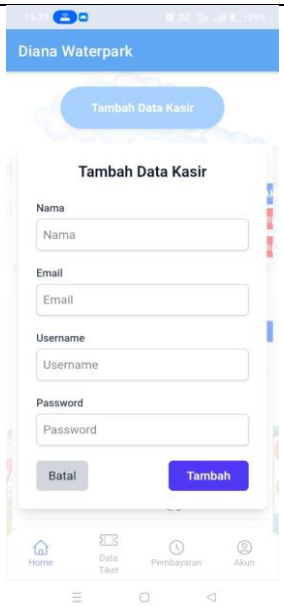
j. *Blackbox testing login* admin

Tabel 4. 14 *Blackbox testing login* admin

Test faktor	Hasil	Keterangan
Memasukkan username dan password yang sesuai	✔	Sistem akan menampilkan halaman login admin
Screenshot		
		

k. *Blackbox testing* tambah kasir

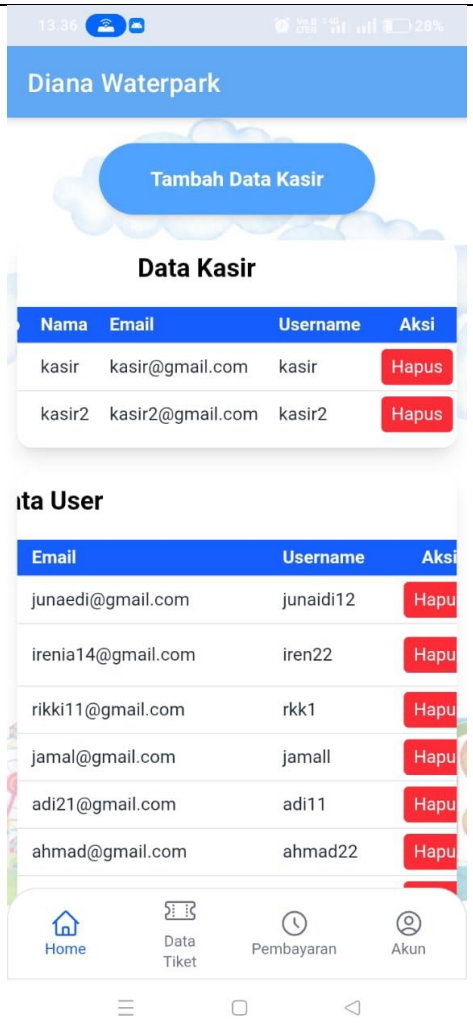
Tabel 4. 15 *Blackbox testing* tambah kasir

Test faktor	Hasil	Keterangan
Mengisi form yang sesuai	✔	Sistem akan menampilkan halaman home admin
Screenshot		
		

l. *Blackbox testing* hapus kasir

Tabel 4. 16 *Blackbox testing* hapus kasir

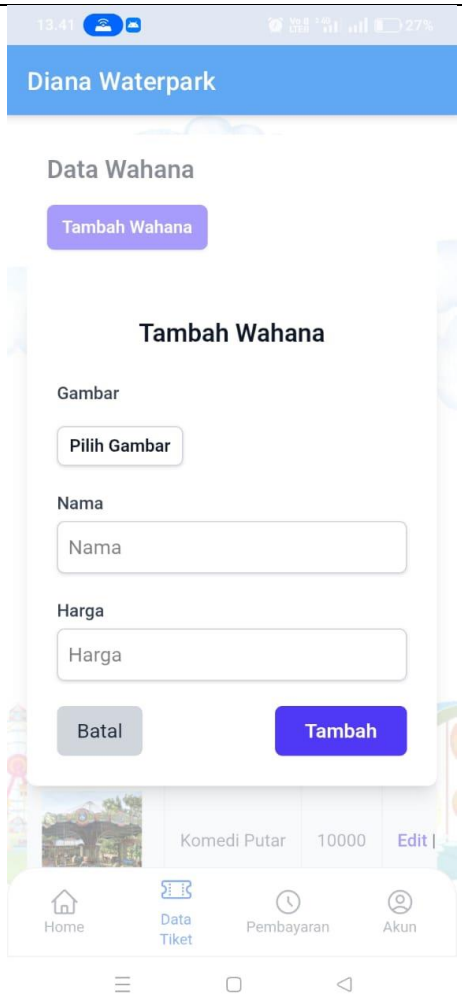
Test faktor	Hasil	Keterangan
Memilih kasir yang ingin dihapus	✔	Sistem akan menampilkan halaman *home* admin
Screenshot		



m. *Blackbox testing* tambah wahana

Tabel 4. 17 *Blackbox testing* tambah wahana

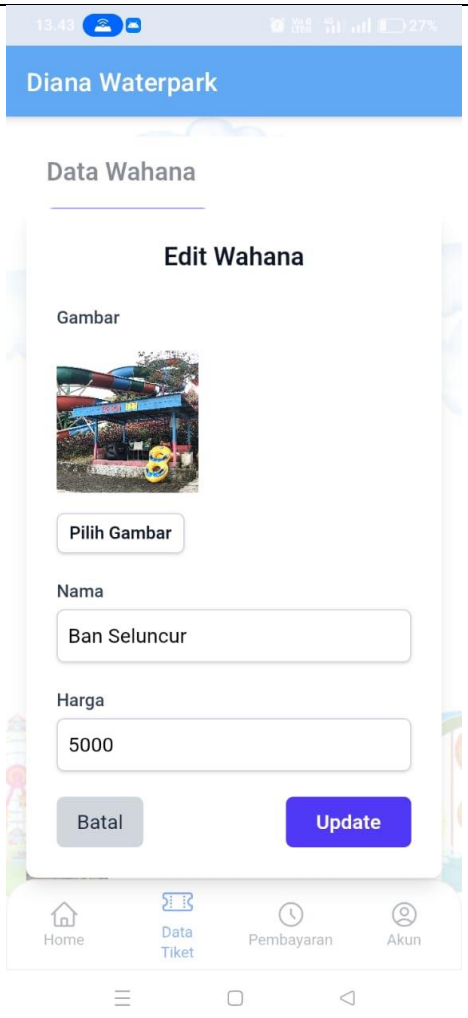
Test faktor	Hasil	Keterangan
Mengisi form yang tersedia	✔	Sistem akan menampilkan halaman admin data tiket
Screenshot		



n. *Blackbox testing* edit wahana

Tabel 4. 18 *Blackbox testing* edit wahana

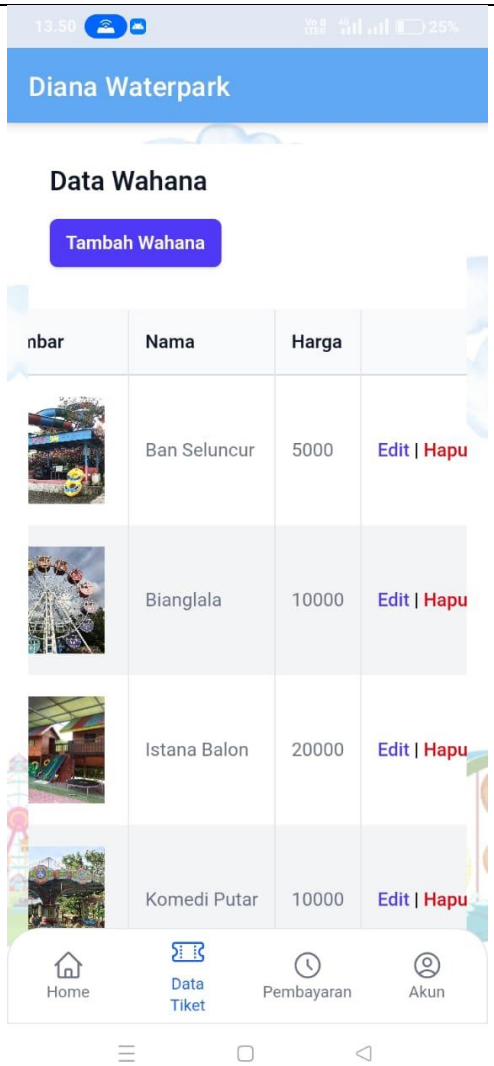
Test faktor	Hasil	Keterangan
Mengisi *form* yang tersedia	✔	Sistem akan menampilkan halaman admin data tiket
Screenshot		



o. *Blackbox testing* hapus wahana

Tabel 4. 19 *Blackbox testing* hapus wahana

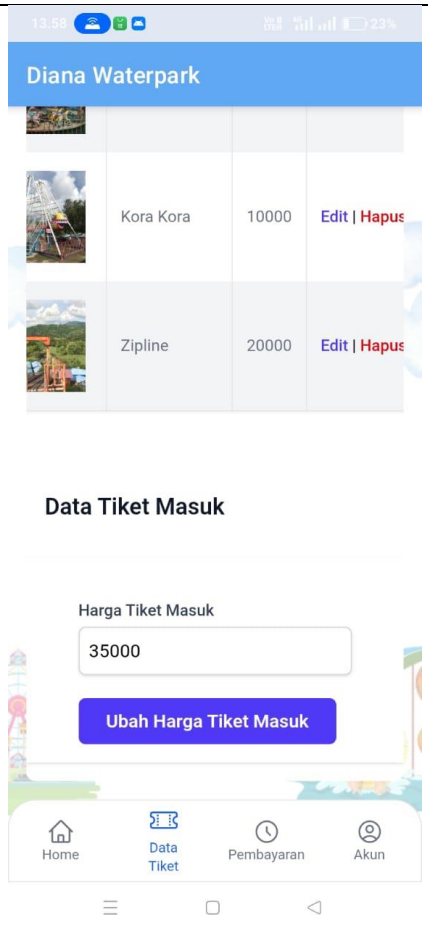
Test faktor	Hasil	Keterangan
Memilih wahana yang akan di hapus	✔	Sistem akan menampilkan halaman admin data tiket
Screenshot		



p. *Blackbox testing* ubah harga tiket masuk

Tabel 4. 20 *Blackbox testing* ubah harga tiket

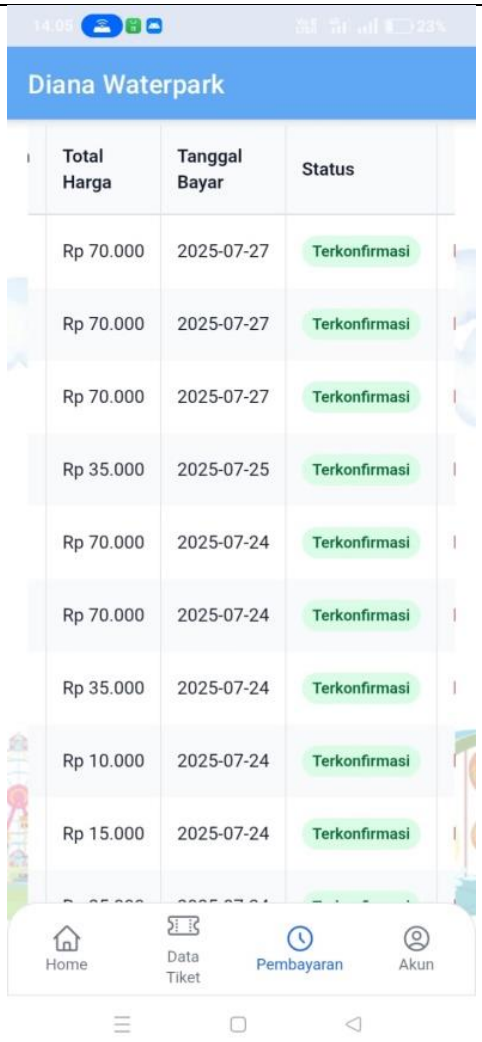
Test faktor	Hasil	Keterangan
Mengisi *form* yang di sediakan	✔	Sistem akan menampilkan halaman admin data tiket
Screenshot		



q. *Blackbox testing* pembayaran

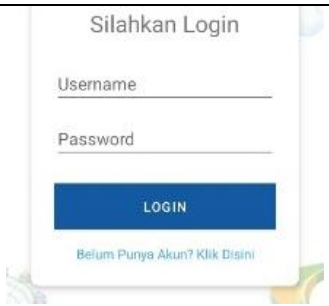
Tabel Tabel 4. 21 *Blackbox testing* pembayaran

Test faktor	Hasil	Keterangan
Mengonfirmasi data pembayaran yang tersedia	✔	Sistem akan menampilkan halaman admin pembayaran
Screenshot		



r. *Blackbox testing login* pimpinan

Tabel 4. 22 *blackbox testing login* pimpinan

Test faktor	Hasil	Keterangan
Memasukkan *username* dan *password* yang sesuai	✔	Sistem akan menampilkan halaman *login*
Screenshot		
		

s. Blackbox testing rating

Tabel 4. 23 *blackbox testing rating*

Test faktor	Hasil	Keterangan
Pimpinan memilih menu *rating*	✔	Sukses, sistem akan menampilkan halaman *rating*
Screenshot		
		

t. *Blackbox testing* laporan

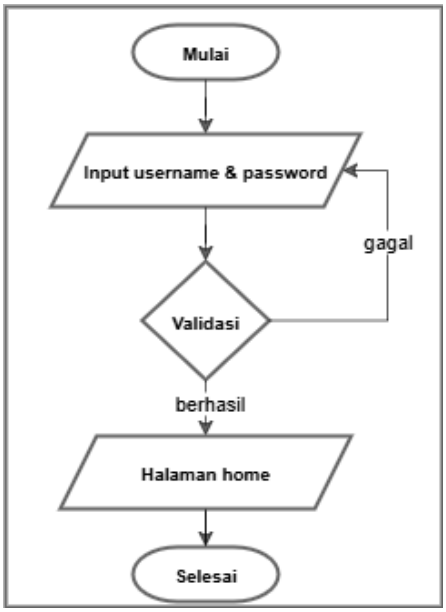
Tabel 4. 24 *blackbox testing* laporan

Test faktor	Hasil	Keterangan
Menampilkan laporan pembelian tiket masuk dan tiket wahana per hari ,per bulan,per tahun dan Memilih rentang waktu laporan serta dapat mendownload laporan	✔	Sistem akan menampilkan laporan pembelian per hari, per bulan serta per tahun dan akan menampilkan berdasarkan rentang waktu yang dipilih serta dapat mendownload laporan
Screenshot		
 		

2. *Whitebox*

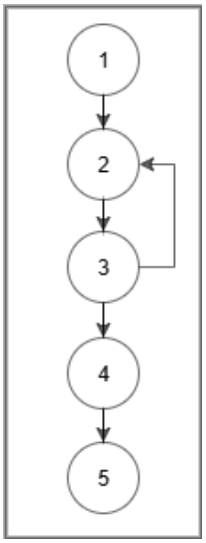
a. *Whitebox testing login user*

1) *Flowchart*



Gambar 4. 65 *Flowchart testing login user*

2) *Flowgraph*



Gambar 4. 66 *Flowgraph testing login user*

Berdasarkan gambar 4. 65 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus: V(G) = E – N + 2

E (*edge*) = 5

N (*node*) = 5

P (predikat *node*) = 1

Penyelesaian:

V(G) = E – N + 2

= 5 – 5 + 2

= 2

Predikat (P) = P + 1

= 1 + 1

= 2

(2) Berdasarkan perhitungan *cyclomatic complexcity* dari *flowgraph* diatas memiliki *region = 2*

(3) *Independent path* pada *flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 2 - 3

Path 2 = 1 – 2 – 3 – 4 – 5

(4) Grafik matriks *login* berhasil

Tabel 4. 25 Grafik matriks login berhasil

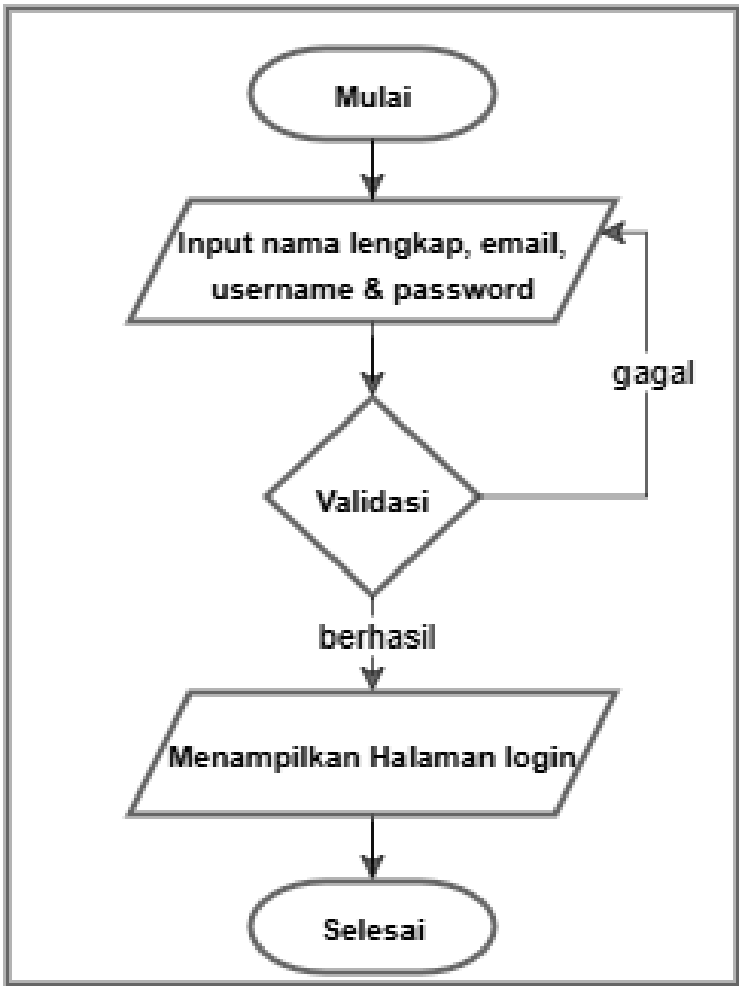
	1	**2**	**3**	**4**	**5**	**E-1**
1		1				1 - 1 = 0

Lanjutan Tabel 4. 26

2			1			1 – 1 = 0
3		1		1		2 – 1 = 1
4					1	1 – 1 = 0
5						0
	SUM (E + 1)					**1 + 1 = 2**

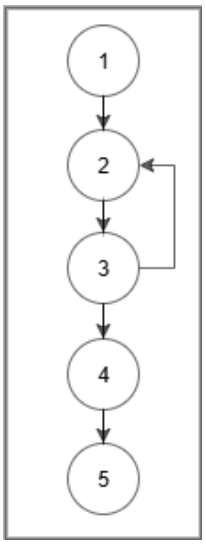
b. *Whitebox testing* registrasi *user*

1) Flowchart



Gambar 4. 67 *Flowchart testing* registrasi *user*

2) Flowgraph



Gambar 4. 68 *Flowgraph testing* registrasi *user*

Berdasarkan gambar 4. 67 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus: V(G) = E – N + 2

E (*edge*) = 5

N (*node*) = 5

P (predikat *node*) = 1

Penyelesaian :

V(G) = E – N + 2

= 5 – 5 + 2

= 2

Predikat (P) = P + 1

= 1 + 1

= 2

(2) Berdasarkan perhitungan *cyclomatic complexcity* dari *flowgraph* diatas memiliki *region = 2*

(3) *Independent path* pada *flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 2 - 3

Path 2 = 1 – 2 – 3 – 4 – 5

(4) Grafik matriks *registrasi* berhasil

Tabel 4. 26 Grafik matriks registrasi berhasil

	1	**2**	**3**	**4**	**5**	**E-1**
1		1				1 - 1 = 0
2			1			1 – 1 = 0
3		1		1		2 – 1 = 1
4					1	1 – 1 = 0
5						0
	SUM (E + 1)					**1 + 1 = 2**

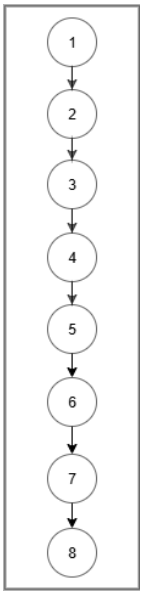
c. *Whitebox testing* tiket masuk *user*

1) Flowchart



Gambar 4. 69 *Flowchart testing* tiket masuk *user*

2) Flowgraph



Gambar 4. 70 *Flowgraph testing* tiket masuk *user*

Berdasarkan gambar 4. 69 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus : V(G) = E – N + 2

E (*edge*) = 7

N (*node*) = 8

P (predikat *node*) = 0

Penyelesaian :

V(G) = 7 – 8 + 2

= 7 – 8 + 2

= 1

Predikat (P) = P + 1

= 0 + 1

= 1

(2) Berdasarkan perhitungan *cyclomatic complexcity* dari *flowgraph* diatas memiliki *region = 1*

(3) *Independent path* pada *flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8

(4) Grafik matriks tiket masuk berhasil

Tabel 4. 27 Grafik matriks tiket masuk berhasil

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**E-1**
1		1							1-1= 0

Lanjutan Tabel 4. 28

2			1						1-1= 0
3				1					1-1= 0
4					1				1-1= 0
5						1			1-1= 0
6							1		1-1= 0
7								1	1-1= 0
8									0
	SUM (E + 1)								**1+1= 2**

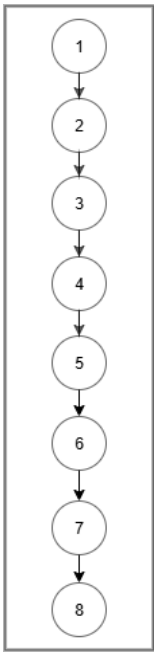
d. *Whitebox testing* wahana *user*

1) Flowchart



Gambar 4. 71 *Flowchart testing* tiket wahana *user*

2) Flowgraph



Gambar 4. 72 *Flowgraph testing* tiket wahana *user*

Berdasarkan gambar 4. 71 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus : V(G) = E – N + 2

E (*edge*) = 7

N (*node*) = 8

P (predikat *node*) = 0

Penyelesaian :

V(G) = 7 – 8 + 2

= 7 – 8 + 2

= 1

Predikat (P) = P + 1

= 0 + 1

= 1

(2) Berdasarkan perhitungan *cyclomatic complexcity* dari *flowgraph* diatas memiliki *region = 1*

(3) *Independent path* pada *flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8

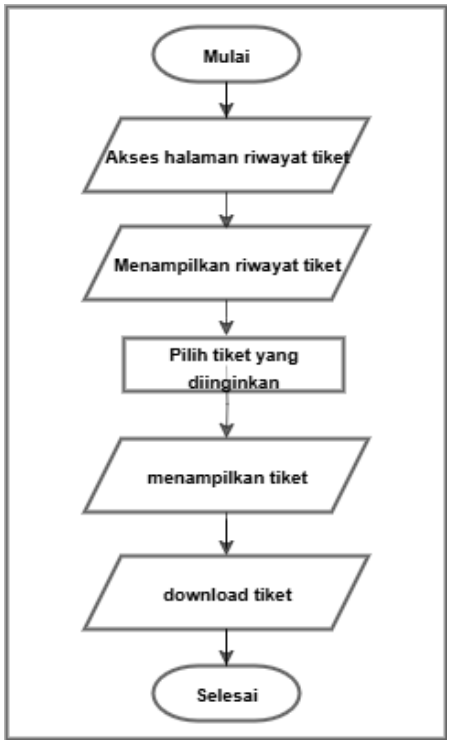
(4) Grafik matriks tiket wahana berhasil

Tabel 4. 28 Grafik matriks tiket wahana berhasil

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**E-1**
1		1							1-1= 0
2			1						1-1= 0
3				1					1-1= 0
4					1				1-1= 0
5						1			1-1= 0
6							1		1-1= 0
7								1	1-1= 0
8									0
	SUM (E + 1)								**1+1= 2**

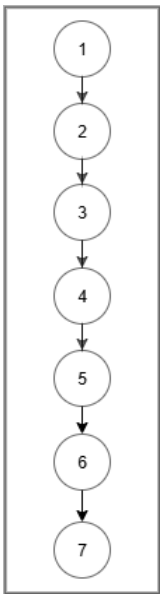
e. *Whitebox testing* riwayat tiket

1) Flowchart



Gambar 4. 73 *Flowchart testing* riwayat tiket *user*

2) Flowgraph



Gambar 4. 74 *Flowgraph testing* riwayat tiket *user*

Berdasarkan gambar 4. 73 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus : V(G) = E – N + 2

E (*edge*) = 6

N (*node*) = 7

P (predikat *node*) = 0

Penyelesaian :

V(G) = 6 – 7 + 2

= 6 – 7 + 2

= 1

Predikat (P) = P + 1

= 0 + 1

= 1

(2) Berdasarkan perhitungan *cyclomatic complexcity* dari *flowgraph* diatas memiliki *region = 1*

(3) *Independent path* pada *flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 4 – 5 – 6 – 7

(4) Grafik matriks riwayat tiket berhasil

Tabel 4. 29 Grafik matriks riwayat tiket berhasil

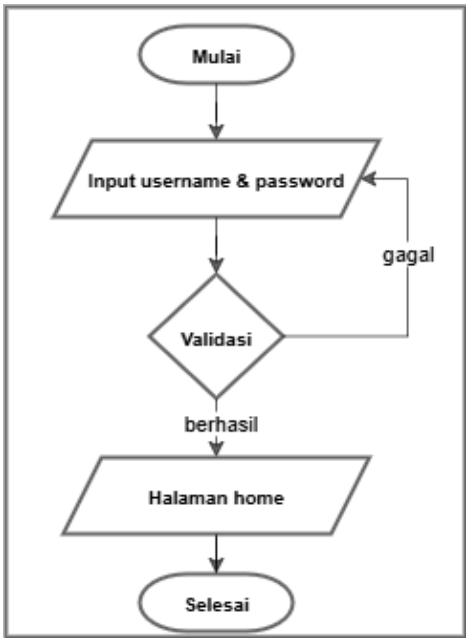
	1	**2**	**3**	**4**	**5**	**6**	**7**	**E-1**
1		1						1 - 1 = 0
2			1					1 – 1 = 0

Lanjutan Tabel 4.30

3				1				1 – 1 = 0
4					1			1 – 1 = 0
5						1		1 – 1 = 0
6							1	1 – 1 = 0
7								0
	SUM (E + 1)							**1 + 1 = 2**

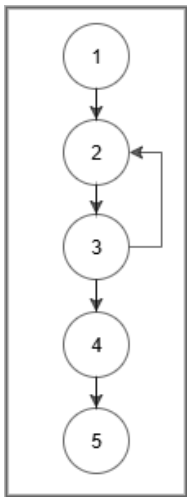
f. *Whitebox testing login* kasir

1) Flowchart



Gambar 4. 75 *Flowchart testing login* kasir

2) Flowgraph



Gambar 4. 76 *Flowgraph testing login* kasir

Berdasarkan gambar 4. 75 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus: V(G) = E – N + 2

E (*edge*) = 5

N (*node*) = 5

P (predikat *node*) = 1

Penyelesaian:

V(G) = E – N + 2

= 5 – 5 + 2

= 2

Predikat (P) = P + 1

= 1 + 1

= 2

(2) Berdasarkan perhitungan *cyclomatic complexcity* dari *flowgraph* diatas memiliki *region = 2*

(3) *Independent path* pada *flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 2 – 3

Path 2 = 1 – 2 – 3 – 4 – 5

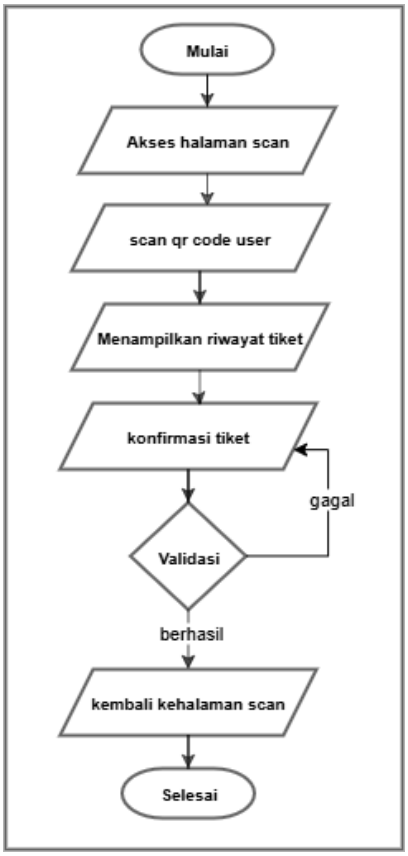
(4) Grafik matriks *login* berhasil

Tabel 4. 30 Grafik matriks *login* berhasil

	1	**2**	**3**	**4**	**5**	**E-1**
1		1				1 - 1 = 0
2			1			1 – 1 = 0
3		1		1		2 – 1 = 1
4					1	1 – 1 = 0
5						0
	SUM (E + 1)					**1 + 1 = 2**

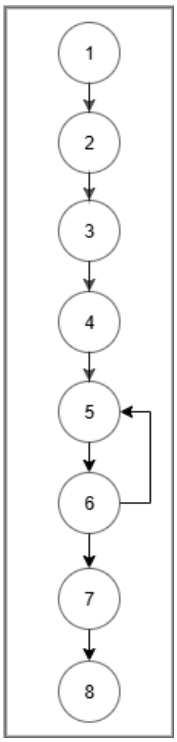
g. *Whitebox testing scan* kasir

1) Flowchart



Gambar 4. 77 *Flowchart testing scan* kasir

2) Flowgraph



Gambar 4. 78 *Flowgraph testing scan* kasir

Berdasarkan gambar 4. 77 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus: V(G) = E – N + 2

E (*edge*) = 8

N (*node*) = 8

P (predikat *node*) = 1

Penyelesaian:

V(G) = E – N + 2

= 8 – 8 + 2

= 2

Predikat (P) = P + 1

= 1 + 1

= 2

(2) Berdasarkan perhitungan *cyclomatic complexcity* dari *flowgraph* diatas memiliki *region = 2*

(3) *Independent path* pada *flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 4 – 5 – 6 – 5

Path 2 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8

(4) Grafik matriks *scan* kasir berhasil

Tabel 4. 31 Grafik matriks *scan* kasir berhasil

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**E-1**
1		1							1 - 1 = 0
2			1						1 – 1 = 0
3				1					1 – 1 = 0
4					1				1 – 1 = 0
5						1			1 – 1 = 0
6					1		1		2 – 1 = 0
7								1	1 – 1 = 0
8									0
	SUM (E + 1)								**1 + 1 = 2**

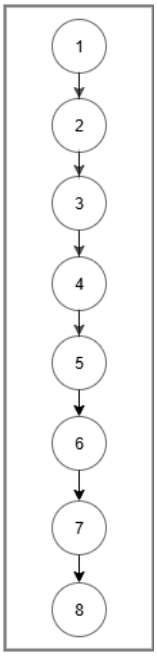
h. *Whitebox testing* laporan

1) Flowchart



Gambar 4. 79 *Flowchart* laporan

2) Flowgraph



Gambar 4. 80 *Flowgraph testing* laporan

Berdasarkan gambar 4. 79 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus: V(G) = E – N + 2

E (*edge*) = 7

N (*node*) = 8

P (predikat *node*) = 0

Penyelesaian:

V(G) = 7 – 8 + 2

= 7 – 8 + 2

= 1

Predikat (P) = P + 1

= 0 + 1

= 1

(2) Berdasarkan perhitungan *cyclomatic complexcity* dari *flowgraph* diatas memiliki *region = 1*

(3) *Independent path* pada *flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8

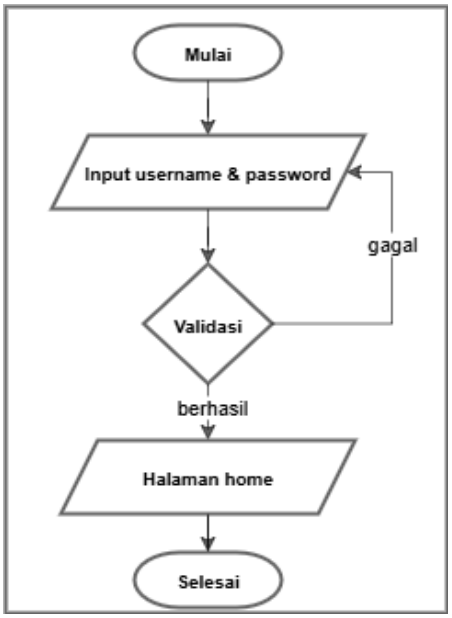
(4) Grafik matriks laporan

Tabel 4. 32 Grafik matriks laporan

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**E-1**
1		1							1-1= 0
2			1						1-1= 0
3				1					1-1= 0
4					1				1-1= 0
5						1			1-1= 0
6							1		1-1= 0
7								1	1-1= 0
8									0
	SUM (E + 1)								**1+1= 2**

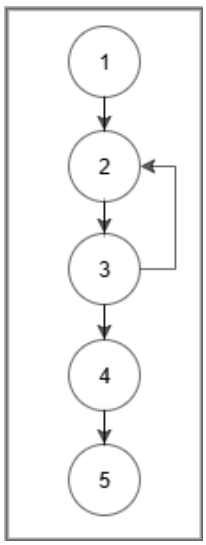
i. *Whitebox testing login admin*

1) Flowchart



Gambar 4. 81 *Flowchart testing login admin*

2) Flowgraph



Gambar 4. 82 *Flowgraph testing login admin*

Berdasarkan gambar 4. 69 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus: V(G) = E – N + 2

E (*edge*) = 5

N (*node*) = 5

P (predikat *node*) = 1

Penyelesaian:

V(G) = E – N + 2

= 5 – 5 + 2

= 2

Predikat (P) = P + 1

= 1 + 1

= 2

(2) Berdasarkan perhitungan *cyclomatic complexcity* dari *flowgraph* diatas memiliki *region = 2*

(3) *Independent path* pada *flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 2 – 3

Path 2 = 1 – 2 – 3 – 4 – 5

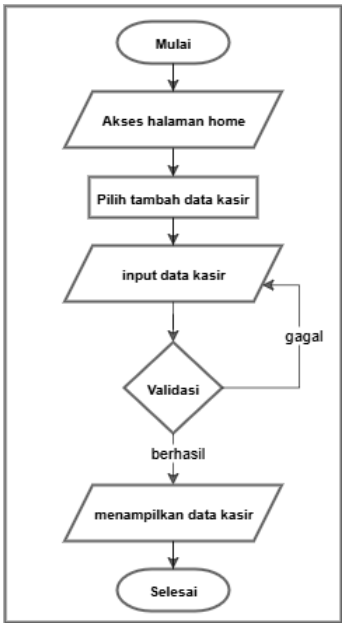
(4) Grafik matriks *login* berhasil

Tabel 4. 33 Grafik matriks *login* berhasil

	1	**2**	**3**	**4**	**5**	**E-1**
1		1				1 - 1 = 0
2			1			1 – 1 = 0
3		1		1		2 – 1 = 1
4					1	1 – 1 = 0
5						0
	SUM (E + 1)					**1 + 1 = 2**

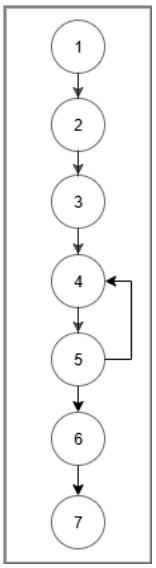
j. *Whitebox testing* tambah kasir

1) Flowchart



Gambar 4. 83 *Flowchart testing* tambah kasir

2) Flowgraph



Gambar 4. 84 *Flowgraph testing* tambah kasir

Berdasarkan gambar 4.83 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus: V(G) = E – N + 2

E (*edge*) = 7

N (*node*) = 7

P (predikat *node*) = 1

Penyelesaian:

V(G) = E – N + 2

= 7 – 7 + 2

= 2

Predikat (P) = P + 1

= 1 + 1

= 2

(2) Berdasarkan perhitungan *cyclomatic complexcity* dari *flowgraph* diatas memiliki *region = 2*

(3) *Independent path* pada *flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 4 – 5 – 4

Path 2 = 1 – 2 – 3 – 4 – 5 – 6 – 7

(4) Grafik matriks tambah kasir berhasil

Tabel 4. 34 Grafik matriks tambah kasir berhasil

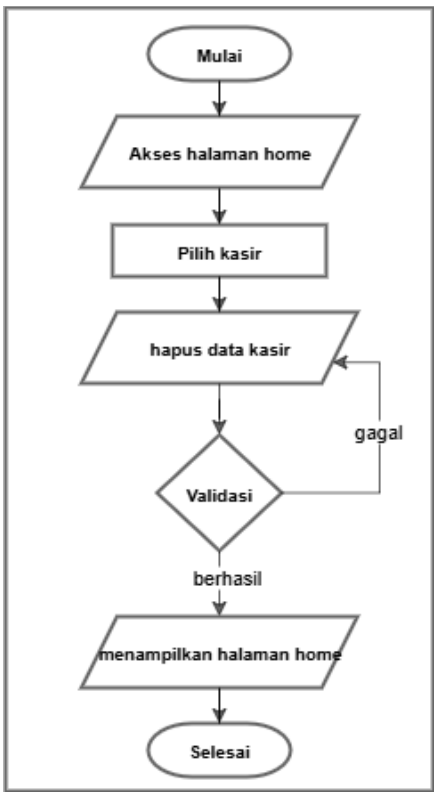
	1	**2**	**3**	**4**	**5**	**6**	**7**	**E-1**
1		1						1 - 1 = 0
2			1					1 – 1 = 0
3				1				1 – 1 = 0

Lanjutan Tabel 4.35

4					1			1 – 1 = 0
5				1		1		2 – 1 = 1
6							1	1 – 1 = 0
7								0
	SUM (E + 1)							**1 + 1 = 2**

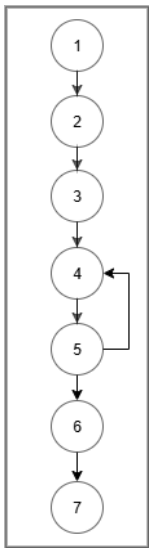
k. *Whitebox testing* hapus kasir

1) Flowchart



Gambar 4. 85 *Flowchart testing* hapus kasir

2) Flowgraph



Gambar 4. 86 *Flowgraph testing* hapus kasir

Berdasarkan gambar 4.85 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus: V(G) = E – N + 2

E (*edge*) = 7

N (*node*) = 7

P (predikat *node*) = 1

Penyelesaian:

V(G) = 7 – 7 + 2

= 7 – 7 + 2

= 2

Predikat (P) = P + 1

= 1 + 1

= 2

(2) Berdasarkan perhitungan *cyclomatic complexcity* dari *flowgraph* diatas memiliki *region = 2*

(3) *Independent path* pada *flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 4 – 5 – 4

Path 2 = 1 – 2 – 3 – 4 – 5 – 6 – 7

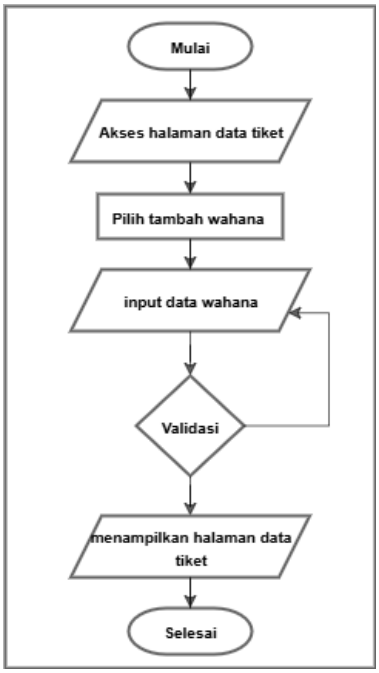
(4) Grafik matriks hapus kasir berhasil

Tabel 4. 35 Grafik matriks hapus kasir berhasil

	1	**2**	**3**	**4**	**5**	**6**	**7**	**E-1**
1		1						1 - 1 = 0
2			1					1 – 1 = 0
3				1				1 – 1 = 0
4					1			1 – 1 = 0
5				1		1		2 – 1 = 1
6							1	1 – 1 = 0
7								0
	SUM (E + 1)							**1 + 1 = 2**

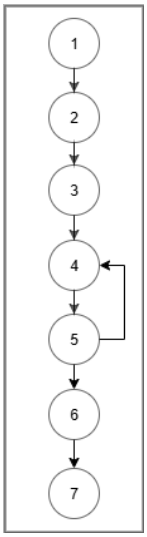
l. Whitebox testing tambah wahana

1) Flowchart



Gambar 4. 87 *Flowchart testing* tambah wahana

2) Flowgraph



Gambar 4. 88 *Flowgraph testing* tambah wahana

Berdasarkan gambar 4.87 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus: V(G) = E – N + 2

E (*edge*) = 7

N (*node*) = 7

P (predikat *node*) = 1

Penyelesaian:

V(G) = E – N + 2

= 7 – 7 + 2

= 2

Predikat (P) = P + 1

= 1 + 1

= 2

(2) Berdasarkan perhitungan *cyclomatic complexcity* dari *flowgraph* diatas memiliki *region = 2*

(3) *Independent path* pada *flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 4 – 5 – 4

Path 2 = 1 – 2 – 3 – 4 – 5 – 6 – 7

(4) Grafik matriks tambah wahana berhasil

Tabel 4. 36 Grafik matriks tambah wahana berhasil

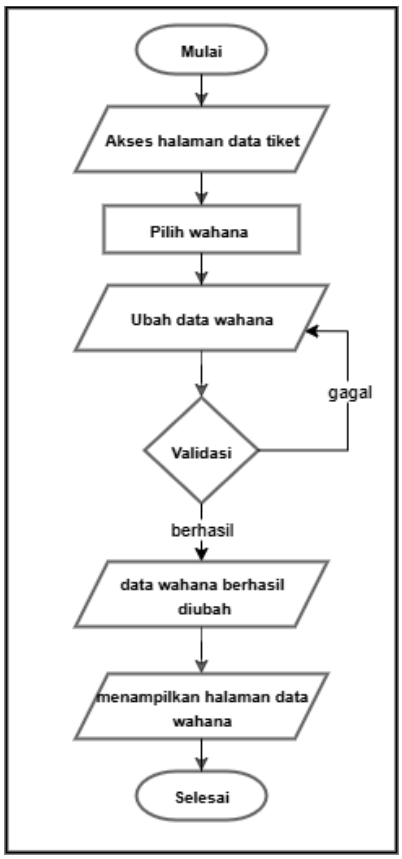
	1	**2**	**3**	**4**	**5**	**6**	**7**	**E-1**
1		1						1 - 1 = 0
2			1					1 – 1 = 0
3				1				1 – 1 = 0

Lanjutan Tabel 4.37

4					1			1 – 1 = 0
5				1		1		2 – 1 = 1
6							1	1 – 1 = 0
7								0
	SUM (E + 1)							**1 + 1 = 2**

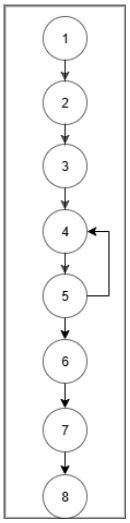
m. *Whitebox testing* edit wahana

1) Flowchart



Gambar 4. 89 *Flowchart testing* edit wahana

2) Flowgraph



Gambar 4. 90 *Flowgraph testing* edit wahana

Berdasarkan gambar 4.89 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus: V(G) = E – N + 2

E (*edge*) = 8

N (*node*) = 8

P (predikat *node*) = 1

Penyelesaian:

V(G) = E – N + 2

= 8 – 8 + 2

= 2

Predikat (P) = P + 1

= 1 + 1

= 2

(2) Berdasarkan perhitungan *cyclomatic complexcity* dari *flowgraph* diatas memiliki *region = 2*

(3) *Independent path* pada *flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 4 – 5 – 4

Path 2 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8

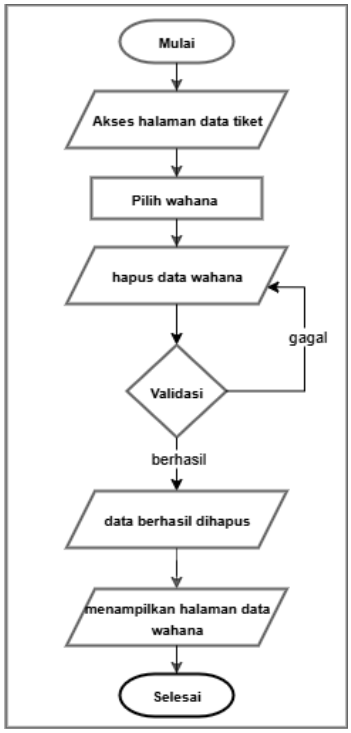
(4) Grafik matriks edit tiket wahana berhasil

Tabel 4. 37 Grafik matriks edit wahana berhasil

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**E-1**
1		1							1 - 1 = 0
2			1						1 – 1 = 0
3				1					1 – 1 = 0
4					1				1 – 1 = 0
5				1		1			2 – 1 = 1
6							1		1 – 1 = 0
7								1	
8									0
	SUM (E + 1)								**1 + 1 = 2**

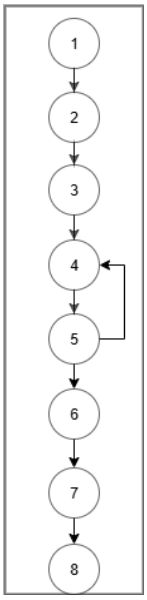
n. *Whitebox testing* hapus wahana

1) Flowchart



Gambar 4. 91 *Flowchart testing* hapus wahana

2) Flowgraph



Gambar 4. 92 *Flowgraph testing* hapus wahana

Berdasarkan gambar 4.91 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus : V(G) = E – N + 2

E (*edge*) = 8

N (*node*) = 8

P (predikat *node*) = 1

Penyelesaian :

V(G) = E – N + 2

= 8 – 8 + 2

= 2

Predikat (P) = P + 1

= 1 + 1

= 2

(2) Berdasarkan perhitungan *cyclomatic complexcity* dari *flowgraph* diatas memiliki *region = 2*

(3) *Independent path* pada *flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 4 – 5 – 4

Path 2 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8

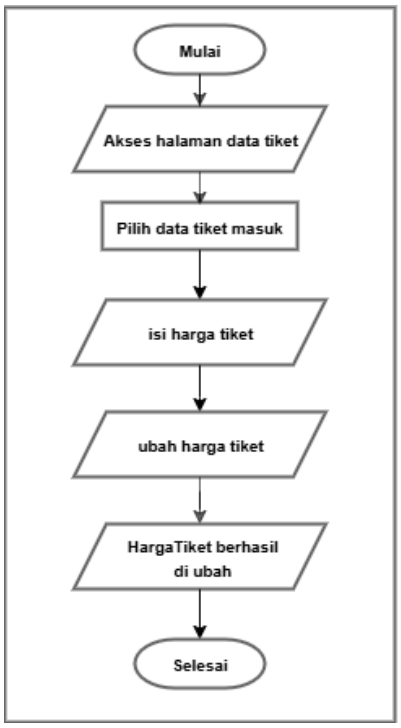
(4) Grafik matriks *hapus tiket* berhasil

Tabel 4. 38 Grafik matriks hapus wahana berhasil

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**E-1**
1		1							1 - 1 = 0
2			1						1 – 1 = 0
3				1					1 – 1 = 0
4					1				1 – 1 = 0
5				1		1			2 – 1 = 1
6							1		1 – 1 = 0
7								1	
8									0
	SUM (E + 1)								**1 + 1 = 2**

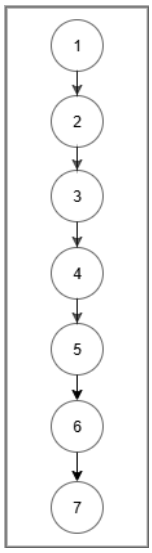
o. *Whitebox testing* ubah harga tiket masuk

1) Flowchart



Gambar 4. 93 *Flowchart testing* ubah harga tiket masuk

2) Flowgraph



gambar 4. 94 *Flowgraph testing* ubah harga tiket masuk

Berdasarkan gambar 4.93 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus: V(G) = E – N + 2

E (*edge*) = 6

N (*node*) = 7

P (predikat *node*) = 0

Penyelesaian:

V(G) = E – N + 2

= 6 – 7 + 2

= 1

Predikat (P) = P + 1

= 0 + 1

= 1

(2) Berdasarkan perhitungan *cyclomatic complexcity* dari *flowgraph* diatas memiliki *region = 1*

(3) *Independent path* pada *flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 4 – 5 – 6 – 7

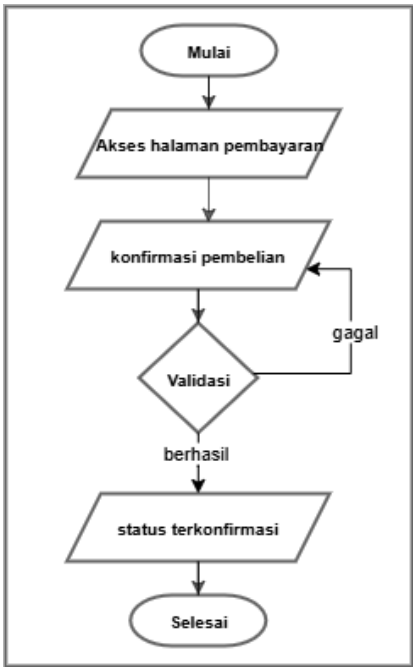
(4) Grafik matriks ubah harga tiket masuk berhasil

Tabel 4. 39 Grafik matriks ubah harga tiket masuk berhasil

	1	**2**	**3**	**4**	**5**	**6**	**7**	**E-1**
1		1						1-1= 0
2			1					1-1= 0
3				1				1-1= 0
4					1			1-1= 0
5						1		1-1= 0
6							1	1-1= 0
7								0
	SUM (E + 1)							**1+1= 2**

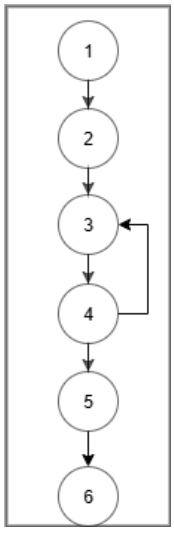
p. *Whitebox testing* pembayaran

1) Flowchart



Gambar 4. 95 *Flowchart Testing* Pembayaran

2) Flowgraph



Gambar 4. 96 *Flowgraph Testing* Pembayaran

Berdasarkan gambar 4.95 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus: V(G) = E – N + 2

E (*edge*) = 6

N (*node*) = 6

P (predikat *node*) = 1

Penyelesaian:

V(G) = E – N + 2

= 6 – 6 + 2

= 2

Predikat (P) = P + 1

= 1 + 1

= 2

(2) Berdasarkan perhitungan *cyclomatic complexcity* dari *flowgraph* diatas memiliki *region = 2*

(3) *Independent path* pada *flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 4 - 3

Path 2 = 1 – 2 – 3 – 4 – 5 – 6

(4) Grafik matriks konfirmasi pembayaran berhasil

Tabel 4. 40 Grafik matriks konfirmasi pembayaran berhasil

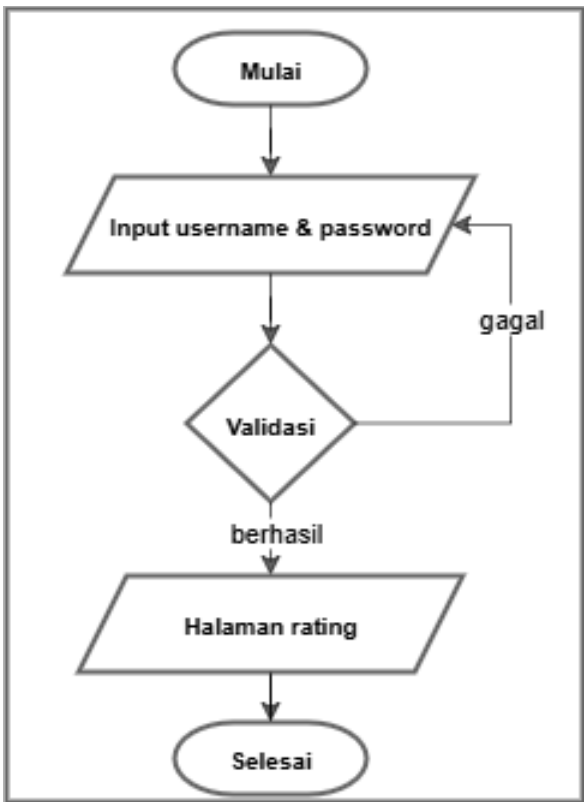
	1	**2**	**3**	**4**	**5**	**6**	**E-1**
1		1					1 - 1 = 0
2			1				1 – 1 = 0
3				1			1 – 1 = 0

Lanjutan Tabel 4.41

4			1		1		2 – 1 = 0
5						1	1 – 1 = 0
6							0
	SUM (E + 1)						**1 + 1 = 2**

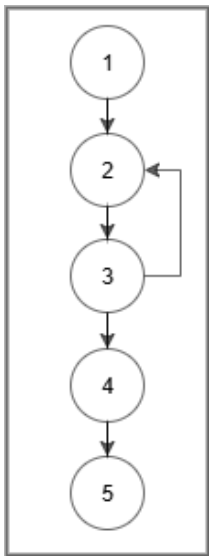
q. *Whitebox testing login* pimpinan

1) Flowchart



Gambar 4. 97 *Flowchart testing* login pimpinan

2) Flowgraph



Gambar 4. 98 *Flowgraph testing login* pimpinan

Berdasarkan gambar 4. 97 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus: V(G) = E – N + 2

E (*edge*) = 5

N (*node*) = 5

P (predikat *node*) = 1

Penyelesaian:

V(G) = E – N + 2

= 5 – 5 + 2

= 2

Predikat (P) = P + 1

= 1 + 1

= 2

(2) Berdasarkan perhitungan *cyclomatic complexcity* dari *flowgraph* diatas memiliki *region = 2*

(3) *Independent path* pada *flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 2

Path 2 = 1 – 2 – 3 – 4 – 5

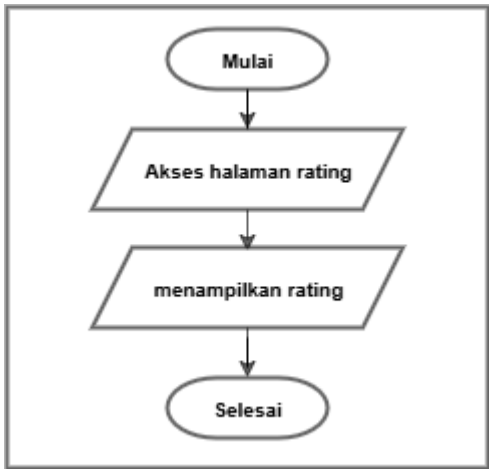
(4) Grafik matriks *login* berhasil

Tabel 4. 41 Grafik matriks *login* berhasil

	1	**2**	**3**	**4**	**5**	**E-1**
1		1				1 - 1 = 0
2			1			1 – 1 = 0
3		1		1		2 – 1 = 1
4					1	1 – 1 = 0
5						0
	SUM (E + 1)					**1 + 1 = 2**

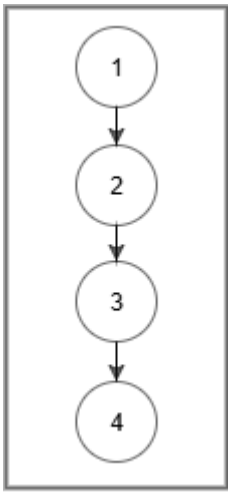
r. *Whitebox testing rating* pimpinan

1) Flowchart



Gambar 4. 99 *Flowchart testing rating* pimpinan

2) Flowgraph



Gambar 4. 100 *Flowgraph testing rating* pimpinan

Berdasarkan gambar 4. 99 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus: V(G) = E – N + 2

E (*edge*) = 3

N (*node*) = 4

P (predikat *node*) = 0

Penyelesaian:

V(G) = E – N + 2

= 3 – 4 + 2

= 1

Predikat (P) = P + 1

= 0 + 1

= 1

(2) Berdasarkan perhitungan *cyclomatic complexcity* dari *flowgraph* diatas memiliki *region = 1*

(3) *Independent path* pada *flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 4

(4) Grafik matriks *rating* pimpinan

Tabel 4. 42 Grafik matriks *rating* pimpinan

	1	**2**	**3**	**4**	**E-1**
1		1			1 - 1 = 0
2			1		1 – 1 = 0
3				1	1 – 1 = 1
4					0
	SUM (E + 1)				**1 + 1 = 2**

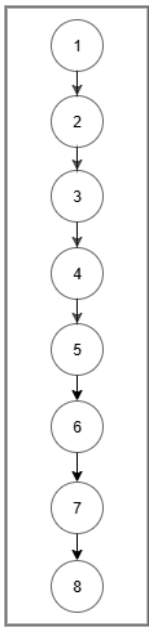
s. *Whitebox testing* laporan

1) Flowchart



Gambar 4. 101 *Flowchart* laporan

2) Flowgraph



Gambar 4. 102 *Flowgraph testing* laporan

Berdasarkan gambar 4. 101 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus: V(G) = E – N + 2

E (*edge*) = 7

N (*node*) = 8

P (predikat *node*) = 0

Penyelesaian:

V(G) = E – N + 2

= 7 – 8 + 2

= 1

Predikat (P) = P + 1

= 0 + 1

= 1

(2) Berdasarkan perhitungan *cyclomatic complexcity* dari *flowgraph* diatas memiliki *region = 1*

(3) *Independent path* pada *flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8

(4) Grafik matriks laporan

Tabel 4. 43 Grafik matriks laporan

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**E-1**
1		1							1-1= 0
2			1						1-1= 0
3				1					1-1= 0
4					1				1-1= 0
5						1			1-1= 0
6							1		1-1= 0
7								1	1-1= 0
8									0
	SUM (E + 1)								**1+1= 2**

BAB V

PENUTUP

A. Kesimpulan

Berdasarkan hasil penelitian dan pengujian terhadap aplikasi pemesanan tiket berbasis QR Code pada Wisata Diana *Waterpark* Barru, maka dapat disimpulkan:

1. Aplikasi berhasil dirancang dan dibangun menggunakan teknologi Android dengan bahasa pemrograman Java (*frontend)* dan PHP (*backend*), serta didukung oleh database MySQL dan sistem manajemen *PhpMyAdmin.*
2. Aplikasi mampu memfasilitasi pemesanan tiket masuk dan tiket wahana dengan menampilkan *QR Code* sebagai bukti pemesanan, yang dapat dipindai oleh petugas (kasir) menggunakan fitur *scan* dalam aplikasi.
3. Sistem pemesanan tiket ini meningkatkan efisiensi layanan dengan mengurangi antrean di loket, mempersingkat waktu transaksi, dan mempercepat proses verifikasi tiket.
4. Implementasi *QR Code* sebagai e-tiket efektif dalam mengelola data transaksi dan meningkatkan keamanan serta keakuratan validasi tiket.
5. Aplikasi mendukung empat peran pengguna utama: *user,* kasir, *admin,* dan pimpinan. Masing-masing peran memiliki fitur dan akses berbeda sesuai tanggung jawabnya, seperti pemesanan tiket oleh *user*, validasi oleh kasir, manajemen data oleh *admin*, dan analisis laporan oleh pimpinan.

6. Hasil pengujian sistem dengan metode *blackbox* dan *whitebox* menunjukkan bahwa seluruh fitur berjalan dengan baik dan sesuai fungsinya, mulai dari proses registrasi, login, pemesanan, pembayaran, pemindaian *QR Code,* hingga pelaporan transaksi.
7. Sistem laporan harian, bulanan, dan tahunan dapat diakses oleh kasir dan pimpinan, mendukung monitoring kinerja secara berkala dan berbasis data aktual yang tersimpan dalam sistem.

B. SARAN

Pada penelitian ini penulis menyadari bahwa masih ada beberapa kekurangan yang sangat di perlu perbaikan dan pengembangan di penelitian selanjutnya. Oleh karena itu, penulis memiliki beberapa saran untuk pengembangan selanjutnya, sebagai berikut:

1. Aplikasi ini dapat dikembangkan kembali dengan tambah fitur-fitur yang lebih lebih lengkap.
2. Pastikan antarmuka aplikasi mudah digunakan dan responsif di berbagai perangkat.
3. Pengembangan Fitur Tambahan disarankan agar aplikasi dikembangkan lebih lanjut dengan menambahkan fitur notifikasi otomatis, metode pembayaran digital (seperti e-wallet), dan integrasi dengan media sosial untuk memperluas jangkauan promosi dan kenyamanan pengguna.

DAFTAR PUSTAKA

Abdurahman, H., Riswaya, A. R., & Id, A. (2014). Aplikasi pinjaman pembayaran secara kredit pada bank yudha bhakti. *Jurnal Computech & Bisnis*, *8*(2), 61-69.

Affandi, E., & Syahputra, T. (2018). Pemodelan Uml Manajeman Sistem Inventory. *Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD*, *1*(2), 14-25.

Akbar, F., & Sembiring, N. S. B. (2023). Perancangan Aplikasi Pemesanan Tiket Masuk Pada Central Park Zoo Medan Berbasis Android Menggunakan QR Code. *Jurnal Rekayasa Sistem (JUREKSI)*, *1*(2), 461-476.

Bagus Tri, M. (2020). Perancangan Sistem Informasi Management Siswa Berprestasi Berbasis Android Pada Smk Pgri Rawalumbu. *Jurnal Sains & Teknologi Fakultas Teknik*, *10*(2), 30-39.

Clivan, T., Sugiarso, B. A., & Sinsuw, A. A. (2019). Aplikasi Website Perpustakaan Berbasis QR-Code. *Jurnal Teknik Informatika*, *14*(1), 1-8.

Faulina, S. T., & Lestari, N. (2021). Penerapan Metode Waterfall pada Aplikasi Pemesanan Soundsystem Dan Organ Tunggal Jefri. *JIK: Jurnal Informatika dan Komputer*, *12*(2), 65-79.

Firdaus, M. (2022). Perancangan aplikasi chat-room dengan prinsip threading melalui pemrograman dengan bahasa java. *TEKNOSAINS: Jurnal Sains, Teknologi Dan Informatika*, *9*(2), 121-135.

Fitriani, B., Angraini, T., & Putra, Y. H. G. (2018, July). Pemodelan Use Case Diagram Sistem Informasi Inventaris Laboratorium Teknik Mesin. In *Proceeding Seminar Nasional Sistem Informasi dan Teknologi Informasi* (Vol. 1, No. 1, pp. 626-631).

Hadiana, A. I. (2016). Pemanfaatan Teknologi QR Code Untuk Verifikasi Akta Notaris (PPAT). *MIND (Multimedia Artificial Intelligent Networking Database) Journal*, *1*(1), 41.

Kurnianda, T. P., & Purwanto, H. (2021). ANALISIS KEPUASAN PENUMPANG TERHADAP PEMESANAN TIKET MELALUI TIKET ONLINE DAN PEMESANAN TIKET SECARA OFFLINE DI BANDARA XYZ. *JSI (Jurnal sistem Informasi) Universitas Suryadarma*, *8*(1), 193-202.

Masuara, R., Sengkey, R., & Tulenan, V. (2015). Rancang Bangun e-Ticketing Bioskop Studio 21 Manado Berbasis Multiplatform. *Jurnal Teknik Elektro dan Komputer*, *4*(2), 41-55.

Nugroho, D. A. A., & Supriyono, H. (2019). Sistem informasi pendaftaran seminar dengan tiket berbasis qr code. *J. Emit*, *19*(1).

Nurhayati, A. N., Josi, A., & Hutagalung, N. A. (2017). Rancang bangun aplikasi penjualan dan pembelian barang pada koperasi kartika samara grawira prabumulih. *Jurnal Teknologi Dan Informasi*, *7*(2), 13-24.

Pradana, A. G. (2019, October). Rancang Bangun Game Edukasi "AMUDRA" Alat Musik Daerah Berbasis Android. In *Prosiding Seminar Nasional Teknologi Informasi dan Komunikasi (SENATIK)* (Vol. 2, No. 1, pp. 49-53).

Pratala, C. T., Asyer, E. M., Prayudi, I., & Saifudin, A. (2020). Pengujian White Box pada Aplikasi Cash Flow Berbasis Android Menggunakan Teknik Basis Path. *J. Inform. Univ. Pamulang*, *5*(2), 111.

Putra, A. P., Andriyanto, F., Karisman, K., & Harti, T. D. M. (2020). Pengujian aplikasi point of sale menggunakan blackbox testing. *Jurnal Bina Komputer*, *2*(1), 74-78.

Praniffa, A. C., Syahri, A., Sandes, F., Fariha, U., Giansyah, Q. A., & Hamzah, M. (2023). Pengujian Sistem Informasi Parkir Berbasis Web Pada UIN SUSKA RIAU Menggunakan White Box dan Black Box Testing. *Jurnal Testing dan Implementasi Sistem Informasi*, *1*(1), 1-16..

Ramadhan, R. F., & Mukhaiyar, R. (2020). Penggunaan Database Mysql dengan Interface PhpMyAdmin sebagai Pengontrolan Smarthome Berbasis Raspberry Pi. *JTEIN: Jurnal Teknik Elektro Indonesia*, *1*(2), 129-134.

Rahman, F., & Santoso, S. (2015). Aplikasi pemesanan undangan online. *Jurnal Sains dan Informatika*, *1*(2).

Rizqya, N. (2020). Perancangan Sistem Informasi Akuntansi Laporan Posisi Keuangan Pada UMKM Berbasis WEB (Studi Kasus UMKM Home Catering): Array. *Jurnal Ilmiah KOMPUTASI*, *19*(3), 383-390.

Safitri, F. (2019). *Pembangunan Sistem One Payment Untuk Pembayaran Tiket Wisata Di Belitung* (Doctoral dissertation, Universitas Komputer Indonesia).

Sahi, A. (2020). Aplikasi Test Potensi akademik seleksi saringan masuk LP3I berbasis web online menggunakan framework codeigniter. *Tematik*, *7*(1), 120-129.

Sari, F. P. (2018). Aplikasi sistem informasi pemesanan koki dan masakan rumahan berbasis android. *Jurnal SITECH: Sistem Informasi dan Teknologi*, *1*(2), 123-126.

Setiawan, M. F., Witama, M. N., & Hikmah, R. (2020). Perancangan Sistem Pengolahan Data Produksi Konveksi Berbasis Java Pada CV Nirwana Bunga Abadi. *Jurnal Nasional Komputasi dan Teknologi Informasi*, *3*(3), 202-208.

Setiyani, L. (2021, November). Desain Sistem: Use Case Diagram. In *Prosiding Seminar Nasional Inovasi dan Adopsi Teknologi (INOTEK)* (Vol. 1, No. 1, pp. 246-260).

Sugiana, D., & Muhtadi, D. (2019, November). Augmented Reality Type QR Code: Pengembangan Perangkat Pembelajaran di Era Revolusi Industri 4.0. In *Prosiding Seminar Nasional & Call For Papers*.

Sukabumi, A. G. A. B. (2016). Penggunaan QR code sarana penyampaian promosi dan informasi kebun binatang berbasis android. *Bianglala Informatika*, *4*(1).

Susanty, W., Astari, I. N., & Thamrin, T. (2019). Aplikasi Gis Menggunakan Metode Location Based Service (Lbs) Berbasis Android. *Explore: Jurnal Sistem informasi dan telematika*, *10*(1), 331261.

Suzana, A. M. (2021). Analisis Dan Perancangan Aplikasi Pemesanan Tiket Bus Berbasis Android. *Jurnal Informatika dan Rekayasa Perangkat Lunak*, *2*(3), 353-360.

Yunus, Y., & Sardiwan, M. (2018). Perancangan dan Pembuatan Media Pembelajaran Berbasis Android pada Mata Pelajaran Sistem Komputer. *Jurnal Pti (Pendidikan Dan Teknologi Informasi) Fakultas Keguruan Ilmu Pendidikan Universita Putra Indonesia" Yptk" Padang*, 31-41.

Zalukhu, A., Purba, S., & Darma, D. (2023). Perangkat Lunak Aplikasi Pembelajaran Flowchart. *Jurnal Teknologi, Informasi dan Industri*, *4*(1), 61-70.