BAB IV
HASIL DAN PEMBAHASAN

4.1 Tahapan Requirement Planning

Pada tahap ini merupakan tahap untuk melakukan proses pengambilan data oleh peneliti untuk melakukan perancangan sistem pencarian destinasi wisata dan UMKM menggunakan algoritma Dijkstra di kecamatan Kemiling.

4.1.1 Analisis Masalah

Berdasarkan analisis masalah pada bab sebelumnya, ditemukan bahwa potensi wisata dan UMKM di Kecamatan Kemiling terhambat oleh keterbatasan informasi yang terfragmentasi dan promosi yang tidak efektif. Untuk mengatasi hal tersebut, telah dirancang dan dibangun sebuah solusi digital berupa aplikasi Android terintegrasi. Aplikasi ini secara langsung menjawab permasalahan dengan menyediakan platform terpusat di mana wisatawan dapat dengan mudah menemukan informasi deskripsi, lokasi, dan harga dari berbagai destinasi dan produk UMKM. Dengan demikian, aplikasi ini menjembatani kesenjangan informasi antara wisatawan dan pelaku usaha, meningkatkan visibilitas potensi lokal, dan menyediakan sarana digital untuk perencanaan perjalanan yang lebih efisien.

4.1.2 Analisis Kebutuhan Sistem

Berdasarkan analisis masalah yang dipaparkan diatas, dapat disimpulkan bahwa sistem pencarian destinasi wisata dan umkm yang ingin di bangun antara lain :

a. Sisi client (Client-Side)

   Aplikasi dapat berjalan pada Smartphone Android dengan spesifikasi minimum RAM 2GB, memiliki fitur GPS, dan Memiliki Internet. Aplikasi akan dikembangkan menggunakan Android Studio dengan bahasa pemrograman Java, untuk fungsionalitas peta dan komunikasi data.

b. Sisi Server (Server-Side)

Sistem akan di hosting pada layanan web server online yang akan dipakai sebagai API (Application Programming Interface) yang menjadi jembatan antara aplikasi dan database yang di bangun menggunakan Bahasa pemrograman PHP dan sistem manajemen database MySQL.

4.2 Tahapan Perancangan Sistem

Pada tahap Dimana pembuat atau pengembang proyek dapat menjabarkan secara rinci apa saja yang akan di lakukan, dan mendesain struktur aplikasi. Mulai dari :

1. Pembuatan use case diagram

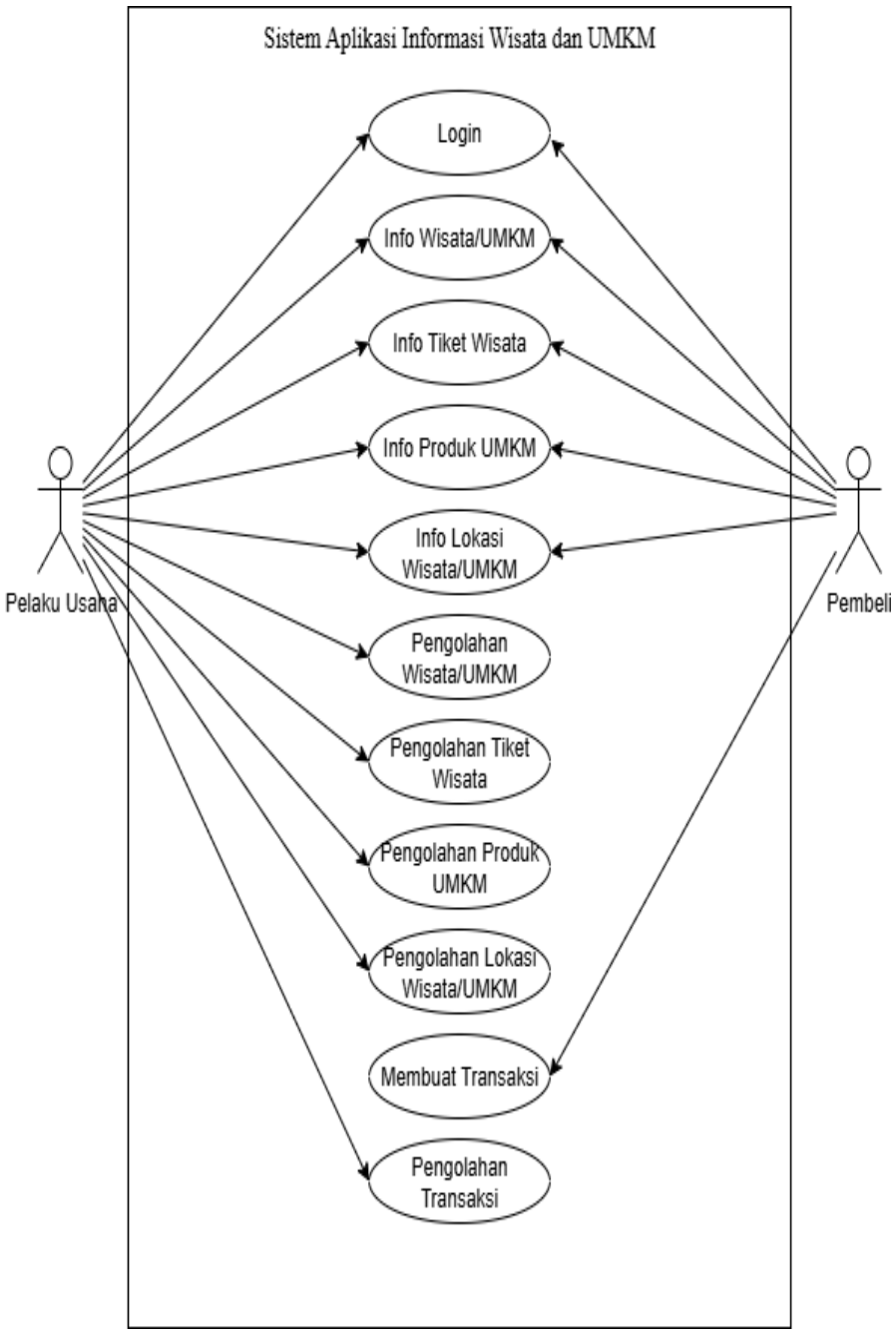
   Pada tahap ini pembuatan use case diagram meliputi :

   Perancangan Use case

Tabel 4.2 Use Case

Aktor	Deskripsi
Pengguna Umum (Wisatawan)	Aktor ini merupakan pengguna utama yang perannya sebagai konsumen. Pengguna dapat melakukan registrasi, login, mencari informasi wisata dan UMKM, melihat detail lokasi, serta mengelola profil pribadi nya.
Pelaku Usaha	Aktor ini adalah pengguna yang memiliki dan mengelola data destinasi wisata atau UMKM. Setelah melakukan login, Pelaku Usaha memiliki hak akses untuk menambah, mengubah, dan menghapus data yang terkait dengan usaha miliknya, seperti informasi usaha, produk, dan tiket.

Berdasarkan use case description diatas, dapat di lihat pada gambar 4.2 dibawah ini.



Gambar 4.1 Use Case

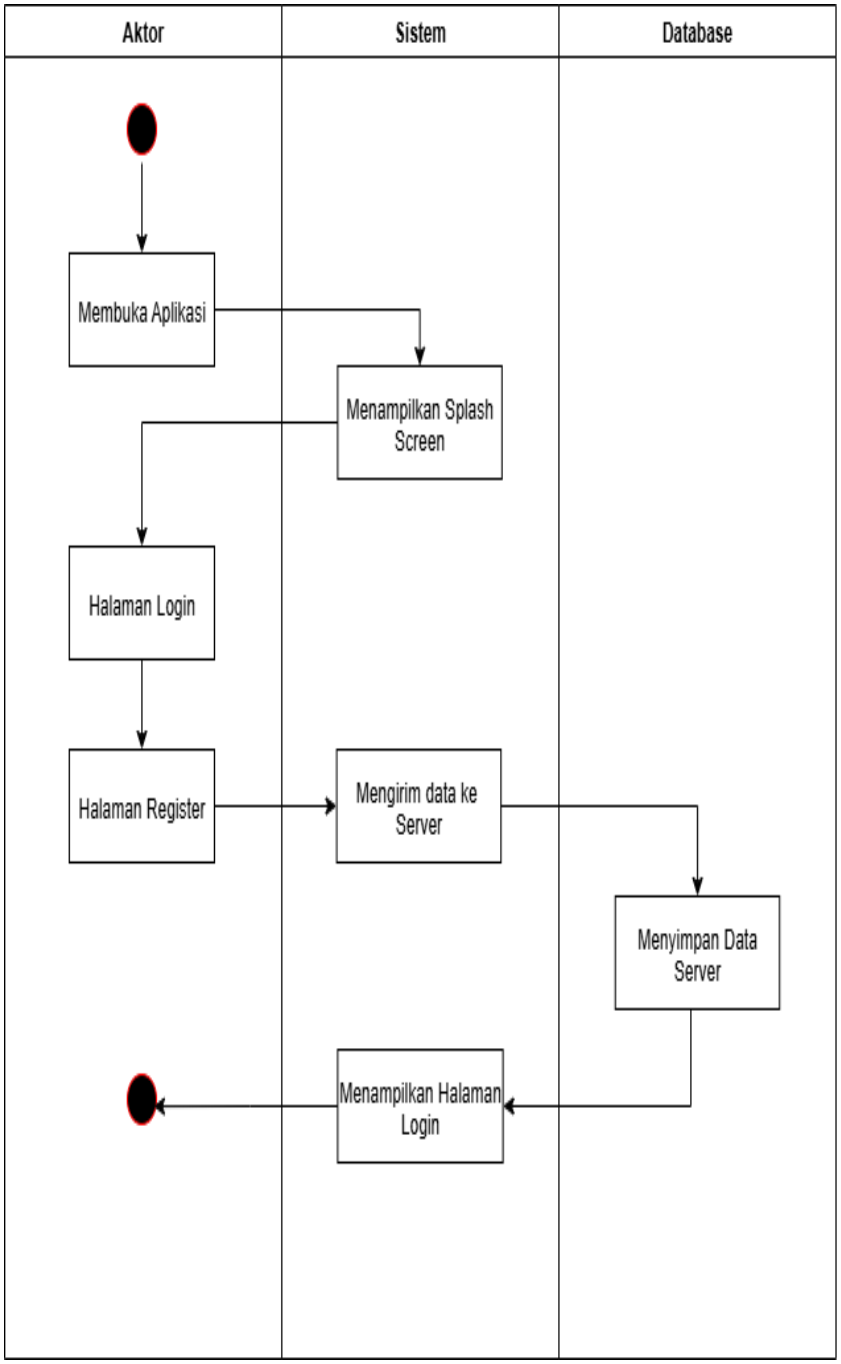
2. Pembuatan Activity diagram

   Pada tahap ini peneliti membuat alur proses aktivitas sistem dan merencanakan setiap alur Activity Diagram.

   1. Activity Diagram Registrasi

      Diagram aktivitas ini merangkum alur kerja pengguna, mulai dari membuka aplikasi dan melakukan pendaftaran akun. Sehingga

setelah melakukan pendaftaran akun, pengguna bisa langsung login dengan akun yang sudah di daftarkan.

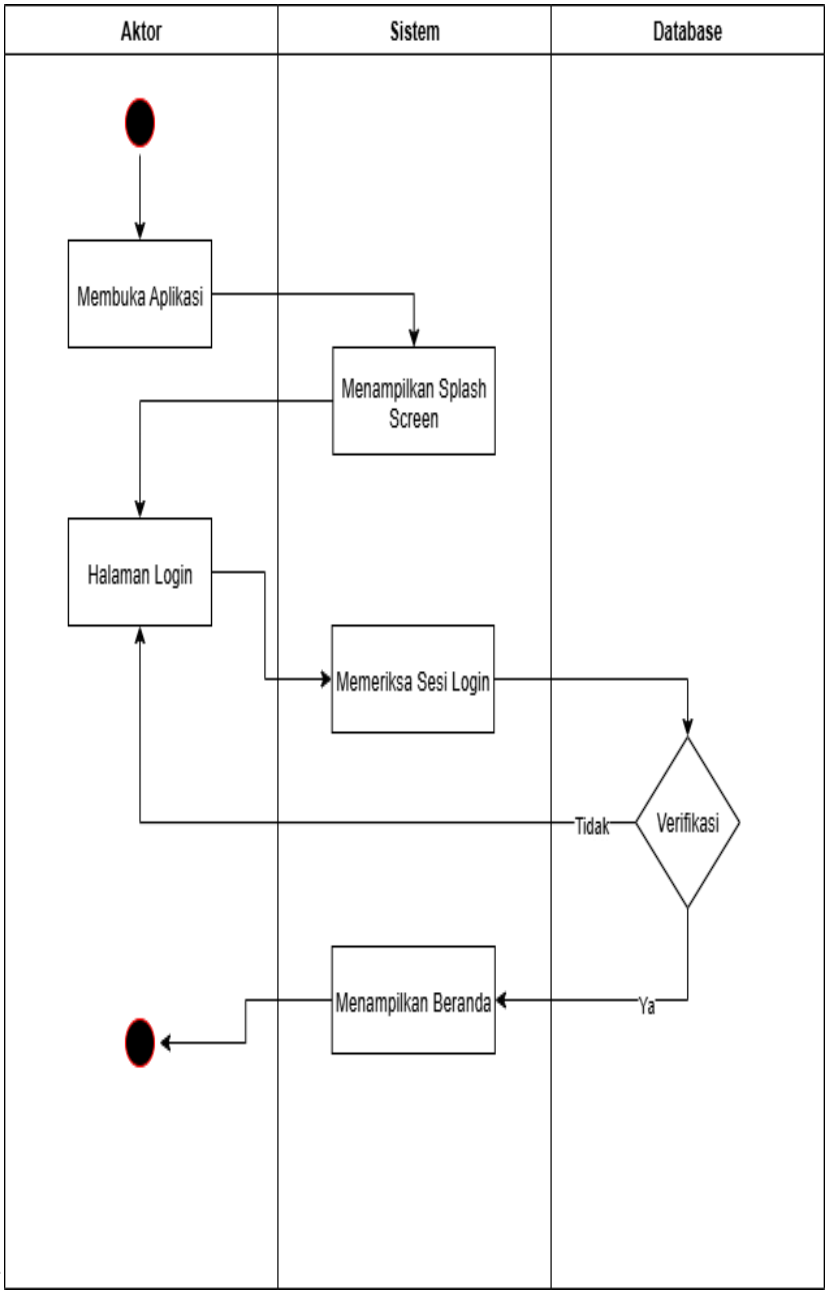


Gambar 4.2 Activity Diagram Registrasi

2. Activitiy Diagram Login

   Diagram aktivitas ini merangkum alur kerja pengguna, mulai dari membuka aplikasi dan melakukan login dengan akun yang sudah

terdaftar. Sehingga setelah melakukan login akun, pengguna bisa langsung membuka aplikasi dan menampilkan halaman beranda.

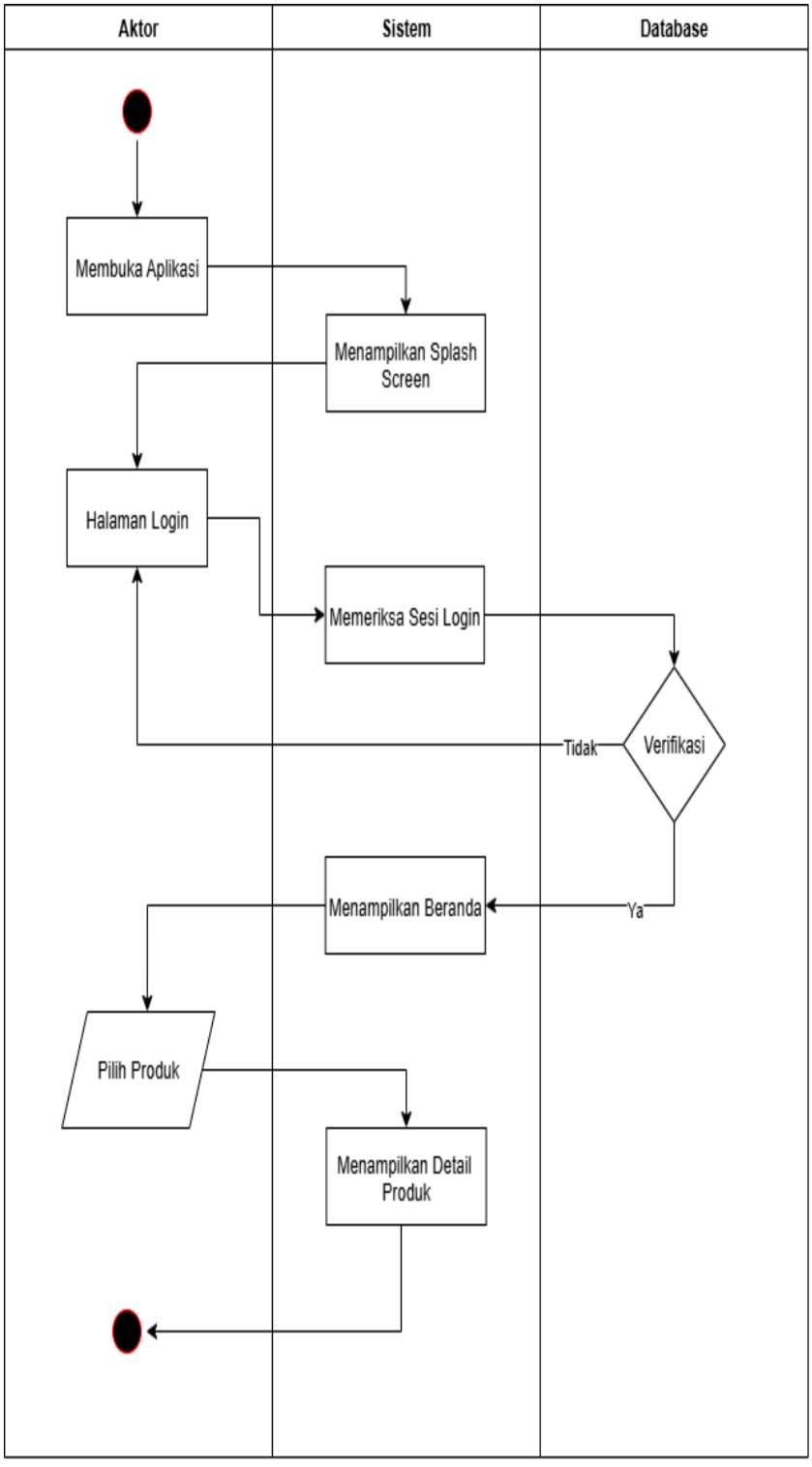


Gambar 4.3 Activity Diagram Login

3. Activity Detail Produk

Diagram aktivitas ini menggambarkan alur kerja pengguna untuk melihat detail dalam produk yang mereka pilih, yang akan

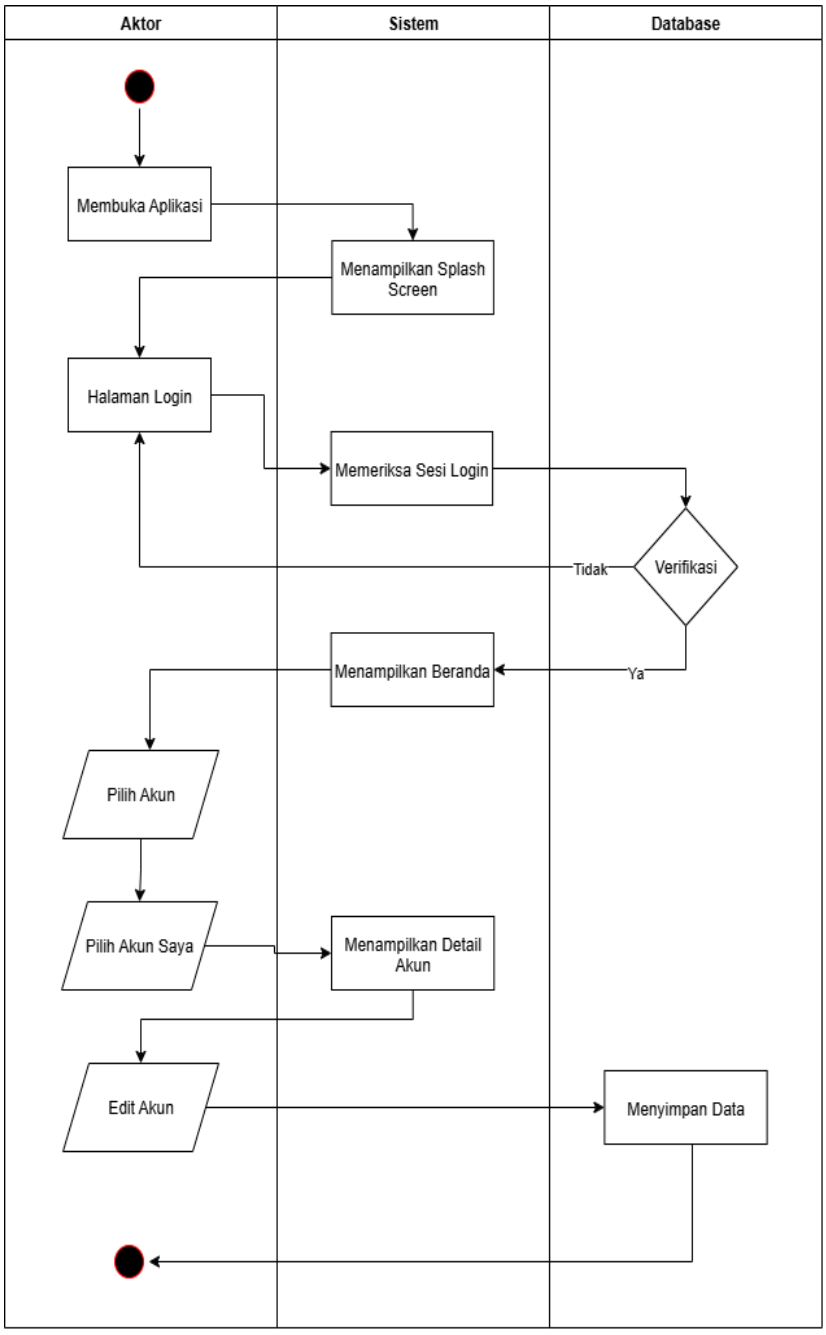
menampilkan deskripsi, gambar, dan rute yang akan di tempuh ke tempat lokasi produk tersebut.



Gambar 4.4 Activity Detail Produk

4. Activity Diagram Edit Profil

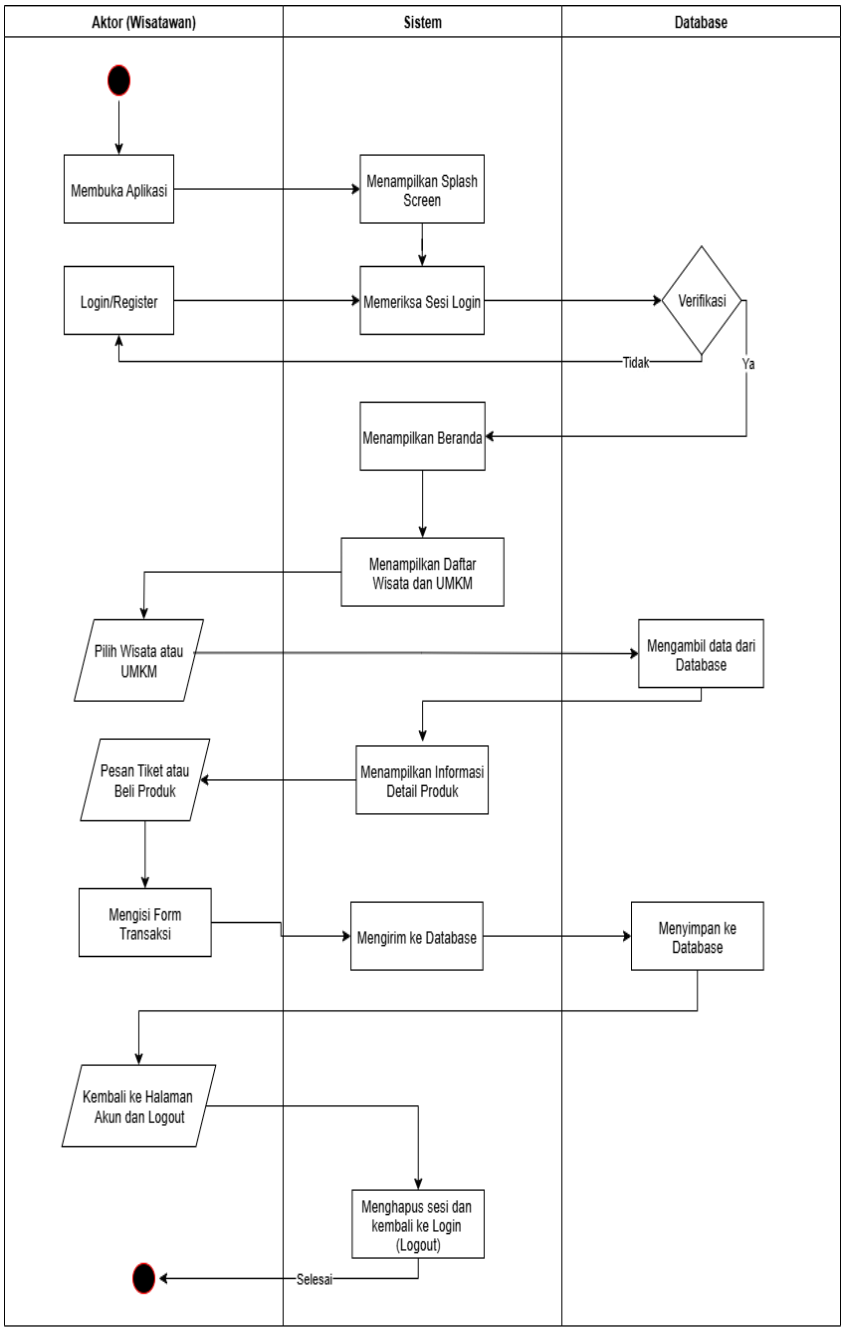
Diagram aktivitas ini menjelaskan alur kerja Pengguna, yang setelah login dapat mengelola profil miliknya di halaman "Akun saya". Fungsi utamanya adalah menyimpan data akun dengan mengisi form, di mana data tersebut akan diproses dan disimpan ke database.



Gambar 4.5 Activity Diagram Edit Profil

5. Activity Diagram Transaksi Wisatawan

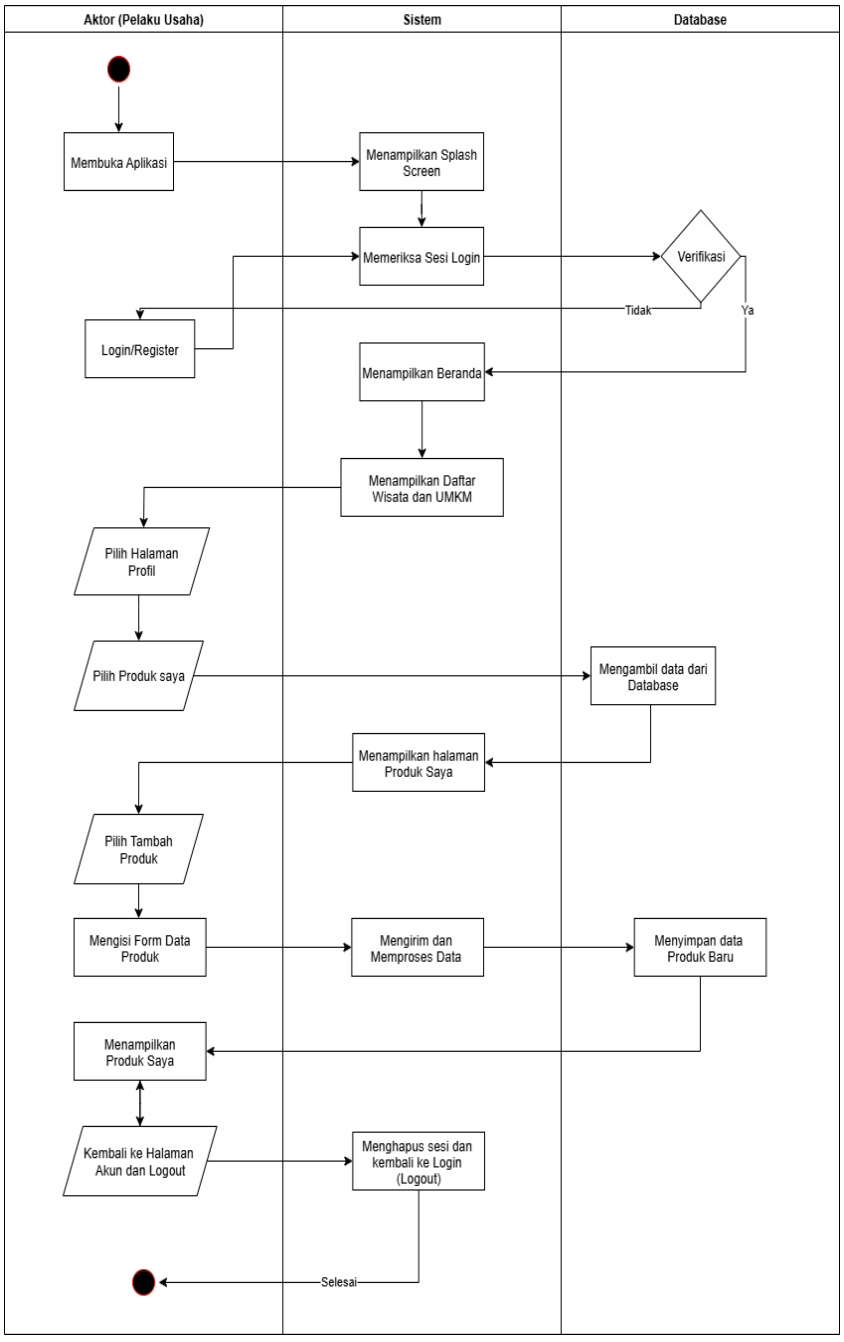
Diagram aktivitas ini menggambarkan alur kerja pengguna (wisatawan) ketika melakukan proses transaksi, seperti pemesanan tiket atau pembelian produk, melalui sistem aplikasi. Alur ini menunjukkan bagaimana sistem menangani dan merekam setiap transaksi yang dilakukan oleh pengguna.



Gambar 4.6 Activity Diagram Transaksi

6. Activity Diagram Tambah Produk

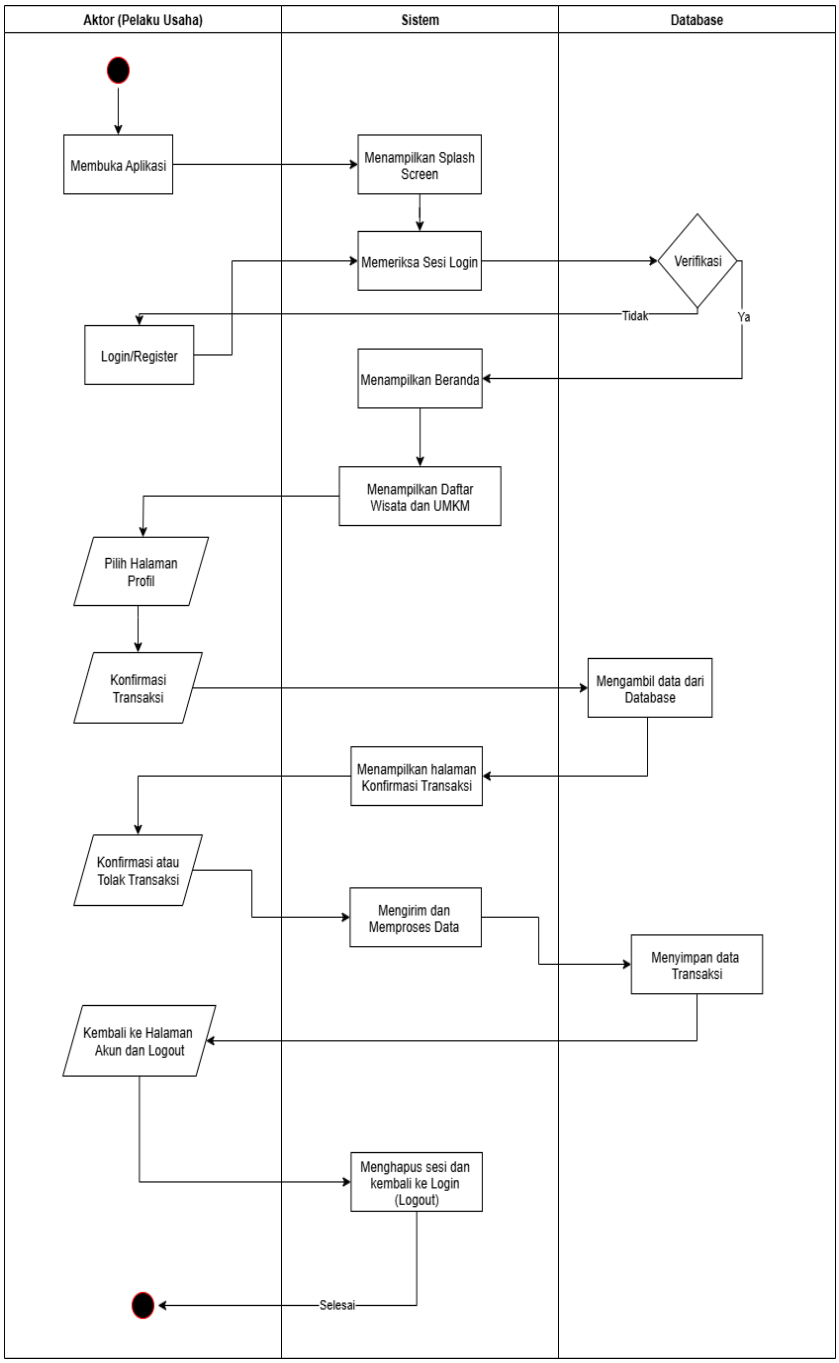
Diagram aktivitas ini menjelaskan alur kerja Pelaku Usaha, yang setelah login dapat mengelola konten miliknya di halaman "Produk Saya". Fungsi utamanya adalah menambah produk baru dengan mengisi form, di mana data tersebut akan diproses dan disimpan ke database.



Gambar 4.7 Activity Diagram Tambah Produk

7. Activity Diagram Konfirmasi Transaksi

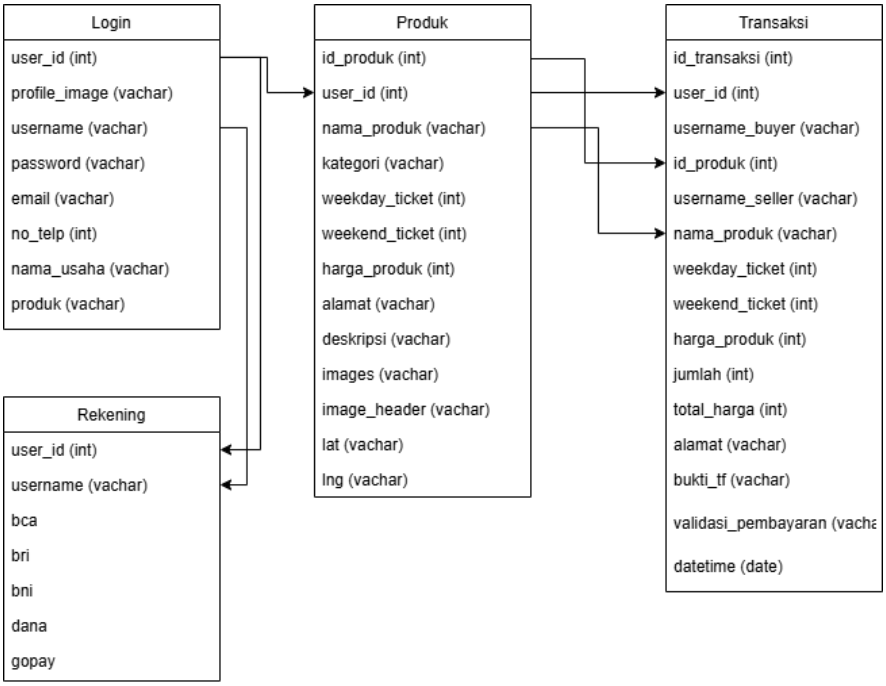
Diagram aktivitas ini menguraikan alur kerja bagi Pelaku Usaha ketika menerima dan mengonfirmasi pesanan atau tiket yang telah dibuat oleh wisatawan. Alur ini memastikan bahwa ada proses validasi dari sisi penyedia jasa sebelum sebuah transaksi dianggap selesai, dan memberikan kepastian bagi wisatawan.



Gambar 4.8 Konfirmasi Transaksi

3. Pembuatan Class Diagram

   Class diagram ini menggambarkan struktur dasar dari sistem aplikasi dengan tiga komponen. Dengan class diagram ini menunjukkan arsitektur sistem yang modular, Dimana setiap class memiliki tanggung jawab yang spesifik.



Gambar 4.9 Class Diagram

4.3 Tahapan Pembangunan dan Prototipe

Pada tahap ini, rancangan yang sudah di buat dapat di wujudkan dalam bentuk Aplikasi. Pada pembuatan aplikasi platform sistem terintegrasi berbasis mobile ini menggunakan android studio. Tahapan ini melakukan pengembangan prototipe awal berdasarkan desain yang sudah dibuat. Berikut ini adalah adalah desain awal dari aplikasi yang akan dibuat.

4.3.1 Pembangunan Desain Awal Aplikasi

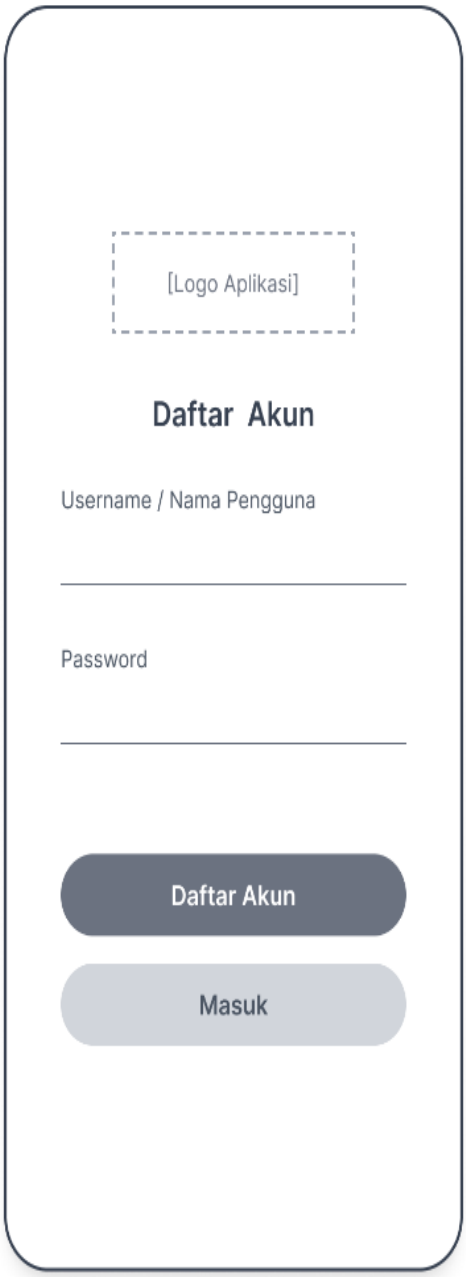
a. Tampilan Login

   Tampilan ini adalah tampilan awal dari ketika membuka aplikasi, untuk masuk kebagian berikutnya, terdapat masukan data seperti username dan password, dan terdapat tombol untuk masuk dan melakukan proses login.

Gambar 4.10 Tampilan Login

b. Tampilan Registrasi

Tampilan ini merupakan bagian untuk dapat menambahkan akun user login konsumen, terdapat masukan data seperti usename dan password serta terdapat tombol daftar untuk menambahkan akun ke database yang dapat di lihat pada gambar dibawah.



Gambar 4.11 Tampilan Daftar

c. Tampilan Beranda

Tampilan ini merupakan bagian antarmuka sistem yang menampilkan konten produk yang diambil dari database. Produk ini ditambahkan pengguna dari pelaku usaha dengan informasi deskripsi dan lokasi yang lengkap. Konten produk ini meliputi Wisata dan UMKM dan dilengkapi dengan navigasi utama yang terdiri dari “Home, Terdekat,

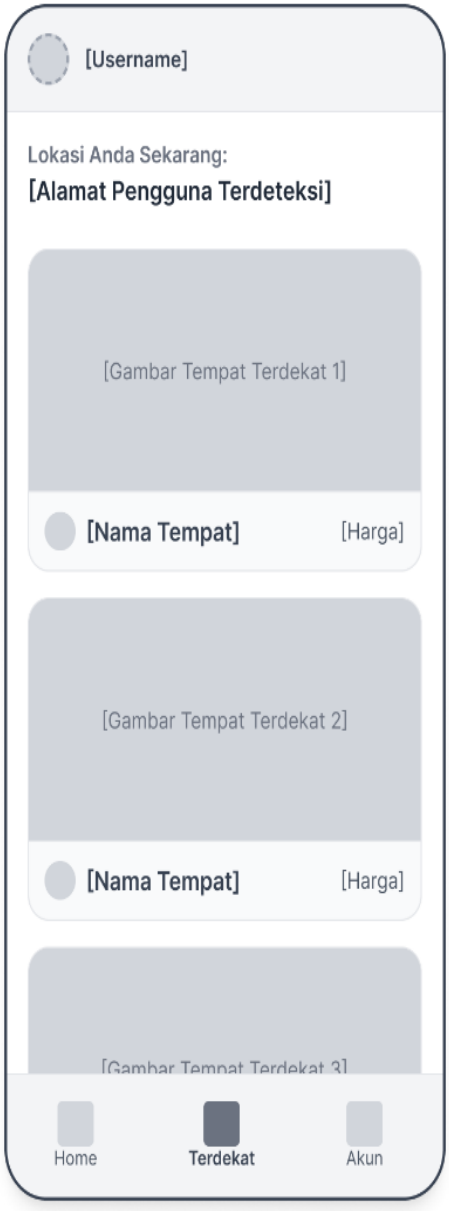
dan Akun", untuk memfasilitasi perpindahan cepat antar fitur utama aplikasi.

Gambar 4.12 Tampilan Beranda

d. Tampilan Terdekat

Tampilan ini merupakan rancangan antarmuka yang dirancang untuk memberikan informasi Lokasi wisata dan UMKM berdasarkan

kedekatannya dengan posisi pengguna. Dengan tata letak ini memudahkan pengguna untuk memilih produk atau destinasi wisata terdekat yang ada di sekitar mereka, bisa di lihat pada gambar dibawah.



Gambar 4.13 Tampilan Terdekat

e. Halaman Akun

Halaman ini merupakan tampilan untuk mengelola profil, produk, dan Logout dari sesi. Dengan tata letak minimalis memastikan pengguna

yang dapat dengan mudah mengakses fungsi-fungsi penting terkait akun mereka, dan bilah navigasi bawa dengan ikon.

Gambar 4.14 Tampilan Akun

f. Tampilan Akun Saya

Tampilan ini merupakan antarmuka yang menampilkan dan mengelola informasi pribadi mereka, yang terdiri dari email, username, nomor telpon, dan nama usaha mereka, dan terdapat tombol simpan yang

berfungsi untuk menyimpan setiap perubahan yang mungkin dilakukan pengguna.

Gambar 4.15 Tampilan Akun Saya

g. Tampilan Produk Saya

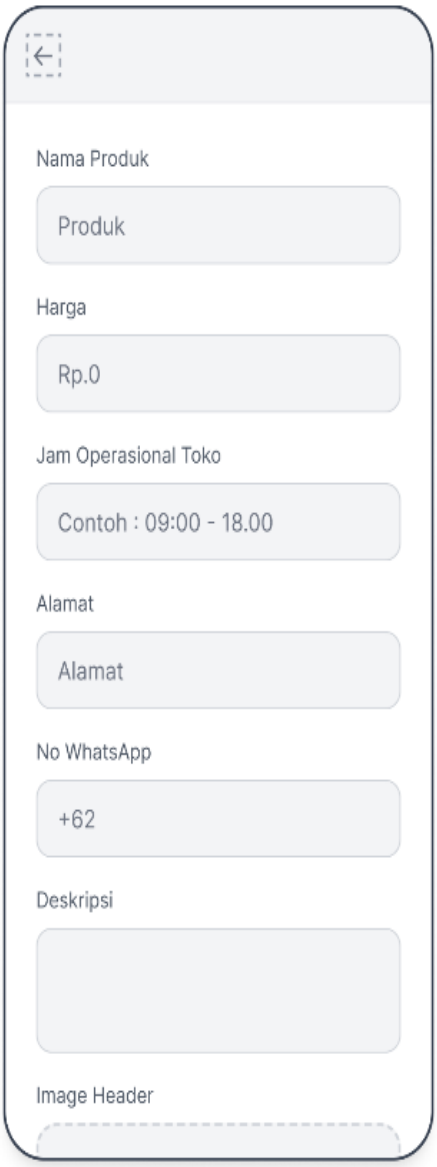
Pada tampilan ini merupakan rancangan khusus untuk pelaku usaha dalam mengelola produk mereka. Dengan dua tombol sebagai penambah konten produk yang akan di upload, dan juga menampilkan beberapa daftar produk atau wisata yang diambil dari dari database

berdasarkan yang pelaku usaha upload. Serta pengguna dapat menghapus produk.

Gambar 4.16 Tampilan Produk Saya

h. Tampilan Tambah Produk

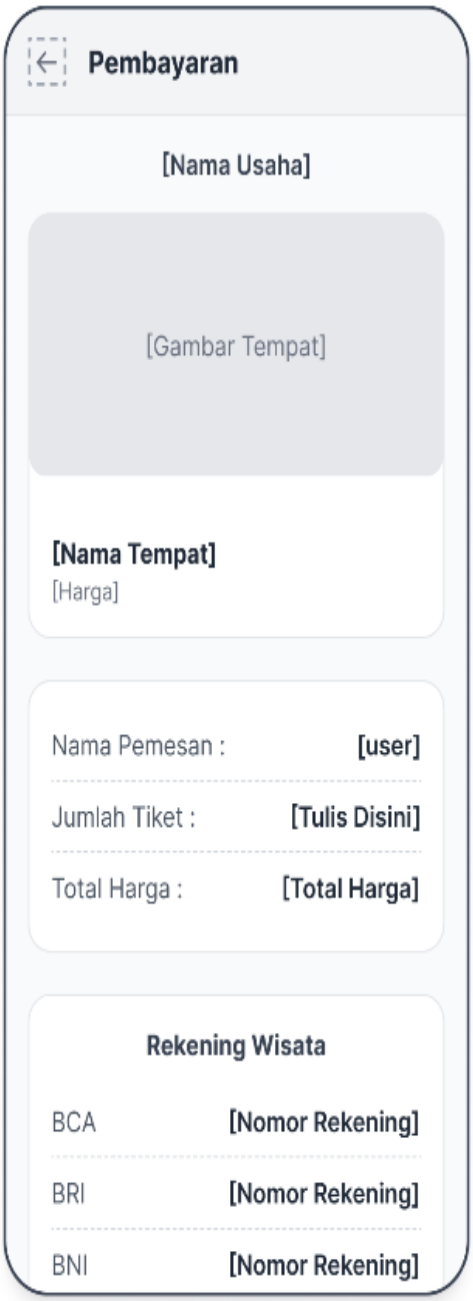
Tampilan ini merupakan sebuah formulir vertikal untuk memasukkan semua detail data yang di perlukan. Dengan desain ini memastikan semua informasi penting dapat di input oleh pelaku usaha secara sistematis dengan satu alur.



Gambar 4.17 Tampilan Tambah Produk

i. Tampilan Pembayaran

rancangan antarmuka untuk Halaman Pembayaran. Halaman ini dirancang untuk memandu pengguna menyelesaikan proses transaksi setelah memesan tiket atau produk. Dengan desain ini memastikan semua informasi penting dapat di input oleh pengguna secara sistematis dengan satu alur.

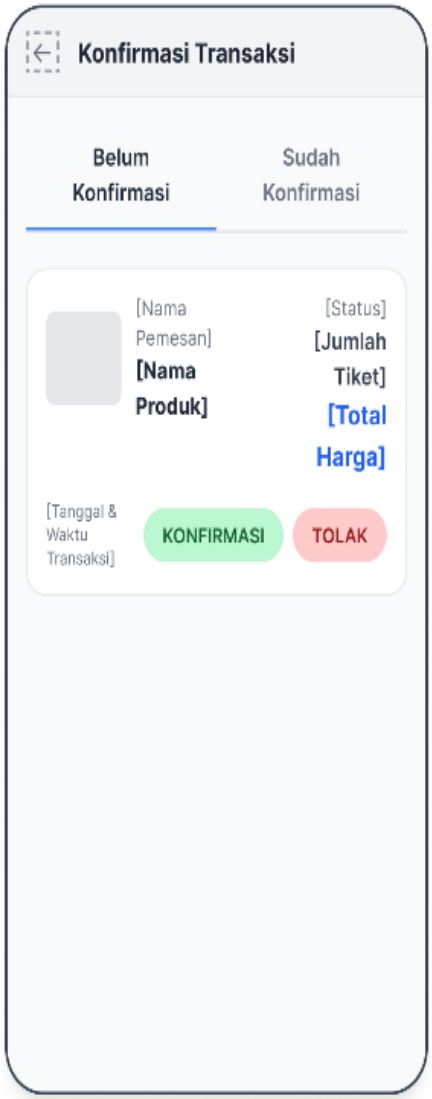


Gambar 4.18 Tampilan Pembayaran

j. Tampilan Konfirmasi Transaksi

Gambar tersebut merupakan rancangan antarmuka untuk Halaman Konfirmasi Transaksi, yang dirancang khusus bagi Pelaku Usaha untuk mengelola pesanan yang masuk. Di bagian bawah setiap kartu, terdapat dua tombol aksi utama, yaitu "Konfirmasi" untuk menyetujui dan

"Tolak" untuk membatalkan pesanan, yang memungkinkan Pelaku Usaha untuk mengelola alur transaksi secara efisien.



Gambar 4.19 Tampilan Konfirmasi

4.4 Penerapan Algoritma Dijkstra

Algoritma Dijkstra diimplementasikan untuk memenuhi tujuan utama sistem, yaitu memberikan rekomendasi tujuan berdasarkan rute perjalanan terpendek, bukan sekadar jarak garis lurus. Untuk menjalankan fungsinya, algoritma ini memerlukan model jaringan jalan dalam bentuk graf yang terdiri dari titik

(node) dan sisi (edge) yang memiliki bobot (jarak). Proses kerja Algoritma Dijkstra yang diimplementasikan dalam kode adalah sebagai berikut:

1. Inisialisasi Sistem menetapkan lokasi pengguna sebagai titik awal dengan jarak 0, sementara semua titik lain diberi jarak tak terhingga.
2. Selanjutnya Algoritma secara berulang memilih titik yang belum dikunjungi dengan jarak terkecil. Dari titik tersebut, algoritma memeriksa setiap tetangganya dan memperbarui jaraknya jika ditemukan rute yang lebih pendek. Proses ini menggunakan rumus inti:
   Jika d(u) + w(u, v) < d(v), maka d(v) := d(u) + w(u, v)
3. Sehingga hasil akhir proses ini berlanjut ketika jarak terpendek dari titik awal ke semua titik lain ditemukan.

   Contoh perhitungan manual :

   Kita akan simulasikan perhitungan Dijkstra pada sebuah graf sederhana yang mempresentasikan beberapa lokasi di Kecamatan Kemiling.

   Dengan model graf :

   a) Titik (Node): A (Lokasi Anda), B (Camp 91), C (Simpang Kedaung), D (Villa LDR), E (Taman Kupu-Kupu):
   b) Sisi (Edge) dan Bobot (Jarak dalam km):
      1) A → B : 5 km
      2) A → C : 2 km
      3) B → D : 6 km
      4) C → B : 1 km
      5) C → D : 4 km
      6) C → E : 7 km
      7) D → E : 3 km

Tabel 4.41 Algoritma Dijkstra

No	Node yang Belum Dikunjungi (Jarak)	Node Terpilih	Tetangga	Perhitungan jarak baru	Jarak final Diperbaharui
1.	A(0), B(∞), C(∞), D(∞), E(∞)	-	-	-	-

2.	A(0), B(∞), C(∞), D(∞), E(∞)	A	B C	d(A)+w(A,B) = 0+5=5d(A)+w(A,C) = 0+2=2	d(B)=5d(C)=2
3.	B(5), C(2), D(∞), E(∞)	C	B D E	d(C)+w(C,B) = 2+1=3 d(C)+w(C,D) = 2+4=6 d(C)+w(C,E) = 2+7=9	d(B)=3 (lebih kecil dari 5) d(D)=6 d(E)=9
4.	B(3), D(6), E(9)	B	D	d(B)+w(B,D) = 3+6=9	(Tidak ada perubahan, 9 > 6)
5.	D(6), E(9)	D	E	d(D)+w(D,E) = 6+3=9	(Tidak ada perubahan, 9 = 9)
6.	E(9)	E	-	-	-

Sehingga hasil akhir berdasarkan tabel diatas menemukan total jarak terpendek dari lokasi anda (A) adalah:

a. Jarak ke Camp 91 (B): 3 km (melalui Simpang Kedaung)
b. Jarak ke Simpang Kedaung (C): 2 km
c. Jarak ke Villa LDR (D): 6 km (melalui Simpang Kedaung)
d. Jarak ke Taman Kupu-Kupu (E): 9 km (melalui Simpang Kedaung)

Dengan contoh ini menunjukkan bagaimana Dijkstra secara cerdas menemukan rute ke Camp 91 lebih pendek jika melewati simpang Kedaung (2+1=3 KM) daripada rute langsung (5 km). Proses manual inilah yang dieksekusi oleh kode PHP di bawah ini secara otomatis dan cepat.

Berikut ini kode Implementasi dari Algoritma Dijkstra pada API *(Application Programming Interface)* menggunakan Bahasa pemrograman PHP (*Hypertext Preprocessor*):

Tabel 4.42 Kode Algoritma Dijkstra

```
<?php
        function getDistanceMatrix($nodes, $apiKey) {
            $locations = [];
            $nodeKeys = array_keys($nodes); // Simpan urutan key/ID
```

```
    foreach ($nodes as $node) {
        $locations[] = $node['lat'] . ',' . $node['lng'];
    }
    $locationStr = implode('|', $locations);

                                        $url =
"https://maps.googleapis.com/maps/api/distancematrix/json?o
rigins=" . urlencode($locationStr) . "&destinations=" .
urlencode($locationStr) . "&key={$apiKey}";

    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($ch, CURLOPT_TIMEOUT, 20);
    $response = curl_exec($ch);
    curl_close($ch);

    $result = json_decode($response, true);
    $matrix = [];

    if (isset($result['rows'])) {
        foreach ($result['rows'] as $i => $row) {
            $originKey = $nodeKeys[$i];
            $matrix[$originKey] = [];
            foreach ($row['elements'] as $j => $element) {
                $destKey = $nodeKeys[$j];
                if ($element['status'] == 'OK') {
                                    $matrix[$originKey][$destKey] =
$element['distance']['value'] / 1000; // Jarak dalam KM
                } else {
```

```
                    $matrix[$originKey][$destKey] = INF; // Tandai
tidak terjangkau
                }
            }
        }
        return $matrix;
    }
    return null;
}

// Implementasi Algoritma Dijkstra
function dijkstra_all_distances($graph, $startNode) {
    $distances = [];
    $queue = new SplPriorityQueue();
    foreach ($graph as $node => $edges) {
        $distances[$node] = INF;
    }
    $distances[$startNode] = 0;
    $queue->insert($startNode, 0);
    while (!$queue->isEmpty()) {
        $currentNode = $queue->extract();
                    if    ($distances[$currentNode]    ===    INF    ||
!isset($graph[$currentNode])) {
            continue;
        }
        foreach ($graph[$currentNode] as $neighbor => $weight)
{
            $alt = $distances[$currentNode] + $weight;
            if ($alt < $distances[$neighbor]) {
                $distances[$neighbor] = $alt;
                $queue->insert($neighbor, -$alt);
```

```
                }
            }
        }
        return $distances;
    }
?>
```

Pengujian dilakukan menggunakan perangkat lunak *Postman* dengan mengirimkan permintaan *POST* yang berisi data *latitude* dan *longitude*. Sebagai studi kasus, digunakan titik koordinat Kampus IIB Darmajaya ("latitude": -5.377201, "longitude": 105.249535).

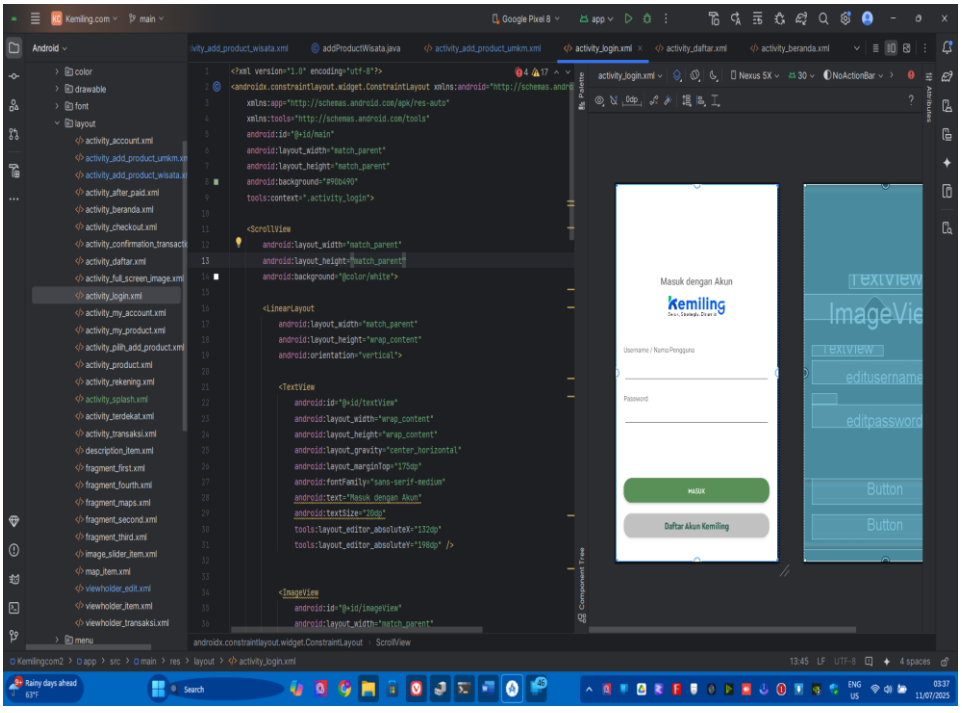

```
{
  "status": "success",
  "data": [
    {
      "id_produk": "141",
      "nama_produk": "Bukit Sakura",
      "kategori": "Tempat Wisata",
      "distance_route_km": 5.58
    },
    {
      "id_produk": "146",
      "nama_produk": "KDI Handmade",
      "kategori": "UMKM",
      "distance_route_km": 7.00
    },
    {
      "id_produk": "138",
      "nama_produk": "Kopi Gunawan",
      "kategori": "UMKM",
```

```
            "distance_route_km": 8.35
        }
    ]
}
```

4.5 Tahapan Pengkodean dan Integrasi Sistem (Contruction)

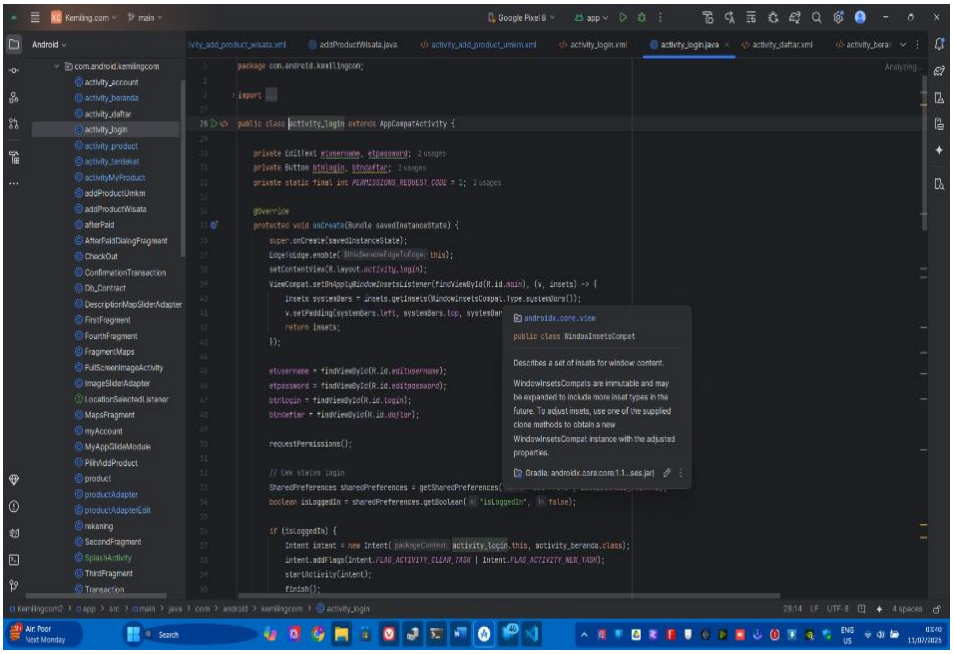
Pada tahap ini rancangan yang sudah di buat akan diimplementasi ke dalam kode program yang sesuai dengan desain sistem. Dan akan dilakukan testing menggunakan blackbox untuk menguji fungsionalitas sistem yang sudah dibangun serta mengetahui apakah aplikasi sudah berfungsi dan sesuai dengan rancangan.

a) Halaman Login



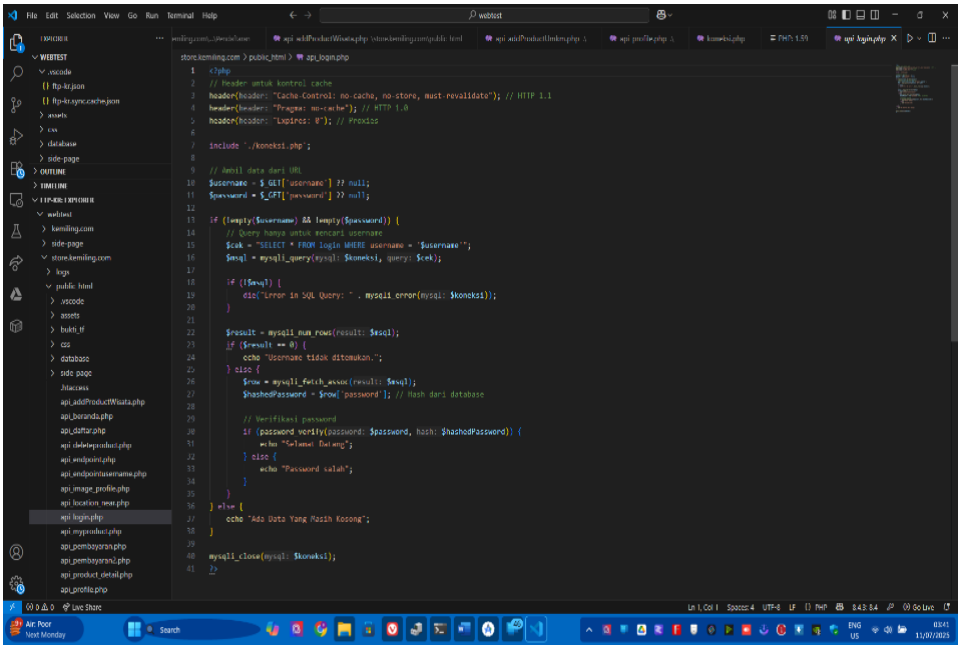
Gambar 4.20 Kode Desain Login

Pada gambar 4.20 diatas mendefinisikan struktur visual dan komponen antarmuka untuk halaman login aplikasi. Secara ringkas, kode ini berfungsi untuk menyusun semua komponen visual secara deklaratif guna menciptakan halaman login yang fungsional bagi pengguna.



Gambar 4.21 Kode Login

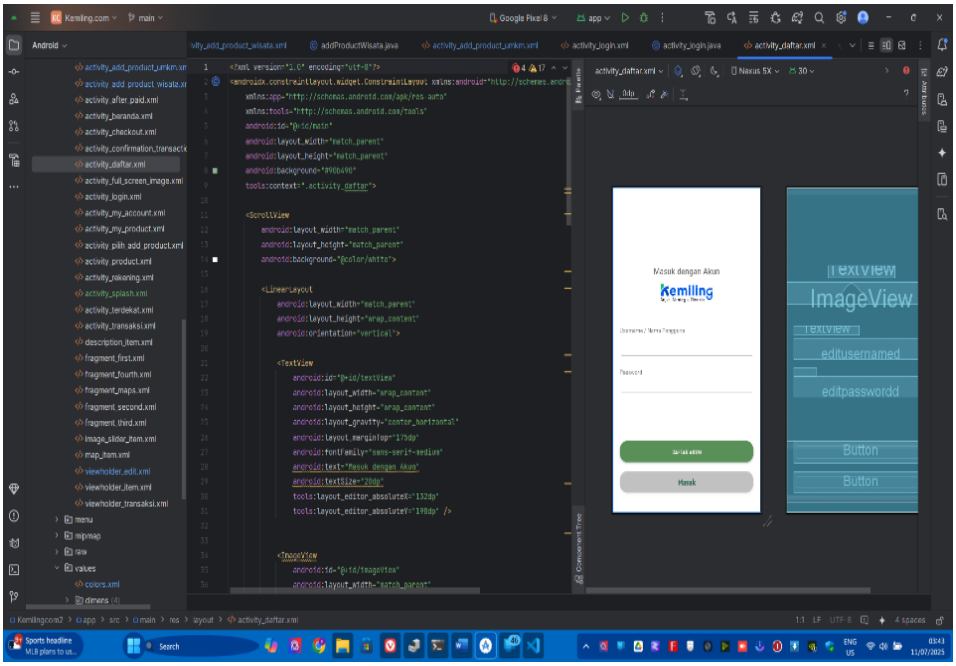
Pada gambar 4.21 diatas berfungsi sebagai logika kode untuk antarmuka halaman login. Di dalamnya kode menghubungkan elemen visual dari file xml ke variabel kode java. Fungsi utamanya adalah menangani input dari interaksi pengguna lalu mengarahkannya ke halaman utama jika sesi masih aktif.



Gambar 4. 22 Api Login

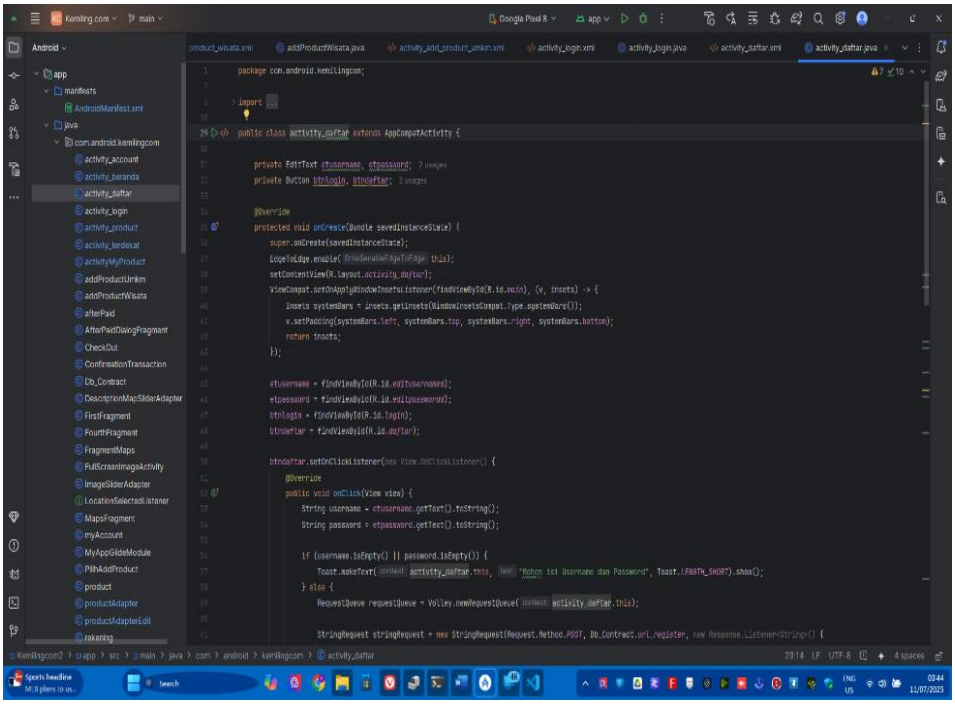
Pada gambar 4.22 diatas adalah kode backend (API) yang berfungsi untuk memproses permintaan login dari aplikasi android. Setelah mendapatkan data username dan password dari pengguna yang dikirimkan, akan dilakukan kueri ke database untuk mencari pengguna berdasarkan data yang dikirim. Jika di temukan, maka akan diverifikasi dan tersimpan ke database.

b) Halaman Daftar



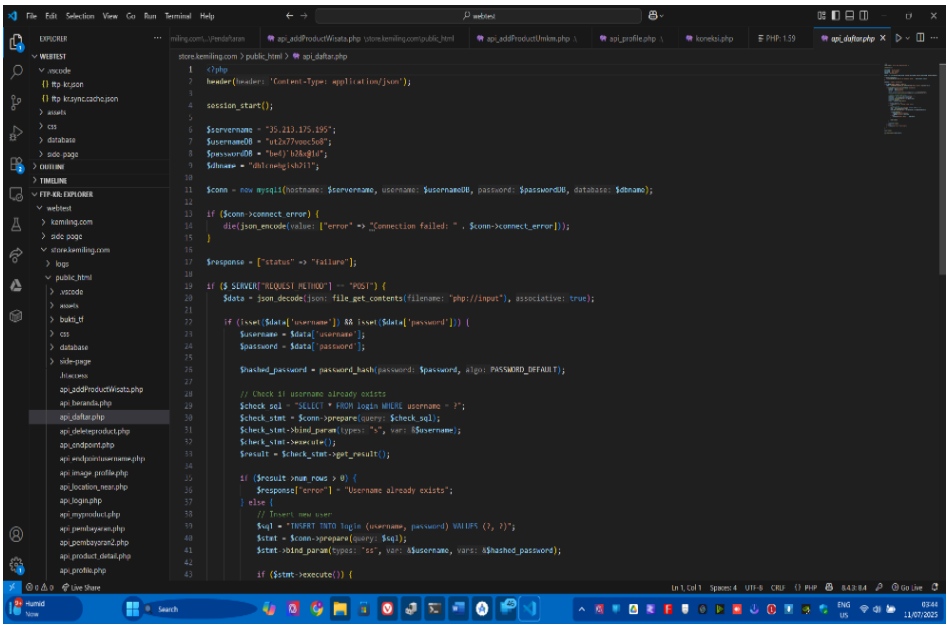
Gambar 4.23 Kode Desain Daftar

Pada gambar 4.23 diatas mendefinisikan struktur visual dan komponen antarmuka untuk halaman login aplikasi. Secara ringkas, kode ini berfungsi untuk menyusun semua komponen visual secara deklaratif guna menciptakan halaman Daftar yang fungsional bagi pengguna.



Gambar 4.24 Kode Daftar

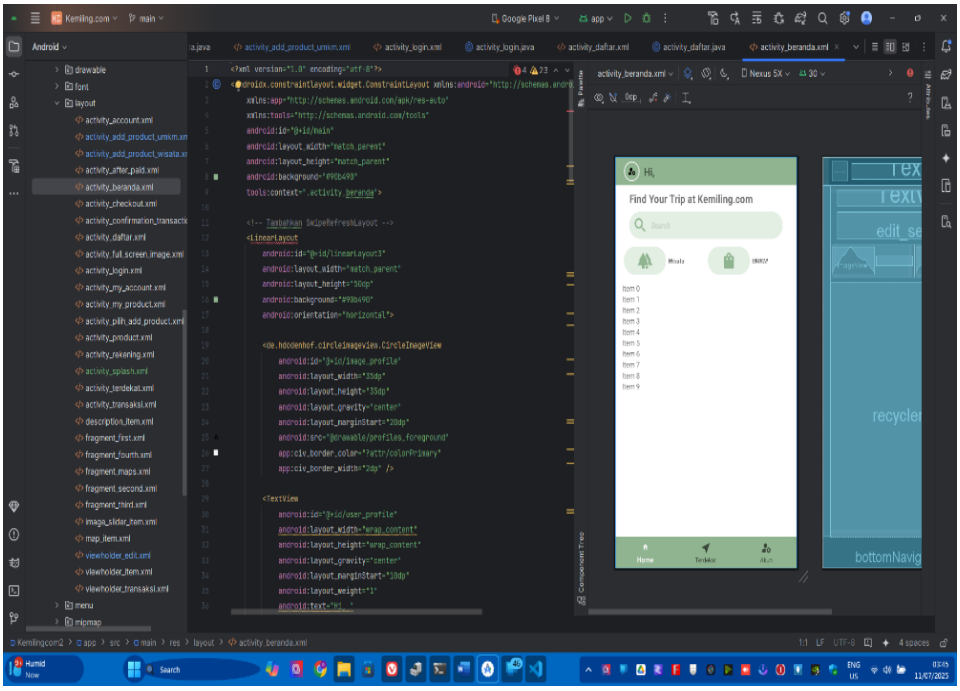
Pada gambar 4.24 diatas berfungsi sebagai logika kode untuk antarmuka halaman daftar. Di dalamnya kode menghubungkan elemen visual dari file xml ke variabel kode java. Fungsi utamanya adalah menangani input dari interaksi pengguna lalu mengarahkannya ke halaman login jika akun berhasil terdaftar.



Gambar 4.25 Kode Api Daftar

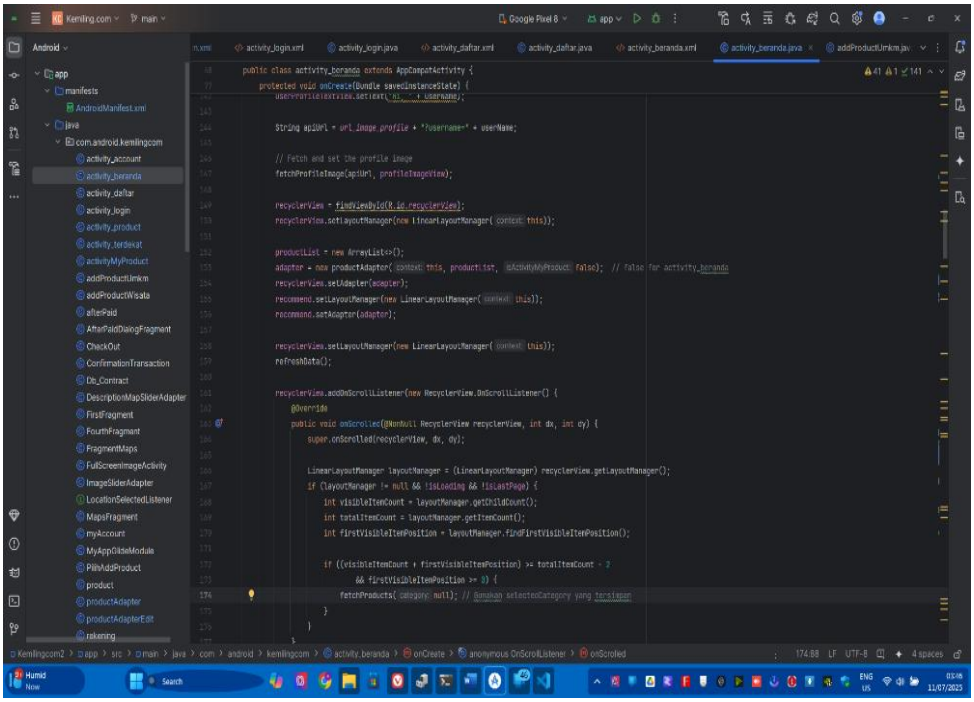
Pada gambar 4.25 diatas adalah kode backend (API) yang berfungsi untuk memproses permintaan daftar dari aplikasi android. Setelah mendapatkan data username dan password dari pengguna yang dikirimkan, akan dilakukan penyimpanan ke database berdasarkan data yang dikirim. Jika berhasil, maka akan diverifikasi dan tersimpan ke database

c) Halaman Beranda



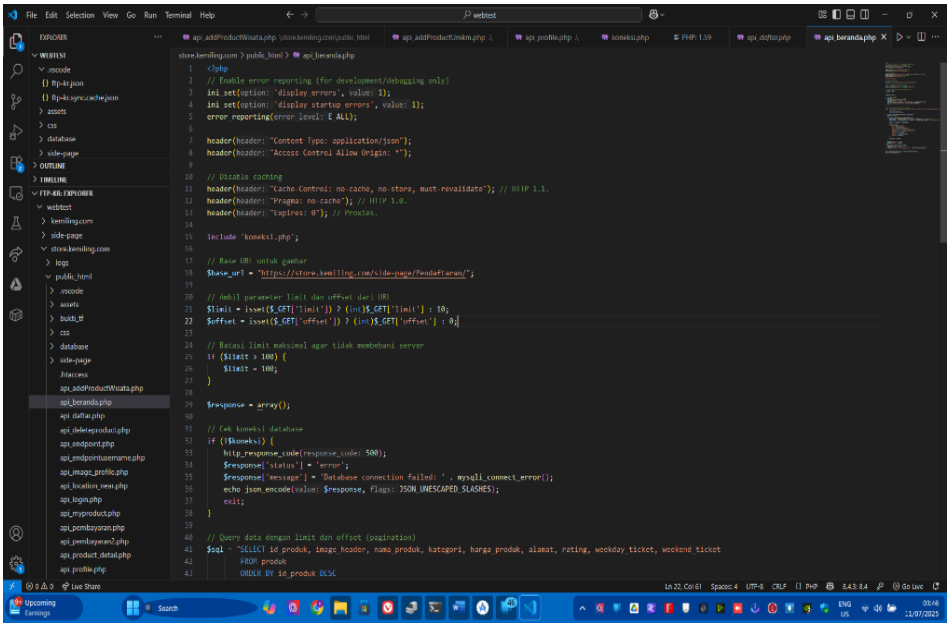
Gambar 4.26 Kode Desain Beranda

Pada gambar 4.26 diatas mendefinisikan struktur visual dan komponen antarmuka untuk halaman beranda aplikasi. Secara ringkas, kode ini berfungsi untuk menyusun semua komponen visual secara deklaratif guna menciptakan halaman beranda yang fungsional bagi pengguna.



Gambar 4.27 Kode Beranda

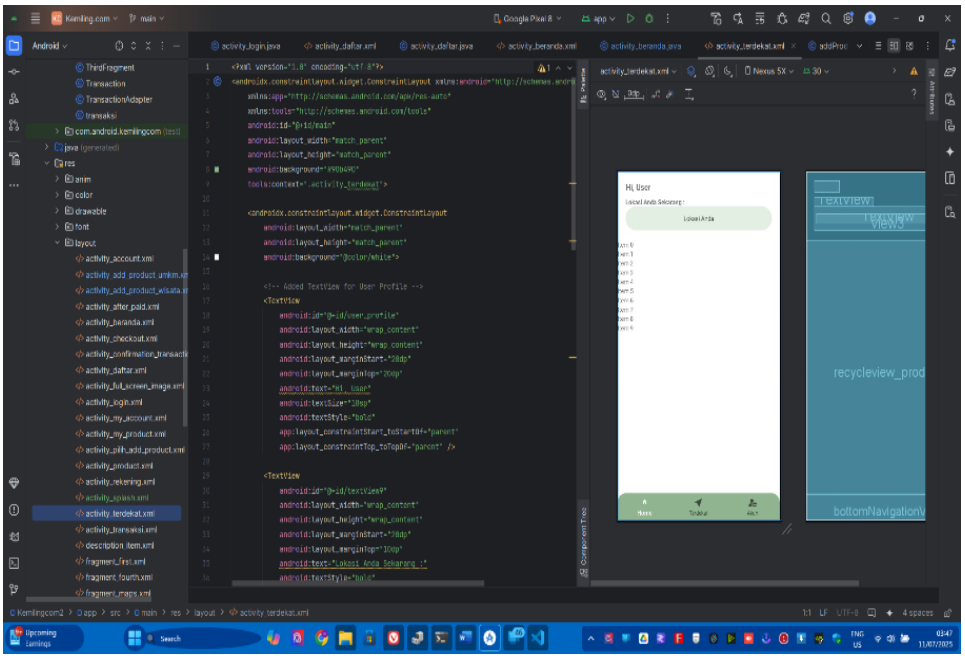
Pada gambar 4.27 diatas berfungsi sebagai logika kode untuk antarmuka halaman beranda. Di dalamnya kode menghubungkan elemen visual dari file xml ke variabel kode java. Fungsi utamanya adalah menampilkan produk wisata dan umkm dari data yang di input oleh pengguna atau pelaku usaha.



Gambar 4.28 Kode Api Beranda

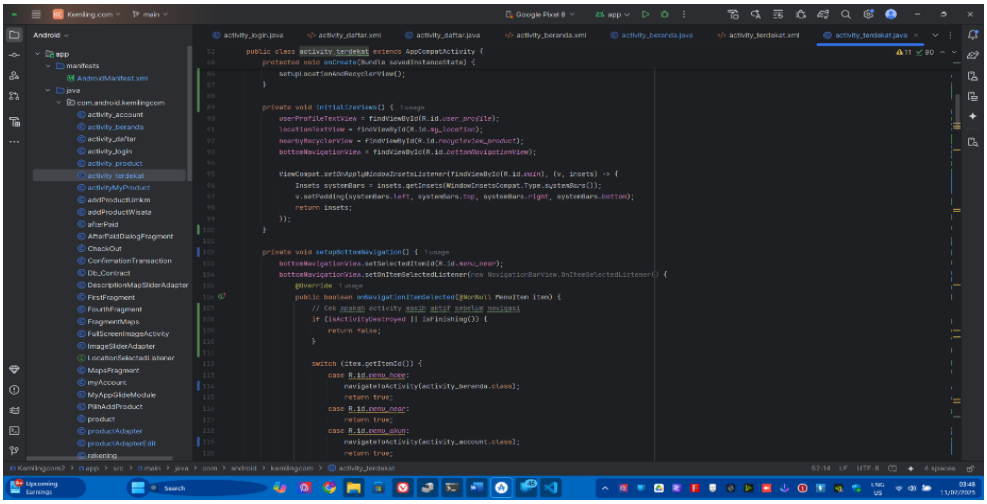
Pada gambar 4.28 diatas adalah kode backend (API) yang berfungsi untuk memproses permintaan get data dari database dan di tampilkan ke halaman beranda aplikasi.

d) Halaman Terdekat



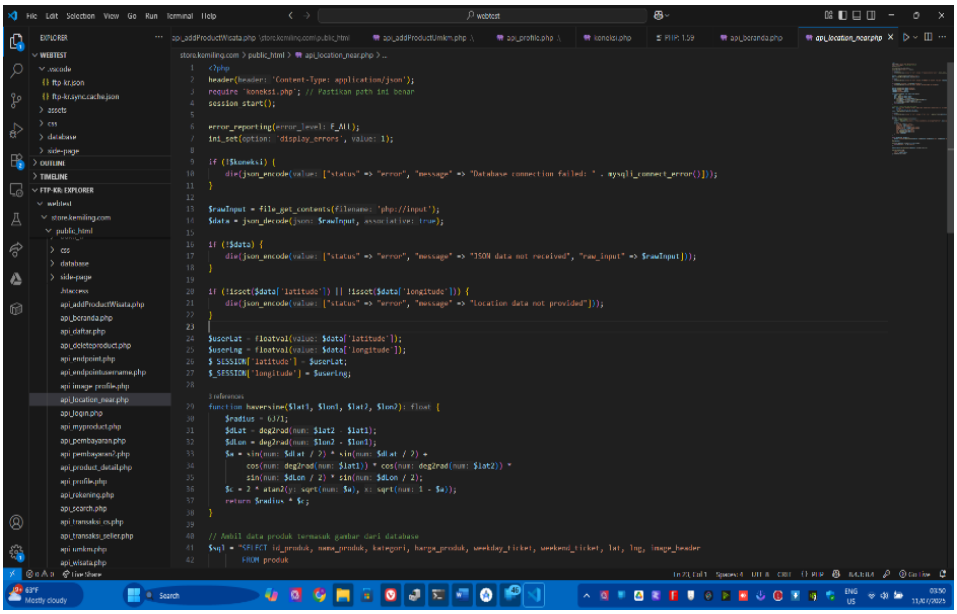
Gambar 4.29 Kode Desain Terdekat

Pada gambar 4.29 diatas mendefinisikan struktur visual dan komponen antarmuka untuk halaman terdekat aplikasi. Secara ringkas, kode ini berfungsi untuk menyusun semua komponen visual secara deklaratif guna menciptakan halaman terdekat yang fungsional bagi pengguna.



Gambar 4.30 Kode Terdekat

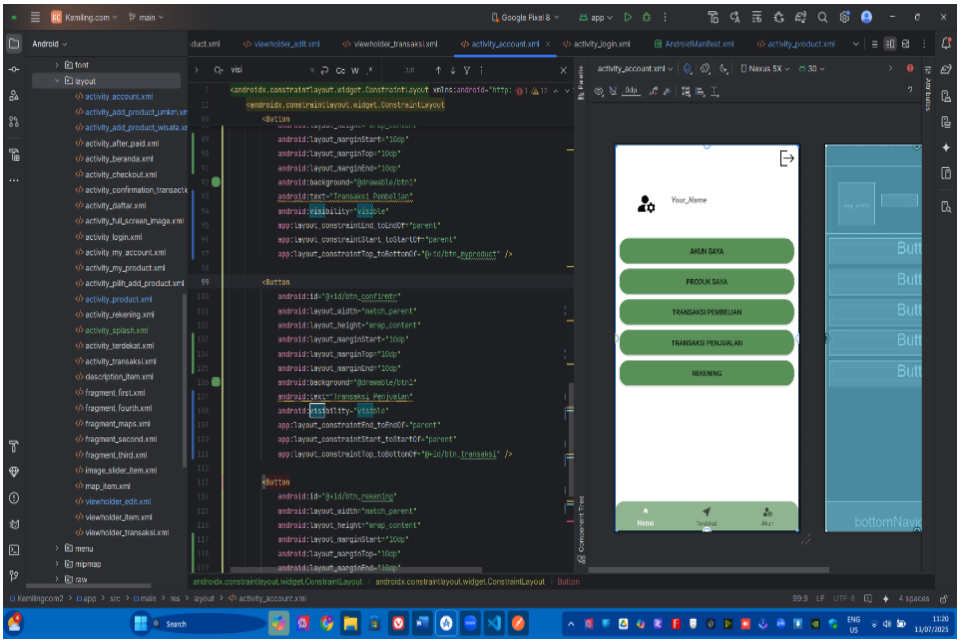
Pada gambar 4.30 diatas berfungsi sebagai logika kode untuk antarmuka halaman terdekat. Di dalamnya kode menghubungkan elemen visual dari file xml ke variabel kode java. Fungsi utamanya adalah menampilkan produk wisata dan umkm dari data yang di input oleh pengguna atau pelaku usaha. Sebelum di tampilkan, data di proses oleh algoritma sehingga data yang tertampil adalah data produk terdekat dari pengguna.



Gambar 4.31 Kode Api Terdekat

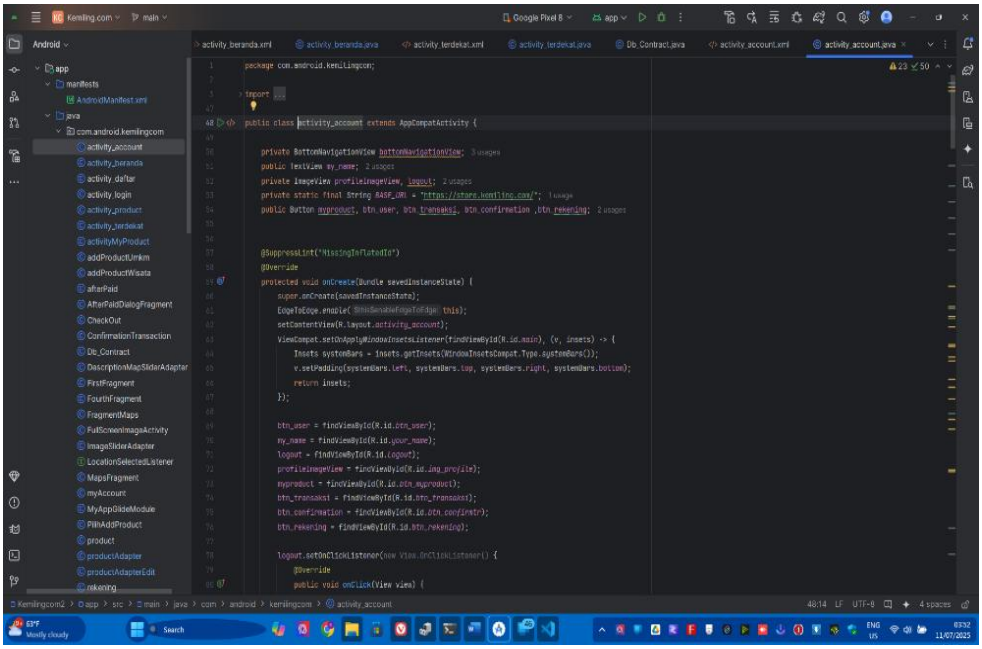
Pada gambar 4.31 diatas adalah kode backend (API) yang berfungsi untuk memproses permintaan get data dari database dan di tampilkan ke halaman terdekat aplikasi. Kode ini terimplementasi algoritma dijkstra, sehingga data yang di tampilkan adalah hasil dari penerapan algoritma dijktra, dan merupakan data terdekat dari pengguna secara *real-time.*

e) Halaman Akun



Gambar 4.32 Kode Desain Akun

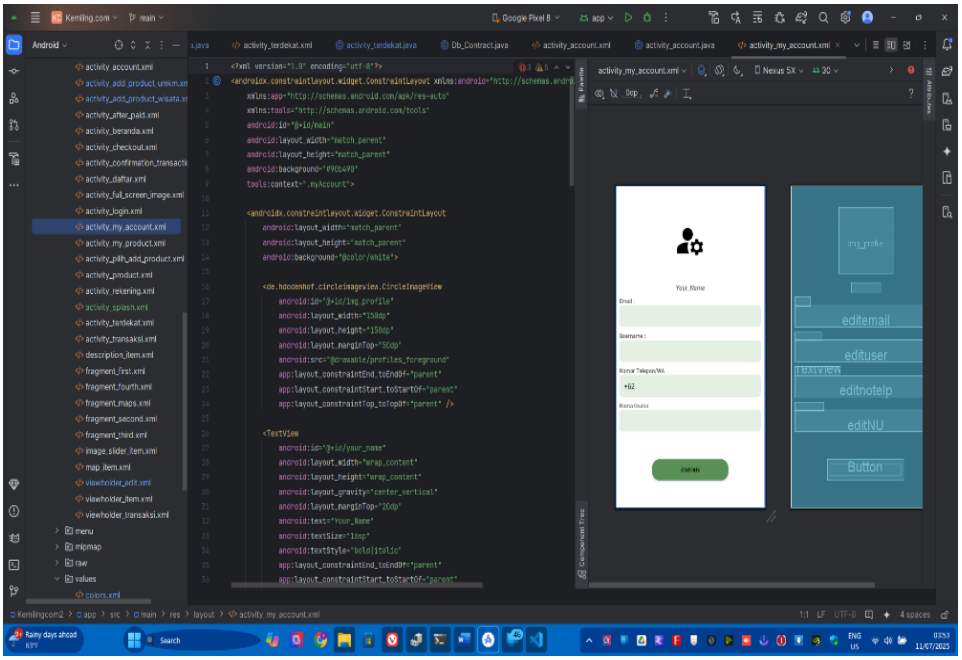
Pada gambar 4.32 diatas mendefinisikan struktur visual dan komponen antarmuka untuk halaman akun aplikasi. Secara ringkas, kode ini berfungsi untuk menyusun semua komponen visual secara deklaratif guna menciptakan halaman akun yang fungsional bagi pengguna.



Gambar 4.33 Kode akun

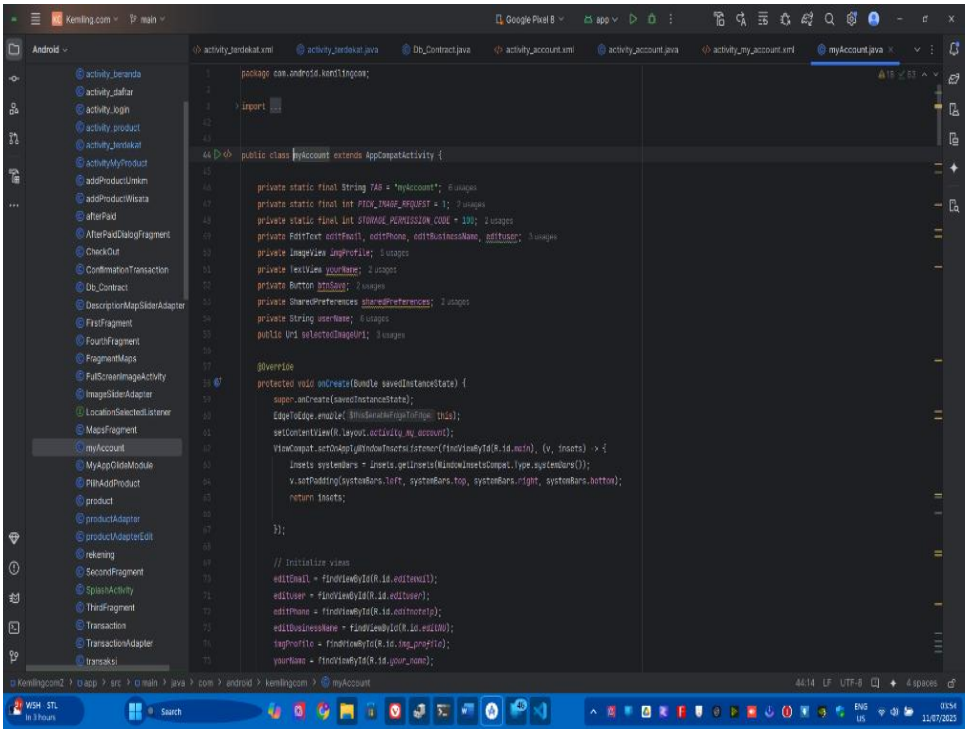
Pada gambar 4.33 diatas berfungsi sebagai logika kode untuk antarmuka halaman akun. Di dalamnya kode menghubungkan elemen visual dari file xml ke variabel kode java. Fungsi utamanya adalah menampilkan beberapa tombol pilihan untuk melakukan perubahan data, baik data akun ataupun data transaksi hingga produk dan akan tersimpan ke dalam database.

f) Halaman Akun Saya



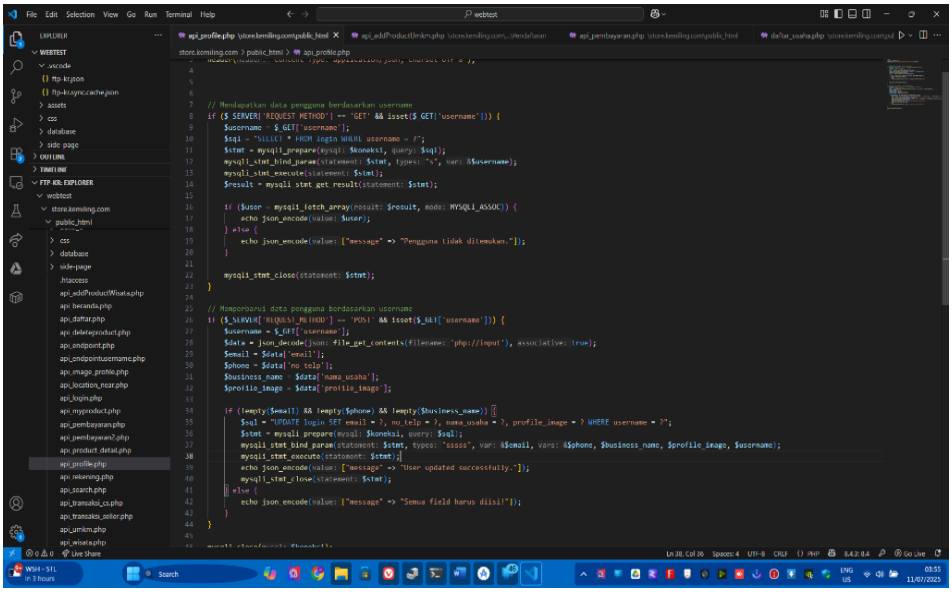
Gambar 4.34 Kode Desain Akun Saya

Pada gambar 4.34 diatas mendefinisikan struktur visual dan komponen antarmuka untuk halaman akun saya aplikasi. Secara ringkas, kode ini berfungsi untuk menyusun semua komponen visual secara deklaratif guna menciptakan halaman akun saya yang fungsional bagi pengguna.



Gambar 4.35 Kode Akun Saya

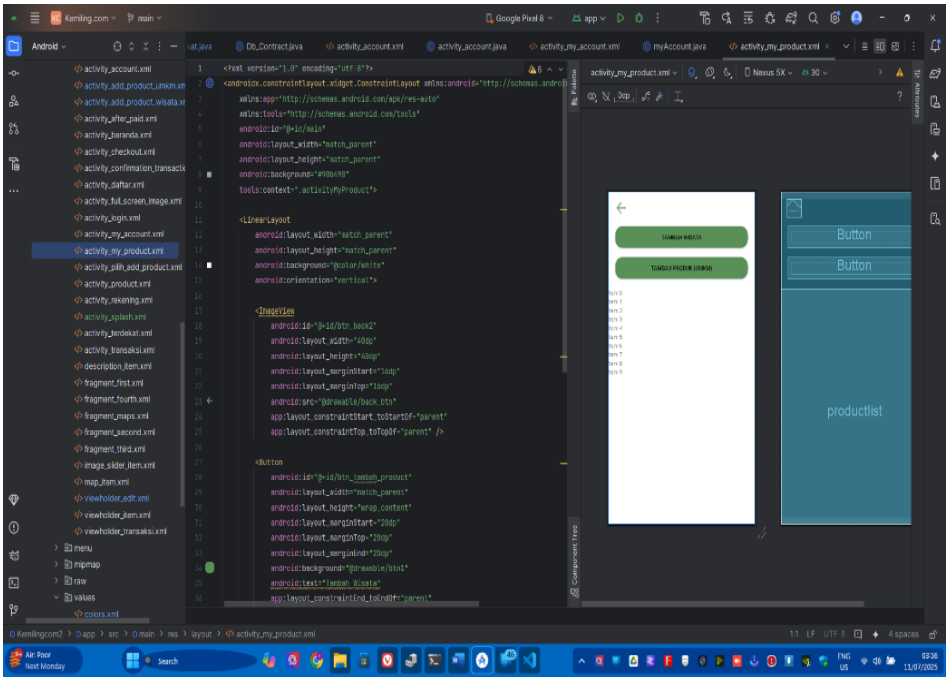
Pada gambar 4.35 diatas berfungsi sebagai logika kode untuk antarmuka halaman akun saya. Di dalamnya kode menghubungkan elemen visual dari file xml ke variabel kode java. Fungsi utamanya adalah menampilkan beberapa field untuk melakukan perubahan data akun dan akan tersimpan ke dalam database.



Gambar 4.36 Kode Api Akun Saya

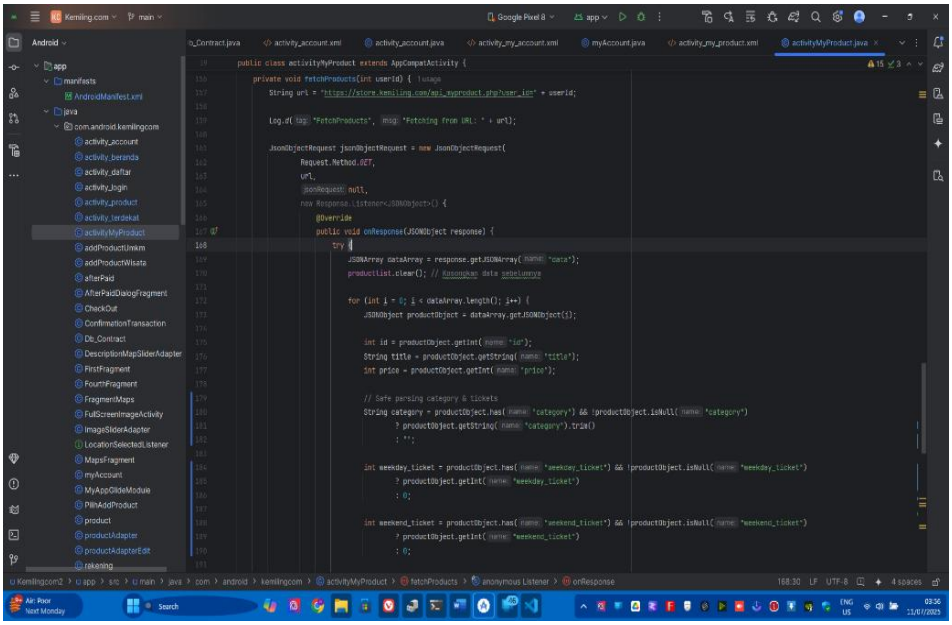
Pada gambar 4.36 diatas adalah kode backend (API) yang berfungsi untuk memproses permintaan data dari database dan di tampilkan ke halaman akun saya di aplikasi. Kode ini dapat menerima perubahan data dan menyimpan ke database.

g) Halaman Produk Saya



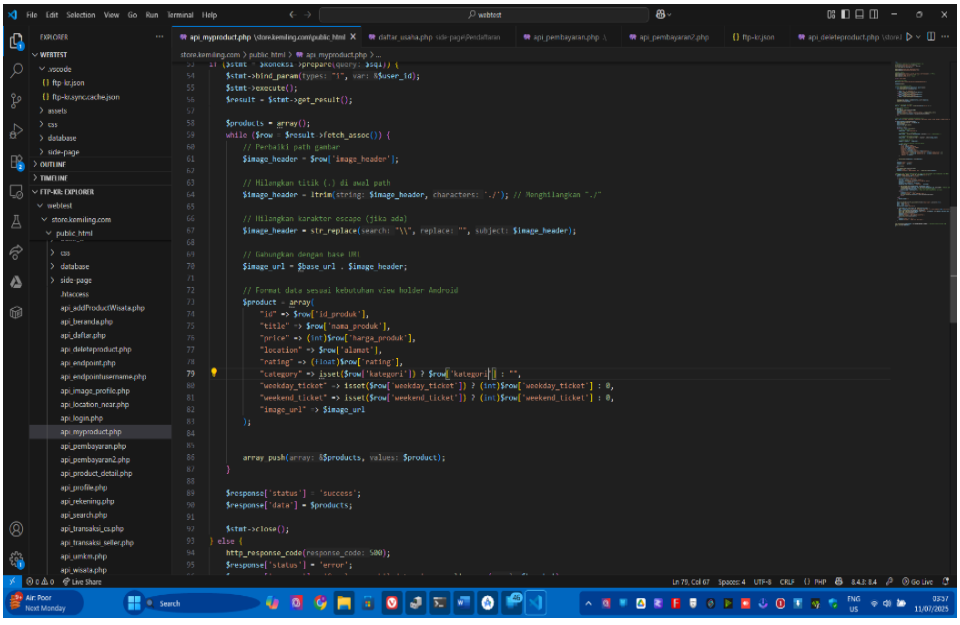
Gambar 4.37 Kode Desain Produk Saya

Pada gambar 4.37 diatas mendefinisikan struktur visual dan komponen antarmuka untuk halaman produk saya aplikasi. Secara ringkas, kode ini berfungsi untuk menyusun semua komponen visual secara deklaratif guna menciptakan halaman produk saya yang fungsional bagi pengguna.



Gambar 4.38 Kode Produk Saya

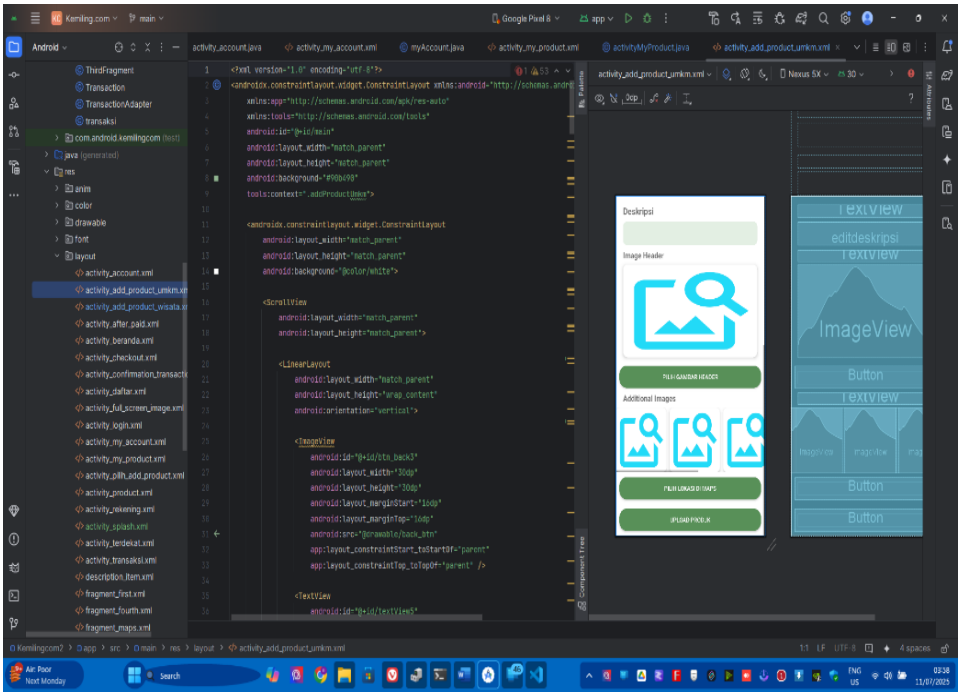
Pada gambar 4.38 diatas berfungsi sebagai logika kode untuk antarmuka halaman produk saya. Di dalamnya kode menghubungkan elemen visual dari file xml ke variabel kode java. Fungsi utamanya adalah menampilkan data produk yang sudah di upload oleh pengguna (pelaku usaha) baik dari wisata maupun umkm.



Gambar 4.39 Kode Api Produk Saya

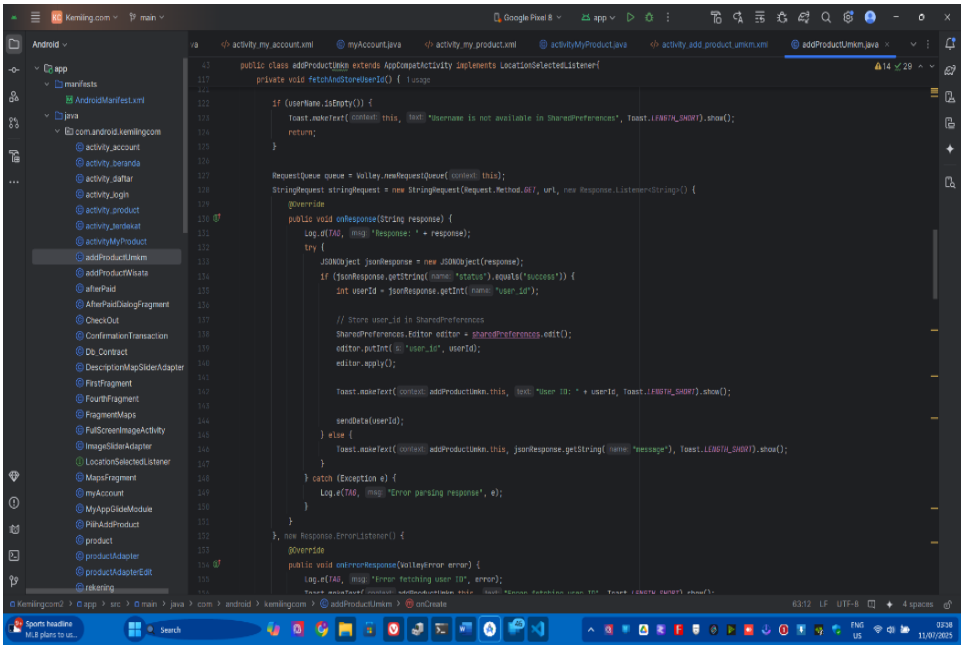
Pada gambar 4.39 diatas adalah kode backend (API) yang berfungsi untuk memproses permintaan data dari database dan di tampilkan ke halaman produk saya di aplikasi. Kode ini dapat mengambil data dari database dan akan di tampilkan ke aplikasi android.

h) Halaman Tambah Produk



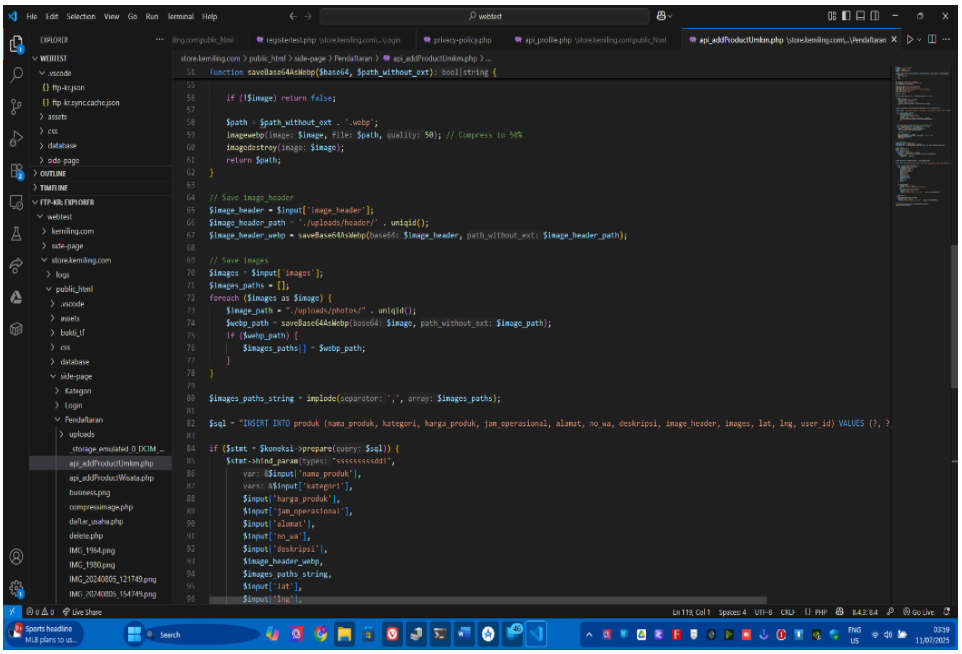
Gambar 4.40 Kode Tambah Produk

Pada gambar 4.40 diatas mendefinisikan struktur visual dan komponen antarmuka untuk halaman tambah produk di aplikasi. Secara ringkas, kode ini berfungsi untuk menyusun semua komponen visual secara deklaratif guna menciptakan halaman tambah produk yang fungsional bagi pengguna.



Gambar 4.41 Kode Tambah Produk

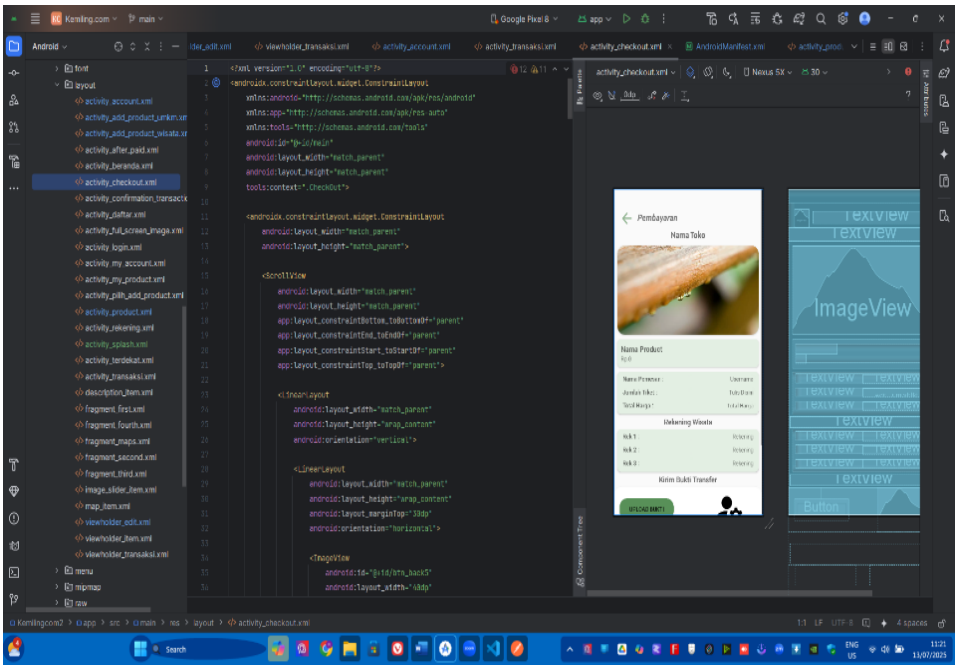
Pada gambar 4.41 diatas berfungsi sebagai logika kode untuk antarmuka halaman tambah produk. Di dalamnya kode menghubungkan elemen visual dari file xml ke variabel kode java. Fungsi utamanya adalah mengupload data produk ke database oleh pengguna (pelaku usaha) baik dari wisata maupun umkm.



Gambar 4.42 Kode Api Tambah Produk

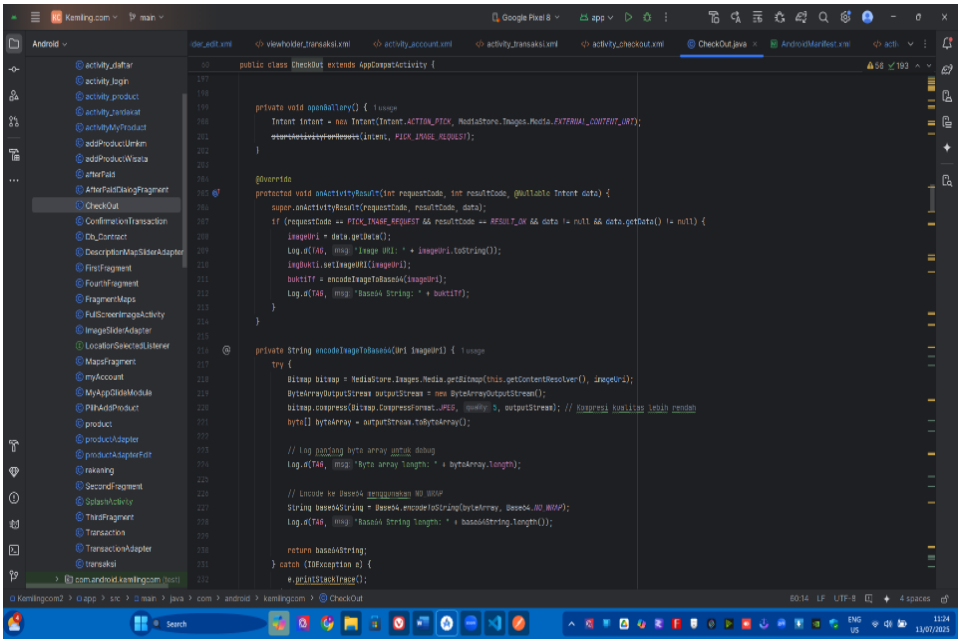
Pada gambar 4.42 diatas adalah kode backend (API) yang berfungsi untuk memproses perngiriman data dari aplikasi ke database. Kode ini dapat mengirim data ke database dan akan di tampilkan ke aplikasi android.

i) Halaman Pembayaran



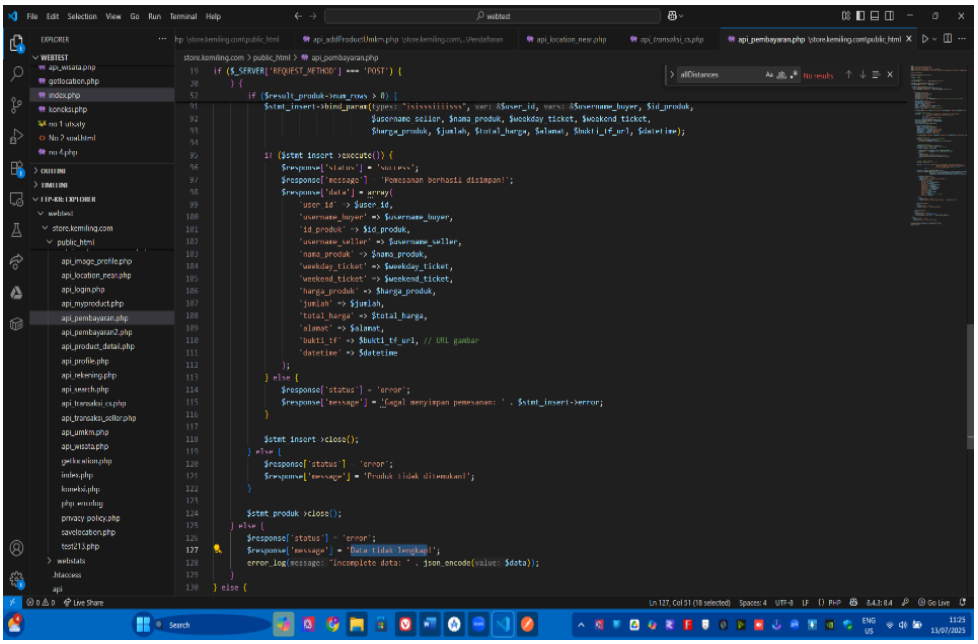
Gambar 4.43 Desain Halaman Pembayaran

Pada gambar 4.43 diatas mendefinisikan struktur visual dan komponen antarmuka untuk halaman pembayaran di aplikasi. Secara ringkas, kode ini berfungsi untuk menyusun semua komponen visual secara deklaratif guna menciptakan halaman pembayaran yang fungsional bagi pengguna.



Gambar 4.44 Kode Pembayaran

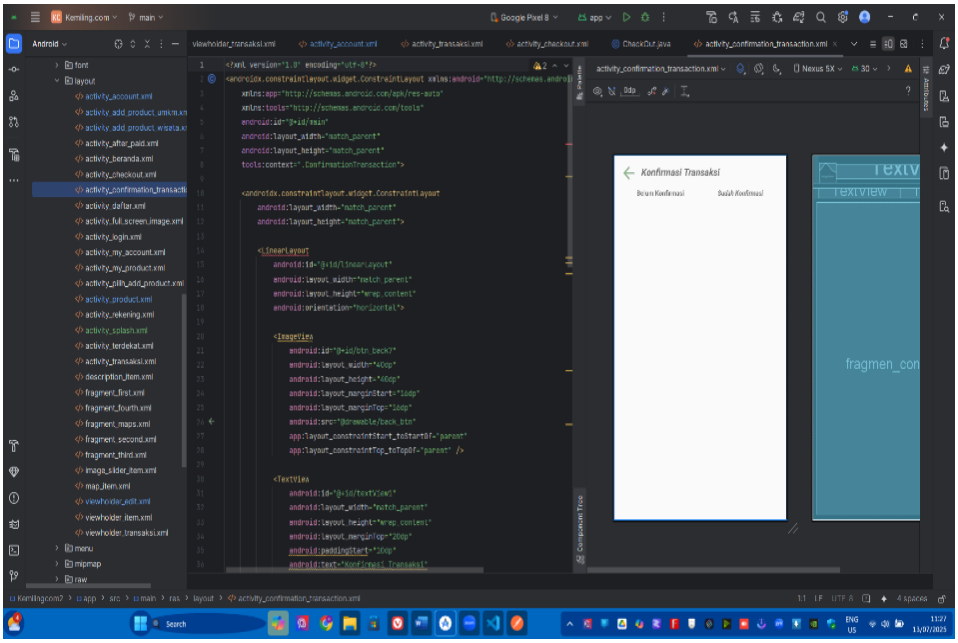
Pada gambar 4.44 diatas berfungsi sebagai logika kode untuk antarmuka halaman Pembayaran. Di dalamnya kode menghubungkan elemen visual dari file xml ke variabel kode java. Fungsi utamanya adalah mengupload data transaksi ke database oleh pengguna dan dapat di verifikasi oleh pelaku usaha.



Gambar 4.45 Kode API Pembayaran

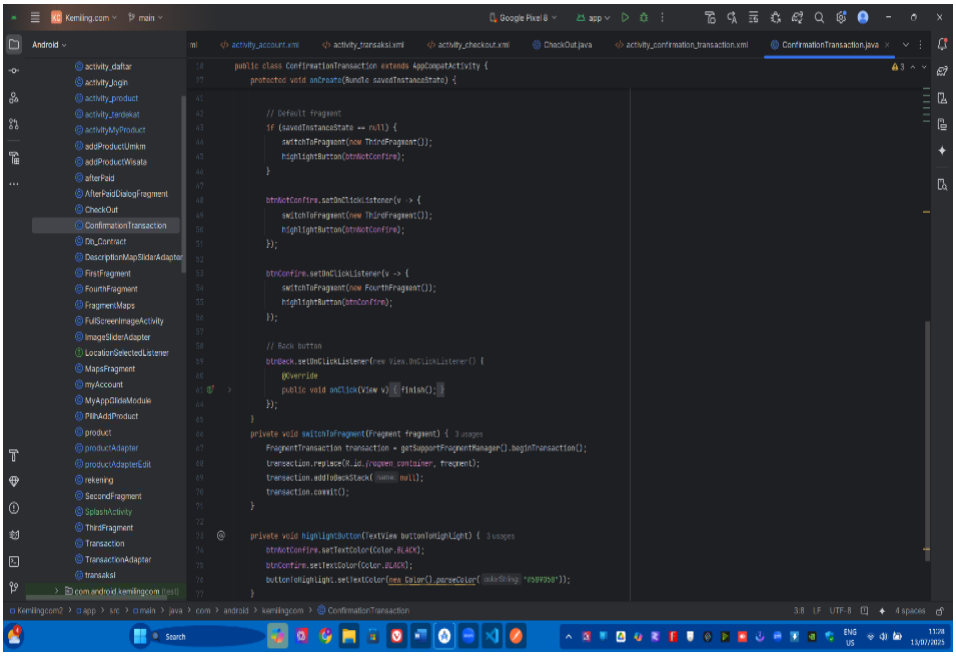
Pada gambar 4.45 diatas adalah kode backend (API) yang berfungsi untuk memproses pengiriman data dari aplikasi ke database. Kode ini dapat mengirim data ke database dan akan di tampilkan ke aplikasi android pada halaman transaksi.

j) Halaman Konfirmasi Transaksi



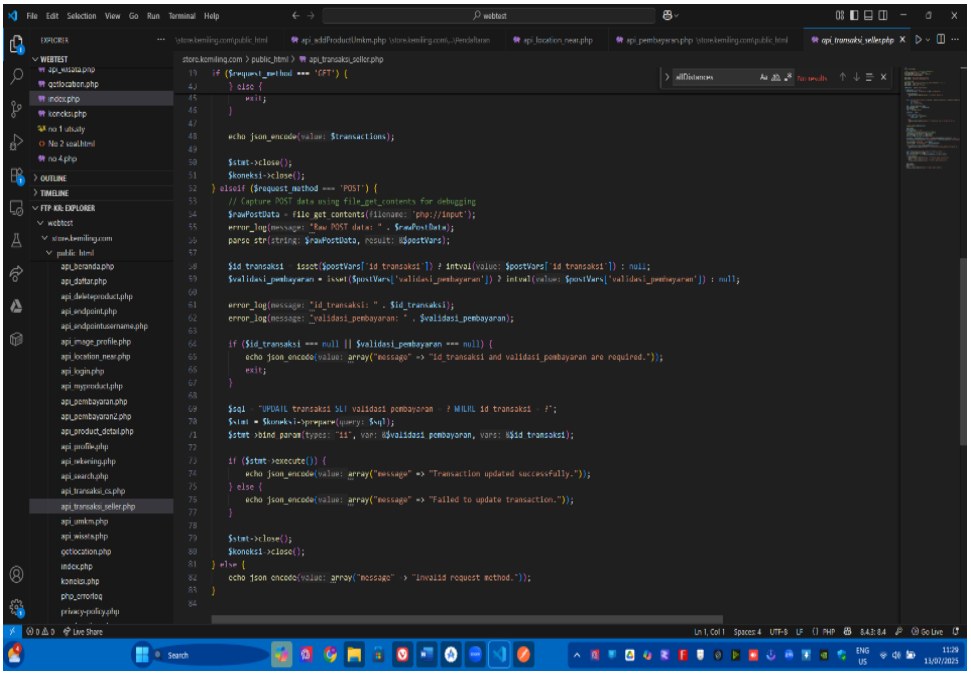
Gambar 4.46 Desain Halaman Konfirmasi

Pada gambar 4.46 diatas mendefinisikan struktur visual dan komponen antarmuka untuk halaman konfirmasi transksi di aplikasi. Secara ringkas, kode ini berfungsi untuk menyusun semua komponen visual secara deklaratif guna menciptakan halaman konfirmasi transaksi yang fungsional bagi pengguna.



Gambar 4.47 Kode Konfirmasi

Pada gambar 4.47 diatas berfungsi sebagai logika kode untuk antarmuka halaman konfirmasi transaksi. Di dalamnya kode menghubungkan elemen visual dari file xml ke variabel kode java. Fungsi utamanya adalah mengupload data transaksi ke database oleh pengguna dan dapat di verifikasi oleh pelaku usaha.



Gambar 4.48 Kode API Konfirmasi

Pada gambar 4.48 diatas adalah kode backend (API) yang berfungsi untuk memproses pengiriman data dari aplikasi ke database. Kode ini dapat mengirim data ke database tentang hasil konfirmasi transaksi yang di lakukan pelaku usaha.

4.6 Tahapan Implementasi (Cutover)

Hasil dari seluruh proses perancangan dan pengembangan adalah sebuah purwarupa aplikasi Android fungsional yang siap untuk digunakan, menyediakan platform terintegrasi bagi wisatawan untuk menemukan destinasi wisata dan produk UMKM, serta bagi pelaku usaha untuk mengelola konten mereka. Fitur-fitur utama seperti pendaftaran pengguna, penjelajahan konten, dan penambahan produk telah diuji dan berjalan sesuai dengan hasil yang diharapkan. Secara khusus, implementasi Algoritma Dijkstra berhasil menyediakan fungsionalitas rekomendasi tujuan terdekat berdasarkan jarak rute perjalanan yang efisien.

4.6.1 Halaman Login

Implementasi halaman Login sebagai halaman awal pada saat aplikasi di buka yang mengharuskan pengguna memasukkan username dan password sebelum membuka aplikasi.

Gambar 4.49 Halaman Login

4.6.2 Halaman Daftar

Implemetasi halaman Daftar yang dirancang sebagai alur pendaftaran bagi pengguna baru. Dengan hanya menampilkan kolom input

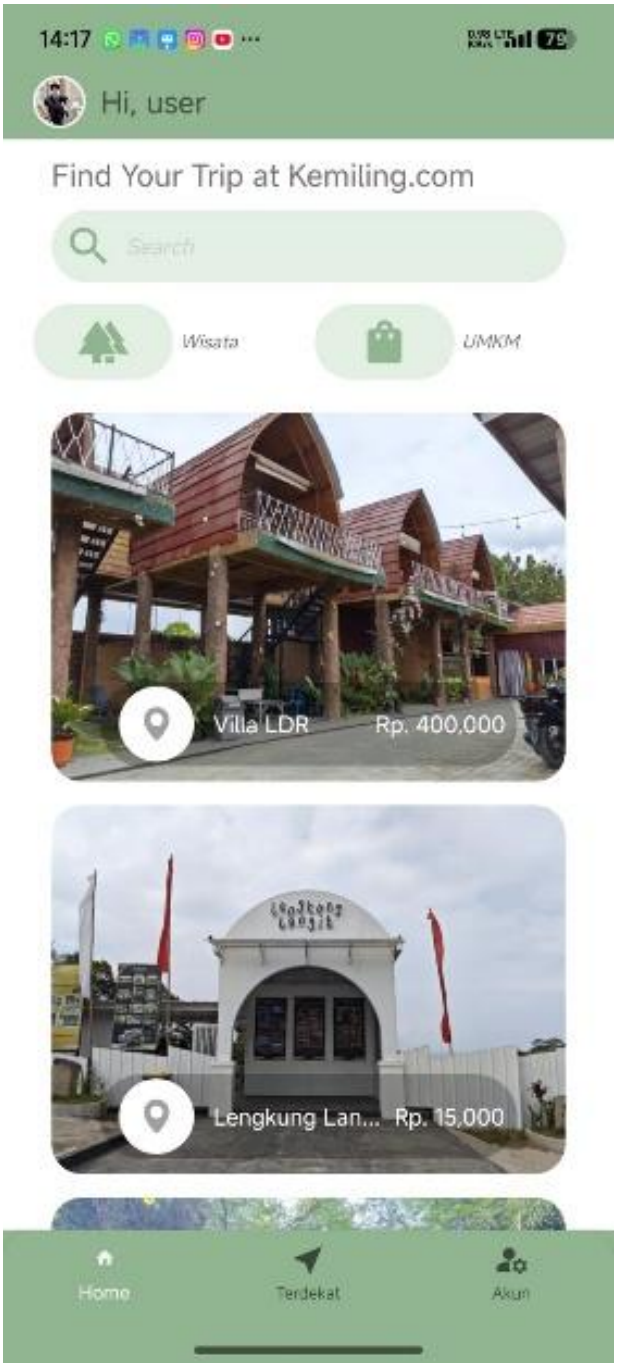
username dan password serta dua tombol aksi seperti “Daftar Akun” yang berfunsi sebagai tombol yang memproses data pendaftaran.

Gambar 4.50 Halaman Daftar Akun

4.6.3 Halaman Beranda

Implementasi halaman Beranda yang dirancang sebagai halaman utama yang menampilkan produk destinasi wisata dan UMKM dari pelaku

usaha kecamatan Kemiling. Dan pengguna juga bisa memilih produk atau destinasi wisata yang ingin mereka tuju.

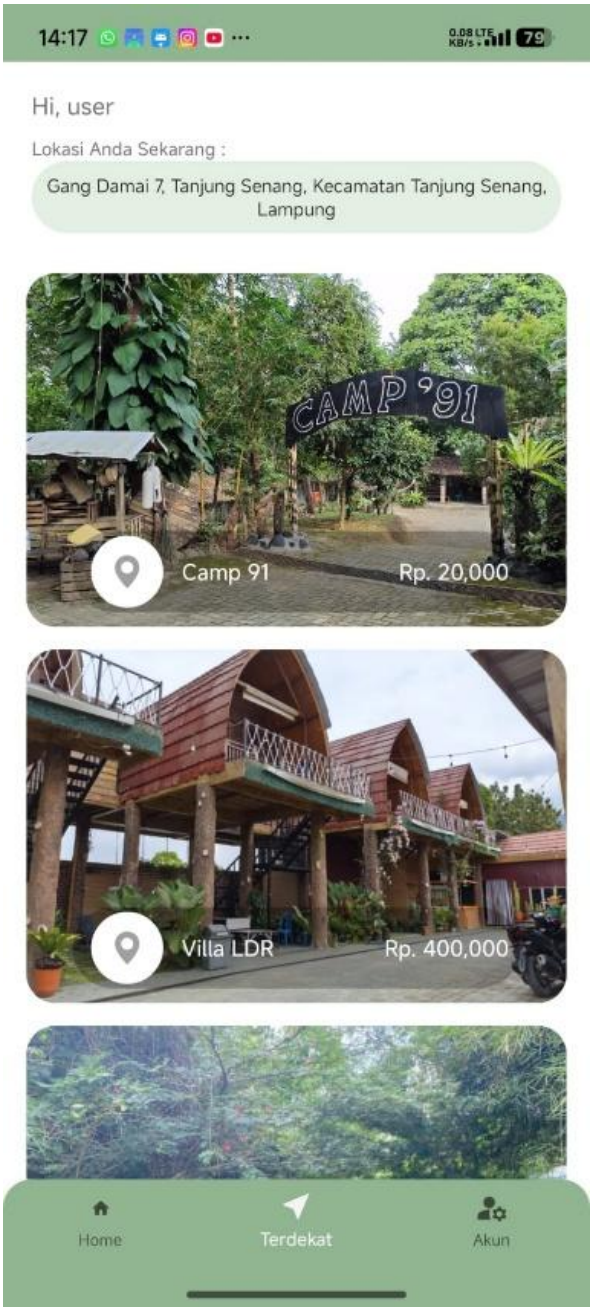


Gambar 4.51 Halaman Beranda

4.6.4 Halaman Terdekat

Implementasi halaman Terdekat sebagai halaman yang akan menampilkan produk atau destinasi wisata terdekat dari tempat

pengguna secara real-time. Dengan menggunakan kalkulasi algoritma Dijkstra sebagai dasar jarak rute terpendek.



Gambar 4.52 Halaman Terdekat

4.6.5 Halaman Akun

Implementasi halaman Akun yang yang dirancang untuk pengguna mengubah data profile mereka, menambahkan produk, atau mengecek

transaksi yang sudah di buat. Serta dapat mengkonfirmasi transaksi sebagai pelaku usaha.

Gambar 4.53 Halaman Akun

4.6.6 Halaman Produk Detail

Implementasi halaman Produk detail yang dirancang untuk memberikan informasi kepada pengguna setelah mereka mereka

memilih produk. Serta memiliki dua tombol untuk melakukan pemesanan dan melihat lokasi produk.

Gambar 4.54 Halaman Produk Detail

4.6.7 Halaman Pemesanan/Pembayaran

Implementasi halaman Pembayaran yang dirancang untuk memfatilitasi proses transaksi setelah pengguna memilih atau memesan tiket. Dengan

menyediakan pengunggah bukti pembayaran melalui tombol “Upload Bukti”. Setelah itu pengguna dapat melakukan konfirmasi pembayaran mereka dengan dua tombol terakhir dibawah.

Gambar 4.55 Halaman Pemesanan

4.6.8 Halaman Lokasi

Implementasi halaman Lokasi sebagai hasil integrasi sistem untuk menampilkan peta interaktif yang berpusat pada koordinat destinasi

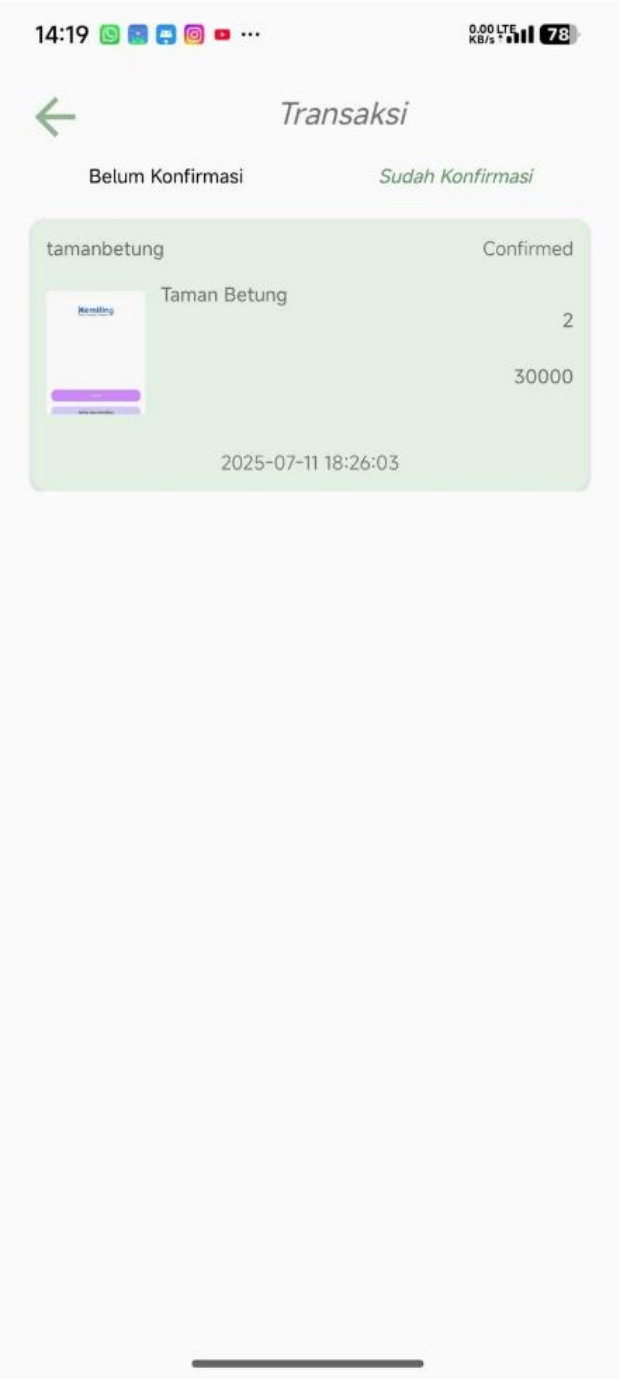
tujuan, yang ditandai dengan pin berwarna merah. Serta ada tombol untuk melihat menggunakan Aplikasi pihak ketiga dan tombol selesai untuk kembali ke halaman sebelumnya.

Gambar 4.55 Halaman Lokasi

4.6.9 Halaman Status Transaksi

Implementasi halaman status Transaksi yang dirancang agar pengguna dapat melihat status pesanan atau transaksi mereka. Implementasi ini

memberikan transparansi kepada pengguna mengenai alur pemesanan mereka.

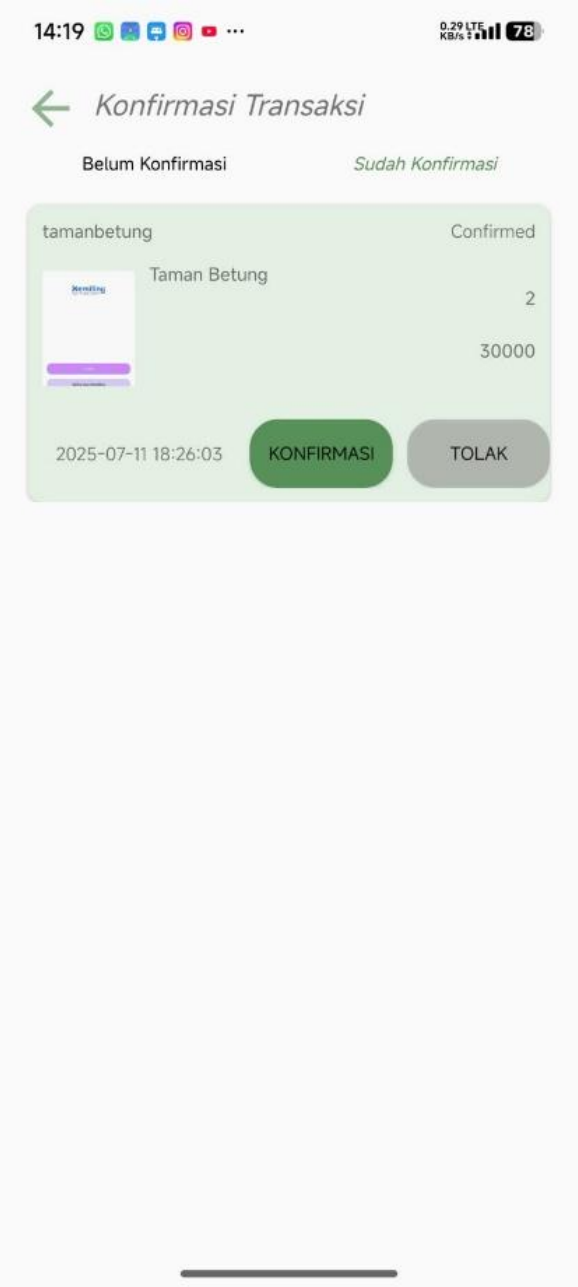


Gambar 4.56 Halaman Status Transaksi

4.6.10 Halaman Konfirmasi Transaksi

Implementasi halaman ini merupakan pusat pengelolaan pesanan yang masuk. Untuk mempermudah manajemen, halaman ini dilengkapi

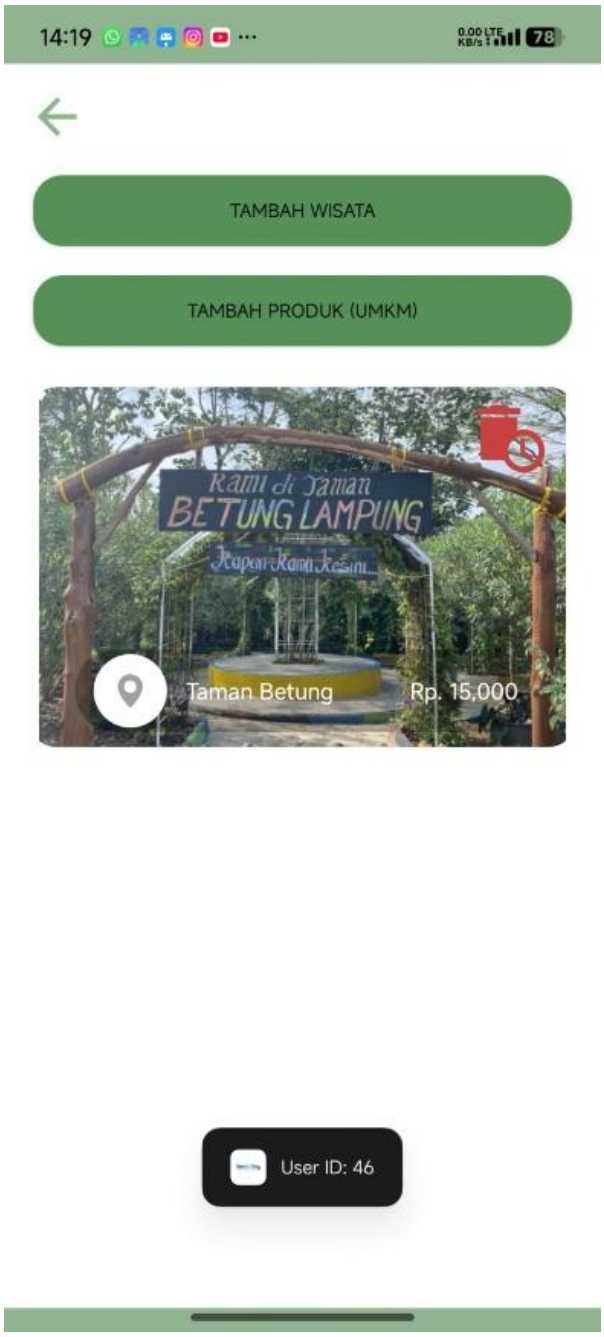
dengan sistem tab yang memisahkan transaksi berdasarkan statusnya, yaitu "Belum Konfirmasi" dan "Sudah Konfirmasi". Fungsi inti dari halaman ini adalah tombol aksi "Konfirmasi" dan "Tolak", yang memungkinkan Pelaku Usaha untuk menyetujui atau menolak pesanan secara langsung, sehingga alur validasi transaksi dapat berjalan dengan efisien.



Gambar 4.57 Halaman Konfirmasi Transaksi

4.6.11 Halaman Produk Saya

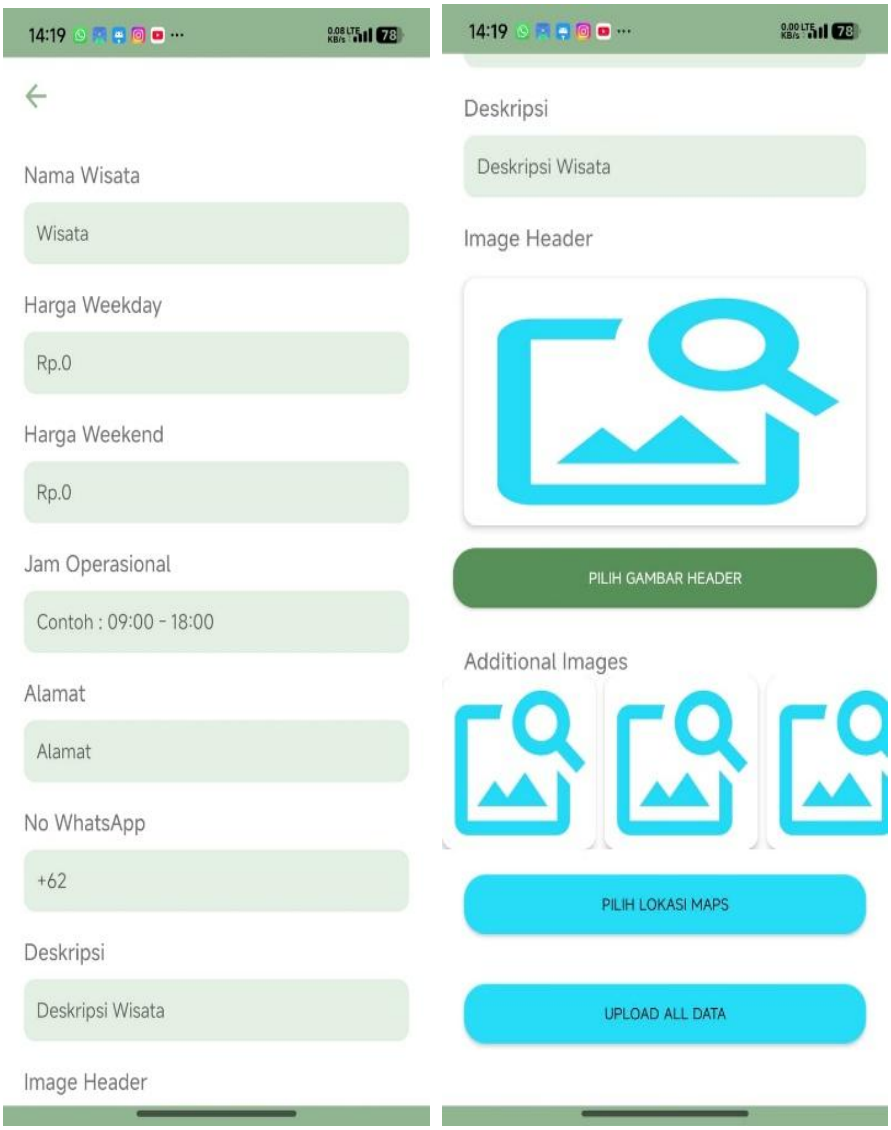
Implementasi halaman ini dirancang sebagai dasbor bagi pelaku usaha untuk mengelola konten mereka. Dibagian atas terdapat dua tombol yang dapat menambahkan produk konten baru, baik Wisata maupun produk UMKM. Dan memungkinkan pengguna untuk menghapus data produk mereka.



Gambar 4.58 Halaman Produk Saya

4.6.12 Halaman Tambah Produk

Implementasi halaman ini dirancang sebagai sebuah formulir untuk pelaku usaha bisa mendaftarkan destinasi wisata atau produk umkm baru. Dengan mencakup kolom input yang terstruktur dan beberapa tombol untuk pilih titik lokasi dan menyimpan data produk ke *Database.*

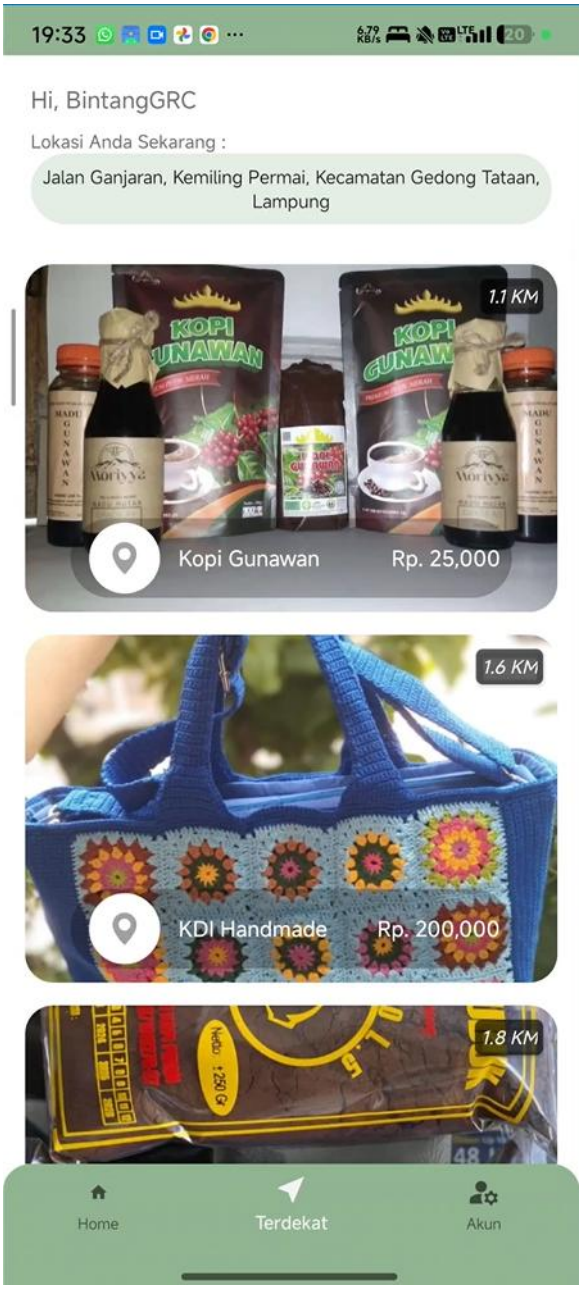


Gambar 4.59 Halaman Tambah Produk

4.6.13 Halaman Hasil Algoritma Dijkstra

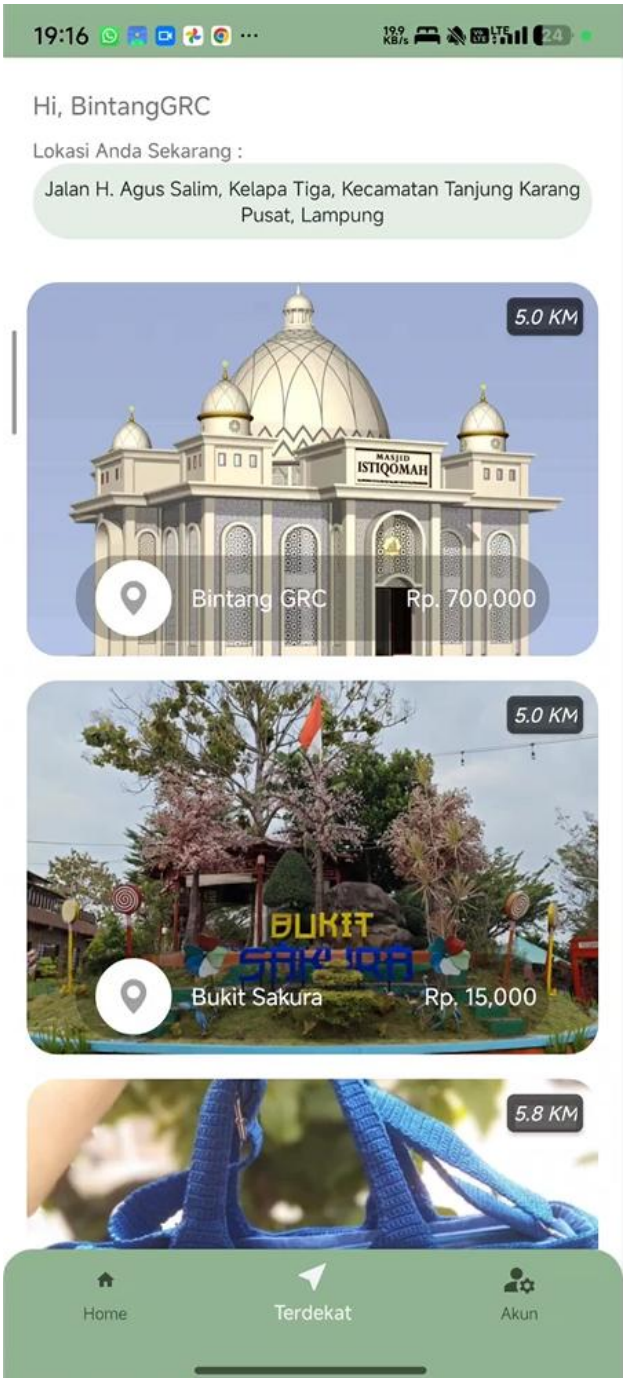
Berdasarkan implementasi Algoritma Dijkstra pada kode PHP yang sudah di terapkan sebelumnya, sistem berhasil menampilkan

rekomendasi terdekat dari lokasi pengguna secara *real-time,* dan berikut ini merupakan beberapa contoh rekomendasi tujuan dari berbagai tempat.



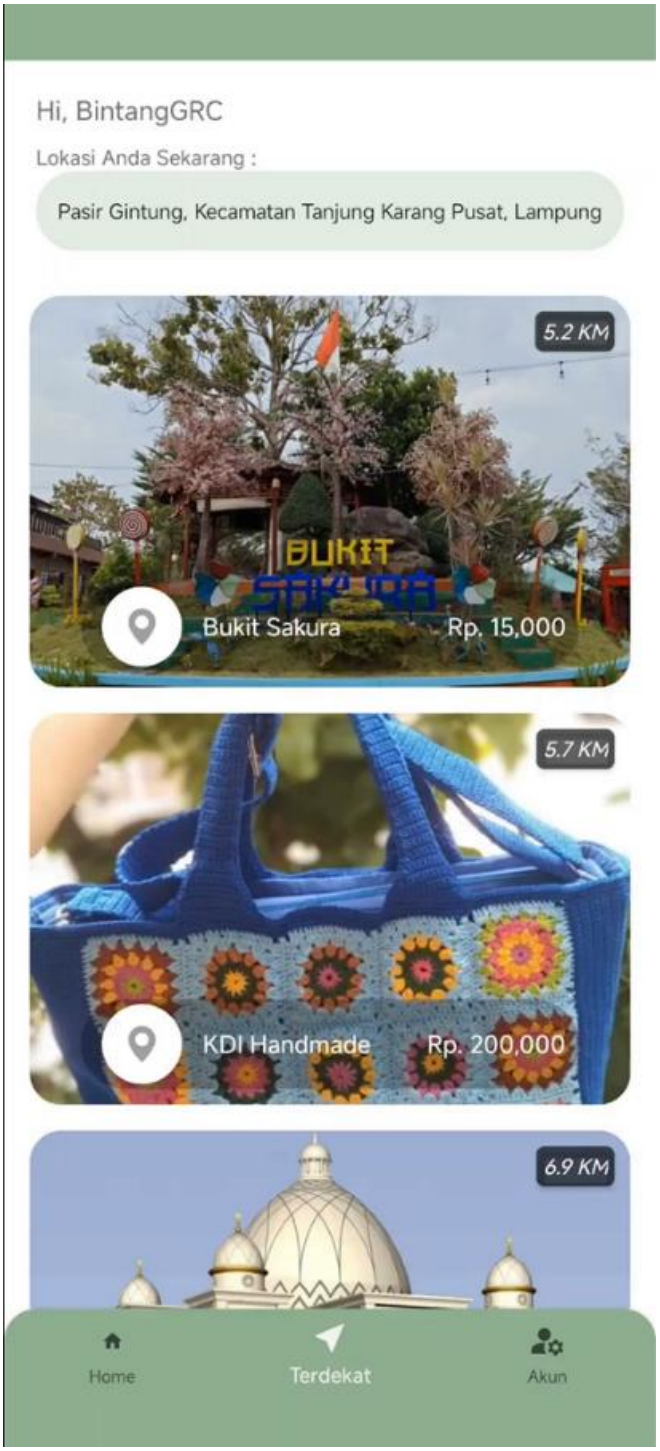
Gambar 4.60 Lokasi Jalan Ganjaran Kemiling

Dari lokasi ini, algoritma merekomendasikan Kopi Gunawan (1.1 KM) sebagai destinasi terdekat, diikuti oleh KDI Handmade (1.6 KM), menunjukkan hasil pengurutan jarak yang benar.



Gambar 4.62 Lokasi Jalan H. Agus Salim

Dari lokasi ini, algoritma dijkstra merekomendasikan Bintang GRC dan Bukit Sakura sebagai destinasi terdekat.



Gambar 4.63 Lokasi Pasar Gintung

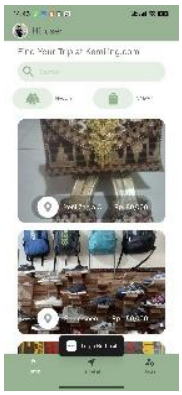
Dan dari lokasi ini, algoritma dikstra merekomendasikan Bukit Sakura sebagai destinasi terdekat yang dapat di kunjungi pengguna berdasarkan lokasi pengguna saat itu.

4.7 Pengujian Sistem

Berikut ini adalah tabel pengujian sistem yang sudah berjalan, tabel ini mencakup scenario pengujian dengan hasil yang di harapkan untuk memastikan setiap fungsi berjalan dengan baik sesuai rancangan.

Tabel 4.7 Pengujian Sistem

Skenario Pengujian	Hasil Pengujian dan Hasil yang Diharapkan		Kesimpulan
Registrasi Pengguna Baru		Deskripsi : Berhasil menampilkan halaman Register sesuai yang di harapkan	Sesuai dengan yang diharapkan
Login Pengguna		Deskripsi : Berhasil menampilkan halaman Login sesuai yang di harapkan	Sesuai dengan yang diharapkan
Login dengan Data Salah		Deskripsi : Berhasil menampilkan halaman Login dengan data yang salah dan sesuai yang di harapkan	Sesuai dengan yang diharapkan

Melihat Daftar Produk		Deskripsi : Berhasil menampilkan halaman Register sesuai yang di harapkan	Sesuai dengan yang diharapkan
Melihat Algoritma Bekerja		Deskripsi : Berhasil menampilkan halaman Terdekat tempat algoritma bekerja sesuai yang di harapkan	Sesuai dengan yang diharapkan
Melihat Profil Pengguna		Deskripsi : Berhasil menampilkan halaman Profil sesuai yang di harapkan	Sesuai dengan yang diharapkan

4.8 Distribusi Aplikasi

Untuk mempermudah proses penyebaran aplikasi, berkas instalasi diunggah ke layanan penyimpanan awan Google drive, Sehingga pengguna dapat mengunduh dan menggunakan aplikasi. Aplikasi dapat di download pada link berikut.

Link : https://drive.google.com/drive/folders/1WanW7MtLZ5NuVY-eODb3iewMW_i9zy2S?usp=sharing

atau di scan pada barcode yang mengarahkan ke tempat aplikasi di upload dibawah ini.

Gambar 4.64 Barcode Aplikasi