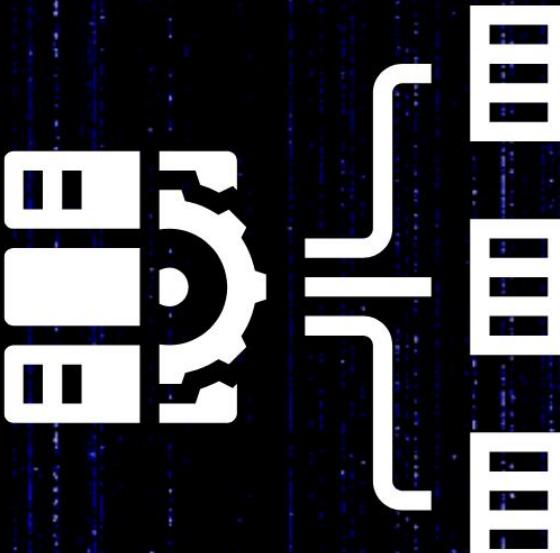


20 PROYEK APLIKASI PERANCANGAN & IMPLEMENTASI DATABASE

Laporan Proyek Besar
Mata Kuliah Basis Data Praktik



Program Studi Informatika
Universitas Teknologi Yogyakarta
2024

KATA PENGANTAR

Big thanks for seven, eight, and ten of database practice class.
PROUD OF YOU.

Muhammad Zakariyah, M.Kom.

DAFTAR ISI

SAMPUL.....	i
KATA PENGANTAR	ii
DAFTAR ISI.....	iii
Madya Regulation Culture (MRC).....	1
Pemesanan dan Info Wahana	15
Dashboard penjualan dan data toko Maestro Copies	27
Aplikasi Penjualan Konter Istar Cell (Icell)	46
Aplikasi Reservasi Restoran.....	74
E-Library Grhatama	87
Management Omset	103
SIMUN: Aplikasi Penjualan, Pembelian dan Konsultasi Produk SidoMuncul	116
Perancangan Sistem Aplikasi Toko Pancing dan Aksesoris.....	128
LaundryHUb: Solusi Modern Untuk Management Laundry	139
Visualisasi Menu Cafe	157
Perancangan Sistem Manejemen Laundry Berbasis Digital	165
Sistem Aplikasi Coffee Shop Berbasis Web.....	178
Sistem Informasi Manajemen Operasional Rumah Makan Nasi Balap Cucun.....	188
Hana Chicks 100% Restaurant	199
Wings Shoes (WS)	209
Manajemen pembelian tiket berbasis Web untuk PO Bus Purba Putra	222
Perancangan Sistem Ticketing Laundry Berbasis Aplikasi.....	253
SPHYNX CARE: Sistem Pemesanan Kamar Cat Hotel Online berbasis Aplikasi Mobile	269
Manajemen Inventaris dan Pemantauan Pendapatan Spare part Motor berbasis Web	303

Judul Aplikasi	
MADYA REGULATION CULTURE (MRC)	
Dikembangkan Oleh	
<ul style="list-style-type: none"> • 5220411253 – Berliandy putri Hasanuddin • 5220411258 – Faiz saputra • 5220411264 – Falah ilhan azmi • 5220411277 – Ahmad Dyan Hakiki 	
Deskripsi Singkat Instansi, Permasalahan, dan Solusi Aplikasi Yang Akan Dikembangkan [10 poin] <p>"Madya Regulation Culture" merupakan aplikasi inovatif yang dirancang khusus untuk membantu pengelola dan penyewa tempat yang ada di wilayah kota Jogja. Aplikasi ini mengatur perizinan, pengelolaan persewaan tempat, serta memastikan aturan yang berlaku di taati dengan baik. Dengan tampilan yang menarik dan fitur yang lengkap juga informatif, aplikasi ini menjadi solusi tepat untuk mengoptimalkan proses perizinan dan pengelolaan tempat dengan efisien.</p> <p>Fitur utama yang terdapat pada "Madya Regulation Culture" meliputi:</p> <ol style="list-style-type: none"> 1. Perizinan Terpusat: Pengguna akan mendapatkan akses mudah terhadap informasi perizinan yang diperlukan di wilayah kota Jogja. Pengguna dapat dengan cepat memperoleh informasi tentang peraturan, persyaratan, dan prosedur perizinan yang berlaku. 2. Pengelolaan Persewaan: Pengelola tempat dapat dengan mudah mengatur dan memantau proses persewaan tempat melalui aplikasi ini. Mereka dapat membuat jadwal ketersediaan tempat, mengelola pemesanan, dan mengatur harga sewa secara efisien. 3. Pemantauan Kepatuhan: Pengguna di berikan fitur pemantauan yang memungkinkan pengguna untuk memastikan kepatuhan terhadap regulasi perizinan dan aturan pengelolaan tempat. Fitur ini juga termasuk pemberitahuan otomatis tentang tenggat waktu perizinan yang akan datang dan pengingat tentang kewajiban administrasi. 4. Notifikasi dan Pemberitahuan: Pengguna akan menerima notifikasi real-time tentang perubahan peraturan perizinan, pemesanan baru, atau perubahan status persewaan tempat. <p>Dengan aplikasi "Madya Regulation Culture", pengelola dan penyewa tempat di wilayah kota Jogja dapat mengelola semua aspek perizinan dan persewaan tempat secara efektif dan efisien, meningkatkan transparansi, dan memastikan kepatuhan terhadap aturan yang berlaku.</p>	

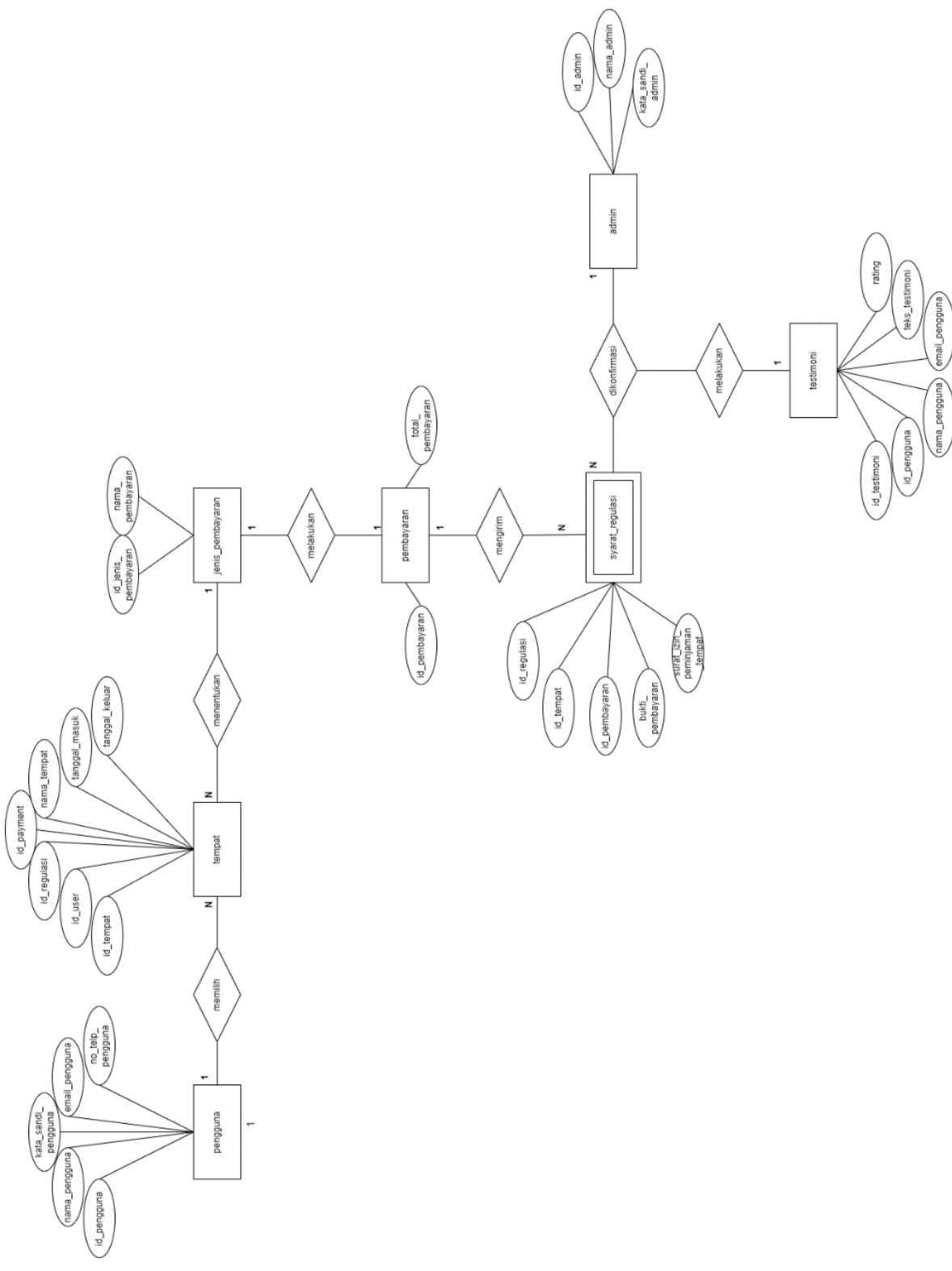
Pengguna Aplikasi [5 poin]

Target pengguna untuk web "Madya Regulation Culture" meliputi:

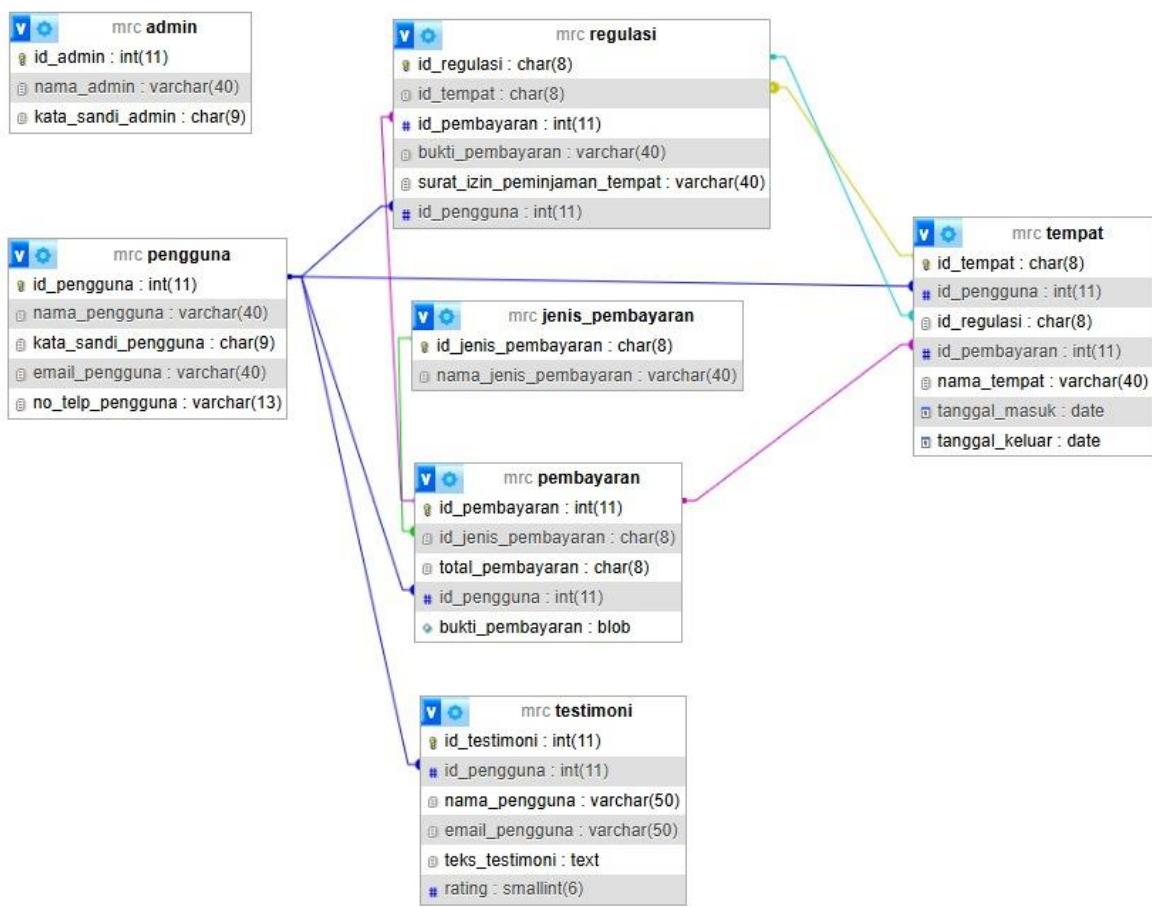
1. Pengelola Tempat: Pemilik properti, manajer acara, atau pengelola pusat kebudayaan yang membutuhkan akses mudah ke informasi perizinan, pengelolaan tempat, dan ketersediaan tempat.
2. Penyewa Tempat: Individu atau organisasi yang ingin menyewa tempat untuk acara atau kegiatan di wilayah kota Jogja. Mereka akan menggunakan aplikasi untuk menyewa, memesan, dan mengatur tempat yang sesuai dengan kebutuhan.
3. Pemerintah Daerah: Instansi pemerintah setempat, seperti dinas perizinan atau departemen kebudayaan, yang bertanggung jawab atas regulasi perizinan dan kegiatan di kota Jogja. Mereka dapat menggunakan aplikasi untuk memantau dan mengelola proses perizinan serta memastikan aturan di patuhi dengan baik.
4. Pengguna Umum: kelompok yang akan menyelenggarakan kegiatan kebudayaan dan akan melakukan penyewaan tempat di wilayah kota Jogja juga dapat menjadi pengguna aplikasi untuk mencari informasi tentang tempat yang tersedia, aturan perizinan, dan biaya sewa.

Dengan beberapa target pengguna ini, aplikasi dapat menjadi alat bantu yang komprehensif dan berguna dalam mengelola dan memfasilitasi proses perizinan dan penyewaan tempat di wilayah kota Jogja.

Desain Entity-Relationship Diagram (ERD) [10 poin]



Desain Relasi Basis Data [10 poin]



Query Data Definition Language (DDL) [25 poin]

```
CREATE DATABASE MRC

CREATE TABLE admin (
    id_admin INT AUTO_INCREMENT PRIMARY KEY,
    nama_admin VARCHAR(40),
    kata_sandi_admin CHAR(9)
);

CREATE TABLE pengguna (
    id_pengguna INT AUTO_INCREMENT PRIMARY KEY,
    nama_pengguna VARCHAR(40),
    kata_sandi_pengguna CHAR(9),
    email_pengguna VARCHAR(40),
    no_telp_pengguna VARCHAR(13)
);
```

```
CREATE TABLE tempat (
    id_tempat CHAR(8) PRIMARY KEY,
    id_pengguna INT,
    id_regulasi CHAR(8),
    id_pembayaran INT,
    nama_tempat VARCHAR(40),
    tanggal_masuk DATE,
    tanggal_keluar DATE,
    FOREIGN KEY (id_pengguna) REFERENCES pengguna(id_pengguna),
    FOREIGN KEY (id_regulasi) REFERENCES regulasi(id_regulasi),
    FOREIGN KEY (id_pembayaran) REFERENCES pembayaran(id_pembayaran)
);

CREATE TABLE pembayaran (
    id_pembayaran INT AUTO_INCREMENT PRIMARY KEY,
    id_jenis_pembayaran CHAR(8),
    id_pengguna INT,
    total_pembayaran CHAR(8),
    bukti_pembayaran BLOB,
    FOREIGN KEY (id_jenis_pembayaran) REFERENCES jenis_pembayaran
    REFERENCES jenis_pembayaran(id_jenis_pembayaran)
    FOREIGN KEY (id_pengguna) REFERENCES pengguna(id_pegguna)
);

CREATE TABLE jenis_pembayaran (
    id_jenis_pembayaran CHAR(8) PRIMARY KEY,
    nama_jenis_pembayaran VARCHAR(40)
);

CREATE TABLE regulasi (
    id_regulasi CHAR(8) PRIMARY KEY,
    id_tempat CHAR(8),
```

```
    id_pengguna INT,  
    id_pembayaran INT,  
    bukti_pembayaran VARCHAR(40),  
    surat_izin_peminjaman_tempat BLOB,  
    FOREIGN KEY (id_tempat) REFERENCES tempat(id_tempat),  
    FOREIGN KEY (id_pengguna) REFERENCES pengguna(id_pengguna),  
    FOREIGN KEY (id_pembayaran) REFERENCES pembayaran(id_pembayaran)  
);
```

```
CREATE TABLE testimoni(  
    id_testimoni INT PRIMARY KEY AUTO_INCREMENT,  
    id_pengguna INT,  
    nama_pengguna VARCHAR(50) NOT NULL,  
    email_pengguna VARCHAR(50) NOT NULL,  
    teks_testimoni TEXT NOT NULL,  
    rating SMALLINT NOT NULL CHECK (rating >= 1 AND rating <=5),  
    FOREIGN KEY (id_pengguna) REFERENCES pengguna(id_pengguna)  
);
```

Desain Interface (Wireframe/Mockup) Beserta Query-nya [poin 40]

1. Beranda



2. About pada website yang dimana menjelaskan bagaimana kebudayaan DIY

DAERAH ISTIMEWA YOGYAKARTA

Ketika Anda mendengar nama Yogyakarta, Anda akan berpikir tentang keunikan kota di Jawa di Indonesia ini. Kota ini disebut unik karena kita bisa melihat berbagai tradisi Jawa yang begitu melekat di kota ini dan masyarakatnya, seperti batik (kain dengan corak khusus yang dibuat dengan tangan (dilukis dan dicetak), kerajinan perak, pertunjukan wayang, musik tradisional atau gamelan, bahkan makanan khas Yogyakarta yang disebut Gudeg yaitu sayuran yang terbuat dari buah nangka muda.

Yogyakarta juga merupakan Ibu kota Daerah Istimewa Yogyakarta di Jawa yang terkenal dengan Istana Sultan Yogyakarta atau Keraton Yogyakarta dimana keluarga Sultan masih tinggal di istana tersebut sampai sekarang . Istana ini juga terbuka untuk umum tetapi tidak semua bangunan, hanya beberapa bagian bangunan yang bisa dilihat oleh masyarakat.

3. Pilihan Tempat

TEMPAT

Taman Budaya Embung Giwangon	Teras Maliboro 2	Taman Pintar	UPT Maliboro
Seluas 3,6 hektar memiliki embung, Jogging Track, Gedung Entrance dirancang dengan gaya Indische, dilengkapi fasilitas ruang rapat dan mini gallery. Fasilitas Terdiri dari 100 unit penginapan dan car park, dan area parkir: 108 unit motor, 50 unit mobil, dan 4 unit bus.	Pengunjung bisa belanja dengan tempat yang lebih rapi, bersih dan banyak bangunan fasilitasnya. Area ini pada saat malam digunakan sebagai tempat yang nyaman untuk jalan-jalan dapat merasakan makanan khas Yogyakarta. Terdapat banyak spot foto dan juga fasilitas playground.	Tempat rekreasi edukasi di Jogja, terdapat planetarium, wahana bahan, zona pengolahan sampah, science theater, playground, pertemuan, konsert, karnaval, gelaran seni dan jika anda ingin membuat event seperti event tarl, wayang dll, maka pihak taman pintar juga bersedia.	Menunjang operasional Pariwisata dan Kebudayaan Maliboro Yogyakarta. Mempunyai tanggungjawab dalam pengelolaan destinasi, keterbukaan, kain halus, usaha pembenihan, penanaman kawasan parkir dan transportasi yang berada di Kawasan Maliboro.
Booking	Booking	Booking	Booking

4. Testimoni

The screenshot shows a testimonial section with three reviews. Each review includes a user profile picture, the user's name, their Instagram handle, and a 5-star rating followed by a testimonial text.

User	Instagram Handle	Rating	Testimonial
Niall Horan	@naillhorann	★★★★★	Saya sangat terkesan dengan layanan yang diberikan oleh website ini. Namanya mudah, informasi lengkap, dan respons dari admin sangat cepat dan membantu. Pengalaman membooking tempat untuk membuat event sangat baik. Sangat direkomendasikan!
Harry Styles	@harrystyles	★★★★★	Website ini luar biasa dan benar-benar membandingkan proses perizinan milik saya ketika menggunakan website lainnya. Dapat direspon dengan cepat dan admin juga sangat informatif. Saya pasti akan menggunakan website ini lagi untuk event event yang akan saya adakan selanjutnya. Rating 5 bintang dari saya
Zayn Malik	@zaynmailkk	★★★★★	Pengalaman terbaik yang pernah saya alami saat menggunakan website ini. Antarmukanya intuitif, dan proses perizinan tempat dan izin yang diberikan sangat mudah. Tempat dan keleluasaan kami terpenuhi dengan baik, dan acara kami berjalan dengan sangat lancar. Lokasinya juga strategis dan fasilitasnya lengkap. Terima kasih banyak atas bantuanmu!

5. Kontak

The screenshot shows a contact page with a dark purple header. It features the MRC logo, navigation links, and a central section for company and service details. Below this is a footer with social media links and a copyright notice.

Company	Service	Follow Us
About	Taman budaya Embung Giwangan	Instagram
Contact	Teras Malioboro 2	Facebook
	Taman Pintar	TikTok
	UPT Malioboro	YouTube

direct by : @madyaculture | All Reserved

6. Profil

The screenshot shows a user profile page titled "My Profil". It displays the user's ID (id_user) and three input fields for "email", "name", and "no telp". There are also "Create Account" and "Login" buttons. On the left, there are links for "My Profil" and "My Booking".

```
SELECT id_pengguna, nama_pengguna, email_pengguna, no_telp_pengguna  
FROM pengguna;
```

7. Registrasi/ membuat akun

Create Account

Email

No telefon

Username

Password

Already have an account ? [Login](#)

[Create](#)

```
INSERT INTO pengguna (email_pengguna, no_telp_pengguna, nama_pengguna,  
kata_sandi_pengguna)
```

8. Login

Login

Welcome Back! please login to access the madya regulation culture
Did You [Forgot Your Password?](#)

Username

Password

Don't have an account ? guest to [Registrasi](#)

[Login](#)

```
SELECT * FROM pengguna  
WHERE nama_pengguna = 'username'  
AND kata_sandi_pengguna = 'password';
```

9. Pilihan Tempat

The screenshot shows the 'TEMPAT' section of the MRC website. It features four cards, each with a thumbnail image and a 'Booking' button:

- Taman Budaya Embung Giwangon**: A 3.6-hectare embung with a Jogging Track and Indische-style building.
- Teras Malioboro 2**: An area for shopping and dining, featuring a food court and playground.
- Taman Pintar**: An educational and recreational space with a planetarium and various attractions.
- UPT Malioboro**: A place for cultural events, including traditional performances.

```
SELECT id_tempat, id_pengguna, nama_tempat FROM tempat;
```

10. Memilih Tempat dengan menyesuaikan tanggal

The screenshot shows the 'TEMPAT' section with the 'Teras Malioboro 2' card selected. A calendar overlay is displayed, showing the month of May 2024. The date May 18 is highlighted in blue, indicating it is the chosen date for the booking.

```
SELECT id_tempat, id_pengguna, nama_tempat, tanggal_masuk, tanggal_keluar  
FROM tempat;
```

11. Regulasi

The screenshot shows a website interface with a navigation bar at the top: BERANDA, KEBUDAYAAN, PEMESANAN, REGULASI (highlighted in pink), MY BOOKING, and PROFIL. A circular logo for 'MRC Madya Regulation Culture' is on the left. The main content area has a title 'Complete the order' and a photo of a building at night with a sign that says 'Taman Budaya'. To the right, there's a form titled 'Taman Budaya Embung Giwangan' with fields for 'Bukti Pembayaran*' (with a file upload button) and 'Surat Izin Peminjaman Tempat*' (with a file upload button). There's also a 'Kosongkan formulir' link and a green 'Kirim' button.

INSERT INTO pembayaran (bukti_pembayaran, surat_izin_peminjaman_tempat)

12. Proses Waiting (untuk tampilan pada pengguna)

The screenshot shows a website interface with a navigation bar at the top: BERANDA, KEBUDAYAAN, PEMESANAN (highlighted in pink), REGULASI, MY BOOKING, and PROFIL. A circular logo for 'MRC Madya Regulation Culture' is on the left. The main content area has a title 'Cheking File' and a photo of the same building at night. To the right, there's a message 'Taman Budaya Embung Giwangan' with the note 'Yogyakarta | 18 may 2024' and a status message 'Menunggu konfirmasi' above a green 'waiting...' button.

13. Tampilan pada admin setelah pengguna mengirimkan dokumen

The screenshot shows the MRC Admin Dashboard with the title "Madya Regulation Culture". On the right, there is a profile picture of "admin". Below the title, there are four summary cards:

- Total Pesanan:** 27 (More Info →)
- Pesanan Pending:** 7 (More Info →)
- Pesanan Approved:** 4 (More Info →)
- Pesanan Ditolak:** 9 (More Info →)

Below these cards, there are three tables of document submissions:

- Pesanan Pending:** One entry by Niall Horan, status Pending, dated 18 Mei 2024, at Taman Budaya Embung Giwangan.
- Pesanan Approved:** One entry by Niall Horan, status Approved, dated 18 Mei 2024, at Taman Budaya Embung Giwangan.
- Pesanan di Tolak:** One entry by Niall Horan, status Failed, dated 18 Mei 2024, at Taman Budaya Embung Giwangan.

14. Tampilan pada admin untuk mengkonfirmasi berkas pengguna

The screenshot shows the MRC Admin Dashboard with the title "Madya Regulation Culture". On the right, there is a profile picture of "admin". Below the title, there are four summary cards:

- Total Pesanan:** 27 (More Info →)
- Pesanan Pending:** 7 (More Info →)
- Pesanan Approved:** 4 (More Info →)
- Pesanan Ditolak:** 9 (More Info →)

Below these cards, there are two sections for file attachments:

- Niall Horan:** Includes a checked checkbox icon and a red circle icon.
- Bukti pembayaran:** A blue document icon labeled "DOCUMENT".
- Surat Izin Meminjam Tempat:** A blue document icon labeled "DOCUMENT".

15. Tampilan pengguna jika admin menolak berkas

The screenshot shows the "Cheking File" page. At the top, there is a navigation bar with links: BERANDA, KEBUDAYAAN, **PEMESANAN**, REGULASI, MY BOOKING, and PROFIL. On the left, there is a circular logo with "MRC". The main content area has the title "Cheking File". It displays a photograph of a building at night with a sign that says "Cafe Budaya". To the right of the photo, there is a pink box containing the following information:

Taman Budaya Embung Giwangan
Yogyakarta | 18 may 2024
*Silahkan periksa kembali dokumen yang anda lampirkan

A red button labeled "Failed" is visible.

16. Tampilan Pengguna jika admin mengacc

The screenshot shows a booking detail page for 'Taman Budaya Embung Giwangan' in Yogyakarta. At the top, there's a navigation bar with links: BERANDA, KEBUDAYAAN, PEMESANAN, REGULASI, MY BOOKING (which is highlighted in pink), and PROFIL. On the left, there's a sidebar with 'My Profil' and 'My Booking' options. The main content area features a photograph of a traditional building at night with a sign that reads 'Taman Budaya Embung Giwangan'. To the right of the photo, the venue name is listed along with its location ('Yogyakarta | 18 may 2024') and a note that the booking is completed ('Pesanan anda telah selesai'). Below that, the user's information is provided: 'Nama Pengguna' (nameyour@gmail.com), phone number ('+62 812 xxxx xxxx'), and the booking date ('18 may 2024'). A 'view map' button is also present. At the bottom of the page, there's a yellow button labeled 'Berikan Penilaian'.

17. Setelah pengguna menyelesaikan prosedur regulasi dan diberikan acc oleh admin maka pengguna diminta untuk memberikan penilaian/ testimoni

The screenshot shows a testimonial page. At the top, there's a navigation bar with links: BERANDA, KEBUDAYAAN, PEMESANAN, REGULASI, MY BOOKING, and PROFIL. The main title is 'Testimoni'. Below it, there are two rating sections: 'Informasi Website' with a 5-star rating icon and 'Pelayanan Admin' with a 5-star rating icon. There's also a checkbox labeled 'Tampilkan username pada penilaian'. To the right, there's a text input field with placeholder text 'Bagikan penilaianmu dan bantu Pengguna lain!' and a green 'Kirim' button.

```
INSERT INTO testimoni (id_pengguna, nama_pengguna, email_pengguna, teks_testimoni, rating)
```

18. tampilan testimoni setelah pengguna memberikan penilaian

The screenshot shows the 'Testimoni' (Testimonials) section of the MRC website. At the top, there is a navigation bar with links to BERANDA, KEBUDAYAAN, PEMESANAN, REGULASI, MY BOOKING, and PROFIL. On the left side, there is a circular logo for 'MRC' (Madya Regulation Culture) featuring a stylized torch. The main content area displays three testimonies in separate boxes:

- Nail Horan** (@nailhorann)
★★★★★
Saya sangat terkesan dengan layanan yang diberikan oleh website ini. Navigasinya mudah dan responsif. Dukungan tim admin sangat cepat dan membantu. Pengalaman membooking tempat untuk membuat event sangat baik. Sangat direkomendasikan!
- Harry Styles** (@harrystyless)
★★★★★
Website ini luar biasa dan benar-benar membantu! Proses perizinan saat saya membuat event di sini sangatlah cepat dan admin juga sangat informatif. Saya pasti akan menggunakan website ini lagi untuk event-event yang akan saya adakan selanjutnya. Rating 5 bintang dari saya
- Zayn Malik** (@zaynmalikk)
★★★★★
Pengalaman terbaik yang pernah saya alami saat menggunakan website ini. Antarmukanya intuitif, dan proses perizinan yang dilakukan yang ditanyakan sangat cepat dan efisien. Semua kebutuhan kami terpenuhi dengan baik, dan acara kami berjalan dengan sangat lancar. Lokasinya juga strategis dan fasilitasnya lengkap. Terima kasih banyak atas bantuanmu!

SELECT * FROM testimoni;

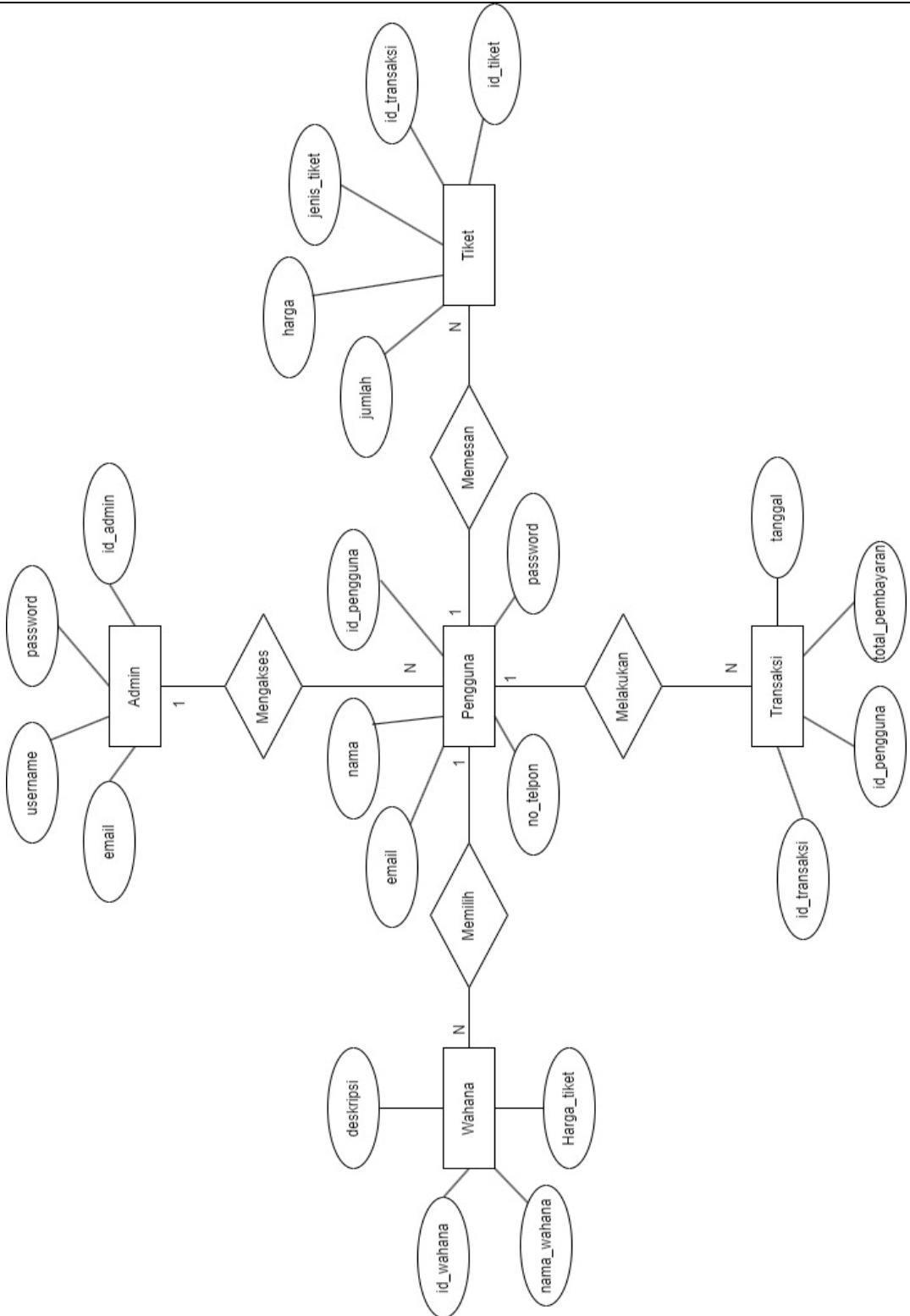
Judul Aplikasi <h2 style="text-align: center;">PEMESANAN DAN INFO WAHANA</h2>	
Dikembangkan Oleh <ul style="list-style-type: none"> 1. 5220411283_Reza Gemazih 2. 5220411278_Abhirama Garda N.H 3. 5220411276_Sirajudin 4. 5220411285_Silvester Stoneiman 5. 5220411271_Muhammad Alwin Prima 	
<p>Deskripsi Singkat Instansi, Permasalahan, dan Solusi Aplikasi Yang Akan Dikembangkan [10 poin]</p> <p>Sebuah taman hiburan atau wahana rekreasi yang Bernama Sindu Kusuma EduPark ingin meningkatkan pengalaman pelanggan dengan menyediakan layanan pemesanan tiket online dan informasi wahana yang lebih efisien.</p> <p>Permasalah yang mereka sering alami adalah banyak pengunjung yang mengeluh karna antrian panjang di lokasi, kurangnya informasi yang jelas tentang wahana, dan kurangnya kemudahan dalam proses pemesanan tiket yang mengakibatkan pengalaman pelanggan yang tidak optimal. Antrian panjang dapat mengganggu pengalaman pelanggan dan menyebabkan ketidaknyamanan, sedangkan kurangnya informasi tentang wahana dapat menyebabkan kebingungan bagi pengunjung dan mengurangi kepuasan mereka. Selain itu, jika proses pemesanan tiket tidak efisien, pelanggan mungkin menjadi frustrasi dan memilih untuk tidak mengunjungi taman hiburan tersebut.</p> <p>Solusi untuk permasalahan tersebut adalah pengembangan aplikasi pemesanan tiket dan informasi wahana yang efisien dan mudah digunakan. Aplikasi ini dapat menyediakan fitur-fitur seperti:</p> <ul style="list-style-type: none"> Pemesanan Tiket Online: Pelanggan dapat dengan mudah memesan tiket mereka melalui aplikasi, menghindari antrian panjang di lokasi, Informasi Wahana yang Lengkap: Aplikasi menyediakan deskripsi lengkap tentang setiap wahana, termasuk fitur, tingkat kesulitan, dan persyaratan umur/tinggi, Perbaruan Ketersediaan dan Waktu Tunggu: Pelanggan dapat melihat informasi real-time tentang ketersediaan tiket dan estimasi waktu tunggu untuk setiap wahana. Pengingat dan Notifikasi: Aplikasi dapat mengirimkan pengingat kepada pelanggan tentang jadwal perjalanan mereka dan memberikan notifikasi tentang penawaran khusus atau promosi yang sedang berlangsung. Feedback dan Evaluasi: Pelanggan dapat memberikan umpan balik tentang 	

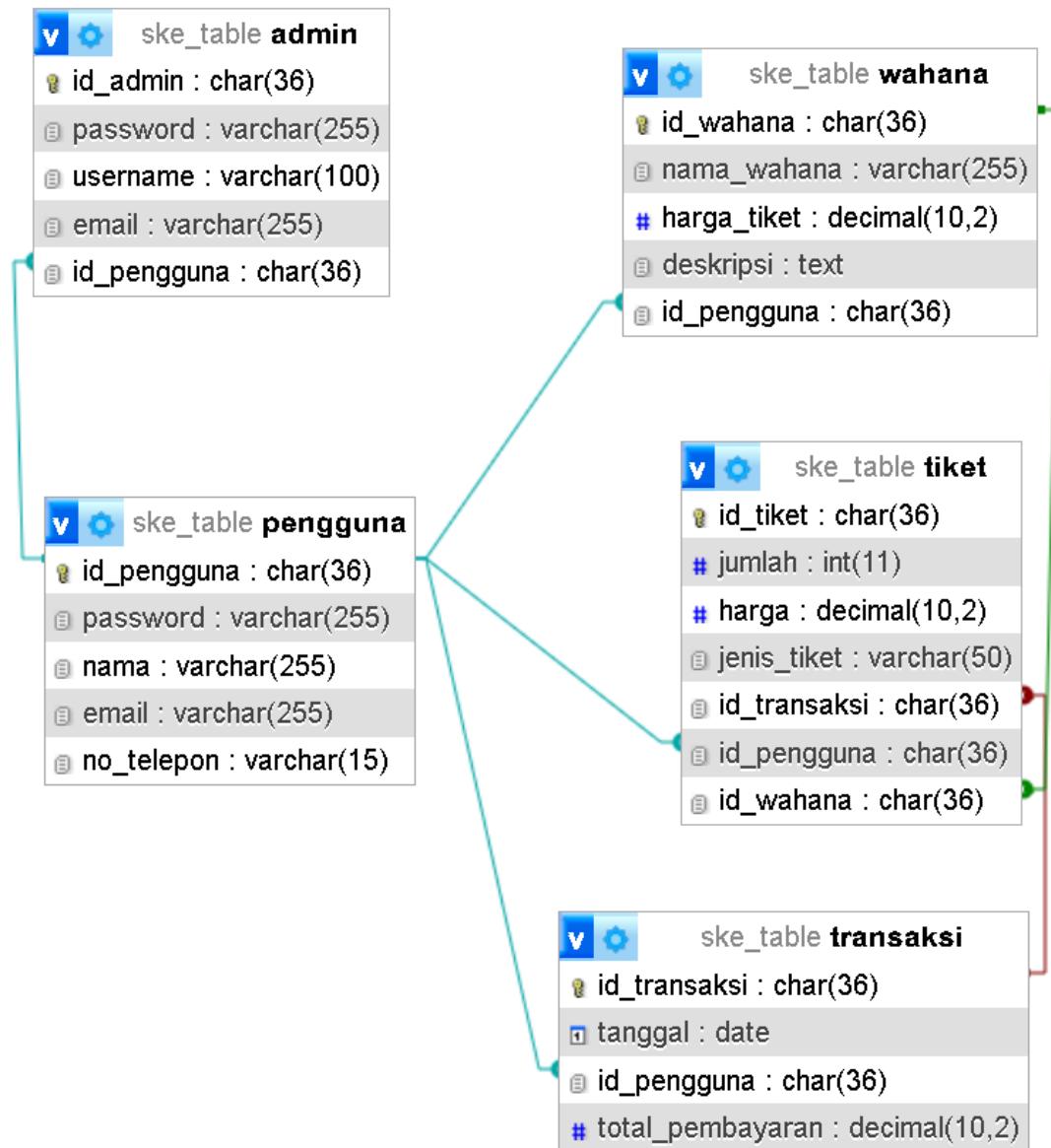
pengalaman mereka melalui aplikasi, yang dapat membantu taman hiburan untuk terus meningkatkan layanan mereka.

Pengguna Aplikasi [5 poin]

Pengunjung Wahana dan Admin Sindu Kusuma EduPark

Desain Entity-Relationship Diagram (ERD) [10 poin]



Desain Relasi Basis Data [10 poin]**Query Data Definition Language (DDL) [25 poin]**

```
CREATE TABLE Pengguna (
    id_pengguna INT(36) PRIMARY KEY AUTO_INCREMENT,
    password VARCHAR(255),
    nama VARCHAR(255),
    email VARCHAR(255),
    no_telepon VARCHAR(15)
);
```

```
CREATE TABLE Wahana (
    id_wahana INT(36) PRIMARY KEY AUTO_INCREMENT,
```

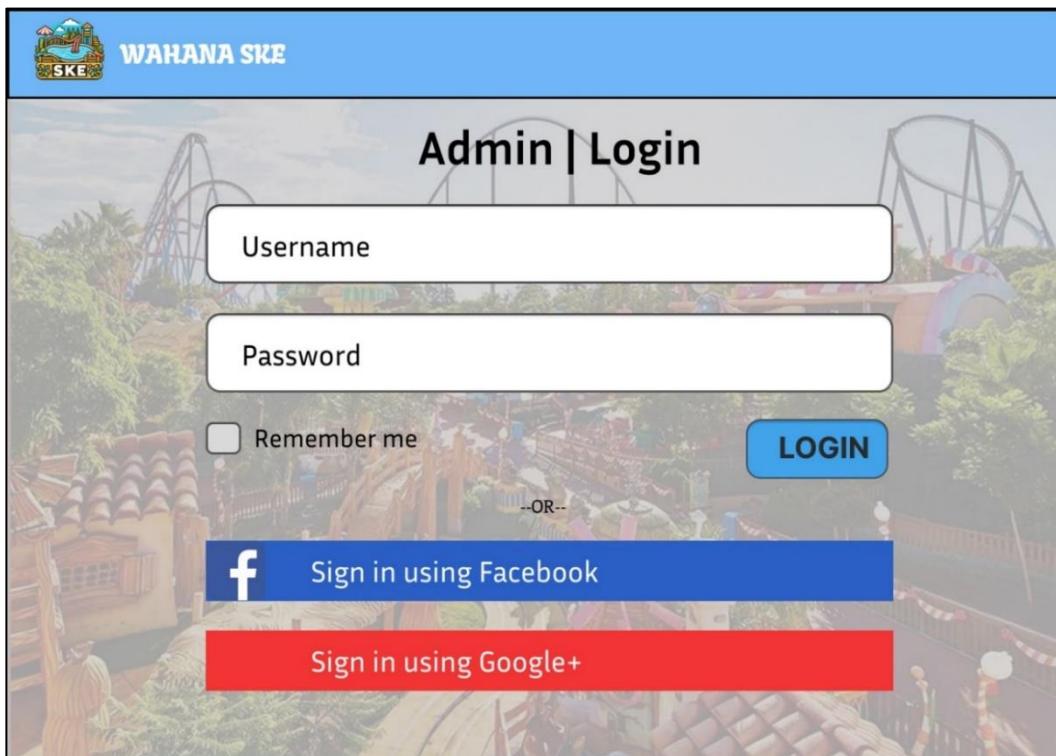
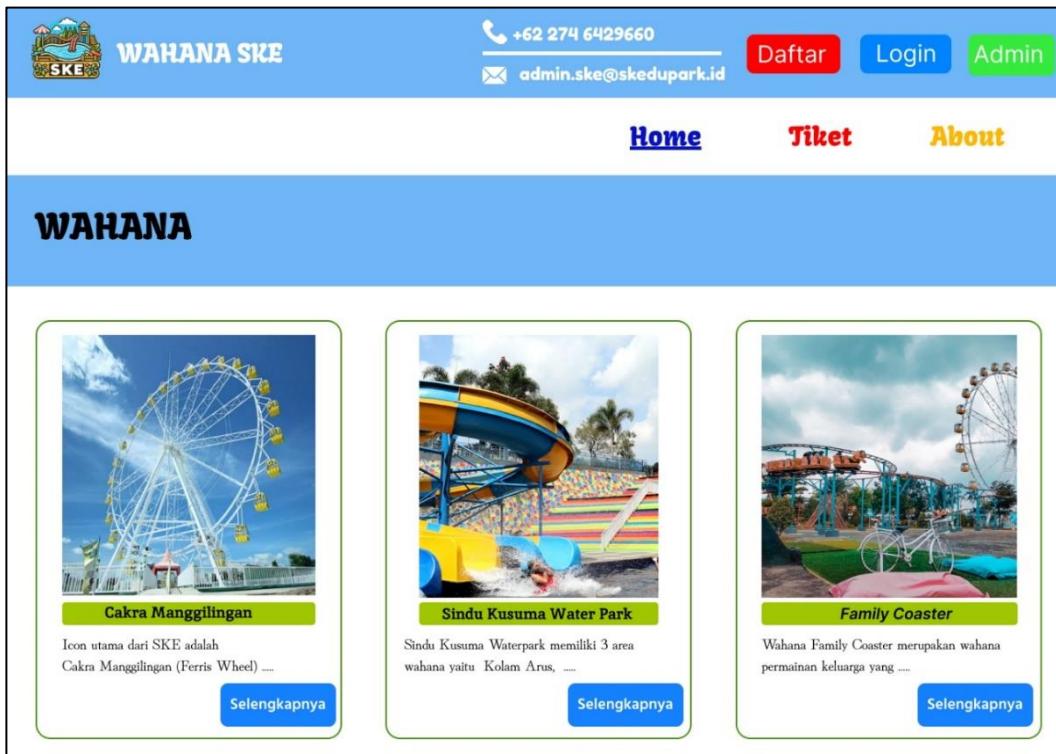
```
nama_wahana VARCHAR(255),
harga_tiket DECIMAL(10, 2),
deskripsi TEXT,
id_pengguna CHAR(36),
FOREIGN KEY (id_pengguna) REFERENCES Pengguna(id_pengguna)
);

CREATE TABLE Transaksi (
    id_transaksi INT(36) PRIMARY KEY AUTO_INCREMENT,
    tanggal DATE,
    id_pengguna CHAR(36),
    total_pembayaran DECIMAL(10, 2),
    FOREIGN KEY (id_pengguna) REFERENCES Pengguna(id_pengguna)
);

CREATE TABLE Tiket (
    id_tiket INT(36) PRIMARY KEY AUTO_INCREMENT,
    jumlah INT,
    harga DECIMAL(10, 2),
    jenis_tiket VARCHAR(50),
    id_transaksi CHAR(36),
    id_pengguna CHAR(36),
    id_wahana CHAR(36),
    FOREIGN KEY (id_transaksi) REFERENCES Transaksi(id_transaksi),
    FOREIGN KEY (id_pengguna) REFERENCES Pengguna(id_pengguna),
    FOREIGN KEY (id_wahana) REFERENCES Wahana(id_wahana)
);

CREATE TABLE Admin (
    id_admin INT(36) PRIMARY KEY AUTO_INCREMENT,
    password VARCHAR(255),
    username VARCHAR(100),
    email VARCHAR(255),
    id_pengguna CHAR(36),
    FOREIGN KEY (id_pengguna) REFERENCES Pengguna(id_pengguna)
);
```

Desain Interface (Wireframe/Mockup) Beserta Query-nya [poin 40]



```
INSERT INTO `admin` (`nama`, `kata_sandi`) VALUES ('admin1@gmail.com', 'admin123');
```

WAHANA SKE

Admin

Search....

- Home**
- Wahana**
- Admin**
- Pelanggan**
- Pembelian**
- Logout**

SELAMAT DATANG DI WAHANA SKE!

Kami sangat senang Anda memilih untuk bergabung dengan kami hari ini. Di sini, Anda akan menemukan petualangan yang tak terlupakan dan pengalaman yang penuh kegembiraan. Sebelum kita mulai, izinkan kami memberikan beberapa informasi penting untuk memastikan keselamatan dan kenyamanan Anda selama berada di wahana ini.

Sindu Kusuma Edupark (SKE)
4.2 ★★★★ (16.871)
Taman Hiburan
Tutup - Buka Sel pukul 09.00

WAHANA SKE

Admin

Search....

- Home**
- Wahana**
- Admin**
- Pelanggan**
- Pembelian**
- Logout**

[+]Tambah Wahana

Data Harga Tiket Masuk Wahana SKE

No.	Nama	Deskripsi	Harga Tiket	Foto	Aksi
1	Harga Tiket Masuk (HTM)		Rp10.000		Ubah Hapus
2	Sindu Kusuma Water Park	Sindu Kusuma Waterpark memiliki 3 area wahana yaitu Kolam Arus,.....	Rp15.000		Ubah Hapus
3	Family Coaster	Wahana Family Coaster merupakan wahana permainan keluarga yang	Rp15.000		Ubah Hapus
4	Cakra Manggilingan	Icon utama dari SKE adalah Cakra Manggilingan (Ferris Wheel)	Rp15.000		Ubah Hapus

```
INSERT INTO `wahana` (`id_wahana`, `nama_wahana`, `harga_tiket`, `deskripsi`) VALUES ('1', 'Harga Tiket Masuk (HTM)', '10.000', '-'), ('2', 'Sindu Kusuma Water Park', '15.000', 'Sindu Kusuma Waterpark memiliki 3 area wahana yaitu Kolam Arus,.....'), ('3', 'Family Coaster', '15.000', 'Wahana Family Coaster merupakan wahana permainan keluarga yang .....'), ('4', 'Cakra Manggilingan', '15.000', 'Icon utama dari SKE adalah Cakra Manggilingan (Ferris Wheel) .....
```

'Cakra Manggilingan', '15.000', 'Icon utama dari SKE adalah \r\nCakra Manggilingan (Ferris Wheel).....')

The screenshot shows the 'Data Admin' section of the Wahana SKE admin interface. On the left sidebar, there is a user icon labeled 'Admin' and a search bar labeled 'Search....'. Below the search bar are navigation links: Home, Wahana, Admin, Pelanggan, Pembelian, and Logout. The main content area is titled 'Data Admin' and contains a table with one row of data:

NO	Nama	Username	Password	Aksi
1	Admin1	Admin1	Admin123	<button>Ubah</button> <button>Hapus</button>

SELECT * FROM `admin`;

The screenshot shows the 'Data Pelanggan' section of the Wahana SKE admin interface. On the left sidebar, there is a user icon labeled 'Admin1' and a search bar labeled 'Search....'. Below the search bar are navigation links: Home, Wahana, Admin, Pelanggan, Pembelian, and Logout. The main content area is titled 'Data Pelanggan' and contains a table with two rows of data:

No.	Nama Pelanggan	Telpon Pelanggan	Email
1	Sumanto	08142653436	Sumntosikopat@gmail.com
2.	Jarjit	08952425262	jarjitsing21@gmail.com

SELECT * FROM `pengguna`;

No.	Nama Pelanggan	Tanggal Pemesanan	Status Pemesanan	Total	Aksi
1	Sumanto	12 mei 2024	Belum Dibayar	Rp50.000	<button>Ubah</button> <button>Hapus</button>

SELECT * FROM `transaksi`;

Halaman daftar pelanggan

Form fields for registering a new customer:

- Nama:
- Email:
- Password:
- Konfirmasi password:
- Alamat:
- No. telpon:

Daftar

INSERT INTO `pengguna` (`nama`, `email`, `password`, `no_telp`) VALUES ('Alwin', 'alwin@gmail.com', 'alwin123', '085253645354');

Daftar Disini'."/>

Pelanggan | Login

Email

Password

Remember me

Login

Belum Punya akun? [Daftar Disini](#)

```
INSERT INTO `pengguna` (`password`, `email`) VALUES
('alwin123','alwin@gmail.com');
```

WAHANA

Cakra Manggilingan

Sindu Kusuma Water Park

Family Coaster

Selengkapnya

Selengkapnya

Selengkapnya

```
SELECT * FROM `wahana`;
```

Pemesanan Tiket

No.	Nama tiket	Harga Tiket	Pesan	Jumlah
1	Harga Tiket Masuk (HTM)	Rp10.000	[+/-]	1x
2	Sindu Kusuma Water Park	Rp15.000	[+/-]	0x
3	Family Coaster	Rp15.000	[+/-]	1x
4	Cakra Manggilingan	Rp15.000	[+/-]	1x

Total: Rp40.000 **Pesan**

```
SELECT * FROM `tiket`;
```

Pilih metode pembayaran

E-Wallet gopay DANA OVO

ATM Bersama PRIMA ALTO

Total: Rp40.000 **Bayar**

```
INSERT INTO transaksi (id_transaksi, tanggal, id_pengguna, total_pembayaran)
VALUES ('1', '2024-05-26', '1', '40.000');
```



PEMESANAN ANDA BERHASIL!

Terima Kasih atas Pembayaran Anda!

*Pembayaran tiket Anda telah berhasil diproses.
selamat menikmati liburan anda.*

KEMBALI

The screenshot shows a blue header bar with the logo 'WAHANA SKE' and a user profile for 'Joko'. Below the header is a navigation menu with 'Home' (blue), 'Tiket' (red), and 'About' (yellow). The main content area features a banner for 'SINDU KUSUMA EDUPARK' with cartoon characters and the park's logo. The text 'SINDU KUSUMA EDUPARK' is prominently displayed below the banner.

Sindu Kusuma Edupark (SKE) selaku Perusahaan yang bergerak di bidang Wisata Themepark dan Edukasi di Yogyakarta, yaitu tempat wahana rekreasi keluarga berbasis Budaya & Edukasi.



WAHANA SKE

Joko
[Log Out](#)



Joko

Nama

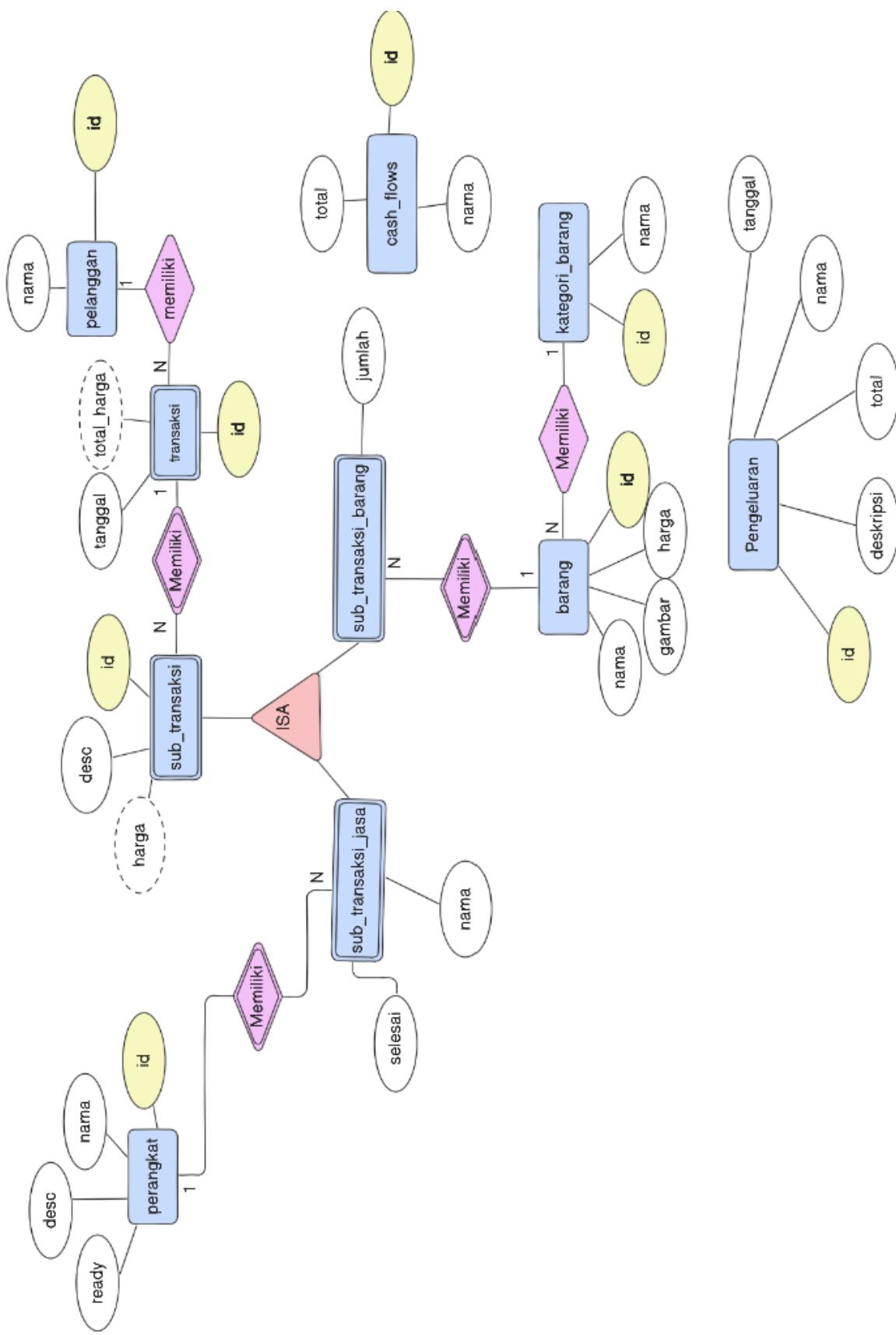
Email

No telpo

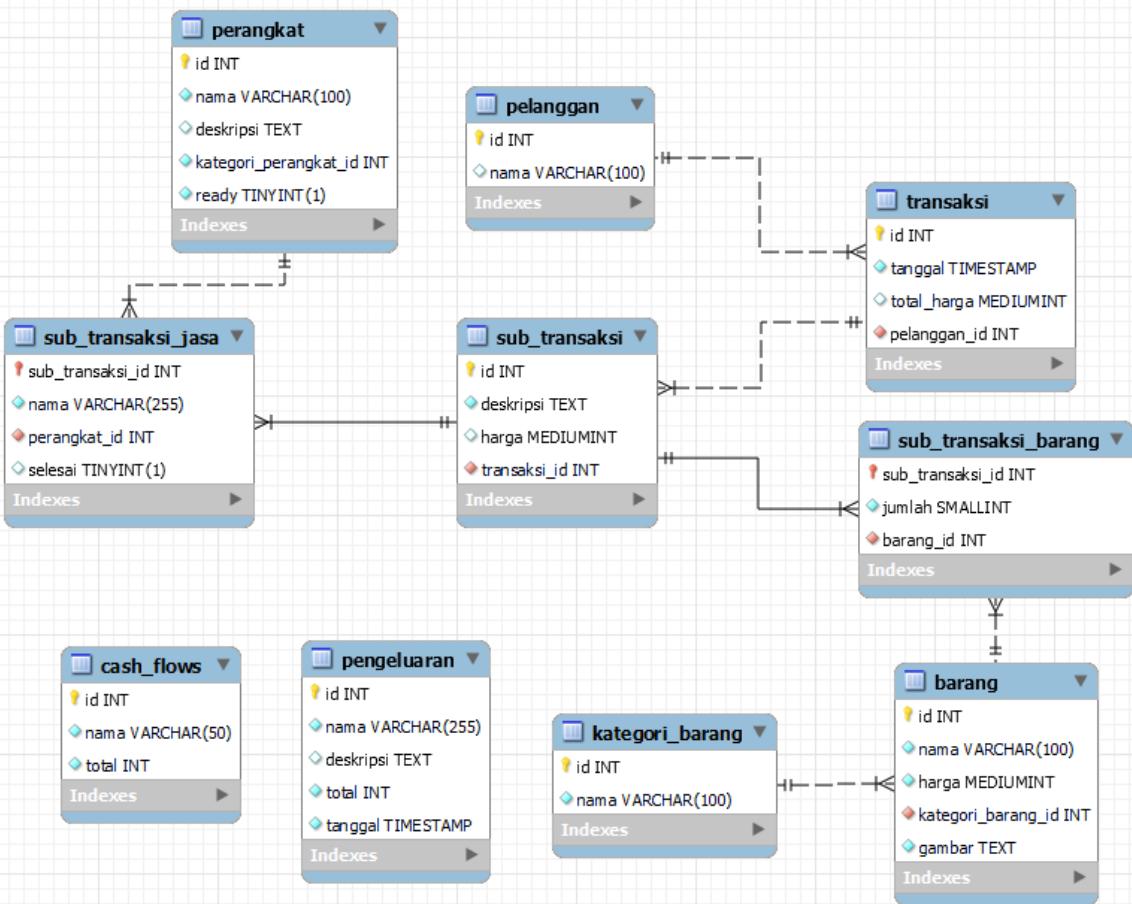
```
SELECT * FROM `pengguna`
```

Judul Aplikasi	
DASHBOARD PENJUALAN DAN DATA TOKO MAESTRO COPIES	
Dikembangkan Oleh	
<ol style="list-style-type: none"> 1. Rayhan Prasetya Ady / 5220411267 2. Echa Apriliyanto / 5220411272 3. Mochamad Fauzan Pratama Putra Sudiana / 5220411275 4. Syahrul Ramadhan / 5220411289 5. Abimanyu Putra Aria / 5220411291 	
Deskripsi Singkat Instansi, Permasalahan, dan Solusi Aplikasi Yang Akan Dikembangkan [10 poin]	<p>Maestro copies adalah toko yang menjual jasa untuk percetakan, laminating, scan dan menjual barang berupa alat alat kantor, instansi ini memiliki masalah pada ketepatan waktu tunggu jika ada banyak pelanggan yang datang untuk menggunakan jasanya, solusi aplikasi yang dapat kami kembangkan adalah dashboard untuk menampung data pesanan, mesin apa saja yang dipakai, kertas apa yang dapat diproses mesin, ketersediaan bahan cetak dan alat tulis; sehingga owner dapat dengan mudah melihat dan menentukan waktu yang tepat untuk mengambil/waktu selesai</p>
Pengguna Aplikasi [5 poin]	
Owner (Pemilik)	

Desain Entity-Relationship Diagram (ERD) [10 poin]



Desain Relasi Basis Data [10 poin]



Query Data Definition Language (DDL) [25 poin]

```
CREATE DATABASE IF NOT EXISTS maestro_copy;
```

```
USE maestro_copy;
```

```
CREATE TABLE IF NOT EXISTS perangkat (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nama VARCHAR(100) NOT NULL,
    deskripsi TEXT,
    kategori_perangkat_id INT NOT NULL,
    ready BOOLEAN NOT NULL DEFAULT TRUE
);
```

```
CREATE TABLE IF NOT EXISTS pelanggan (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nama VARCHAR(100)
);
```

```
CREATE TABLE IF NOT EXISTS transaksi (
    id INT AUTO_INCREMENT PRIMARY KEY,
    tanggal TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    total_harga MEDIUMINT UNSIGNED DEFAULT 0,
    pelanggan_id INT NOT NULL
);

CREATE TABLE IF NOT EXISTS sub_transaksi (
    id INT AUTO_INCREMENT PRIMARY KEY,
    deskripsi TEXT NOT NULL,
    harga MEDIUMINT UNSIGNED DEFAULT 0,
    transaksi_id INT NOT NULL
);

CREATE TABLE IF NOT EXISTS sub_transaksi_jasa (
    sub_transaksi_id INT NOT NULL REFERENCES sub_transaksi(id),
    nama VARCHAR(255) NOT NULL,
    perangkat_id INT NOT NULL,
    selesai BOOLEAN DEFAULT FALSE
);

CREATE TABLE IF NOT EXISTS sub_transaksi_barang (
    sub_transaksi_id INT NOT NULL REFERENCES sub_transaksi(id),
    jumlah SMALLINT UNSIGNED NOT NULL,
    barang_id INT NOT NULL
);

CREATE TABLE IF NOT EXISTS kategori_barang (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nama VARCHAR(100) NOT NULL
);

CREATE TABLE IF NOT EXISTS barang (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nama VARCHAR(100) NOT NULL,
    harga MEDIUMINT UNSIGNED NOT NULL,
```

```

kategori_barang_id INT NOT NULL,
gambar TEXT NOT NULL
);

CREATE TABLE IF NOT EXISTS pengeluaran (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nama VARCHAR(255) NOT NULL,
    deskripsi TEXT,
    total INT NOT NULL,
    tanggal TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE IF NOT EXISTS cash_flows (
    id INT PRIMARY KEY,
    nama VARCHAR(50) NOT NULL,
    total INT NOT NULL
);

ALTER TABLE sub_transaksi_jasa
ADD CONSTRAINT fk_subtransaksijasa_perangkat
FOREIGN KEY (perangkat_id)
REFERENCES perangkat (id)
ON DELETE CASCADE
ON UPDATE CASCADE;

ALTER TABLE sub_transaksi_jasa
ADD CONSTRAINT pk_subtransaksijasa
PRIMARY KEY (sub_transaksi_id);

ALTER TABLE sub_transaksi_barang
ADD CONSTRAINT pk_subtransaksibarang
PRIMARY KEY (sub_transaksi_id);

ALTER TABLE sub_transaksi
ADD CONSTRAINT fk_transaksi_subtransaksi

```

```
FOREIGN KEY (transaksi_id)
REFERENCES transaksi (id)
ON DELETE CASCADE
ON UPDATE CASCADE;

ALTER TABLE transaksi
ADD CONSTRAINT fk_pelanggan_transaksi
FOREIGN KEY (pelanggan_id)
REFERENCES pelanggan (id)
ON DELETE CASCADE
ON UPDATE CASCADE;

ALTER TABLE sub_transaksi_barang
ADD CONSTRAINT fk_barang_subtransaksibarang
FOREIGN KEY (barang_id)
REFERENCES barang (id)
ON DELETE CASCADE
ON UPDATE CASCADE;

ALTER TABLE barang
ADD CONSTRAINT fk_barang_kategoribarang
FOREIGN KEY (kategori_barang_id)
REFERENCES kategori_barang (id)
ON DELETE CASCADE
ON UPDATE CASCADE;

ALTER TABLE sub_transaksi_jasa
ADD CONSTRAINT fk_subtransaksi_subtransaksijasa
FOREIGN KEY (sub_transaksi_id)
REFERENCES sub_transaksi (id);

ALTER TABLE sub_transaksi_barang
ADD CONSTRAINT fk_subtransaksi_subtransaksibarang
FOREIGN KEY (sub_transaksi_id)
REFERENCES sub_transaksi (id);
```

Desain Interface (Wireframe/Mockup) Beserta Query-nya [poin 40]

```

CREATE TRIGGER
    tgr_setelah_insert_transaksi_pemasukan
AFTER INSERT ON
    transaksi
FOR EACH ROW
UPDATE
    cash_flows
SET
    cash_flows.total = cash_flows.total + NEW.total_harga
WHERE
    cash_flows.id = 1;

CREATE TRIGGER
    tgr_setelah_insert_pengeluaran
AFTER INSERT ON
    pengeluaran
FOR EACH ROW
UPDATE
    cash_flows
SET
    cash_flows.total = cash_flows.total + NEW.total
WHERE
    cash_flows.id = 2;

CREATE TRIGGER tgr_profit_bulan_ini_setelah_insert_transaksi
AFTER INSERT ON transaksi
FOR EACH ROW
UPDATE cash_flows
SET cash_flows.total = cash_flows.total +
    SELECT SUM(total_harga)
    FROM transaksi
    WHERE MONTH(tanggal) = MONTH(CURRENT_DATE())
        AND YEAR(tanggal) = YEAR(CURRENT_DATE())
)
WHERE cash_flows.id = 3;

CREATE TRIGGER tgr_profit_bulan_ini_setelah_insert_pengeluaran

```

```

AFTER INSERT ON pengeluaran
FOR EACH ROW
UPDATE cash_flows
    SET cash_flows.total = cash_flows.total - (
        SELECT SUM(total)
        FROM pengeluaran
        WHERE MONTH(tanggal) = MONTH(CURRENT_DATE())
            AND YEAR(tanggal) = YEAR(CURRENT_DATE())
    )
WHERE cash_flows.id = 3;

INSERT INTO
    cash_flows (id, nama, total)
VALUES
(1, 'total_pemasukan', 0),
(2, 'total_pengeluaran', 0),
(3, 'profit_bulan_ini', 0);

INSERT INTO
    perangkat (nama, deskripsi, kategori_perangkat_id)
VALUES
('perangkat1', 'deskripsi1', 1),
('perangkat2', 'deskripsi2', 2),
('perangkat3', 'deskripsi3', 3),
('perangkat4', 'deskripsi4', 4),
('perangkat5', 'deskripsi5', 5),
('perangkat6', 'deskripsi6', 6),
('perangkat7', 'deskripsi7', 7),
('perangkat8', 'deskripsi8', 8),
('perangkat9', 'deskripsi9', 9),
('perangkat10', 'deskripsi10', 10);

INSERT INTO
    pelanggan (nama)
VALUES
('Djoko Marwoto'),
('Mulya Kurniawan'),

```

```
('Limar Suwarno'),  
('Suci Lestari'),  
('Rudi Suryono'),  
('Kusuma Adriansyah'),  
('Kayla Nasyidah'),  
('Rika Pratiwi'),  
('Julia Novitasari'),  
('Agnes Kuswandari');
```

```
INSERT INTO  
    transaksi (pelanggan_id)  
VALUES  
(1),(2),(3),(4),(5),(6),(7),(8),(9),(10);
```

```
INSERT INTO  
    sub_transaksi (deskripsi, transaksi_id)  
VALUES  
('Buku tulis 10 pcs', 1), -- transaksi 1 barang  
('Pulpen 2 pcs', 2), -- transaksi 2  
('Isi ulang stapler 2 pcs', 3), -- transaksi 3  
('Kaos kaki size L', 4), -- transaksi 4  
('Penghapus 1 pc', 5), -- transaksi 5  
('Sabuk', 6), -- transaksi 6  
('Gunting', 7), -- transaksi 7  
('Penggaris 30 cm', 8), -- transaksi 8  
('Materai 10k 2pcs', 9), -- transaksi 9  
('Buku gambar A3', 10); -- transaksi 10
```

```
INSERT INTO  
    sub_transaksi (harga, deskripsi, transaksi_id)  
VALUES  
(10000, 'Ngeprint tugas basis data', 1), -- transaksi 1 jasa  
(12000, 'Jilid 24 halaman', 2), -- transaksi 2  
(7500, 'Print 15 lembar', 3), -- transaksi 3  
(1500, 'Fotokopi warna 3 dokumen', 4), -- transaksi 4  
(4000, 'Cetak foto 4x4 2pcs', 5), -- transaksi 5  
(2000, 'Laminating kertas', 6), -- transaksi 6
```

```
(3000, 'Scan dokumen', 7), -- transaksi 7  
(15000, 'Binding kertas menjadi buku', 8), -- transaksi 8  
(20000, 'Scan dokumen kuliah', 9), -- transaks5000, i 9  
(30000, 'Scan dokumen BPJS', 10); -- transaksi 10
```

```
INSERT INTO  
    kategori_barang (nama)  
VALUES  
    ('Alat tulis'), ('Alat kantor'), ('Alat sekolah'),  
    ('Peralatan umum');
```

```
INSERT INTO  
    barang (nama, harga, gambar, kategori_barang_id)  
VALUES  
    ('Pensil Standar', 2000, 'gambar1', 1),  
    ('Kertas HVS A4', 75, 'gambar2', 1),  
    ('Penggaris Butterfly', 2000, 'gambar3', 1),  
    ('Meterai Tempel 10Rb', 11000,'gambar4', 2),  
    ('Amplop 10pcs', 2500,'gambar5', 2),  
    ('Meterai Tempel 6Rb', 7000,'gambar6', 2),  
    ('Tepak Pensil Unicorn', 20000,'gambar7', 3),  
    ('Tongkat Pramuka', 1500,'gambar8', 3),  
    ('Buku Saku', 5000,'gambar9', 3),  
    ('Gunting', 10000,'gambar10', 4),  
    ('Stapler', 10000,'gambar10', 4),  
    ('Gunting Kuku', 7500,'gambar10', 4);
```

```
INSERT INTO pengeluaran (nama, deskripsi, total)  
VALUES  
    ('beli gunting', 'gunting besar 1 dus', 50000),  
    ('beli kertas', 'hvs A4 1 rim', 7500),  
    ('beli bolpen', 'standard 1 box', 35000),  
    ('bayar wifi', 'langganan bulanan', 350000),  
    ('bayar listrik', '2000 watt', 400000),  
    ('beli kertas', 'hvs A3 1 rim', 10000),  
    ('beli kertas', 'art paper 50 lembar', 150000),
```

```
('beli kertas', 'bc indah 1 rim', 245000),
('beli tinta printer', 'Tinta Canon', 50000),
('beli buku tulis', 'buku tulis Campus', 60000);
```

```
INSERT INTO
```

```
    sub_transaksi_jasa (nama, sub_transaksi_id, perangkat_id, selesai)
```

```
VALUES
```

```
("nama 1", 11, 6, FALSE),
("nama 2", 12, 6, FALSE),
("nama 3", 13, 7, FALSE),
("nama 4", 14, 7, FALSE),
("nama 5", 15, 8, FALSE),
("nama 6", 16, 8, FALSE),
("nama 7", 17, 9, FALSE),
("nama 8", 18, 9, FALSE),
("nama 9", 19, 10, FALSE),
("nama 10", 20, 10, FALSE);
```

```
INSERT INTO
```

```
    sub_transaksi_barang (sub_transaksi_id, jumlah, barang_id)
```

```
VALUES
```

```
(1, 10, 1), (2, 10, 1),(3, 30, 2), (4, 10, 3),(5, 10, 4), (6, 10, 5),(7, 10, 6),(8, 50, 7), (9, 70, 8), (10, 90, 9);
```

```
CREATE OR REPLACE VIEW
```

```
    view_pemasukan_minggu_ini
```

```
SELECT
```

```
CASE DAYOFWEEK(tanggal)
    WHEN 1 THEN 'Minggu'
    WHEN 2 THEN 'Senin'
    WHEN 3 THEN 'Selasa'
    WHEN 4 THEN 'Rabu'
    WHEN 5 THEN 'Kamis'
    WHEN 6 THEN 'Jumat'
    WHEN 7 THEN 'Sabtu'
```

```
END AS hari,
SUM(total_harga) AS pemasukan
FROM
transaksi
WHERE
tanggal >= DATE_SUB(CURDATE(), INTERVAL 1 WEEK)
GROUP BY
DAYOFWEEK(tanggal)
ORDER BY
DAYOFWEEK(tanggal);

CREATE OR REPLACE VIEW
view_pemasukan_bulan_ini
SELECT

DAY(tanggal) AS tanggal,
SUM(total_harga) AS pemasukan
FROM
transaksi
WHERE
MONTH(tanggal) = MONTH(CURRENT_DATE())
AND YEAR(tanggal) = YEAR(CURRENT_DATE())
GROUP BY
DAY(tanggal)
ORDER BY
DAY(tanggal);

CREATE OR REPLACE VIEW
view_pemasukan_bulan_ini
SELECT

DAY(tanggal) AS tanggal,
SUM(total_harga) AS pemasukan
FROM
transaksi
```

```
WHERE
    MONTH(tanggal) = MONTH(CURRENT_DATE())
    AND YEAR(tanggal) = YEAR(CURRENT_DATE())
GROUP BY
    DAY(tanggal)
ORDER BY
    DAY(tanggal);
```

```
CREATE OR REPLACE VIEW
    view_pemasukan_tahun_ini
SELECT
    MONTH(tanggal) as bulan,
    SUM(total_harga) as pemasukan
FROM
    transaksi
GROUP BY
    MONTH(tanggal);
```

```
CREATE OR REPLACE VIEW
    view_total_pemasukan
SELECT
    *
FROM
    cash_flows
WHERE
    id = 1;
```

```
CREATE OR REPLACE VIEW
    view_pengeluaran_minggu_ini
SELECT
    CASE DAYOFWEEK(tanggal)
        WHEN 1 THEN 'Minggu'
        WHEN 2 THEN 'Senin'
        WHEN 3 THEN 'Selasa'
        WHEN 4 THEN 'Rabu'
```

```

        WHEN 5 THEN 'Kamis'
        WHEN 6 THEN 'Jumat'
        WHEN 7 THEN 'Sabtu'
    END AS hari,
    SUM(total) AS pemasukan
FROM
    pengeluaran
WHERE
    tanggal >= DATE_SUB(CURDATE(), INTERVAL 1 WEEK)
GROUP BY
    DAYOFWEEK(tanggal)
ORDER BY
    DAYOFWEEK(tanggal);

CREATE OR REPLACE VIEW
    view_pengeluaran_bulan_ini
SELECT

    DAY(tanggal) AS tanggal,
    SUM(total) AS pengeluaran
FROM
    pengeluaran
WHERE
    MONTH(tanggal) = MONTH(CURRENT_DATE())
    AND YEAR(tanggal) = YEAR(CURRENT_DATE())
GROUP BY
    DAY(tanggal)
ORDER BY
    DAY(tanggal);

CREATE OR REPLACE VIEW
    view_pengeluaran_tahun_ini
SELECT
    MONTH(tanggal) AS bulan,
    SUM(total) AS pengeluaran
FROM
    pengeluaran

```

```
GROUP BY
    MONTH(tanggal);

CREATE OR REPLACE VIEW
    view_total_pengeluaran
SELECT
    *
FROM
    cash_flows
WHERE
    id = 2;

CREATE OR REPLACE VIEW
    view_profit_bulan_ini
SELECT
    *
FROM
    cash_flows
WHERE
    id = 3;

CREATE OR REPLACE VIEW
    view_barang
SELECT
    *
FROM
    barang;

SELECT * FROM view_barang LIMIT 8;

CREATE OR REPLACE VIEW
    view_transaksi
SELECT
    *
FROM
```

```
transaksi;

SELECT * FROM view_transaksi LIMIT 7;

CREATE OR REPLACE VIEW
    view_perangkat
SELECT
    *
FROM
    perangkat;

SELECT * FROM view_perangkat LIMIT 7;

CREATE OR REPLACE VIEW
    view_pengeluaran
SELECT
    *
FROM
    pengeluaran;

SELECT * FROM view_pengeluaran
LIMIT 7;

SELECT * FROM ub_transaksi
HERE
ub_transaksi.transaksi_id = 1;

DELETE FROM pengeluaran
WHERE id = 1;

DELETE FROM barang
WHERE id = 1;

DELETE FROM transaksi WHERE id = 1;

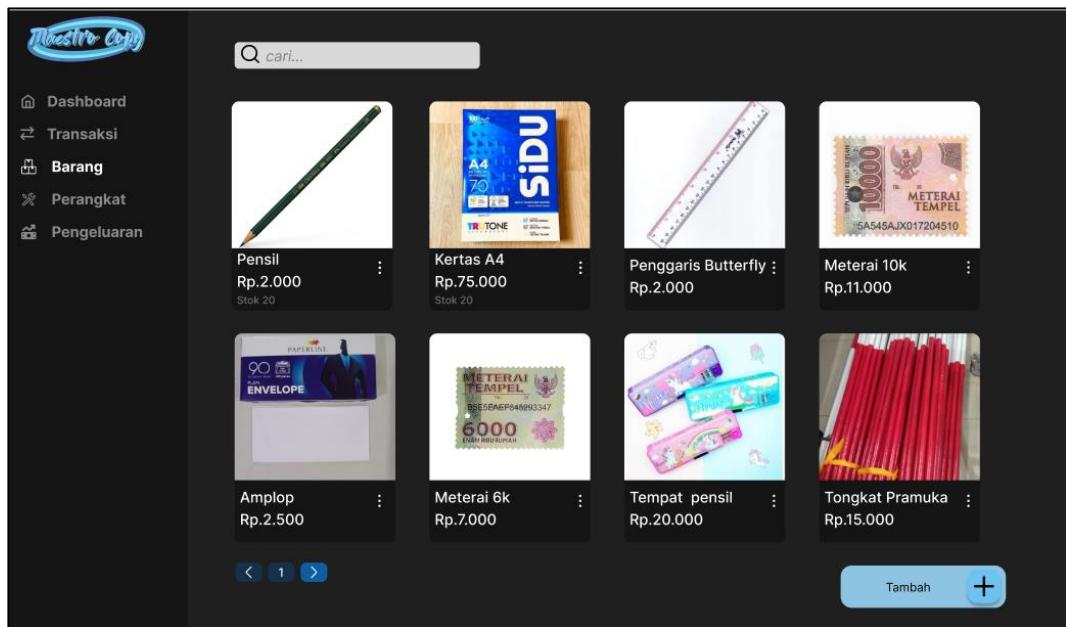
UPDATE barang
SET
```

```
nama = 'nama_baru',
harga = 5000,
gambar = 'gambar_baru.jpg'
WHERE id = 1;
```

```
UPDATE perangkat
SET
    IF ready = TRUE
        THEN ready = FALSE
    ELSE
        THEN ready = TRUE
WHERE
    id = 1;
```

Mockup





The screenshot shows the Maestro Copies transaction history table. The sidebar on the left includes icons for Dashboard, Transaksi, Barang, Perangkat, and Pengeluaran. A search bar with the placeholder "cari..." is at the top. The table has a header row with columns: Title, No, Nama pelanggan, Tanggal, and Total Harga. The data rows are:

Title	No	Nama pelanggan	Tanggal	Total Harga
Row 1		Djoko Marwoto	2024-05-19 22:15:29	Rp.25.000
Row 2		Mulya Kurniawan		456
Row 3		Limar Suwarno		456
Row 4		Rudi Suryono		456
Row 5		Kayla Nasyidah		456
Row 6		Rika Pratiwi		456
Row 7				456

At the bottom right is a blue button labeled "Transaksi Baru" with a plus sign.

Maestro Copy

cari...

Perangkat

No	Title	deskripsi	Status
Row 1	Kanon e69	print warna	READY
Row 2	Epson o7	print hitam putih	BUSY
Row 4			
Row 5			
Row 6			
Row 7			

< 1 >

Tambah

Pengeluaran

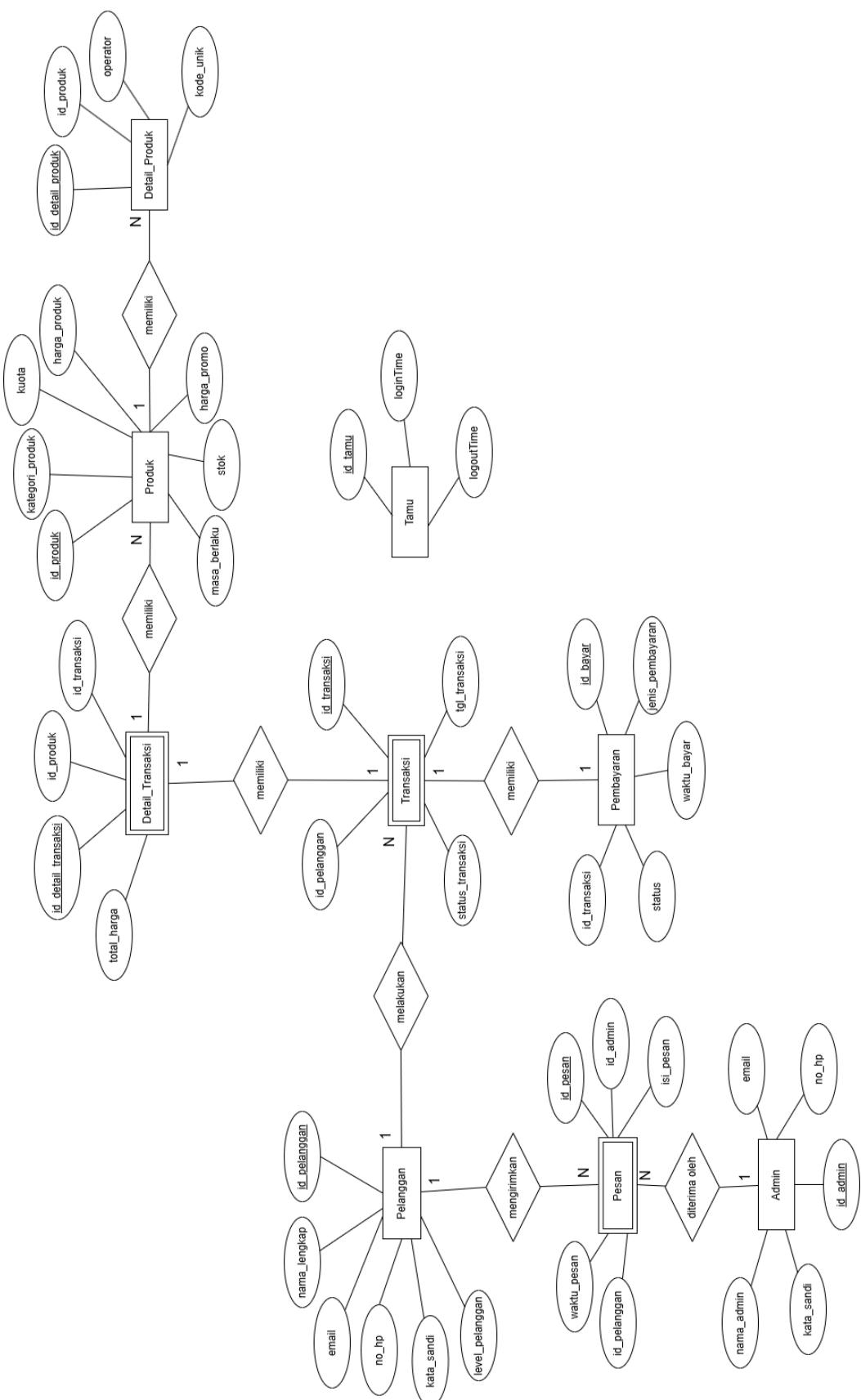
No	Nama pengeluaran	Tanggal	deskripsi	total
1	beli gunting	2024-05-25 22:15:29	gunting besar 1 dus	Rp.50.000
2	beli kertas	2024-05-27 22:15:29	hvs A4 1 rim	Rp.75.000
3	beli bolpen	2024-05-28 22:15:29	stabard 1 box	Rp.35.000
4	bayar wifi	2024-05-29 22:15:29	langganan bulanan	Rp.350.000
5	bayar listrik	2024-05-30 22:15:29	2000 watt	Rp.400.000
6	beli kertas	2024-06-01 22:15:29	hvs A3 1 rim	Rp.90.000
7	beli kertas	2024-06-02 22:15:29	Art Paper 50 Lembar	Rp.150.000

< 1 >

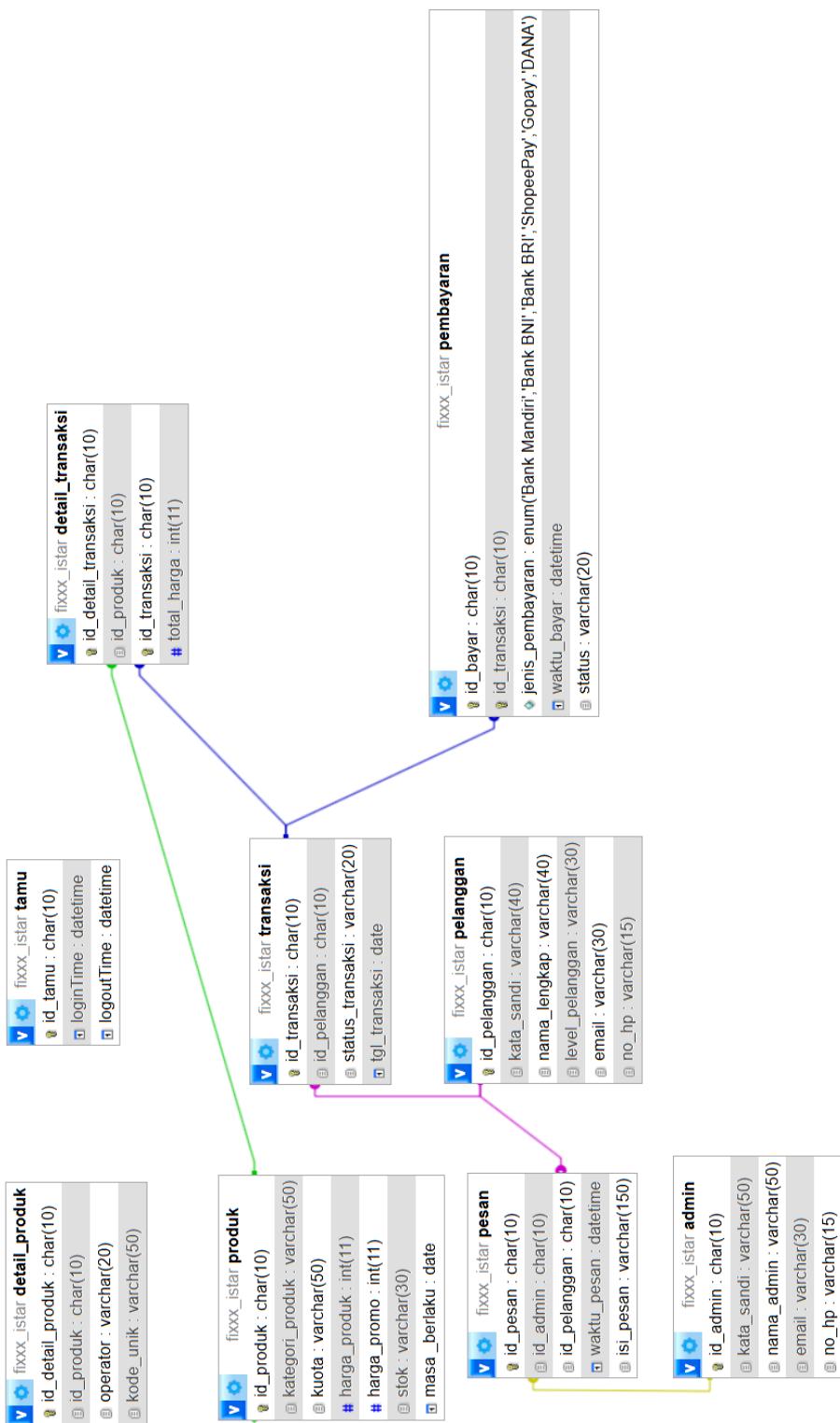
Tambah

Judul Aplikasi	APLIKASI PENJUALAN KONTER ISTAR CELL (ICELL.)	
Dikembangkan Oleh		
1. 5220411281 – Yahya Abdulhafidz 2. 5220411284 – Enggi Wira Praja Putri Taufani 3. 5220411286 – Ahmad Salim Hibatulloh 4. 5220411287 – Sukma Indriya Wati 5. 5220411290 – Rizka Octa Setiani		
Deskripsi Singkat Instansi, Permasalahan, dan Solusi Aplikasi Yang Akan Dikembangkan [10 poin]		
<ul style="list-style-type: none"> - Instansi : Istar Cell adalah sebuah toko konter yang berada di Jalan Kaliurang No.27 Km 5.2, Karang Wuni, Caturtunggal, Kec. Depok, Kabupaten Sleman, DIY yang menyediakan berbagai layanan dan produk terkait teknologi, termasuk penjualan pulsa, kuota internet, kartu perdana, serta beragam aksesoris hp. Mereka juga menawarkan layanan pembayaran tagihan PLN dan pembelian token PLN. Selain itu, mereka menyediakan layanan top up untuk berbagai e-wallet, serta melayani transfer uang antar bank. - Permasalahan : Istar Cell ingin meningkatkan penjualan produk-produk utama seperti pulsa, kuota internet, kartu perdana, pembayaran tagihan PLN, dan pembelian token PLN di Istar Cell, serta meningkatkan penggunaan layanan top up e-wallet yang disediakan. Mereka ingin memperluas jangkauan penjualan dengan menggunakan aplikasi, bukan hanya bergantung pada penjualan offline. - Solusi : Aplikasi seluler yang memungkinkan pelanggan untuk membeli produk-produk utama seperti pulsa, kuota internet, kartu perdana, pembayaran tagihan PLN, dan pembelian token PLN secara online. Aplikasi tersebut juga dapat menyediakan fitur top up untuk berbagai e-wallet. Dengan menggunakan aplikasi, pelanggan dapat dengan mudah melakukan transaksi kapan pun dan di mana pun, memperluas jangkauan penjualan Istar Cell secara signifikan serta meningkatkan penggunaan layanan top up e-wallet yang disediakan. 		
Pengguna Aplikasi [5 poin]		
Target dari aplikasi ini adalah pelanggan dan admin.		

Desain Entity-Relationship Diagram (ERD) [10 poin]



Desain Relasi Basis Data [10 poin]



Query Data Definition Language (DDL) [25 poin]

```
--  
-- Table structure for table `admin`  
  
--  
  
CREATE TABLE `admin` (  
    `id_admin` char(10) NOT NULL,  
    `kata_sandi` varchar(50) NOT NULL,  
    `nama_admin` varchar(50) NOT NULL,  
    `email` varchar(30) NOT NULL,  
    `no_hp` varchar(15) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;  
  
--  
-- Triggers `admin`  
  
--  
  
DELIMITER $$  
CREATE TRIGGER `trg_insert_admin` BEFORE INSERT ON `admin` FOR EACH  
ROW BEGIN  
    DECLARE next_id INT;  
  
    SELECT SUBSTRING(id_admin, 4) + 1 INTO next_id  
    FROM admin  
    ORDER BY id_admin DESC  
    LIMIT 1;  
  
    IF next_id IS NULL THEN  
        SET next_id = 1;  
    END IF;  
  
    SET NEW.id_admin = CONCAT('ADM', LPAD(next_id, 3, '0'));  
END  
$$  
DELIMITER ;  
-- -----  
-- Table structure for table `detail_produk`  
--
```

```

CREATE TABLE `detail_produk` (
  `id_detail_produk` char(10) NOT NULL,
  `id_produk` char(10) NOT NULL,
  `operator` varchar(20) NOT NULL,
  `kode_unik` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

-- 
-- Triggers `detail_produk`
-- 

DELIMITER $$

CREATE TRIGGER `trg_insert_detail_produk` BEFORE INSERT ON
`detail_produk` FOR EACH ROW BEGIN
    DECLARE next_id INT;

    SELECT SUBSTRING(id_detail_produk, 4) + 1 INTO next_id
    FROM detail_produk
    ORDER BY id_detail_produk DESC
    LIMIT 1;

    IF next_id IS NULL THEN
        SET next_id = 1;
    END IF;

    SET NEW.id_detail_produk = CONCAT('DPR', LPAD(next_id, 3, '0'));
END
$$
DELIMITER ;

-- -----
-- 
-- Table structure for table `detail_transaksi`
-- 

CREATE TABLE `detail_transaksi` (

```

```

`id_detail_transaksi` char(10) NOT NULL,
`id_produk` char(10) NOT NULL,
`id_transaksi` char(10) NOT NULL,
`total_harga` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

-- 
-- Triggers `detail_transaksi`

-- 
DELIMITER $$

CREATE TRIGGER `trg_insert_detail_transaksi` BEFORE INSERT ON
`detail_transaksi` FOR EACH ROW BEGIN
    DECLARE next_id INT;

    SELECT SUBSTRING(id_detail_transaksi, 4) + 1 INTO next_id
    FROM detail_transaksi
    ORDER BY id_detail_transaksi DESC
    LIMIT 1;

    IF next_id IS NULL THEN
        SET next_id = 1;
    END IF;

    SET NEW.id_detail_transaksi = CONCAT('DTR', LPAD(next_id, 3, '0'));
END
$$
DELIMITER ;

-- -----
-- 
-- Table structure for table `pelanggan`
-- 

CREATE TABLE `pelanggan` (
`id_pelanggan` char(10) NOT NULL,
`kata_sandi` varchar(40) NOT NULL,

```

```

`nama_lengkap` varchar(40) NOT NULL,
`level_pelanggan` varchar(30) NOT NULL DEFAULT 'Bronze',
`email` varchar(30) NOT NULL,
`no_hp` varchar(15) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

-- 
-- Triggers `pelanggan`

-- 
DELIMITER $$

CREATE TRIGGER `trg_insert_pelanggan` BEFORE INSERT ON `pelanggan`
FOR EACH ROW BEGIN
    DECLARE next_id INT;

    SELECT SUBSTRING(id_pelanggan, 4) + 1 INTO next_id
    FROM Pelanggan
    ORDER BY id_pelanggan DESC
    LIMIT 1;

    IF next_id IS NULL THEN
        SET next_id = 1;
    END IF;

    SET NEW.id_pelanggan = CONCAT('PLG', LPAD(next_id, 3, '0'));
END
$$
DELIMITER ;

-- -----
-- 
-- Table structure for table `pembayaran`

-- 

CREATE TABLE `pembayaran` (
`id_bayar` char(10) NOT NULL,
`id_transaksi` char(10) NOT NULL,

```

```

`jenis_pembayaran` enum('Bank Mandiri','Bank BNI','Bank
BRI','ShopeePay','Gopay','DANA') NOT NULL,
`waktu_bayar` datetime NOT NULL,
`status` varchar(20) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

-- 
-- Triggers `pembayaran`
-- 

DELIMITER $$

CREATE TRIGGER `trg_insert_pembayaran` BEFORE INSERT ON `pembayaran`
FOR EACH ROW BEGIN
    DECLARE next_id INT;

    SELECT SUBSTRING(id_bayar, 4) + 1 INTO next_id
    FROM pembayaran
    ORDER BY id_bayar DESC
    LIMIT 1;

    IF next_id IS NULL THEN
        SET next_id = 1;
    END IF;

    SET NEW.id_bayar = CONCAT('BYR', LPAD(next_id, 3, '0'));
END
$$
DELIMITER ;

-----  

-- 
-- Table structure for table `pesan`  

-- 

CREATE TABLE `pesan` (
`id_pesan` char(10) NOT NULL,
`id_admin` char(10) NOT NULL,
`id_pelanggan` char(10) NOT NULL,

```

```

`waktu_pesanan` datetime NOT NULL,
`isi_pesanan` varchar(150) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
-- 
-- Triggers `pesan`
-- 

DELIMITER $$

CREATE TRIGGER `trg_insert_pesanan` BEFORE INSERT ON `pesan` FOR EACH
ROW BEGIN
    DECLARE next_id INT;

    SELECT SUBSTRING(id_pesanan, 4) + 1 INTO next_id
    FROM pesan
    ORDER BY id_pesanan DESC
    LIMIT 1;

    IF next_id IS NULL THEN
        SET next_id = 1;
    END IF;

    SET NEW.id_pesanan = CONCAT('PSN', LPAD(next_id, 3, '0'));
END
$$

DELIMITER ;

-- -----
-- 
-- Table structure for table `produk`
-- 

CREATE TABLE `produk` (
`id_produk` char(10) NOT NULL,
`kategori_produk` varchar(50) NOT NULL,
`kuota` varchar(50) NOT NULL,
`harga_produk` int(11) NOT NULL,
`harga_promo` int(11) DEFAULT NULL,
`stok` varchar(30) NOT NULL,

```

```

`masa_berlaku` date NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

-- 
-- Triggers `produk`
-- 

DELIMITER $$

CREATE TRIGGER `trg_insert_produk` BEFORE INSERT ON `produk` FOR
EACH ROW BEGIN
    DECLARE next_id INT;

    SELECT SUBSTRING(id_produk, 4) + 1 INTO next_id
    FROM Produk
    ORDER BY id_produk DESC
    LIMIT 1;

    IF next_id IS NULL THEN
        SET next_id = 1;
    END IF;

    SET NEW.id_produk = CONCAT('PRD', LPAD(next_id, 3, '0'));
END
$$

DELIMITER ;

-- -----
-- 

-- Table structure for table `tamu`

-- 

CREATE TABLE `tamu` (
    `id_tamu` char(10) NOT NULL,
    `loginTime` datetime NOT NULL,
    `logoutTime` datetime NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

```

```

-- 
-- Triggers `tamu` 

-- 

DELIMITER $$

CREATE TRIGGER `trg_insert_tamu` BEFORE INSERT ON `tamu` FOR EACH
ROW BEGIN
    DECLARE next_id INT;

    SELECT SUBSTRING(id_tamu, 4) + 1 INTO next_id
    FROM tamu
    ORDER BY id_tamu DESC
    LIMIT 1;

    IF next_id IS NULL THEN
        SET next_id = 1;
    END IF;

    SET NEW.id_tamu = CONCAT('TMU', LPAD(next_id, 3, '0'));
END
$$

DELIMITER ;

-----


-- 
-- Table structure for table `transaksi` 

-- 

CREATE TABLE `transaksi` (
    `id_transaksi` char(10) NOT NULL,
    `id_pelanggan` char(10) NOT NULL,
    `id_admin` char(10) NOT NULL,
    `tanggal_transaksi` datetime NOT NULL,
    `total_bayar` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

-- 

```

```

-- Triggers `transaksi`

-- 

DELIMITER $$

CREATE TRIGGER `trg_insert_transaksi` BEFORE INSERT ON `transaksi` FOR
EACH ROW BEGIN
    DECLARE next_id INT;

    SELECT SUBSTRING(id_transaksi, 4) + 1 INTO next_id
    FROM transaksi
    ORDER BY id_transaksi DESC
    LIMIT 1;

    IF next_id IS NULL THEN
        SET next_id = 1;
    END IF;

    SET NEW.id_transaksi = CONCAT('TRS', LPAD(next_id, 3, '0'));
END
$$

DELIMITER ;

-- 
-- Indexes for dumped tables
-- 
-- 
-- 
-- Indexes for table `admin`
-- 
ALTER TABLE `admin`
    ADD PRIMARY KEY (`id_admin`);

-- 
-- Indexes for table `detail_produk`
-- 
ALTER TABLE `detail_produk`
    ADD PRIMARY KEY (`id_detail_produk`),
    ADD KEY `id_produk` (`id_produk`);

```

```
--  
-- Indexes for table `detail_transaksi`  
--  
ALTER TABLE `detail_transaksi`  
    ADD PRIMARY KEY (`id_detail_transaksi`),  
    ADD UNIQUE KEY `id_transaksi` (`id_transaksi`),  
    ADD KEY `id_produk` (`id_produk`);  
  
--  
-- Indexes for table `pelanggan`  
--  
ALTER TABLE `pelanggan`  
    ADD PRIMARY KEY (`id_pelanggan`);  
  
--  
-- Indexes for table `pembayaran`  
--  
ALTER TABLE `pembayaran`  
    ADD PRIMARY KEY (`id_bayar`),  
    ADD UNIQUE KEY `id_transaksi` (`id_transaksi`);  
  
--  
-- Indexes for table `pesan`  
--  
ALTER TABLE `pesan`  
    ADD PRIMARY KEY (`id_pesan`),  
    ADD KEY `id_admin` (`id_admin`),  
    ADD KEY `id_pelanggan` (`id_pelanggan`);  
  
--  
-- Indexes for table `produk`  
--  
ALTER TABLE `produk`  
    ADD PRIMARY KEY (`id_produk`);  
--
```

```

-- Indexes for table `tamu`
--

ALTER TABLE `tamu`
    ADD PRIMARY KEY (`id_tamu`);


-- 
-- Indexes for table `transaksi`
-- 

ALTER TABLE `transaksi`
    ADD PRIMARY KEY (`id_transaksi`),
    ADD KEY `id_pelanggan` (`id_pelanggan`);


-- 
-- Constraints for dumped tables
-- 

-- 
-- Constraints for table `detail_produk`
-- 

ALTER TABLE `detail_produk`
    ADD CONSTRAINT `detail_produk_ibfk_1` FOREIGN KEY (`id_produk`)
REFERENCES `produk` (`id_produk`);


-- 
-- Constraints for table `detail_transaksi`
-- 

ALTER TABLE `detail_transaksi`
    ADD CONSTRAINT `detail_transaksi_ibfk_1` FOREIGN KEY (`id_produk`)
REFERENCES `produk` (`id_produk`),
    ADD CONSTRAINT `detail_transaksi_ibfk_2` FOREIGN KEY (`id_transaksi`)
REFERENCES `transaksi` (`id_transaksi`);


-- 
-- Constraints for table `pembayaran`
-- 

ALTER TABLE `pembayaran`

```

```
ADD CONSTRAINT `pembayaran_ibfk_1` FOREIGN KEY (`id_transaksi`)
REFERENCES `transaksi` (`id_transaksi`);

-- 
-- Constraints for table `pesan`
-- 

ALTER TABLE `pesan`
    ADD CONSTRAINT `pesan_ibfk_1` FOREIGN KEY (`id_admin`) REFERENCES
`admin` (`id_admin`),
    ADD CONSTRAINT `pesan_ibfk_2` FOREIGN KEY (`id_pelanggan`)
REFERENCES `pelanggan` (`id_pelanggan`);

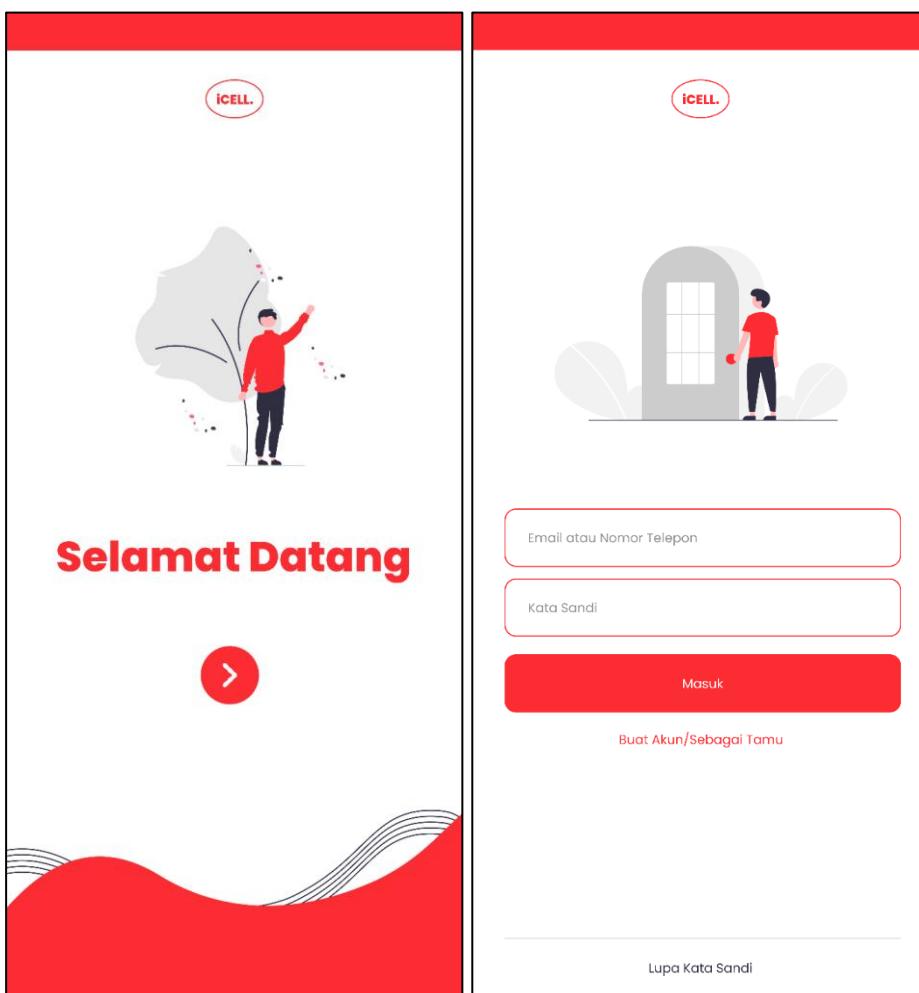
-- 
-- Constraints for table `transaksi`
-- 

ALTER TABLE `transaksi`
    ADD CONSTRAINT `transaksi_ibfk_1` FOREIGN KEY (`id_pelanggan`)
REFERENCES `pelanggan` (`id_pelanggan`);

COMMIT;
```

Desain Interface (Wireframe/Mockup) Beserta Query-nya [poin 40]

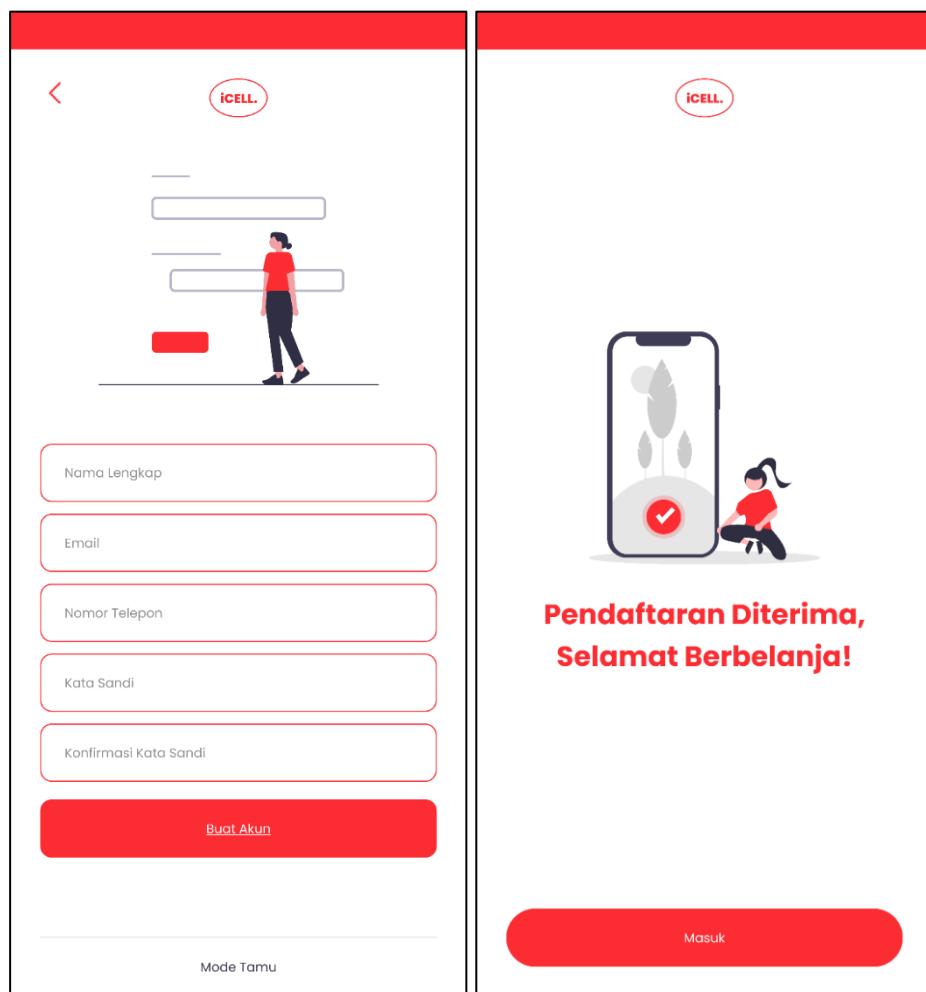
1. Tampilan Awal dan Masuk



Query tampilan masuk:

```
# ini mencari data pengguna pada database
SELECT * FROM pelanggan
WHERE (email = 'anonym.email@gmail.com' OR no_hp = '081234567890')
AND kata_sandi = anonym123';
```

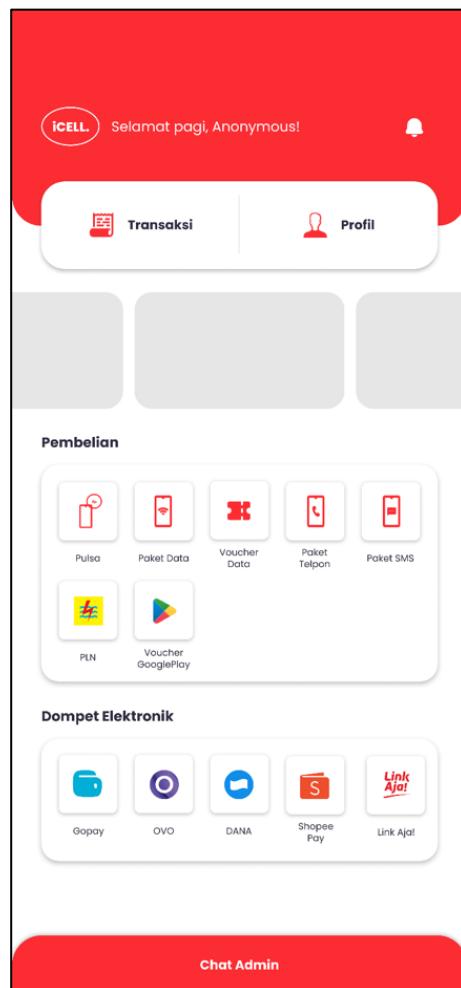
2. Tampilan Daftar



Query tampilan daftar:

```
# ini akan mengirimkan data pengguna ke database
INSERT INTO pelanggan (id_pelanggan, kata_sandi, nama_lengkap, email, no_hp)
VALUES ('PLG001', 'anonym123', 'Anonymous', 'anonym.email@gmail.com',
'081234567890');
```

3. Tampilan halaman utama

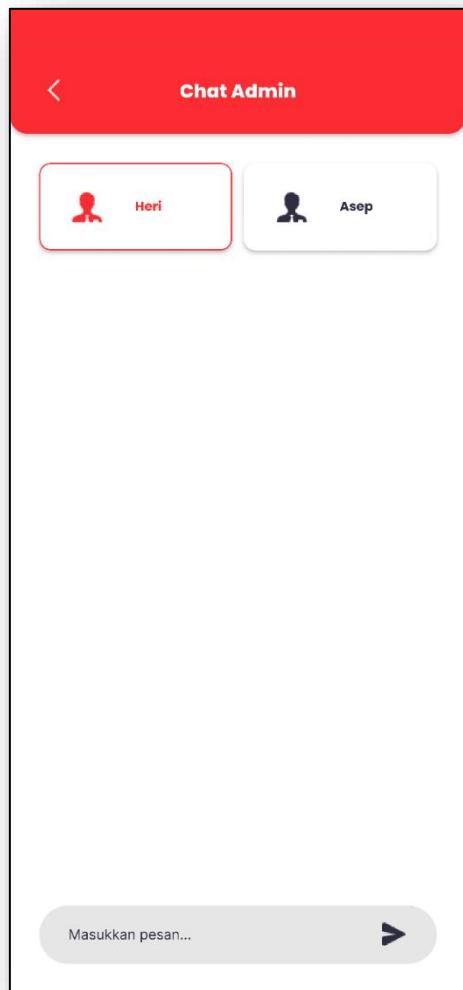


Query tampilan halaman utama:

hanya menampilkan nama pelanggan

```
SELECT nama_lengkap FROM pelanggan
```

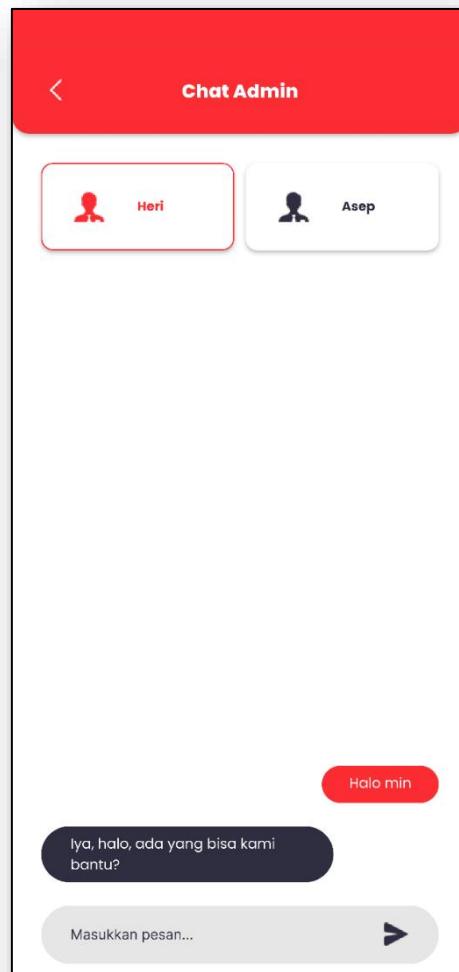
4. Tampilan chat admin



Query tampilan chat admin:

```
# ini saat memilih pesan akan dikirimkan ke admin
SELECT isi_pesan, nama_admin, waktu_pesan
FROM pesan, admin
WHERE nama_admin IN ('Heri');
```

5. Tampilan pelanggan mengirim pesan ke admin

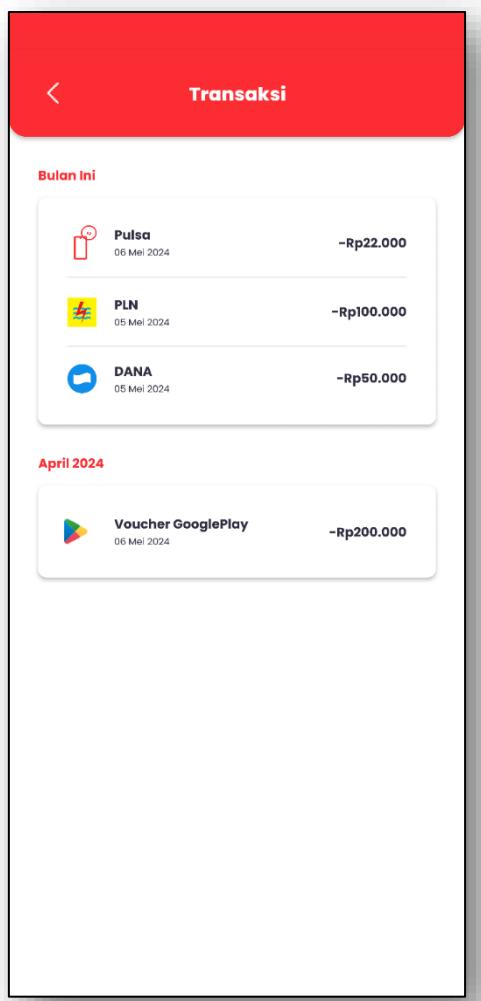


Query tampilan chat admin (mengirim pesan):

ini dilakukan saat menekan tombol kirim

```
INSERT INTO pesan (id_pelanggan, id_admin, isi_pesanan)
VALUES (
    (SELECT id_pelanggan FROM pelanggan WHERE nama_lengkap =
'Anonymous'),
    (SELECT id_admin FROM admin WHERE nama_admin = 'Heri'),
    'Halo min'
);
```

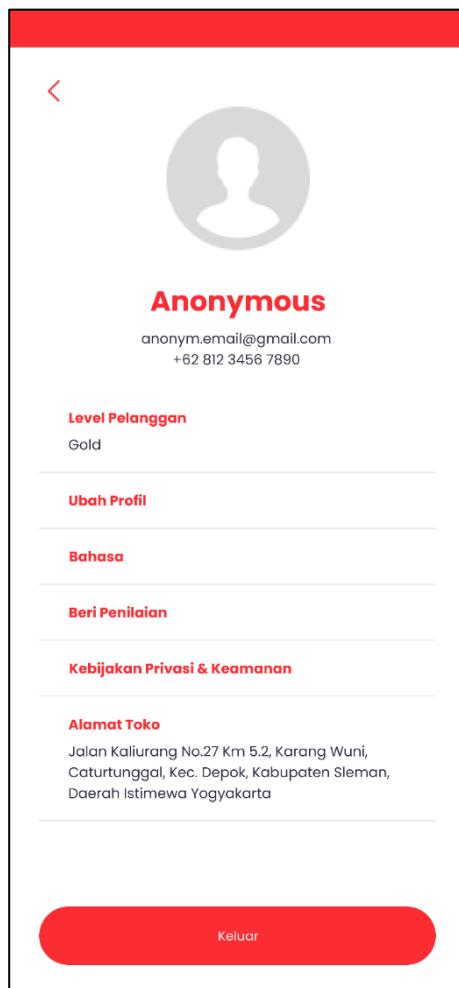
6. Halaman Transaksi



Query tampilan transaksi:

```
# ini akan melihatkan semua transaksi pengguna
SELECT transaksi.tgl_transaksi, produk.kategori_produk,
detail_transaksi.total_harga
FROM transaksi
JOIN pelanggan ON transaksi.id_pelanggan = pelanggan.id_pelanggan
JOIN detail_transaksi ON transaksi.id_transaksi = detail_transaksi.id_transaksi
JOIN produk ON detail_transaksi.id_produk = produk.id_produk
ORDER BY
transaksi.tgl_transaksi DESC;
```

7. Tampilan Profil

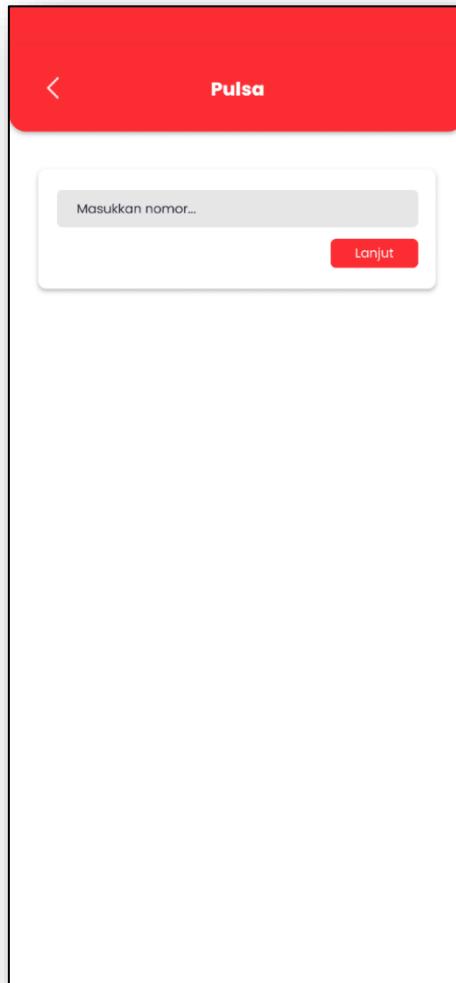


Query tampilan profil:

untuk menampilkan tentunya perlu kode pada aplikasi dan web

```
SELECT pelanggan.nama_lengkap, pelanggan.email, pelanggan.no_hp,  
pelanggan.level_pelanggan  
FROM pelanggan  
WHERE id_pelanggan = 'PLG001';
```

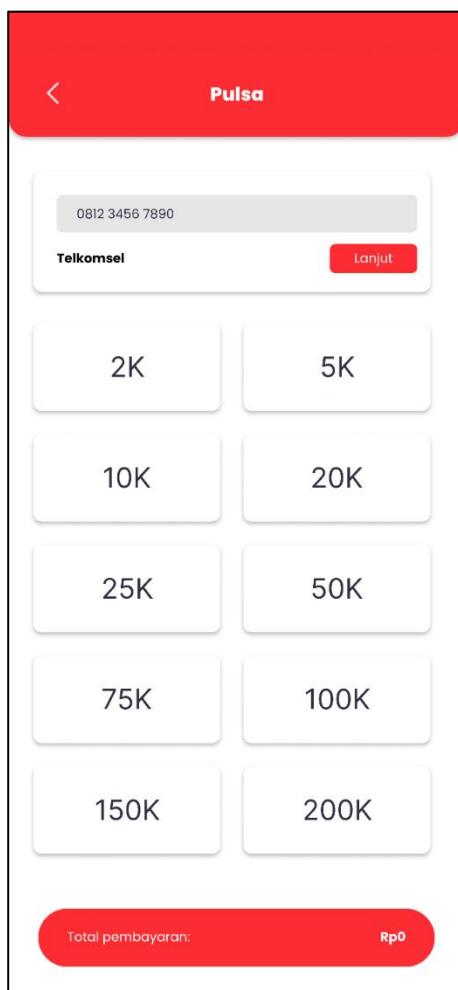
8. Tampilan Produk (Pulsa)



Query memilih pulsa:

```
# hanya mengambil database  
SELECT kategori_produk  
FROM produk  
WHERE  
kategori_produk = 'Pulsa';
```

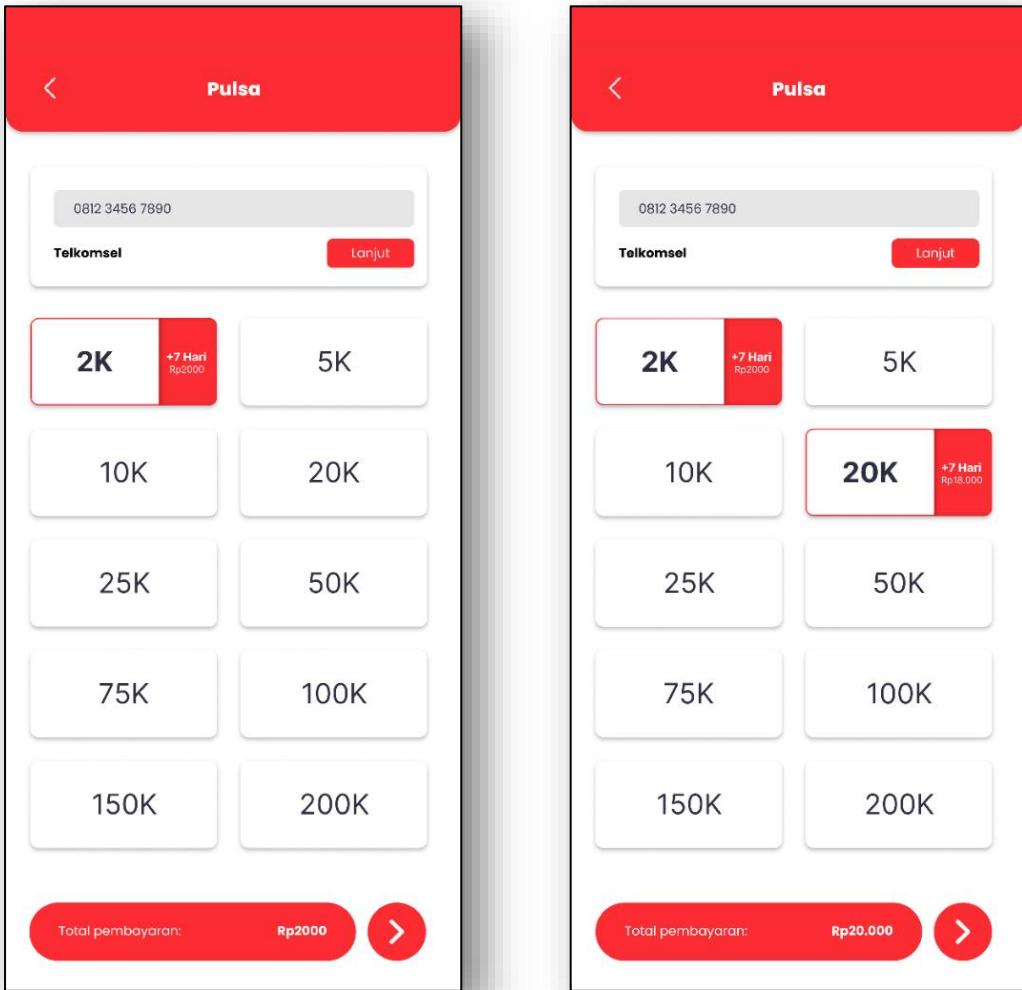
9. Tampilan melakukan pembelian produk



Query tampilan pulsa (saat sudah mengisi nomor pulsa):

```
# ini untuk memastikan operator
SELECT
    detail_produk.operator,
    produk.kategori_produk,
    produk.kuota
FROM
    detail_produk
JOIN
    produk ON detail_produk.id_produk = produk.id_produk
WHERE
    detail_produk.kode_unik = LEFT('081234567890', 4);
```

10. Tampilan pengguna memilih bagian dari produk



Query tampilan pulsa (saat memilih kuota):

ini memperlihatkan isi produk

SELECT

```
detail_produk.operator,  
produk.kategori_produk,  
produk.kuota,  
produk.harga_produk,  
produk.harga_promo,  
produk.masa_berlaku
```

FROM

```
detail_produk
```

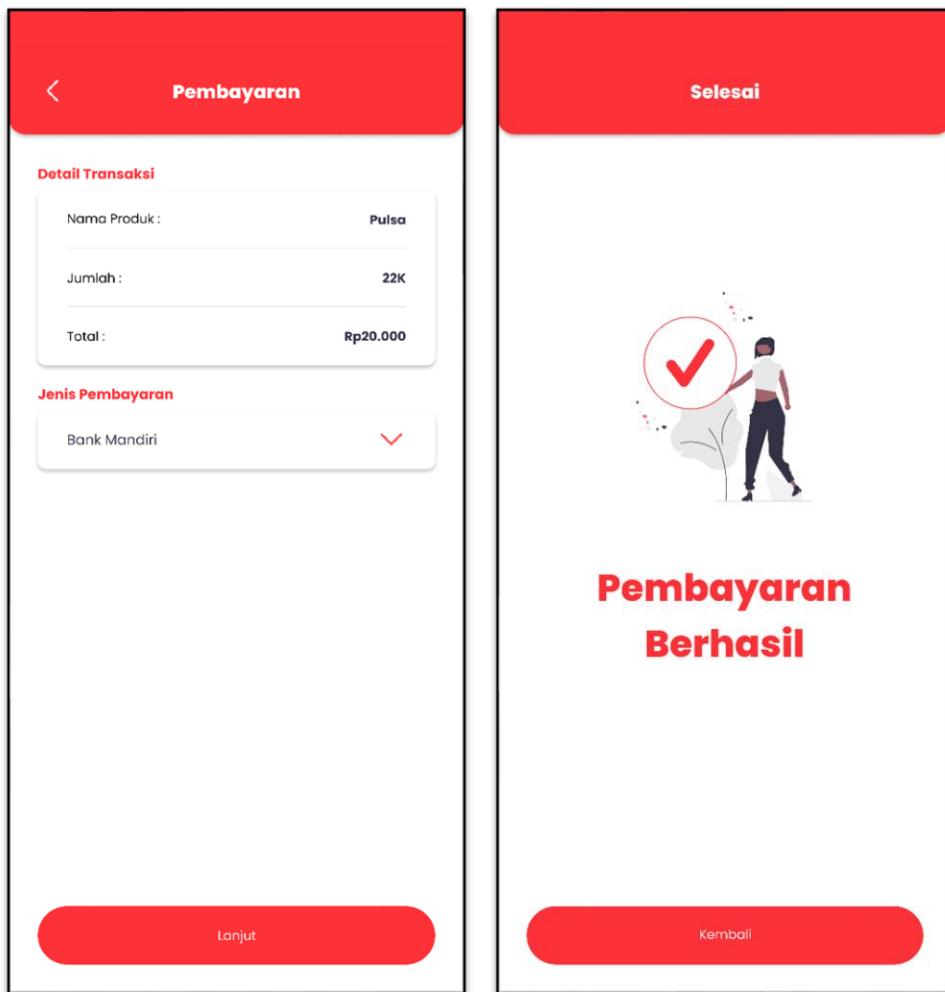
JOIN

```
produk ON detail_produk.id_produk = produk.id_produk
```

WHERE

```
detail_produk.kode_unik = LEFT('081234567890', 4);
```

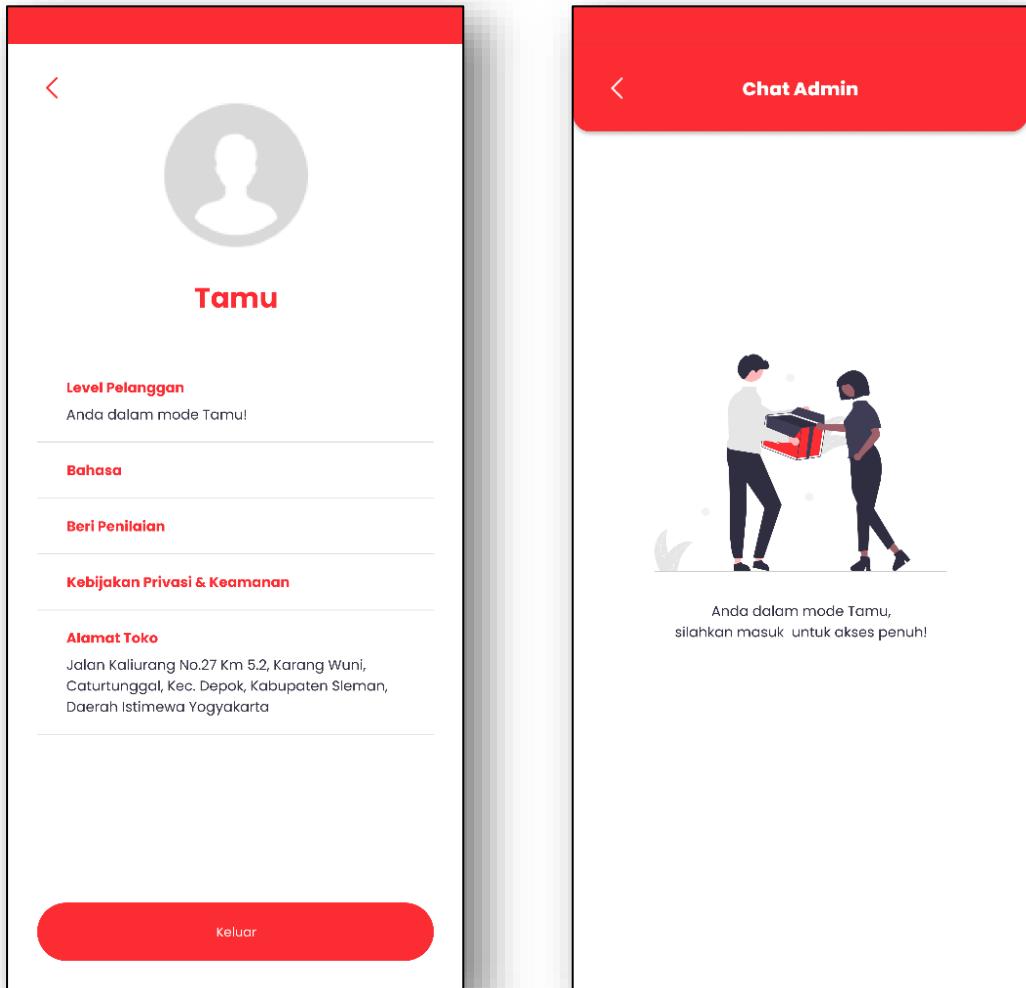
11. Halaman pembayaran

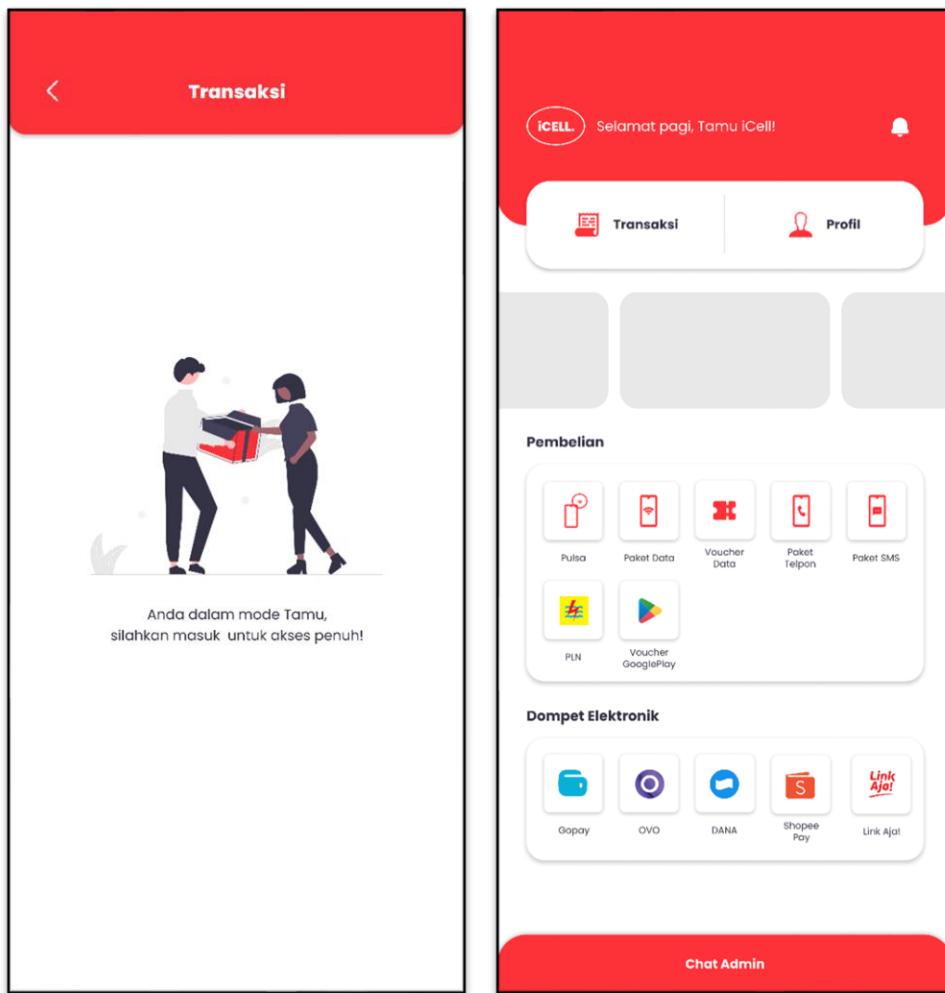


Query tampilan pembayaran:

```
INSERT INTO pembayaran
(id_bayar, id_transaksi, jenis_pembayaran, waktu_bayar, status)
VALUES ('BYR001', 'TRS001', 'Bank Mandiri', '2024-05-23 18:22:05',
'Berhasil');
```

12. Halaman mode tamu





QUERY MODE TAMU:

```
INSERT INTO tamu (id_tamu, loginTime, logoutTime)
VALUES ('TMU001', '2024-05-24 09:00:00', '2024-05-24 12:00:00');
```

```
INSERT INTO tamu (id_tamu, loginTime, logoutTime)
VALUES ('TMU001', NOW(), '2024-05-24 12:00:00');
```

```
INSERT INTO tamu (id_tamu, loginTime, logoutTime)
VALUES ('TMU001', CURRENT_TIMESTAMP, '2024-05-24 12:00:00');
```

<p>Judul Aplikasi</p> <h2 style="text-align: center;">APLIKASI RESERVASI RESTORAN</h2>	
<p>Dikembangkan Oleh</p> <ul style="list-style-type: none"> 1. 5220411235 – Ainur Dwi Utmawati 2. 5220411263 – Naufal Nur Faiq 3. 5220411270 – Didik Sukma Putra 4. 5220411279 – Andina Aura Hayuningtyas 5. 5220411292 – Shafira Nur Alfiah 	
<p>Deskripsi Singkat Instansi, Permasalahan, dan Solusi Aplikasi Yang Akan Dikembangkan [10 poin]</p> <p>Deskripsi Instansi:</p> <p>Downtown Diner adalah salah satu tempat casual dining dengan konsep unik dan menarik yang menawarkan sejuta kenikmatan western food, sekaligus membawa kita ke suasana Amerika di tahun 1950an yang terkesan vintage, terlihat dari berbagai ornamen-ornamen yang akan membawa para pengunjungnya masuk dalam dimensi tahun 50an. Seperti televisi dan radio kuno, mural-mural cantik ala jadul dan juga suasana kota New York di era tersebut. Downtown salah satu tempat nongkrong sekaligus tempat makan yang estetik di Kota Gudeg adalah Downtown Diner Jogja. Tempat makan ini juga menyediakan berbagai macam jenis maknanan dan minuman. Mulai dari yang halal hingga non halal. Downtown Diner merupakan tempat yang cocok bagi pengunjung yang menyukai menu internasional dan ingin merasakan nuansa seperti di luar negeri ketika menyantap hidangan.</p> <p>Downtown Diner Jogja berada di Jl. Seturan Baru No.2, RT.016/RW.095, Kledokan, Caturtunggal, Kec. Depok, Kabupaten Sleman, Daerah Istimewa Yogyakarta. tempat makan ini buka setiap hari. Tempat makan ini buka mulai dari hari Senin sampai Kamis pada pukul 10.00-23.00 WIB. Sedangkan di hari Jumat sampai Sabtu, tempat makan ini buka mulai pukul 10.00-00.00 WIB.</p> <p>Menu yang ditawarkan di sini yaitu menu internasional. Mulai dari burger, kentang, salad, dan lain-lain dengan plating yang indah. Rasanya pun tidak kalah dengan platingannya. Tetapi bagi pengunjung yang tidak menyukai menu barat tidak perlu khawatir, karena ada juga menu non-barat.</p>	

Permasalahan:

Beberapa permasalahan yang mungkin dihadapi oleh aplikasi reservasi restoran meliputi:

- Ketersediaan tempat:

Restoran mungkin memiliki keterbatasan jumlah meja atau kapasitas ruang, yang bisa membuat sulit bagi pengguna untuk menemukan tempat yang tersedia pada waktu yang diinginkan.

- Ketepatan waktu:

Ada kemungkinan bahwa reservasi yang dibuat oleh pengguna tidak selalu dihormati oleh restoran, menyebabkan keterlambatan atau bahkan pembatalan tanpa pemberitahuan.

- Pembaruan informasi:

Informasi tentang ketersediaan tempat, menu, atau jam operasional restoran mungkin tidak selalu diperbarui dengan akurat, yang dapat menyebabkan kebingungan atau ketidakpuasan bagi pengguna.

- Kebutuhan untuk pembayaran di muka:

Beberapa restoran meminta pembayaran di muka untuk reservasi, yang mungkin menjadi kendala bagi beberapa pengguna yang lebih suka membayar setelah makan

Solusi Aplikasi:

Berikut adalah beberapa solusi untuk mengatasi permasalahan dalam aplikasi reservasi restoran:

- Ketersediaan tempat yang transparan:

Memperbarui secara real-time ketersediaan tempat dan memberikan estimasi waktu tunggu kepada pengguna. Hal ini dapat membantu pengguna untuk memilih table yang memiliki tempat kosong sesuai dengan kebutuhan mereka.

- Konfirmasi reservasi yang tepat waktu:

Menerapkan sistem konfirmasi reservasi otomatis atau pemberitahuan langsung kepada pengguna untuk memastikan bahwa reservasi mereka telah diterima oleh restoran.

- Pembaruan informasi secara berkala:

Memastikan bahwa informasi tentang restoran, seperti menu, jam operasional, dan kebijakan reservasi, diperbarui secara berkala agar selalu akurat.

- Peningkatan kualitas layanan:

Mendorong restoran untuk meningkatkan kualitas layanan dengan memberikan pelatihan kepada staff, memperhatikan umpan balik pelanggan dan memperbaiki masalah yang diidentifikasi.

- Fleksibilitas pembayaran:

Memberikan pilihan kepada pengguna untuk membayar di muka atau setelah makan, serta menawarkan opsi pembayaran yang beragam seperti kartu kredit, transfer bank, atau dompet digital.

Pengguna Aplikasi [5 poin]

Berikut adalah deskripsi lengkap penggunaan aplikasi reservasi restoran:

- **Pendaftaran Pengguna:**

Pengguna pertama kali harus membuat akun di aplikasi dengan mengisi informasi pribadi seperti nama, alamat email, nomor telepon, dan kata sandi.

- **Pemilihan Tanggal dan Waktu:**

Pengguna dapat memilih tanggal dan waktu kunjungan yang diinginkan. Aplikasi akan menampilkan ketersediaan meja sesuai dengan preferensi pengguna.

- **Pemesanan Meja:**

Pengguna dapat memesan meja untuk jumlah tamu tertentu sesuai dengan kebutuhan mereka. Mereka juga dapat menambahkan permintaan khusus, seperti tempat duduk yang lebih tenang atau persyaratan diet.

- **Konfirmasi Reservasi:**

Setelah memilih tanggal, waktu, dan jumlah tamu, pengguna akan diminta untuk mengonfirmasi reservasi mereka. Setelah itu, pengguna akan menerima konfirmasi melalui email atau notifikasi dalam aplikasi.

- **Pengelolaan Reservasi:**

Pengguna dapat mengelola atau membatalkan reservasi mereka melalui aplikasi jika ada perubahan rencana. Mereka juga dapat melihat riwayat reservasi sebelumnya.

- **Penggunaan di Pihak Restoran**

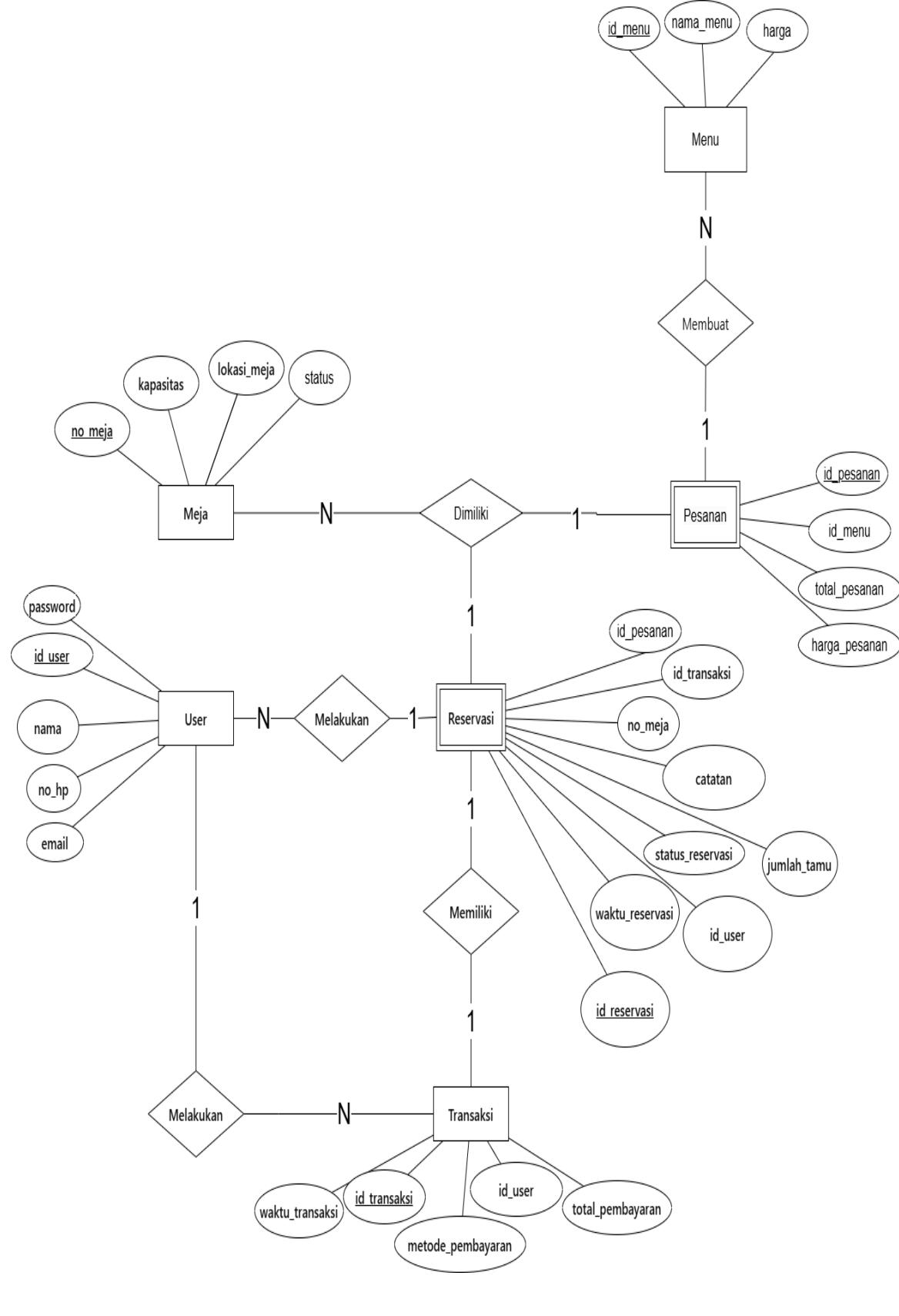
Pihak restoran juga dapat menggunakan aplikasi untuk mengelola reservasi, mengonfirmasi pesanan, dan memantau kapasitas restoran mereka.

Dengan aplikasi reservasi restoran ini, pengguna dapat dengan mudah merencanakan kunjungan ke restoran kami tanpa harus repot menghubungi restoran secara langsung atau menunggu antrian di tempat.

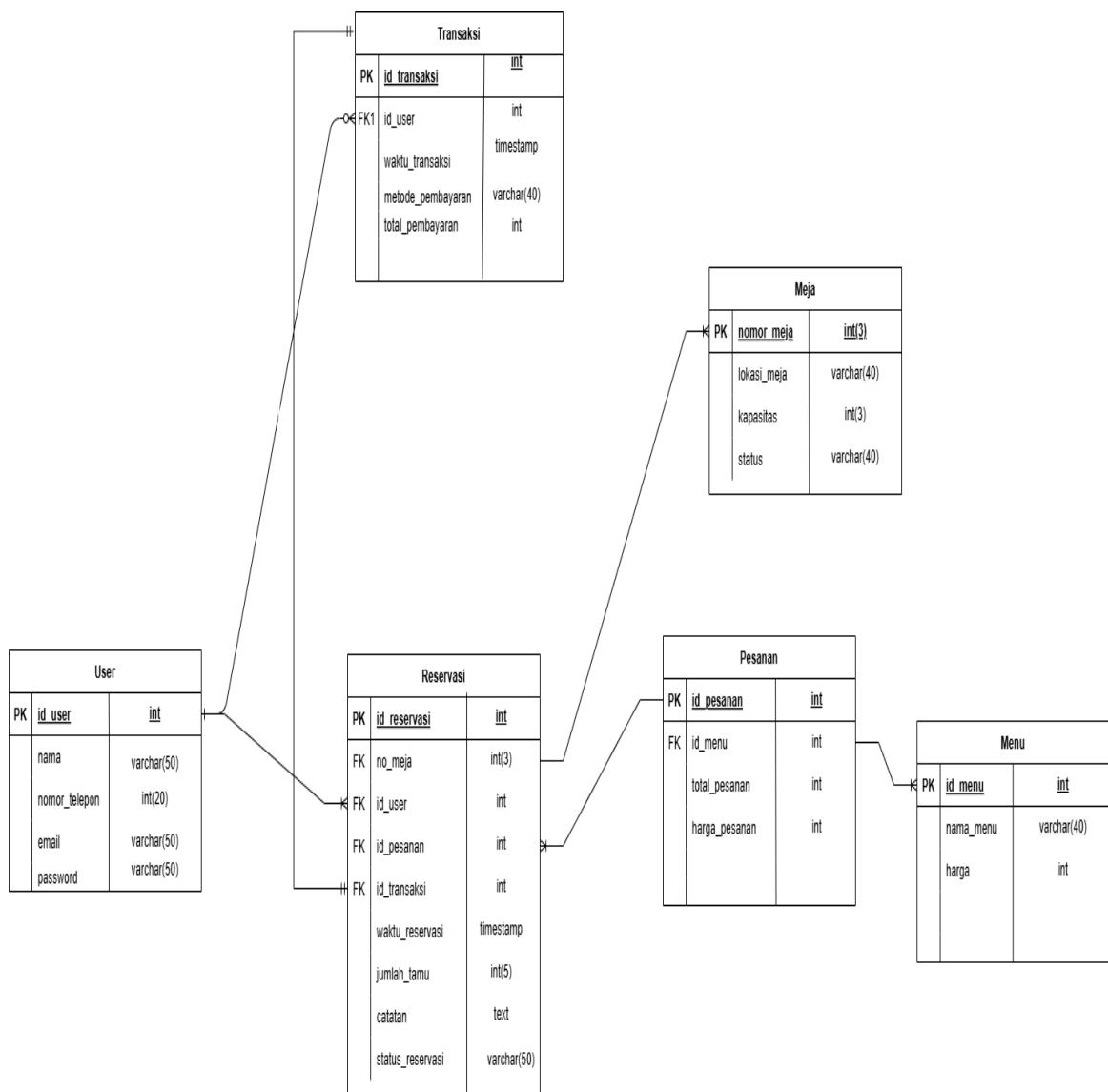
Target pengguna untuk aplikasi reservasi restoran Downtown Diner bisa mencakup:

- Orang-orang yang memiliki jadwal padat dan membutuhkan cara cepat dan mudah untuk memesan tempat di restoran tanpa harus menelepon dan mengantri.
- Wisatawan yang sedang mengunjungi kota dan ingin merencanakan makan malam di salah satu restoran populer di pusat kota.
- Pasangan yang mencari tempat makan yang nyaman dan berkesan untuk makan malam romantis di pusat kota.
- Orang-orang yang bekerja di sekitar pusat kota dan mencari tempat makan siang cepat atau tempat makan malam bersama rekan kerja.
- Orang-orang yang ingin mencoba makanan baru di restoran yang populer di pusat kota dan mencari pengalaman kuliner yang unik.
- Orang tua yang ingin merencanakan makan malam keluarga di restoran yang ramah keluarga di pusat kota.

Desain Entity-Relationship Diagram (ERD) [10 poin]



Desain Relasi Basis Data [10 poin]



Query Data Definition Language (DDL) [25 poin]

```
CREATE TABLE User (
    id_User INT AUTO_INCREMENT PRIMARY KEY,
    nama VARCHAR(50) NOT NULL,
    email VARCHAR(50) NOT NULL,
    password VARCHAR(50) NOT NULL,
    no_hp INT(20) NOT NULL
);

CREATE TABLE Meja(
    no_meja INT(3) PRIMARY KEY,
    kapasitas INT(3) NOT NULL,
    lokasi_meja VARCHAR(40) NOT NULL,
    status_meja VARCHAR(40) NOT NULL
);

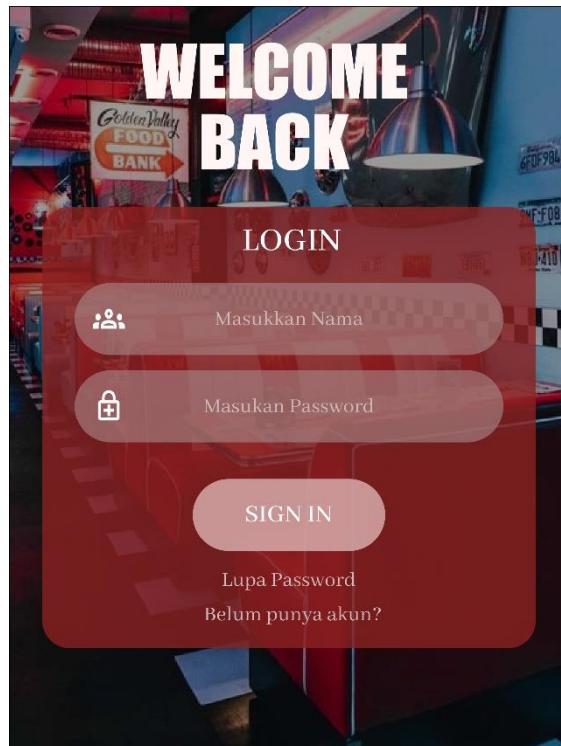
CREATE TABLE Menu(
    id_menu INT AUTO_INCREMENT PRIMARY KEY,
    nama_menu VARCHAR(40) NOT NULL,
    harga INT NOT NULL
);

CREATE TABLE Pesanan (
    id_pesanan INT AUTO_INCREMENT PRIMARY KEY,
    id_menu INT NOT NULL,
    total_pesanan INT NOT NULL,
    harga_pesanan INT NOT NULL,
    FOREIGN KEY (id_menu) REFERENCES Menu(id_menu)
);

CREATE TABLE Transaksi (
    id_transaksi INT AUTO_INCREMENT PRIMARY KEY,
    waktu_transaksi TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    metode_pembayaran VARCHAR(40) NOT NULL,
    total_pembayaran INT NOT NULL,
    id_user INT NOT NULL,
    FOREIGN KEY (id_user) REFERENCES User(id_user)
);
```

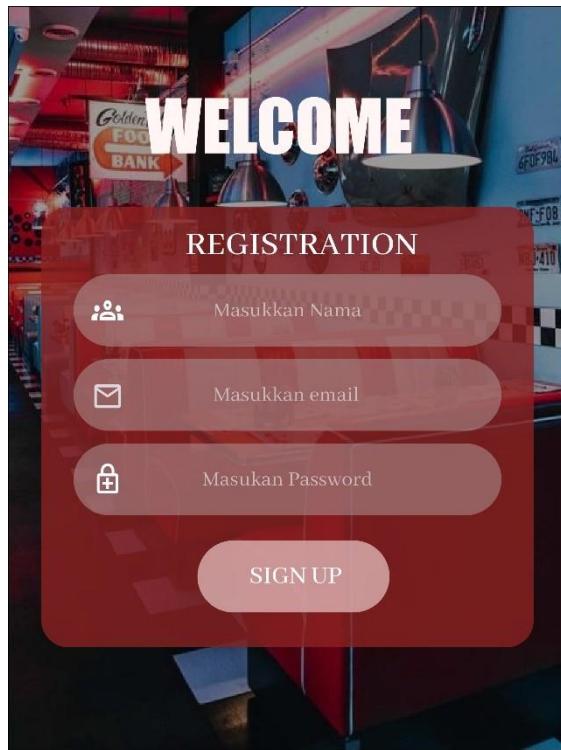
```
CREATE TABLE Reservasi(
    id_reservasi INT AUTO_INCREMENT PRIMARY KEY,
    waktu_reservasi TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    jumlah_tamu INT(5) NOT NULL,
    catatan TEXT,
    status_reservasi VARCHAR(50) NOT NULL,
    no_meja INT(3) NOT NULL,
    id_user INT NOT NULL,
    id_pesanan INT NOT NULL,
    id_transaksi INT NOT NULL,
    FOREIGN KEY (no_meja) REFERENCES Meja(no_meja),
    FOREIGN KEY (id_transaksi) REFERENCES Transaksi(id_transaksi),
    FOREIGN KEY (id_user) REFERENCES User(id_user),
    FOREIGN KEY (id_pesanan) REFERENCES Pesanan(id_pesanan)
);
```

Desain Interface (Wireframe/Mockup) Beserta Query-nya [poin 40]



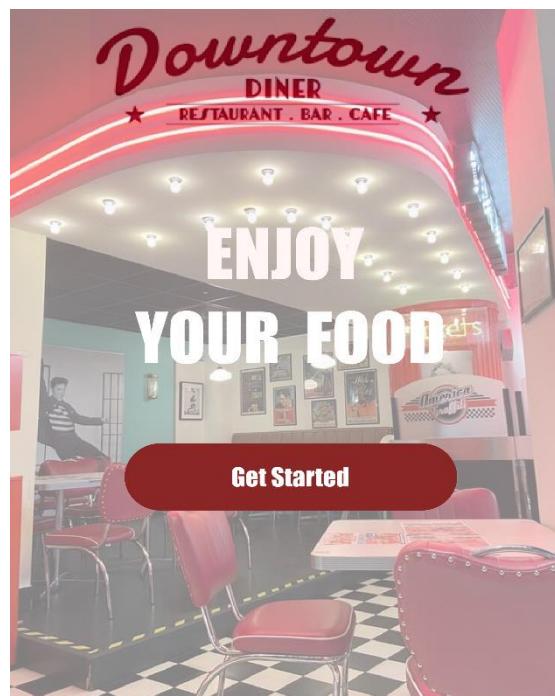
*SELECT * FROM User*

Where nama = 'input_nama' AND password = 'input_password';



USER Register:

```
INSERT INTO User (nama, email, password, no_hp)
VALUES ('yayat', 'yayat12@gmail.com', '12345678', '085123456789');
```



Reservation Details

Nama Pemesanan

Telp/Hp

Tanggal

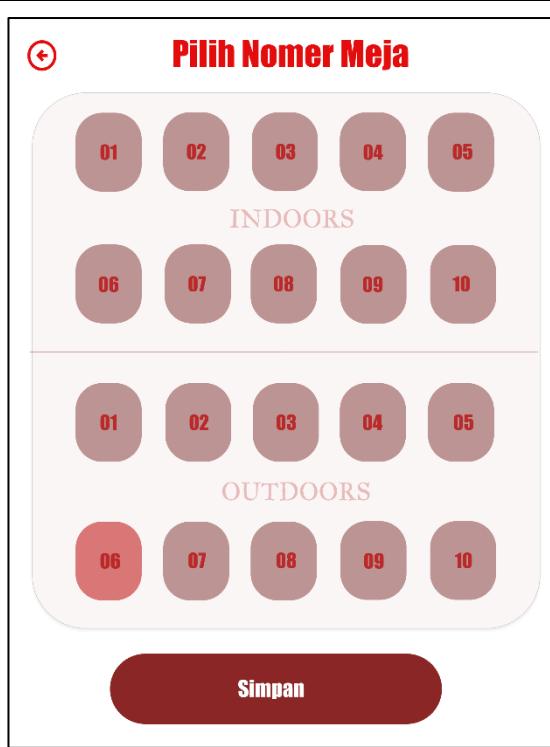
Nomer Meja

Catatan Tambahan

Confirm Reservation

RESERVASI:

```
INSERT INTO reservasi (jumlah_tamu, catatan, status_reservasi, no_meja,
id_user, id_transaksi, id_pesanan)
SELECT 4, 'makanan pedas', 'Dipesan', m.no_meja, u.id_user, t.id_transaksi,
p.id_pesanan
FROM meja m
JOIN user u ON u.id_user = m.id_user
JOIN pesanan p ON p.id_pesanan = p.id_pesanan
JOIN transaksi t ON t.id_transaksi = t.id_transaksi
WHERE m.status_meja = 'tersedia' and m.no_meja = 1 and t.id_transaksi = 1
and u.id_user = 1;
```



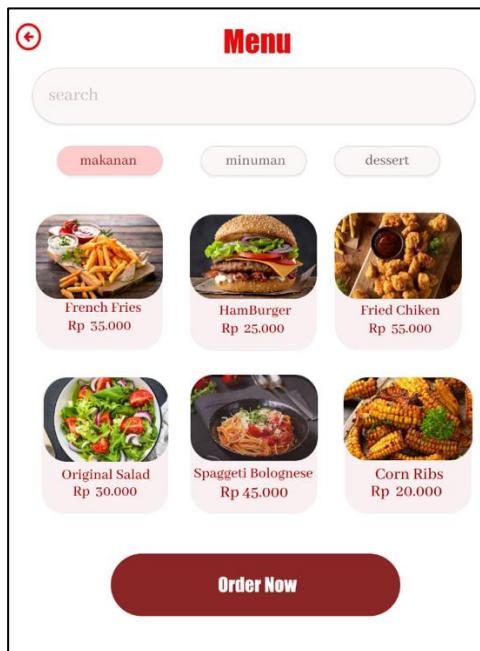
MEJA:

```
INSERT INTO Meja (no_meja, kapasitas,  
lokasi_meja, status_meja)  
VALUES ('01', '4', 'INDOORS', 'tersedia'),  
('02', '4', 'INDOORS', 'tersedia'),  
('03', '4', 'INDOORS', 'tersedia'),  
('04', '4', 'INDOORS', 'tersedia'),  
('05', '4', 'INDOORS', 'tersedia'),  
('06', '6', 'INDOORS', 'tersedia'),  
('07', '6', 'INDOORS', 'tersedia'),  
('08', '6', 'INDOORS', 'tersedia'),  
('09', '6', 'INDOORS', 'tersedia'),  
('10', '6', 'INDOORS', 'tersedia'),  
('1', '6', 'OUTDOORS', 'tersedia'),  
('2', '6', 'OUTDOORS', 'tersedia'),  
('3', '6', 'OUTDOORS', 'tersedia'),  
('4', '6', 'OUTDOORS', 'tersedia'),  
('5', '6', 'OUTDOORS', 'tersedia'),  
('6', '6', 'OUTDOORS', 'tersedia'),  
('7', '6', 'OUTDOORS', 'tersedia'),  
('8', '6', 'OUTDOORS', 'tersedia'),  
('9', '6', 'OUTDOORS', 'tersedia'),
```

```
('10', '6', 'OUTDOORS', 'tersedia');
```

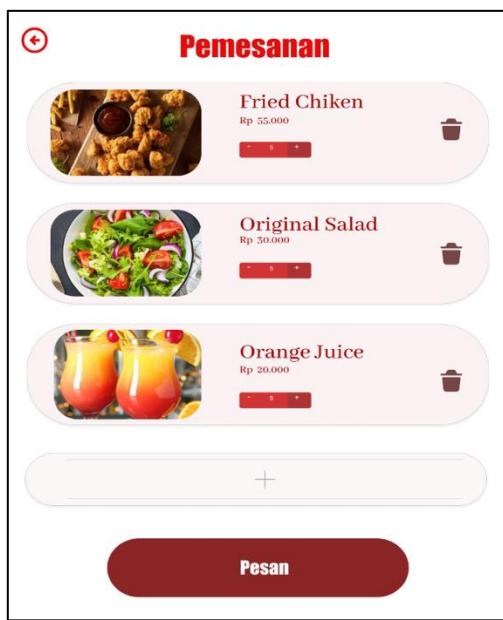
UPDATE Meja

```
-> SET status_meja = 'Tidak Tersedia'  
-> WHERE no_meja = 2 and lokasi_meja = 'OUTDOORS';
```



MENU:

```
INSERT INTO menu (nama_menu, harga)  
VALUES ('French Fries', 35000), ('HamBurger', 25000), ('Fried Chicken',  
55000), ('Original Salad', 30000), ('Spagheti Bolognese', 45000), ('Corn Ribs',  
20000);
```



PESANAN:

```
INSERT INTO pesanan (id_menu, total_pesanan, harga_pesanan)
-> SELECT id_menu, 2, harga * 2 FROM menu WHERE nama_menu = 'French
Fries';
```



TRANSAKSI:

```
INSERT INTO transaksi (metode_pembayaran, total_pembayaran, id_user)
SELECT 'tunai', p.harga_pesanan, u.id_user
FROM pesanan p
JOIN user u ON u.id_user = u.id_user
WHERE p.id_pesanan = 1 and u.id_user = 2;
```



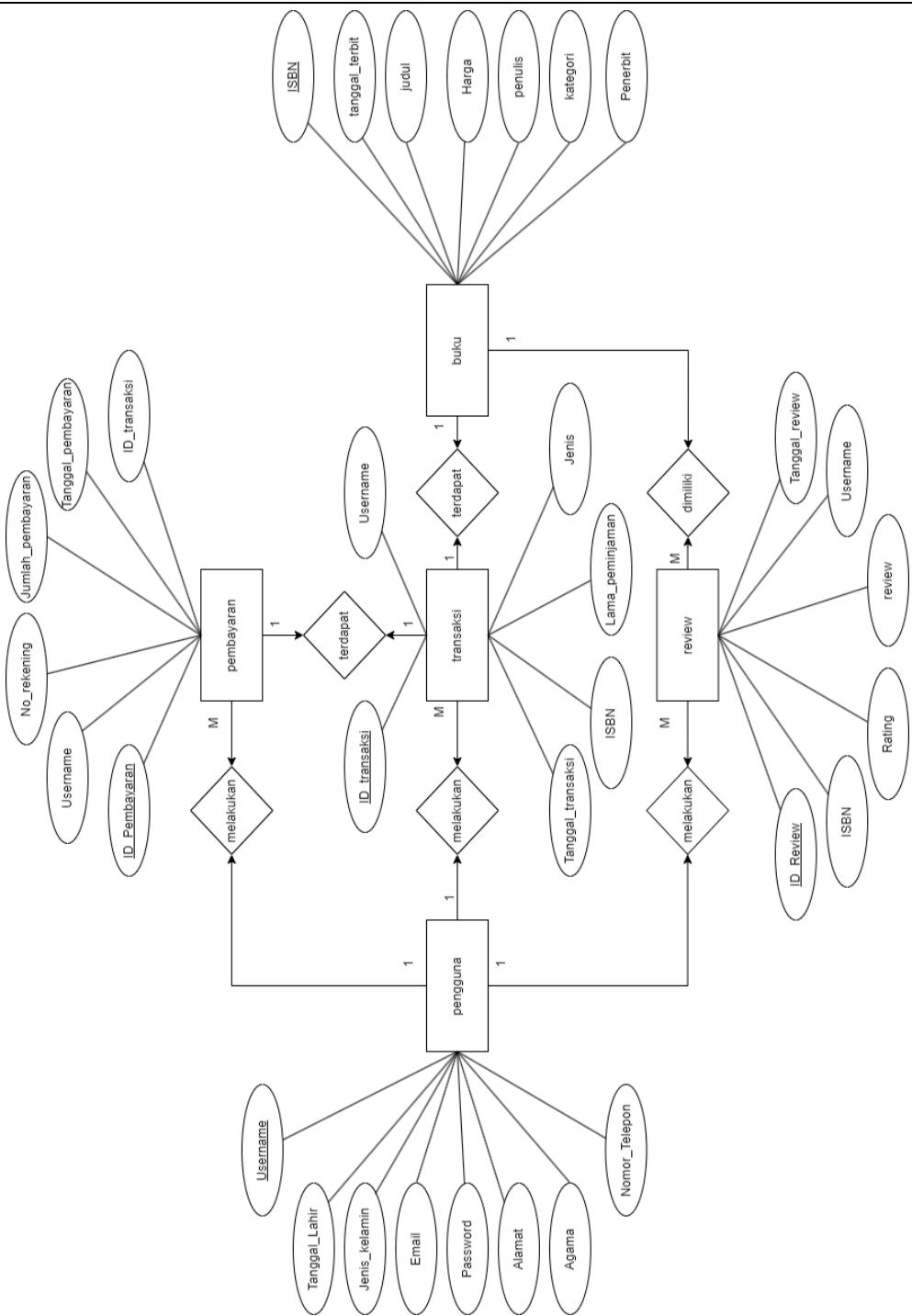
Judul Aplikasi	 E-LIBRARY GRHATAMA
Dikembangkan Oleh	
<ul style="list-style-type: none"> • 5220411259 - Hanan Alif Mustofa • 5220411255 - Ragil Gigih Utomo • 5220411280 - Muh Indra Tri Kusuma • 5220411256 - Muhammad Iqbal Ravelino • 5220411260 - Auxylium Briton Edyn Bunga 	
Deskripsi Singkat Instansi, Permasalahan, dan Solusi Aplikasi Yang Akan Dikembangkan [10 poin]	
<p>Grhatama Pustaka Nama tersebut diberikan langsung oleh Gubernur DIY untuk Gedung yang beralamat di Jalan Janti Banguntapan Bantul. Grhatama Pustaka merupakan penyedia layanan perpustakaan dari pemda. dengan tujuan utama guna meningkatkan literasi masyarakat di Yogyakarta. namun karena beberapa alasan seperti, jarak yang jauh, keterbatasan waktu atau lebih suka membaca secara daring, menyebabkan kurangnya pengunjung yang datang ke perpustakaan Grhatama. dengan latar belakang tersebut, kami menemukan salah satu Solusi untuk permasalah tersebut yaitu dengan membuat aplikasi perpustakaan online atau e-library. Tujuan utamanya adalah untuk memudahkan peminat membaca buku tanpa harus datang langsung atau meluangkan banyak waktu untuk membaca di tempat. Disisi lain hal ini memiliki segi positif lainnya seperti efisiensi, aksesibilitas untuk difabel, dan pemeliharaan yang mudah. dengan aplikasi ini pengguna dapat melakukan peminjaman/pembelian buku secara online kemudian pengguna juga dapat dengan mudah mencari buku yang ingin dibaca termasuk ketersediaan buku sehingga tentunya ini akan sangat membantu pengguna dan juga untuk bukunya sendiri akan tersedia dalam bentuk e-book sehingga akan selalu tersedia untuk dibeli/dipinjam oleh pengguna. aplikasi kami sendiri memiliki alur seperti berikut, pengguna melakukan login/registrasi kemudian pengguna dapat melihat koleksi buku, lalu pengguna dapat memilih opsi peminjaman/pembelian dan pemilihan durasi pinjam lalu pengguna dapat melakukan pembayaran secara online kemudian untuk peminjaman saat sudah habis waktu peminjaman buku akan otomatis ditarik kemudian pengguna dapat memberikan rating dan review yang bersifat opsional hal tersebut dapat membantu pengguna lainnya sebagai pertimbangan meminjam/membeli. Untuk fitur lainnya dalam aplikasi ini terdapat</p>	

koleksi buku yang sudah dibeli/dipinjam kemudian rating dan review buku dari pengguna. Sehingga, dengan adanya aplikasi ini selain dapat memudahkan pengguna juga dapat meningkatkan literasi Masyarakat Yogyakarta yang mana merupakan tujuan utama dari Grhatama Pustaka itu sendiri.

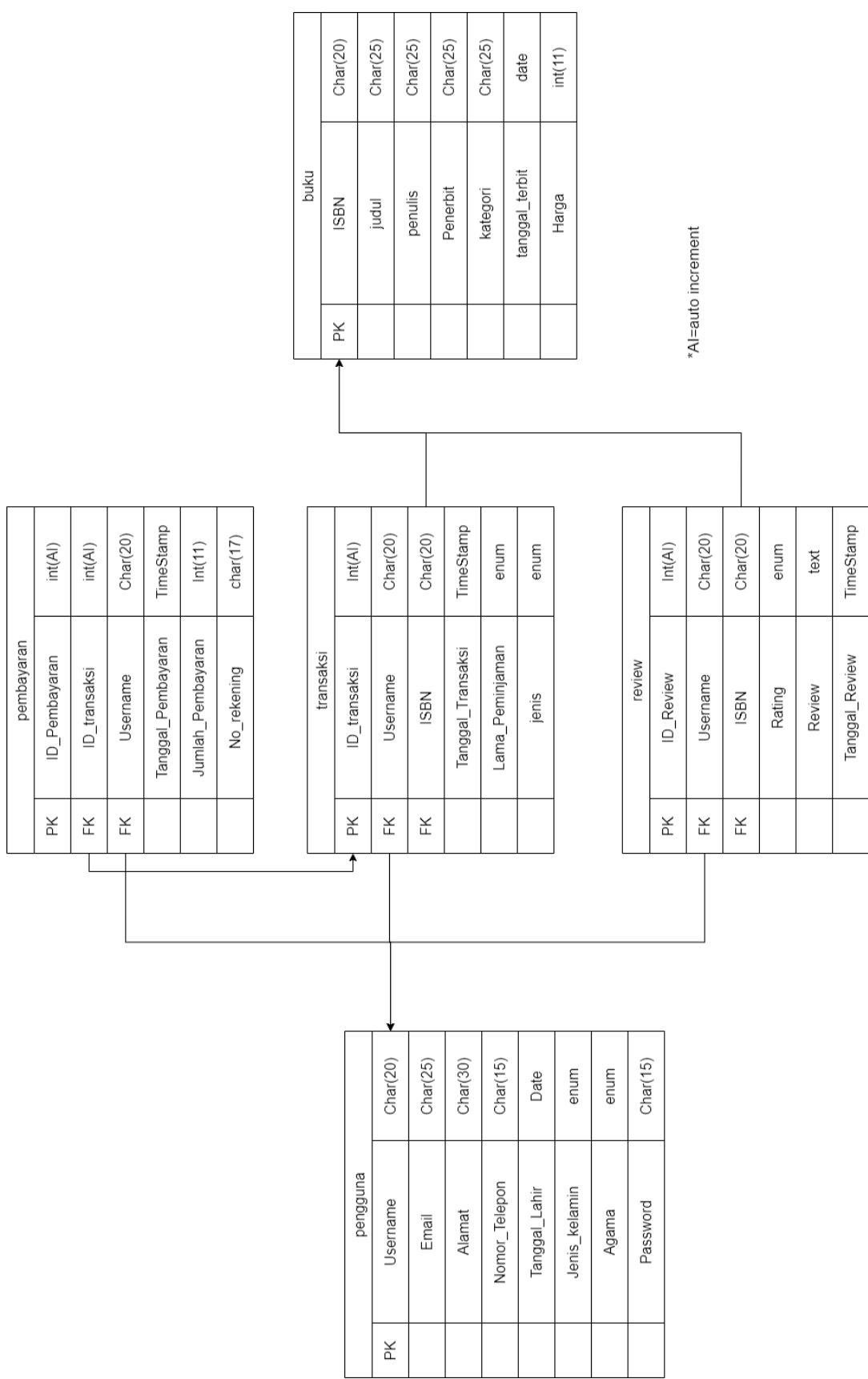
Pengguna Aplikasi [5 poin]

Masyarakat

Desain Entity-Relationship Diagram (ERD) [10 poin]



Desain Relasi Basis Data [10 poin]



*AI=auto increment

Query Data Definition Language (DDL) [25 poin]

```
-- phpMyAdmin SQL Dump
-- version 5.2.1
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1
-- Generation Time: May 24, 2024 at 04:34 AM
-- Server version: 10.4.28-MariaDB
-- PHP Version: 8.2.4

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET
@OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET
@OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET
@OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

-- Database: `grhatama`


-- Table structure for table `buku`


CREATE TABLE `buku` (
  `ISBN` char(20) NOT NULL,
  `judul` char(25) NOT NULL,
  `penulis` char(25) NOT NULL,
  `Penerbit` char(25) NOT NULL,
  `kategori` char(25) NOT NULL,
  `tanggal_terbit` date NOT NULL,
  `Harga` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
--
```

```
-- Table structure for table `pembayaran`


CREATE TABLE `pembayaran` (
  `ID_Pembayaran` int(11) NOT NULL,
  `ID_transaksi` int(11) NOT NULL,
  `Username` char(20) NOT NULL,
  `Tanggal_Pembayaran` timestamp NOT NULL DEFAULT current_timestamp(),
  `Jumlah_Pembayaran` int(11) NOT NULL,
  `No_rekening` int(17) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
--


-- Table structure for table `pengguna`


CREATE TABLE `pengguna` (
  `Username` char(20) NOT NULL,
  `Password` char(15) NOT NULL,
  `Alamat` char(30) NOT NULL,
  `Nomor_telepon` int(16) NOT NULL,
  `Tanggal_Lahir` date NOT NULL,
  `Jenis_kelamin` enum('L','P') NOT NULL,
  `Agama` enum('Islam','Kristen','Budha','Hindu','Konghucu') NOT NULL,
  `Email` char(25) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
--


-- Table structure for table `review`


CREATE TABLE `review` (
  `ID_Review` int(11) NOT NULL,
  `Username` char(20) NOT NULL,
  `ISBN` char(20) NOT NULL,
  `Rating` enum('1','2','3','4','5') NOT NULL,
  `Review` text NOT NULL,
  `Tanggal_Review` timestamp NOT NULL DEFAULT current_timestamp() ON
UPDATE current_timestamp()

```

```

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
-- -----
-- Table structure for table `transaksi`


CREATE TABLE `transaksi` (
  `ID_transaksi` int(11) NOT NULL,
  `Username` char(20) NOT NULL,
  `ISBN` char(20) NOT NULL,
  `Tanggal_transaksi` timestamp NOT NULL DEFAULT current_timestamp(),
  `Lama_Peminjaman` enum('1','2','3','4','5') DEFAULT NULL,
  `jenis` enum('sewa','beli') NOT NULL
) ;

-- Indexes for dumped tables--


-- Indexes for table `buku`


ALTER TABLE `buku`
  ADD PRIMARY KEY (`ISBN`);


-- Indexes for table `pembayaran`


ALTER TABLE `pembayaran`
  ADD PRIMARY KEY (`ID_Pembayaran`),
  ADD UNIQUE KEY `Username` (`Username`),
  ADD UNIQUE KEY `ID_transaksi` (`ID_transaksi`);


-- Indexes for table `pengguna`


ALTER TABLE `pengguna`
  ADD PRIMARY KEY (`Username`);


-- Indexes for table `review`


ALTER TABLE `review`
  ADD PRIMARY KEY (`ID_Review`),
  ADD UNIQUE KEY `Username` (`Username`),
  ADD UNIQUE KEY `ISBN` (`ISBN`);


-- Indexes for table `transaksi`
```

```

ALTER TABLE `transaksi`
    ADD PRIMARY KEY (`ID_transaksi`),
    ADD KEY `Username` (`Username`),
    ADD KEY `ISBN` (`ISBN`);

ALTER TABLE `transaksi`
    ADD CONSTRAINT chk_lama_peminjaman
        CHECK (`jenis` = 'beli' OR (`jenis` = 'sewa' AND `Lama_Peminjaman` IS NOT NULL));
-- AUTO_INCREMENT for dumped tables

-- AUTO_INCREMENT for table `pembayaran`
ALTER TABLE `pembayaran`
    MODIFY `ID_Pembayaran` int(11) NOT NULL AUTO_INCREMENT;

-- AUTO_INCREMENT for table `review`
ALTER TABLE `review`
    MODIFY `ID_Review` int(11) NOT NULL AUTO_INCREMENT,
    AUTO_INCREMENT=2;

-- AUTO_INCREMENT for table `transaksi`
ALTER TABLE `transaksi`
    MODIFY `ID_transaksi` int(11) NOT NULL AUTO_INCREMENT;
-- Constraints for dumped tables
-- Constraints for table `pembayaran`
ALTER TABLE `pembayaran`
    ADD CONSTRAINT `pembayaran_ibfk_1` FOREIGN KEY (`ID_transaksi`)
REFERENCES `transaksi` (`ID_transaksi`),
    ADD CONSTRAINT `pembayaran_ibfk_2` FOREIGN KEY (`Username`)
REFERENCES `pengguna` (`Username`);

-- Constraints for table `review`
ALTER TABLE `review`
    ADD CONSTRAINT `review_ibfk_1` FOREIGN KEY (`ISBN`) REFERENCES
`buku` (`ISBN`),
    ADD CONSTRAINT `review_ibfk_2` FOREIGN KEY (`Username`)
REFERENCES `pengguna` (`Username`);

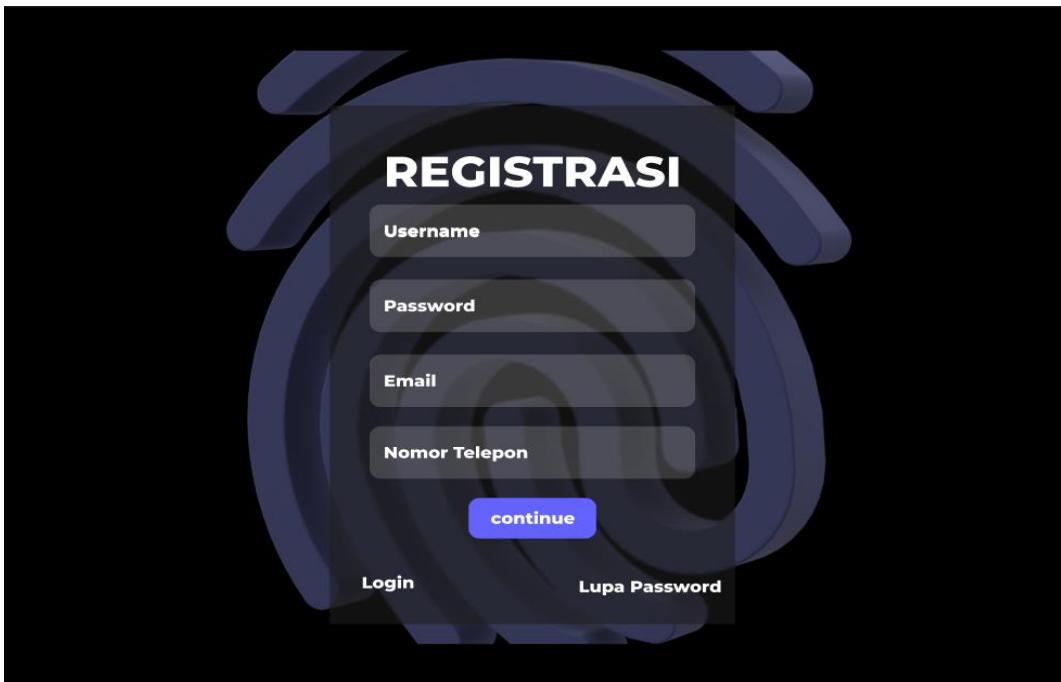
```

```
-- Constraints for table `transaksi`
ALTER TABLE `transaksi`
    ADD CONSTRAINT `transaksi_ibfk_1` FOREIGN KEY (`Username`)
REFERENCES `pengguna` (`Username`),
    ADD CONSTRAINT `transaksi_ibfk_2` FOREIGN KEY (`ISBN`) REFERENCES
`buku` (`ISBN`);
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT
*/;
/*!40101 SET
CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION
*/;
```

Desain Interface (Wireframe/Mockup) Beserta Query-nya [poin 40]

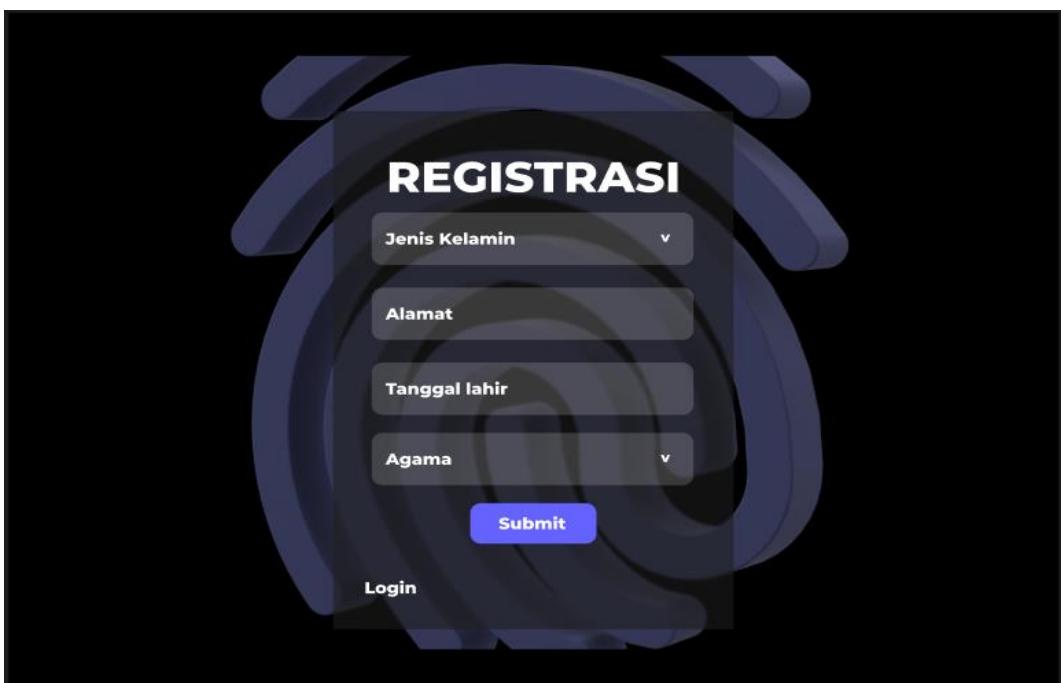
Registrasi



Query :

```
INSERT INTO pengguna (Username, Password, Email,Nomor_telepon)
VALUES ('Username_input', 'Password_input', 'Email_input','Nomor
Telepon_input')
```

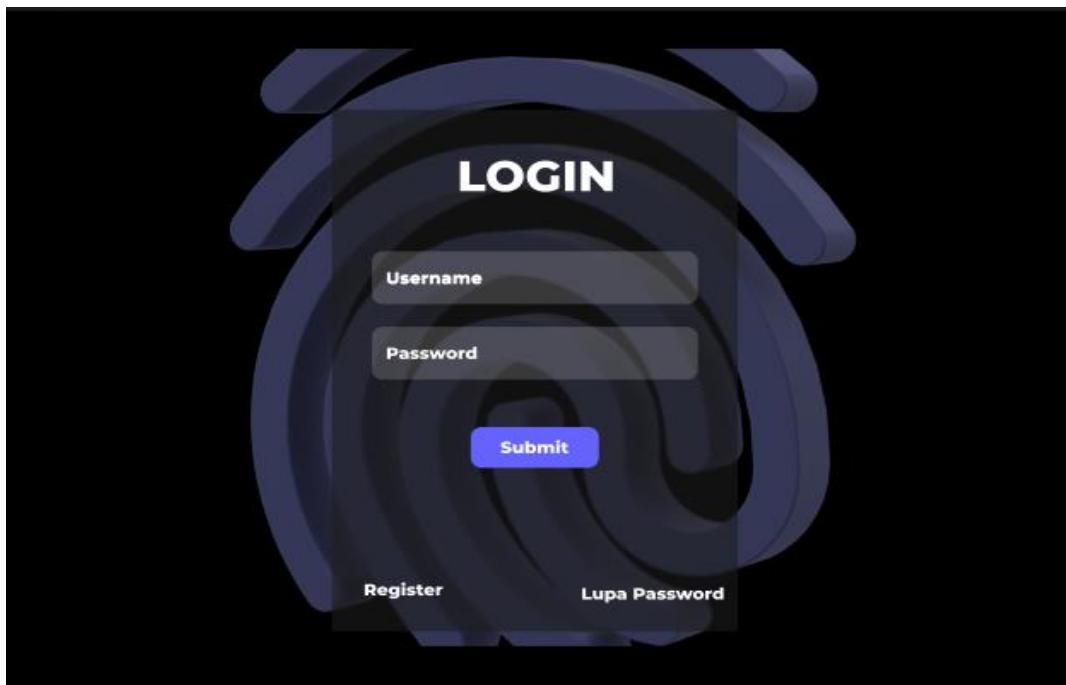
Registrasi



Query : `INSERT INTO pengguna (Jenis_kelamin, Alamat, Tanggal_lahir,Agama)`

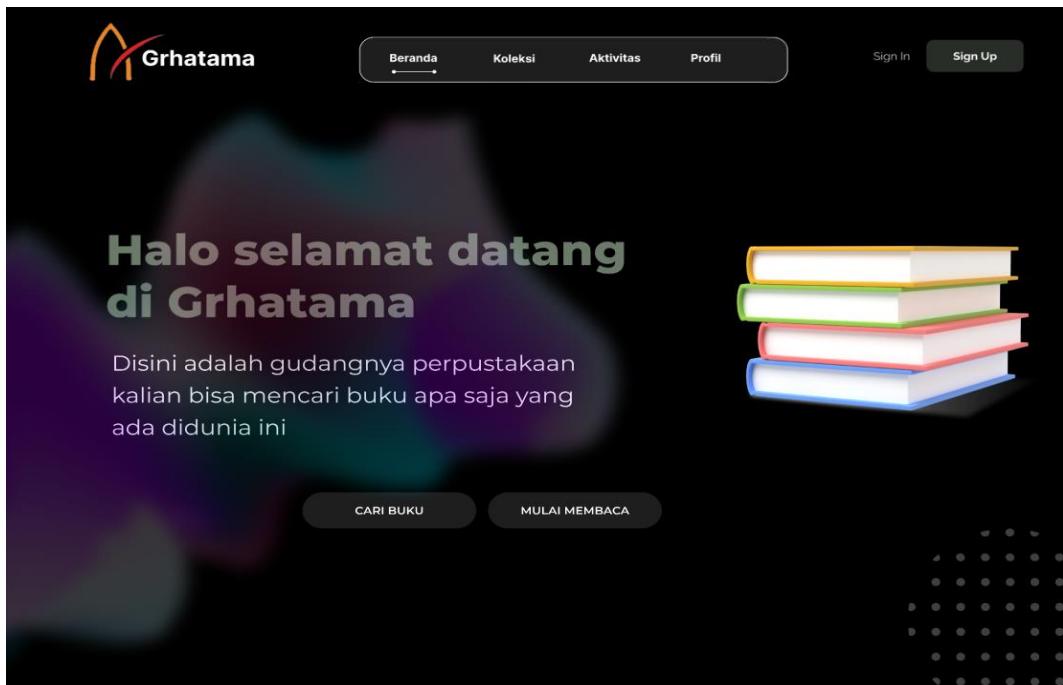
```
VALUES ('jenis_kelamin_input', 'alamat_input',
'tanggal_lahir_input','Agama_input')
```

Login

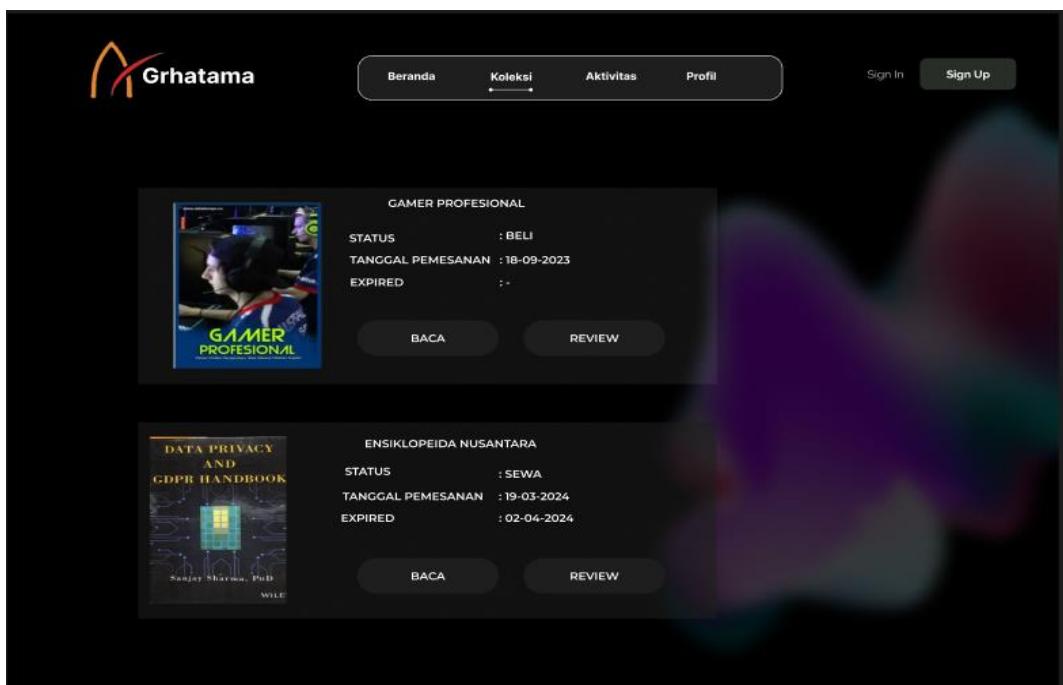


```
if request.method == 'POST':
    username = request.form['username']
    password = request.form['password']
    user = User.query.filter_by(username=username).first()
    if user and check_password_hash(user.password, password):
        session['username'] = username
```

Beranda



Koleksi



Query :

```
SELECT buku.judul,CONCAT('STATUS : ', transaksi.jenis) AS  
status,CONCAT('TANGGAL PEMESANAN : ',  
DATE_FORMAT(transaksi.Tanggal_transaksi, '%d-%m-%Y')) AS  
tanggal_pemesanan,  
CASE
```

```

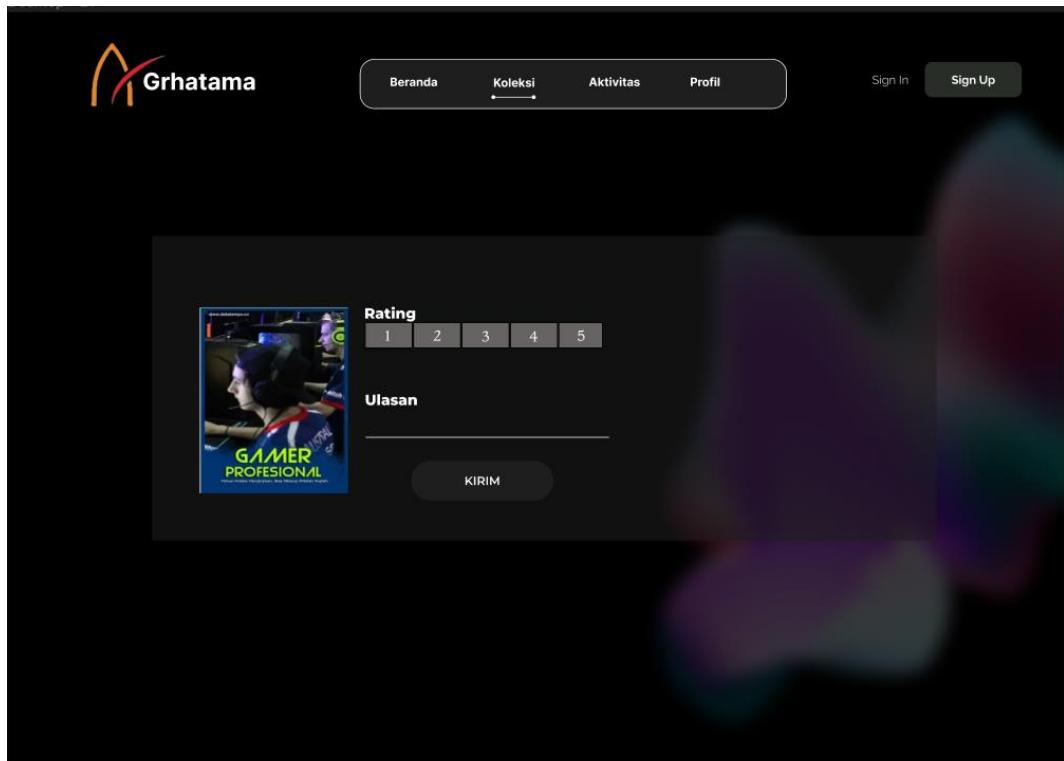
WHEN transaksi.jenis = 'sewa' THEN CONCAT('EXPIRED : ',
DATE_FORMAT(DATE_ADD(transaksi.Tanggal_transaksi, INTERVAL 1 WEEK),
'%d-%m-%Y'))
ELSE ""
END AS expired
FROM transaksi
JOIN buku ON buku.ISBN = transaksi.ISBN
WHERE
transaksi.Username = :username";

```

Note :

```
username = session['Username']
```

Rating/Review



Query :

```

INSERT INTO review(Rating, Review, Tanggal_review, Username, ISBN)
VALUES ('5','bagus',NOW(),'username','isbn')

```

Note:

```
username = session['Username']
```

```
isbn = request.form['isbn']
```

Aktivitas

The screenshot shows the 'Aktivitas' section of the Ghatama website. At the top, there is a navigation bar with the Ghatama logo, 'Beranda', 'Koleksi', 'Aktivitas' (which is highlighted with a double arrow), and 'Profil'. There are also 'Sign In' and 'Sign Up' buttons. Below the navigation bar, the title 'KOLEKSI TERPOPULER' is displayed in large white letters. Four book cards are shown in a grid:

- ENSIKLOPEDIA NUSANTARA**
by ABSTRAK
IGREA SISWANTO DKK
2022-10-10
- MANAJEMEN TERITORIAL**
by ABSTRAK
DWI JOKO SISWANTO & TEGOR
2021-11-23
- GAMER PROFESIONAL**
by ABSTRAK
HITLER
2018-3-10
- DATA PRIVACY AND GDPR HANDBOOK**
by ABSTRAK
SANJAY SHARMA, PHD
2012-10-10

At the bottom center of the grid, there is a button labeled 'LIHAT SEMUA'.

SELECT judul, kategori, Penulis, tanggal_terbit

FROM buku

ORDER BY RAND()

LIMIT 4;

Aktivitas (lihat buku)

The screenshot shows a detailed view of a book card for 'ENSIKLOPEDIA NUSANTARA'. The book cover is red with illustrations of traditional Indonesian figures and animals. To the right of the cover, the book's details are listed:

ENSIKLOPEDIA NUSANTARA
978-602-8519-93-9
ABSTRAK
IGREA SISWANTO DKK
TANGGAL 2022-10-10
GRAMEDIA
RP65.000

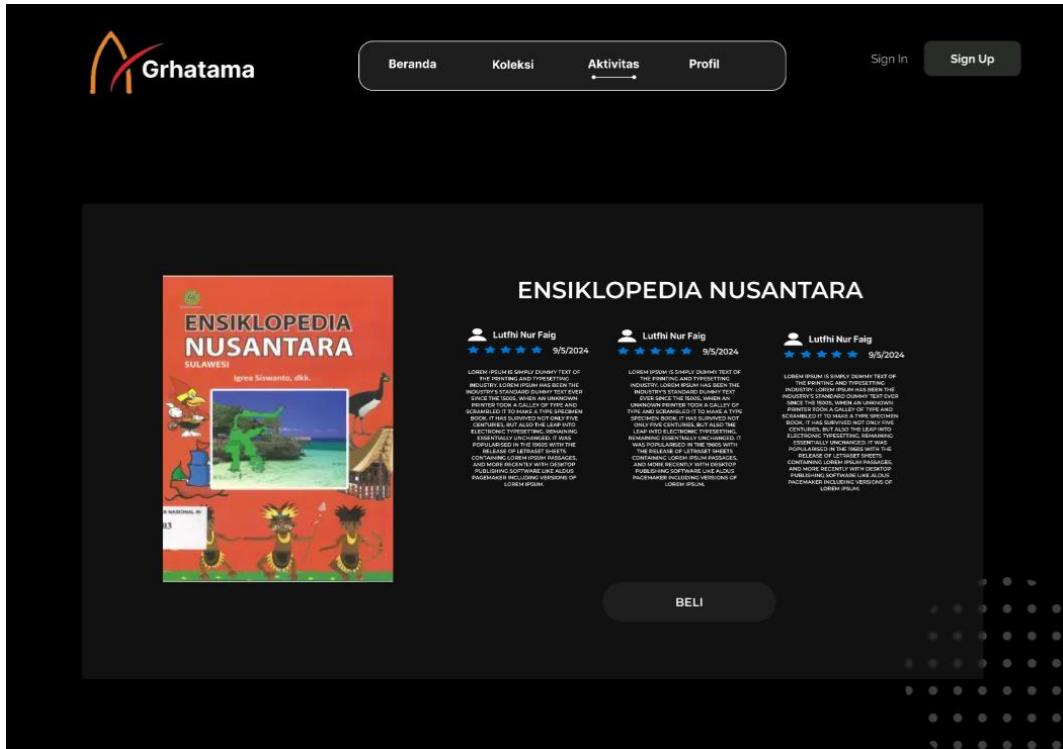
Below the book details are two buttons: 'REVIEW' and 'BELI'.

```
SELECT judul, ISBN, kategori, penulis, CONCAT('tanggal ', tanggal_terbit), penerbit,
CONCAT('Rp ', FORMAT(Harga, 0))
FROM buku WHERE ISBN = :isbn;
```

Note : isbn =

```
request.form['isbn'] #diambil sesui dengan buku yg dipilih
```

Ativitas(Review buku)



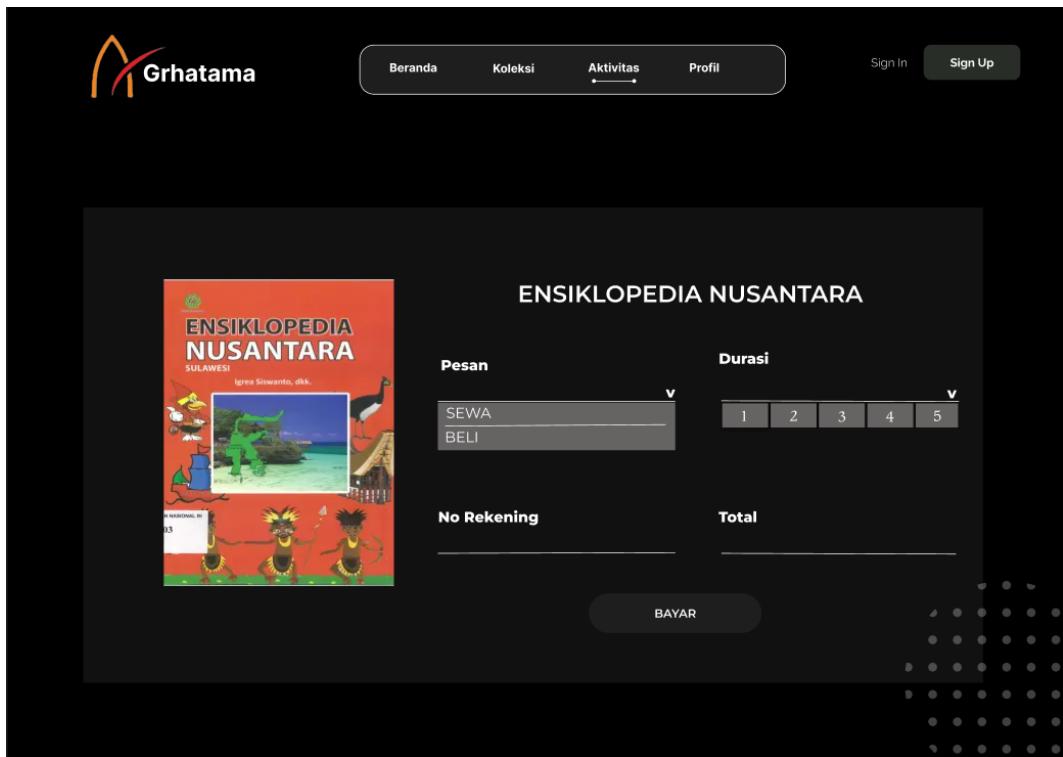
Query :

```
SELECT buku.judul, review.Username, review.Tanggal_Review, review.Rating,
review.Review FROM review WHERE ISBN = :isbn;
JOIN buku ON buku.ISBN=review.ISBN
ORDER BY ASC()
LIMIT 3;
```

Note:

```
daftar_buku = [{isbn: '1234567890', 'judul': 'Buku A'}, {'isbn': '0987654321',
'judul': 'Buku B'}] #diambil dari database buku
isbn = request.form['isbn']
```

Aktivitas(transaksi)



Query:

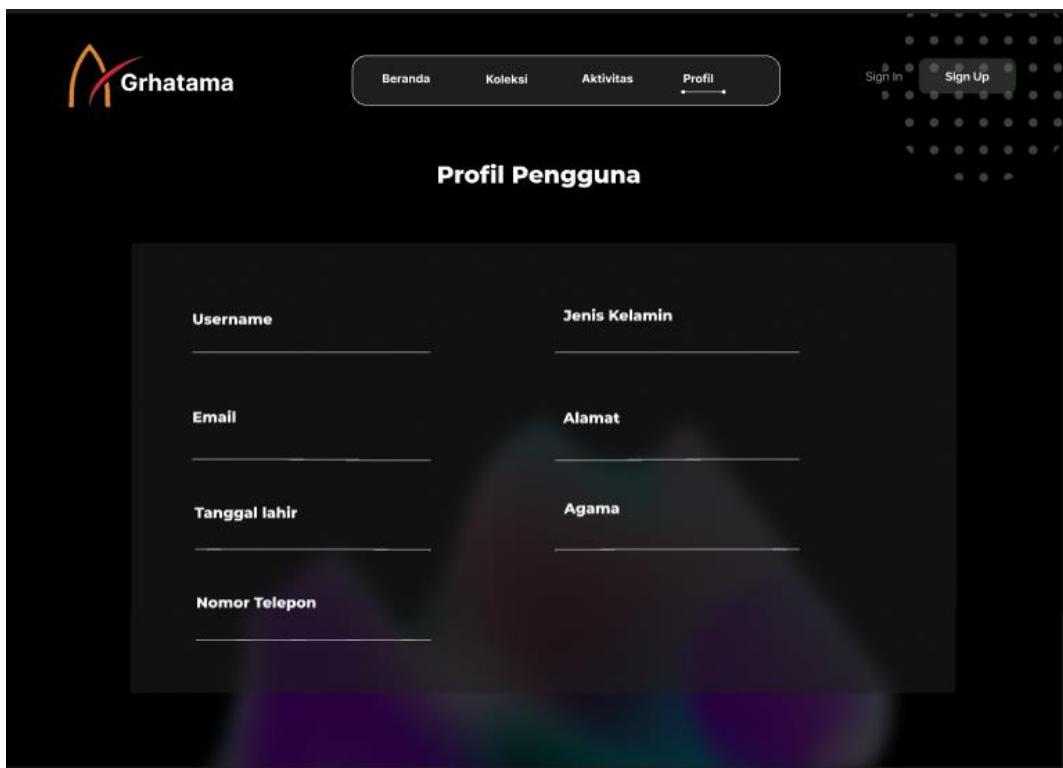
```
INSERT INTO transaksi (jenis, no_rekening, Username, ISBN, Lama_peminjaman)
VALUES ('sewa', '1234567890', 'username', 'isbn', '2');

SELECT
CASE
WHEN transaksi.jenis = 'sewa' THEN buku.Harga
ELSE transaksi.Lama_peminjaman*5000
END as Total
FROM transaksi
JOIN buku on buku.ISBN = transaksi.ISBN;
```

Note:

```
username = session['Username']
daftar_buku = [{isbn: '1234567890', judul: 'Buku A'}, {isbn: '0987654321',
'judul': 'Buku B'}] #diambil dari database buku
isbn = request.form['isbn']
```

Profil



Query :

```
SELECT Username, Email, Tanggal_Lahir AS 'Tanggal lahir',
Nomor_telepon AS 'Nomor Telepon',
Jenis_kelamin AS 'Jenis Kelamin', Alamat, Agama
FROM pengguna
WHERE
Username = :username";
```

Note :

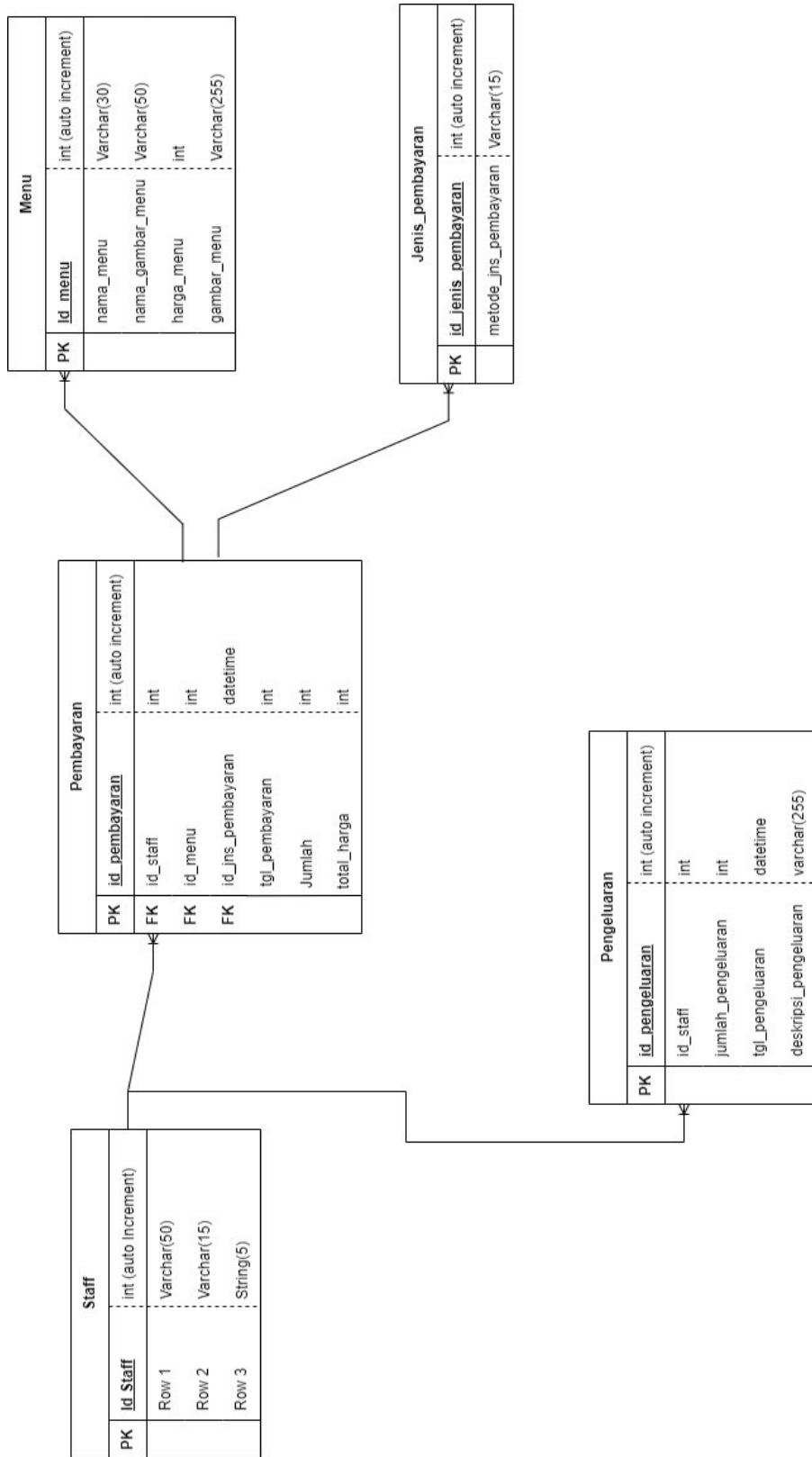
```
username = session['Username']
```

Judul Aplikasi	MANAGEMENT OMSET	
Dikembangkan Oleh		
	<ol style="list-style-type: none"> 1. 5220411261 – Febrian Aditya 2. 5220411262 – Afrizal Halim Naindhy Tuasikal 3. 5220411265 – I Nyoman Darma Yoga 4. 5220411274 – Anandava Eka Buana Baskara 5. 5220411288 – Muhammad Shiddiq 	
Deskripsi Singkat Instansi, Permasalahan, dan Solusi Aplikasi Yang Akan Dikembangkan [10 poin]		
Deskripsi :		
	<p>Posh Coffee merupakan perusahaan yang bergerak di bidang Makanan dan Minuman (FnB). Perusahaan ini menargetkan pasar pelajar dan mahasiswa untuk mengerjakan tugas mereka. Dengan fokus pada memberikan lingkungan yang kondusif untuk aktivitas akademik, Posh Coffee bertujuan untuk menjadi tempat yang nyaman di mana para pelajar dapat dengan tekun menyelesaikan tugas-tugas akademik mereka sambil menikmati minuman dan makanan berkualitas tinggi.</p>	
Permasalahan :		
	<p>Setiap Pengeluaran dan pemasukan omset per shift antar pegawai masih dilakukan secara manual dan tidak transparan, sehingga menyebabkan perhitungan omset secara total sulit untuk di kalkulasi.</p>	
Solusi :		
	<p>Membuat sistem manajemen omset yang di dapatkan dari setiap shift pegawai secara digital dan otomatis, sehingga data omset yang dimiliki oleh perusahaan lebih terstruktur, transparan, dan memiliki kejelasan antara uang yang keluar dan masuk</p>	
Pengguna Aplikasi [5 poin]		
	Karyawan	

Desain Entity-Relationship Diagram (ERD) [10 poin]



Desain Relasi Basis Data [10 poin]



Query Data Definition Language (DDL) [25 poin]

```
CREATE TABLE staff (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nama_staff VARCHAR(50) NOT NULL,
    nomortlp_staff VARCHAR(20) NOT NULL,
    status ENUM('Off', 'On') NOT NULL
);

CREATE TABLE menu (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nama_menu VARCHAR(50) NOT NULL,
    nama_gambar_menu VARCHAR(255) NOT NULL,
    harga_menu INT NOT NULL,
    gambar_menu TEXT NOT NULL
);

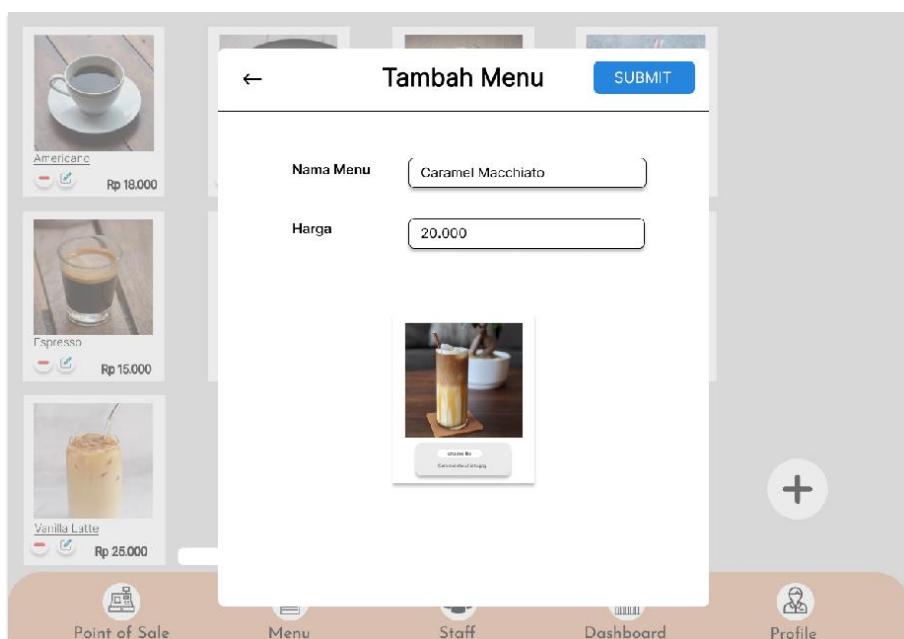
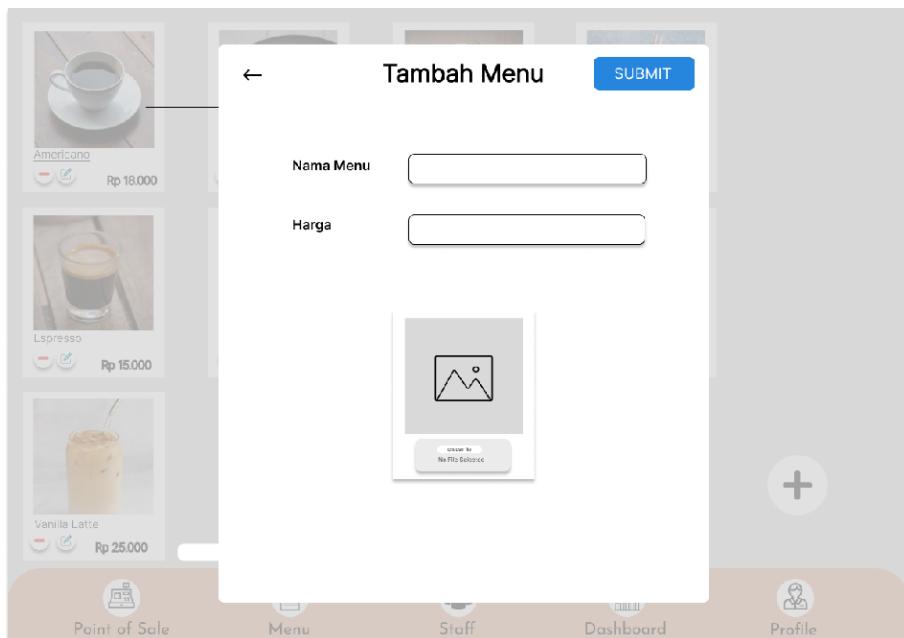
CREATE TABLE jenis_pembayaran (
    id INT AUTO_INCREMENT PRIMARY KEY,
    metode_jns_pembayaran VARCHAR(255) NOT NULL
);

CREATE TABLE pembayaran (
    id INT AUTO_INCREMENT PRIMARY KEY,
    id_staff INT NOT NULL,
    id_menu INT NOT NULL,
    id_jns_pembayaran INT NOT NULL,
    tgl_pembayaran DATETIME NOT NULL,
    jumlah INT NOT NULL,
    total_harga INT NOT NULL,
    FOREIGN KEY (id_staff) REFERENCES staff(id),
    FOREIGN KEY (id_menu) REFERENCES menu(id),
    FOREIGN KEY (id_jns_pembayaran) REFERENCES jenis_pembayaran(id)
);

CREATE TABLE pengeluaran (
    id INT AUTO_INCREMENT PRIMARY KEY,
    id_staff INT NOT NULL,
    tgl_pengeluaran DATETIME NOT NULL,
    jumlah_pengeluaran INT NOT NULL,
    deskripsi_pengeluaran TEXT NOT NULL,
    FOREIGN KEY (id_staff) REFERENCES staff(id)
);
```

Desain Interface (Wireframe/Mockup) Beserta Query-nya [poin 40]

- Tambah Menu



Query:

```
INSERT INTO menu (nama_menu, nama_gambar_menu, harga_menu,
gambar_menu) VALUES ("Caramel Macchiato", "CM.jpg", 20000,
"https://www.readyseteat.com/recipes-Iced-Caramel-Macchiato-9611");
```

- Input Pembelian

The screenshot shows a POS interface. On the left is a menu grid with nine items:

Item	Description	Price
Americano	Cup of coffee	Rp 18.000
Bolognese	Spaghetti with meat sauce	Rp 27.000
Chicken Karage	Breaded chicken pieces	Rp 27.000
Chocolate	Milkshake with chocolate flavor	Rp 25.000
Espresso	Strong coffee shot	Rp 15.000
French Fries	Fried potatoes with ketchup	Rp 20.000
Lemon Tea	Iced tea with lemon slices	Rp 17.000
Sausage Fries	French fries with sausage	Rp 25.000
Vanilla Latte	Iced coffee with vanilla flavor	Rp 25.000

To the right is a receipt section titled "Dine IN" with one item:

Item	Price
Americano	Rp 18.000

Total: Rp 18.000

Buttons at the bottom include "Clear Sale" and "Charge Rp 18.000".

The screenshot shows a payment screen for a total amount of Rp 18.000. It includes options for:

- Cash:** A "Cash Amount" input field.
- E-Wallet:** Logos for ShopeePay, OVO, DANA, gopay, and LinkAja.
- EDC:** Logos for BCA, Mandiri, BNI, BSI, and BRI.

Buttons at the bottom include "Point of Sale", "Menu", "Staff", "Dashboard", and "Profile".

Query :

```

INSERT INTO pembayaran
(id_staff,id_menu,id_jns_pembayaran,jumlah,total_harga) VALUES
(1,5,1,2,18000);
SELECT * FROM menu;

```

- Input Pengeluaran

DASHBOARD

tanggal : 20-May-2024

Pemasukan Pengeluaran

Tambah Pengeluaran

No.	Date Time	Staff	Deskripsi	Total Pengeluaran
1	2024-05-20T08:10:57.581061+00:00	Sidik	Es Kristal	25000
2	2024-05-20T08:41:41.05378+00:00	Dava	Galon 2	10000

Point of Sale Menu Staff Dashboard Profile

Query :

```
INSERT INTO pengeluaran (id_staff,jumlah_pengeluaran,deskripsi_pengeluaran)
VALUES (1,27000,"Es Kristal");
```

```
SELECT p.id_staff, s.nama_staff, p.jumlah_pengeluaran,
p.deskripsi_pengeluaran, p.tgl_pengeluaran
FROM pengeluaran p
JOIN staff s ON p.id_staff = s.id_staff
WHERE DATE(p.tgl_pengeluaran) = CURDATE();
```

- Dashboard (Pemasukan)

DASHBOARD

tanggal : 20-May-2024

Pemasukan Pengeluaran

Pemasukan Hari Ini: Rp. 213600

No.	Date Time	Staff	Menu	Harga	Jumlah	Total
1	2024-05-20T06:05:01.5 93112+00:00	Darma	Diana12	7500	1	7500
2	2024-05-20T06:06:09.7 60344+00:00	Darma	Testing	16000	1	16000
3	2024-05-20T06:08:58.9 70389+00:00	Darma	Martini	6000	1	6000
4	2024-05-20T06:19:04.2 44435+00:00	Sidik	Americano	8900	2	17800
5	2024-05-20T06:12:54.8 87973+00:00	Dava	Bijer2	8900	1	8900
6	2024-05-20T06:12:54.8 87973+00:00	Dava	Bijer2	8900	1	8900

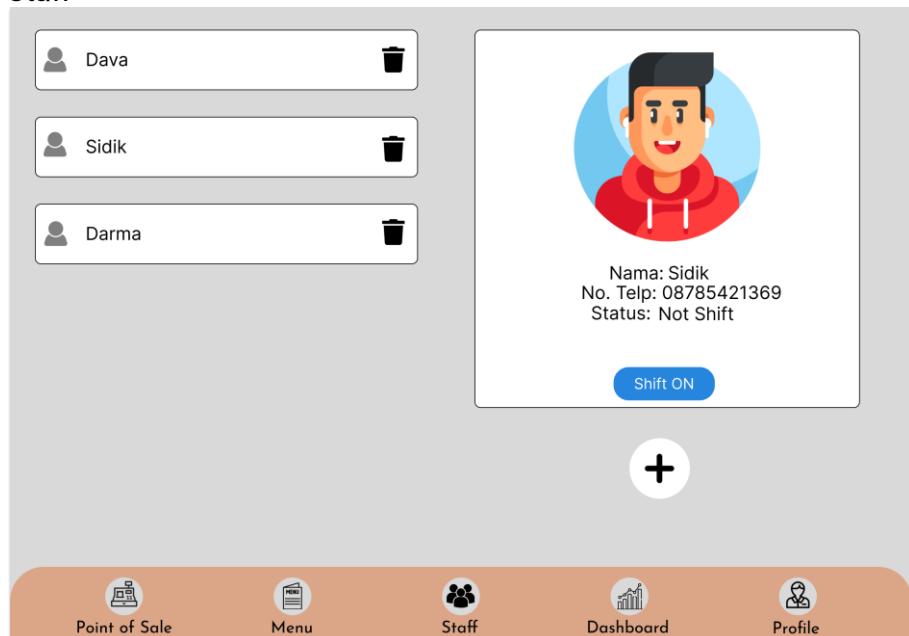
 Point of Sale
  Menu
  Staff
  Dashboard
  Profile

Query :

```
SELECT staff.nama_staff, menu.nama_menu, menu.harga_menu,
pembayaran.tgl_pembayaran, pembayaran.jumlah, pembayaran.total_harga
FROM pembayaran
JOIN staff ON staff.id = pembayaran.id_staff
JOIN menu ON menu.id = pembayaran.id_menu
WHERE DATE(pembayaran.tgl_pembayaran) = CURDATE();
```

```
- SELECT SUM(pembayaran.total_harga) AS total_harga_hari_ini
FROM pembayaran
WHERE DATE(pembayaran.tgl_pembayaran) = CURDATE();
```

- Staff



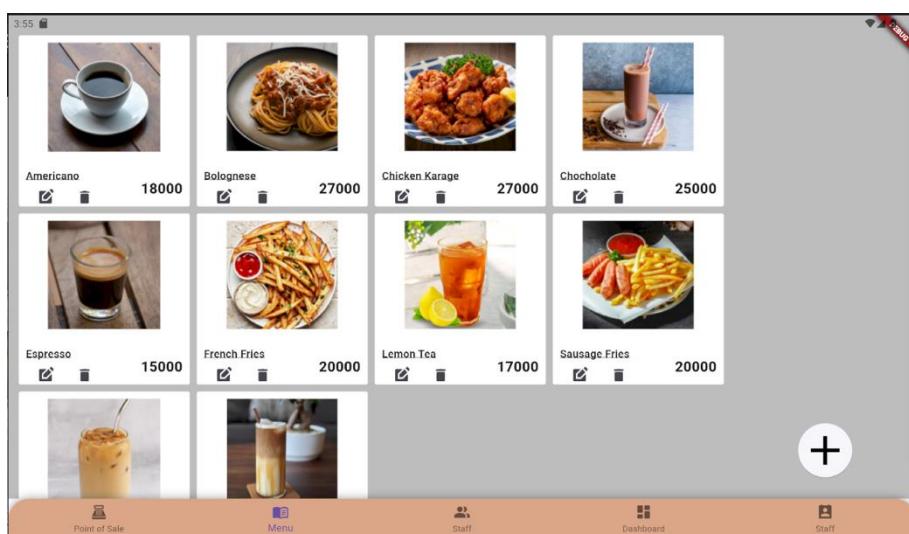
Query :

```
INSERT INTO staff (nama_staff, nomortlp_staff, status) VALUES ("Anandava", "087869545124", "On");
```

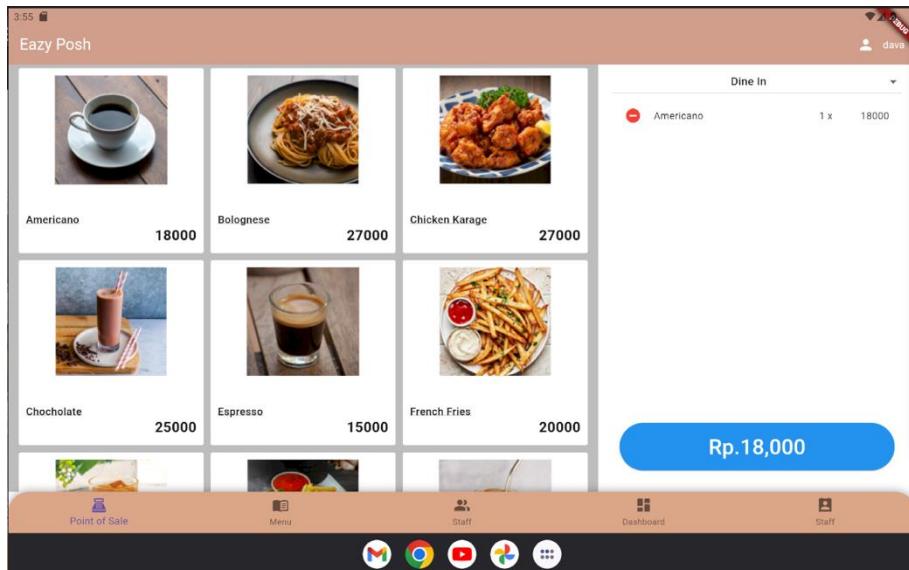
```
UPDATE staff SET status = "On" WHERE id_staff = 1;
```

Implementasi mockup ke aplikasi menggunakan Bahasa pemrograman dart, framework flutter dan database supabase:

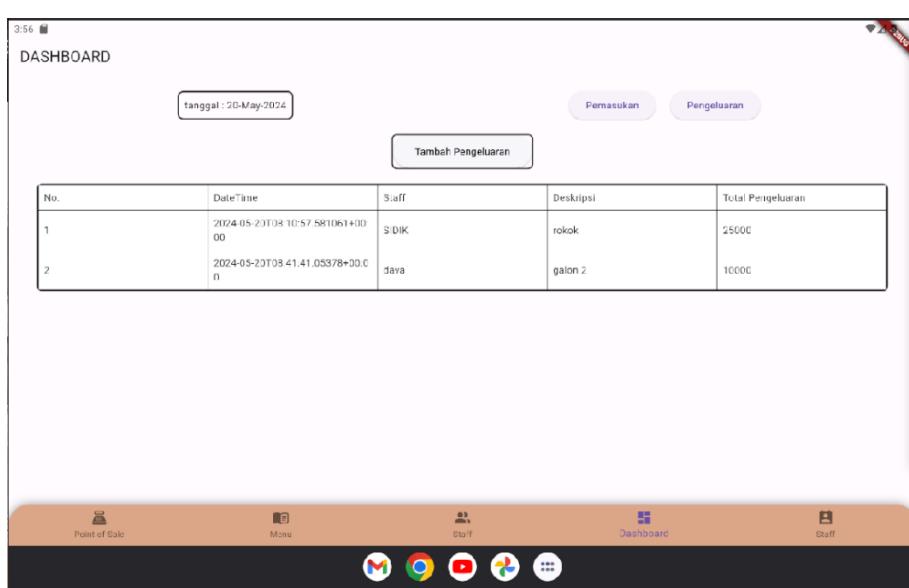
- Tampilan menu CRUD



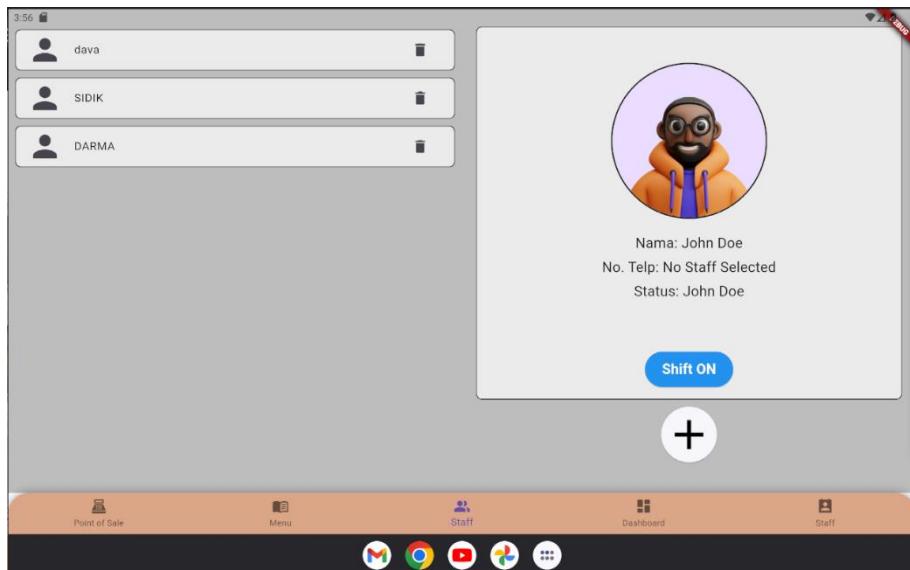
- Input Pembelian



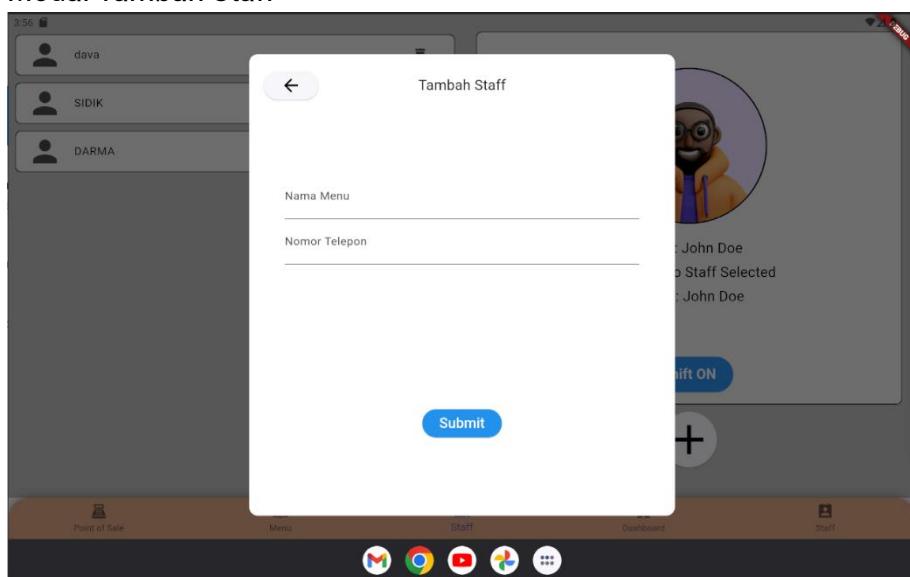
- Dashboard Pengeluaran



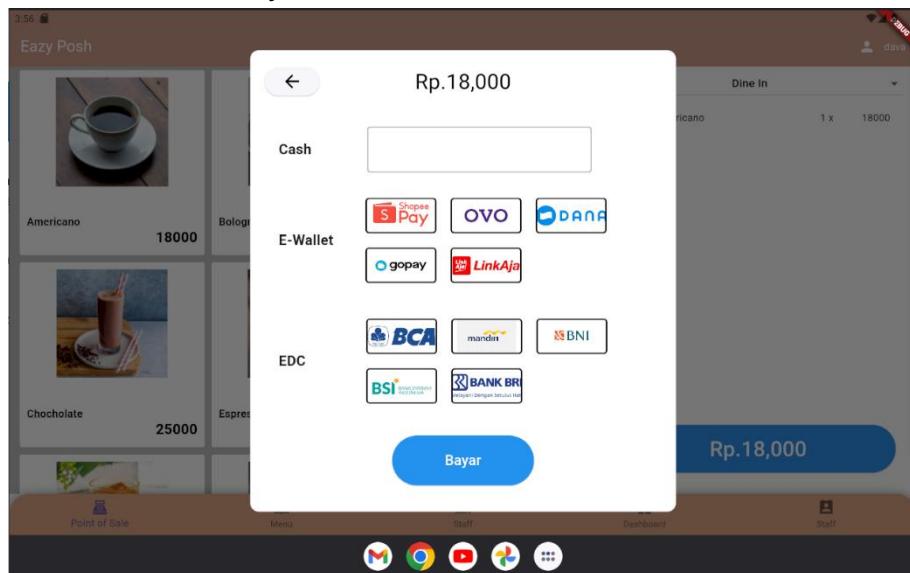
- Tampilan staff CRUD



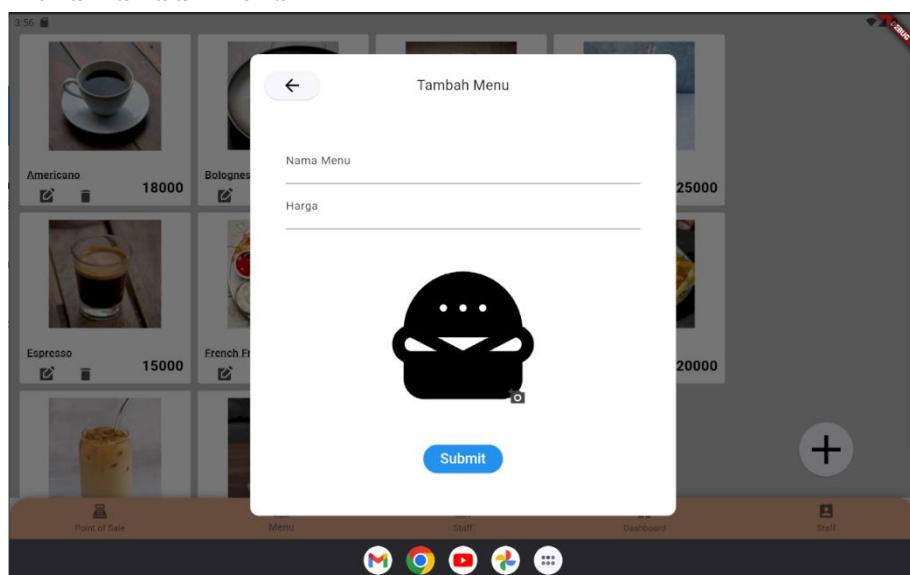
- Modal Tambah Staff



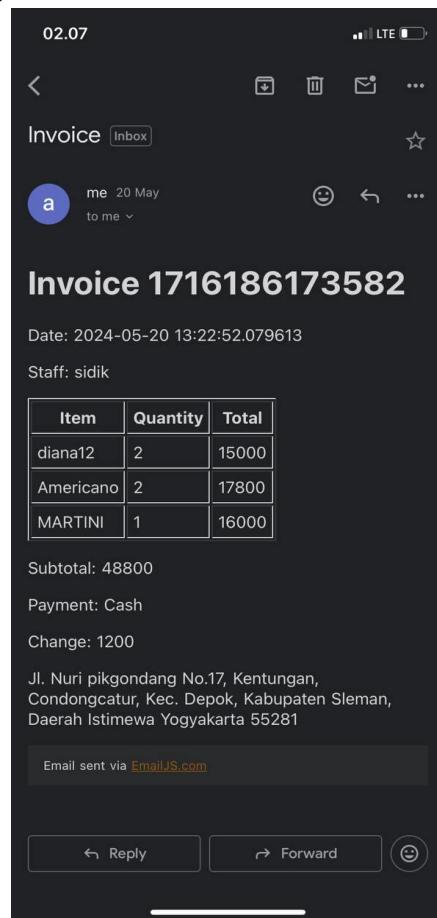
- Modal Jenis Pembayaran



- Modal Tambah Menu

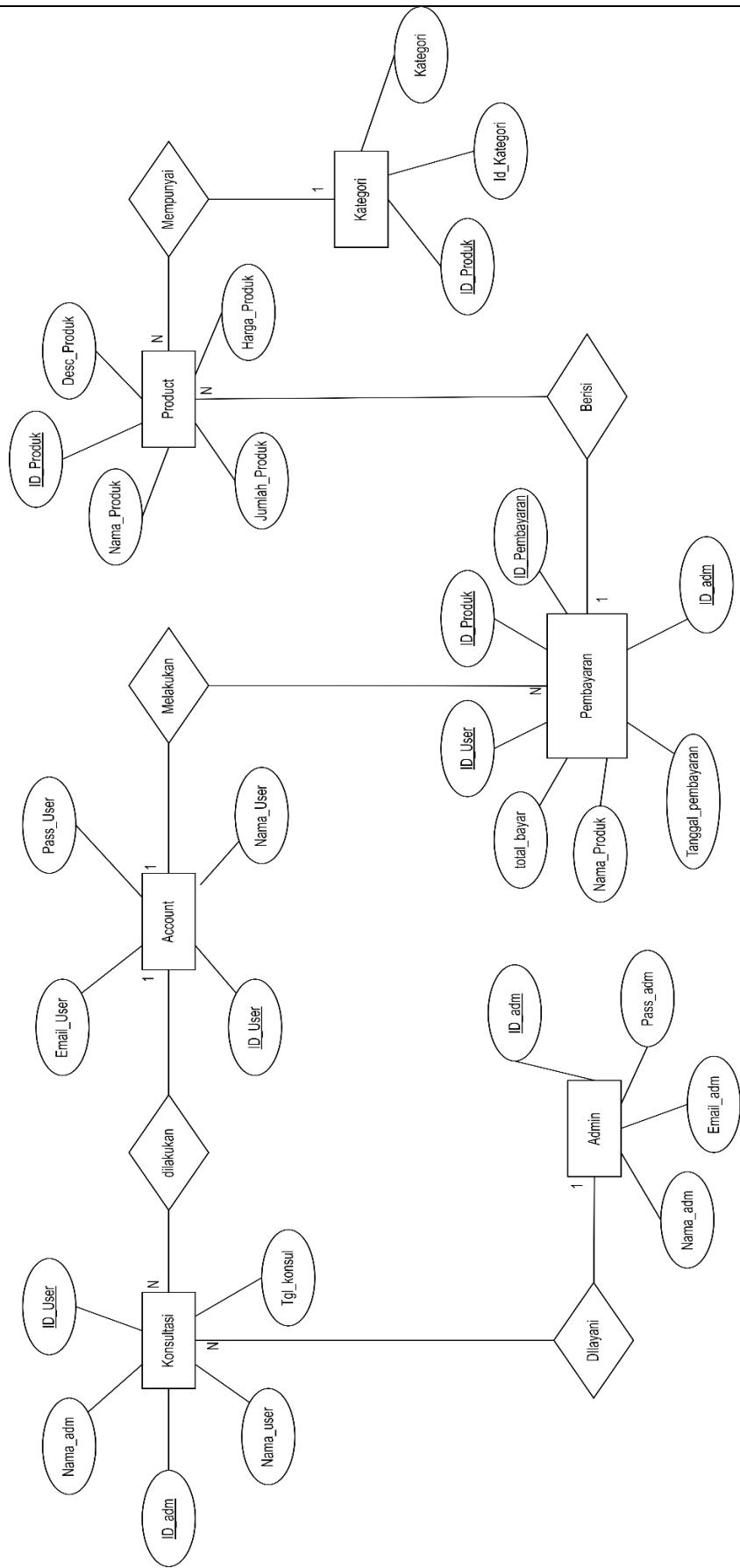


- Invoice Sederhana dengan Email

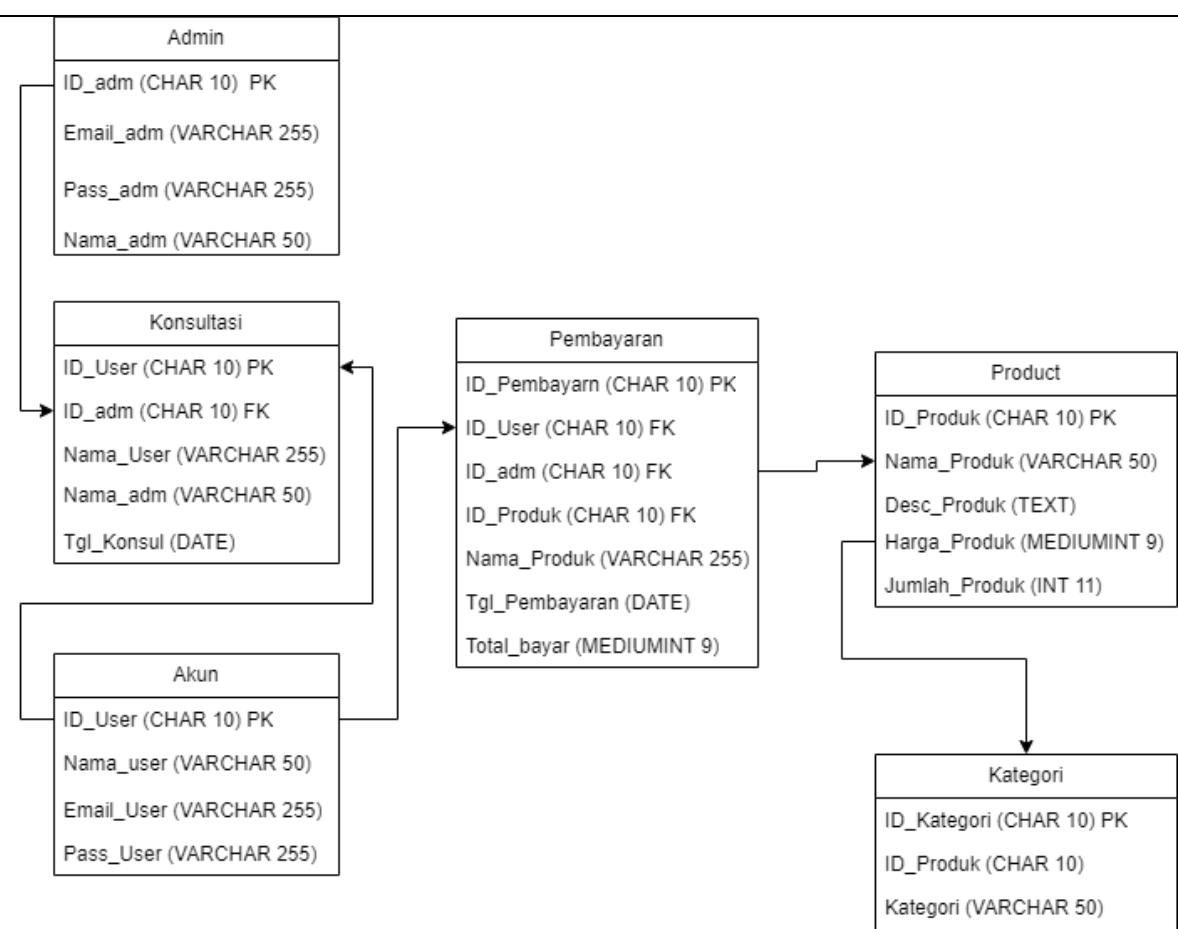


Judul Aplikasi	
SIMUN: APLIKASI PENJUALAN, PEMBELIAN DAN KONSULTASI PRODUK SIDOMUNCUL	
Dikembangkan Oleh	<ul style="list-style-type: none"> • 5220411257 – ZahfarAziz Ferdian • 5220411268 – Martina Diana Sari • 5220411254 – Luthfi Nur Faradillah • 5220411269 – Chaterine Laura Panggabean • 5220411282 – Vina Nasyi Atul Lailiyah
Deskripsi Singkat Instansi, Permasalahan, dan Solusi Aplikasi Yang Akan Dikembangkan [10 poin]	<p>PT Industri Jamu dan Farmasi Sidomuncul Tbk adalah perusahaan terkenal di Indonesia, terutama dalam industri jamu dan farmasi. Dikenal dengan produk-produk herbal yang berkualitas terutama dalam produksi Jamu dan Farmasi. PT Industri Jamu dan Farmasi Sidomuncul Tbk menjadi bagian penting dalam perusahaan penting kesehatan Indonesia.</p> <p>Permasalahan:</p> <ol style="list-style-type: none"> 1. Kurang terdapatnya perbedaan tiap kategori jamu 2. Kurangnya informasi lengkap tentang produk yang ditawarkan 3. Tidak adanya Konsultasi client ke Pakar jamu 4. Tidak adanya platform penjualan dan pembelian jamu <p>Solusi:</p> <ol style="list-style-type: none"> 1. Pada aplikasi yang dikembangkan dengan database akan mengkategorikan produk jamu yang ditawarkan sesuai kebutuhan client, untuk mempermudah pencarian. 2. Pada katalog produk yang ditawarkan akan ditambahkan deskripsi, resep atau racikan jamu, serta manfaat jamu bagi client yang melihat detail. 3. Pada aplikasi yang dikembangkan Client yang merasa kesulitan dalam memilih produk yang tepat dapat berkonsultasi pada Pakar Jamu. 4. Untuk mempromosikan produk lebih luas dan laku aplikasi dari PT. Sidomuncul akan menambahkan fitur Jual serta beli jamu serta farmasi pada aplikasi.
Pengguna Aplikasi [5 poin]	Masyarakat Umum

Desain Entity-Relationship Diagram (ERD) [10 poin]



Desain Relasi Basis Data [10 poin]



Query Data Definition Language (DDL) [25 poin]

```

CREATE TABLE admin (
    Id_adm char(10) primary key,
    nama_adm varchar(50),
    email_adm varchar(255),
    pass_adm varchar(255)
);

CREATE TABLE Pembayaran (
    Id_Pembayaran char(10) PRIMARY KEY,
    Tgl_Pembayaran DATE,
    Total_Bayar MEDIUMINT (9)
);

CREATE TABLE Account (
    Id_User char (10) PRIMARY KEY,
    Nama_User VARCHAR (50),
    ...
);
    
```

```
Email_User VARCHAR (255),
Pass_User VARCHAR(255)
);
DESC Account;

CREATE TABLE Konsultasi(
    Id_Konsul char(10) PRIMARY KEY,
    Tgl_Konsul DATE
);

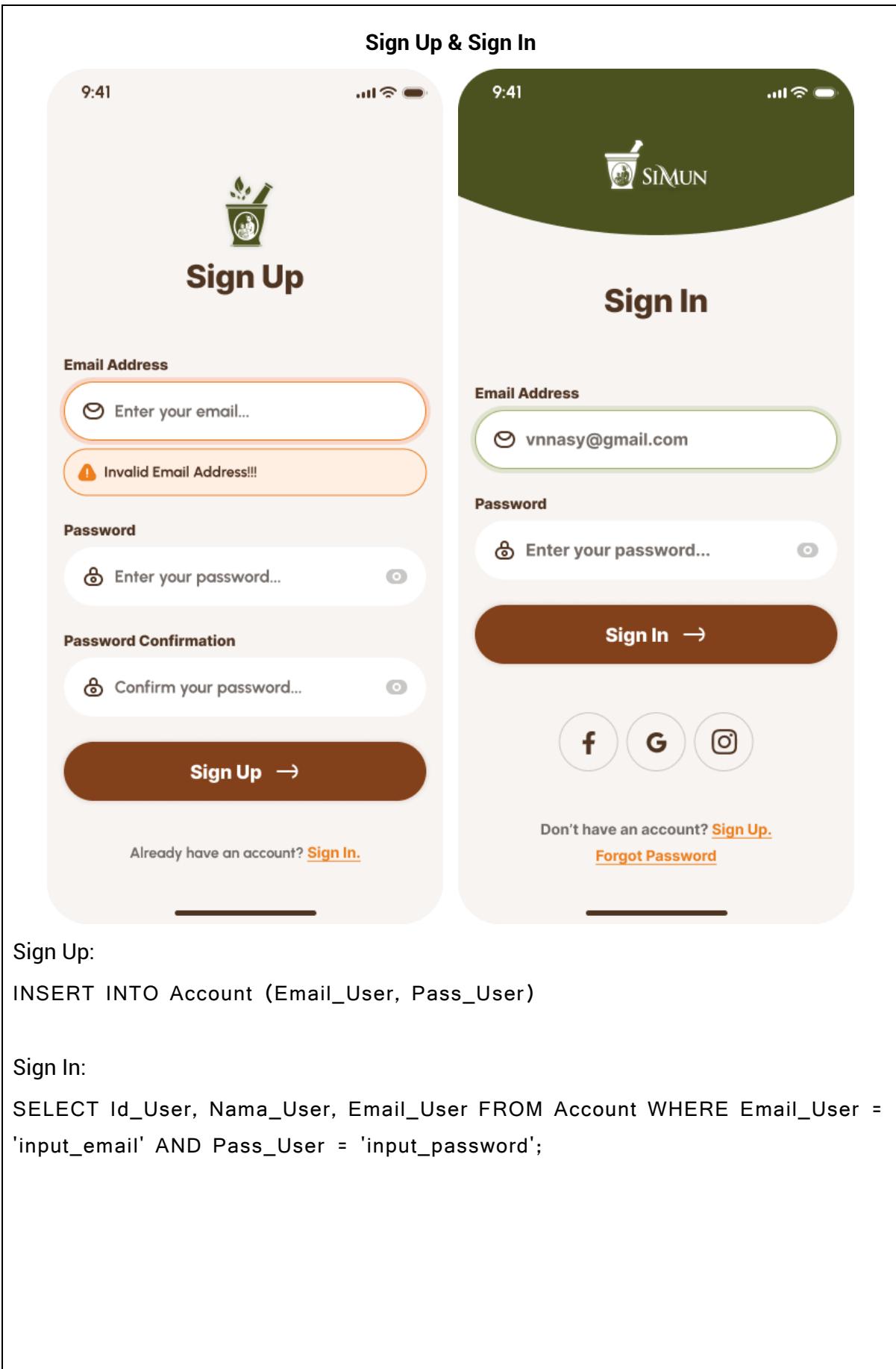
CREATE TABLE Product (
    Id_Produk char(10) PRIMARY KEY,
    Nama_Produk VARCHAR(50),
    Desc_Produk Text,
    Harga_Produk Mediumint(9),
    Jumlah_Produk INT (11)
);

CREATE TABLE Kategori (
    Id_Kategori char(10) PRIMARY KEY,
    Kategori VARCHAR (50)
);
```

Desain Interface (Wireframe/Mockup) Beserta Query-nya [poin 40]

First Page



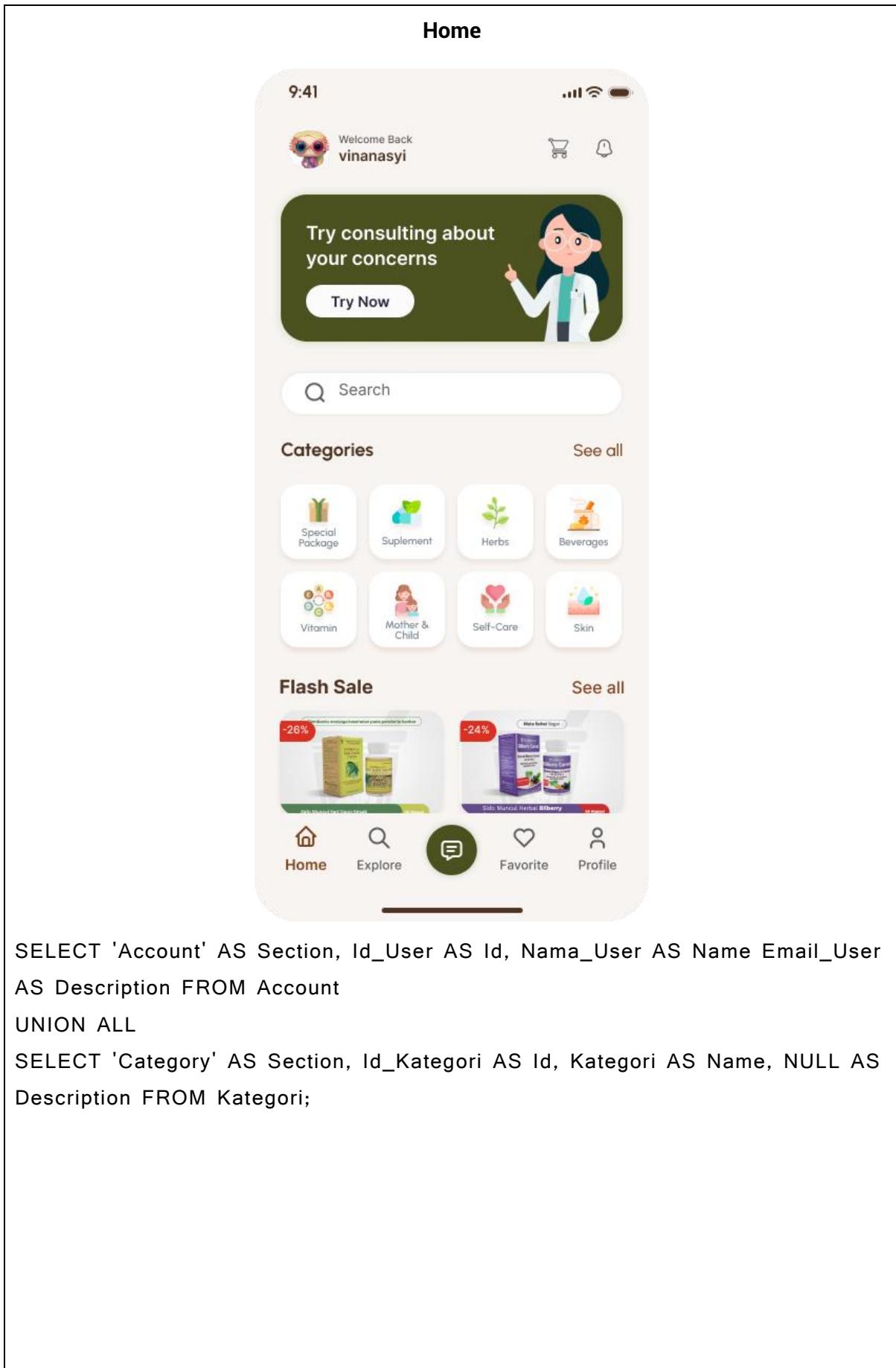


Sign Up:

```
INSERT INTO Account (Email_User, Pass_User)
```

Sign In:

```
SELECT Id_User, Nama_User, Email_User FROM Account WHERE Email_User = 'input_email' AND Pass_User = 'input_password';
```



```
SELECT 'Account' AS Section, Id_User AS Id, Nama_User AS Name Email_User AS Description FROM Account
UNION ALL
SELECT 'Category' AS Section, Id_Kategori AS Id, Kategori AS Name, NULL AS Description FROM Kategori;
```

Explore & Simun Consult

The image displays two side-by-side screenshots of the SIMUN mobile application interface.

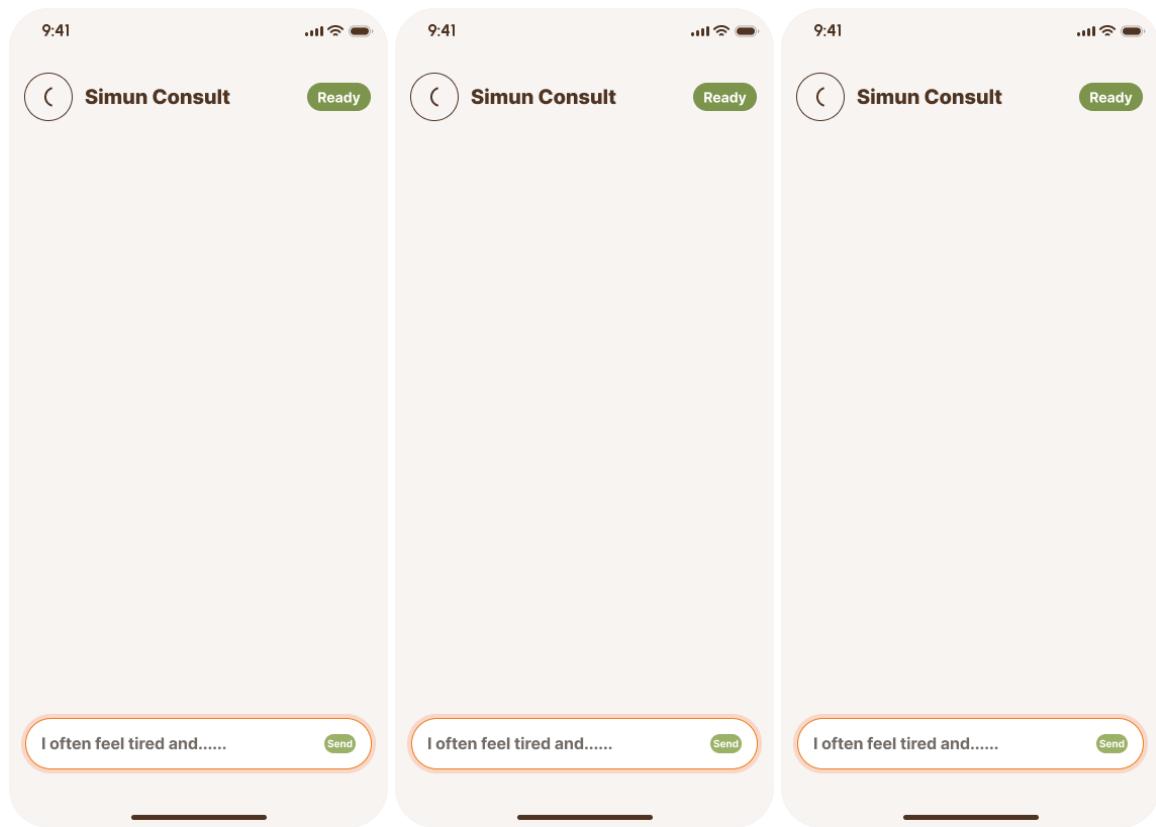
Left Screenshot (Home Screen):

- Top Bar:** Shows the time as 9:41 and signal strength.
- Search Bar:** Labeled "Search" with a magnifying glass icon.
- Category Buttons:** "vitamin" (highlighted in red), "Supplement", and "Mother & Child".
- Section: Best-Selling Products**
 - A grid titled "Produk Herbal Terlaris" showing various products like Vitamin C 1000, Telaik Rangin, and Kukuh Basah.
- Section: Shop by Brand**
 - Grid of brand logos: SusuJaya, Telaik Rangin, Alami Sariyah, madusido, Anak Sehat, Vitamix C 1000, Esemag, and Kukuh Basah ENER-G.
- Section: Exclusive Collaboration**
 - Grid featuring products from SidoMuncul and TokoAja.
- Bottom Navigation Bar:** Icons for Home, Explore (highlighted in green), Favorite, and Profile.

Right Screenshot (Simun Consult Screen):

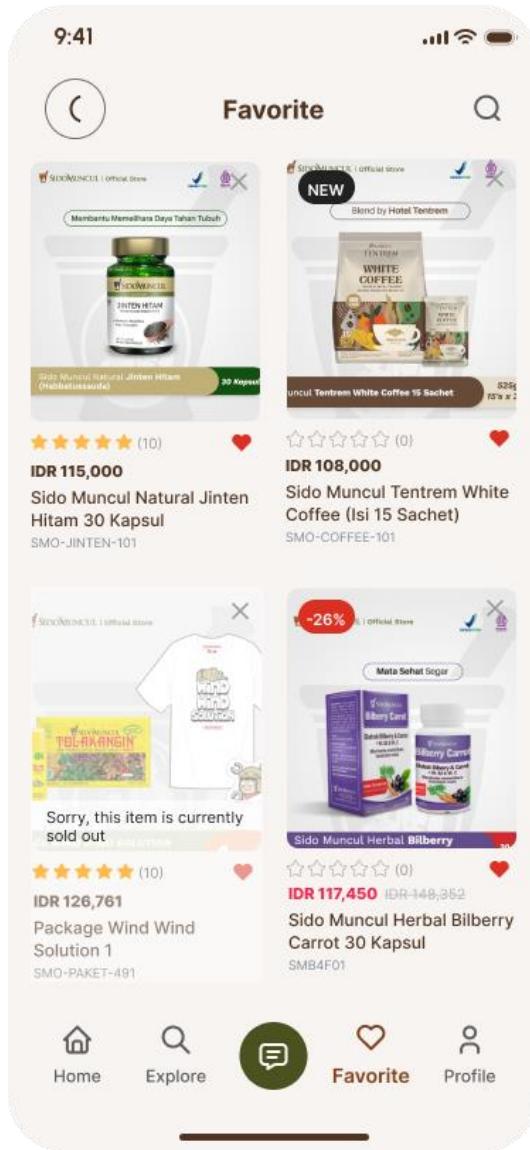
- Top Bar:** Shows the time as 9:41 and signal strength.
- Logo:** A green stylized logo of a person holding a leaf.
- Title:** "Welcome to Simun Consult"
- Text:** "Empowering your health journey with expert herbal advice, anytime, anywhere." followed by a green leaf emoji.
- Illustration:** A cartoon doctor with glasses and a stethoscope, pointing upwards.
- Call-to-Action:** A brown button labeled "Let's Chat" with a right-pointing arrow.
- Bottom Navigation Bar:** Icons for Home, Explore, Favorite, and Profile.

Chat



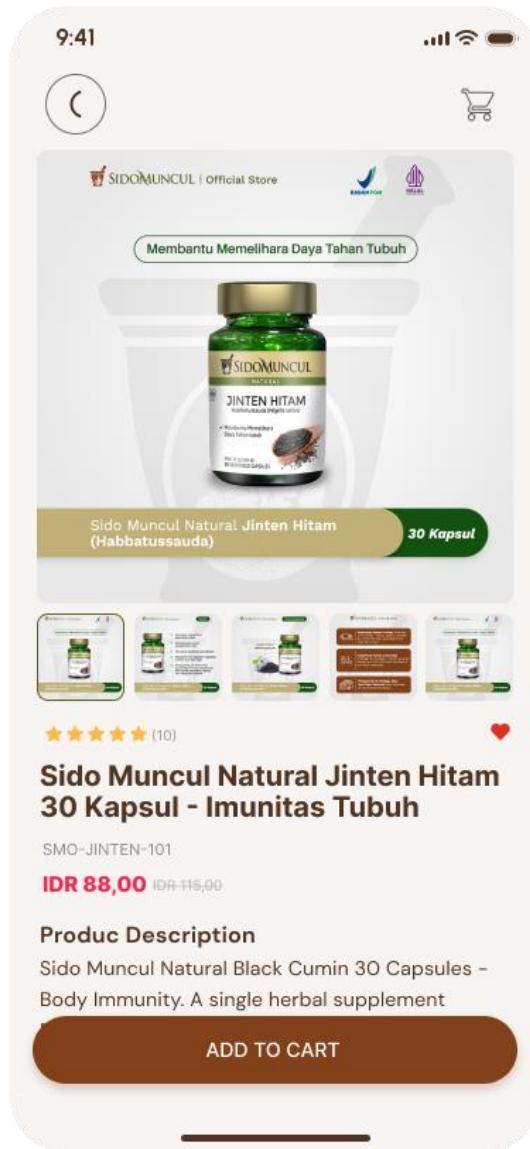
```
SELECT k.Id_Konsul, k.Tgl_Konsul, a.nama_adm, acc.Nama_User FROM  
Konsultasi k JOIN Admin a ON k.Id_adm = a.Id_adm JOIN Account acc ON  
k.Id_User = acc.Id_User;
```

Favorite



```
SELECT p.Id_Prod, p.Nama_Prod, k.Kategori FROM Product p JOIN
Produk_Kategori pk ON p.Id_Prod = pk.Id_Prod JOIN Kategori k ON
pk.Id_Kategori = k.Id_Kategori;
```

Product View



SELECT

a.Id_User, a.Nama_User, p.Id_Prod, p.Nama_Prod,

k.Kategori, pb.Tgl_Pembayaran, pb.Total_Bayar

FROM

Account a

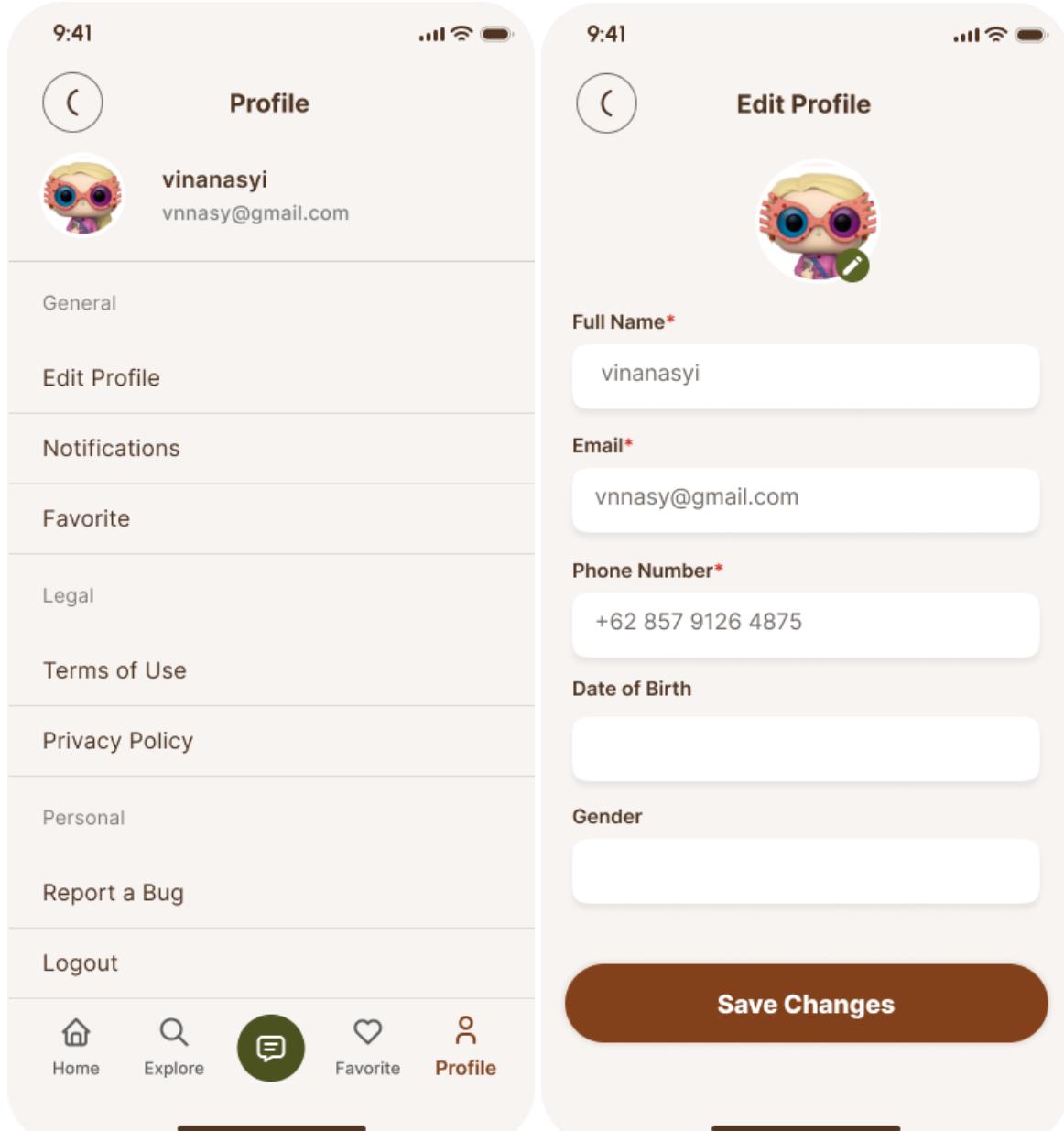
JOIN Pembelian pe ON a.Id_User = pe.Id_User

JOIN Product p ON pe.Id_Prod = p.Id_Prod

JOIN Produk_Kategori pk ON p.Id_Prod = pk.Id_Prod

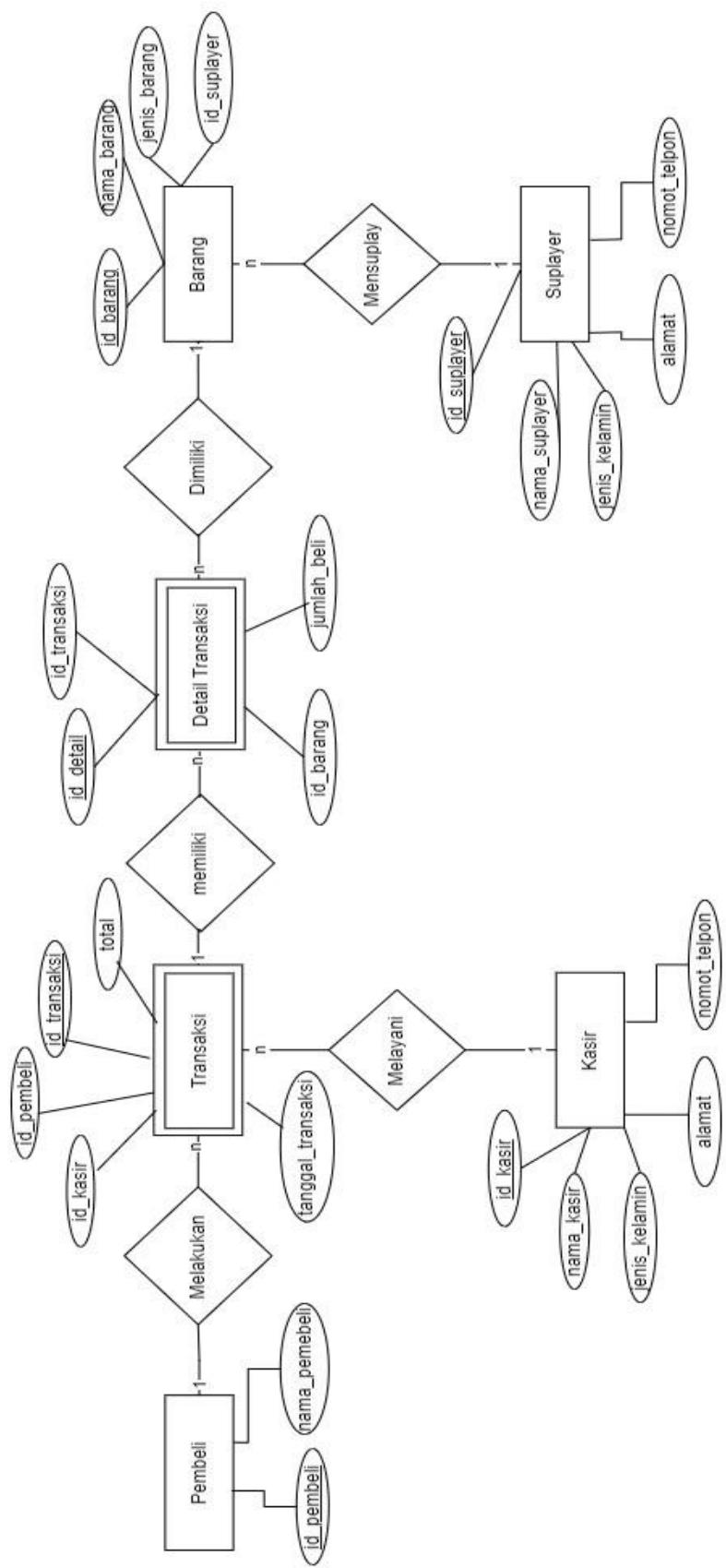
```
JOIN Kategori k ON pk.Id_Kategori = k.Id_Kategori  
JOIN Pembayaran pb ON pe.Id_Pembelian = pb.Id_Pembelian  
WHERE k.Kategori = 'kategori_yang_dipilih';
```

Profile

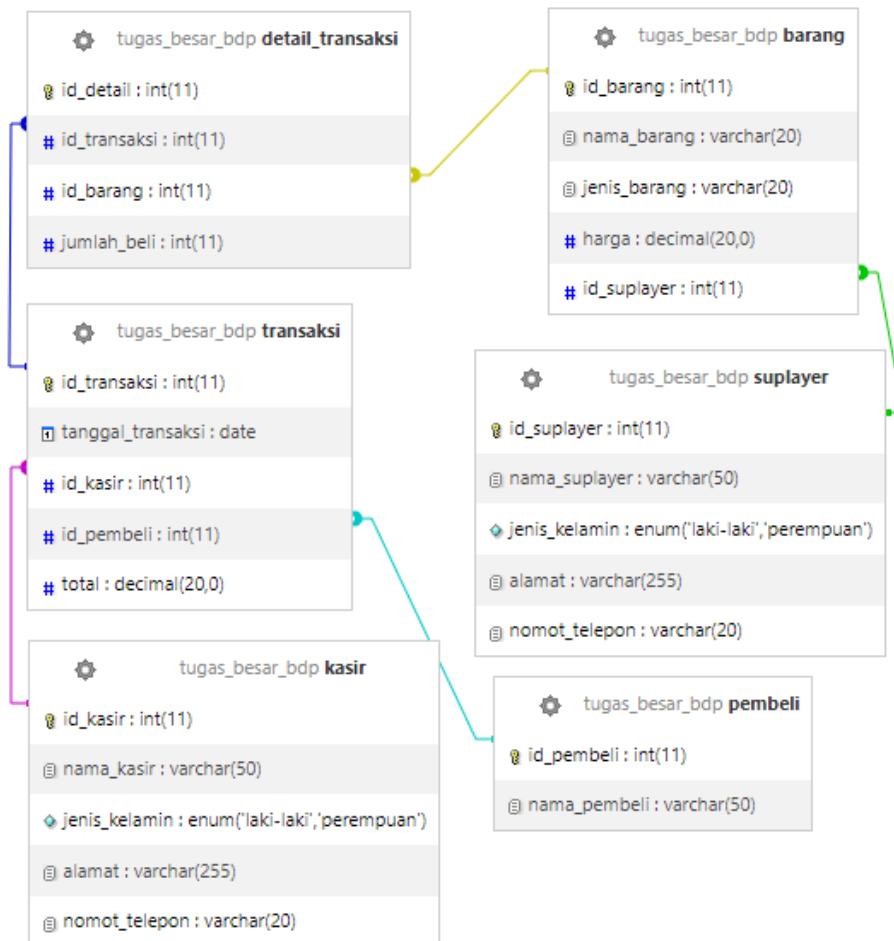


Judul Aplikasi	
PERANCANGAN SISTEM APLIKASI TOKO PANCING DAN AKSESORIS	
Dikembangkan Oleh	<ul style="list-style-type: none"> • 5220411297 – Mahmud Isti Panggalih • 5220411323 – Revaldo • 5220411326 – Muh Agna Khairul Aqila • 5220411328 – Muhammad Adin Musababa • 5220411321 – Ali sahat Pardomuan
Deskripsi Singkat Instansi, Permasalahan, dan Solusi Aplikasi Yang Akan Dikembangkan [10 poin]	<p>Instansi yang akan digunakan untuk dikembangkan dalam permasalahan ini adalah sebuah toko aksesoris akuarium dan peralatan pancing yang beralamatkan di Desan Cepor Lor, Kecamatan Palbapang, Kabupaten Bantul, Provinsi Daerah Istimewa Yogyakarta. Toko ini bernama The First Aquarium. Toko ini telah ada sejak tahun 2002 dengan pemilik toko bernama "". Namun, selama toko ini berdiri proses pelayanan transaksi pembelian dan proses stok barang masih manual. Pada proses transaksi biasanya hanya dituliskan dalam buku atau note saja. Dan pada pengecekan barang masih dilakukan satu persatu. Oleh karena itu perlu dibuatkan sistem informasi untuk Toko The First Aquarium.</p> <p>Sistem ini dibuat agar nanti kedepannya dapat membantu mempercepat proses penyelesaian pekerjaan. Pada proses penjualan, pembuatan nota sudah secara otomatis, sehingga tidak perlu merekap dua kali ke buku. Pengecekan barang dapat dilakukan dengan cepat tanpa harus mengecek satu persatu. Dengan demikian sistem dapat meningkatkan kinerja di Toko The First Aquarium.</p>
Pengguna Aplikasi [5 poin]	<ol style="list-style-type: none"> 1. Admin/pegawai 2. Suplayer

Desain Entity-Relationship Diagram (ERD) [10 poin]



Desain Relasi Basis Data [10 poin]



Query Data Definition Language (DDL) [25 poin]

```
CREATE TABLE Pembeli (
    id_pembeli INT PRIMARY KEY,
    nama_pembeli VARCHAR(255) NOT NULL
);

CREATE TABLE Kasir (
    id_kasir INT PRIMARY KEY,
    nama_kasir VARCHAR(255) NOT NULL,
    jenis_kelamin ENUM('laki-laki', 'perempuan') NOT NULL,
    alamat VARCHAR(255) NOT NULL,
    nomot_telepon VARCHAR(20) NOT NULL
);
DELIMITER //
```

```

CREATE TRIGGER before_insert_kasir
BEFORE INSERT ON Kasir
FOR EACH ROW
BEGIN
    SET NEW.id_kasir = CONCAT('adm', NEW.increment_id);
END //

DELIMITER ;

CREATE TABLE Suplayer (
    id_suplayer INT PRIMARY KEY,
    nama_suplayer VARCHAR(255) NOT NULL,
    jenis_kelamin ENUM('laki-laki', 'perempuan') NOT NULL,
    alamat VARCHAR(255) NOT NULL,
    nomot_telepon VARCHAR(20) NOT NULL
);

CREATE TABLE Barang (
    id_barang INT PRIMARY KEY,
    nama_barang VARCHAR(20) NOT NULL,
    jenis_barang VARCHAR(20) NOT NULL,
    id_suplayer INT NOT NULL,
    FOREIGN KEY (id_suplayer) REFERENCES Suplayer(id_suplayer)
);

CREATE TABLE Transaksi (
    id_transaksi INT PRIMARY KEY,
    tanggal_transaksi DATE NOT NULL,
    id_kasir INT,
    id_pembeli INT,
    FOREIGN KEY (id_kasir) REFERENCES Kasir(id_kasir),
    FOREIGN KEY (id_pembeli) REFERENCES Pembeli(id_pembeli)
);

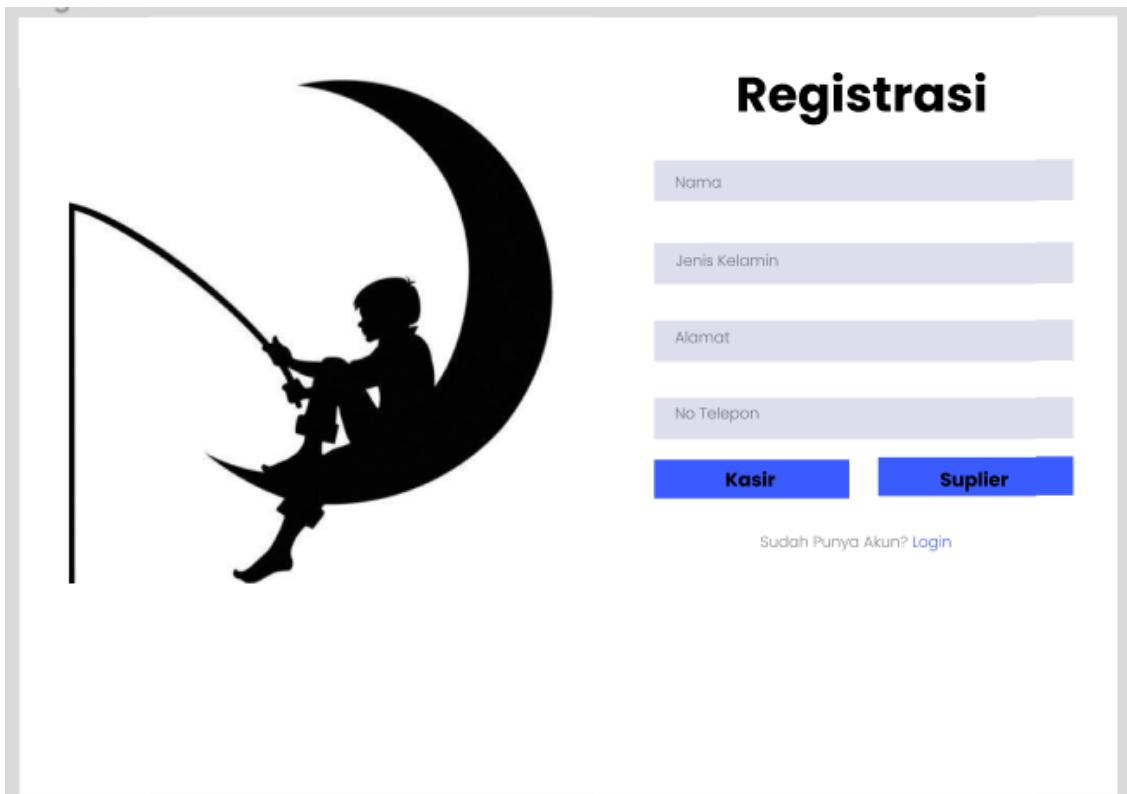
CREATE TABLE Detail_Transaksi (
    id_detail INT PRIMARY KEY,
    id_transaksi INT,

```

```
id_barang INT,  
jumlah_beli INT NOT NULL,  
FOREIGN KEY (id_transaksi) REFERENCES Transaksi(id_transaksi),  
FOREIGN KEY (id_barang) REFERENCES Barang(id_barang)  
);
```

Desain Interface (Wireframe/Mockup) Beserta Query-nya [poin 40]

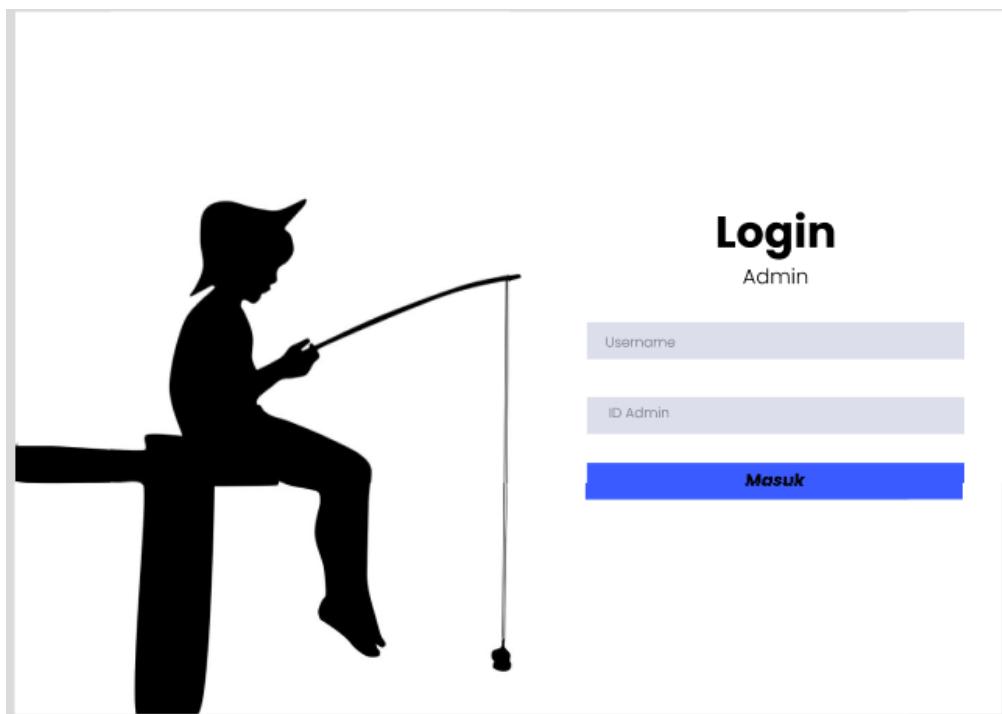
1. Query form register



The wireframe shows a registration form titled "Registrasi". On the left side, there is a large black silhouette of a person fishing from a crescent moon. The form has four input fields: "Nama", "Jenis Kelamin", "Alamat", and "No Telepon". Below the input fields are two blue buttons labeled "Kasir" and "Suplier". At the bottom right, there is a link "Sudah Punya Akun? Login".

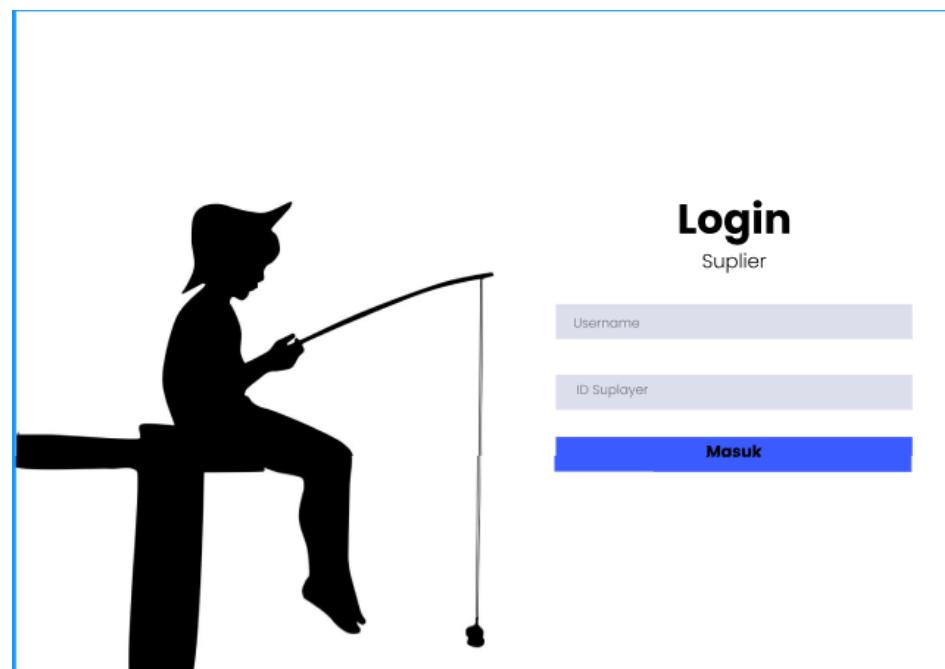
```
INSERT INTO  
CASE  
    WHEN :selected_role = 'admin' THEN admin  
    WHEN :selected_role = 'suplayer' THEN suplayer  
END  
(id, username, jenis_kelamin, alamat, no_telepon)  
VALUES  
(UUID(), :username, :jenis_kelamin, :alamat, :no_telepon);
```

2. Qwery from login admin



```
SELECT * FROM admin WHERE USERNAME = 'username_input' AND  
PASSWORD = 'Id_input';
```

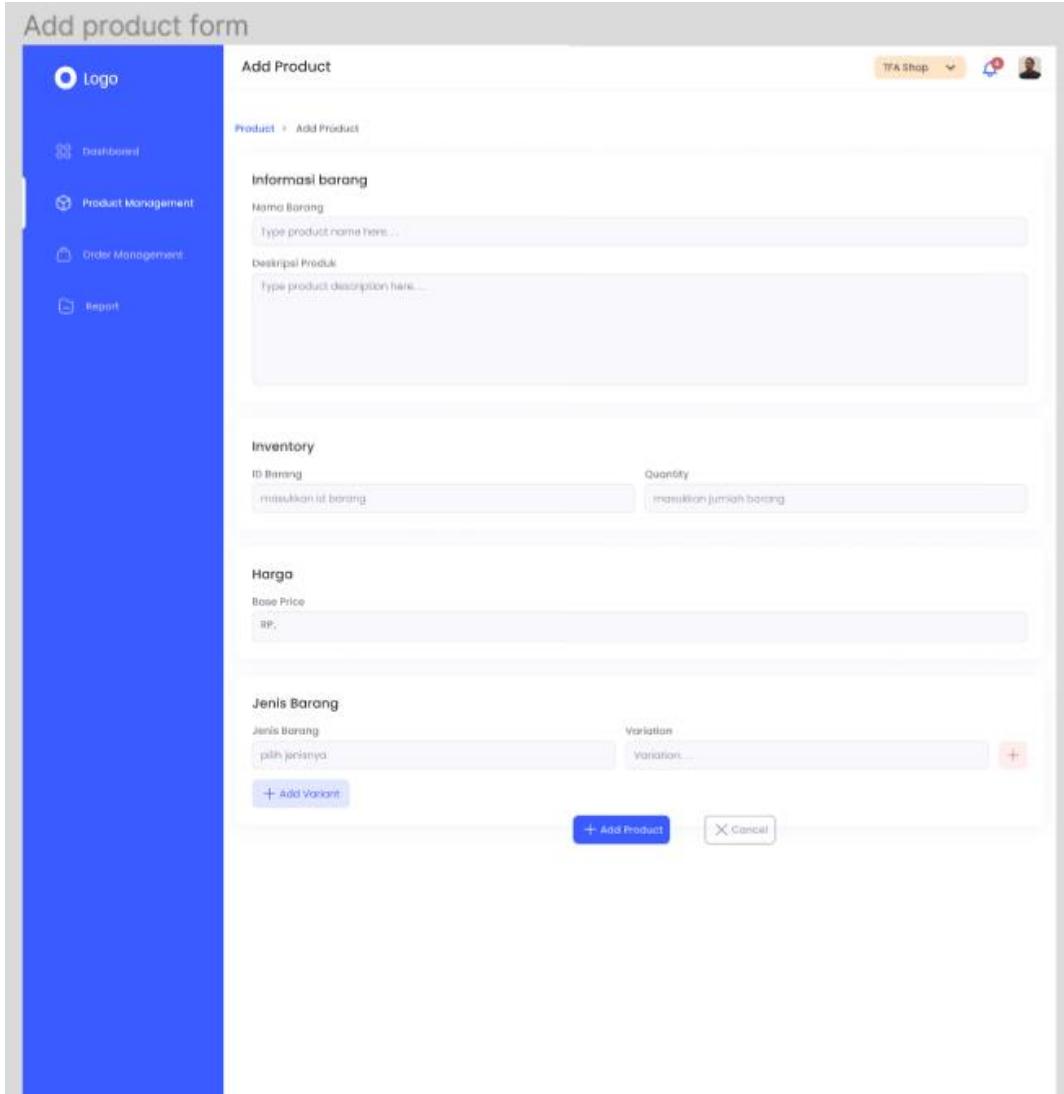
3. Query login suplayer



```
SELECT * FROM suplayer
```

```
WHERE USERNAME = 'username_input' AND  
PASSWORD = 'Id_input';
```

4. Query menambahkan barang



```
INSERT INTO barang (nama_item, jumlah_item, harga, jenis_barang, id_suplier,  
stock_item)  
VALUES (:nama_item, :jumlah_item, :harga, :jenis_barang, :id_suplier, :  
stock_item );
```

5. Query untuk order

```
SELECT * FROM Barang WHERE id_barang = [selected_product_id];
SET @jumlah_barang = [input_quantity];
SET @id_transaksi = (SELECT MAX(id_transaksi) + 1 FROM Transaksi);
INSERT INTO Transaksi (id_transaksi, tanggal_transaksi, id_kasir, id_pembeli)
VALUES (@id_transaksi, NOW(), [admin_id], [customer_id]);

INSERT INTO Detail_Transaksi (id_transaksi, id_barang, jumlah_barang,
subtotal)
VALUES (@id_transaksi, [selected_product_id], @jumlah_barang,
[selected_product_price] * @jumlah_barang);
```

The screenshot shows a web application interface with a sidebar on the left containing navigation links: Logo, Dashboard, Product Management, Order Management, and Report. The main area is titled 'Order' and displays a table of products. The table has columns: Product, ID, Category, Stock, Harga, Added, and Jumlah. The data in the table is as follows:

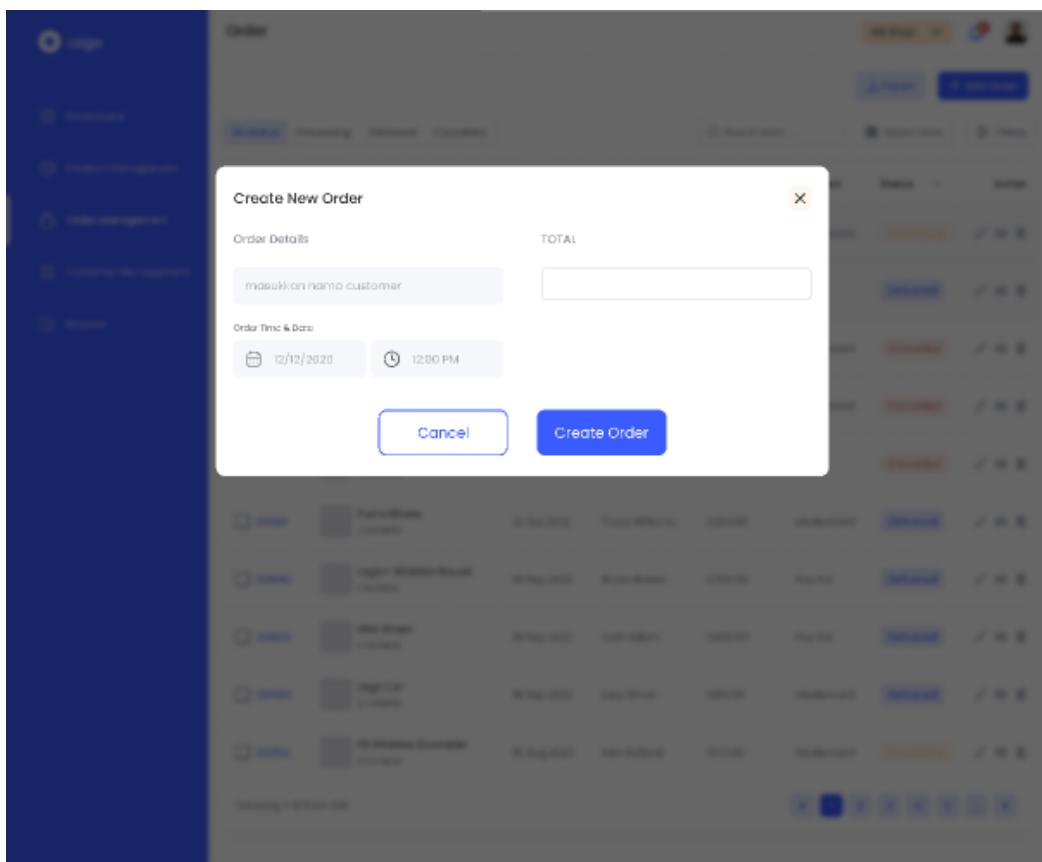
Product	ID	Category	Stock	Harga	Added	Jumlah
Joran yamada 2 Variants	A001	Alat Pancing	10	Rp 200.000,00	29 Dec 2022	
Kail 3mm 2 Variants	A002	Alat pancing	204	Rp 5.000,00	24 Dec 2022	
Tokoni 50ml 3 Variants	B001	Aksesoris dan pokan ikan	40	Rp 25.000,00	12 Dec 2022	
Barong 3m 1 Variants	A003	Alat Pancing	401	Rp 20.000,00	28 Oct 2022	
Aquarium 20x40 4 Variants	B002	Aksesoris dan pokan ikan	120	Rp 70.000,00	28 Oct 2022	
Pompa air Shimizu 3 Variants	B003	Aksesoris Dan pokan ikan	432	Rp 45.000,00	28 Oct 2022	
Ikan Stasi Air 1 Variants	C001	Ikan	0	Rp 20.000,00	18 Sep 2022	
Ikan Kol 3 Variants	C002	Ikan	247	Rp 10.000,00	18 Sep 2022	
Joran 3m 2 Variants	A004	Alat Pancing	299	Rp 10.000,00	18 Sep 2022	
Aquoscope 2 Variants	B004	Aksesoris dan pokan ikan	38	Rp 60.000,00	10 Aug 2022	

At the bottom of the table, it says 'Showing 1-10 from 100'. To the right of the table, there are navigation buttons for page numbers 1 through 5.

```

SELECT
    Transaksi.id_transaksi,
    DATE_FORMAT(Transaksi.tanggal_transaksi, '%Y-%m-%d %H:%i:%s') AS tanggal_waktu,
    SUM(Detail_Transaksi.subtotal) AS total_belanja
FROM
    Transaksi
JOIN
    Detail_Transaksi ON Transaksi.id_transaksi = Detail_Transaksi.id_transaksi
WHERE
    Transaksi.id_transaksi = [input_transaction_id];

```



6. Query untuk report

```

SELECT
    t.id_transaksi AS "ID Order",
    t.tanggal_transaksi AS "Tanggal Transaksi",
    k.nama_kasir AS "Kasir",
    p.nama_pembeli AS "Customer",

```

```

b.nama_barang AS "Product",
dt.jumlah_beli AS "Jumlah",
(dt.jumlah_beli * b.harga_barang) AS "Total Price"

FROM
    Transaksi t
JOIN
    Kasir k ON t.id_kasir = k.id_kasir
JOIN
    Pembeli p ON t.id_pembeli = p.id_pembeli
JOIN
    Detail_Transaksi dt ON t.id_transaksi = dt.id_transaksi
JOIN
    Barang b ON dt.id_barang = b.id_barang
ORDER BY
    t.tanggal_transaksi DESC;

```

Product	Id Order	Customer	Jumlah	Price	Kasir
Ikan Sapi Air	Ikan	Susanto	12	Rp240.000.00	Marni
Joran 3m +1 other products	Alat Pancing	Suyud	6	Rp 900.000.00	Marni
Ikan Koi	Ikan	Sulis	3	Rp 30.000.00	Marni
Aquascope +1 other products	Aksesoris Dan Pakan	Bayu	9	Rp5.400.000.00	Marni
Kail Pancing	Alat Pancing	Rizky	10	Rp50.000.00	Marni

Hug x Hug

7. Detail Transaksi:

SELECT

```
t.id_transaksi,  
t.nama_pelanggan AS 'Bpk/ibu/sdr',  
t.alamat,  
t.no_hp AS 'No Hp',  
dt.banyaknya,  
dt.nama_item AS 'Nama',  
dt.harga AS 'Harga',  
dt.jumlah AS 'Jumlah'
```

FROM

```
transaksi t
```

JOIN

```
detail_transaksi dt
```

ON

```
t.id_transaksi = dt.id_transaksi
```

WHERE

```
t.id_transaksi; = [ID_TRANSAKSI]
```

The screenshot shows a receipt template titled "Struck TFA Shop". At the top right, there is a dropdown menu labeled "TFA Shop" with a user icon. Below the title, the address "Alamat: Cepor Lor Palbapang Bantul" and phone number "No Hp : 088123456786" are listed. To the right, there are fields for "Bpk/ibu/sdr:" and "Di:". A large table below has columns for "Banyaknya", "Nama", "Harga", and "Jumlah". At the bottom right of the table, there is a field labeled "jumlah Rp." and a signature area labeled "Tanda Terima".

<p>Judul Aplikasi</p> <h2 style="text-align: center;">LAUNDRYHUB: SOLUSI MODERN UNTUK MANAGEMENT LAUNDRY</h2>	
<p>Dikembangkan Oleh</p>	
<ol style="list-style-type: none"> 1. 5220411306 – Muhammad Sirajul Huda 2. 5220411308 – Dara Shaliha Happyananda Imani 3. 5220411329 – Amalia Nur Hanifah 4. 5220411296 – Bayu Sagara Handrika 5. 5220411311 – Muhammad Hilmy Prasetyi Himawan 	
<p>Deskripsi Singkat Instansi, Permasalahan, dan Solusi Aplikasi Yang Akan Dikembangkan [10 poin]</p>	
<p>Aplikasi Manajemen Laundry dengan Sistem Modern adalah solusi inovatif untuk instansi atau usahalaundry yang menghadapi tantangan dalam mengelola operasional mereka secara efisien. Instansi laundry seringkali menghadapi berbagai permasalahan, seperti:</p> <p>Pengelolaan Inventori: Sulitnya melacak stok bahan kimia, deterjen, atau perlengkapan lainnya yang dibutuhkan untuk mencuci pakaian.</p> <p>Penjadwalan dan Pelacakan Pesanan: keterlambatan dalam menyelesaikan laundry sering kali terjadi karena tidak ada jadwal atau target yang di catat hal itu menyebabkan customer kecewa</p> <p>Manajemen Pembayaran: Proses pembayaran yang manual bisa menjadi lama dan meningkatkan risiko kesalahan.</p> <p>Komunikasi dengan Pelanggan: Kekurangan sistem yang memungkinkan pelanggan untuk memantau status pesanan mereka dan berkomunikasi dengan instansi laundry secara manual.</p> <p>Untuk mengatasi permasalahan-permasalahan tersebut, aplikasi manajemen laundry akan dikembangkan. Solusi yang ditawarkan meliputi:</p> <p>Sistem Inventori Terintegrasi: Aplikasi akan memungkinkan instansi laundry untuk melacak inventori mereka dengan mudah, memperbarui stok secara otomatis setiap kali ada penggunaan, dan memberikan notifikasi saat stok mendekati kehabisan.</p> <p>Penjadwalan dan Pelacakan Pesanan Otomatis: Pengguna dapat membuat jadwal cuci secara langsung</p>	

melalui aplikasi, dan pelanggan dapat melacak status pesanan mereka secara real-time, termasuk pemberitahuan saat pesanan telah selesai.

Pembayaran Digital: Aplikasi akan menyediakan sistem pembayaran digital yang aman dan mudah digunakan, memungkinkan pelanggan untuk membayar pesanan mereka secara online atau melalui aplikasi.

Portal Pelanggan: Pelanggan akan memiliki akses ke portal khusus yang memungkinkan mereka untuk membuat pesanan, melacak status pesanan, memberikan umpan balik, dan berkomunikasi dengan instansi laundry dengan mudah.

Dengan aplikasi ini, instansi laundry akan dapat meningkatkan efisiensi operasional mereka, meningkatkan kepuasan pelanggan, dan mengurangi risiko kesalahan dalam manajemen bisnis mereka.

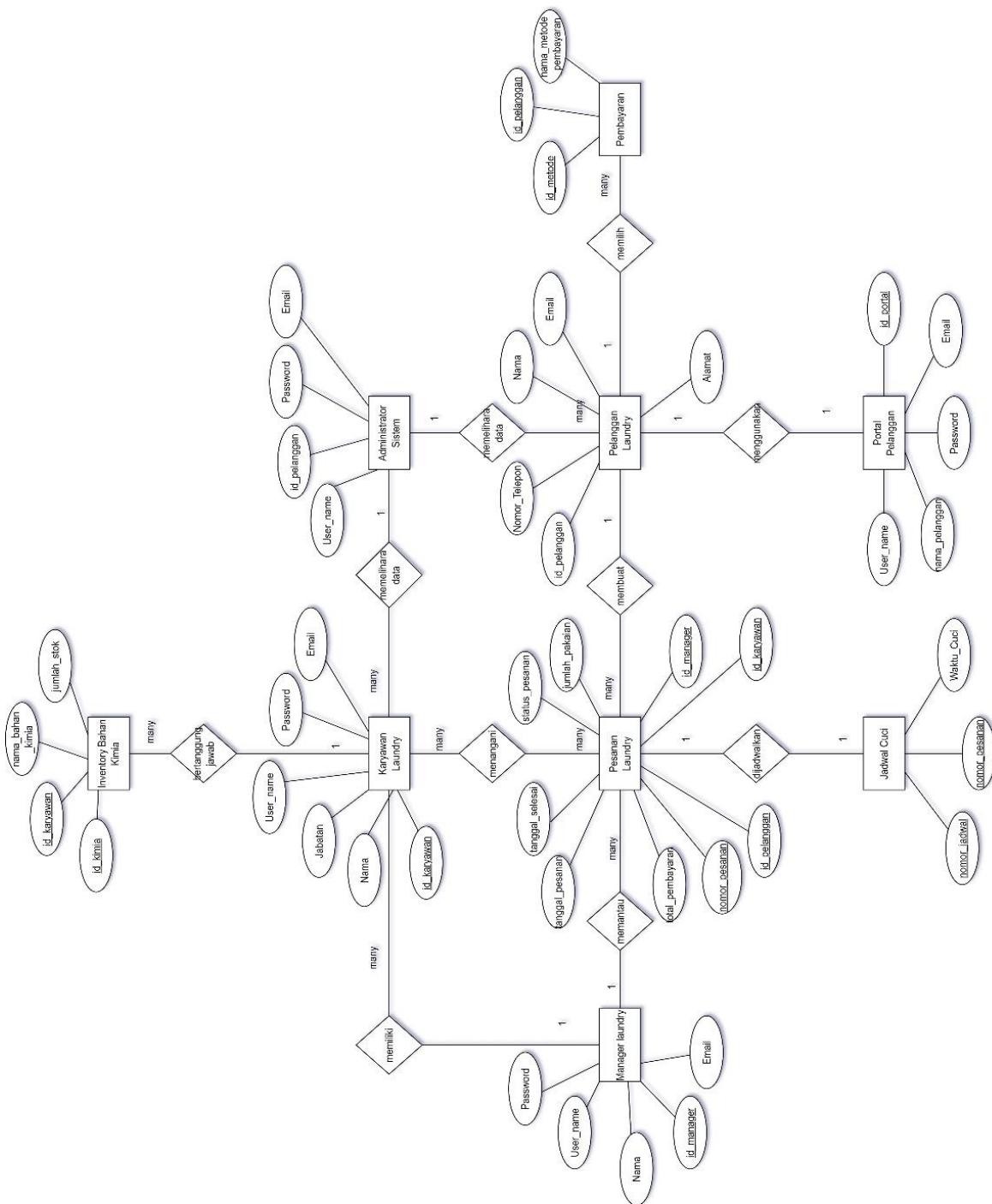
Pengguna Aplikasi [5 poin]

Pengguna aplikasi manajemen laundry dengan sistem modern meliputi:

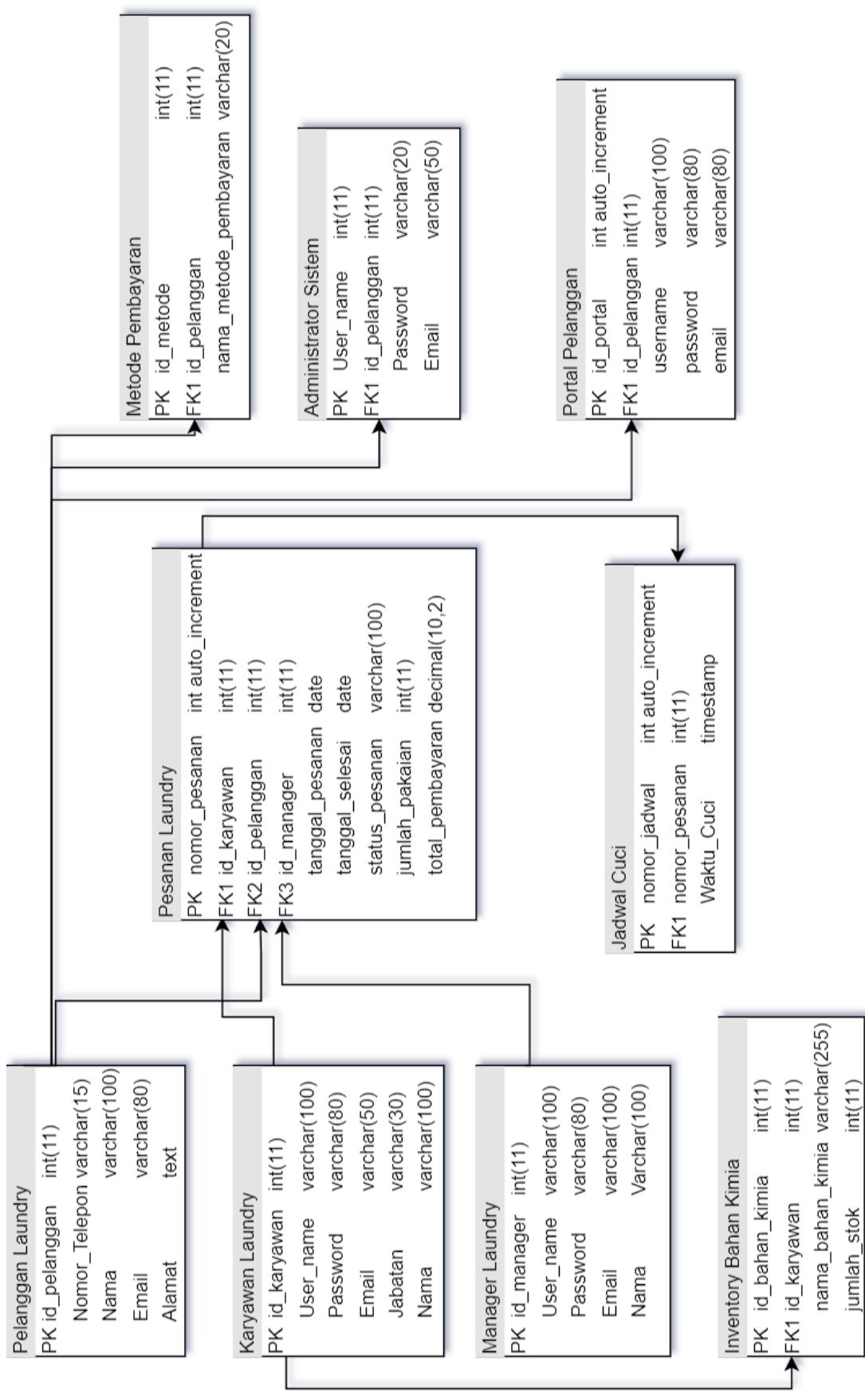
1. Manajer Laundry: Mereka akan menggunakan aplikasi untuk mengelola operasional harian, mengecek antrian, memantau inventori, dan mengelola pesanan pelanggan.
2. Karyawan Laundry: Karyawan akan menggunakan aplikasi untuk melihat jadwal kerja, mengetahui tugas yang harus dilakukan, dan mencatat informasi tentang pesanan yang sedang diproses.
3. Pelanggan Laundry: Pelanggan akan mengakses aplikasi untuk melakukan pemesanan, melacak status pesanan mereka, memberikan umpan balik, dan melakukan pembayaran.
4. Administrator Sistem: Mereka bertanggung jawab atas pemeliharaan dan pembaruan aplikasi, serta memastikan bahwa semua sistem berjalan dengan lancar dan aman.

Dengan penggunaan aplikasi yang terintegrasi, semua pihak terlibat dapat berinteraksi dengan mudah, menghemat waktu, dan meningkatkan produktivitas secara keseluruhan.

Desain Entity-Relationship Diagram (ERD) [10 poin]



Desain Relasi Basis Data [10 poin]



Query Data Definition Language (DDL) [25 poin]

```
CREATE DATABASE laundry;
CREATE TABLE pelanggan (
    id_pelanggan INT PRIMARY KEY,
    nomor_telepon VARCHAR(15),
    nama VARCHAR(100),
    email VARCHAR(80),
    alamat TEXT
);

CREATE TABLE karyawan (
    id_karyawan INT PRIMARY KEY,
    username VARCHAR(100),
    password VARCHAR(80),
    nama VARCHAR(100),
    email VARCHAR(50)
);

CREATE TABLE manager (
    id_manager INT PRIMARY KEY,
    username VARCHAR(100),
    password VARCHAR(80),
    nama VARCHAR(100),
    email VARCHAR(100)
);

CREATE TABLE inventory_bahan_kimia (
    id_bahan_kimia INT PRIMARY KEY,
    id_karyawan INT,
    nama_bahan_kimia VARCHAR(255),
    jumlah_stok INT,
    FOREIGN KEY (id_karyawan) REFERENCES karyawan(id_karyawan)
);

CREATE TABLE pesanan_laundry (
    nomor_pesanan INT AUTO_INCREMENT PRIMARY KEY ,
    id_karyawan INT,
```

```

id_pelanggan INT,
id_manager INT,
tanggal_pesanan DATE,
tanggal_selesai DATE,
status_pesanan VARCHAR(100),
jumlah_pakaian INT,
total_pembayaran DECIMAL(10,2),
FOREIGN KEY (id_karyawan) REFERENCES karyawan(id_karyawan),
FOREIGN KEY (id_pelanggan) REFERENCES pelanggan(id_pelanggan),
FOREIGN KEY (id_manager) REFERENCES manager(id_manager)
);

CREATE TABLE jadwal_cuci (
nomor_jadwal INT AUTO_INCREMENT PRIMARY KEY,
nomor_pesanan INT,
waktu_cuci DATETIME,
FOREIGN KEY (nomor_pesanan) REFERENCES
pesanan_laundry(nomor_pesanan)
);

CREATE TABLE portal_pelanggan (
id_portal INT AUTO_INCREMENT PRIMARY KEY,
id_pelanggan INT,
username VARCHAR(100),
password VARCHAR(80),
email VARCHAR(80),
FOREIGN KEY (id_pelanggan) REFERENCES pelanggan(id_pelanggan)
);

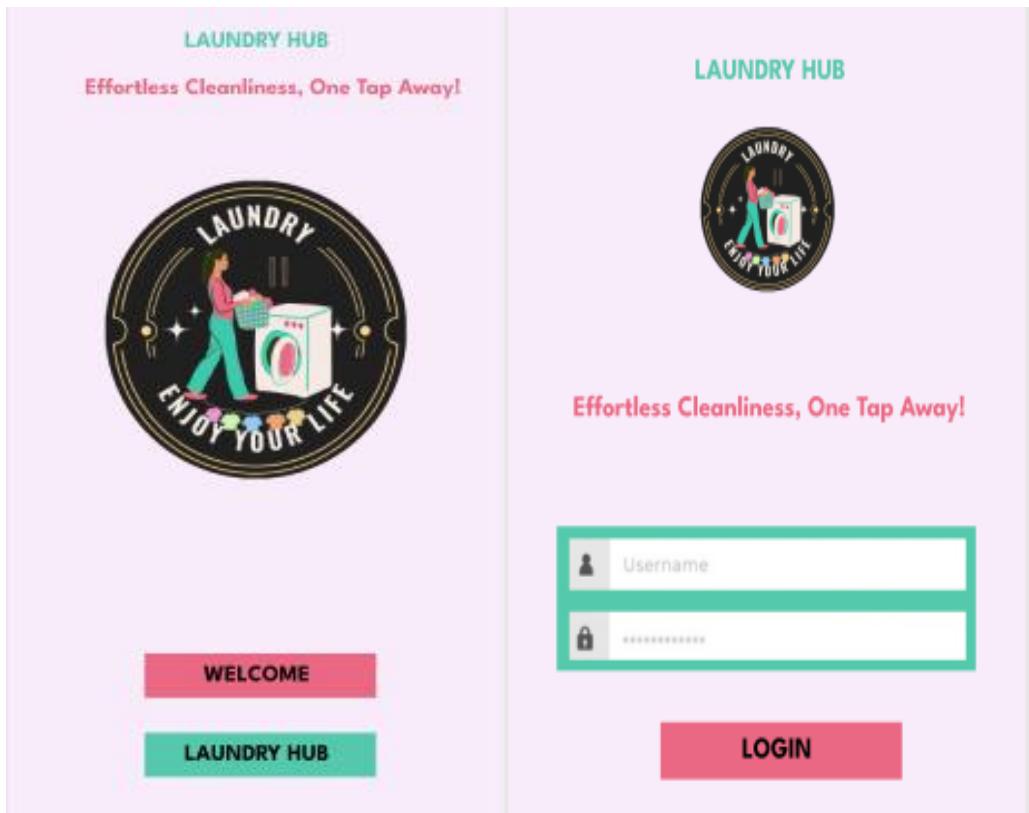
CREATE TABLE pembayaran (id_metode_pembayaran INT PRIMARY KEY,
Id_pelanggan INT,
metode_pembayaran varchar (20),
FOREIGN KEY (id_pelanggan) REFERENCES pelanggan(id_pelanggan)
);

```

Desain Interface (Wireframe/Mockup) Beserta Query-nya [poin 40]

1. Pelanggan

Berikut ini merupakan laman masuk untuk pengguna.

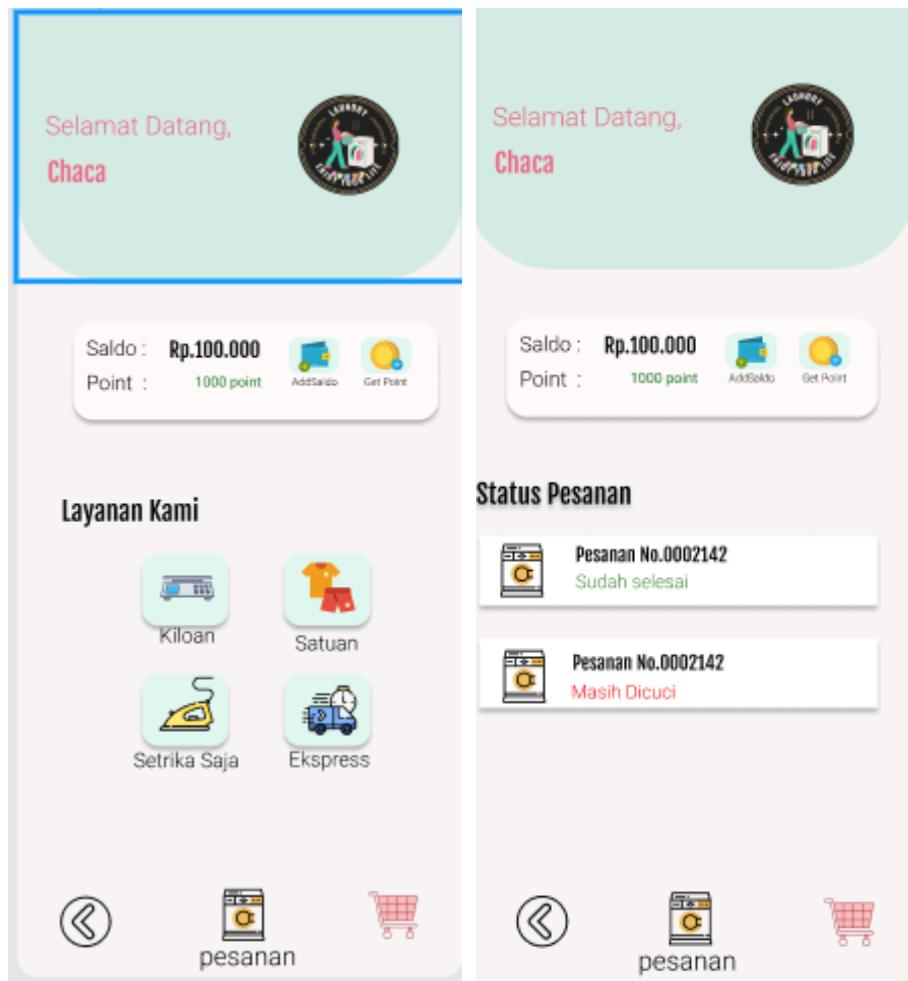


SELECT * FROM pelanggan

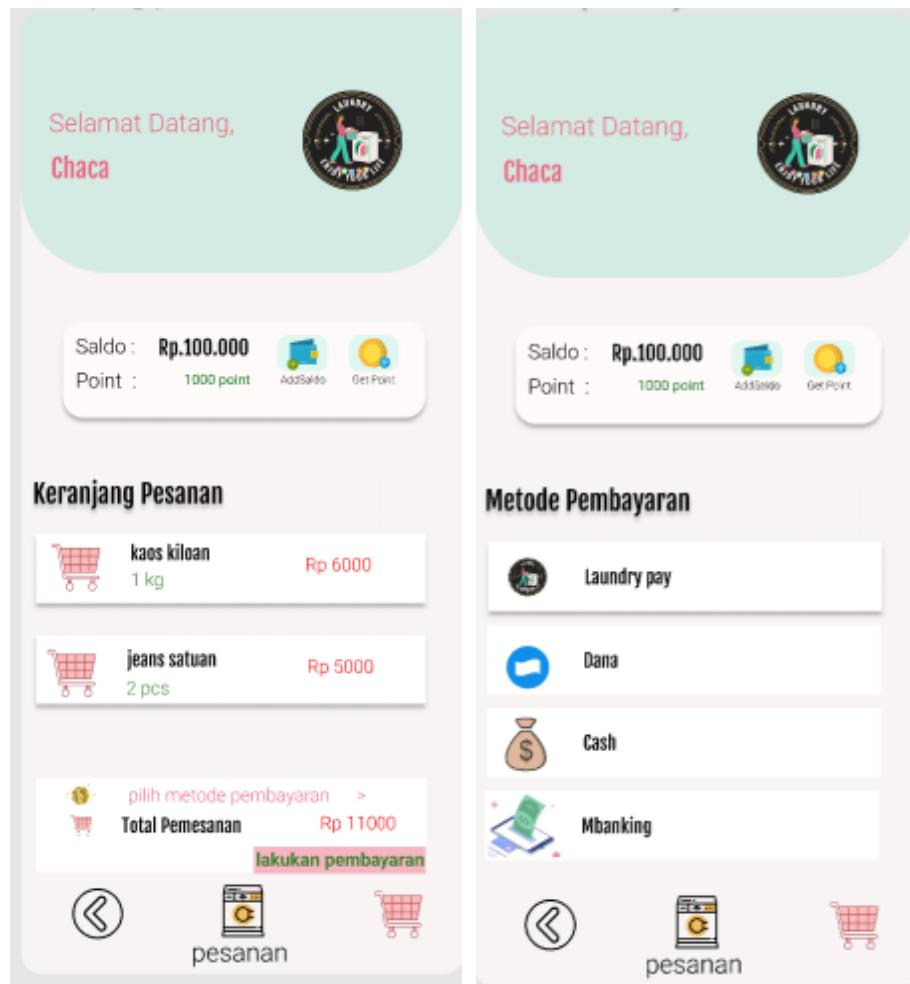
WHERE username = username_input AND password = password_input

```
CREATE TABLE portal_pelanggan (
    id_portal INT AUTO_INCREMENT PRIMARY KEY,
    id_pelanggan INT,
    username VARCHAR(100),
    password VARCHAR(80),
    email VARCHAR(80),
    FOREIGN KEY (id_pelanggan) REFERENCES pelanggan(id_pelanggan)
);
```

Menu utama pengguna



```
SELECT status_pesanan FROM pesanan_laundry
```



SELECT jumlah_pakaian AND total_pembayaran FROM pesanan_laundry

Halaman Pembayaran dan Bukti bayar ketika pelanggan sudah melakukan pembayaran.

The image displays two side-by-side screenshots of the LaundryHub mobile application interface, illustrating the payment process for a laundry order.

Left Screenshot (Info Pembayaran):

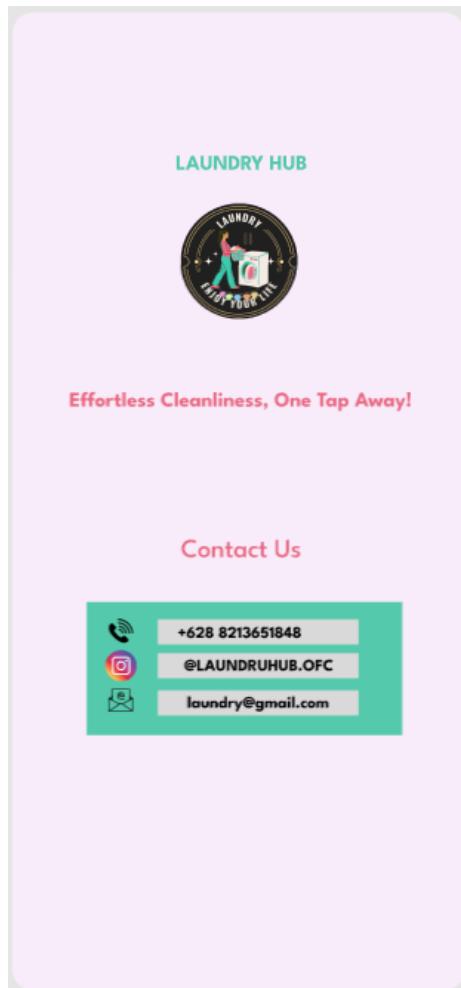
- Welcome:** "Selamat Datang, Chaca" with a user icon.
- Balance:** Saldo : Rp.100.000
- Points:** Point : 1000 point, AddSaldo, Get Point
- Payment Information:**
 - Total Pembayaran:** Rp 11.000
 - Metode Pembayaran:** Mbanking
 - No Virtual Account:** 246176892
- Buttons:** "lakukan pembayaran" (Perform payment), back arrow, "pesanan" (orders), and shopping cart icon.

Right Screenshot (Rincian Pembayaran):

- Welcome:** "Selamat Datang, Chaca" with a user icon.
- Balance:** Saldo : Rp.100.000
- Points:** Point : 1000 point, AddSaldo, Get Point
- Payment Details:**
 - ke:** LaundryHub
 - Metode Pembayaran:** Mbanking
 - Total Biaya:** Rp 11.000
 - No Transaksi:** 5217219
- Status:** A large green circle with a white checkmark and the word "selesai" (finished).
- Buttons:** back arrow, "pesanan" (orders), and shopping cart icon.

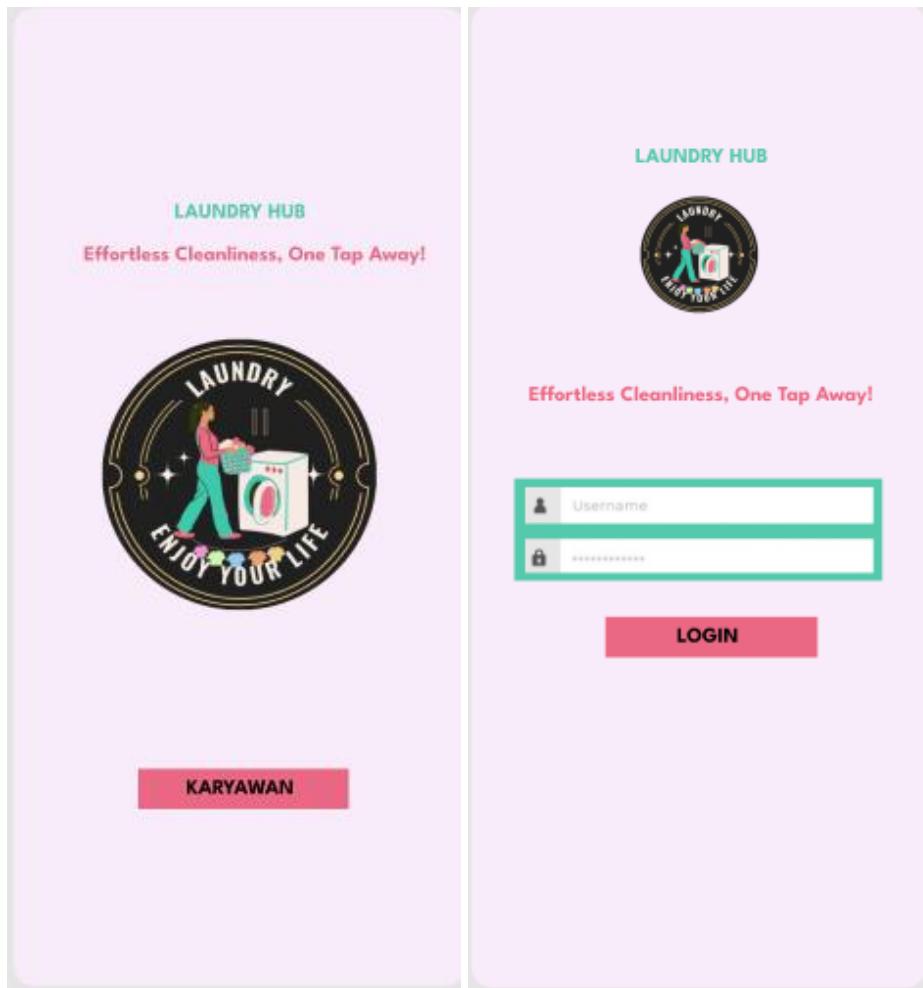
SELECT total_pembayaran FROM pesanan_laundry;

Laman bantuan pengguna yang menyediakan kontak



2. Karyawan Laundry

Menu awal untuk karyawan ketika mau membuka aplikasi



SELECT * FROM karyawan

WHERE username = username_input AND password = passwor_input;

The image shows three separate mobile application screens arranged horizontally. Each screen has a light blue header with a circular logo containing a person washing laundry.

- Karyawan Apps** (Left Screen):
 - Welcome message: Selamat Datang Karyawan, Dani
 - Icon: LaundryHUB logo with text "semangat bekerja"
 - Section: **Jadwal Kerja** (Working Schedule)

Hari	Waktu
Senin	07.00 - 12.00
Selasa	07.00 - 12.00
Rabu	12.00 - 17.00
Kamis	12.00 - 17.00
Jumat	07.00 - 12.00
Sabtu	12.00 - 17.00
 - Section: **Tugas** (Tasks)

Tugas	No pesanan
Cuci	No pesanan. 124353
Jemur	No pesanan. 124353
Setrika	No pesanan. 124353
Setrika	No pesanan. 122353
Cuci	No pesanan. 125553
Jemur	No pesanan. 122353
- Jadwal Kerja** (Middle Screen):
 - Welcome message: Selamat Bekerja Dani
 - Icon: LaundryHUB logo
 - Section: **Jadwal Kerja** (Working Schedule)

Hari	Waktu
Senin	07.00 - 12.00
Selasa	07.00 - 12.00
Rabu	12.00 - 17.00
Kamis	12.00 - 17.00
Jumat	07.00 - 12.00
Sabtu	12.00 - 17.00
- Tugas** (Right Screen):
 - Welcome message: Selamat Bekerja Dani
 - Icon: LaundryHUB logo
 - Section: **Tugas** (Tasks)

Tugas	No pesanan
Cuci	No pesanan. 124353
Jemur	No pesanan. 124353
Setrika	No pesanan. 124353
Setrika	No pesanan. 122353
Cuci	No pesanan. 125553
Jemur	No pesanan. 122353

```
SELECT nomor_pesanan AND tanggal_pesanan FROM pesanan_laundry;
```

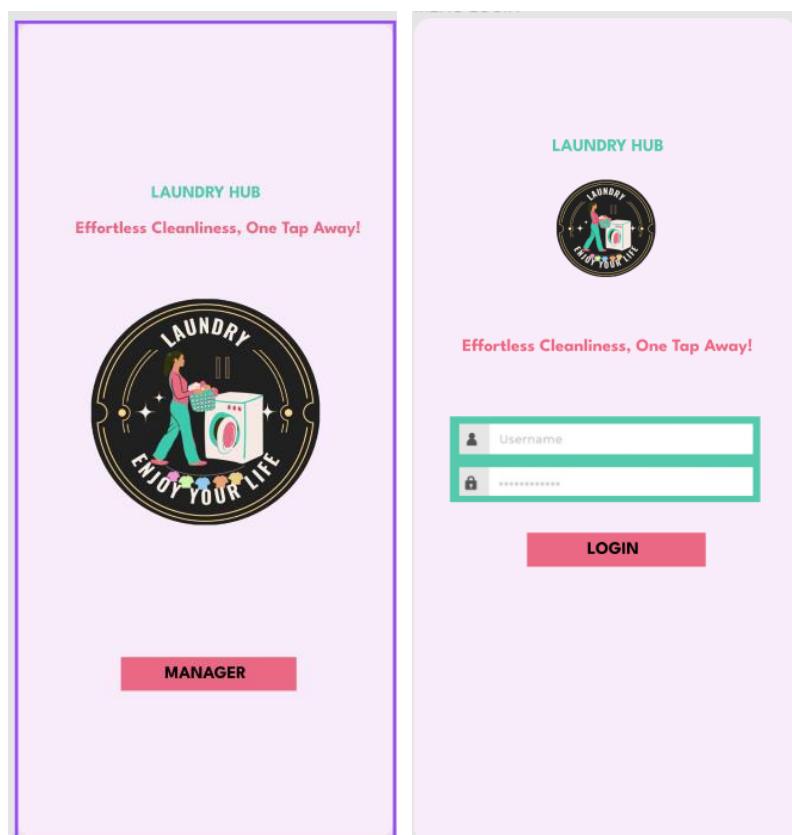
The image shows a single mobile application screen displaying laundry order details. It features a light blue header with a circular logo containing a person washing laundry.

- Informasi Pesanan** (Order Information):
 - Icon: LaundryHUB logo
 - Text: Pesanan No.0002142
cuci
 - Text: Pesanan No.0002142
setrika
 - Text: Pesanan No.0002142
cuci
 - Text: Pesanan No.0002142
setrika saja

```
SELECT * FROM jadwal_cuci;
```

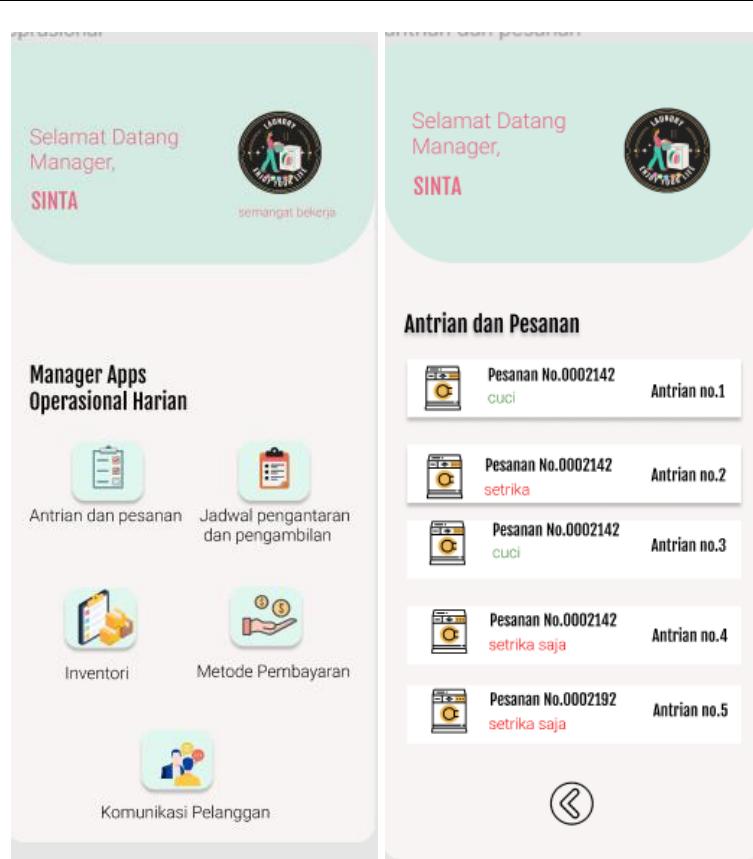
3. Manager

Menu awal untuk manager ketika ingin membuka aplikasi



SELECT * FROM MANAGER

WHERE username = username_input AND password = password_input



SELECT * FROM pesanan_laumdry



SELECT * FROM pesanan_laundry

Stok Inventori

Stok 10	Stok 12	Stok 10
Deterjen	Pewangi pakaian	Pemutih pakaian
Stok 13	Stok 11	Kemasan dan Tas
Penghilang noda		

Metode Pembayaran

Pesanan No.0002142 Rp 15.000	Cash
Pesanan No.0002142 Rp 20.000	Cash
Pesanan No.0002142 Rp 21.000	Bank
Pesanan No.0002142 Rp 17.000	Bank
Pesanan No.0002192 Rp 20.000	Cash

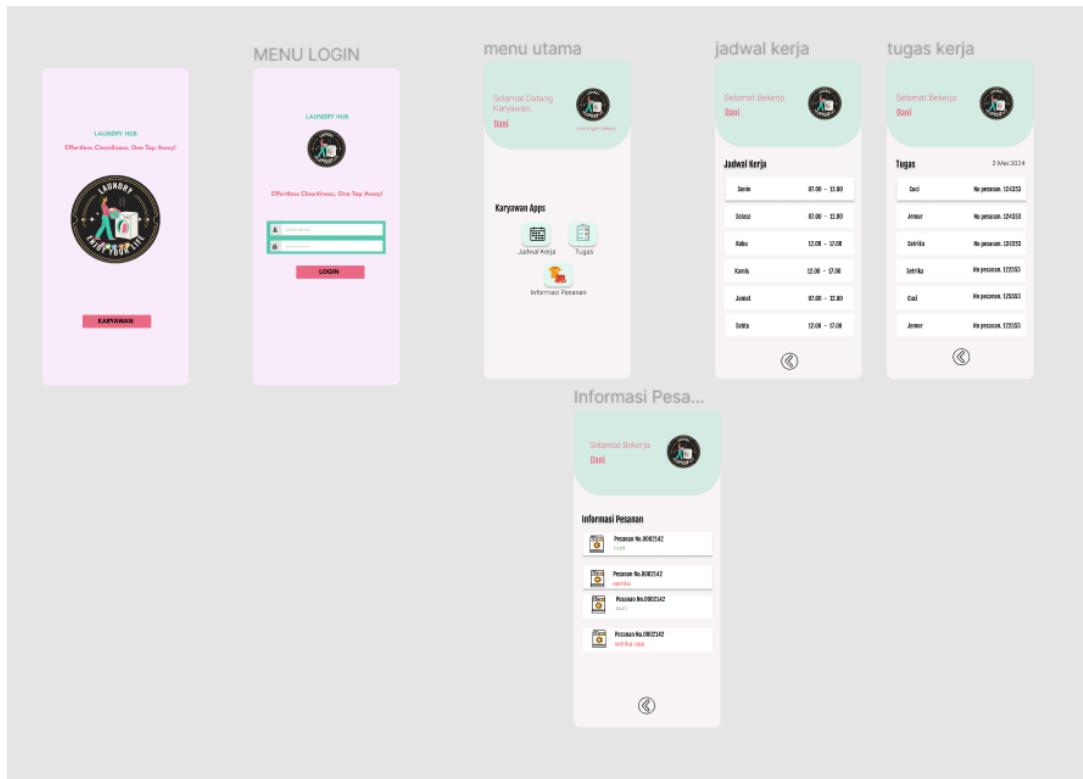
Chat Pelanggan

- Dania: wahh harum banget bajunya...
- Azam: makasih min udah di anter cepet...
- Cia: bagus banget pelayanannya nih..
- Husen: thanks min...
- Akbar: iya min nanti saya ambil...

The wireframe shows the following sections:

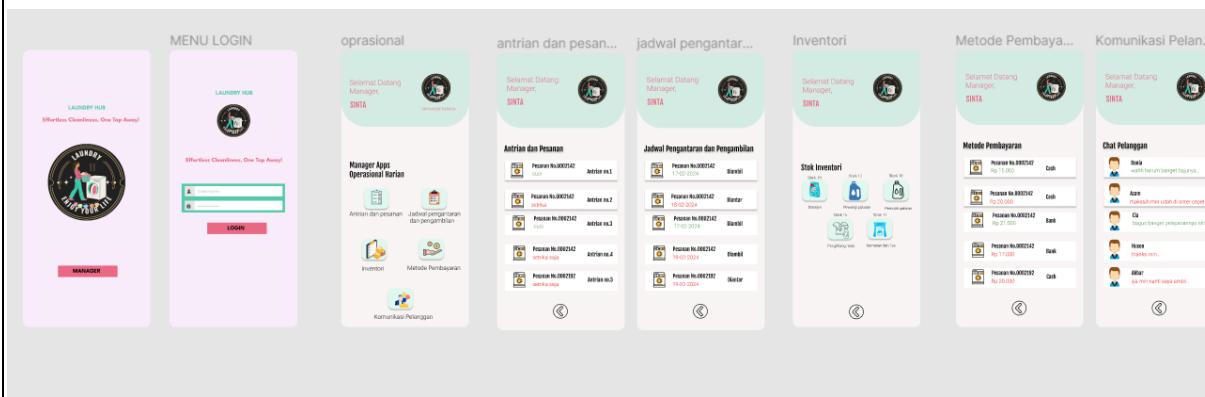
- Top Navigation:** MENU AW..., MENU LOGIN, menu pesanan, pesanan aktif, keranjang pe..., metode pem..., info pembay..., info pembay...
- Customer Support (CS):** CS screen with contact information.
- Product Categories:** kiloan, SATUAN, Setrika saja, EKSPRESS.
- Order Status:** Delivered, Pending, Processing, Shipped, Ready to Collect.
- Payment Methods:** CIMB Clicks, Mandiri Clickpay, Mandiri Clickpay QR, Mandiri Clickpay Mandiri Debit/Credit, Mandiri Clickpay Mandiri Debit/Credit QR, Mandiri Clickpay Mandiri Debit/Credit Mandiri Debit/Credit QR.
- Delivery Options:** Same Day, Next Day, 24 Hours, 48 Hours, 72 Hours.

App pelanggan Laundry



App Karyawan Laundry

UI KESELURUHAN



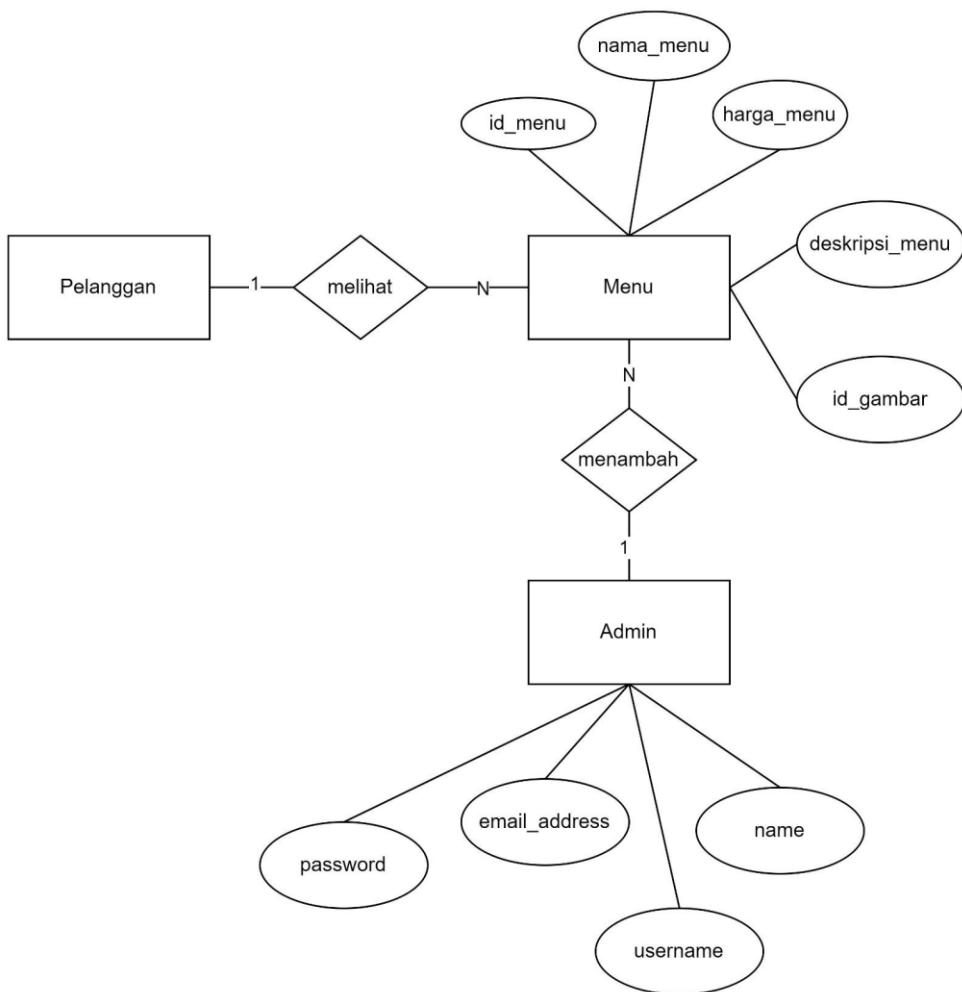
App Manager Laundry



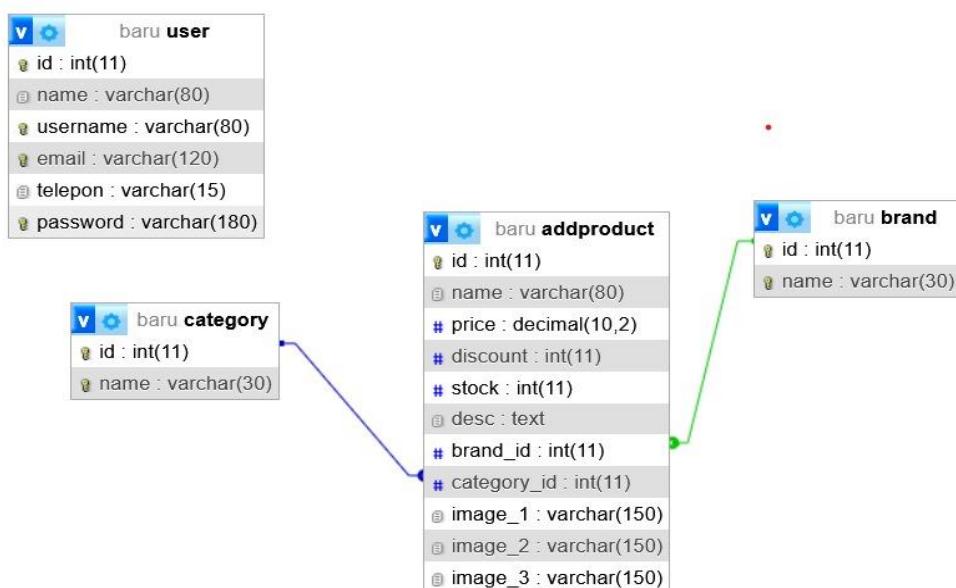
Administrator sistem

Judul Aplikasi	
VISUALISASI MENU CAFE	
Dikembangkan Oleh	<ul style="list-style-type: none"> • 5220411312_Dyah Prameswari Cinta Herlanti • 5220411298_Moachmamad Aurell Ernandi • 5220411302_Haikal Fawwaz Karim • 5220411307_Dian Febry Setyadi • 5220411314_Firza Mauli • 52204411318_Faras Sulthan Pratama
Deskripsi Singkat Instansi, Permasalahan, dan Solusi Aplikasi Yang Akan Dikembangkan [10 poin]	Kopi Pengantar dikategorikan sebagai usaha kecil menengah (UKM) yang dikelola secara mandiri. Instansi ini memiliki produk yang di jual di tempatnya yang dapat di beli oleh orang yang akan memesan kopi di sana, instansi ini tidak memiliki tampilan produk yang di tampilkan di sana secara visual, sehingga pelanggan dapat melihat dulu produk yang di dapat sebelum membeli produknya
Pengguna Aplikasi [5 poin]	Targetnya adalah Palanggan atau pengunjung cafe

Desain Entity-Relationship Diagram (ERD) [10 poin]



Desain Relasi Basis Data [10 poin]



Query Data Definition Language (DDL) [25 poin]**Query DDL Website:**

```
CREATE TABLE Addproduct (
    id SERIAL PRIMARY KEY,
    name VARCHAR(80) NOT NULL,
    price NUMERIC(10,2) NOT NULL,
    discount INTEGER DEFAULT 0,
    stock INTEGER NOT NULL,
    desc TEXT NOT NULL,
    brand_id INTEGER NOT NULL,
    category_id INTEGER NOT NULL,
    FOREIGN KEY (brand_id) REFERENCES Brand(id),
    FOREIGN KEY (category_id) REFERENCES Category(id)
);

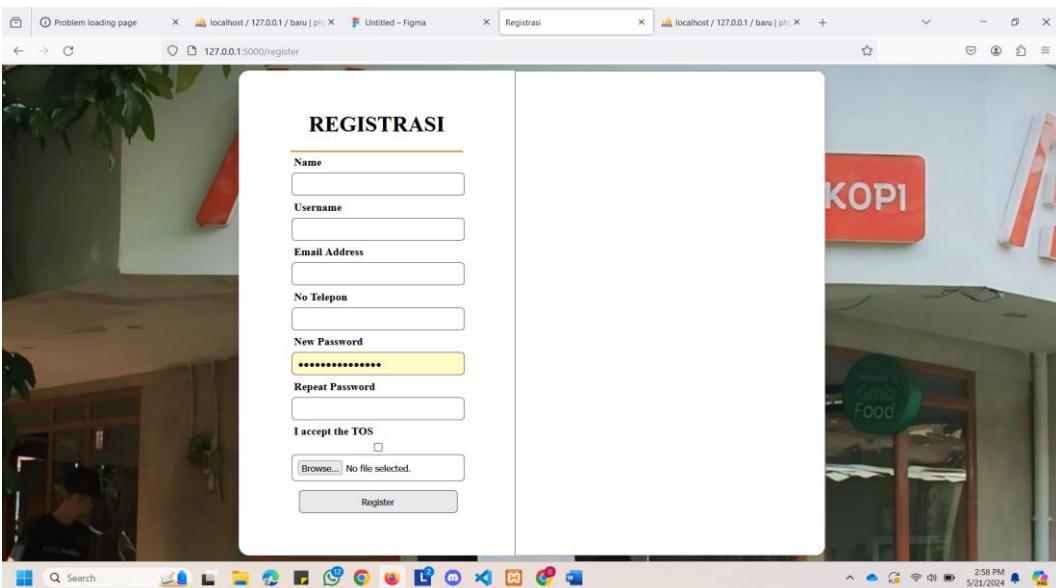
CREATE TABLE User (
    id SERIAL PRIMARY KEY,
    name VARCHAR(80) NOT NULL,
    username VARCHAR(80) UNIQUE NOT NULL,
    email VARCHAR(120) UNIQUE NOT NULL,
    telepon VARCHAR(120) UNIQUE NOT NULL,
    password VARCHAR(180) NOT NULL,
);

CREATE TABLE Brand (
    id SERIAL PRIMARY KEY,
    name VARCHAR(30) NOT NULL UNIQUE
);

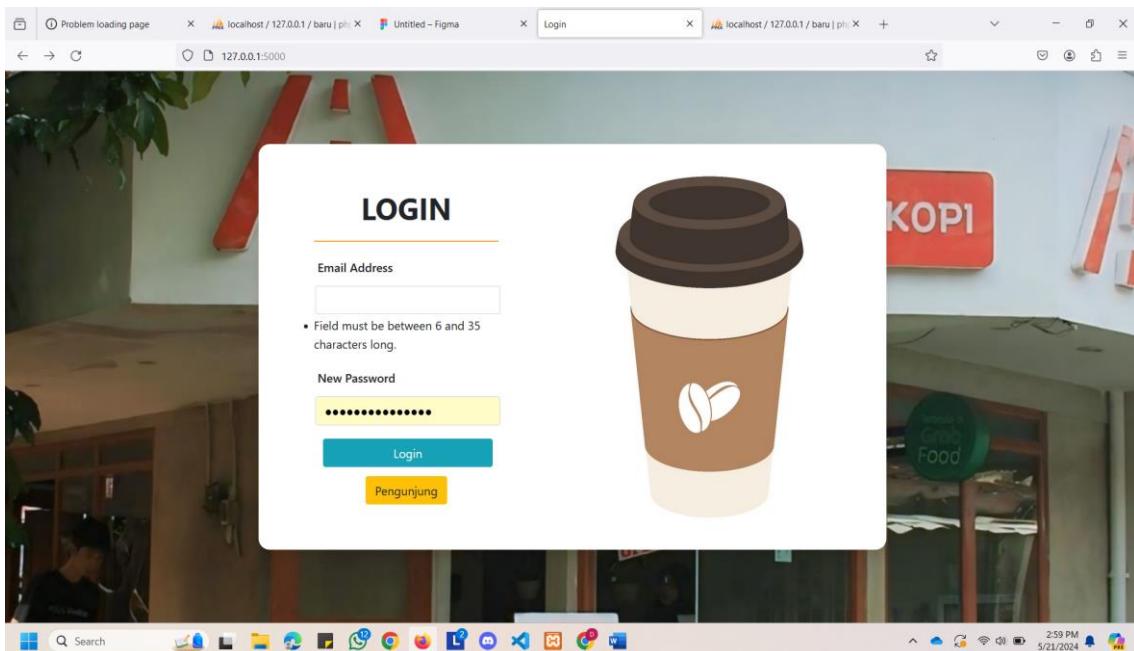
CREATE TABLE Category (
    id SERIAL PRIMARY KEY,
    name VARCHAR(30) NOT NULL UNIQUE
);
```

Desain Interface (Wireframe/Mockup) Beserta Query-nya [poin 40]

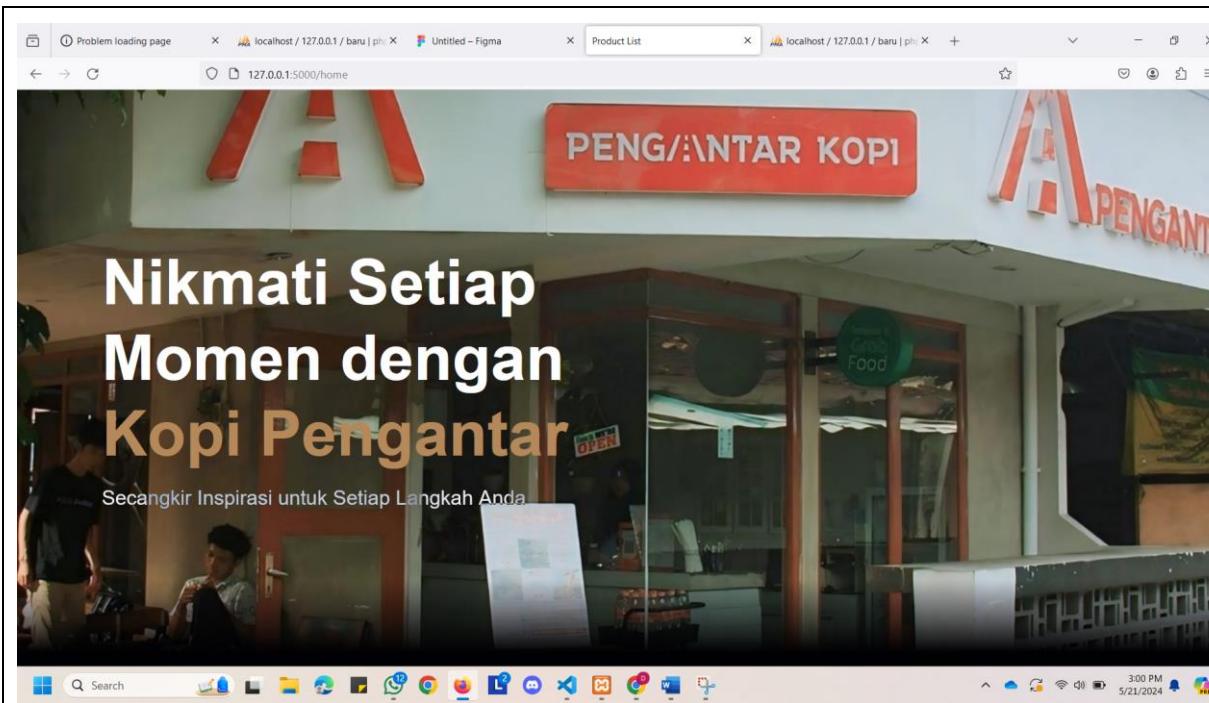
Desain Interface Website:



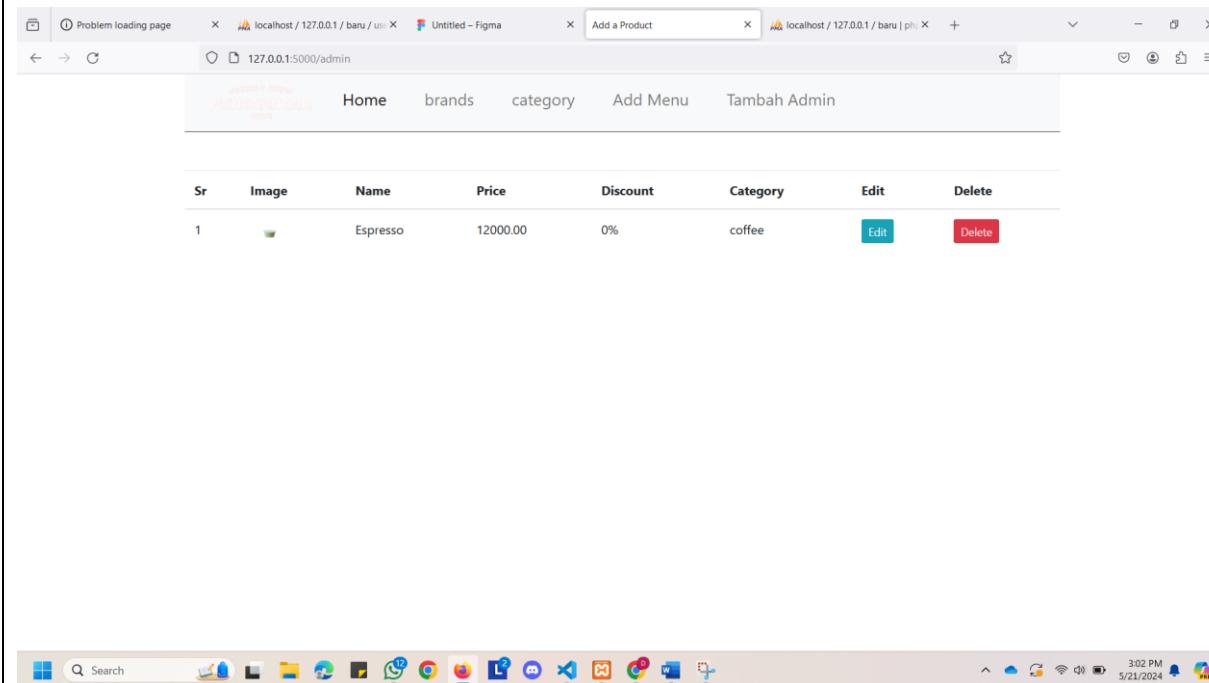
```
user = User(name=form.name.data, username=form.username.data,
email=form.email.data,
telepon=form.telepon.data, password=hash_password)
db.session.add(user)
db.session.commit()
```



```
user = User.query.filter_by(email = form.email.data).first()
```



```
products = Addproduct.query.all()
```



Sr	Image	Name	Price	Discount	Category	Edit	Delete
1		Espresso	12000.00	0%	coffee	Edit	Delete

```
products = Addproduct.query.all()
```

Add a Menu

Name

Price

Discount

Stock

Description

Add a brand

Select A brand

Add a category

Add a Category

Image 1

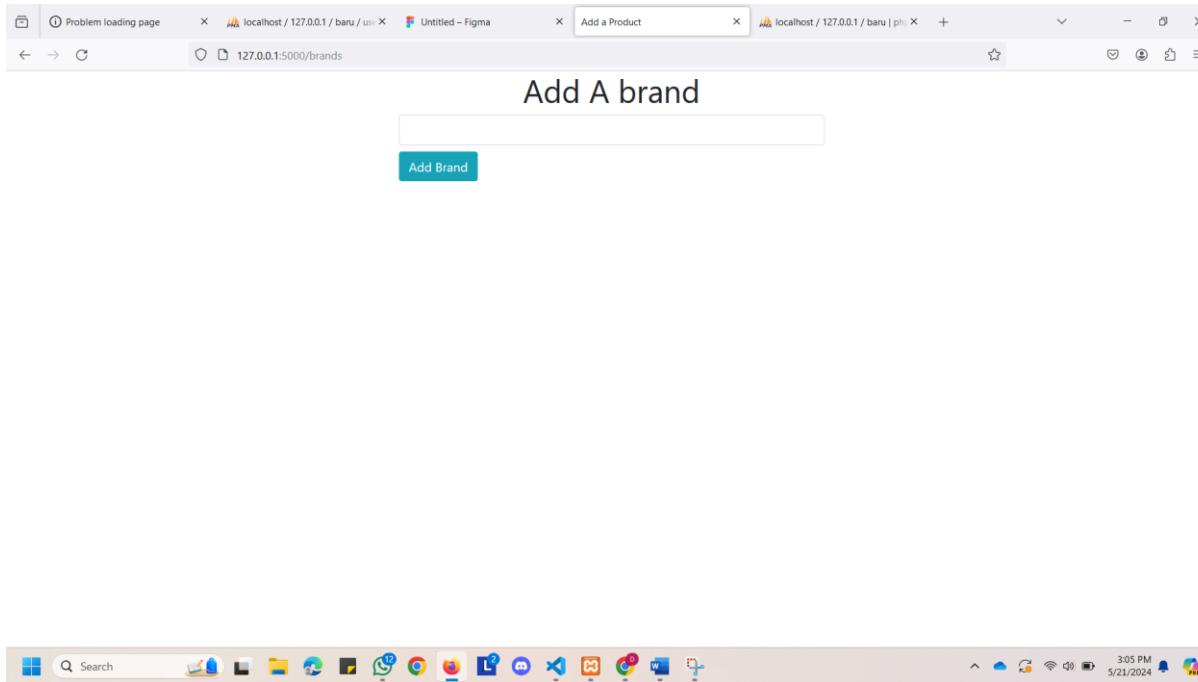
Browse... No file selected.

```
addproduct = Addproduct(name=name, price=price, discount=discount,
stock=stock,
desc=desc, category_id=category,
brand_id=brand,
image_1=image_1, image_2=image_2,
image_3=image_3)
```

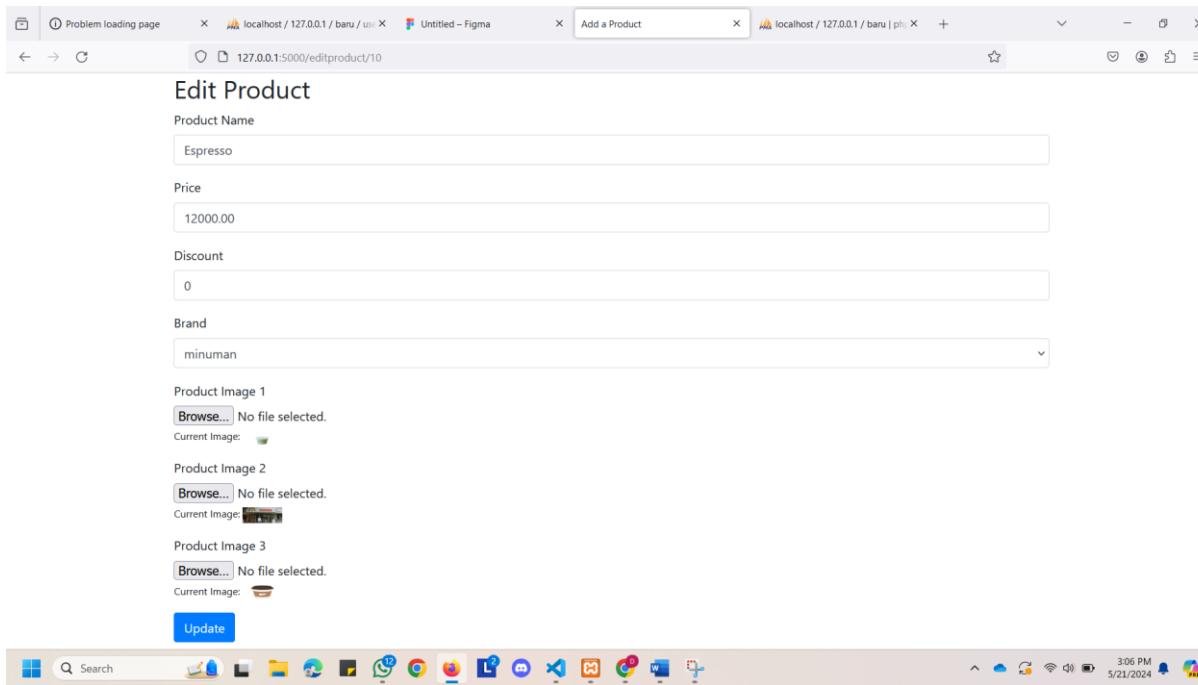
Add A category

```
getcat = request.form.get('category')
category = Category(name=getcat)
```

```
db.session.add(category)
```



```
getbrand = request.form.get('brand')
brand = Brand(name=getbrand)
db.session.add(brand)
```



```
product = Addproduct.query.get_or_404(product_id)
```

```
db.session.commit()  
brands = Brand.query.all()
```

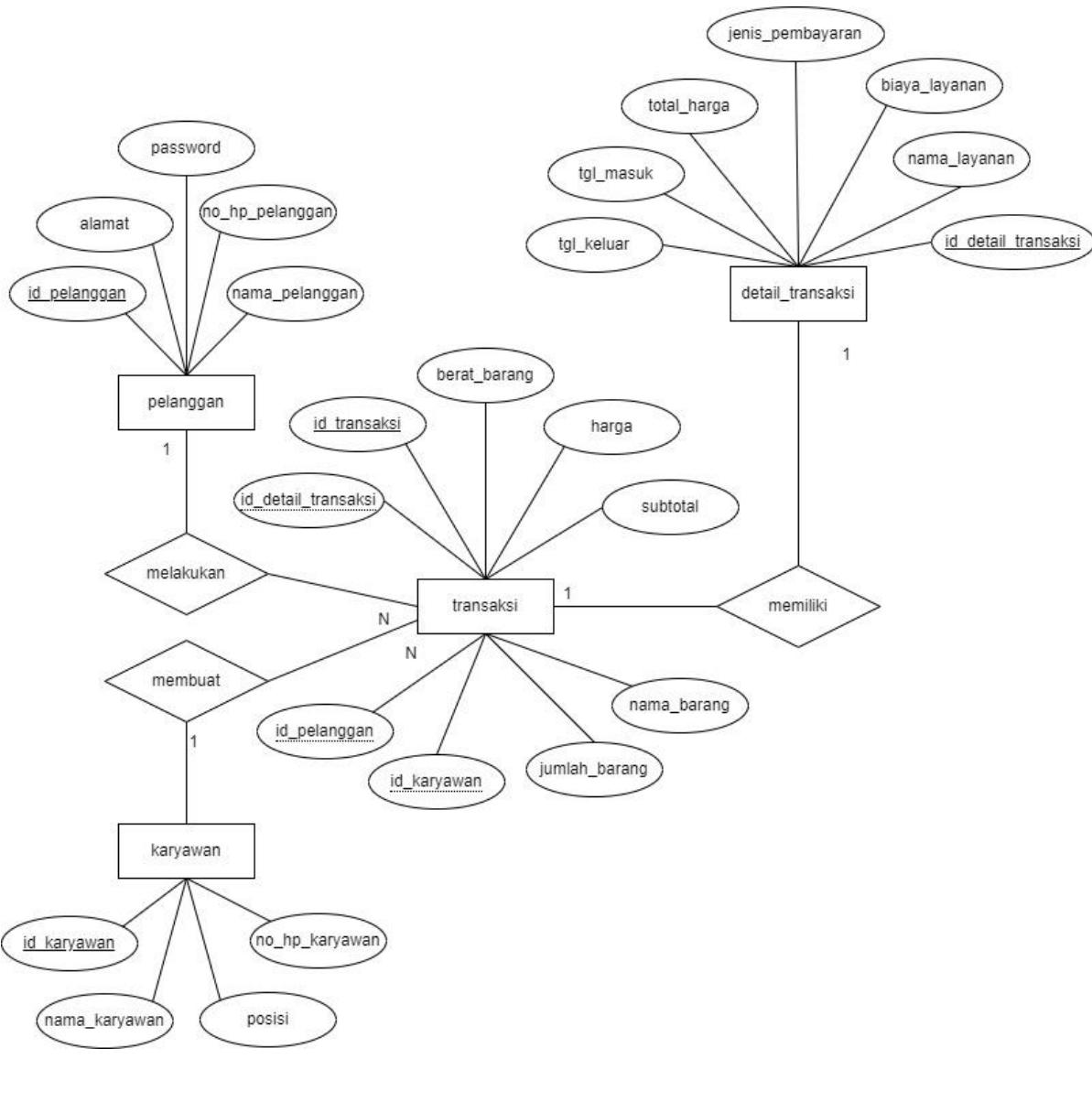
<p>Judul Aplikasi</p> <h1>PERANCANGAN SISTEM MANAJEMEN LAUNDRY BERBASIS DIGITAL</h1>	
<p>Dikembangkan Oleh</p> <ul style="list-style-type: none"> • 5220411294 – Andata Leonardo • 5220411304 – Ahrenz Dianto • 5220411313 – Tariska Indah Rahmawati • 5220411315 – Restyandito Vega Manuhara • 5220411319 – Siti Inayatul Mufidah 	
<p>Deskripsi Singkat Instansi, Permasalahan, dan Solusi Aplikasi Yang Akan Dikembangkan [10 poin]</p>	
<p>Hans Laundry, yang terletak di Kutu Patran, Sinduadi, Kecamatan Mlati, Kabupaten Sleman, Daerah Istimewa Yogyakarta 55284. Merupakan sebuah bidang yang fokus pada kualitas dan kebersihan, Hans Laundry menyediakan layanan cuci, setrika, dan dry cleaning yang unggul. Menggunakan teknologi pencucian terkini dan deterjen berkualitas tinggi, Hans Laundry memastikan setiap pakaian tidak hanya bersih, tetapi juga dirawat dengan teliti agar serat kain dan warna tetap terjaga. Layanan setrika mereka menghasilkan pakaian yang rapi dan bebas kerutan, sementara dry cleaning menggunakan pelarut khusus yang aman untuk kain halus dan sensitif.</p>	
<p>Permasalahan:</p> <ol style="list-style-type: none"> 1. Nota masih berupa kertas tulisan 2. Pengumpulan Informasi nota masih berupa pembukuan manual 3. Masih belum terdapat nomor resi 4. Pembayaran masih terbatas 5. Pemesanan belum bisa dilakukan secara online 	
<p>Solusi Permasalahan:</p> <ol style="list-style-type: none"> 1. Untuk permasalahan nota yang masih berupa kertas tulisan, kita memberikan solusi yang memungkinkan pelanggan untuk menerima nota digital melalui whatsapp, hal ini berguna untuk mengurangi penggunaan kertas dan mempermudah manajemen catatan. 2. Untuk mengatasi masalah pengumpulan informasi nota masih berupa pembukuan manual, kita memberikan solusi berupa membuat rancangan sistem database sesuai dengan kebutuhan laundry. 	

3. Untuk permasalahan belum terdapat nomor resi, kita menawarkan solusi untuk setiap pesanan diberikan nomor resi unik yang dapat digunakan pelanggan untuk melacak status pesanan.
4. Untuk permasalahan pembayaran, kita menambahkan opsi pembayaran selain cash, yang berupa QRIS, transfer bank, kartu kredit, dan dompet digital lainnya.
5. Untuk masalah pemesanan belum bisa dilakukan secara online, kita memberikan solusi dengan menciptakan platform/aplikasi yang memungkinkan pelanggan untuk memesan layanan laundry secara online.

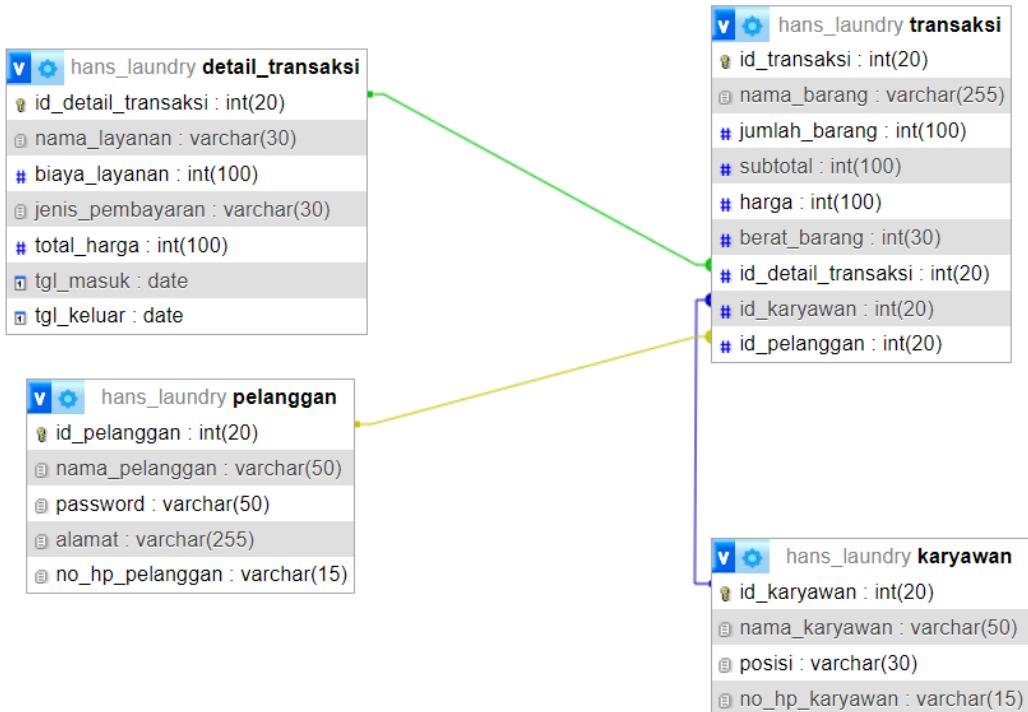
Pengguna Aplikasi [5 poin]

Aplikasi ini rencananya akan digunakan oleh pelanggan dan pemilik

Desain Entity-Relationship Diagram (ERD) [10 poin]



Desain Relasi Basis Data [10 poin]



Query Data Definition Language (DDL) [25 poin]

```
CREATE TABLE pelanggan(
    id_pelanggan INT(20) NOT NULL AUTO_INCREMENT PRIMARY KEY,
    nama_pelanggan VARCHAR(50) NOT NULL,
    password VARCHAR(50) NOT NULL,
    alamat VARCHAR(255) NOT NULL,
    no_hp_pelanggan VARCHAR(15) NOT NULL);

CREATE TABLE karyawan(
    id_karyawan INT(20) NOT NULL AUTO_INCREMENT PRIMARY KEY,
    nama_karyawan VARCHAR(50) NOT NULL,
    posisi VARCHAR(30) NOT NULL,
    no_hp_karyawan VARCHAR(15) NOT NULL);

CREATE TABLE transaksi(
    id_transaksi INT(20) NOT NULL AUTO_INCREMENT PRIMARY KEY,
    nama_barang VARCHAR(255) NOT NULL,
    jumlah_barang INT(100) NOT NULL,
```

```
subtotal INT(100) NOT NULL,  
harga INT(100) NOT NULL,  
berat_barang INT(30) NOT NULL,  
id_detail_transaksi INT(20) NOT NULL,  
id_karyawan INT(20) NOT NULL,  
id_pelanggan INT(20) NOT NULL);  
  
CREATE TABLE detail_transaksi(  
    id_detail_transaksi INT(20) NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    nama_layanan VARCHAR(30) NOT NULL,  
    biaya_layanan INT(100) NOT NULL,  
    jenis_pembayaran VARCHAR(30) NOT NULL,  
    total_harga INT(100) NOT NULL,  
    tgl_masuk DATE NOT NULL,  
    tgl_keluar DATE NOT NULL);  
  
ALTER TABLE transaksi  
ADD CONSTRAINT FK_detail_transaksi  
FOREIGN KEY (id_detail_transaksi)  
REFERENCES detail_transaksi(id_detail_transaksi);  
  
ALTER TABLE transaksi  
ADD CONSTRAINT FK_karyawan  
FOREIGN KEY (id_karyawan)  
REFERENCES karyawan(id_karyawan);  
  
ALTER TABLE transaksi  
ADD CONSTRAINT FK_pelanggan  
FOREIGN KEY (id_pelanggan)  
REFERENCES pelanggan(id_pelanggan);
```

Desain Interface (Wireframe/Mockup) Beserta Query-nya [poin 40]

1. Menu Login



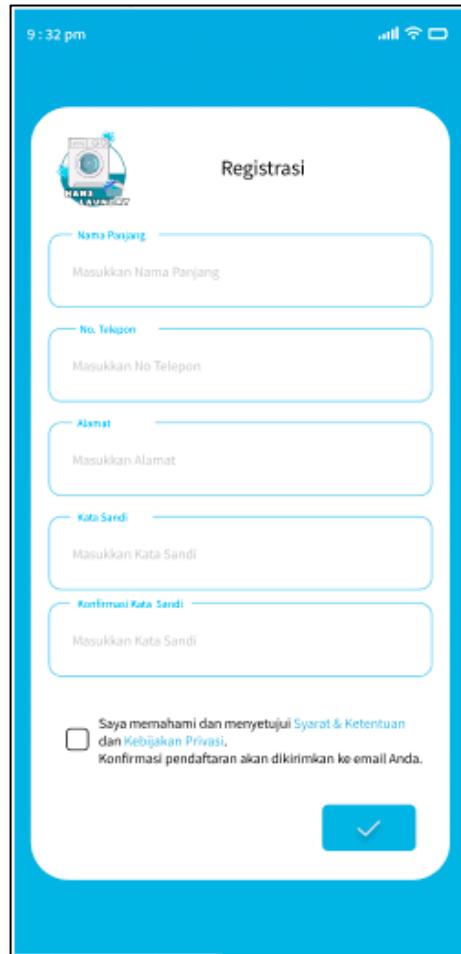
SELECT * FROM pelanggan WHERE phone_number = '+62xxxxxxxxx';

2. Menu Login 2



```
SELECT* FROM pelanggan WHERE password = 'password_input';
```

3. Registrasi



```
INSERT INTO pelanggan (nama_pelanggan, no_hp_pelanggan, alamat, password) VALUES ('andata', '085673248912', 'godean', '12345');
```

4. Lupa Password



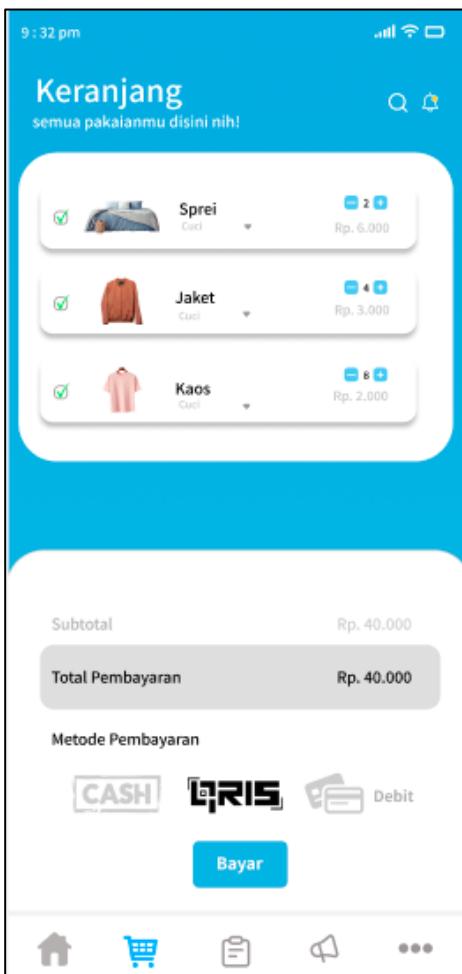
```
UPDATE tabel_pelanggan  
SET password = 'kata_sandi_baru'  
WHERE no_hp_pelanggan = '+62nomor_telepon';
```

5. Memasukkan password baru



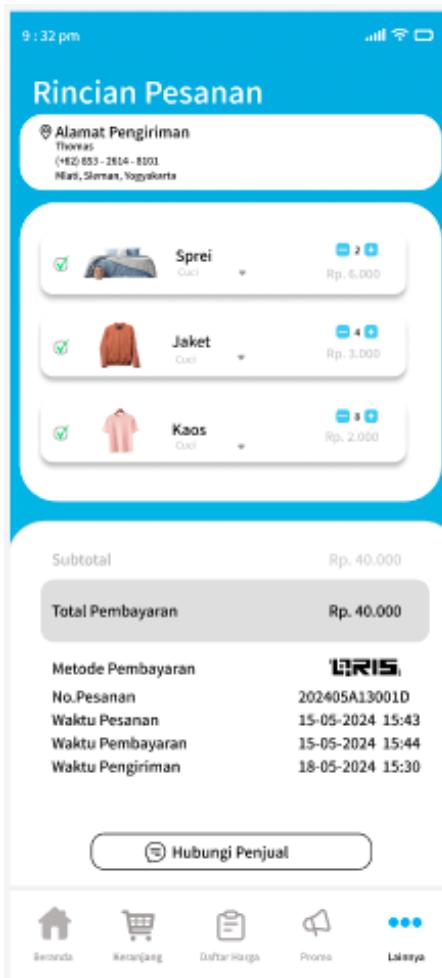
```
UPDATE tabel_pelanggan  
SET password = 'kata_sandi_baru'  
WHERE no_hp_pelanggan = '+62nomor_telepon';
```

6. Menu Keranjang



```
SELECT nama_barang, jumlah_barang, subtotal, harga  
FROM transaksi;
```

7. Menu Detail Transaksi

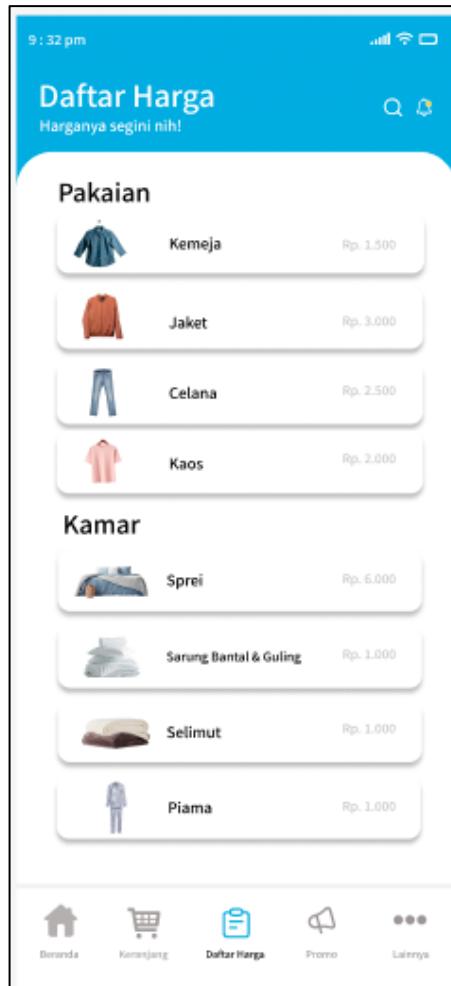


SELECT

```
dt.id_detail_transaksi,  
dt.nama_layanan,  
dt.jenis_pembayaran,  
dt.total_harga,  
dt.tgl_masuk,  
dt.tgl_keluar,  
t.nama_barang,  
t.jumlah_barang,  
t.subtotal,  
t.harga,  
p.nama_pelanggan,  
p.alamat,  
p.no_hp_pelanggan
```

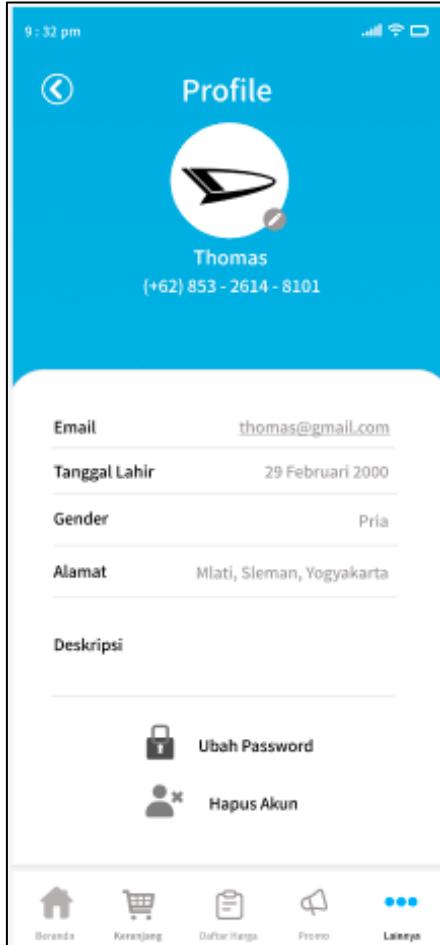
```
FROM  
    detail_transaksi dt  
JOIN  
    transaksi t ON dt.id_detail_transaksi = t.id_detail_transaksi  
JOIN  
    pelanggan p ON t.id_pelanggan = p.id_pelanggan;
```

8. Menu Daftar Harga



```
SELECT nama_barang, harga  
FROM transaksi;
```

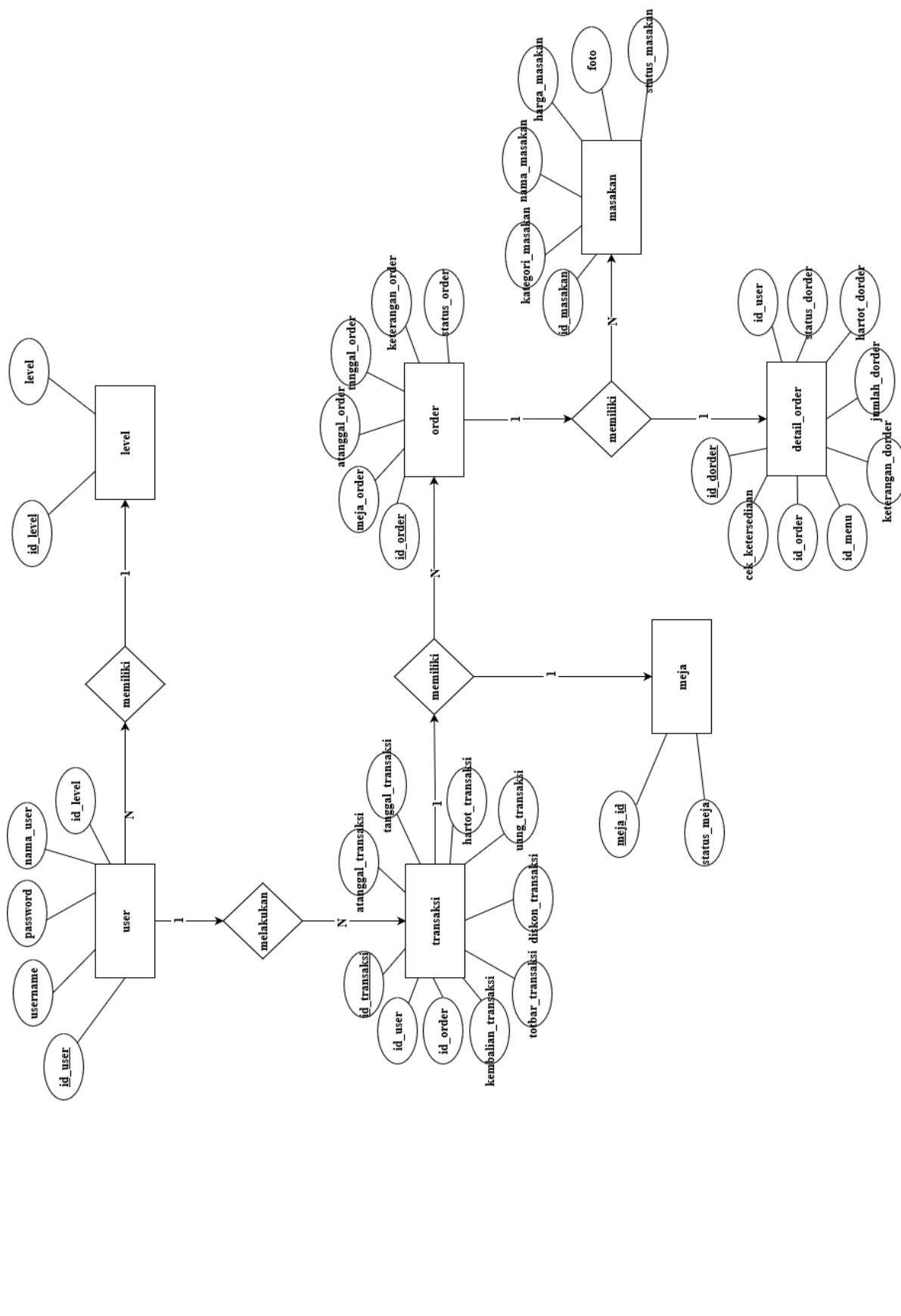
9. Menu Profile



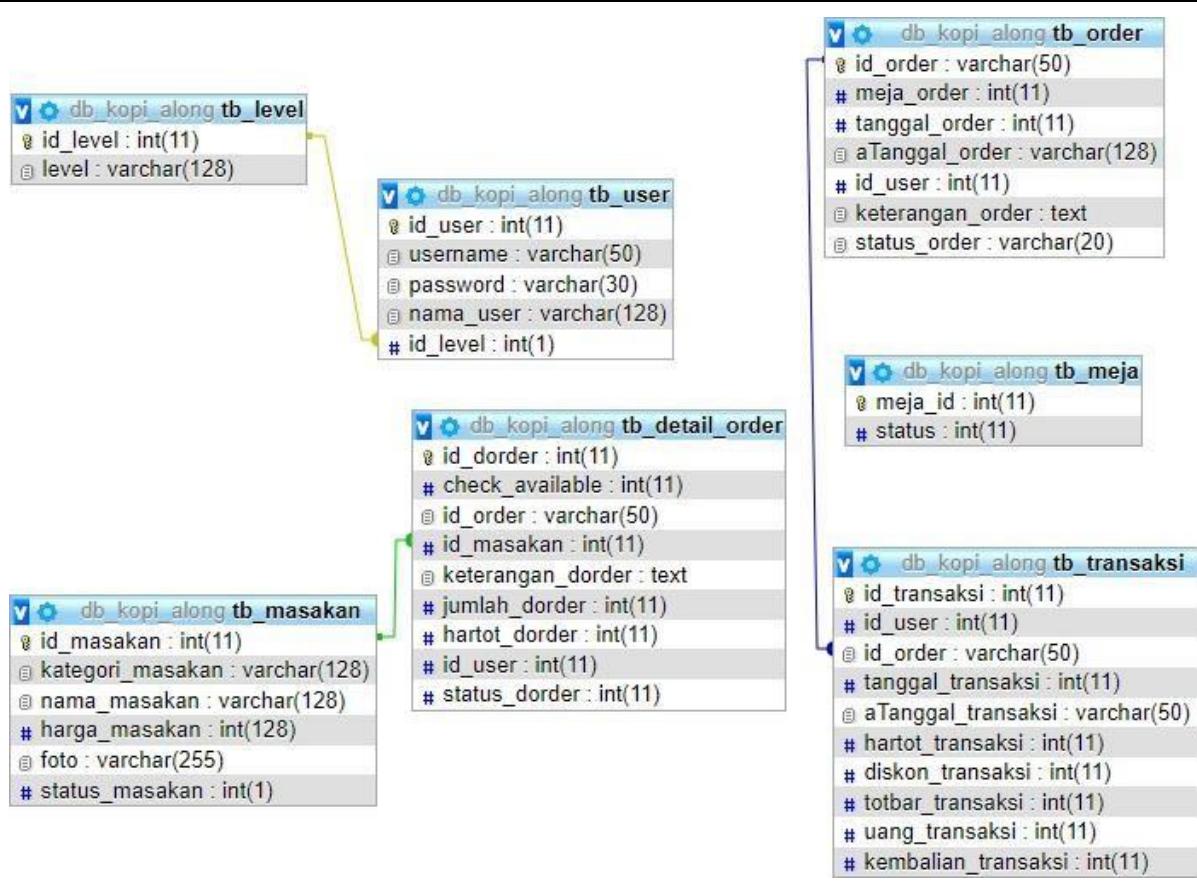
```
SELECT nama_pelanggan, alamat, no_hp_pelanggan  
FROM pelanggan  
WHERE nama_pelanggan = 'Thomas';
```

Judul Aplikasi	
<h2>SISTEM APLIKASI COFFEE SHOP BERBASIS WEB</h2>	
Dikembangkan Oleh	<ul style="list-style-type: none"> • 5220411295 – Kaka Nisfa Kurniawan • 5220411322 – Raihan Muflih • 5220411320 – Tri Jaka Satria • 5220411325 – Shandika Sayyid Ammar Pangesty • 5220411331 – Faiz Ahmad Fauzan • 5220411317 – Alfian Surya Alam
Deskripsi Singkat Instansi, Permasalahan, dan Solusi Aplikasi Yang Akan Dikembangkan [10 poin]	<p>'Kopi Along merupakan cafe yang berlokasi di kalimantan tengah yang di bangun dan dimiliki oleh Muhammad Asary pada tahun 2019'. (Asary, 2019)</p> <p>Permasalah di Kopi Along adalah aktifitas pemesanan, pencatatan pesanan, transaksi pembayaran, pembuatan struk dan laporan transaksi masih dilakukan secara manual.</p> <p>Solusinya kami berikan suatu platform berbasis website yang bisa digunakan untuk melakukan pemesan, mencatat pesanan, mencatat semua transaksi yang ada di kopi along, serta menampilkan struk dan laporan transaksi yang bisa disimpan dengan format pdf.</p>
Pengguna Aplikasi [5 poin]	<ul style="list-style-type: none"> • Owner • Waiters • Kasir • Pelanggan

Desain Entity-Relationship Diagram (ERD) [10 poin]



Desain Relasi Basis Data [10 poin)



Data Definition Language (DDL)

```

CREATE TABLE `tb_level` (
`id_level` int(11) NOT NULL,
`level` varchar(128) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

ALTER TABLE `tb_level`
ADD PRIMARY KEY (`id_level`);

```

```

CREATE TABLE `tb_meja` (
`meja_id` int(11) NOT NULL,
`status` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

ALTER TABLE `tb_meja`
ADD PRIMARY KEY (`meja_id`);

```

```
ALTER TABLE `tb_meja`  
MODIFY `meja_id` int(11) NOT NULL AUTO_INCREMENT,  
AUTO_INCREMENT=11;  
  
CREATE TABLE `tb_user` (  
`id_user` int(11) NOT NULL,  
`username` varchar(50) NOT NULL,  
`password` varchar(30) NOT NULL,  
`nama_user` varchar(128) NOT NULL,  
`id_level` int(1) NOT NULL,  
FOREIGN KEY (id_level) REFERENCES tb_level(id_level)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
  
ALTER TABLE `tb_user`  
ADD PRIMARY KEY (`id_user`),  
ADD KEY `id_level` (`id_level`);  
  
ALTER TABLE `tb_user`  
MODIFY `id_user` int(11) NOT NULL AUTO_INCREMENT,  
AUTO_INCREMENT=12;  
  
CREATE TABLE `tb_order` (  
`id_order` varchar(50) NOT NULL,  
`meja_order` int(11) NOT NULL,  
`tanggal_order` int(11) NOT NULL,  
`aTanggal_order` varchar(128) NOT NULL,  
`id_user` int(11) NOT NULL,  
`keterangan_order` text DEFAULT NULL,  
`status_order` varchar(20) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
  
ALTER TABLE `tb_order`  
ADD PRIMARY KEY (`id_order`);  
  
CREATE TABLE `tb_masakan` (  
`id_masakan` int(11) NOT NULL,  
`kategori_masakan` varchar(128) NOT NULL,
```

```

`nama_masakan` varchar(128) NOT NULL,
`harga_masakan` int(128) NOT NULL,
`foto` varchar(255) NOT NULL,
`status_masakan` int(1) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `tb_masakan`
ADD PRIMARY KEY (`id_masakan`); CREATE TABLE `tb_transaksi` (
`id_transaksi` int(11) NOT NULL,
`id_user` int(11) DEFAULT NULL,
`id_order` varchar(50) NOT NULL,
`tanggal_transaksi` int(11) NOT NULL,
`aTanggal_transaksi` varchar(50) NOT NULL,
`hartot_transaksi` int(11) NOT NULL,
`diskon_transaksi` int(11) NOT NULL,
`totbar_transaksi` int(11) NOT NULL,
`uang_transaksi` int(11) NOT NULL,
`kembalian_transaksi` int(11) NOT NULL,
FOREIGN KEY (id_order) REFERENCES tb_order(id_order)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `tb_transaksi`
ADD PRIMARY KEY (`id_transaksi`);

ALTER TABLE `tb_transaksi`
MODIFY `id_transaksi` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=27;

CREATE TABLE `tb_detail_order` (
`id_dorder` int(11) NOT NULL,
`check_available` int(11) NOT NULL,
`id_order` varchar(50) NOT NULL,
`id_masakan` int(11) NOT NULL,
`keterangan_dorder` text DEFAULT NULL,
`jumlah_dorder` int(11) NOT NULL,

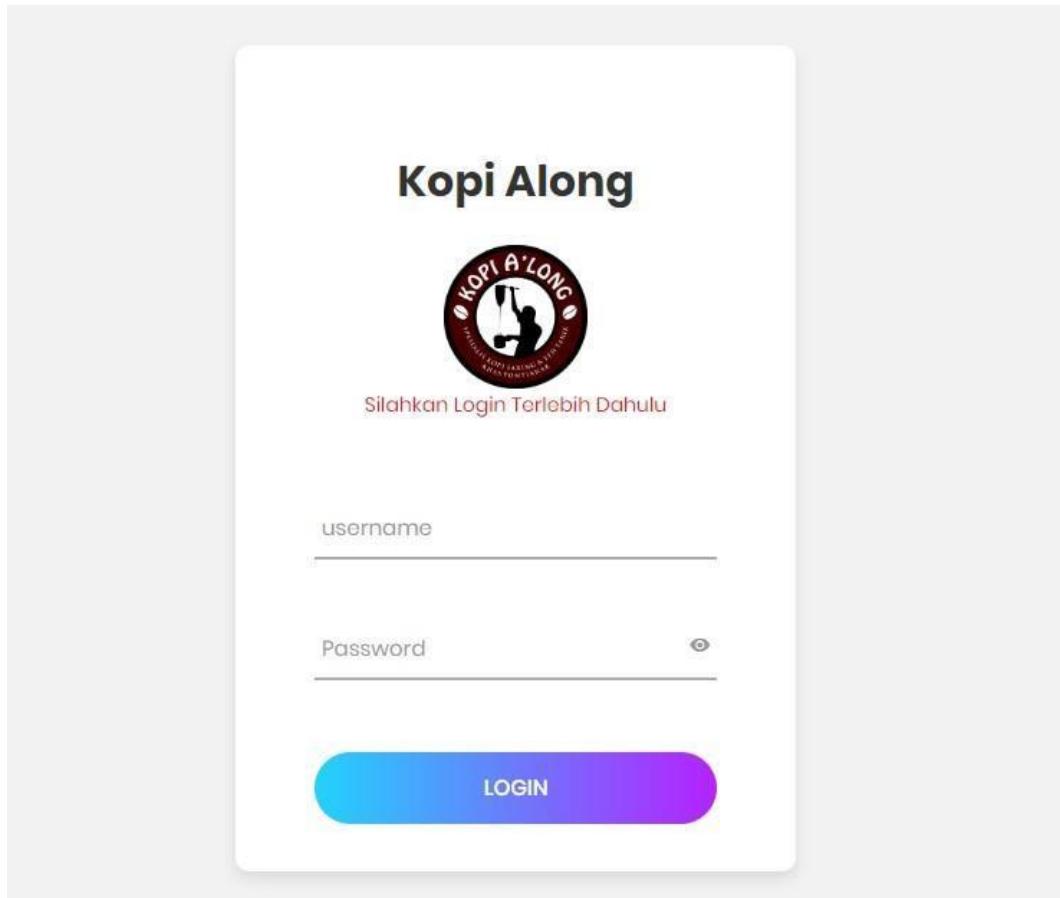
```

```
'hartot_dorder` int(11) NOT NULL,  
'id_user` int(11) NOT NULL,  
'status_dorder` int(11) NOT NULL,  
FOREIGN KEY (id_masakan) REFERENCES tb_masakan(id_masakan)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
ALTER TABLE `tb_detail_order`
```

Desain Interface (Wireframe/Mockup) Beserta Query-nya [point 40]

1. Login



```
Query : $username = $_POST['username'];  
$password = $_POST['password'];  
  
$login = mysqli_query($kon, "SELECT * FROM tb_user WHERE username  
= '$username' AND password= '$password'");  
$cek = mysqli_num_rows($login);
```

2. Halaman Home

The screenshot shows the homepage of a website for 'Kopi Along'. At the top, there's a red header bar with the logo 'Kopi Along' and a 'Home' link. On the right side of the header are 'Login' and 'Pesanan' buttons. Below the header, a welcome message 'Selamat Datang di Kopi Along' is displayed, followed by a sub-instruction 'Silahkan Pesan Makanan Dan Minuman Anda'. A section titled 'Makanan/Food' lists four food items with images, names, prices, and 'Beli' (Buy) buttons:

No	Nama	Keterangan	Harga	Jumlah	Total	Option
1	Ayam Geprek		Rp. 30.000,00	1	Rp. 30.000,00	
2	Ayam Bakar		Rp. 35.000,00	1	Rp. 35.000,00	
3	Ayam Betutu	Tidak Tersedia	Rp. 45.000,00	1	Rp. 45.000,00	
4	Ayam Goreng	Tersedia	Rp. 20.000,00	1	Rp. 20.000,00	

```
Query: $query = "SELECT * FROM tb_masakan WHERE
kategori_masakan='Makanan' ORDER BY id_masakan LIMIT 7";
$sql = mysqli_query($kon, $query);
while($data = mysqli_fetch_array($sql)) :
```

3. Pesanan

The screenshot shows a modal window titled 'Pesanan' (Order). It contains fields for 'Order no.' (set to 'ORD0002') and 'No Meja' (set to '2'). The main area displays a table of items already added to the order:

No	Nama	Keterangan	Harga	Jumlah	Total	Option
1	Ayam Geprek		Rp. 25.000,00	1	Rp. 25.000,00	
2	Es Teh		Rp. 5.000,00	1	Rp. 5.000,00	

Below the table, a total price 'Total Harga : Rp. 30.000,00' is shown. At the bottom right are 'Batal' (Cancel) and 'Proses' (Process) buttons. In the background, there are small preview cards for other menu items: Ayam Bakar (Rp. 35.000,00), Ayam Betutu (Rp. 45.000,00), and Ayam Goreng (Rp. 20.000,00).

```
$query_order = mysqli_query($kon, "SELECT count(id_order) as no_order
FROM tb_order");
$order = mysqli_fetch_assoc($query_order);
$no_order = $order['no_order'] + 1;
$no_meja = mysqli_query($kon, "SELECT * FROM tb_meja WHERE status != 1");
```

```

$list_pesanan = mysqli_query($kon, "SELECT * FROM tb_detail_order WHERE
id_order = 'ORD000$no_order' AND id_user = '$_SESSION[id_user]'");
$nono = 'ORD000' . $no_order;
$query_hartot = mysqli_query($kon, "SELECT sum(hartot_dorder) as hartot
FROM tb_detail_order WHERE id_order = '$nono'");
$hartot = mysqli_fetch_assoc($query_hartot);

```

4. Halaman Pembayaran

The screenshot shows a web application interface for managing coffee shop transactions. At the top, there's a navigation bar with a logo, the text 'I Kopi Along', and links for 'Home', 'Entri Data', 'Entri Transaksi', and 'Laporan'. On the right side of the header, there are links for 'Kasir' and 'Pesanan'. The main content area is titled 'Entri Transaksi'. It displays the following information:

- No order : ORD0002
- No Meja : 3
- Tanggal Pesan : 14-05-2024
- Keterangan :
- Show: 10 entries Search:
- A table showing order details:

No	Nama Pesanan	Jumlah	Harga	Harga Total
1	Ayam Geprek	1	Rp. 25.000,00	Rp. 25.000,00
2	Es Teh	1	Rp. 5.000,00	Rp. 5.000,00
- No Meja dropdown set to 3.
- Member dropdown.
- Total Harga: 30000.
- Diskon (%): 0.
- Total Bayar: 30000.
- Uang input field.
- Kembalian input field.

```

Queiy : $id_order = htmlspecialchars($_POST['id_order']);
$meja = htmlspecialchars($_POST['meja']);
$member = htmlspecialchars($_POST['member']);
$total_harga = htmlspecialchars($_POST['total_harga']);
$diskon = htmlspecialchars($_POST['diskon']);
$total_bayar = htmlspecialchars($_POST['total_bayar']);
$uang = htmlspecialchars($_POST['uang']);
$kembalian = htmlspecialchars($_POST['kembalian']);
$tanggal = time();
$tanggal2 = date('d-m-Y');

$q_detailOrder = mysqli_query($kon, "SELECT * FROM tb_detail_order WHERE
id_order = '$id_order'");

```

5. Halaman Laporan

The screenshot shows a report page titled 'Laporan' for the 'Kopi Along' application. At the top, there is a navigation bar with links for Home, Entri Data, Entri Order, and Laporan. On the right side of the header, there is a user profile for 'Walter' and a link to 'Pesanan'. Below the header, the main content area has a title 'Laporan' and a button labeled 'Print semua laporan'. It includes a search bar and a table with 10 entries. The table columns are: No, No Order, No Meja, Pelanggan, Tanggal Transaksi, Total Bayar, Diskon, Total Bayar (Diskon), and Option. A single row is visible, showing entry 1 with ID ORD0001, table 1, customer 'Yogas Destariansyah', date '02-06-2020 12:02', total 'Rp. 159.000,00', discount '50%', final total 'Rp. 79.500,00', and an 'Option' column with a small blue icon. Below the table, it says 'Showing 1 to 1 of 1 entries' and has 'Previous' and 'Next' buttons.

```
Query: $transaksi = mysqli_query($kon, "SELECT * FROM tb_transaksi ORDER BY id_transaksi DESC");
foreach ($transaksi as $row) :
$user_query = mysqli_query($kon, "SELECT * FROM tb_user WHERE id_user = '$row[id_user]'");
$user = mysqli_fetch_assoc($user_query);
$order_query = mysqli_query($kon, "SELECT * FROM tb_order WHERE id_order = '$row[id_order]'");
$oq = mysqli_fetch_assoc($order_query);
```

6. Struk

The screenshot shows a receipt page for the 'Kopi Along' application. At the top, there is a logo and the text 'Kopi Along'. Below the logo, the receipt details are listed: 'TANGGAL : 02-06-2020 12:03', 'Member : Yogas Destariansyah', and 'NO ORDER : ORD0001'. The receipt then lists the items purchased: 'Ayam Geprek 6 Rp. 25.000,00 Rp. 150.000,00' and 'Jus Mangga 1 Rp. 9.000,00 Rp. 9.000,00'. Following this is a summary table:

Subtotal	Rp. 159.000,00
Diskon	50 %
Total	Rp. 79.500,00
Tunai	Rp. 100.000,00
Kembalian	Rp. 20.500,00

At the bottom of the receipt, there is a message: 'Terima Kasih Atas Kunjungan Anda'.

```
Queiy : $id = $_GET['id_order'];
$q_struk = mysqli_query($kon, "SELECT * FROM tb_transaksi WHERE id_order =
'$id'");
$struk = mysqli_fetch_assoc($q_struk);
$q_mem = mysqli_query($kon, "SELECT * FROM tb_user WHERE id_user =
'$struk[id_user]'");
$mem = mysqli_fetch_assoc($q_mem);
$detail_order = mysqli_query($kon, "SELECT * FROM tb_detail_order WHERE
id_order = '$id'");
$q_hartot = mysqli_query($kon, "SELECT sum(hartot_dorder) as hartot FROM
tb_detail_order WHERE id_order = '$id'");
$hartot = mysqli_fetch_assoc($q_hartot);
```

Judul Aplikasi	
SISTEM INFORMASI MANAJEMEN OPERASIONAL RUMAH MAKAN NASI BALAP CUCUN	
Dikembangkan Oleh	
<ul style="list-style-type: none"> • 5220411300 – Nurwahdaniah Maulida • 5220411301 – Dwita Dhara Nanda • 5220411310 – Amelia Gusviani • 5220411330 – Andi Muhammad Bintang • 5220411333 – Anis'syah Mahdarani 	
Deskripsi Singkat Instansi, Permasalahan, dan Solusi Aplikasi Yang Akan Dikembangkan [10 poin]	
<p>Rumah Makan Nasi Balap Cucun yang berlokasi di Jalan Kaliurang KM. 5 Gang Srinindito No. 14, Sleman, merupakan salah satu rumah makan di DI Yogyakarta. Rumah makan ini menjual berbagai macam jenis menu masakan yang enak dan lezat. Namun, seiring dengan berkembangnya bisnis, rumah makan ini dihadapkan pada beberapa tantangan dalam mengelola operasionalnya secara efisien, terutama terkait dengan:</p> <ul style="list-style-type: none"> • Proses pengelolaan menu, penjualan, dan pelanggan masih dilakukan secara manual, yang tidak efisien dan rentan terhadap kesalahan. • Pemesanan pencatatan pembelian menu makanan oleh pelanggan dilakukan secara manual menggunakan nota oleh pelayan sehingga proses tersebut dirasa rentan terhadap kekeliruan dan menimbulkan keraguan oleh pelanggan terhadap keakuratan transaksi. <p>Solusi aplikasi yang akan dikembangkan:</p> <ul style="list-style-type: none"> • Sistem Manajemen Data Menyimpan dan mengelola data menu, pemesanan, dan pelanggan secara terpusat dan terstruktur. • Sistem Manajemen Pemesanan Mengotomatisasi proses pemesanan, sehingga meminimalisir human error dan mempercepat proses transaksi serta menyediakan data transaksi yang akurat dan real-time. <p>Dengan mengimplementasikan sistem basis data terintegrasi ini, Rumah Makan Nasi Balap Cucun dapat mencapai pengelolaan operasional yang lebih efisien, meningkatkan kepuasan pelanggan, dan pada akhirnya mendorong pertumbuhan profitabilitas bisnis. Sistem ini akan menjadi fondasi yang kuat untuk mendukung perkembangan dan kesuksesan Nasi Balap Cucun.</p>	

Pengguna Aplikasi [5 poin]

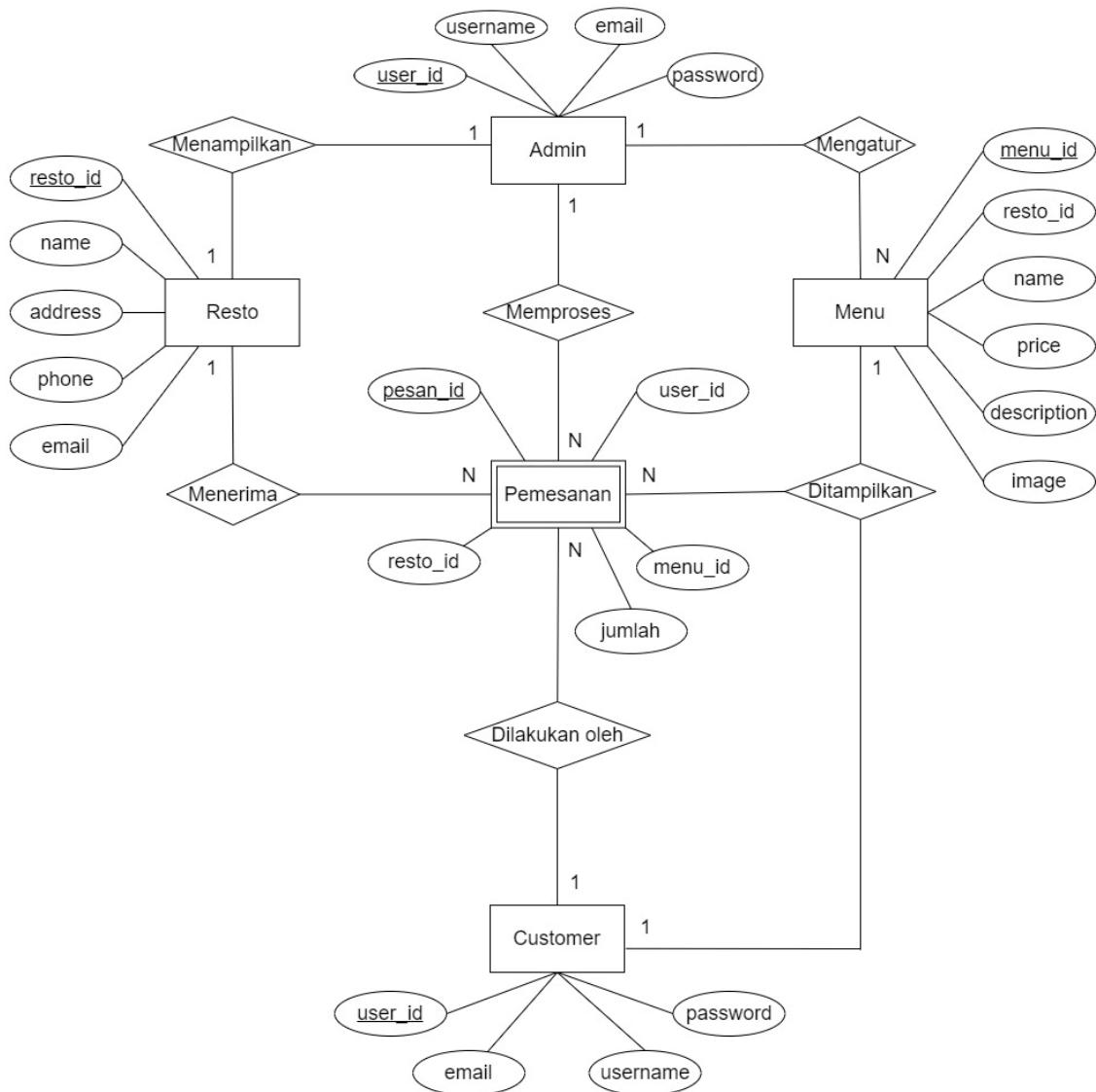
1. Admin:

- Sebagai pemegang otoritas tertinggi dalam sistem, admin bertanggung jawab atas manajemen keseluruhan operasi.
- Admin memiliki akses penuh terhadap fitur dan fungsi sistem, termasuk pengaturan menu, inventaris, dan harga.
- Admin menggunakan sistem untuk mengambil pesanan dari pelanggan dan memprosesnya.

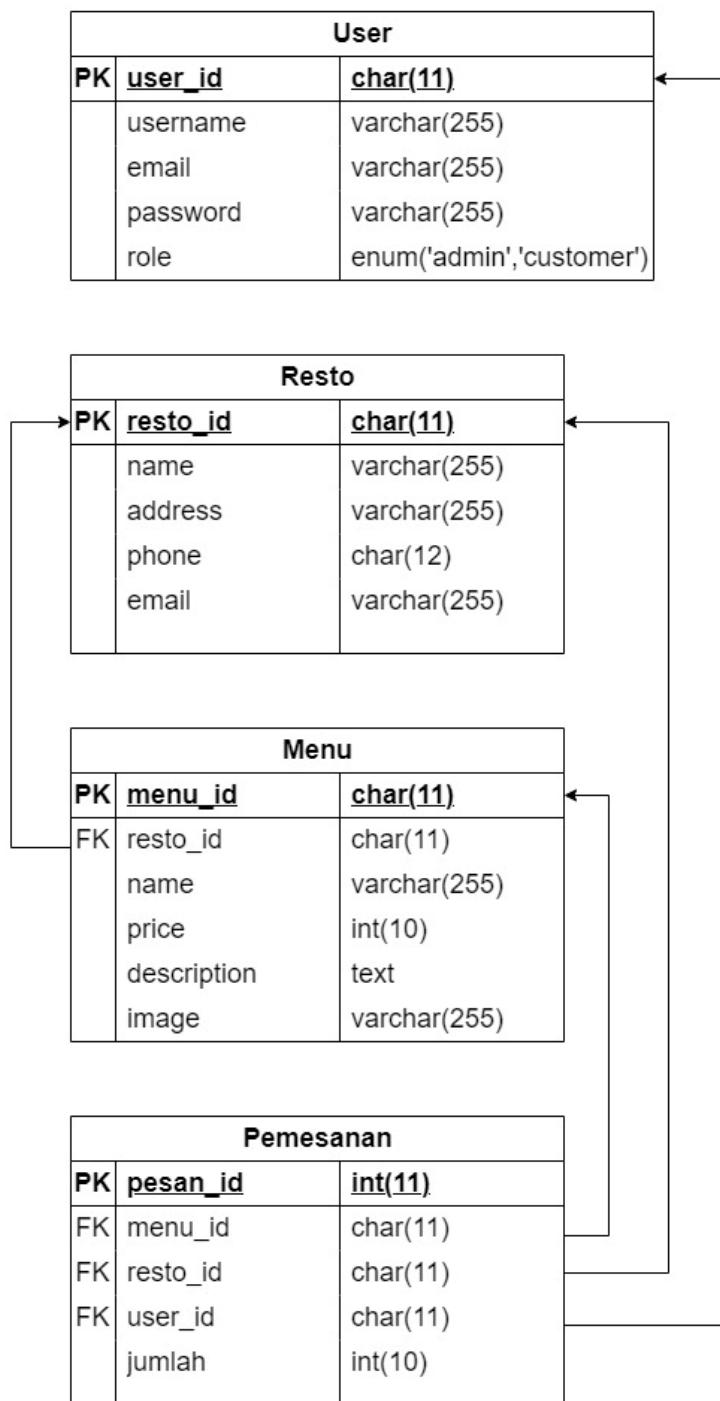
2. Customer:

- Customer adalah pengguna eksternal yang menggunakan layanan rumah makan. Customer dapat menggunakan sistem untuk memesan makanan dan minuman, melihat menu dan harga.

Desain Entity-Relationship Diagram (ERD) [10 poin]



Desain Relasi Basis Data [10 poin]



Data Definition Language (DDL)

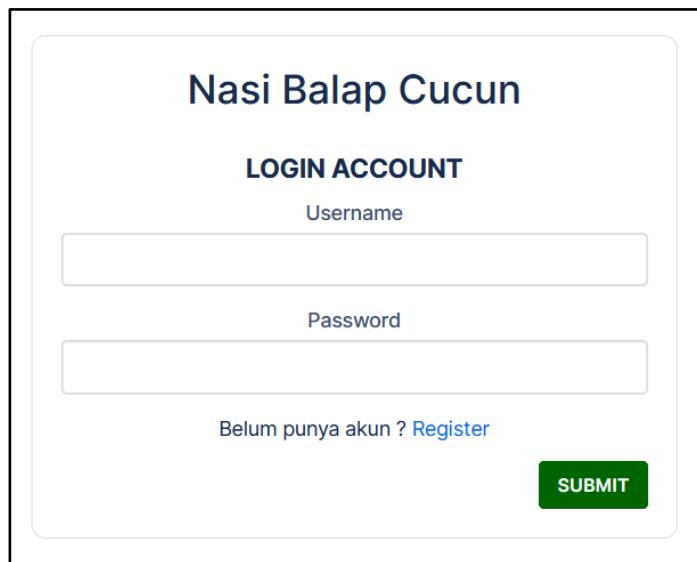
```

CREATE TABLE User (
    user_id CHAR(11) PRIMARY KEY,
    username VARCHAR(255) NOT NULL,
    email VARCHAR(255) NOT NULL,
)
    
```

```
password VARCHAR(255) NOT NULL,
role enum('admin','customer') NOT NULL
);
CREATE TABLE Resto (
    resto_id CHAR(11) PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    address VARCHAR(255) NOT NULL,
    phone CHAR(12) NOT NULL,
    email VARCHAR(255) NOT NULL
);
CREATE TABLE Menu (
    menu_id CHAR(11) PRIMARY KEY,
    resto_id CHAR(11),
    name VARCHAR(255) NOT NULL,
    price INT(10) NOT NULL,
    description text NOT NULL,
    image varchar(255) DEFAULT NULL,
    FOREIGN KEY (resto_id) REFERENCES resto(resto_id)
);
CREATE TABLE Pemesanan (
    pesan_id INT(11) AUTO_INCREMENT PRIMARY KEY,
    menu_id CHAR(11),
    resto_id CHAR(11),
    user_id CHAR(11),
    jumlah INT(10) NOT NULL,
    FOREIGN KEY (menu_id) REFERENCES menu(menu_id),
    FOREIGN KEY (resto_id) REFERENCES resto(resto_id),
    FOREIGN KEY (user_id) REFERENCES user(user_id)
);
```

Desain Interface (Wireframe/Mockup) Beserta Query-nya [poin 40]

1. Login

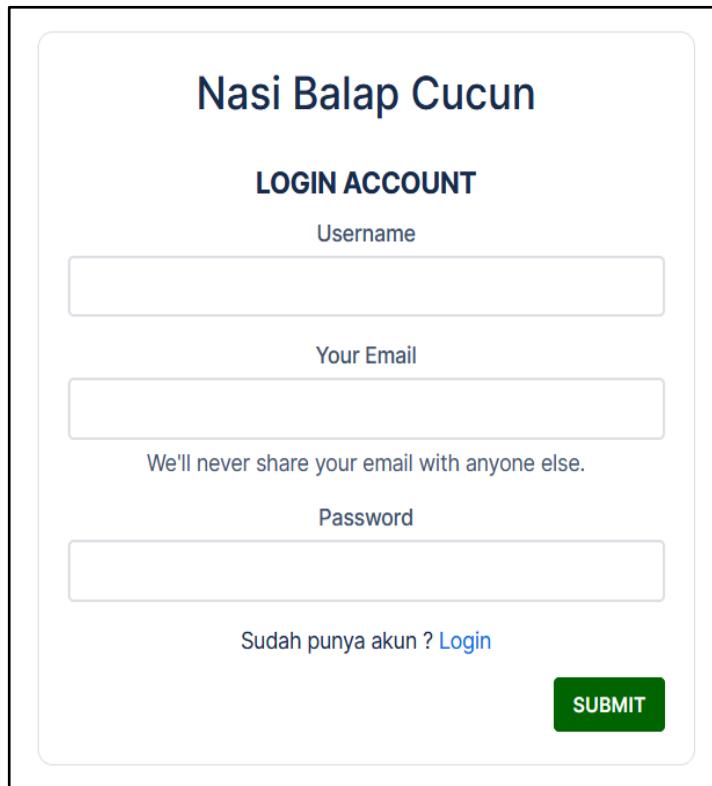


The login form is titled "Nasi Balap Cucun" at the top center. Below it is a section labeled "LOGIN ACCOUNT". It contains two input fields: "Username" and "Password", each with a placeholder text inside. Below the password field is a link "Belum punya akun ? Register". At the bottom right is a green "SUBMIT" button.

Query :

```
SELECT * FROM user  
WHERE username = 'username' AND password = 'password';
```

2. Register



The register form is titled "Nasi Balap Cucun" at the top center. Below it is a section labeled "LOGIN ACCOUNT". It contains three input fields: "Username", "Your Email", and "Password", each with a placeholder text inside. Between the first two fields is a note "We'll never share your email with anyone else.". Below the password field is a link "Sudah punya akun ? Login". At the bottom right is a green "SUBMIT" button.

Query :

```
INSERT INTO user (username, email, password)  
VALUES ('username','email','password');
```

3. Menampilkan Tabel Resto

Name	Address	Phone	Email	Action
Nasi Balap Cucun	Jl.Kaliurang Yogyakarta	0812-1244-6677	nasibalapcucun@gmail.com	<button>EDIT</button> <button>DELETE</button>

Query :

```
SELECT * FROM resto;
```

4. Menambahkan Resto

Create New Resto

Name Resto

Address

Phone

Email

CREATE

Query :

```
INSERT INTO resto (resto_id, name, address, phone, email)
VALUES ('resto_id','name','address','phone','email');
```

5. Edit Resto

A

Edit Resto

Name Resto
Nasi Balap Cucun

Address
Jl. Karangjati

Phone
0872-5376-5464

Email
nasibalapcucun2@gmail.com

SAVE

Query :

```
UPDATE resto SET
name='name',address='address',phone='phone',email='email'
WHERE resto_id='resto_id';
```

6. Hapus Resto

A

CREATE

Name	Address	Phone	Email	Action
Nasi Balap Cucun	Jl.Kalurang Yogyakarta	0812-1244-6677	nasibalapcucun@gmail.com	EDIT DELETE

Query :

```
DELETE FROM resto WHERE resto_id='resto_id';
```

7. Menampilkan Tabel Menu

Name	Price	Resto	Description	Image	Aksi
Nasi Balap Cucun Paket A	25000	Nasi Balap Cucun	Nasi balap cucun paket A merupakan hidangan utama yang terdiri dari nasi putih yang disajikan dengan berbagai macam lauk pauk seperti ayam suwir, tempe, tahu, telur kecap, dan sambal		<button>EDIT</button> <button>DELETE</button>
Nasi Balap Cucun Paket B	20000	Nasi Balap Cucun	Nasi balap cucun paket B merupakan hidangan utama yang terdiri dari nasi putih yang disajikan dengan berbagai macam lauk pauk seperti sayur kacang, tempe, telur dadar, dan sambal		<button>EDIT</button> <button>DELETE</button>
Nasi Balap Cucun Paket C	27000	Nasi Balap Cucun	Nasi balap cucun paket C merupakan hidangan utama yang terdiri dari nasi putih yang disajikan dengan berbagai macam lauk pauk seperti ayam suwir, tahu, telur dadar, dan sambal		<button>EDIT</button> <button>DELETE</button>
Nasi Balap Cucun Paket D	23000	Nasi Balap Cucun	Nasi balap cucun paket A merupakan hidangan utama yang terdiri dari nasi putih yang disajikan dengan berbagai macam lauk pauk seperti sayur kacang, tempe, tahu, telur kecap, dan sambal		<button>EDIT</button> <button>DELETE</button>

Query :

```
SELECT * FROM menu;
```

8. Menambahkan Menu

Create New Menu

Name Menu

Resto

Price

Description

Image No file selected.

CREATE

Query :

```
INSERT INTO menu(menu_id, name, resto_id, price, description, image)
VALUES ('menu_id','name','price','description','image');
```

9. Edit Menu

Edit Menu

Name Menu
Nasi Balap Cucun Paket A

Resto
Resto

Price
25000

Description
Nasi balap cucun paket A merupakan hidangan utama yang terdiri dari nasi putih yang disajikan dengan berbagai macam lauk pauk seperti ayam suwir, tempe, tahu, telur kecap, dan sambal

Image Browse... No file selected.

SAVE

Query :

UPDATE menu SET

resto_id='resto_id',name='name',price='price',description='description',image='image'

WHERE menu_id='menu_id';

10. Hapus Menu

Name	Harga	Resto	Description	Image	Aksi
Nasi Balap Cucun Paket A	25000	Nasi Balap Cucun	Nasi balap cucun paket A merupakan hidangan utama yang terdiri dari nasi putih yang disajikan dengan berbagai macam lauk pauk seperti ayam suwir, tempe, tahu, telur kecap, dan sambal		<button>EDIT</button> <button>DELETE</button>
Nasi Balap Cucun Paket B	20000	Nasi Balap Cucun	Nasi balap cucun paket B merupakan hidangan utama yang terdiri dari nasi putih yang disajikan dengan berbagai macam lauk pauk seperti sayur kacang, tempe, telur dadar, dan sambal		<button>EDIT</button> <button>DELETE</button>
Nasi Balap Cucun Paket C	27000	Nasi Balap Cucun	Nasi balap cucun paket C merupakan hidangan utama yang terdiri dari nasi putih yang disajikan dengan berbagai macam lauk pauk seperti ayam suwir, tahu, telur dadar, dan sambal		<button>EDIT</button> <button>DELETE</button>
Nasi Balap Cucun Paket D	23000	Nasi Balap Cucun	Nasi balap cucun paket A merupakan hidangan utama yang terdiri dari nasi putih yang disajikan dengan berbagai macam lauk pauk seperti sayur kacang, tempe, tahu, telur kecap, dan sambal		<button>EDIT</button> <button>DELETE</button>

Query :

DELETE FROM menu WHERE menu_id='menu_id';

11. Menampilkan Tabel Pemesanan

The screenshot shows a table titled 'Pemesanan' with the following data:

Name	Menu	Jumlah	Price	Resto	Total	Aksi
amelia	Nasi Balap Cucun Paket C	10	Rp. 27000	Nasi Balap Cucun	Rp. 270000	PROSES PESANAN
anissyah	Nasi Balap Cucun Paket A	2	Rp. 25000	Nasi Balap Cucun	Rp. 50000	PROSES PESANAN
dania	Nasi Balap Cucun Paket B	2	Rp. 20000	Nasi Balap Cucun	Rp. 40000	PROSES PESANAN
dhara	Nasi Balap Cucun Paket C	2	Rp. 27000	Nasi Balap Cucun	Rp. 54000	PROSES PESANAN
bintang	Nasi Balap Cucun Paket D	2	Rp. 23000	Nasi Balap Cucun	Rp. 46000	PROSES PESANAN

Query :

```
SELECT * FROM pemesanan;
```

12. Pemesanan

The screenshot shows a form for placing an order with the following fields:

- Nama Customer: amelia
- Email Customer: amelia@gmail.com
- Menu: Nasi Balap Cucun Paket C
- Description: Nasi balap cucun paket C merupakan hidangan utama yang terdiri dari nasi putih yang disajikan dengan berbagai macam lauk pauk seperti ayam suwir, tahu, telur dadar, dan sambal
- Price: 27000
- Resto: Nasi Balap Cucun
- Jumlah: (empty input field)
- Pesan: (green button)

Query :

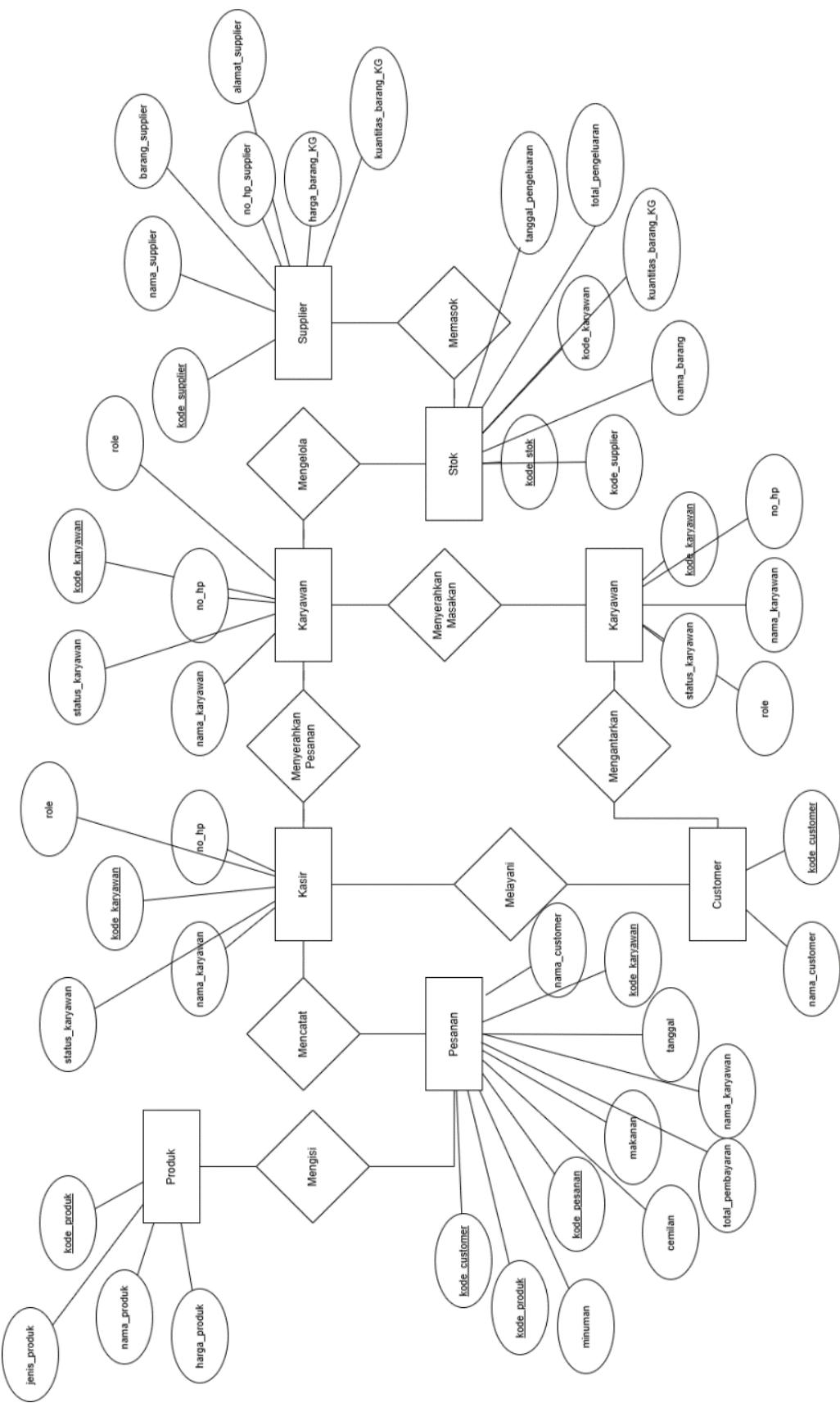
```
SELECT pemesanan.*,
       resto.name AS resto_name,
       menu.name AS menu_name,
       user.name AS user_name
  FROM pemesanan
 LEFT JOIN
       resto ON pemesanan.resto_id = resto.resto_id
LEFT JOIN
```

```
menu ON pemesanan.menu_id = menu.menu_id
LEFT JOIN
    user ON pemesanan.user_id = user.user_id;

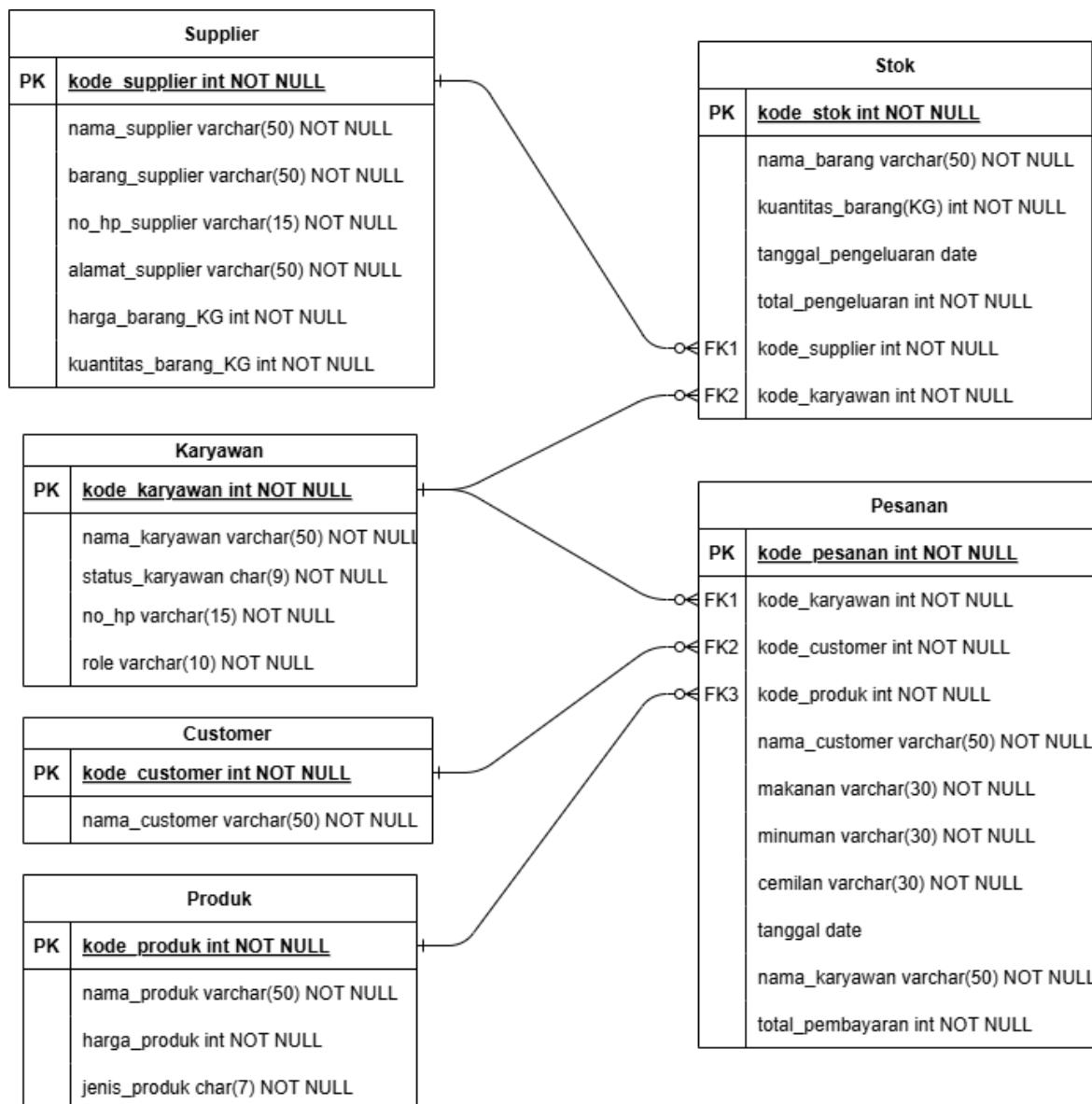
INSERT INTO pemesanan(pesan_id, menu_id, resto_id, user_id, jumlah)
VALUES ('pesan_id','menu_id','resto_id','user_id','jumlah');
```

Judul Aplikasi	
HANA CHICKS 100% RESTAURANT	
Dikembangkan Oleh	
<ul style="list-style-type: none"> • 5220411293 – Syaifulah Yusuf Fadililah • 5220411299 – Latif Hilmie Romadhoni • 5220411303 – Alvero Galen Gaviota • 5220411305 – Adam Hrlambang • 5220411316 – Dimas Sabda Putra 	
Deskripsi Singkat Instansi, Permasalahan, dan Solusi Aplikasi Yang Akan Dikembangkan [10 poin]	
<p>Hana Chicks adalah instansi yang bergerak pada bidang <i>Food & Beverages</i> yang berada di lingkungan kampus Universitas Teknologi Yogyakarta. Hana Chicks menawarkan makanan cepat saji berbasis ayam dengan aneka minuman serta menyediakan fasilitas tempat yang luas.</p> <p>Pembukuan keuangan dari restoran ini yang perlu di komputerisasikan untuk mempermudah jalannya operasi restoran pada bagian administrasi. Karena pembukuan yang dilakukan Hana Chicks masih menggunakan buku. Sehingga dapat meminimalisir kesalahan yang terjadi pada pembukuan.</p> <p>Aplikasi ini dirancang untuk mempermudah proses pembukuan karena dapat dilakukan secara <i>real time</i> setelah terjadinya transaksi yang dilakukan oleh pelanggan. Dan Apabila pada masa yang akan datang diperlukan data yang sebelumnya maka Aplikasi ini dapat digunakan sebagai solusi karena aplikasi tersebut berisi data sebelumnya yang sudah terkomputerisasi.</p>	
Pengguna Aplikasi [5 poin]	
Kasir Dan Karyawan Hana Chicks.	

Desain Entity-Relationship Diagram (ERD) [10 poin]



Desain Relasi Basis Data [10 poin]



Query Data Definition Language (DDL) [25 poin]

```
CREATE TABLE pesanan (
    kode_pesanan int(11) NOT NULL,
    kode_karyawan int(11) NOT NULL,
    kode_customer int(11) NOT NULL,
    kode_produk int(11) NOT NULL,
    nama_customer varchar(50) NOT NULL,
    makanan varchar(30) NOT NULL,
    minuman varchar(30) NOT NULL,
    cemilan varchar(30) NOT NULL,
    tanggal date,
```

```

        total_pembayaran int NOT NULL,
        FOREIGN KEY (kode_karyawan) REFERENCES
karyawan(kode_karyawan),
        FOREIGN KEY (kode_customer) REFERENCES customer(kode_customer),
        FOREIGN KEY (kode_produk) REFERENCES produk(kode_produk)
);

CREATE TABLE customer (
    kode_customer int(11) NOT NULL,
    nama_customer varchar(50) NOT NULL
);

CREATE TABLE karyawan (
    kode_karyawan int(11) NOT NULL PRIMARY KEY,
    nama_karyawan varchar(50) NOT NULL,
    status_karyawan char(9) NOT NULL,
    no_hp varchar(15) NOT NULL,
    role varchar(10) NOT NULL
);

CREATE TABLE produk (
    kode_produk int(11) NOT NULL PRIMARY KEY,
    nama_produk varchar(50) NOT NULL,
    harga_produk int(11) NOT NULL,
    jenis_produk char(7) NOT NULL
);

CREATE TABLE stok (
    kode_stok int(11) NOT NULL,
    nama_barang varchar(50) NOT NULL,
    kuantitas_barang int(11) NOT NULL,
    kode_supplier int(11) NOT NULL,
    kode_karyawan int(11) NOT NULL,
    tanggal_pengeluaran date,
    total_pengeluaran int NOT NULL,
    FOREIGN KEY (kode_supplier) REFERENCES supplier(kode_supplier),
    FOREIGN KEY (kode_karyawan) REFERENCES
karyawan(kode_karyawan)
);

```

```
);
```

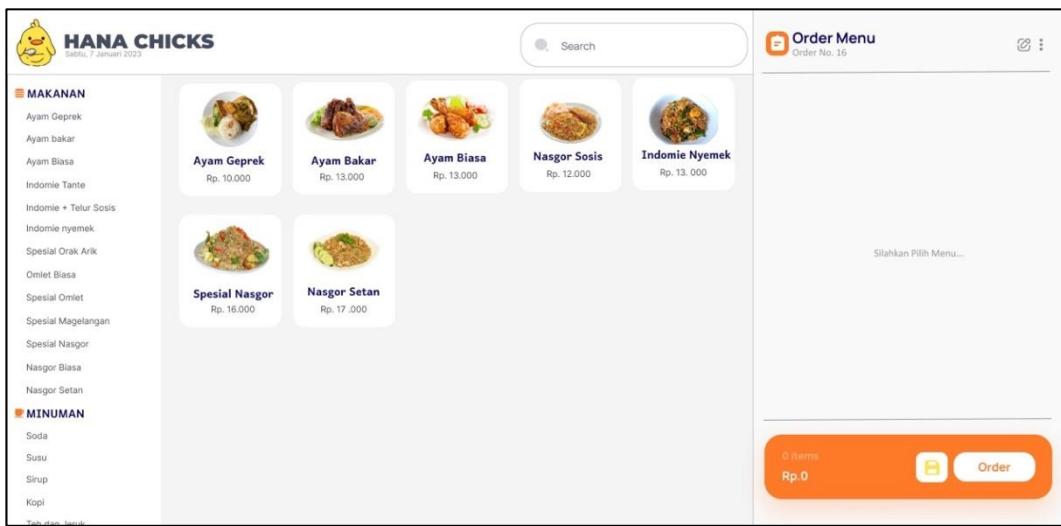
```
CREATE TABLE supplier (
    kode_supplier int(11) NOT NULL,
    nama_supplier varchar(50) NOT NULL,
    barang_supplier varchar(50) NOT NULL,
    no_hp_supplier varchar(15) NOT NULL,
    alamat_supplier varchar(50) NOT NULL,
    harga_barang_per_KG int(11) NOT NULL,
    kuantitas_barang_KG int(11) NOT NULL
);
```

```
CREATE OR REPLACE VIEW view_detail_order AS
SELECT kode_pesanan, nama_customer, makanan, minuman, cemilan, tanggal,
total_pembayaran
FROM pesanan;
```

```
CREATE OR REPLACE VIEW view_cashflow AS
SELECT kode_pesanan, nama_customer, kode_supplier, nama_supplier, tanggal
AS tanggal_pemasukan, total_pembayaran AS pemasukan, tanggal_pengeluaran,
total_pengeluaran
FROM pesanan JOIN stok ON pesanan.kode_karyawan = stok.kode_karyawan
JOIN supplier ON stok.kode_supplier = supplier.kode_supplier;
```

Desain Interface (Wireframe/Mockup) Beserta Query-nya [poin 40]

User Interface awal sebelum memasukan data pemasukan dan pengeluaran



User Interface untuk memasukan data pemasukan

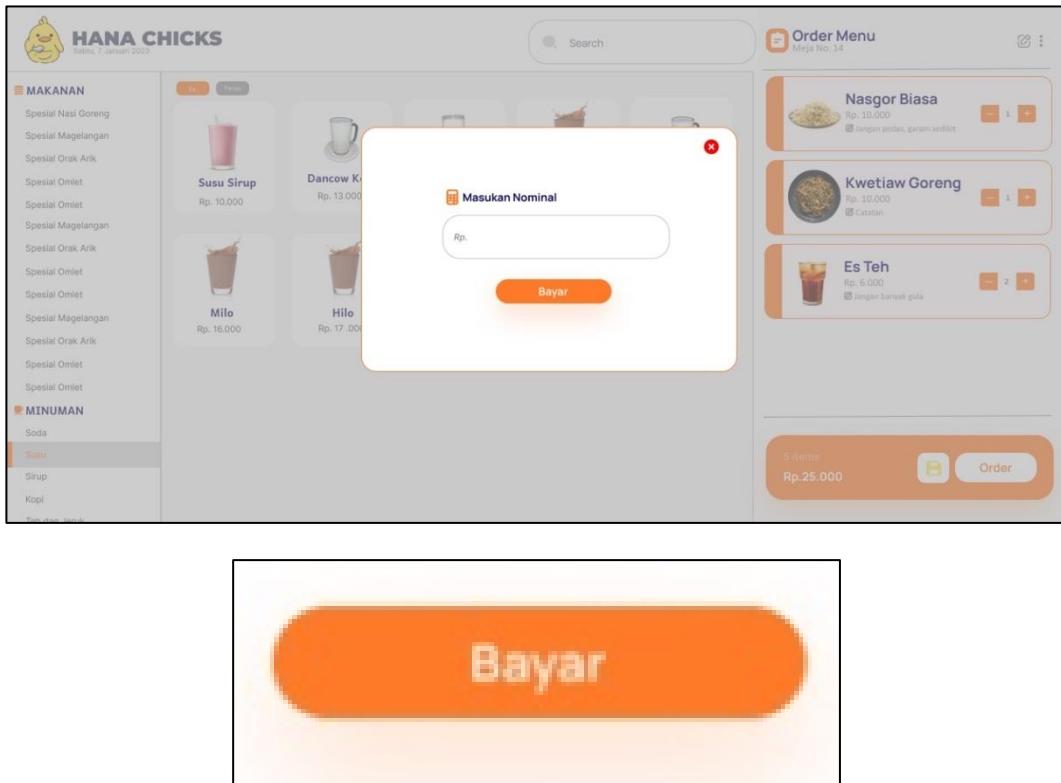
The screenshot shows the HANA CHICKS mobile ordering application interface. At the top left is the restaurant logo and name "HANA CHICKS" with the date "Sabtu, 7 Januari 2023". A search bar is at the top right. On the left, there's a sidebar menu with sections for MAKANAN and MINUMAN. Under MAKANAN, several items like Spesial Nasi Goreng, Spesial Magelangan, etc., are listed. Under MINUMAN, Soda, Susu (highlighted in orange), Sirup, and Kopi are listed. The main area displays a grid of food and drink items with their names, prices, and descriptions. To the right, the "Order Menu" section shows three items added to the cart: Nasgor Biasa (Rp. 10.000), Kwetiauw Goreng (Rp. 10.000), and Es Teh (Rp. 6.000). The total is 5 items for Rp. 25.000. An "Order" button is at the bottom.

Item	Description	Price
Nasgor Biasa	Rp. 10.000 jangan pedas, garam sedikit	1
Kwetiauw Goreng	Rp. 10.000 Catatan	1
Es Teh	Rp. 6.000 jangan banyak gula	2



Tombol Order: INSERT INTO pesanan (makanan, minuman, tanggal) VALUES ('nama makanan', 'nama minuman', 'tanggal transaksi')

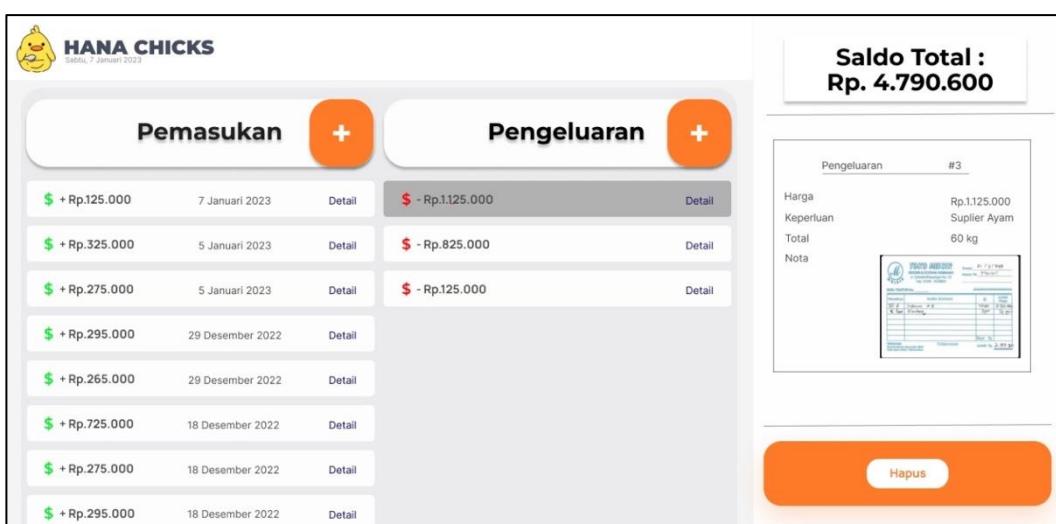
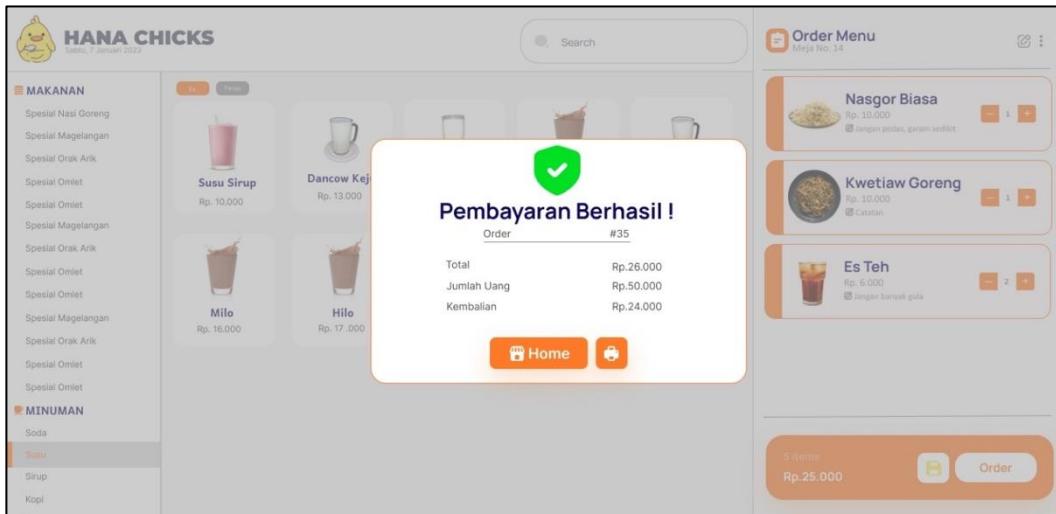
User Interface untuk memasukkan nominal dari total pembayaran.



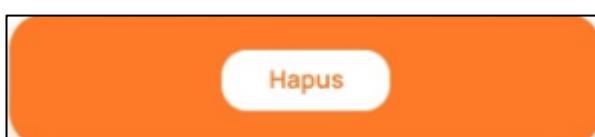
Bayar

Tombol Bayar: INSERT INTO pesanan(total_pembayaran) VALUES('nominal')

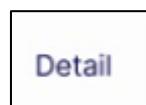
User Interface pop-up status pembayaran.



User Interface riwayat pemasukan dan pengeluaran.



Tombol Hapus: DELETE FROM pesanan WHERE kode_pesanan = 'kode pesanan'



Tombol Detail: SELECT * FROM [view_detail_order]

The screenshot displays a mobile application interface for a restaurant named "HANA CHICKS".

Cash Flow History:

- 7 Januari 2023: 3 Item, Rp. 36.000 (16.43)
- 5 Januari 2023: 3 Item, Rp. 36.000 (16.43)
- 29 Desember 2022: 3 Item, Rp. 36.000 (16.43)
- 18 Desember 2022: 3 Item, Rp. 36.000 (16.43)

Detail Order (Order No.16):

Item	Quantity	Description
Nasgor Biasa	1	Rp. 10.000 Jangan pedas, garam sedikit
Kwetiaw Goreng	1	Rp. 10.000 Catatan
Es Teh	2	Rp. 6.000 Jangan banyak gula

Hapus button is located at the bottom right of the detail order section.

Tombol Hapus: DELETE FROM view_detail_order WHERE kode_pesanan = 'kode pesanan'



Tombol Dolar: SELECT * FROM [view_cashflow]

User Interface memasukan data pengeluaran

HANA CHICKS
Saldo, 7 Januari 2023

Pengeluaran

Bulan: Select Tahun: Select Tanggal: ***

Nama Supplier: Lengkap ya.... Barang Suplai: Jangan disingkat... Jumlah: ...kg

Deskripsi / Tambahan: Tambahan

Simpan

Tambah Foto Barang, Tarik atau Upload

Saldo Total :
Rp. 4.790.600

Pengeluaran	#
Harga	...
Keperluan	...
Total	...
Nota	

Tambah

Tambah

Tombol Tambah: `INSERT INTO stok (nama_barang, kuantitas_barang) VALUES ('barang suplai', 'jumlah barang')`

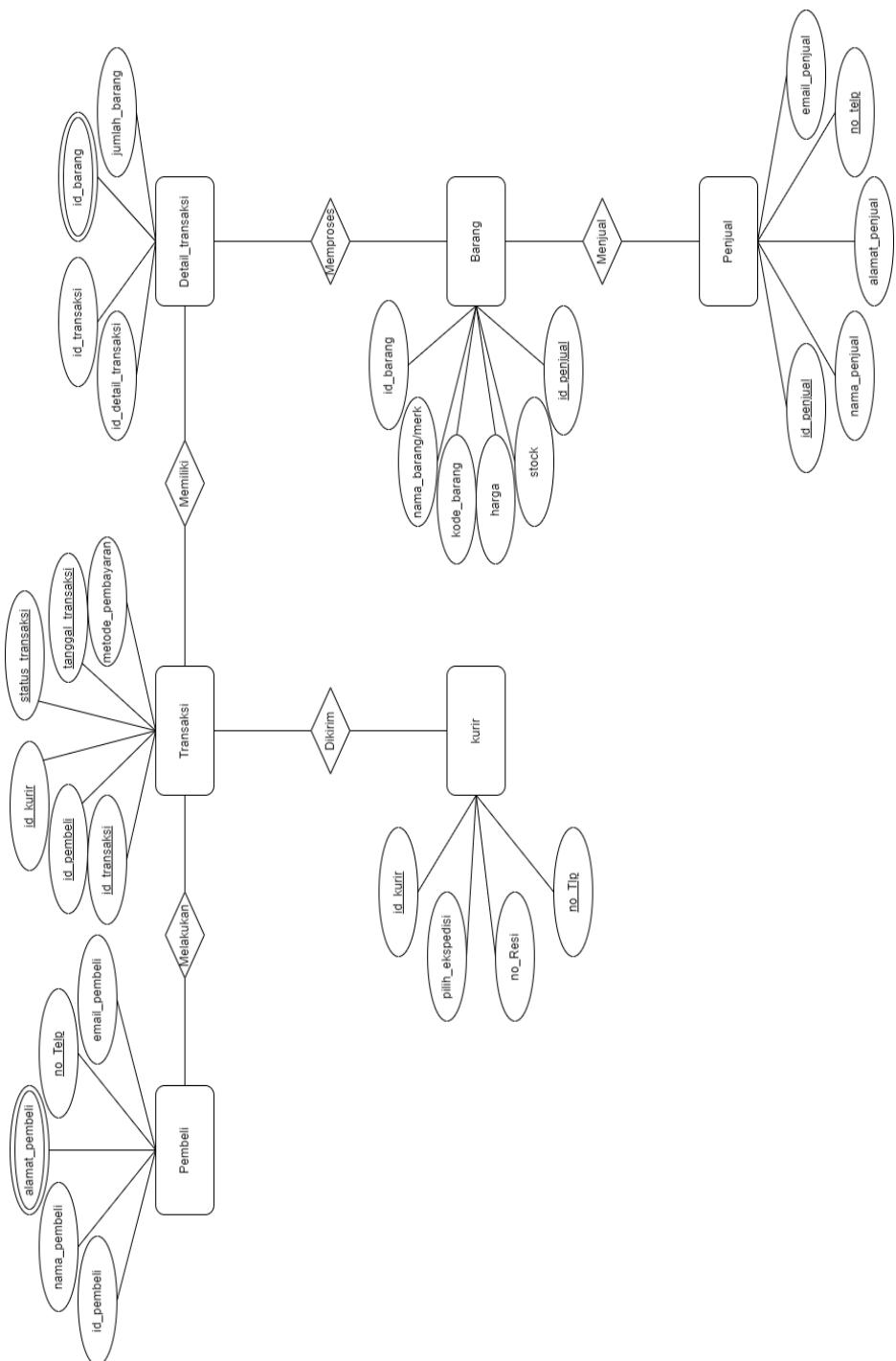
Judul Aplikasi	WINGS SHOES (WS)	
Dikembangkan Oleh		
<ul style="list-style-type: none"> • 5220411406 – Mohamad Styvani Hendi S* • 5220411408 – Fajar Jati Satata • 5220411409 – Ilham Wisudawan • 5220411413 – Ade Prima Krifitsky • 5220411421 – Abdul Ambya • 5220411384 – Muhammad Luqmanul hakim 		
Deskripsi Singkat Instansi, Permasalahan, dan Solusi Aplikasi Yang Akan Dikembangkan [10 poin]		
<p>Instansi</p> <p>Toko sepatu "Wings Shoes"</p> <p>Permasalahan</p> <p>Saat ini, pelanggan kami kesulitan menavigasi katalog sepatu kami secara efisien dan melakukan transaksi dengan lancar. Selain itu, kami ingin meningkatkan kesadaran merek kami di pasar yang semakin kompetitif.</p> <p>Solusi Aplikasi</p> <p>Kami akan mengembangkan aplikasi mobile untuk "Wings Shoes" yang akan memberikan pengalaman berbelanja yang lebih nyaman dan menyenangkan bagi pelanggan kami. Aplikasi ini akan memungkinkan pengguna untuk dengan mudah menjelajahi berbagai koleksi sepatu kami, melakukan pencarian produk, menambahkan barang ke keranjang belanja, dan menyelesaikan pembayaran dengan aman dan mudah. Selain itu, kami akan</p>		

menyertakan fitur pemberitahuan tentang penjualan, promosi, dan pembaruan katalog produk untuk meningkatkan keterlibatan pengguna.

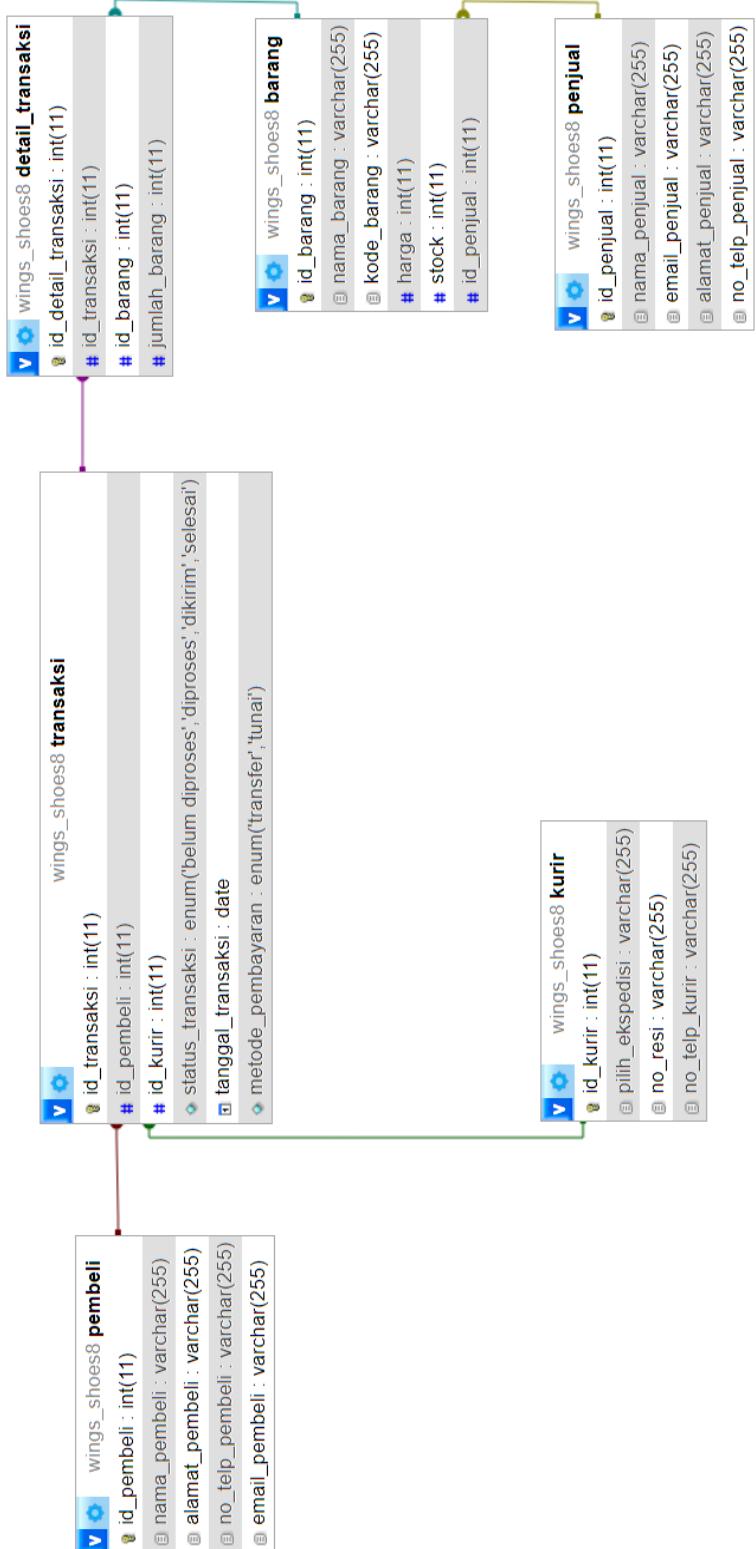
Pengguna Aplikasi [5 poin]

1. Pengguna smartphone yang familiar dengan berbelanja online.
2. Pelanggan yang mencari kenyamanan dalam berbelanja sepatu secara online tanpa harus mengunjungi toko fisik.
3. Semua kelompok usia yang memiliki minat dalam mode dan sepatu.

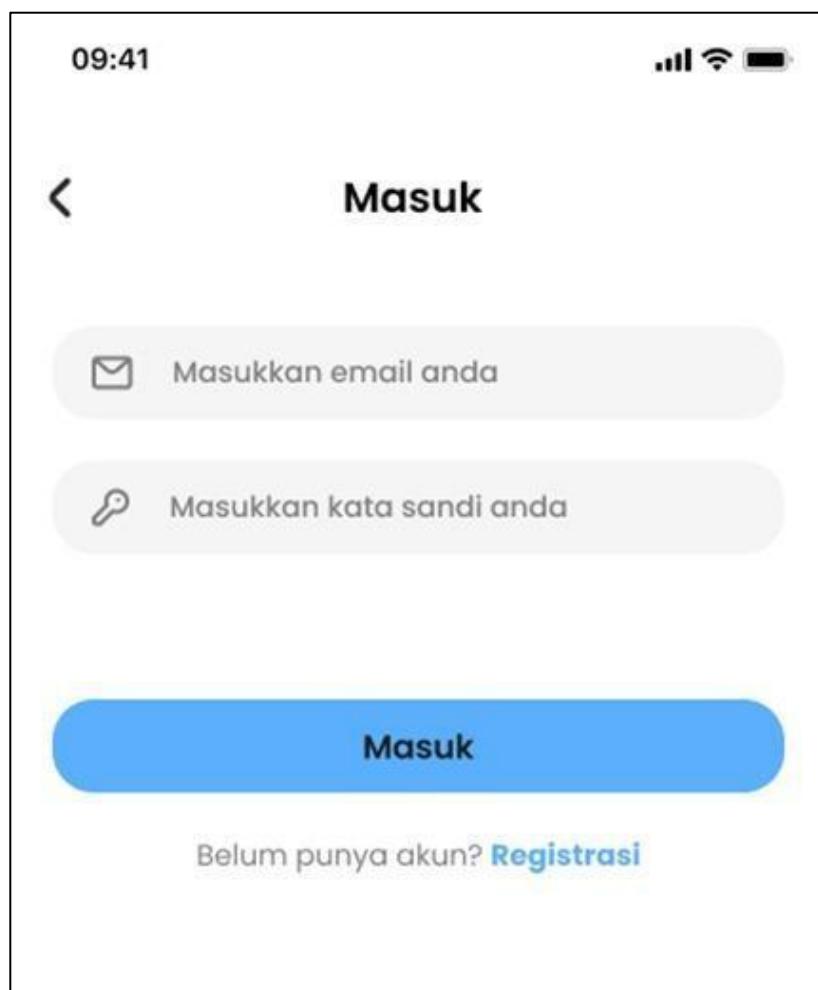
Desain Entity-Relationship Diagram (ERD) [10 poin]



Desain Relasi Basis Data



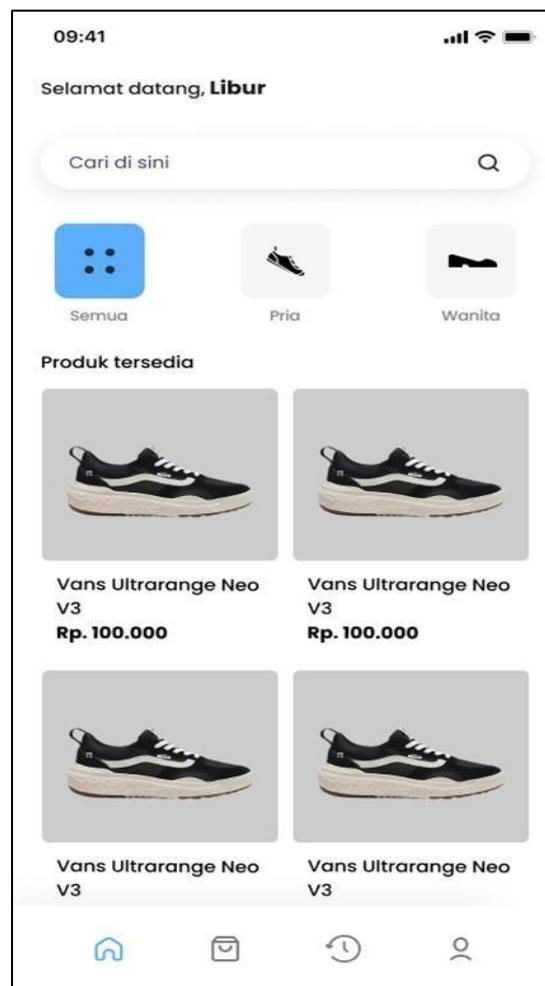
Desain Interface (Wireframe/Mockup) Beserta Query-nya [poin 40]



Query yang diperlukan

```
SELECT * FROM users WHERE username = 'nama_pengguna' AND password =  
'kata_sandi';
```

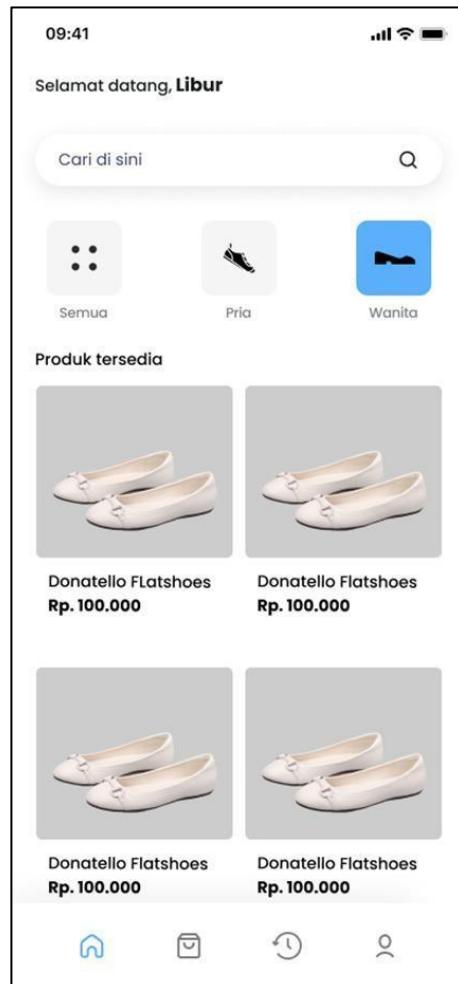
Tampilan Home



Query untuk tampilan home =

```
SELECT p.nama_produk, p.harga, k.nama_kategori  
FROM produk p  
JOIN kategori_produk k ON p.id_kategori = k.id_kategori;  
INSERT INTO produk (nama_produk, harga, id_kategori)  
VALUES ('Nama Produk', 100000, 1);
```

Tampilan Home Kategori Wanita



Query untuk tampilan
home kategori wanita

```
SELECT p.nama_produk, p.harga
FROM produk p
JOIN kategori_produk k ON p.id_kategori = k.id_kategori
INSERT INTO produk (nama_produk, harga, id_kategori)
VALUES ('Nama Produk', 100000, 1);
WHERE k.nama_kategori = 'Nama_Kategori';
```

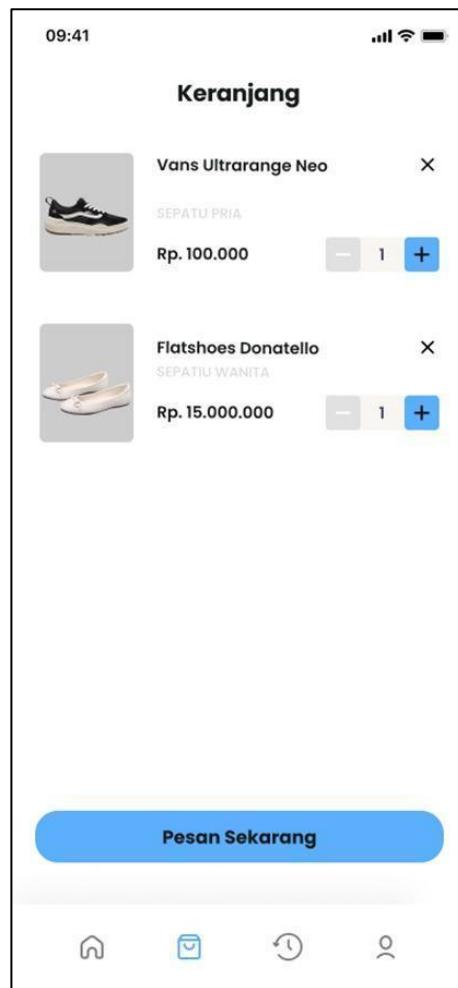
Tampilan Detail Produk



Query untuk detail produk :

```
SELECT p.nama_produk, p.deskripsi, p.harga, k.nama_kategori  
FROM produk p  
JOIN kategori_produk k ON p.id_kategori = k.id_kategori  
WHERE p.id_produk = 'ID_Produk';
```

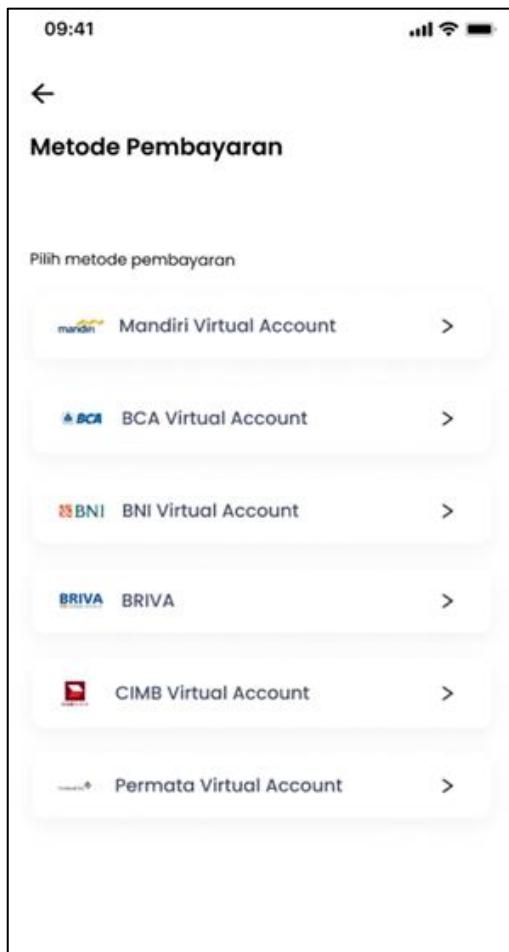
Tampilan Halaman keranjang



Query Keranjang :

```
SELECT p.nama_produk, p.harga, c.jumlah  
FROM keranjang c  
JOIN produk p ON c.id_produk = p.id_produk  
WHERE c.id_pelanggan = 'ID_Pelanggan';  
INSERT INTO keranjang (id_pelanggan, id_produk, jumlah)  
VALUES ('ID_Pelanggan', 'ID_Prod', 'Jumlah');  
INSERT INTO keranjang (id_pelanggan, id_produk, jumlah)  
VALUES (5, 101, 2);
```

Tampilan Metode Pembayaran



Query Metode Pembayaran :

```
INSERT INTO metode_pembayaran (nama_metode, detail_metode)
VALUES
('Bank Mandiri', 'Transfer Bank melalui Mandiri'),
('BCA', 'Transfer Bank melalui BCA'),
('BNI', 'Transfer Bank melalui BNI'),
('BRIVA', 'Pembayaran Virtual Account melalui BRI'),
('CIMB', 'Transfer Bank melalui CIMB Niaga'),
('Permata', 'Transfer Bank melalui Bank Permata');
SELECT id_metode, nama_metode, detail_metode
FROM metode_pembayaran;
```

Tampilan Konfirmasi Pembayaran

09:41 Signal Strength | Wi-Fi | Battery

[←](#)

Konfirmasi Pembayaran

Metode Pembayaran

 Mandiri Virtual Account

Detail Pembelian

Nama Penerima:	Cloud
Alamat:	Jalan, Griya Anyar, Gg. Bandeng, No. 37, Denpasar Selatan, Denpasar, Bali, 80221

Ringkasan Pembelian

Ix: Vans Ultrarange Neo	Rp. 100.000
Ix: Flatshoes Donatello	Rp. 15.000.000
Biaya pengiriman	Rp. 300.000
Total	Rp. 15.400.000

[Konfirmasi Pembelian](#)

Query tampilan konfirmasi pembayaran :

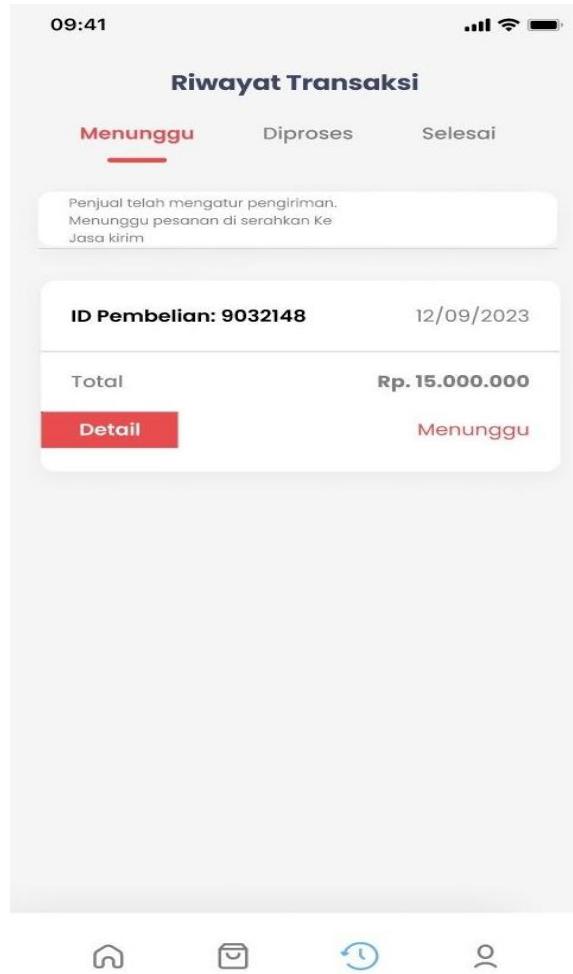
```

SELECT t.id_transaksi, t.tanggal_transaksi, t.total_harga, p.nama_produk,
d.jumlah, d.harga_satuan
FROM transaksi t
JOIN detail_transaksi d ON t.id_transaksi = d.id_transaksi
JOIN produk p ON d.id_produk = p.id_produk
WHERE t.id_pelanggan = 'ID_Pelanggan' AND t.status_pembayaran = 'Belum
Dikonfirmasi';
INSERT INTO pembayaran (id_transaksi, tanggal_pembayaran,
bukti_pembayaran)
VALUES ('ID_Transaksi', '2024-05-29', 'path/to/bukti_pembayaran.jpg');
UPDATE transaksi

```

```
SET status_pembayaran = 'Dikonfirmasi'  
WHERE id_transaksi = 'ID_Transaksi';
```

Tampilan Riwayat Transaksi (Menunggu)



Query tampilan riwayat transaksi(Selesai) :

```
SELECT  
    id_pembelian,  
FROM transaksi  
WHERE status = 'Menunggu'  
AND tanggal_transaksi = '2023-09-12';
```

Tampilan Riwayat Transaksi (Diproses)



Query tampilan riwayat transaksi(Selesai) :

SELECT

```
id_pembelian,  
id_pemesanan,  
id_pembayaran,  
waktu_pemesanan,  
waktu_pembayaran,  
waktu_pengiriman,  
total,  
status  
FROM riwayat_transaksi  
ORDER BY waktu_pengiriman DESC;
```

Tampilan Riwayat Transaksi (Selesai)



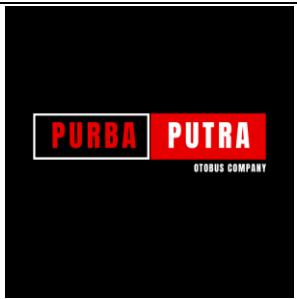
Query tampilan riwayat transaksi(Selesai) :

SELECT

```
id_pembelian,  
id_pemesanan,  
id_pembayaran,  
waktu_pemesanan,  
waktu_pembayaran,  
waktu_pengiriman,  
total,  
status
```

FROM riwayat_transaksi

WHERE status = 'Selesai';

<p>Judul Aplikasi</p> <h2 style="text-align: center;">MANAJEMEN PEMBELIAN TIKET BERBASIS WEB UNTUK PO BUS PURBA PUTRA</h2>	
<p>Dikembangkan Oleh</p> <ul style="list-style-type: none"> • 5220411380 - Muhammad Raihan • 5220411378 - Joanna Andini Prabaningrum Pasya • 5220411381 - Nasir Winanto • 5220411385 - Fajar Permana Putra • 5220411387 - Damar Wahyu Hidayat • 5220411407 - Fahreza Syahrul Ramadhan 	
<p>Deskripsi Singkat Instansi, Permasalahan, dan Solusi Aplikasi Yang Akan Dikembangkan [10 poin]</p> <p>PO Bus Purba Putra merupakan sebuah perusahaan otobus yang bergerak di bidang transportasi, yang berlokasi di Kec. Muntilan, Kab. Magelang, Prov. Jawa Tengah. Mereka menjadi bagian integral dari sistem transportasi perkotaan dan antar kota di lingkup wilayah Prov. Jawa Tengah, dengan sistem mengangkut penumpang dari satu lokasi ke lokasi lain dengan biaya yang terjangkau. Bus juga memiliki peran penting dalam pengurangan kemacetan dan polusi udara dengan menyediakan alternatif transportasi massal. Selain itu, bis juga bisa menjadi sumber penghasilan bagi pengemudi dan operatornya.</p> <p>Pembelian tiket bus secara langsung/konvensional seringkali memakan waktu dan tidak efisien bagi pelanggan. Prosesnya bisa menjadi rumit dengan antrian panjang di loket tiket atau mengharuskan pelanggan untuk datang langsung ke terminal bus. Selain itu, terkadang informasi tentang jadwal, harga, dan ketersediaan kursi tidak tersedia secara lengkap atau terupdate secara real-time.</p> <p>Pengembangan aplikasi pembelian tiket berbasis web untuk PO Bus Purba Putra dapat menjadi solusi yang efektif. Web tersebut dapat menyediakan platform yang user-friendly</p>	

bagi pelanggan untuk membeli tiket secara online berbasis Web. Fitur-fitur yang bisa disertakan dalam web tersebut antara lain:

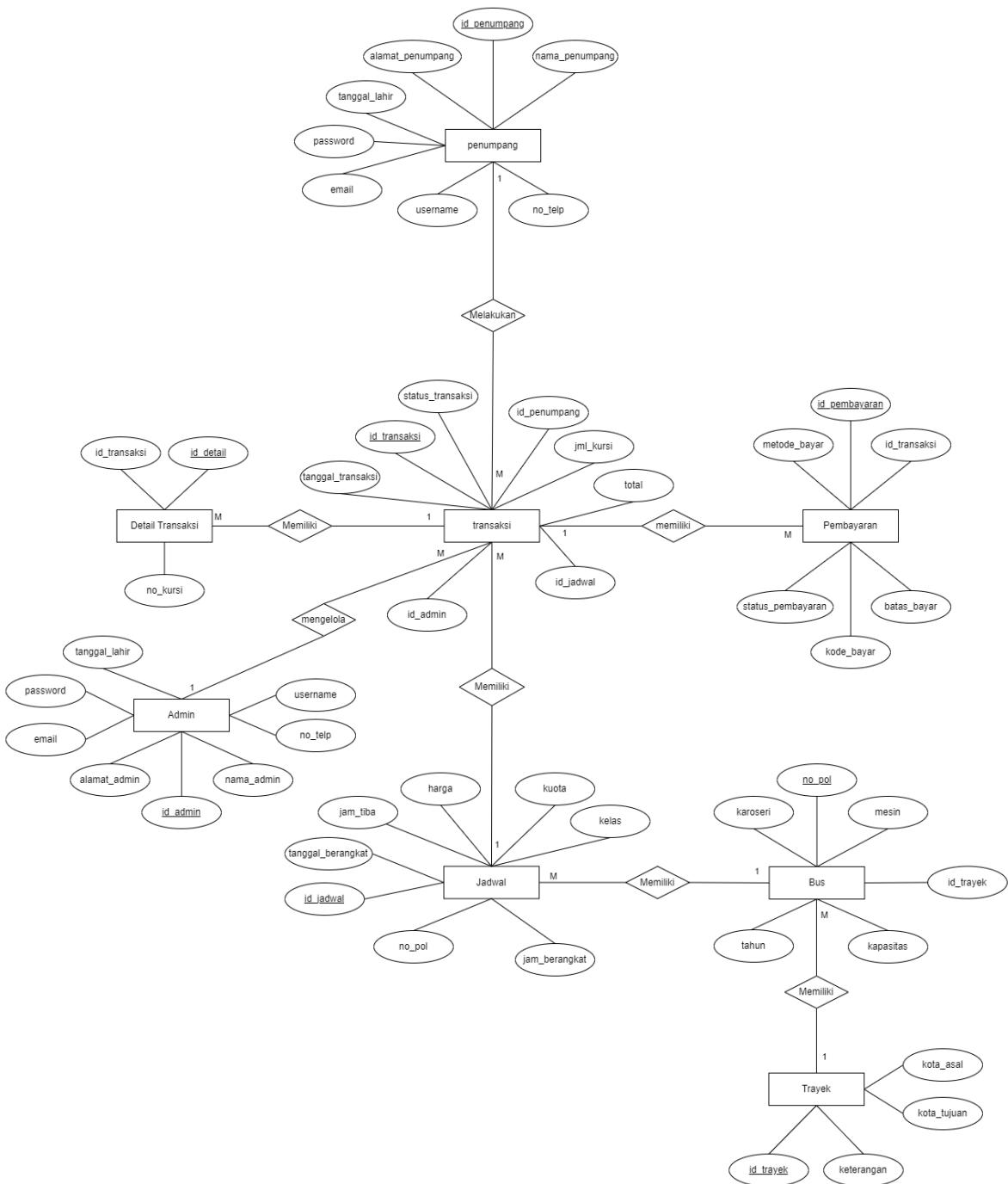
1. Pencarian Jadwal dan Rute: Pelanggan dapat dengan mudah mencari jadwal keberangkatan bus dan rute yang tersedia.
2. Pembelian Tiket Online: Pelanggan dapat memesan dan membayar tiket bus secara online melalui aplikasi, menghemat waktu dan tenaga.
3. Pemilihan Tempat Duduk: Pelanggan dapat memilih tempat duduk yang diinginkan di dalam bus jika tersedia.
4. Notifikasi dan Konfirmasi: Pelanggan akan menerima notifikasi dan konfirmasi pembelian tiket melalui email atau SMS.
5. Informasi Terkini: Aplikasi akan menyediakan informasi terkini tentang jadwal, harga, dan ketersediaan kursi secara real-time.
6. Histori Transaksi: Pelanggan dapat melihat histori transaksi mereka untuk referensi masa depan.

Dengan adopsi pembelian tiket berbasis Web, Po Bus Purba Putra dapat meningkatkan pengalaman pelanggan, meningkatkan efisiensi operasional, serta memperluas jangkauan pasar mereka melalui platform digital.

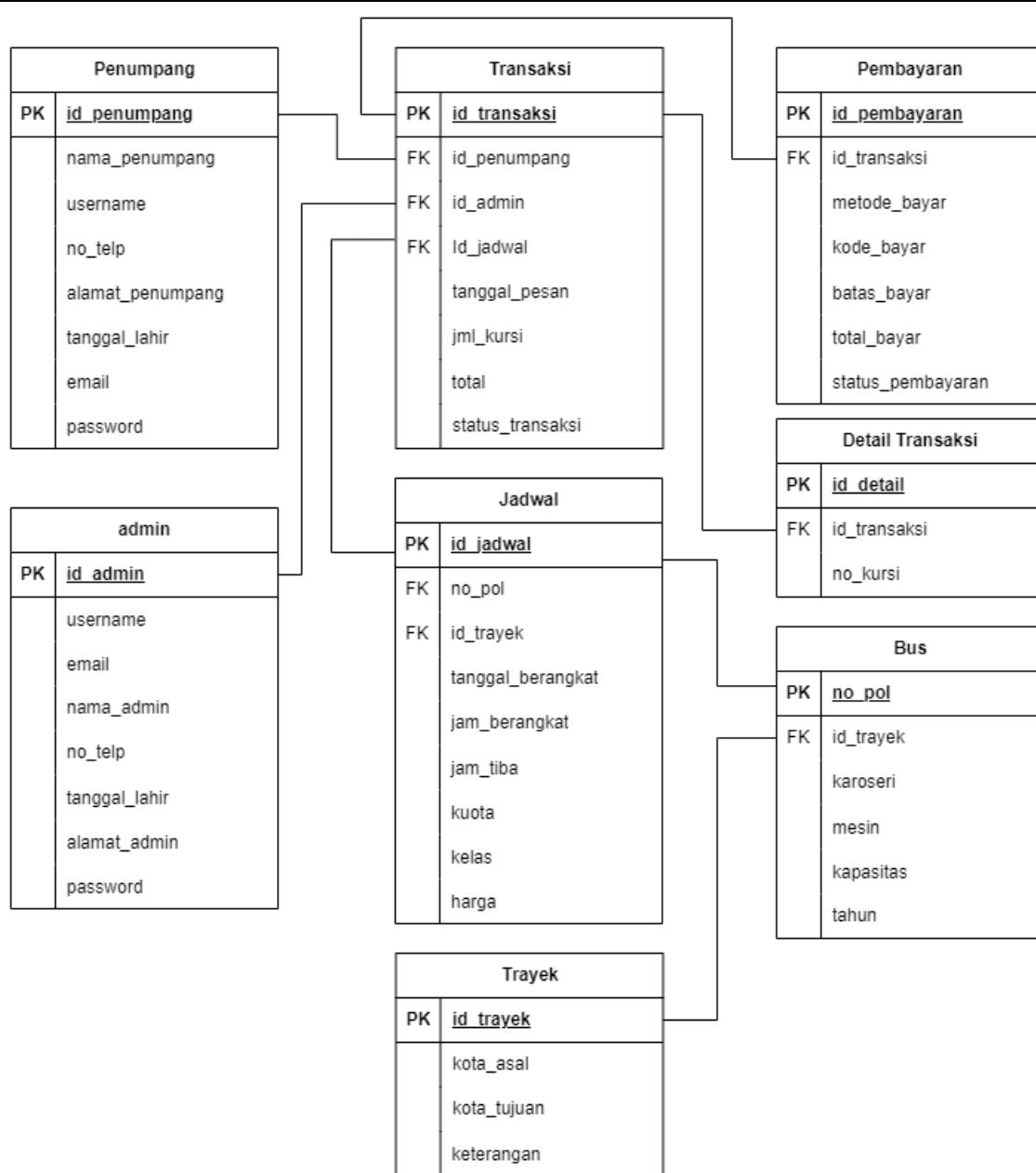
Pengguna Aplikasi [5 poin]

Pengguna web ini adalah penumpang dan admin

Desain Entity-Relationship Diagram (ERD) [10 poin]



Desain Relasi Basis Data [10 poin]



Query Data Definition Language (DDL) [25 poin]

- Membuat database

```
CREATE DATABASE purbaPutra;
```

- Membuat Table Penumpang

```
CREATE TABLE penumpang(
    id_penumpang CHAR(8) PRIMARY KEY,
    nama_penumpang VARCHAR(50) NOT NULL,
    username VARCHAR(30) NOT NULL,
    no_telp VARCHAR(15) NOT NULL,
    alamat_penumpang VARCHAR(100) NOT NULL,
    tanggal_lahir DATE NOT NULL,
    email VARCHAR(50) NOT NULL,
    PASSWORD VARCHAR(80) NOT NULL
);
```

- Membuat Table Admin

```
CREATE TABLE admin(
    id_admin CHAR(6) PRIMARY KEY,
    email VARCHAR(50) NOT NULL,
    username VARCHAR(30) NOT NULL,
    nama_admin VARCHAR(50) NOT NULL,
    no_telp VARCHAR(15) NOT NULL,
    tanggal_lahir DATE NOT NULL,
    alamat_admin VARCHAR(100) NOT NULL,
    PASSWORD VARCHAR(80) NOT NULL
);
```

- Membuat Table Trayek

```
CREATE TABLE trayek(  
    id_trayek CHAR(6) PRIMARY KEY,  
    kota_asal VARCHAR(20) NOT NULL,  
    kota_tujuan VARCHAR(20) NOT NULL,  
    keterangan VARCHAR(30) NOT NULL  
) ;
```

- Membuat Table Bus

```
CREATE TABLE bus(  
    no_pol CHAR(10) PRIMARY KEY,  
    id_trayek CHAR(6) NOT NULL,  
    karoseri VARCHAR(15) NOT NULL,  
    mesin VARCHAR(20) NOT NULL,  
    kapasitas TINYINT(2) NOT NULL,  
    tahun YEAR NOT NULL,  
    FOREIGN KEY (id_trayek) REFERENCES trayek(id_trayek) ON DELETE CASCADE  
) ;
```

- Membuat Table Jadwal

```
CREATE TABLE jadwal(  
    id_jadwal CHAR(9) PRIMARY KEY,  
    no_pol VARCHAR(10) NOT NULL,  
    tanggal_berangkat DATE NOT NULL,  
    jam_berangkat TIME NOT NULL,  
    jam_tiba TIME NOT NULL,  
    kelas ENUM ('ekonomi', 'patas', 'executive', 'super-executive') NOT NULL,  
    kuota TINYINT(2) NOT NULL,  
    harga FLOAT(14) NOT NULL,  
    FOREIGN KEY (no_pol) REFERENCES bus(no_pol) ON DELETE CASCADE  
) ;
```

- Membuat Table Transaksi

```
CREATE TABLE transaksi(
    id_transaksi CHAR(8) PRIMARY KEY,
    id_penumpang CHAR(8) NOT NULL,
    id_jadwal CHAR(9) NOT NULL,
    id_admin CHAR(8) NOT NULL,
    tanggal_transaksi DATE DEFAULT CURRENT_DATE() NOT NULL,
    jml_kursi TINYINT NOT NULL,
    status_transaksi ENUM('selesai', 'menunggu-pembayaran', 'batal', 'dibatalkan-admin', 'orefund') DEFAULT 'menunggu-pembayaran' NOT NULL,
    total INT(14) NOT NULL,
    FOREIGN KEY (id_penumpang) REFERENCES penumpang(id_penumpang) ON DELETE CASCADE,
    FOREIGN KEY (id_jadwal) REFERENCES jadwal(id_jadwal) ON DELETE CASCADE,
    FOREIGN KEY (id_admin) REFERENCES ADMIN(id_admin) ON DELETE CASCADE
);
```

- Membuat Table Detail Transaksi

```
CREATE TABLE detail_transaksi(
    id_detail INT PRIMARY KEY AUTO_INCREMENT,
    id_transaksi VARCHAR(8) NOT NULL,
    no_kursi VARCHAR(2) NOT NULL,
    FOREIGN KEY (id_transaksi) REFERENCES transaksi(id_transaksi) ON DELETE CASCADE
);
```

- Membuat Table Pembayaran

```
CREATE TABLE pembayaran(
    id_pembayaran CHAR(6) PRIMARY KEY,
    id_transaksi VARCHAR(8) NOT NULL,
    metode_bayar ENUM('bank-transfer', 'qrис') NOT NULL,
    kode_bayar VARCHAR(30) NOT NULL,
    batas_bayar DATETIME NOT NULL,
    status_pembayaran ENUM('berhasil', 'gagal', 'menunggu') DEFAULT 'menunggu' NOT NULL,
    FOREIGN KEY (id_transaksi) REFERENCES transaksi(id_transaksi) ON DELETE CASCADE
);
```

- Membuat Trigger Untuk Mengurangi kuota jadwal

```
DELIMITER //

CREATE TRIGGER kurangi_kuota_jadwal
AFTER INSERT ON transaksi
FOR EACH ROW
BEGIN
    UPDATE jadwal
    SET kuota = kuota - NEW.jml_kursi
    WHERE id_jadwal = NEW.id_jadwal;
END//

DELIMITER ;
```

- Membuat Trigger untuk set kuota pada jadwal

```
CREATE TRIGGER set_kuota
BEFORE INSERT ON jadwal
FOR EACH ROW
BEGIN
    DECLARE kuota_jadwal INT;
    SELECT kapasitas INTO kuota_jadwal FROM bus WHERE no_pol = NEW.no_pol;
    SET NEW.kuota = kuota_jadwal;
END//;

DELIMITER ;
```

- Membuat Trigger untuk set status transaksi setelah pembayaran

```
DELIMITER //

CREATE TRIGGER set_status_transaksi
AFTER UPDATE ON pembayaran
FOR EACH ROW
BEGIN
    IF NEW.status_pembayaran = 'berhasil' THEN
        UPDATE transaksi
        SET status_transaksi = 'selesai'
        WHERE id_transaksi = NEW.id_transaksi;
    ELSEIF NEW.status_pembayaran = 'gagal' THEN
        UPDATE transaksi
        SET status_transaksi = 'batal'
        WHERE id_transaksi = NEW.id_transaksi;
    END IF;
END//;

DELIMITER ;
```

- Membuat Trigger untuk set total transaksi

```
DELIMITER //

CREATE TRIGGER set_total_bayar
BEFORE INSERT ON transaksi
FOR EACH ROW
BEGIN
    DECLARE harga_jadwal INT;
    SELECT harga INTO harga_jadwal FROM jadwal WHERE id_jadwal = NEW.id_jadwal;
    SET NEW.total = NEW.jml_kursi * harga_jadwal;
END//

DELIMITER ;
```

Desain Interface (Wireframe/Mockup) Beserta Query-nya [poin 40]

- Halaman Registrasi

The wireframe shows a registration form titled "DAFTAR AKUN". It includes fields for Username, Nama Lengkap, Tanggal Lahir, Alamat, Nomor Telpon, Email, Password, and Password Lagi. There are also "Sudah punya akun?" and "Daftar" buttons.

```
INSERT INTO penumpang
(id_penumpang, username, nama_penumpang, tanggal_lahir, alamat_penumpang, no_telp, email, PASSWORD) VALUES
("PNP23974", "Mmat", "Mamat Syahputra", "1995-05-23", "Jl. Magelang Km.8", "083427485706", "mamat@gmail.com", "12345");
```

- Halaman Login



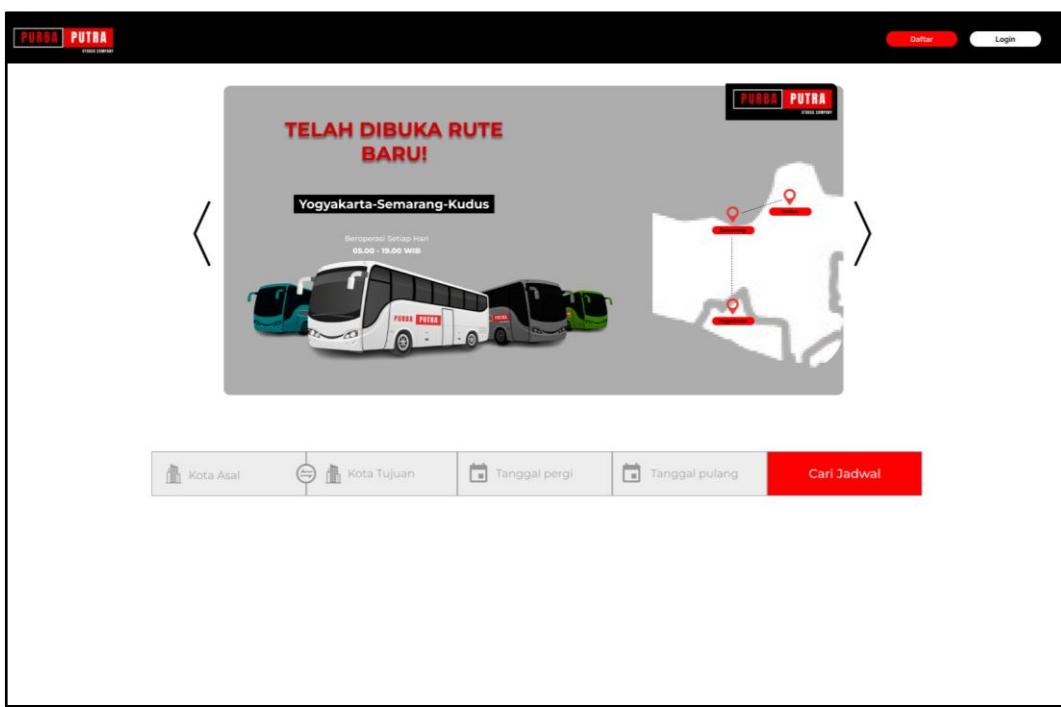
1. Login Penumpang

```
● ● ●  
SELECT id_penumpang, username FROM penumpang WHERE email="mamat@gmail.com" AND PASSWORD="12345";
```

2. Login Admin

```
● ● ●  
SELECT id_admin, username FROM ADMIN WHERE email="admin@xyz.com" AND PASSWORD="admin123";
```

- Halaman Home & Daftar Jadwal



PURBA PUTRA

Hi, Mintika

Terminal Jombor → Terminal Kudus

9 Perjalanan Di Temukan

Jam Keberangkatan	Jam kedatangan	Kelas	Kursi	Harga	Pilih	
06:00	Estimasi Perjalanan 6 Jam	11:00	Executive	25 Kursi	Rp 100.000	Pilih
07:30	Estimasi Perjalanan 6 Jam	12:30	Executive	20 Kursi	Rp 100.000	Pilih
09:00	Estimasi Perjalanan 6 Jam	14:00	Executive	18 Kursi	Rp 100.000	Pilih
11:00	Estimasi Perjalanan 6 Jam	16:00	Executive	7 Kursi	Rp 100.000	Pilih
12:30	Estimasi Perjalanan 6 Jam	17:30	Executive	1 Kursi	Rp 100.000	Pilih
14:00	Estimasi Perjalanan 6 Jam	19:00	Super Executive	12 Kursi	Rp 100.000	Pilih
16:00	Estimasi Perjalanan 6 Jam	21:00	Super Executive	10 Kursi	Rp 100.000	Pilih
17:30	Estimasi Perjalanan 6 Jam	22:30	Executive	25 Kursi	Rp 100.000	Pilih
19:00	Estimasi Perjalanan 6 Jam	00:00	Executive	15 Kursi	Rp 100.000	Pilih

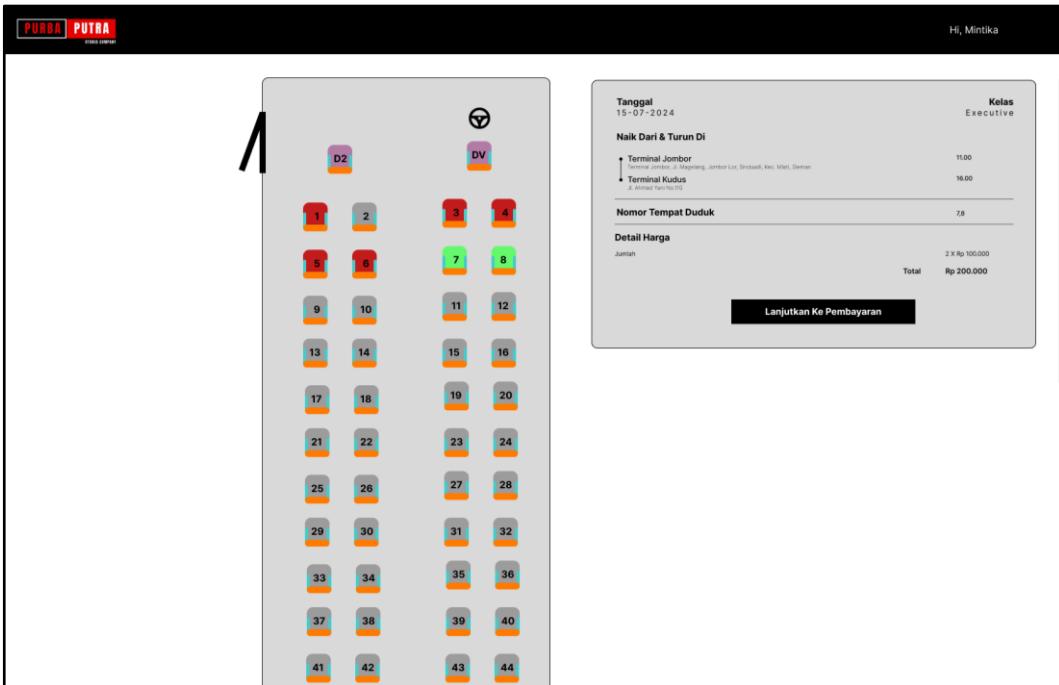
1. Cari dan daftar Jadwal tanpa tanggal pulang.

```
SELECT j.*, t.*  
FROM jadwal j  
JOIN bus b USING(no_pol)  
JOIN trayek t USING(id_trayek)  
WHERE (t.kota_asal = "Jakarta" AND t.kota_tujuan = "Yogyakarta" AND j.tanggal_berangkat="2024-05-29")
```

2. Cari dan daftar Jadwal dengan tanggal pulang.

```
SELECT j.*, t.*  
FROM jadwal j  
JOIN bus b USING(no_pol)  
JOIN trayek t USING(id_trayek)  
WHERE (t.kota_asal = "Jakarta" AND t.kota_tujuan = "Yogyakarta" AND j.tanggal_berangkat="2024-05-29") OR  
(t.kota_asal = "Yogyakarta" AND t.kota_tujuan = "Jakarta" AND j.tanggal_berangkat="2024-05-30");
```

- Halaman Pemesanan Pilih Kursi



1. Kueri untuk menampilkan data transaksi dan kursi not available (terjual).

```
SELECT j.tanggal_berangkat, j.harga, t.kota_asal, t.kota_tujuan, dt.no_kursi
FROM jadwal j
JOIN bus b USING(no_pol)
JOIN trayek t USING(id_trayek)
JOIN transaksi ts USING(id_jadwal)
JOIN detail_transaksi dt USING(id_transaksi)
WHERE id_jadwal = "JKTJOGAJD" AND (ts.status_transaksi = "selesai" OR ts.status_transaksi = "menunggu-pembayaran");
```

2. Mengambil data admin.

Admin disini adalah admin yang bertugas pada trayek tersebut ditentukan oleh bagian front end. Jika terdapat complain atau permintaan pembatalan akan dilayani oleh admin tersebut.

```
SELECT id_admin FROM ADMIN;
```

3. Kueri untuk menyimpan data transaksi.



```
INSERT INTO transaksi (id_transaksi, id_penumpang, id_jadwal, id_admin, jml_kursi) VALUES ("24523146", "PNP23974", "JKTJOGAJD", "ADM485", 2);
```

4. Kueri untuk menyimpan detail transaksi



```
INSERT INTO detail_transaksi (id_transaksi, no_kursi) VALUES ("24523146", "A1");
```

- Halaman Pilih Metode Pembayaran

The screenshot shows a user interface for selecting a payment method. At the top, there's a header with 'PURBA PUTRA' and a user profile 'Hi, Mintika'. Below the header, the transaction details are displayed:

Tanggal	15-07-2024
Naik Dari & Turun Di	
Terminal Jombor	11.00
Terminal Kudus	16.00
Nomor Tempat Duduk	7.8
Detail Harga	
Jumlah	2 x Rp 100.000
Total	Rp 200.000

Below the transaction details is a section titled 'Pilih metode pembayaran' (Select payment method). It lists several options:

- Internet Banking
 - BCA Mobile
 - BRI Mobile
 - BNI mobile
 - Livin by Mandiri
- Kartu Debit/ Kredit
- Transfer ATM
- E wallet
- Mitra Agen

A red button at the bottom right says 'Lanjutkan Pembayaran' (Continue Payment).

1. Kueri untuk menampilkan data transaksi



```
SELECT t.tanggal_transaksi, tr.kota_asal, tr.kota_tujuan,
       GROUP_CONCAT(dt.no_kursi ORDER BY dt.no_kursi SEPARATOR ',') AS no_kursi,
       t.jml_kursi, j.harga, t.total
  FROM transaksi t
 JOIN detail_transaksi dt USING(id_transaksi)
 JOIN jadwal j USING(id_jadwal)
 JOIN bus b USING(no_pol)
 JOIN trayek tr USING(id_trayek)
 WHERE id_transaksi="24523146"
 GROUP BY t.id_transaksi;
```

2. Kueri untuk menyimpan metode bayar dan kode bayar

```
INSERT INTO pembayaran (id_pembayaran, id_transaksi, metode_bayar, kode_bayar)
VALUES ("BYR002", "24523146", "bank-transfer", "43657278392");
```

3. Kueri untuk update batas pembayaran.

```
UPDATE pembayaran SET batas_bayar="2024-05-24 13:00:00" WHERE id_transaksi="24523146";
```

- Halaman Pembayaran

The screenshot shows a mobile application interface for a bus company named 'PURBA PUTRA'. At the top, there's a header with the company name and a user greeting 'Hi, Mintika'. Below the header, a large button with three colored dots (red, yellow, green) is visible. The main content area displays a message 'Menunggu Pembayaran' with a clock icon, indicating that payment is due by '25-05-2024 07:14:32'. It also shows a virtual account number '690952302342572680' and a 'Salin' (Copy) button. Below this, there are several sections of information: 'Tanggal' (Date) '15-07-2024', 'Naik Dari & Turun Di' (Boarding & alighting points), 'Nomor Tempat Duduk' (Seat Number), and 'Detail Harga' (Price Details). The price details show two items at 'Rp 100.000' each, totaling 'Rp 200.000'.

1. Kueri untuk menampilkan data transaksi

```
SELECT t.tanggal_transaksi, tr.kota_asal, tr.kota_tujuan,
       GROUP_CONCAT(dt.no_kursi ORDER BY dt.no_kursi SEPARATOR ',') AS no_kursi,
       t.jml_kursi, j.harga, t.total, p.metode_bayar, p.kode_bayar
  FROM transaksi t
 JOIN pembayaran p USING(id_transaksi)
 JOIN detail_transaksi dt USING(id_transaksi)
 JOIN jadwal j USING(id_jadwal)
 JOIN bus b USING(no_pol)
 JOIN trayek tr USING(id_trayek)
 WHERE id_transaksi="24523146";
```

2. Kueri untuk update status pembayaran

```
UPDATE pembayaran SET status_pembayaran="berhasil" WHERE id_transaksi='24523146';
```

- Halaman E-ticket



```
SELECT t.id_transaksi, j.tanggal_berangkat,
       GROUP_CONCAT(dt.no_kursi ORDER BY dt.no_kursi SEPARATOR ',') AS no_kursi,
       t.jml_kursi, j.harga, t.total
  FROM transaksi t
 JOIN detail_transaksi dt USING(id_transaksi)
 JOIN jadwal j USING(id_jadwal)
 WHERE id_transaksi="24523146"
 GROUP BY t.id_transaksi;
```

- Halaman Profil

The screenshot shows a user profile page for a user named Reza Haji. At the top, there's a navigation bar with the Purba Putra logo and a greeting 'Hi, Mintika'. On the left, there are two menu items: 'PERJALAN SAYA' (Journey Log) and 'PROFILE SAYA' (My Profile). The main content area is titled 'PROFILE'. It displays the user's name 'REZA HAJI', birth date '39-12-1876', gender 'PRA - WANITA', and email 'REZA_HAJI@gmail.com'. There are tabs for 'RINCIAN KONTAK' (Contact Details), 'KODE AREA' (Area Code), and 'NO TELEPON' (Phone Number). Below the contact details, there's a placeholder for a profile picture.

1. Kueri untuk menampilkan data profil penumpang.

```
SELECT nama_penumpang, alamat_penumpang, tanggal_lahir, no_telp, username, email  
FROM penumpang WHERE id_penumpang = "PNP23974";
```

2. Kueri untuk mengupdate data profil penumpang.

```
UPDATE penumpang SET  
    nama_penumpang="Mamat Syahputra Andrianto",  
    alamat_penumpang="Jl. Magelang Km.8",  
    tanggal_lahir="1995-05-23",  
    no_telp="083427485706"  
WHERE id_penumpang="PNP23974"
```

- Halaman Riwayat Perjalanan

The screenshot shows a travel history page from the Purba Putra website. At the top, there's a header with the logo 'PURBA PUTRA' and a greeting 'Hi, Mintika'. On the left, there are navigation links for 'PERJALAN SAYA' and 'PROFILE SAYA'. The main content area is titled 'PERJALANAN' and displays three travel records in a grid format.

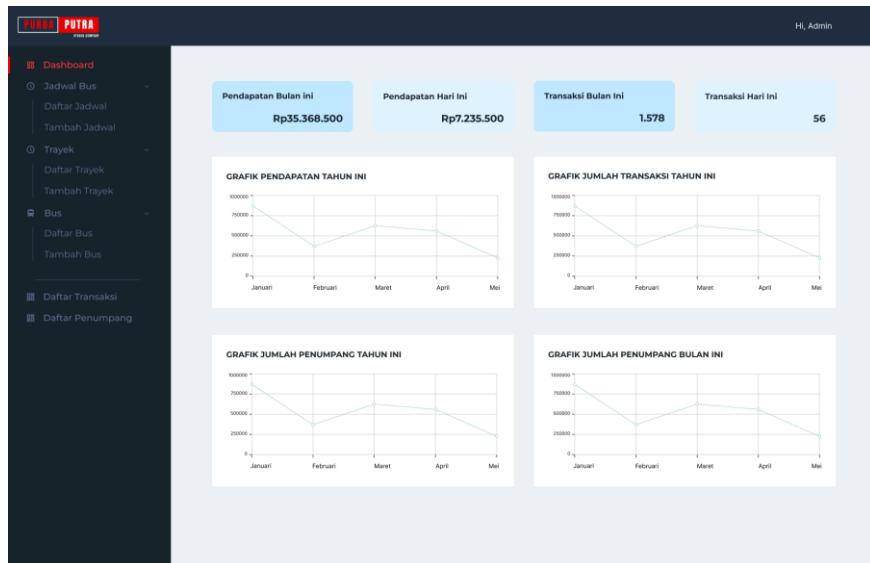
Tanggal Transaksi	Class	Status		
22-04-2024	Super Executive	Menunggu Pembayaran		
TANGGAL 29-04-2024	TIME 16:00	KOTA ASAL Jakarta	KOTA TUJUAN Jakarta	No. Kursi A1, A2
TOTAL		2 x 250.000 Rp500.000		

Tanggal Transaksi	Class	Status		
22-04-2024	Executive	Selesai		
TANGGAL 29-04-2024	TIME 16:00	KOTA ASAL Jakarta	KOTA TUJUAN Jakarta	No. Kursi A1, A2
TOTAL		2 x 250.000 Rp500.000		

Tanggal Transaksi	Class	Status		
22-04-2024	Executive	Selesai		
TANGGAL 29-04-2024	TIME 16:00	KOTA ASAL Jakarta	KOTA TUJUAN Jakarta	No. Kursi A1, A2

```
SELECT t.id_transaksi, t.tanggal_transaksi, j.tanggal_berangkat,
       tr.kota_asal, tr.kota_tujuan,
       GROUP_CONCAT(dt.no_kursi ORDER BY dt.no_kursi SEPARATOR ',') AS no_kursi,
       t.jml_kursi, j.harga, t.total, j.kelas, t.status_transaksi
  FROM transaksi t
 JOIN detail_transaksi dt USING(id_transaksi)
 JOIN jadwal j USING(id_jadwal)
 JOIN bus b USING(no_pol)
 JOIN trayek tr USING(id_trayek)
 WHERE t.id_penumpang="PNP23974"
 GROUP BY id_transaksi;
```

- Halaman Dashboard Admin



1. Query untuk mengambil data pendapatan bulan ini.

```
SELECT SUM(total)
FROM transaksi
WHERE
    MONTH(tanggal_transaksi)=MONTH(NOW()) AND
    status_transaksi = 'selesai';
```

2. Query untuk mengambil data pendapatan Hari Ini.

```
SELECT IFNULL(SUM(total), 0) AS 'total hari ini'
FROM transaksi
WHERE
    tanggal_transaksi=CURDATE() AND
    status_transaksi = 'selesai';
```

3. Query untuk mengambil data jumlah transaksi bulan ini

```
SELECT COUNT(*)
FROM transaksi
WHERE
    MONTH(tanggal_transaksi)=MONTH(NOW()) AND
    YEAR(tanggal_transaksi)=YEAR(NOW())
    AND status_transaksi = 'selesai';
```

4. Query untuk mengambil data pendapatan hari ini

```
SELECT COUNT(*)
FROM transaksi
WHERE
    tanggal_transaksi=CURDATE() AND
    status_transaksi = 'selesai';
```

5. Query untuk mengambil data grafik pendapatan tahun ini.

```
SELECT  
    MONTHNAME(tanggal_transaksi) AS bulan,  
    SUM(total) AS total_bulanan  
FROM  
    transaksi  
GROUP BY  
    MONTH(tanggal_transaksi);
```

6. Query untuk mengambil data grafik jumlah transaksi tahun ini.

```
SELECT  
    MONTHNAME(tanggal_transaksi) AS bulan,  
    COUNT(*) AS jml_transaksi  
FROM  
    transaksi  
GROUP BY  
    MONTH(tanggal_transaksi);
```

7. Query untuk mengambil data grafik jumlah penumpang tahun ini.

```
SELECT  
    MONTHNAME(tanggal_transaksi) AS bulan,  
    SUM(jml_kursi) AS jml_pnp  
FROM  
    transaksi  
WHERE  
    YEAR(tanggal_transaksi) = YEAR(NOW())  
GROUP BY  
    MONTH(tanggal_transaksi);
```

8. Query untuk mengambil data grafik jumlah penumpang bulan ini.

```
SELECT  
    SUM(jml_kursi) AS jml_pnp  
FROM  
    transaksi  
WHERE  
    MONTH(tanggal_transaksi) = MONTH(NOW()) AND  
    YEAR(tanggal_transaksi) = YEAR(NOW());
```

- Halaman Daftar Jadwal

ID Jadwal	No. Pol	Asal	Tujuan	Tanggal	Brngkt	Tiba	Kelas	Harga	Kuota	Aksi
JOGJKT001	AB 8888 SC	Yogyakarta	Jakarta	28 Desember 2024	16:30	05:00	Super Exe	Rp150.000	28	
JOGSBY001	B 3275 KJ	Yogyakarta	Surabaya	28 Mei 2024	10:30	18:00	Ekonomi	Rp120.000	1	
JOCSBY002	AB 3456 JD	Yogyakarta	Surabaya	28 Mei 2024	12:30	20:00	Patas	Rp150.000	35	
JOGMGL001	AB 3759 AI	Yogyakarta	Magelang	28 Mei 2024	10:30	18:00	Executive	Rp150.000	5	
JOGSMG001	AB 1394 GD	Yogyakarta	Semarang	28 Mei 2024	12:30	20:00	Patas	Rp150.000	Full	
JKTJOG001	AB 1394 GD	Jakarta	Yogyakarta	28 Mei 2024	15:30	04:00	Executive	Rp150.000	35	
JKTJOG002	AB 1394 GD	Jakarta	Yogyakarta	28 Mei 2024	16:00	04:30	Executive	Rp150.000	Full	
JKTJOG003	AB 1394 GD	Jakarta	Yogyakarta	28 Mei 2024	15:30	04:00	Executive	Rp150.000	35	
JKTJOG004	AB 1394 GD	Jakarta	Yogyakarta	28 Mei 2024	16:30	04:00	Executive	Rp150.000	10	
JKTJOG005	AB 1394 GD	Jakarta	Yogyakarta	28 Mei 2024	15:00	03:00	Executive	Rp150.000	35	

1. Query untuk menampilkan seluruh data jadwal.

```
SELECT j.id_jadwal, j.no_pol, t.kota_asal,
       t.kota_tujuan, j.tanggal_berangkat, j.jam_berangkat,
       j.jam_tiba, j.kelas, j.harga, j.kuota
  FROM jadwal j
 JOIN bus b USING(no_pol)
 JOIN trayek t USING(id_trayek)
```

2. Query untuk mengupdate data jadwal.

```
UPDATE jadwal SET  
    tanggal_berangkat='2024-05-30',  
    jam_berangkat="16:00:00",  
    jam_tiba="04:00:00",  
    kuota="20",  
    kelas="super-executive",  
    harga=350000  
WHERE id_jadwal="JKTJOGAJD";
```

3. Query untuk menghapus data jadwal.

```
DELETE FROM jadwal WHERE id_jadwal="JKTJOGAJD";
```

- Halaman Tambah Jadwal

The screenshot shows the 'Tambah Jadwal' (Add Schedule) page of a web application. The left sidebar has a dark theme with white text and icons. The main area has a light gray background. The form fields are as follows:

- ID Jadwal: Input field containing "ID Jadwal".
- Bus: Select dropdown menu showing "Pilih Bus".
- Kelas: Select dropdown menu showing "Pilih Kelas".
- Harga: Input field containing "Harga Tiket".
- Tanggal Keberangkatan: Date input field with a calendar icon.
- Jam Berangkat: Time input field showing "00:00".
- Jam Tiba: Time input field showing "00:00".

A large black 'Simpan' (Save) button is located at the bottom right of the form area.

1. Query menampilkan seluruh data bus untuk select bus.

```
● ● ●  
  
SELECT no_pol FROM bus;
```

2. Query untuk menyimpan data jadwal.

```
● ● ●  
  
INSERT INTO jadwal (id_jadwal, no_pol, tanggal_berangkat, jam_berangkat, jam_tiba, kelas, harga)  
VALUES ("JKTJOGAJD", "AB 3758 SH", "2024-05-29", "15:00", "04:00", 'executive', 325000)
```

- Halaman Daftar Trayek

ID Trayek	Kota Asal	Kota Tujuan	Keterangan	Aksi
JKTSCJ	Yogyakarta	Jakarta	Term. Jombor - Term. Magelang - Term. Pulo Gebang	
JKTSCJ	Yogyakarta	Jakarta	Yogyakarta - Magelang - Secang - Jakarta	
JKTSCJ	Yogyakarta	Jakarta	Yogyakarta - Magelang - Secang - Jakarta	
JKTSCJ	Yogyakarta	Jakarta	Yogyakarta - Magelang - Secang - Jakarta	
JKTSCJ	Yogyakarta	Jakarta	Yogyakarta - Magelang - Secang - Jakarta	
JKTSCJ	Yogyakarta	Jakarta	Yogyakarta - Magelang - Secang - Jakarta	
JKTSCJ	Yogyakarta	Jakarta	Yogyakarta - Magelang - Secang - Jakarta	
JKTSCJ	Yogyakarta	Jakarta	Yogyakarta - Magelang - Secang - Jakarta	
JKTSCJ	Yogyakarta	Jakarta	Yogyakarta - Magelang - Secang - Jakarta	

1. Query menampilkan seluruh data trayek.

```
● ● ●  
  
SELECT * FROM trayek
```

2. Query mengupdate data trayek.

```
UPDATE trayek SET keterangan="jakarta-yogyakarta" WHERE id_trayek="JKTJOG";
```

3. Query menghapus data trayek.

```
DELETE FROM trayek WHERE id_trayek = "JOGSJS";
```

- Halaman Tambah Trayek

The screenshot shows a dark-themed web application interface. On the left is a sidebar with navigation links: Dashboard, Jadwal Bus, Trayek (selected), Daftar Trayek, Tambah Trayek, Bus, Daftar Transaksi, and Daftar Penumpang. The main content area has a title 'TAMBAH TRAYEK'. It contains four input fields: 'ID Trayek' (Normor Polisi), 'Kota Asal' (Pilih Karoseri), 'Kota Tujuan' (Pilih Mesin), and 'Keterangan' (Keterangan). A 'Simpan' (Save) button is located at the bottom right of the form.

```
INSERT INTO trayek
(id_trayek, kota_asal, kota_tujuan, keterangan) VALUES
('JKTJOG', 'Jakarta', 'YOGYAKARTA', "jakarta-magelang-jogja")
```

- Halaman Daftar Bus

The screenshot shows a dark-themed web application interface. At the top left is the logo 'PURBA PUTRA'. On the top right, it says 'Hi, Admin'. A vertical sidebar on the left contains navigation links: 'Dashboard', 'Jadwal Bus', 'Trayek', 'Bus' (which is currently selected), 'Daftar Bus', 'Tambah Bus', 'Daftar Transaksi', and 'Daftar Penumpang'. The main content area is titled 'DAFTAR BUS'. It features a table with columns: No. Pol, Karoseri, Mesin, Kapasitas, Tahun, and Aksi. Below the table is a navigation bar with buttons for page numbers 1, 2, 3, and a search bar.

No. Pol	Karoseri	Mesin	Kapasitas	Tahun	Aksi
AB 3628 SF	Adiputro	Mercedes-Benz	55	2021	
AB 1254 DO	Adiputro	Hino	50	2019	
B 4875 KF	Adiputro	Mercedes-Benz	28	2023	
B 5759 FS	Laksana	Mercedes-Benz	28	2021	
B 4875 KF	Adiputro	Mercedes-Benz	28	2023	
B 4875 KF	Adiputro	Mercedes-Benz	28	2023	
B 4875 KF	Adiputro	Mercedes-Benz	28	2023	
B 4875 KF	Adiputro	Mercedes-Benz	28	2023	
B 4875 KF	Adiputro	Mercedes-Benz	28	2023	
B 4875 KF	Adiputro	Mercedes-Benz	28	2023	

1. Query menampilkan seluruh data bus.

The terminal window has three colored dots (red, yellow, green) at the top. The command 'SELECT * FROM bus;' is displayed in blue text.

```
SELECT * FROM bus;
```

2. Query untuk mengupdate data bus.

The terminal window has three colored dots (red, yellow, green) at the top. The command 'UPDATE bus SET id_trayek="JOGJKT", kapasitas="35" WHERE no_pol="AB 3758 SH";' is displayed in blue text.

```
UPDATE bus SET id_trayek="JOGJKT", kapasitas="35" WHERE no_pol="AB 3758 SH";
```

3. Query untuk menghapus data bus.

The terminal window has three colored dots (red, yellow, green) at the top. The command 'DELETE FROM bus WHERE no_pol = "AB 5839 KF";' is displayed in blue text.

```
DELETE FROM bus WHERE no_pol = "AB 5839 KF";
```

- Halaman Tambah Bus

PURBA PUTRA
SISTEM PENJUALAN

Hi, Admin

TAMBAH BUS

Nomor Polisi
Nomor Polisi

Trayek
Pilih Trayek

Karoseri
Pilih Karoseri

Mesin
Pilih Mesin

Kapasitas Penumpang
Kapasitas

Tahun
Pilih Tahun

Simpan

1. Query untuk menampilkan data trayek.

```
SELECT CONCAT(kota_asal, '-' , kota_tujuan) AS trayek FROM trayek;
```

2. Query untuk menambahkan data bus.

```
INSERT INTO bus (no_pol, id_trayek, karoseri, mesin, kapasitas, tahun) VALUES  
("AB 3758 SH", "JKTJOG", "Laksana", "Scania", "50", "2022");
```

- Daftar Transaksi

ID Transaksi	Tanggal	Asal	Tujuan	Jumlah Penumpang	Status	Aksi
24523121	28 Desember 2024	Jakarta	Jakarta	28	Selesai	
SBYUSO	25 Mei 2024	Surabaya	Surabaya	1	Refund	
SBYSJY	25 Mei 2024	Surabaya	Surabaya	15	Batal	
MGLHSJ	25 Mei 2024	Magelang	Magelang	5	Menunggu	
SMGHSJ	25 Mei 2024	Semarang	Semarang	12	Selesai	
JOGDOF	25 Mei 2024	Yogyakarta	Yogyakarta	35	Selesai	
JOGSJK	25 Mei 2024	Yogyakarta	Yogyakarta	23	Selesai	
JOGDOF	25 Mei 2024	Yogyakarta	Yogyakarta	35	Selesai	
JOGHSY	25 Mei 2024	Yogyakarta	Yogyakarta	10	Selesai	
JOGSTU	25 Mei 2024	Yogyakarta	Yogyakarta	35	Selesai	

1. Query untuk menampilkan data transaksi.

```
SELECT t.id_transaksi, t.tanggal_transaksi, tr.kota_asal,
       tr.kota_tujuan, t.jml_kursi, t.total, t.status_transaksi
FROM transaksi t
JOIN detail_transaksi USING(id_transaksi)
JOIN jadwal USING(id_jadwal)
JOIN bus b USING(no_pol)
JOIN trayek tr USING(id_trayek)
WHERE id_admin="ADM485"
GROUP BY t.id_transaksi;
```

2. Query untuk admin membatalkan transaksi.

```
UPDATE transaksi SET status_transaksi="dibatalkan-admin" WHERE id_transaksi="24523146"
```

3. Query untuk admin melakukan refund.

```
● ● ●  
UPDATE transaksi SET status_transaksi="refund" WHERE id_transaksi="24523146"
```

- Daftar Penumpang

ID Jadwal	No. Pol	Asal	Tujuan	Tanggal	Jumlah Penumpang	Aksi
JKTSCJ	AB 8888 SC	Yogyakarta	Jakarta	28 Desember 2024	28	<button>Lihat Manifest</button>
SBYUSO	B 3275 KJ	Yogyakarta	Surabaya	28 Mei 2024	1	<button>Lihat Manifest</button>
SBVSYJ	AB 3456 JD	Yogyakarta	Surabaya	28 Mei 2024	15	<button>Lihat Manifest</button>
MOLHSJ	AB 3759 AI	Yogyakarta	Magelang	28 Mei 2024	5	<button>Lihat Manifest</button>
SMGHSJ	AB 1394 GD	Yogyakarta	Semarang	28 Mei 2024	12	<button>Lihat Manifest</button>
JOGDOF	AB 1394 GD	Jakarta	Yogyakarta	28 Mei 2024	35	<button>Lihat Manifest</button>
JOGSJK	AB 1394 GD	Jakarta	Yogyakarta	28 Mei 2024	23	<button>Lihat Manifest</button>
JOCDOF	AB 1394 GD	Jakarta	Yogyakarta	28 Mei 2024	35	<button>Lihat Manifest</button>
JOGHSY	AB 1394 GD	Jakarta	Yogyakarta	28 Mei 2024	10	<button>Lihat Manifest</button>
JOGSTU	AB 1394 GD	Jakarta	Yogyakarta	28 Mei 2024	35	<button>Lihat Manifest</button>

1. Query menampilkan seluruh data jadwal.

```
● ● ●  
SELECT  
    j.id_jadwal, j.no_pol, tr.kota_asal, tr.kota_tujuan,  
    j.tanggal_berangkat, SUM(t.jml_kursi) AS jml_penumpang  
FROM  
    jadwal j  
JOIN  
    transaksi t USING(id_jadwal)  
JOIN  
    bus b USING(no_pol)  
JOIN  
    trayek tr USING(id_trayek)  
GROUP BY  
    j.id_jadwal;
```

- Detail Daftar Penumpang

The screenshot shows a web-based application interface for managing bus bookings. On the left, there's a sidebar with navigation links: Dashboard, Jadwal Bus, Trayek, Bus, Daftar Transaksi, and Daftar Penumpang. The main content area displays a summary of a trip and a detailed list of passengers.

Trip Summary:

ID Jadwal	:	JKTJOGAJD
No. Pol	:	AB 9403 C
Kelas	:	Executive
Kota Asal	:	YOGYAKARTA
Kota Tujuan	:	JAKARTA
Tanggal Berangkat	:	28 Mei 2024
Jam Berangkat	:	16:00

Passenger List:

ID Penumpang	Nama Penumpang	No. Telp	No. Kursi
PNP380	MUHAMMAD RAIHAN	0835-2353-1849	A1
PNP378	JOANNA ANDINI PRABANINGRUM PASYA	0856-3489-2849	A2
PNP381	NASIR WINANTO	0825-3478-9348	B1
PNP385	FAJAR PERMANA PUTRA	0856-1740-6840	B2
PNP387	DAMAR WAHYU HIDAYAT	0854-6830-9378	C1
PNP407	FAHREZA SYAHRUL RAMADHAN	0854-8593-1287	C2

At the bottom right of the passenger list, there is a page navigation bar showing pages 1, 2, 3, and more.

1. Query menampilkan data detail dari transaksi yang manifest yang dilihat.

```
SELECT
    j.id_jadwal, j.no_pol, j.kelas, j.tanggal_berangkat,
    j.jam_berangkat, tr.kota_asal, tr.kota_tujuan
FROM jadwal j
JOIN bus b USING(no_pol)
JOIN trayek tr USING(id_trayek)
WHERE id_jadwal='JKTJOGAJD';
```

2. Query menampilkan seluruh data penumpang dari jadwal tersebut. Query menampilkan seluruh data trayek.

```
SELECT
    p.id_penumpang, p.nama_penumpang, p.no_telp,
    dt.no_kursi
FROM penumpang p
JOIN transaksi t USING(id_penumpang)
JOIN detail_transaksi dt USING(id_transaksi)
WHERE id_jadwal = 'JKTJOGAJD'
ORDER BY dt.no_kursi ASC;
```

<p>Judul Aplikasi</p> <h1 style="text-align: center;">PERANCANGAN SISTEM TICKETING LAUNDRY BERBASIS APLIKASI</h1>	 <p>Fortuna Laundry</p>
<p>Dikembangkan Oleh</p> <ul style="list-style-type: none"> • 5220411388 – Alfian Nur Kusuma • 5220411396 – Muhammad Rizqi Raka Siwi • 5220411393 – Afriza Choirie Saputra • 5220411382 – Resky Wandoyo • 5220411383 – Difa Nur Sahara 	
<p>Deskripsi Singkat Instansi, Permasalahan, dan Solusi Aplikasi Yang Akan Dikembangkan [10 poin]</p> <p>Fortuna Laundry Pulowatu adalah sebuah usaha layanan mencuci pakaian yang berlokasi di Jamblangan, Purwobinangun, Pakem, Sleman. Sebagai salah satu penyedia jasa laundry di daerah tersebut, Fortuna Laundry Pulowatu berkomitmen untuk memberikan pelayanan terbaik kepada para pelanggannya. Namun, dalam operasional sehari-hari, terdapat beberapa permasalahan yang sering muncul, yaitu:</p> <ol style="list-style-type: none"> 1. Pakaian Hilang: Kasus di mana pakaian pelanggan tidak ditemukan atau hilang saat proses laundry. 2. Pakaian Tertukar: Terjadi kesalahan pengembalian pakaian, di mana pakaian satu pelanggan tercampur dengan pelanggan lainnya. 3. Pakaian Tidak Sesuai: Pelanggan menerima pakaian yang bukan milik mereka, yang menyebabkan ketidaknyamanan dan ketidakpuasan. <p>Untuk mengatasi permasalahan tersebut dan meningkatkan kualitas layanan, kami akan mengembangkan sebuah aplikasi berbasis ticketing. Aplikasi ini dirancang untuk:</p> <ol style="list-style-type: none"> 1. Mencatat Keluhan Pelanggan: Pelanggan dapat dengan mudah melaporkan setiap masalah yang mereka alami melalui aplikasi, seperti kehilangan pakaian, tertukar, atau menerima pakaian yang salah. 2. Penanganan Keluhan yang Efisien: Setiap keluhan yang masuk akan diterima sebagai tiket yang dapat ditelusuri dan diselesaikan satu per satu oleh tim layanan pelanggan, memastikan tidak ada keluhan yang terabaikan. 3. Transparansi Proses: Pelanggan dapat memantau status penyelesaian keluhan mereka secara real-time, memberikan rasa aman dan kepercayaan terhadap layanan Fortuna Laundry Pulowatu. 	

Dengan implementasi aplikasi ticketing ini, diharapkan dapat meminimalisir terjadinya permasalahan serupa di masa mendatang, serta meningkatkan kepuasan dan loyalitas pelanggan terhadap layanan Fortuna Laundry Pulowatu.

Pengguna Aplikasi [5 poin]

Aplikasi ini dirancang untuk melayani tiga jenis pengguna utama:

1. Pemilik Usaha Laundry:

- Pemantauan Operasional: Memantau secara real-time semua keluhan yang masuk dan status penyelesaiannya.
- Analisis Data: Mendapatkan laporan dan analisis data untuk memperbaiki layanan di masa depan.

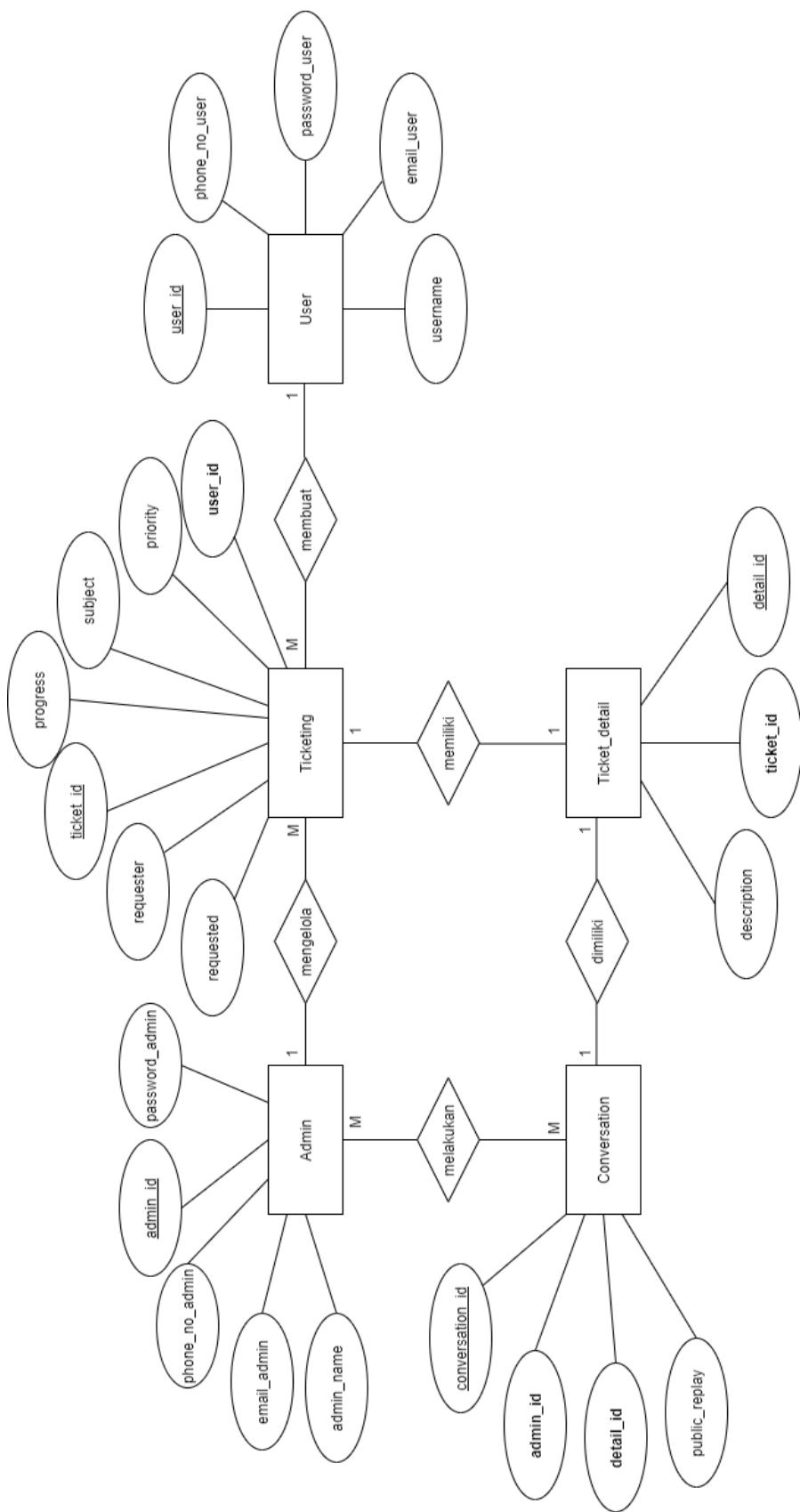
2. Pegawai:

- Penanganan Keluhan: Mengakses dan menangani keluhan pelanggan dengan efisien melalui sistem tiket.
- Pengelolaan Tugas: Mendapatkan tugas yang jelas dan terstruktur untuk menyelesaikan keluhan berdasarkan prioritas.

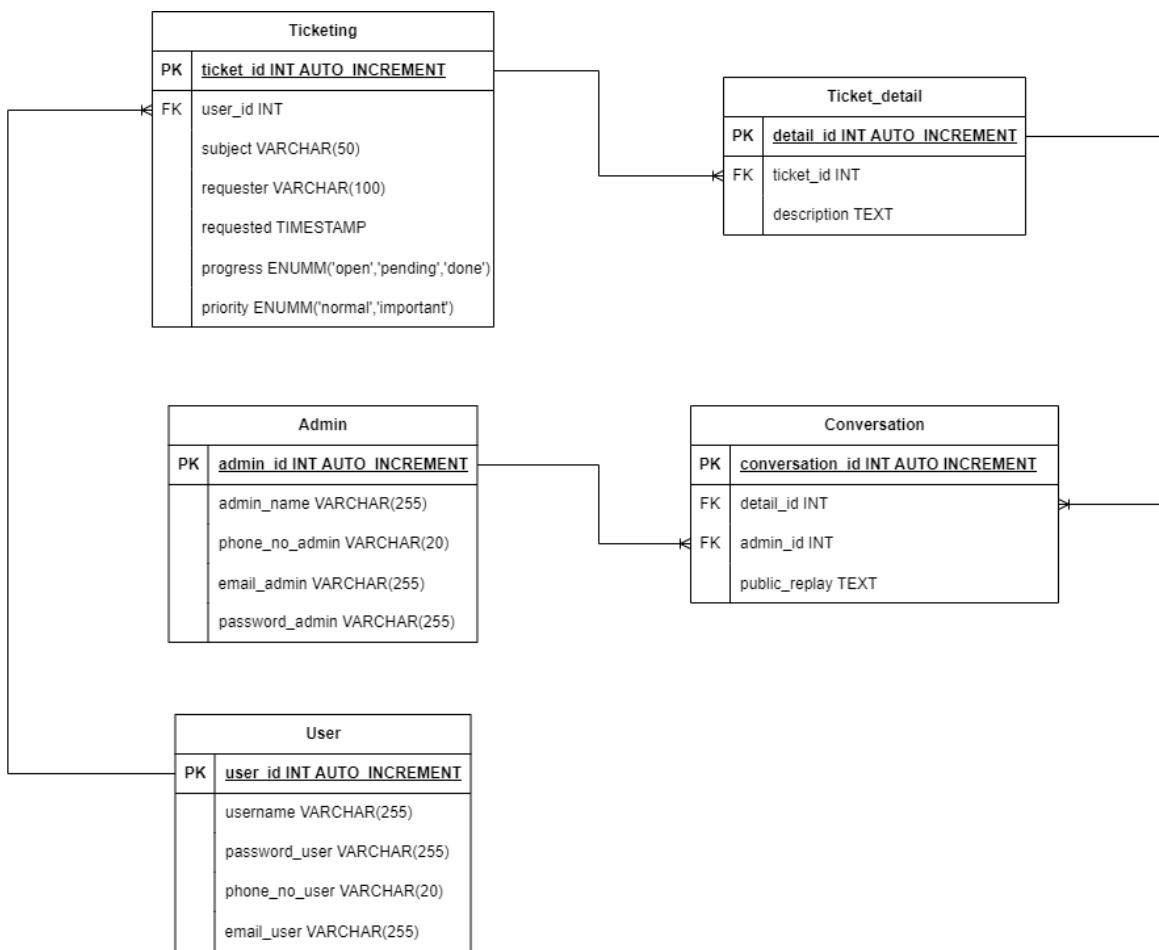
3. Pelanggan:

- Pelaporan Masalah: Mengajukan keluhan dengan mudah jika terjadi masalah seperti kehilangan atau pakaian tertukar.
- Pemantauan Status: Melihat status penyelesaian keluhan mereka secara real-time, memberikan transparansi dan rasa aman.

Desain Entity-Relationship Diagram (ERD) [10 poin]



Desain Relasi Basis Data [10 poin]



Query Data Definition Language (DDL) [25 poin]

1. Query membuat table Ticketing

```
CREATE TABLE Ticketing (
    ticket_id INT PRIMARY KEY AUTO_INCREMENT,
    user_id INT,
    subject VARCHAR(255),
    requester VARCHAR(100),
    requested TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    progress ENUM('open', 'pending', 'done'),
    priority ENUM('normal', 'important'),
    FOREIGN KEY (user_id) REFERENCES User(user_id)
);
```

2. Query membuat table Ticket_Detail

```
CREATE TABLE Ticket_Detail (
```

```
    detail_id INT PRIMARY KEY AUTO_INCREMENT,  
    ticket_id INT,  
    description TEXT,  
    FOREIGN KEY (ticket_id) REFERENCES Ticketing(ticket_id)  
);
```

3. Query membuat table Conversation

```
CREATE TABLE Conversation (  
    conversation_id INT PRIMARY KEY AUTO_INCREMENT,  
    detail_id INT,  
    admin_id INT,  
    public_reply TEXT,  
    FOREIGN KEY (detail_id) REFERENCES Ticket_Detail(detail_id),  
    FOREIGN KEY (admin_id) REFERENCES Admin(admin_id)  
);
```

4. Query membuat table Admin

```
CREATE TABLE Admin (  
    admin_id INT PRIMARY KEY AUTO_INCREMENT,  
    password_admin VARCHAR(255)  
    admin_name VARCHAR(255),  
    phone_no_admin VARCHAR(20),  
    email_admin VARCHAR(255)  
);
```

5. Query membuat table User

```
CREATE TABLE User (  
    user_id INT PRIMARY KEY AUTO_INCREMENT,  
    password_user VARCHAR(255)  
    username VARCHAR(255),  
    phone_no_user VARCHAR(20),  
    email_user VARCHAR(255)  
);
```

Desain Interface (Wireframe/Mockup) Beserta Query-nya [poin 40]

• UI Tickets

The wireframe shows a dark-themed application window titled 'Ticket'. On the left is a sidebar with a 'Fortuna Laundry' logo and icons for Home, Ticket (selected), Settings, and User. The main area is titled 'All Tickets' with a sub-count of '>21 Tickets'. It includes a header with 'New Ticket', 'Search', 'Filter', 'Sort by', and a more options menu. A table lists 15 tickets with columns for Subject, Requester, Requested Date, Progress, and Priority (color-coded). The table has a header row and a status filter row.

<input type="checkbox"/> Subject	Requester	Requested	Progress	Priority
Status : All				
<input type="checkbox"/> Baju Hilang	John Doe	24 January 2024	Open	Normal
<input type="checkbox"/> Baju Kotor	John Doe	26 Desember 2023	Pending	Important
<input type="checkbox"/> Terdapat Baju Lain	John Doe	18 Desember 2023	Done	Important
<input type="checkbox"/> Baju Rusak	John Doe	09 Desember 2023	Done	Normal
<input type="checkbox"/> Baju Hilang	John Doe	02 Desember 2023	Open	Important
<input type="checkbox"/> Baju Hilang	John Doe	28 November 2023	Done	Important
<input type="checkbox"/> Baju Kotor	John Doe	26 November 2023	Pending	Normal
<input type="checkbox"/> Terdapat Baju Lain	John Doe	17 November 2023	Open	Important
<input type="checkbox"/> Baju Rusak	John Doe	09 November 2023	Done	Important
<input type="checkbox"/> Baju Hilang	John Doe	21 October 2024	Pending	Normal
<input type="checkbox"/> Terdapat Baju Lain	John Doe	09 October 2023	Open	Important
<input type="checkbox"/> Baju Rusak	John Doe	26 October 2023	Done	Important

Query :

SELECT

```
subject AS "Subject",  
requester AS "Requester",  
requested AS "Requested",  
progress AS "Progress",  
priority AS "Priority"
```

FROM ticketing

ORDER BY requested DESC;

a) Search

```
SELECT * FROM tickets  
WHERE subject LIKE '%keyword%'  
OR requester LIKE '%keyword%'  
OR requested_date LIKE '%keyword%'  
OR progress LIKE '%keyword%'  
OR priority LIKE '%keyword%'
```

b) Filter

```
SELECT * FROM tickets  
WHERE progress = 'Open, Pending, Done';
```

```
SELECT * FROM tickets  
WHERE priority = 'Important,Normal';
```

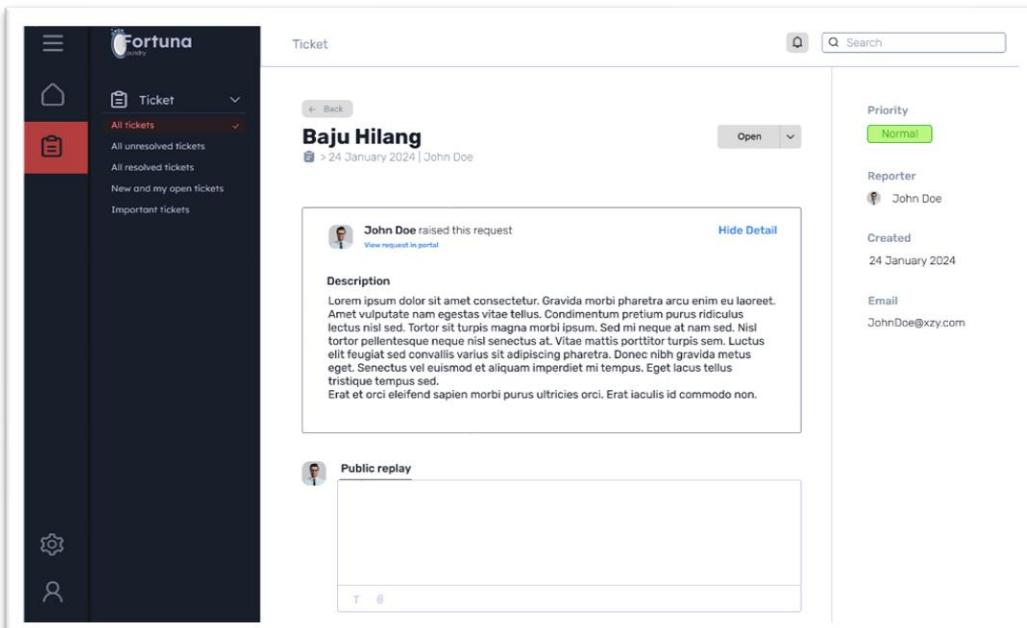
```
SELECT * FROM tickets  
WHERE requested_date BETWEEN '2023-12-01' AND '2024-01-31';
```

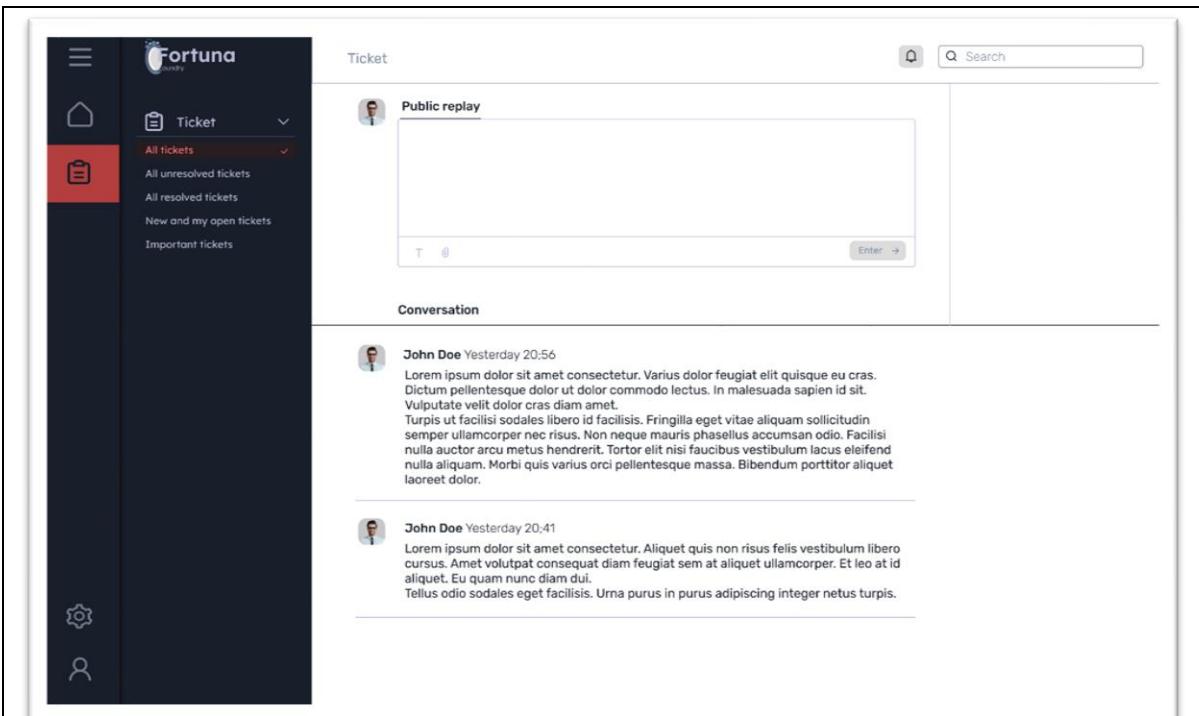
c) Sort by

```
SELECT * FROM tickets  
ORDER BY requested_date DESC;
```

```
SELECT * FROM tickets  
ORDER BY priority ASC;
```

- **UI Detail**





Query :

SELECT

```
t.ticket_id,
t.subject AS ticket_title,
td.description AS ticket_description,
t.progress AS ticket_status,
t.priority AS ticket_priority,
u.username AS reporter_name,
u.email_user AS reporter_email,
t.requested AS ticket_created_date
c.public_replay AS public_replay
```

FROM

```
Ticketing t
```

JOIN

```
Ticket_Detail td ON t.ticket_id = td.ticket_id
```

JOIN

```
User u ON t.user_id = u.user_id
```

JOIN

```
Conversation c ON td.detail_id = c.detail_id
```

WHERE

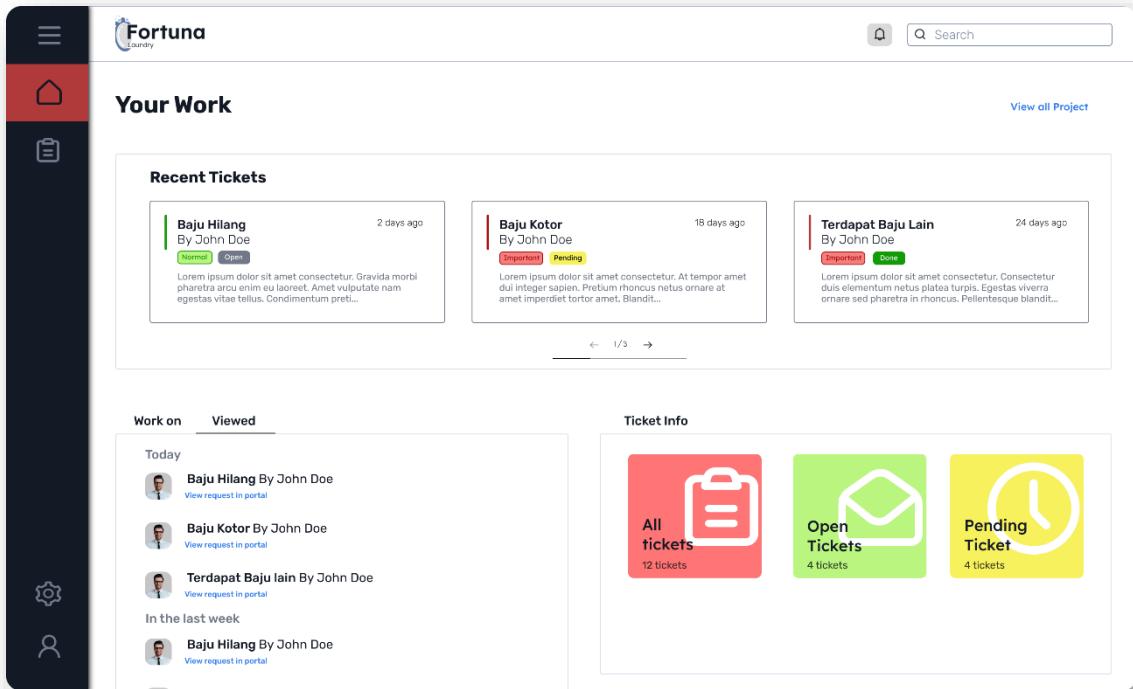
```
t.ticket_id = T001;
```

- **Insert data Chat:**

SET @public_replay = 'Ini adalah balasan publik untuk laporan Anda. Kami akan segera menindaklanjuti laporan ini.';

```
INSERT INTO Conversation (detail_id, admin_id, public_replay)
VALUES (@detail_id, 1, @public_replay);
```

• **UI Dashboard**



Query :

a) **Recent Tickets**

```
SELECT
    t.ticket_id,
    t.subject AS ticket_title,
    u.username AS reporter_name,
    t.priority AS ticket_priority,
    t.progress AS ticket_status,
    t.requested AS ticket_requested_date,
    td.description AS ticket_description
FROM
    Ticketing t
JOIN
    Ticket_Detail td ON t.ticket_id = td.ticket_id
JOIN
```

```

User u ON t.user_id = u.user_id
ORDER BY
    t.requested DESC
LIMIT 3;

b) Viewed
- Tickets worked on today
SELECT
    t.ticket_id,
    t.subject AS ticket_title,
    u.username AS reporter_name,
    t.requested AS ticket_requested_date
FROM
    Ticketing t
JOIN
    Ticket_Detail td ON t.ticket_id = td.ticket_id
JOIN
    User u ON t.user_id = u.user_id
WHERE
    DATE(t.requested) = CURDATE()
ORDER BY
    t.requested DESC;

- Tickets worked on in the last week
SELECT
    t.ticket_id,
    t.subject AS ticket_title,
    u.username AS reporter_name,
    t.requested AS ticket_requested_date
FROM
    Ticketing t
JOIN
    Ticket_Detail td ON t.ticket_id = td.ticket_id
JOIN
    User u ON t.user_id = u.user_id
WHERE
    t.requested >= CURDATE() - INTERVAL 1 WEEK
ORDER BY

```

```
t.requested DESC;

c) Ticket Info
- Total tickets
SELECT
    COUNT(*) AS total_tickets
FROM
    Ticketing;

Open tickets
SELECT
    COUNT(*) AS open_tickets
FROM
    Ticketing
WHERE
    progress = 'Open';

Pending tickets
SELECT
    COUNT(*) AS pending_tickets
FROM
    Ticketing
WHERE
    progress = 'Pending';
```

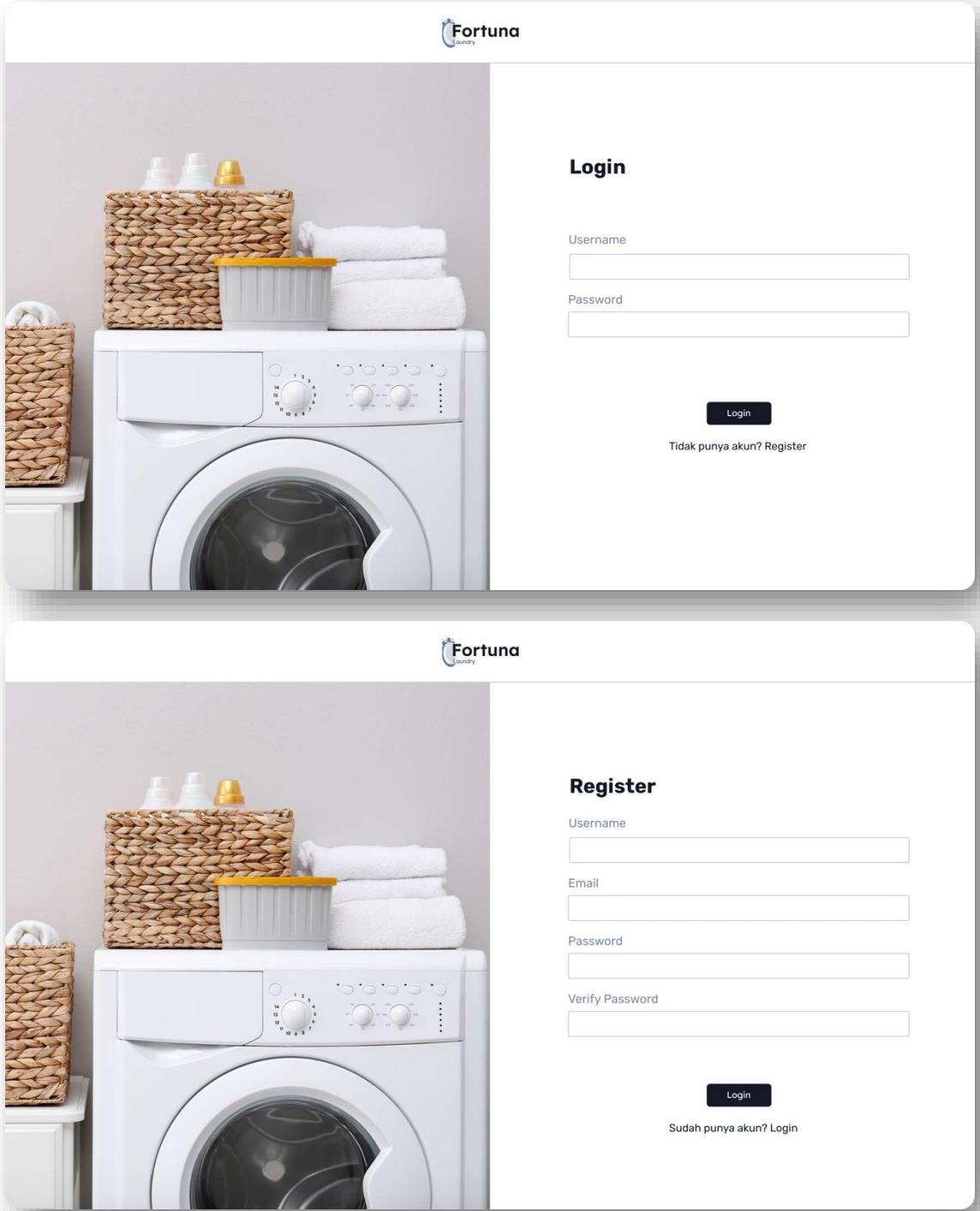
- UI Create Ticket

The screenshot shows the 'Create Ticket' dialog box from the Fortuna Laundry application. The dialog has a dark header bar with the 'Ticket' tab selected. Below the header, the title 'Create Ticket' is centered. The form contains several input fields: 'Subject' (empty), 'Reporter' (empty), 'Email' (empty), and a large 'Description' text area (empty). To the right of the form is a vertical sidebar titled 'Priority' with two columns of three priority levels each. The first column contains 'Normal' (green) and 'Important' (red) priority levels. The second column contains 'Normal' (green), 'Important' (red), and another 'Important' (red) priority level. At the bottom of the dialog are 'Create' and 'Cancel' buttons.

Query :

```
INSERT INTO tickets (subject, reporter, email, description, priority,  
requested_date, progress)  
VALUES ('[subject]', '[reporter]', '[email]', '[description]', '[priority]', NOW(),  
'Open');
```

- **UI Login dan Register**



The image displays two screenshots of a laundry application interface. Both screenshots feature a background image of a white front-loading washing machine with laundry supplies (towels, detergent bottles) on top.

Login Screen:

- The title "Login" is centered above the input fields.
- Two input fields are labeled "Username" and "Password".
- A "Login" button is located below the password field.
- A link "Tidak punya akun? Register" is at the bottom right.

Register Screen:

- The title "Register" is centered above the input fields.
- Four input fields are labeled "Username", "Email", "Password", and "Verify Password".
- A "Login" button is located below the verify password field.
- A link "Sudah punya akun? Login" is at the bottom right.

Query :

- **Memasukkan data pengguna baru**

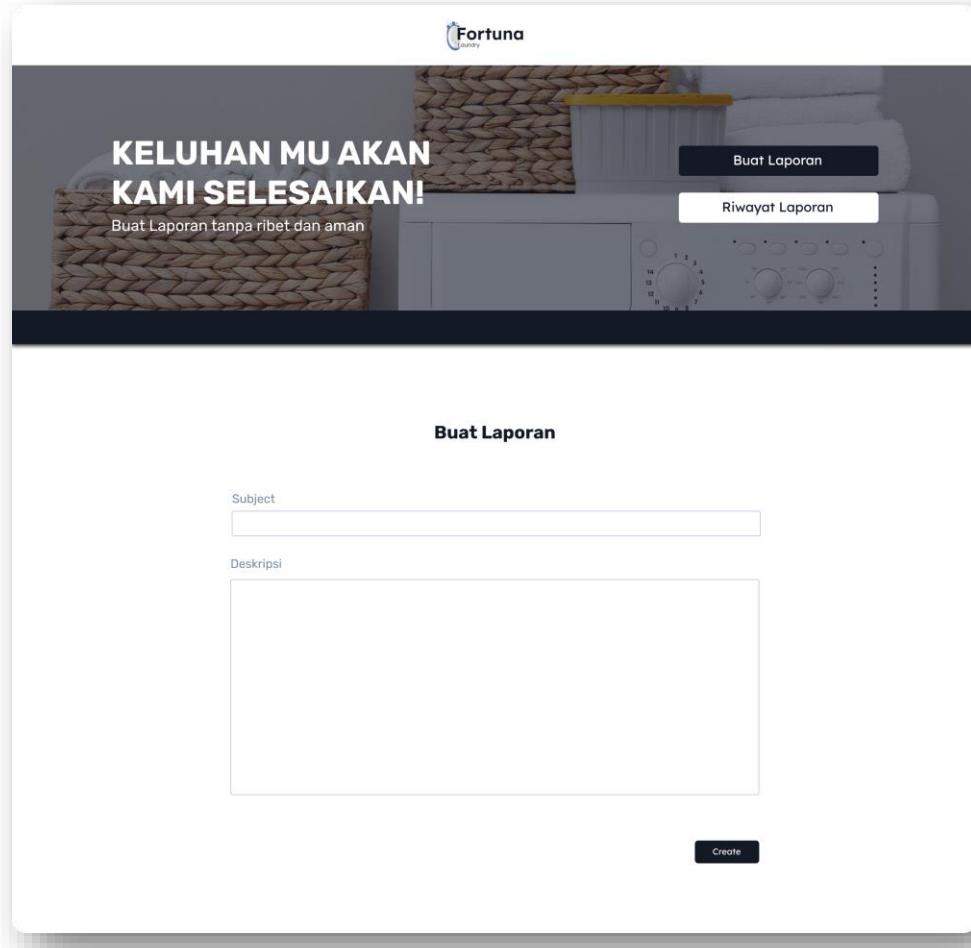
```
INSERT INTO User (username, phone_no_user, email_user, password_user)
VALUES ('John Doe', '081345276851', 'johndoe@xzy.com', '1234567890');
```

- **Verifikasi login pengguna**

```
SELECT * FROM User
```

```
WHERE username = 'John Doe' AND password_user = '1234567890';
```

- **UI Input Ticketing User**

**Query :**

```
GET user_id dari pengguna yang dimasukkan atau yang sudah ada
```

```
SET @user_id = (SELECT user_id FROM User WHERE email_user = @email);
```

a) Masukkan data ke dalam tabel Ticketing

```
INSERT INTO Ticketing (user_id, subject, requester, requested, progress, priority)
```

```
VALUES (@user_id, @subject, @name, NOW(), 'open', 'normal');
```

b) Dapatkan ticket_id dari tiket yang baru dimasukkan

```
SET @ticket_id = LAST_INSERT_ID();
```

Masukkan data ke dalam tabel Ticket_detail

```
INSERT INTO Ticket_detail (ticket_id, description)
```

```
VALUES (@ticket_id, @description);
```

- UI Detail Inputan User & Conversation**

The screenshot shows a web-based ticketing system interface. At the top, there's a header with the logo 'Fortuna Laundry'. Below the header, a sidebar on the left lists several tickets with their titles, posters, and creation dates:

- Baju Hilang (2 days ago)
- Baju Kotor (18 days ago)
- Terdapat Baju L... (24 days ago)
- Baju Rusak (30 days ago)

The main content area displays a detailed view of a ticket titled "Baju Hilang" created on 24 January 2024 by John Doe. The ticket details include:

- Description:** A large text area containing placeholder text about a lost item.
- Priority:** Normal
- Reporter:** John Doe
- Created:** 24 January 2024
- Email:** JohnDoe@xyz.com
- Progress:** Open

Below the ticket details, there's a "Public replay" section with a text input field and an "Enter" button. At the bottom, there's a "Conversation" section showing two messages from John Doe:

- Yesterday 20:56: Placeholder text about a lost item.
- Yesterday 20:41: Placeholder text about a lost item.

Query :

```
SELECT
```

```
t.ticket_id,  
t.subject AS ticket_title,  
td.description AS ticket_description,  
t.progress AS ticket_status,  
t.priority AS ticket_priority,  
u.username AS reporter_name,
```

```
        u.email_user AS reporter_email,  
        t.requested AS ticket_created_date  
        c.public_replay AS public_replay  
        c.internal_note AS internal_note  
  
    FROM  
        Ticketing t  
    JOIN  
        Ticket_Detail td ON t.ticket_id = td.ticket_id  
    JOIN  
        User u ON t.user_id = u.user_id  
    JOIN  
        Conversation c ON td.detail_id = c.detail_id  
  
    WHERE  
        t.ticket_id = T001;
```

- **Insert data Chat:**

```
SET @public_replay = 'Ini adalah balasan publik untuk laporan Anda. Kami akan  
segera menindaklanjuti laporan ini.';  
INSERT INTO Conversation (detail_id, admin_id, public_replay)  
VALUES (@detail_id, 1, @public_replay);
```

Judul Aplikasi	SPHYNX CARE: SISTEM PEMESANAN KAMAR CAT HOTEL ONLINE BERBASIS APLIKASI MOBILE	
Dikembangkan Oleh	<ul style="list-style-type: none"> • 5220411416–Muhammad Ali Pratama Putra • 5220411412– Ni Made Cahyani Dewi • 5220411414– Muchammad Hasbi Ashshiddiqi • 5220411418 – Zelia Latifah • 5220411419 – Muhammad Saiful Anwar 	
Deskripsi Singkat Instansi, Permasalahan, dan Solusi Aplikasi Yang Akan Dikembangkan [10 poin]	<p>Projek aplikasi Sphynx Care merupakan inspirasi dari sistem yang sudah berjalan di Chana Pet Shop & Penitipan Kucing (Cat Hotel). Chana cat house adalah sebuah usaha yang berawal dari kepedulian terhadap kesejahteraan kucing domestik. Berangkat dari pengalaman pribadi dalam memelihara kucing, pemiliknya merasa tergerak untuk memberikan solusi bagi para pecinta kucing yang membutuhkan tempat aman dan nyaman bagi hewan peliharaan mereka.</p>	
		
	<p>Gambar 1. Dokumentasi tim di instansi</p>	
	<p>Awal mula Chana cat house tidak bermula dari usaha penitipan kucing, melainkan dari kebutuhan akan kandang yang lebih besar bagi anabul (istilah untuk kucing dalam bahasa yang digunakan) yang dimiliki. Pemilik merasa kasihan dengan kandang-kandang kecil yang umumnya tersedia di pasaran, sehingga memutuskan untuk membuat kandang sendiri yang lebih besar dan nyaman bagi kucing peliharaannya. Inisiatif ini kemudian meluas menjadi ide untuk memproduksi kandang-kandang yang terjangkau bagi sesama pecinta kucing.</p>	

Ekspansi Usaha:

Dengan permintaan yang terus meningkat, pemilik memutuskan untuk memperluas usaha dengan menawarkan layanan penitipan kucing. Awalnya, penitipan dilakukan di rumah dengan kandang-kandang yang luas dan nyaman. Harga penitipan dijaga tetap terjangkau tanpa mengorbankan kualitas perawatan kucing, dengan komitmen untuk menjaga kebersihan dan kesejahteraan psikis kucing.

Manajemen Penitipan Kucing:

Manajemen penitipan kucing dilakukan secara manual, dengan pemasaran utama melalui media sosial. Admin toko dan pengelolaan penitipan dilakukan langsung oleh pemilik. Meskipun demikian, dalam situasi ramai, bantuan dari orang lain terkadang diperlukan untuk membantu dalam tugas-tugas sehari-hari seperti membersihkan kandang dan memberi makan.

Proses Penitipan:

Proses penitipan kucing dimulai dengan komunikasi melalui pesan langsung atau pesan WhatsApp untuk melakukan pemesanan. Syarat-syarat penitipan yang diberlakukan antara lain: kucing dalam keadaan sehat, bebas kutu dan jamur, tidak dalam kondisi birahi, dan tidak sedang hamil. Setelah syarat-syarat terpenuhi, pemilik kucing membawa hewan peliharaannya ke tempat penitipan.

Permasalahan:

Meskipun Chana cat house telah mengalami pertumbuhan yang pesat, pengelolaan penitipan kucing masih dilakukan secara manual dan terbatas pada pemasaran melalui media sosial. Hal ini menyebabkan keterbatasan dalam pengelolaan data pelanggan, proses pemesanan yang kurang efisien, dan kurangnya informasi yang tersedia bagi pemilik kucing yang ingin menggunakan layanan penitipan.

Solusi Aplikasi Yang Akan Dikembangkan:

1. **Pemesanan Online:** Pemilik kucing dapat melakukan pemesanan penitipan secara online melalui aplikasi, termasuk menentukan tanggal penitipan dan memberikan informasi tentang kucing mereka.
2. **Manajemen Data Pemesanan:** Aplikasi akan menyediakan sistem manajemen data pelanggan yang terintegrasi, termasuk informasi tentang kucing pelanggan, riwayat penitipan, dan preferensi perawatan kucing.
3. **Melihat checkup anabul:** Aplikasi dapat menampilkan riwayat checkup anabul yang dilakukan oleh admin

Pengguna Aplikasi [5 poin]**Pemilik Kucing/parents:**

- Pemilik kucing yang ingin menggunakan layanan penitipan kucing untuk hewan peliharaan mereka.
- Mereka menggunakan aplikasi untuk melakukan pemesanan penitipan, memberikan informasi tentang kucing mereka, mengatur jadwal penitipan, dan melakukan pembayaran.
- Penggunaan aplikasi ini akan memberikan mereka kemudahan dalam mengelola perawatan kucing mereka, serta memberikan akses cepat dan mudah ke informasi tentang aktivitas penitipan dan kebutuhan kucing mereka.

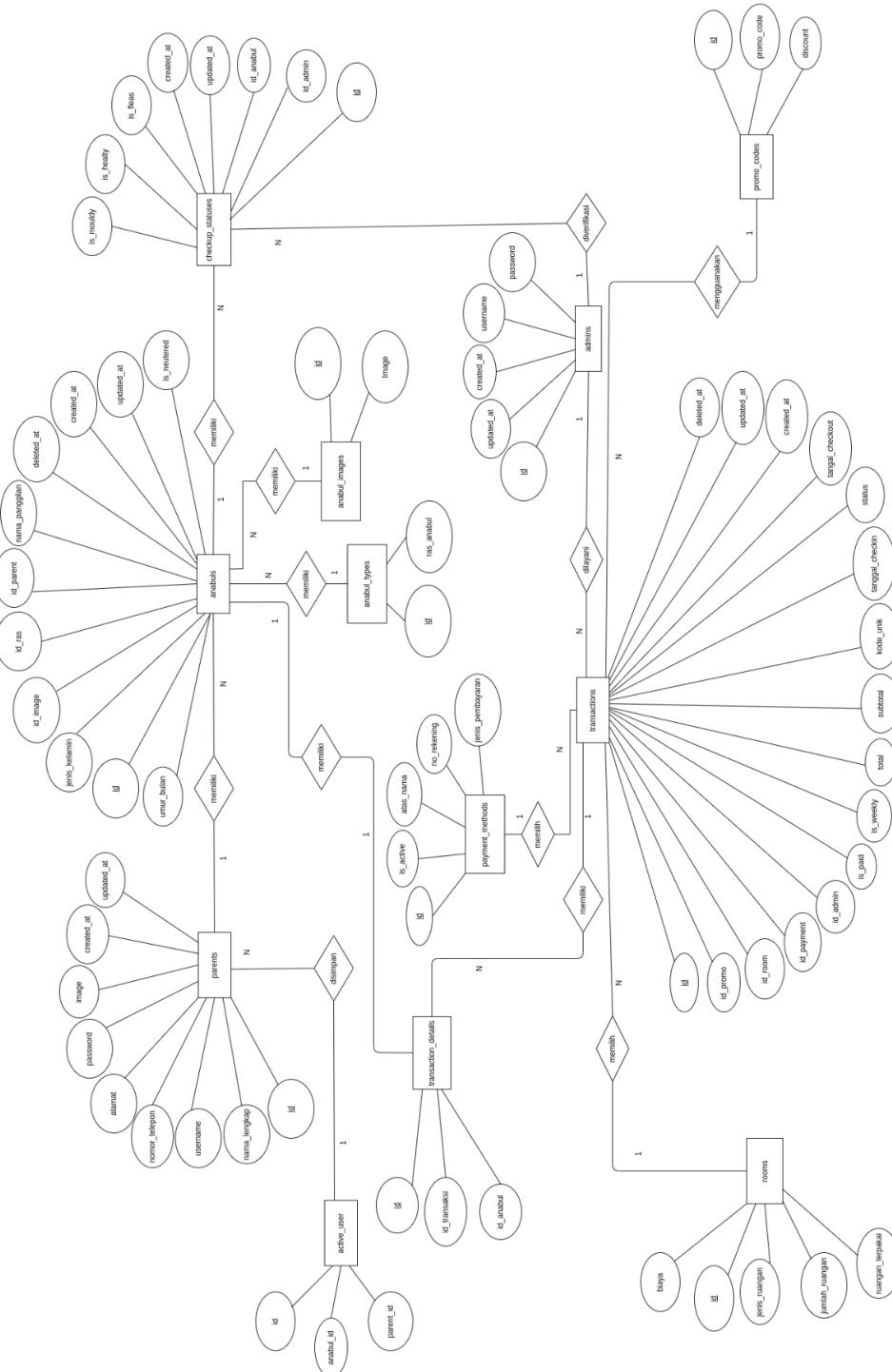
Pada laporan ini, kami hanya terbatas untuk merancang tampilan aplikasi untuk parents, akan tetapi dari segi basis data sudah siap dipakai untuk Admin. Jika laporan ini ditindaklanjuti, kami menyarankan untuk membuat dokumentasi query dan mockup untuk interface pengguna Admin.

Desain Entity-Relationship Diagram (ERD) [10 poin]

Link detail ERD dapat dilihat di:

https://drive.google.com/file/d/140bBjx6HGnj6z2RLI3yp_MPiMGpI2N88/view

ERD
SPHYNXCARE

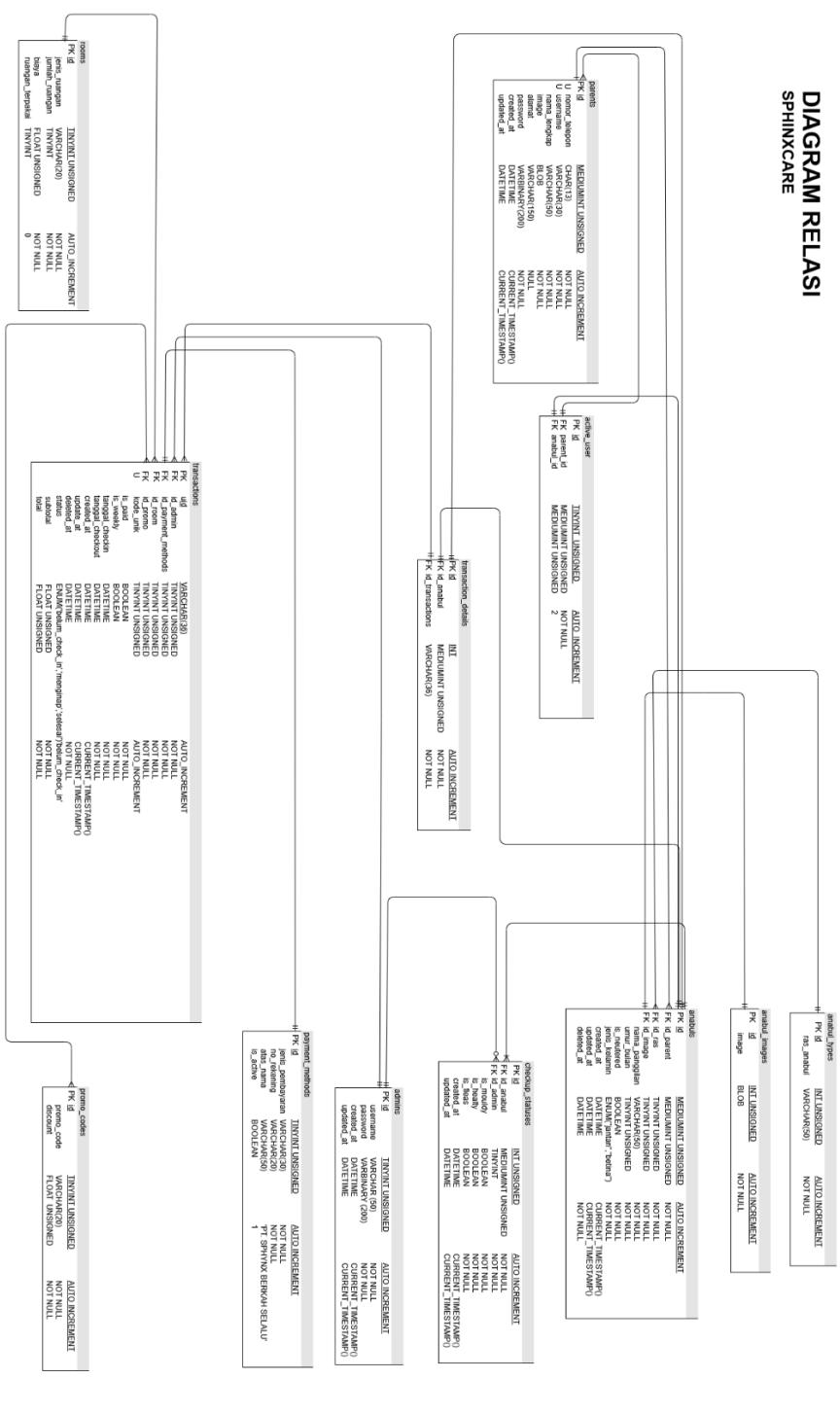


Desain Relasi Basis Data [10 poin]

Link detail desain dapat dilihat di:

<https://drive.google.com/file/d/1vZ3dAo1QuxMRdg40nEgEEEnzBHeCix8d3/view>

DIAGRAM RELASI SPHYNXCARE



Query Data Definition Language (DDL) [25 poin]

Berikut adalah struktur database dari aplikasi yang nampak pada client. Namun pada DDL ini tabel 'active_user' hanya berperan sebagai virtual state agar operasi query nanti dapat dilihat dengan lebih jelas. Namun pada implementasinya nanti, tabel tersebut akan melebur ke state management yang dimiliki oleh interface client dan tidak disimpan di server database.

1. Tabel active_user

Tabel ini berfungsi untuk handling dalam state management saja, bukan masuk dalam entitas pada sistem.

```
create table active_user
(
    id tinyint unsigned auto_increment primary key,
    parent_id mediumint unsigned not null,
    anabul_id mediumint unsigned default 2,
    constraint active_user_anabuls_id_fk
        foreign key (anabul_id) references anabuls (id),
    constraint active_user_parents_id_fk
        foreign key (parent_id) references parents (id)
);
```

2. Tabel admins

Tabel ini berfungsi untuk mengelola data admin yang bertugas.

```
create table admins
(
    id tinyint unsigned auto_increment primary key,
    username varchar(50) not null,
    password varbinary(200) not null,
    created_at datetime default current_timestamp(),
    updated_at datetime default current_timestamp() on update current_timestamp()
);
```

3. Tabel anabul_images

Tabel ini berfungsi untuk menyimpan avatar/foto dari anabul.

```
create table anabul_images
(
    id tinyint unsigned auto_increment primary key,
    image blob not null
);
```

4. Tabel anabul_types

Tabel ini berfungsi untuk menyimpan jenis/ras anabul.

```
create table anabul_types
(
    id tinyint unsigned auto_increment primary key,
    ras_anabul varchar(50) not null
);
```

5. Tabel anabuls

Tabel ini berfungsi untuk mengelola semua data anabul.

```
create table anabuls
(
    id mediumint unsigned auto_increment primary key,
    nama_panggilan varchar(50) not null,
    jenis_kelamin enum ('jantan', 'betina') not null,
    umur_bulan tinyint unsigned not null,
    is_neutered boolean not null,
    id_image tinyint unsigned not null,
    id_ras tinyint unsigned not null,
    id_parent mediumint unsigned not null,
    created_at datetime default current_timestamp(),
    updated_at datetime default current_timestamp() on update current_timestamp(),
    deleted_at datetime null,
    constraint anabuls_anabul_images_id_fk
        foreign key (id_image) references anabul_images (id),
    constraint anabuls_anabul_types_id_fk
        foreign key (id_ras) references anabul_types (id),
    constraint anabuls_parents_id_fk
        foreign key (id_parent) references parents (id)
        on delete cascade
);
```

6. Tabel checkup_statuses

Tabel ini berfungsi untuk mengelola data hasil checkup dari anabul.

```
create table checkup_statuses
```

```

(
    id int unsigned auto_increment primary key,
    is_mouldy boolean not null,
    is_healty boolean not null,
    is_fleas boolean not null,
    id_anabul mediumint unsigned not null,
    id_admin tinyint unsigned not null,
    created_at datetime default current_timestamp(),
    updated_at datetime default current_timestamp() on update current_timestamp(),
    constraint checkup_statuses_admins_id_fk
        foreign key (id_admin) references admins (id),
    constraint checkup_statuses_anabuls_id_fk
        foreign key (id_anabul) references anabuls (id)
        on delete cascade
);

```

7. Tabel parents

Tabel ini berfungsi untuk mengelola data parents atau akun.

```

create table parents
(
    id mediumint(9) unsigned auto_increment primary key,
    nama_lengkap varchar(50) not null,
    username varchar(30) not null,
    password varbinary(200) not null,
    nomor_telepon char(13) not null,
    image blob null,
    alamat varchar(150) null,
    created_at datetime default current_timestamp(),
    updated_at datetime default current_timestamp() on update current_timestamp(),
    constraint parents_pk
        unique (username, nomor_telepon)
);

```

8. Tabel payment_methods

Tabel ini berfungsi untuk menyimpan metode pembayaran.

```

create table payment_methods
(
    id tinyint unsigned auto_increment primary key,
    jenis_pembayaran varchar(30) not null,

```

```
    no_rekening varchar(20) not null,  
    atas_nama varchar(50) default 'PT. SPHYNX BERKAH SELALU',  
    is_active boolean default 1  
);
```

9. Tabel promo_codes

Tabel ini berfungsi untuk menyimpan kode promo dan diskon.

```
create table promo_codes  
(  
    id tinyint unsigned auto_increment primary key,  
    promo_code varchar(20) null,  
    discount float unsigned not null  
);
```

10. Tabel rooms

Tabel ini berfungsi untuk mengelola data ruangan anabul yang tersedia.

```
create table rooms  
(  
    id tinyint unsigned auto_increment primary key,  
    jenis_ruangan varchar(20) not null,  
    biaya float unsigned not null,  
    jumlah_ruangan tinyint not null,  
    ruangan_terpakai tinyint default 0,  
    constraint rooms_pk_2  
        unique (jenis_ruangan)  
);
```

11. Tabel transaction_details

Tabel ini berfungsi untuk mengelola detail transaksi.

```
create table transaction_details  
(  
    id int unsigned auto_increment primary key,  
    id_anabul mediumint unsigned not null,  
    id_transaksi varchar(36) not null,  
    constraint transaction_details_anabuls_id_fk  
        foreign key (id_anabul) references anabuls (id)  
);
```

12. Tabel transactions

Tabel ini berfungsi untuk mengelola semua data transaksi.

```
create table transactions
(
    uid varchar(36) default uuid() primary key,
    tanggal_checkin datetime not null,
    tanggal_checkout datetime not null,
    kode_unik tinyint unsigned auto_increment,
    is_paid boolean default 0,
    is_weekly boolean default 0,
    subtotal float unsigned default 0,
    total float unsigned default 0,
    status enum ('belum_check_in', 'menginap', 'selesai') default 'belum_check_in',
    id_promo tinyint unsigned default 1,
    id_payment tinyint unsigned not null,
    id_admin tinyint unsigned not null,
    id_room tinyint unsigned not null,
    created_at datetime default current_timestamp(),
    updated_at datetime default current_timestamp() on update current_timestamp(),
    deleted_at datetime null,
    constraint transactions_pk_2
        unique (kode_unik),
    constraint transactions_admins_id_fk
        foreign key (id_admin) references admins (id),
    constraint transactions_payment_methods_id_fk
        foreign key (id_payment) references payment_methods (id),
    constraint transactions_promo_codes_id_fk
        foreign key (id_promo) references promo_codes (id),
    constraint transactions_rooms_id_fk
        foreign key (id_room) references rooms (id)
);
```

13. Trigger ins_pass_admins

Trigger ini aktif ketika melakukan operasi INSERT pada tabel admins. Bertujuan untuk enkripsi data pada kolom password.

```
create definer = root@localhost trigger ins_pass_admins
before insert
on admins
for each row
begin
```

```
    SET NEW.password = aes_encrypt(NEW.password, 'SPHYNX');
end;
```

14. Trigger ins_pass_parents

Trigger ini aktif ketika melakukan operasi INSERT pada tabel parents. Bertujuan untuk enkripsi data pada kolom password.

```
create definer = root@localhost trigger ins_pass_parent
before insert
on parents
for each row
begin
    SET NEW.password = AES_ENCRYPT(NEW.password, 'SPHYNX');
end;
```

15. Trigger upt_pass_parents

Trigger ini aktif ketika melakukan operasi UPDATE pada tabel parents. Bertujuan untuk enkripsi data pada kolom password.

```
create definer = root@localhost trigger upt_pass_parent
before update
on parents
for each row
begin
    SET NEW.password = AES_ENCRYPT(NEW.password, 'SPHYNX');
end;
```

16. Trigger ins_anabul_neutered

Trigger ini aktif ketika melakukan operasi INSERT pada tabel anabuls. Bertujuan untuk validasi data pada kolom is_neutered dengan syarat jenis kelamin jantan lebih dari 9 bulan dan betina lebih dari 6 bulan wajib sterilisasi.

```
create definer = root@localhost trigger ins_anabul_neutered
before insert
on anabuls
for each row
begin
    IF (NEW.jenis_kelamin='jantan' AND NEW.umur_bulan > 9) OR
(NEW.jenis_kelamin='betina' AND NEW.umur_bulan > 6) THEN
        SET NEW.is_neutered = 1;
end if;
```

```
end;
```

17. Trigger upt_anabul_neutered

Trigger ini aktif ketika melakukan operasi UPDATE pada tabel anabuls. Bertujuan untuk validasi data pada kolom is_neutered dengan syarat jenis kelamin jantan lebih dari 9 bulan dan betina lebih dari 6 bulan wajib sterilisasi.

```
create definer = root@localhost trigger upt_anabul_neutered
before update
on anabuls
for each row
begin
IF (NEW.jenis_kelamin='jantan' AND NEW.umur_bulan > 9) OR
(NEW.jenis_kelamin='betina' AND NEW.umur_bulan > 6) THEN
SET NEW.is_neutered = 1;
end if;
end;
```

18. Trigger ins_anabul_detail

Trigger ini aktif ketika melakukan operasi INSERT pada tabel transaction_details. Bertujuan untuk validasi data pada kolom subtotal dari tabel transactions dengan syarat jika melebihi dari 5 anabul akan dikenakan denda 25rb/anabul tambahan.

```
create definer = root@localhost trigger ins_anabul_detail
before insert
on transaction_details
for each row
begin
IF (SELECT COUNT(*)
FROM transaction_details
WHERE id_transaksi=NEW.id_transaksi) > 4 THEN
UPDATE transactions
SET subtotal=subtotal+((SELECT COUNT(*)
FROM transaction_details
WHERE id_transaksi=NEW.id_transaksi) - 4) * 25000
WHERE uid=NEW.id_transaksi;
end if;
end;
```

19. Trigger ins_price_transaction

Trigger ini aktif ketika melakukan operasi INSERT pada tabel transactions. Bertujuan untuk kalkulasi data otomatis seperti, menghitung subtotal, melakukan validasi promo kode, memberikan diskon, harga mingguan dan menghitung semua total. Bertujuan juga untuk validasi data ruangan yang tersedia pada tabel rooms, jika ruangan penuh akan memberikan pesan error pada SQL

```
create definer = root@localhost trigger ins_price_transaction
before insert
on transactions
for each row
begin
    DECLARE subtotal FLOAT;
    DECLARE grand_total FLOAT;
    DECLARE potongan FLOAT;
    IF (SELECT jumlah_ruangan = ruangan_terpakai
        FROM rooms
        WHERE id=NEW.id_room) > 0 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Invalid: Ruangan penuh';
    end if;
    SELECT TIMESTAMPDIFF(DAY, NEW.tanggal_checkin, NEW.tanggal_checkout) * biaya
    INTO subtotal
    FROM rooms
    WHERE id=NEW.id_room;
    SET NEW.subtotal = subtotal;
    SET grand_total = subtotal;
    IF NEW.is_weekly > 0 THEN
        SET grand_total = grand_total - grand_total * 15 / 100;
    end if;
    IF NEW.id_promo > 1 THEN
        SELECT discount INTO potongan
        FROM promo_codes WHERE id = NEW.id_promo;
        SET grand_total = grand_total - grand_total * potongan / 100;
    end if;
    SET NEW.total = grand_total + NEW.kode_unik;
end;
```

20. Trigger upt_price_transaction

Trigger ini aktif ketika melakukan operasi UPDATE pada tabel transactions. Bertujuan untuk kalkulasi data otomatis seperti, menghitung subtotal, melakukan validasi promo kode, memberikan diskon, harga mingguan dan menghitung semua total. Bertujuan juga untuk validasi data ruangan yang tersedia pada tabel rooms, jika ruangan penuh akan memberikan pesan error pada SQL

```
create definer = root@localhost trigger upt_price_transaction
before update
on transactions
for each row
begin
    DECLARE grand_total FLOAT;
    DECLARE potongan FLOAT;
    SET grand_total = NEW.subtotal;
    IF NEW.is_weekly > 0 THEN
        SET grand_total = grand_total - grand_total * 15 / 100;
    end if;
    IF NEW.id_promo > 1 THEN
        SELECT discount INTO potongan
        FROM promo_codes WHERE id = NEW.id_promo;
        SET grand_total = grand_total - grand_total * potongan / 100;
    end if;
    SET NEW.total = grand_total + NEW.kode_unik;
end;
```

21. Trigger ins_room_transaction

Trigger ini aktif ketika melakukan operasi INSERT pada tabel transactions. Bertujuan untuk sinkronisasi data kolom ruangan terpakai pada tabel rooms.

```
create definer = root@localhost trigger ins_room_transaction
after insert
on transactions
for each row
begin
    UPDATE rooms SET ruangan_terpakai =
    SELECT COUNT(*)
    FROM transactions
    WHERE NOW() BETWEEN tanggal_checkin AND tanggal_checkout)
    WHERE rooms.id=NEW.id_room;
```

```
end;
```

22. Trigger upt_room_transaction

Trigger ini aktif ketika melakukan operasi UPDATE pada tabel transactions. Bertujuan untuk sinkronisasi data kolom ruangan terpakai pada tabel rooms.

```
create definer = root@localhost trigger upt_room_transaction
  after update
    on transactions
      for each row
begin
  UPDATE rooms SET ruangan_terpakai=(
  SELECT COUNT(*)
  FROM transactions
  WHERE NOW() BETWEEN tanggal_checkin AND tanggal_checkout)
  WHERE rooms.id=NEW.id_room;
end;
```

23. Function check_login_parents

Fungsi ini bertujuan untuk enkapsulasi dari operasi autentikasi pada aplikasi. Fungsi ini akan melakukan return 1 jika autentikasi berhasil, sebaliknya akan melakukan return 0.

```
create
  definer = root@localhost function check_login_parents(auth_username varchar(255), auth_password
  varchar(255)) returns tinyint(1)
    deterministic
BEGIN
  DECLARE is_valid BOOLEAN DEFAULT 0;

  IF (SELECT COUNT(*) FROM parents WHERE
    (username = auth_username OR nomor_telepon = auth_username)
    AND password = AES_ENCRYPT(auth_password, 'SPHYNX')) > 0 THEN
    UPDATE active_user SET parent_id=(SELECT id FROM parents WHERE
      (username = auth_username OR nomor_telepon = auth_username)
      AND password = AES_ENCRYPT(auth_password, 'SPHYNX'))
    WHERE id=2;
    SET is_valid = 1;
  END IF;
  RETURN is_valid;
END;
```

24. Function check_available_username

Fungsi ini bertujuan untuk mengecek username pada tabel parents yang valid. Karena username bersifat unik. Fungsi ini hanya membantu dalam user experience di aplikasi.

```
create
    definer = root@localhost function check_available_username(username varchar(100)) returns tinyint(1)
deterministic
BEGIN
    DECLARE is_valid BOOLEAN DEFAULT 1;
    IF (SELECT COUNT(*) FROM parents WHERE
        parents.username = username) > 0 THEN
        SET is_valid = 0;
    END IF;
    RETURN is_valid;
END;
```

25. Function *change_parent_password*

Fungsi ini bertujuan untuk mengubah sekaligus melakukan validasi password pada tabel parents. Fungsi ini sudah termasuk melakukan enkripsi data.

```
create
    definer = root@localhost function change_parent_password(id_parent mediumint, old_pass
varchar(100),
new_pass varchar(100),
new_conf_pass varchar(100)) returns tinyint(1)
deterministic
BEGIN
    DECLARE is_valid BOOLEAN DEFAULT 0;
    IF (new_pass = new_conf_pass) AND
    (SELECT COUNT(*) FROM parents WHERE
        (id = id_parent AND password = AES_ENCRYPT(old_pass, 'SPHYNX')) > 0) THEN
        UPDATE parents SET password = new_pass WHERE id = id_parent;
        SET is_valid = 1;
    END IF;

    RETURN is_valid;
END;
```

26. *view_cat_checkuplists*

View ini bertujuan untuk menampilkan data checkup pada anabul yang datanya sedang dibuka.

```

CREATE OR REPLACE VIEW view_cat_checkuplists AS
    SELECT DATE_FORMAT(cs.created_at, '%d %M %Y') tanggal_checkup,
        cs.is_mouldy,
        cs.is_fleas,
        cs.is_healty
    FROM checkup_statuses cs
    INNER JOIN db_sphynxcare.anabuls a ON cs.id_anabul = a.id
    INNER JOIN active_user au ON a.id = au.anabul_id
    WHERE au.id=2
    ORDER BY cs.created_at DESC;

```

27. view_cat_detail

View ini bertujuan untuk menampilkan data detail anabul yang datanya sedang dibuka.

```

CREATE OR REPLACE VIEW view_cat_detail AS
SELECT ai.image,
    a.nama_panggilan,
    CONCAT(umur_bulan + TIMESTAMPDIFF(MONTH, NOW(), a.created_at), ' bulan') umur_bulan,
    a.jenis_kelamin,
    aty.ras_anabul,
    IF(a.is_neutered, 'Sudah Steril', 'Belum Steril') is_neutered
FROM anabuls a
    INNER JOIN anabul_images ai ON a.id_image = ai.id
    INNER JOIN anabul_types aty ON a.id_ras = aty.id
    INNER JOIN active_user au ON a.id = au.anabul_id
WHERE au.id=2;

```

28. view_cat_transactions

View ini bertujuan untuk menampilkan data transaksi yang berkaitan dengan anabul yang datanya sedang dibuka.

```

CREATE OR REPLACE VIEW view_cat_transactions AS
SELECT
    t.uid,
    CONCAT(DAY(t.tanggal_checkin), ' ', MONTHNAME(t.tanggal_checkin)) AS `check_in`,
    CONCAT(DAY(t.tanggal_checkout), ' ', MONTHNAME(t.tanggal_checkout)) AS `check_out`,
    CONCAT(TIMESTAMPDIFF(DAY, t.tanggal_checkin, t.tanggal_checkout) + 1, ' Hari '),
    TIMESTAMPDIFF(DAY, t.tanggal_checkin, t.tanggal_checkout), ' Malam') AS `durasi`,
    COUNT(id_parent) AS `jumlah_anabul`,
    r.jenis_ruangan
FROM transactions t

```

```

INNER JOIN transaction_details td ON td.id_transaksi=t.uid
INNER JOIN anabuls a ON td.id_anabul = a.id
INNER JOIN parents p ON a.id_parent = p.id
INNER JOIN rooms r ON t.id_room = r.id
INNER JOIN active_user au ON a.id = au.anabul_id
WHERE au.id=2
GROUP BY a.id_parent, t.created_at
ORDER BY t.created_at DESC;

```

29. view_catlists_all

View ini menampilkan seluruh data anabul yang dimiliki oleh parents yang datanya sedang dibuka.

```

CREATE OR REPLACE VIEW view_catlists_all AS
SELECT a.id,
a.nama_panggilan,
CONCAT(a.umur_bulan, ' bulan') AS 'umur_bulan',
a.jenis_kelamin,
aty.ras_anabul,
ai.image
FROM anabuls a
INNER JOIN anabul_images ai ON a.id_image=ai.id
INNER JOIN anabul_types aty ON a.id_ras=aty.id
INNER JOIN parents p ON a.id_parent=p.id
INNER JOIN active_user au ON p.id = au.parent_id
WHERE au.id=2
ORDER BY a.created_at DESC;

```

30. view_parent_account

View ini menampilkan data parents yang datanya sedang dibuka.

```

CREATE OR REPLACE VIEW view_parent_account AS
SELECT p.nama_lengkap, p.image
FROM parents p
INNER JOIN active_user au
ON p.id = au.parent_id
WHERE au.id=2;

```

31. view_paymentmethods_all

View ini menampilkan data metode pembayaran yang tersedia.

```

CREATE OR REPLACE VIEW view_paymentmethods_all AS

```

```
SELECT jenis_pembayaran,  
       no_rekening,  
       atas_nama  
  FROM payment_methods  
 WHERE is_active=1;
```

32. view_transaction_catlists

View ini menampilkan seluruh data anabul yang berhubungan dengan transaksi yang datanya sedang dibuka.

```
CREATE OR REPLACE VIEW view_transaction_catlists AS  
SELECT vca.nama_panggilan,  
       vca.umur_bulan,  
       vca.jenis_kelamin,  
       vca.ras_anabul,  
       vca.image  
  FROM transaction_details td  
 INNER JOIN transactions t ON td.id_transaksi=t.uid  
 INNER JOIN anabuls a ON td.id_anabul = a.id  
 INNER JOIN view_catlists_all vca ON vca.id=a.id  
 INNER JOIN view_transaction_detail vtd ON t.uid = vtd.uid;
```

33. view_transaction_detail

View ini menampilkan detail transaksi yang datanya sedang dibuka.

```
CREATE OR REPLACE VIEW view_transaction_detail AS  
SELECT  
       t.uid,  
       CONCAT(TIMESTAMPDIFF(DAY, t.tanggal_checkin, t.tanggal_checkout) + 1, ' Hari ',  
              TIMESTAMPDIFF(DAY, t.tanggal_checkin, t.tanggal_checkout), ' Malam') AS `durasi`,  
       t.tanggal_checkin,  
       t.tanggal_checkout,  
       r.jenis_ruangan,  
       t.subtotal,  
       pc.discount,  
       pc.promo_code,  
       t.total,  
       t.status  
  FROM transactions t  
 INNER JOIN rooms r ON t.id_room = r.id  
 INNER JOIN promo_codes pc ON t.id_promo = pc.id
```

```

INNER JOIN transaction_details td ON td.id_transaksi=t.uid
INNER JOIN anabuls a ON td.id_anabul = a.id
INNER JOIN parents p ON a.id_parent = p.id
INNER JOIN active_user au ON a.id = au.anabul_id
WHERE au.id=2
GROUP BY t.uid, t.created_at
ORDER BY t.created_at DESC
LIMIT 1;

```

34. view_transaction_success

View ini menampilkan detail transaksi sukses yang datanya sedang dibuka.

```

CREATE OR REPLACE VIEW view_transaction_success AS
SELECT *,
CONCAT(
(SELECT COUNT(*)
FROM view_transaction_catlists),
' ekor'
) jumlah_anabul
FROM view_transaction_detail;

```

35. view_transaction_all

View ini menampilkan seluruh data transaksi yang dimiliki oleh parents yang datanya sedang dibuka.

```

CREATE OR REPLACE VIEW view_transactions_all AS
SELECT
t.uid,
CONCAT(DAY(t.tanggal_checkin), ' ', MONTHNAME(t.tanggal_checkin)) AS `check_in`,
CONCAT(DAY(t.tanggal_checkout), ' ', MONTHNAME(t.tanggal_checkout)) AS `check_out`,
CONCAT(TIMESTAMPDIFF(DAY, t.tanggal_checkin, t.tanggal_checkout) + 1, ' Hari '),
TIMESTAMPDIFF(DAY, t.tanggal_checkin, t.tanggal_checkout), ' Malam') AS `durasi`,
COUNT(id_parent) AS `jumlah_anabul`,
r.jenis_ruangan,
t.status
FROM transactions t
INNER JOIN transaction_details td ON td.id_transaksi=t.uid
INNER JOIN anabuls a ON td.id_anabul = a.id
INNER JOIN parents p ON a.id_parent = p.id

```

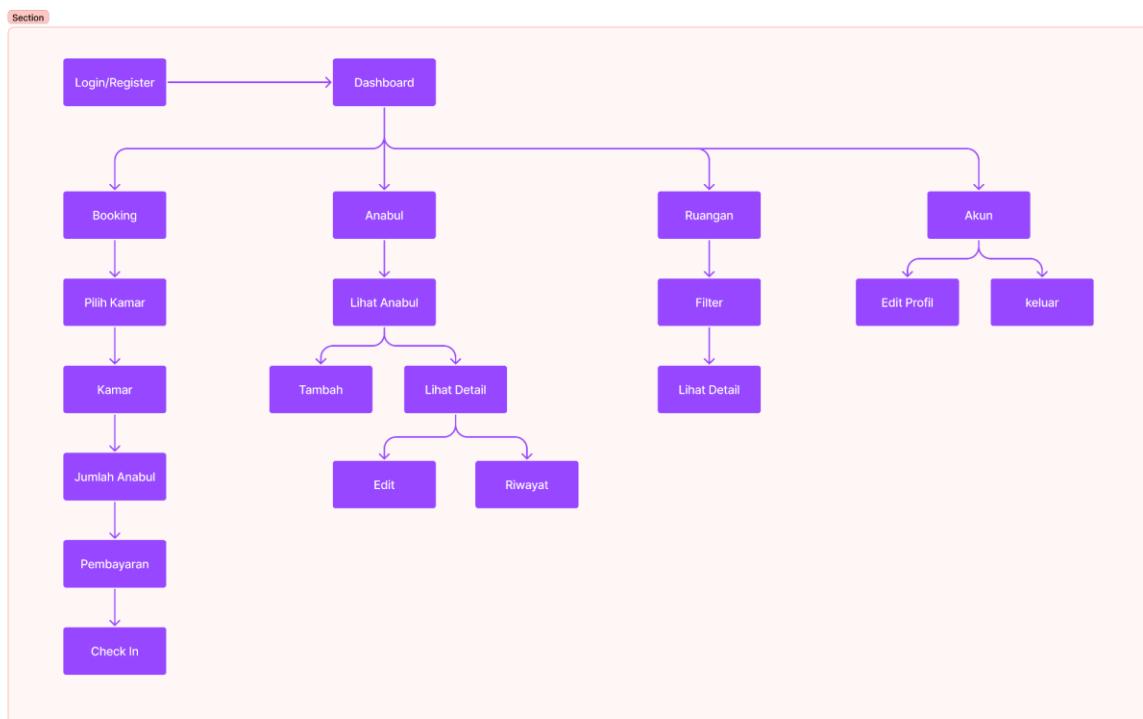
```

INNER JOIN rooms r ON t.id_room = r.id
INNER JOIN active_user au ON p.id = au.parent_id
WHERE au.id=2
GROUP BY a.id_parent, t.created_at
ORDER BY t.created_at DESC;

```

Desain Interface (Wireframe/Mockup) Beserta Query-nya [poin 40]

Userflow diagram:

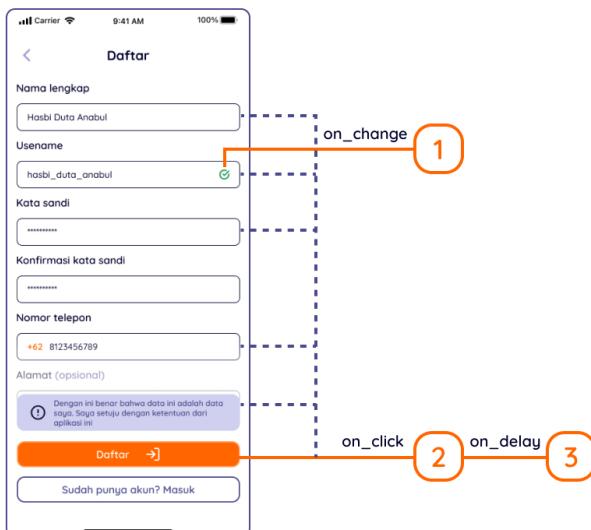


Detail prototype dari interface dapat dilihat di:

<https://www.figma.com/proto/eLBkMrrhV1VuMvkCe0odQi/Sphynx-Care?node-id=22-558&t=g9g3L4MeMGYk3tbk-0&scaling=scale-down&page-id=0%3A1&starting-point-node-id=22%3A558>



Nomor	Aksi	Query
1	Klik tombol	<code>SELECT check_login_parents('hasbi_duta_anabul','hasbi123');</code>

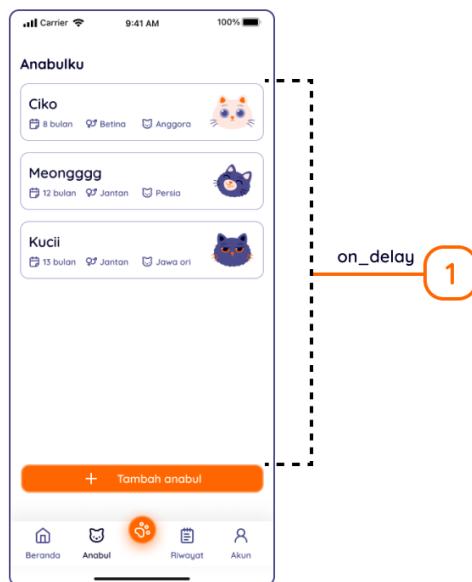


Nomor	Aksi	Query
1	Perubahan username	<code>SELECT check_available_username('hasbi_duta_anabul');</code>

2	Klik tombol	<pre>INSERT INTO parents(nama_lengkap, username, password, nomor_telepon) VALUES ('hasbi duta anabul', 'hasbi_duta_anabul', 'pass_hasbi','0812345678913');</pre>
3	Menunggu waktu berhasil	<pre>SELECT check_login_parents('0812345678913','pass_hasbi');</pre>

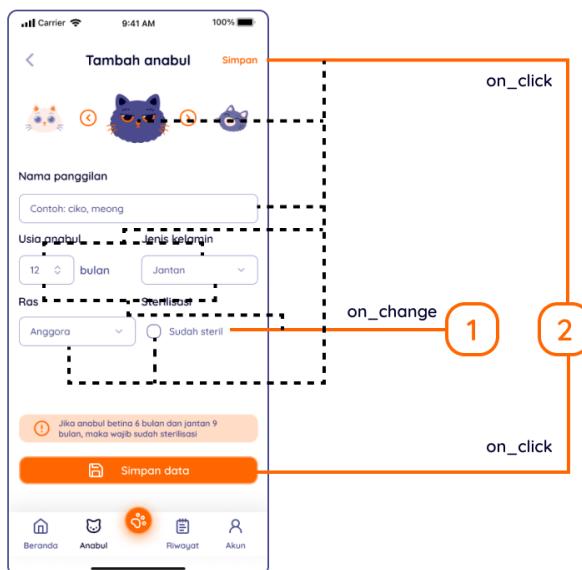


Nomor	Aksi	Query
1	Menunggu	<pre>SELECT * FROM view_parent_account;</pre>
2	Menunggu	<pre>SELECT * FROM view_catlists_all LIMIT 2;</pre>
3	Menunggu	<pre>SELECT * FROM view_transactions_all LIMIT 2;</pre>



on_delay 1

Nomor	Aksi	Query
1	Menunggu	<code>SELECT * FROM view_catlists_all;</code>



on_click 1

on_change 2

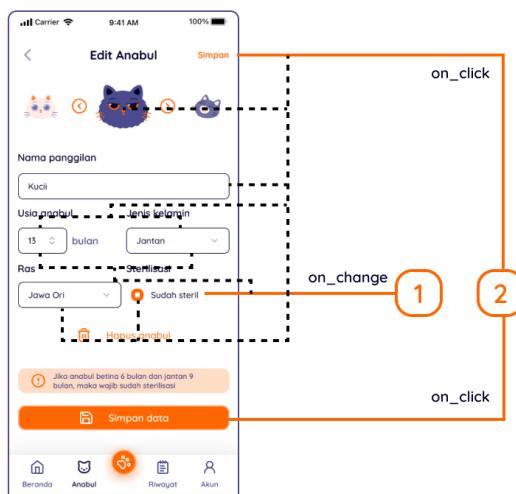
on_click

Nomor	Aksi	Query
1	Perubahan kolom ras	# <otomatis saat aksi 2>

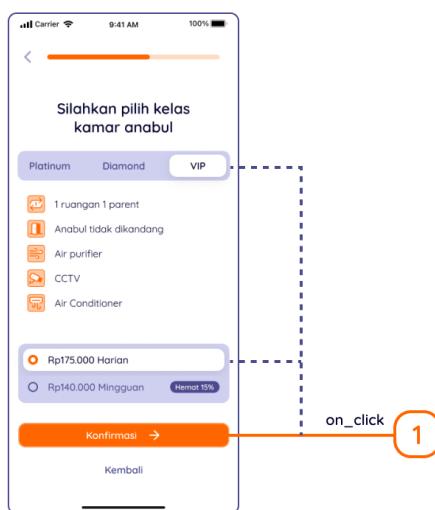
	dan sudah steril	
2	Klik tombol	<pre>INSERT INTO anabuls(nama_panggilan, jenis_kelamin, umur_bulan, is_neutered, id_image, id_ras, id_parent) VALUES ('meng', 'jantan', 9, 0, 2, 3, 2);</pre>



Nomor	Aksi	Query
1	Menunggu	<code>SELECT * FROM view_cat_detail;</code>
2	Menunggu	<code>SELECT * FROM view_cat_transactions LIMIT 2;</code>
3	Menunggu	<code>SELECT * FROM view_cat_checkuplists;</code>

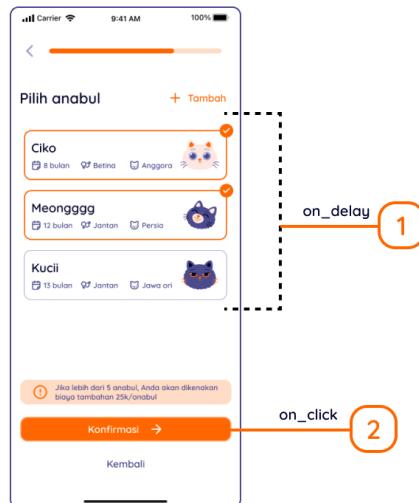


Nomor	Aksi	Query
1	Perubahan kolom ras dan sudah steril	# <otomatis saat aksi 2>
2	Klik tombol	<pre>UPDATE anabuls SET nama_panggilan='brownie', jenis_kelamin='jantan', umur_bulan=10, is_neutered=0, id_image=2, id_ras=3, id_parent=2 WHERE id=(SELECT anabul_id FROM active_user LIMIT 1);</pre>

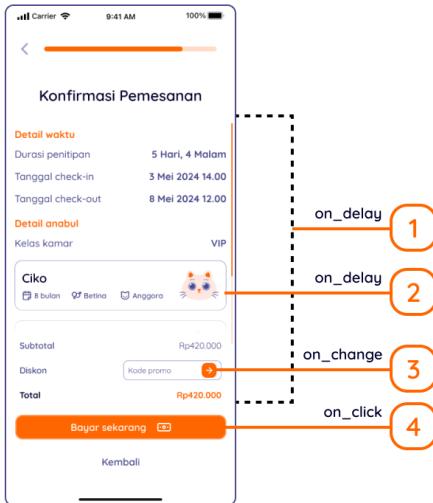


Nomor	Aksi	Query

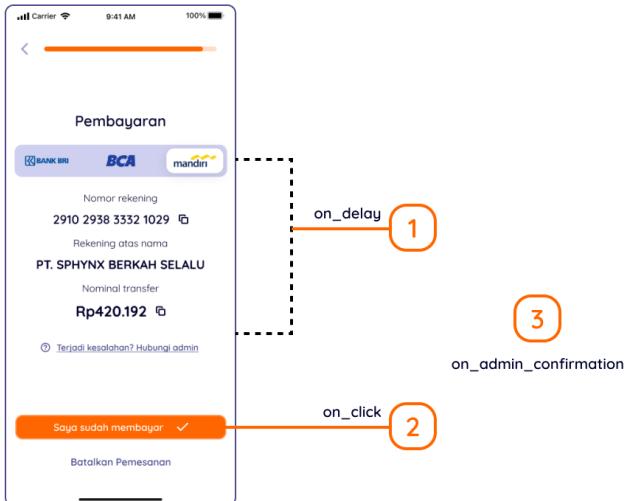
1	Klik tombol	<pre>INSERT INTO transactions(tanggal_checkin, tanggal_checkout, id_promo, id_payment, id_admin, id_room) VALUES ('2024-05-15 14:00:00', '2024-05-18 12:00:00', 2, 1, 3, 2);</pre>
---	-------------	--



Nomor	Aksi	Query
1	Menunggu	<pre>SELECT * FROM view_catlists_all;</pre>
2	Klik tombol	<pre>INSERT INTO transaction_details(id_anabul, id_transaksi) VALUES (2, '29c97229-1305-11ef-b75c-446d579ab8ce'), (3, '29c97229-1305-11ef-b75c-446d579ab8ce');</pre>



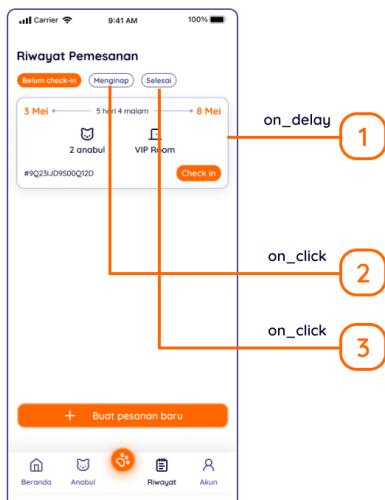
Nomor	Aksi	Query
1	Menunggu	<code>SELECT * FROM view_transaction_detail;</code>
2	Menunggu	<code>SELECT * FROM view_transaction_catlists;</code>
3	Mengubah kode promo	<code>SELECT id FROM promo_codes WHERE promo_code='CATKAREEM';</code>
4	Klik tombol	<code>UPDATE transactions SET id_promo=2 WHERE uid=(SELECT uid FROM view_transaction_detail LIMIT 1);</code>



Nomor	Aksi	Query
1	Menunggu	<code>SELECT * FROM view_paymentmethods_all;</code>
2	Klik tombol	<code>UPDATE transactions SET id_payment=3 WHERE uid=(SELECT uid FROM view_transaction_detail LIMIT 1);</code>
3	Menunggu konfirmasi admin	<code>UPDATE transactions SET is_paid=1 WHERE uid=(SELECT uid FROM view_transaction_detail LIMIT 1);</code>

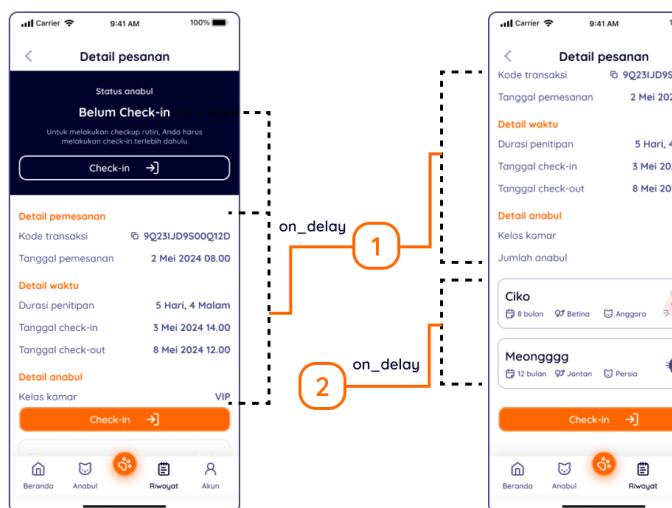


Nomor	Aksi	Query
1	Menunggu	<code>SELECT * FROM view_transaction_success;</code>



Nomor	Aksi	Query
1	Menunggu	<code>SELECT * FROM view_transactions_all WHERE status='belum_check_in';</code>

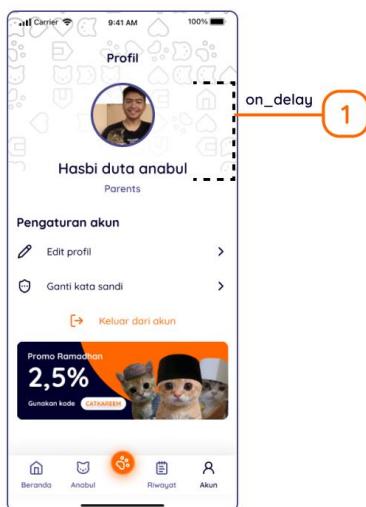
2	Klik filter	<pre>SELECT * FROM view_transactions_all WHERE status='menginap';</pre>
3	Klik filter	<pre>SELECT * FROM view_transactions_all WHERE status='selesai';</pre>



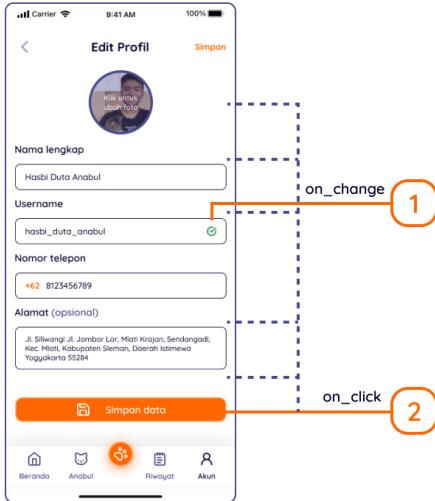
Nomor	Aksi	Query
1	Menunggu	<pre>SELECT * FROM view_transaction_detail;</pre>
2	Menunggu	<pre>SELECT * FROM view_transaction_catlists;</pre>



Nomor	Aksi	Query
1	Menunggu	<code>SELECT uid FROM view_transaction_detail;</code>
2	Konfirmasi dari Admin	<code>UPDATE transactions SET status='menginap'</code> <code>WHERE uid=(</code> <code>SELECT uid FROM view_transaction_detail LIMIT 1);</code>



Nomor	Aksi	Query
1	Menunggu	<code>SELECT * FROM view_parent_account;</code>



Nomor	Aksi	Query
1	Menunggu	<code>SELECT check_available_username('hasbi_duta_anabul');</code>
2	Klik tombol	<code>UPDATE parents</code> <code>SET nama_lengkap='hasbi_duta_anabul',</code> <code>username='hasbi_duta_anabul',</code> <code>password='pass_hasbi',</code> <code>nomor_telepon='0812345678913'</code> <code>WHERE id=(</code> <code>SELECT parent_id FROM active_user LIMIT 1</code> <code>);</code>



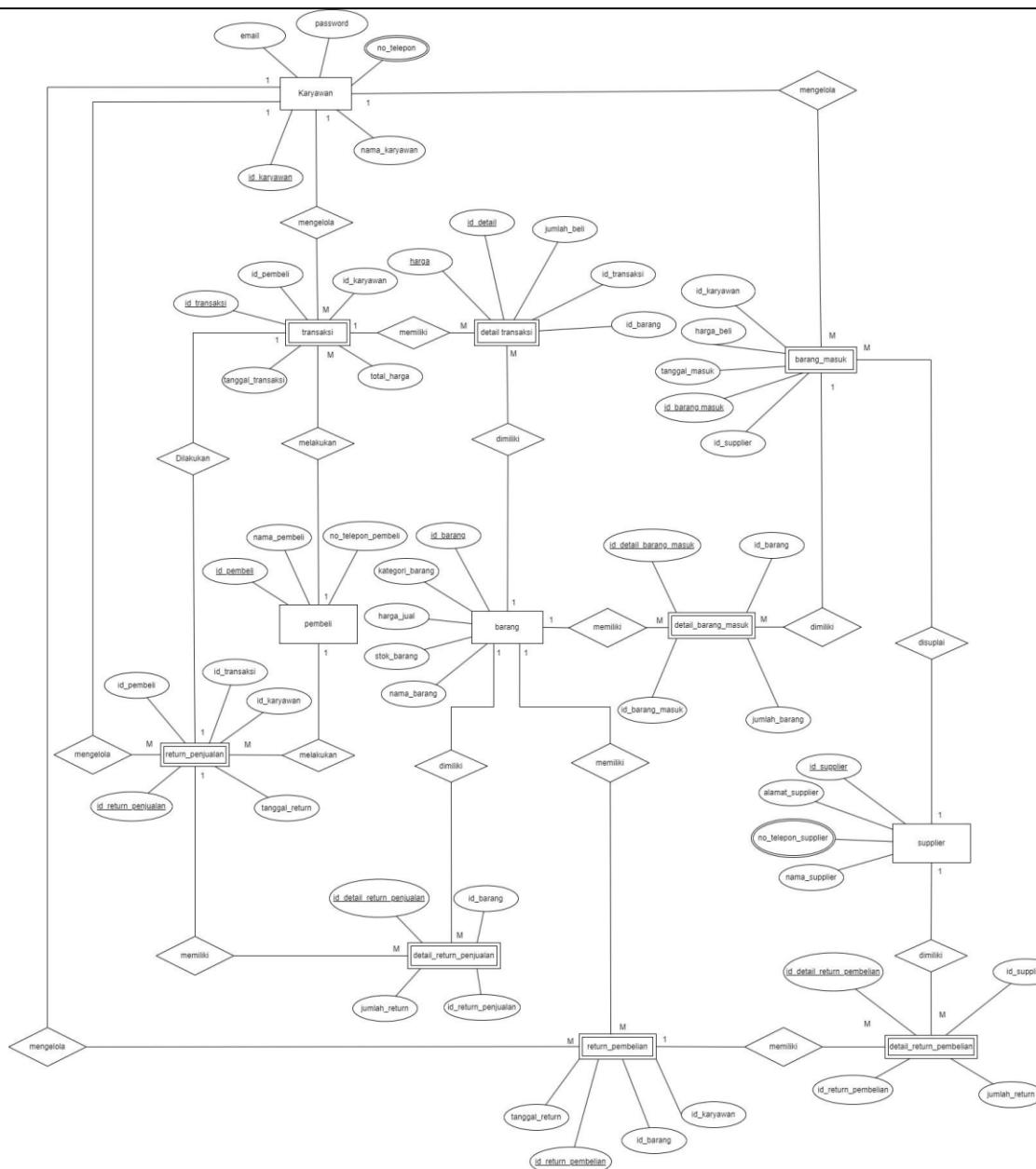
Nomor	Aksi	Query
1	Klik tombol	<pre>SELECT change_parent_password('pass_hasbi', 'ganti_pass', 'ganti_pass');</pre>

<p>Judul Aplikasi</p> <h1 style="text-align: center;">MANAJEMEN INVENTARIS DAN PEMANTAUAN PENDAPATAN SPARE PART MOTOR BERBASIS WEB</h1>	
<p>Dikembangkan Oleh</p>	
<ul style="list-style-type: none"> • 5220411377 – Moch Daffa Yudis Averill • 5220411386 – Tegar Wahyu Adi Saputra • 5220411402 – Adinesa Mukharam • 5220411403 – Hery Setiyawan Jordi • 5220411420 – Aziz Nurdiansyah • 5220411390 – Wahyu Akbar 	
<p>Deskripsi Singkat Instansi, Permasalahan, dan Solusi Aplikasi Yang Akan Dikembangkan [10 poin]</p>	
<p>Bengkel Athaya merupakan sebuah toko yang bergerak di bidang penjualan spare part motor, yang berlokasi di Blaburan, Rt 006/010, Mecanan, Bligo, Kec. Ngluwar, Kabupaten Magelang, Jawa Tengah 56485, Indonesia. Di toko ini menyediakan berbagai sparepart berbagai merk sepeda motor seperti Honda, Yamaha, Suzuki, dan Kawasaki. Dan Bengkel Athaya hanya mempunyai 2 karyawan saja.</p> <p>Salah satu alasan yang menjadi kunci perkembangan suatu perusahaan adalah dengan mampu mengelola inventaris barang serta memantau pendapatan bulanan toko mereka seakurat mungkin. pelaporan stok barang dan pemantauan pendapatan terbilang lambat dikarenakan proses pencatatanya masih menggunakan pencatatan manual, yaitu mencatatnya ke dalam buku sehingga membutuhkan waktu dalam menyesuaikan data. Hal tersebut menjadikan sistem yang ada pada kampus motor menjadi kurang dalam keakuratan informasi dan bisa berdampak pada proses laju bisnis perusahaan.</p> <p>Berdasarkan paparan diatas maka diperlukan sebuah aplikasi berupa sistem Manajemen Inventaris dan Pemantauan Pendapatan berbasis Website, yang sesuai dengan kebutuhan yang ada pada perusahaan. Sistem ini diharapkan dapat mempermudah admin dalam melakuakan pencatatan barang dan pemantauan pendapatan yang ada pada Kampus Motor baik dari segi waktu maupun tenaga. Dengan adanya sistem ini, semua data akan tersimpan ke dalam sistem, sehingga kerusakan file atau data yang diakibatkan oleh kelalaian admin seperti buku pencatatan yang hilang, ataupun permasalahan lainnya dapat teratasi. Sistem ini juga diharapkan dapat membantu admin dalam pembuatan laporan barang dan data pemantauan pendapatan yang ada pada perusahaan.</p>	

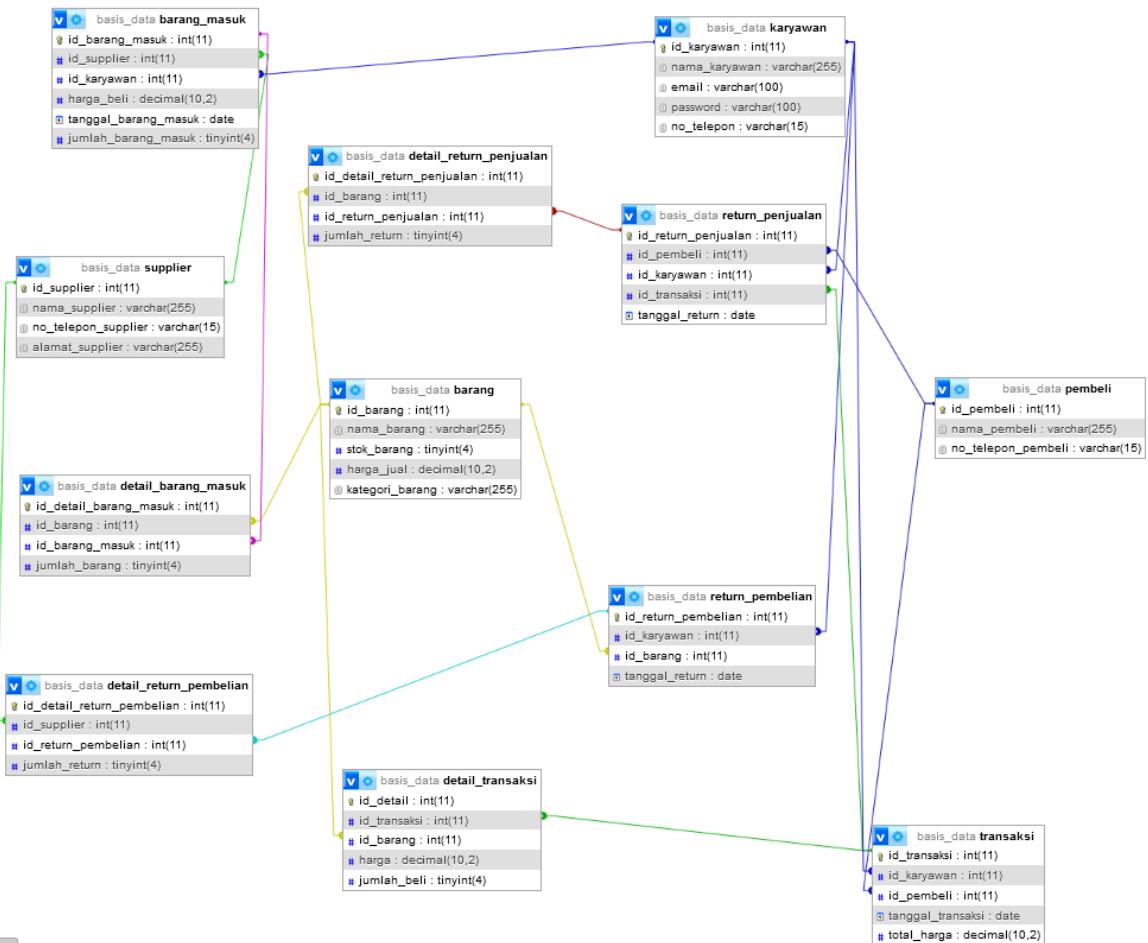
Pengguna Aplikasi [5 poin]

Pengguna aplikasi ini adalah karyawan

Desain Entity-Relationship Diagram (ERD) [10 poin]



Desain Relasi Basis Data [10 poin]



Query Data Definition Language (DDL) [25 poin]

- Membuat database Bernama bengkel:

```
1 CREATE DATABASE bengkel;
```
- Membuat tabel karyawan

```
CREATE TABLE karyawan (
    id_karyawan INT PRIMARY KEY AUTO_INCREMENT,
    nama_karyawan VARCHAR(255) NOT NULL,
    email VARCHAR(100) NOT NULL,
    password VARCHAR(100) NOT NULL,
    no_telepon VARCHAR(15) NOT NULL
);
```

- Membuat tabel barang

```
CREATE TABLE barang (
    id_barang INT PRIMARY KEY AUTO_INCREMENT,
    nama_barang VARCHAR(255) NOT NULL,
    stok_barang TINYINT NOT NULL,
    harga_jual DECIMAL(10,2) NOT NULL,
    kategori_barang VARCHAR(255) NOT NULL
);
```

- Membuat tabel supplier

```
CREATE TABLE supplier (
    id_supplier INT PRIMARY KEY AUTO_INCREMENT,
    nama_supplier VARCHAR(255) NOT NULL,
    no_telepon_supplier VARCHAR(15) NOT NULL,
    alamat_supplier VARCHAR(255) NOT NULL
);
```

- Membuat tabel barang_masuk

```
CREATE TABLE barang_masuk (
    id_barang_masuk INT PRIMARY KEY AUTO_INCREMENT,
    id_supplier INT NOT NULL,
    harga_beli DECIMAL(10,2) NOT NULL,
    tanggal_barang_masuk DATE NOT NULL,
    jumlah_barang_masuk TINYINT NOT NULL,
    FOREIGN KEY (id_supplier) REFERENCES supplier(id_supplier)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);
```

- Membuat tabel detail_barang_masuk

```
CREATE TABLE detail_barang_masuk (
    id_detail_barang_masuk INT PRIMARY KEY AUTO_INCREMENT,
    id_barang INT NOT NULL,
    id_barang_masuk INT NOT NULL,
    jumlah_barang TINYINT NOT NULL,
    FOREIGN KEY (id_barang) REFERENCES barang(id_barang)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (id_barang_masuk) REFERENCES barang_masuk(id_barang_masuk)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);
```

- Membuat tabel pembeli

```
CREATE TABLE pembeli (
    id_pembeli INT PRIMARY KEY AUTO_INCREMENT,
    nama_pembeli VARCHAR(255) NOT NULL,
    no_telepon_pembeli VARCHAR(15) NOT NULL
);
```

- Membuat tabel return penjualan

```
CREATE TABLE detail_return_penjualan (
    id_detail_return_penjualan INT PRIMARY KEY AUTO_INCREMENT,
    id_barang INT NOT NULL,
    id_return_penjualan INT NOT NULL,
    jumlah_return TINYINT NOT NULL,
    FOREIGN KEY (id_barang) REFERENCES barang(id_barang)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (id_return_penjualan) REFERENCES return_penjualan(id_return_penjualan)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);
```

- Membuat tabel detail_return_penjualan

```
CREATE TABLE detail_return_penjualan (
    id_detail_return_penjualan INT PRIMARY KEY AUTO_INCREMENT,
    id_barang INT NOT NULL,
    id_return_penjualan INT NOT NULL,
    jumlah_return TINYINT NOT NULL,
    FOREIGN KEY (id_barang) REFERENCES barang(id_barang)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (id_return_penjualan) REFERENCES return_penjualan(id_return_penjualan)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);
```

- Membuat tabel return_pembelian

```
CREATE TABLE return_pembelian (
    id_return_pembelian INT PRIMARY KEY AUTO_INCREMENT,
    id_karyawan INT NOT NULL,
    id_barang INT NOT NULL,
    tanggal_return DATE NOT NULL,
    FOREIGN KEY (id_barang) REFERENCES barang(id_barang)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (id_karyawan) REFERENCES karyawan(id_karyawan)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);
```

- Membuat tabel detail_return_pembelian

```
CREATE TABLE detail_return_pembelian (
    id_detail_return_pembelian INT PRIMARY KEY AUTO_INCREMENT,
    id_supplier INT NOT NULL,
    id_return_pembelian INT NOT NULL,
    jumlah_return TINYINT NOT NULL,
    FOREIGN KEY (id_supplier) REFERENCES supplier(id_supplier)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (id_return_pembelian) REFERENCES return_pembelian(id_return_pembelian)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);
```

- Membuat tabel transaksi

```
CREATE TABLE transaksi (
    id_transaksi INT PRIMARY KEY AUTO_INCREMENT,
    id_karyawan INT NOT NULL,
    id_pembeli INT NOT NULL,
    tanggal_transaksi DATE NOT NULL,
    total_harga DECIMAL(10,2) NOT NULL,
    FOREIGN KEY (id_pembeli) REFERENCES pembeli(id_pembeli)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (id_karyawan) REFERENCES karyawan(id_karyawan)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);
```

- Membuat detail_transaksi

```
CREATE TABLE detail_transaksi (
    id_detail INT PRIMARY KEY AUTO_INCREMENT,
    id_transaksi INT NOT NULL,
    id_barang INT NOT NULL,
    harga DECIMAL(10,2) NOT NULL,
    jumlah_beli TINYINT NOT NULL,
    FOREIGN KEY (id_transaksi) REFERENCES transaksi(id_transaksi)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (id_barang) REFERENCES barang(id_barang)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);
```

- Triger menambah stok_barang ketika jumlah_barang dimasukan di tabel detail_barang_masuk

```
CREATE TRIGGER after_detail_barang_masuk_insert
AFTER INSERT ON detail_barang_masuk
FOR EACH ROW
BEGIN
    -- Perbarui stok barang di tabel barang berdasarkan id_barang
    UPDATE barang
    SET stok_barang = stok_barang + NEW.jumlah_barang
    WHERE id_barang = NEW.id_barang;
END$$

DELIMITER ;
```

- Trigger mengurangi stok_barang pada saat proses transaksi lewat jumlah_beli di tabel detail_transaksi

```

DELIMITER $$

CREATE TRIGGER after_detail_transaksi_insert
AFTER INSERT ON detail_transaksi
FOR EACH ROW
BEGIN
    -- Kurangi stok barang di tabel barang berdasarkan id_barang
    UPDATE barang
    SET stok_barang = stok_barang - NEW.jumlah_beli
    WHERE id_barang = NEW.id_barang;
END$$

DELIMITER ;

```

- Trigger mengurangi Stok barang saat melakukan return pembelian kepada supplier kepada supplier lewat jumlah_return di kolom detail_retrurn_pembelian

```

DELIMITER ;;
CREATE TRIGGER reduce_stock_after_return_pembelian
AFTER INSERT ON detail_return_pembelian
FOR EACH ROW
BEGIN
    DECLARE stock_before INT;
    DECLARE stock_after INT;

    -- Dapatkan stok barang sebelum return
    SELECT stok_barang INTO stock_before FROM barang WHERE id_barang = (SELECT id_barang FROM return_pembelian WHERE id_return_pembelian = NEW.id_return_pembelian);

    -- Hitung stok barang setelah return
    SET stock_after = stock_before - NEW.jumlah_return;

    -- Update stok barang di tabel barang
    UPDATE barang SET stok_barang = stock_after WHERE id_barang = (SELECT id_barang FROM return_pembelian WHERE id_return_pembelian = NEW.id_return_pembelian);
END ;;
DELIMITER ;

```

Trigger mengurangi jumlah_beli di tabel detail_transaksi saat proses return penjualan dengan jumlah_return di tabel detail_return_penjualan

```

DELIMITER $$

CREATE TRIGGER after_detail_return_penjualan_insert
AFTER INSERT ON detail_return_penjualan
FOR EACH ROW
BEGIN
    -- Kurangi jumlah beli di tabel detail_transaksi berdasarkan id_barang
    UPDATE detail_transaksi AS dt
    INNER JOIN transaksi AS t ON dt.id_transaksi = t.id_transaksi
    SET dt.jumlah_beli = dt.jumlah_beli - NEW.jumlah_return
    WHERE dt.id_barang = NEW.id_barang AND t.id_transaksi = (SELECT id_transaksi FROM return_penjualan WHERE id_return_penjualan = NEW.id_return_penjualan);
END$$

DELIMITER ;

```

- Trigger menambah stok barang dari input jumlah_return di tabel detail_return_penjualan

```
DELIMITER //
```

```
CREATE TRIGGER tambah_stok_barang AFTER INSERT ON detail_return_penjualan
FOR EACH ROW
BEGIN
    DECLARE jumlah_return INT;

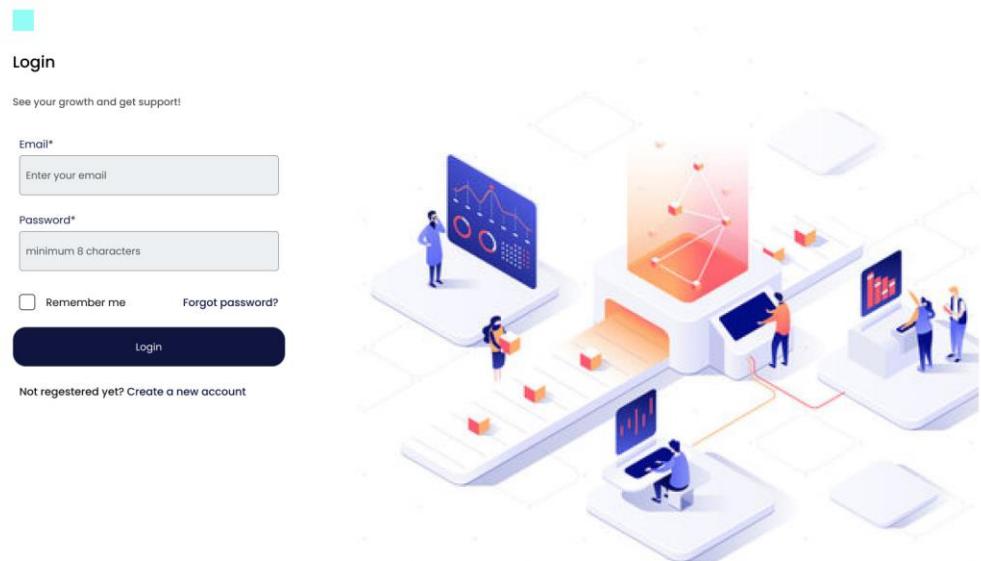
    -- Ambil nilai jumlah return yang dimasukkan
    SET jumlah_return = NEW.jumlah_return;

    -- Update stok barang di tabel barang
    UPDATE barang
    SET stok_barang = stok_barang + jumlah_return
    WHERE id_barang = NEW.id_barang;
END;
//
```

DELIMITER ;

Desain Interface (Wireframe/Mockup) Beserta Query-nya [poin 40]

Halaman Sign In



Query:

```
● ● ●
1 SELECT * FROM karyawan WHERE email = %s AND password = %s
```

Halaman Registrasi



Register

Name

Email
 Phone no.

Password

I agree to all terms, privacy policies, and fees

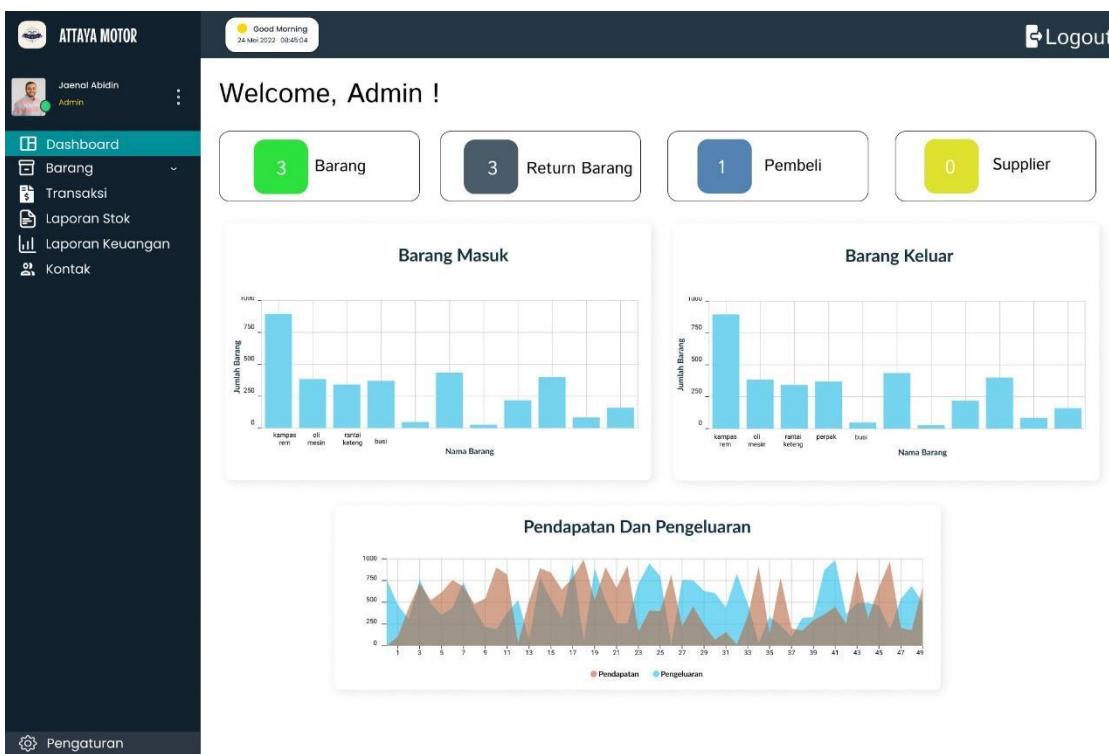
[Sign up](#)

Already have an account? [Log in](#)

Query:

```
1  INSERT INTO karyawan (password, email, nama_karyawan, no_telepon) VALUES (%s, %s, %s, %s)
```

Halaman Dashboard



Query:

```
1 SELECT COUNT(id_supplier) AS jumlah_supplier FROM Supplier;
2 SELECT COUNT(id_pembeli) AS jumlah_pembeli FROM pembeli;
3 SELECT COUNT(id_return_barang) AS jumlah_return FROM return_barang;
4 SELECT COUNT(id_barang) AS jumlah_barang FROM barang;
5
6 SELECT
7     barang.nama_barang,
8     SUM(barang_masuk.jumlah_barang_masuk)
9 FROM barang
10    INNER JOIN barang_masuk ON barang.id_barang = barang_masuk.id_barang GROUP BY barang.nama_barang;
11
12 SELECT
13     barang.nama_barang,
14     SUM(detail_transaksi.jumlah_beli)
15 FROM detail_transaksi
16    INNER JOIN barang ON detail_transaksi.id_barang = barang.id_barang
17 GROUP BY
18     barang.nama_barang;
```

```

1  SELECT
2      MONTH(tanggal_transaksi) AS bulan,
3      SUM(total_harga) AS total_pendapatan
4  FROM
5      transaksi
6  GROUP BY
7      MONTH(tanggal_transaksi);
8
9  SELECT
10     MONTH(barang_masuk.tanggal_masuk) AS bulan,
11     SUM(barang.harga_beli * barang_masuk.jumlah_barang_masuk) AS total_pengeluaran *
12  FROM
13     barang_masuk
14  JOIN
15     barang ON barang_masuk.id_barang = barang.id_barang
16  GROUP BY
17     MONTH(barang_masuk.tanggal_masuk);

```

Halaman Transaksi

The screenshot shows the ATTAYA MOTOR web application interface. On the left is a sidebar with a user profile (Joenal Abidin, Admin) and navigation links: Dashboard, Barang, Transaksi (highlighted in teal), Laporan Stok, Laporan Keuangan, and Kontak. The main content area has a header with a home icon, the title 'Transaksi', and a 'Logout' button. Below the header, there are input fields for 'Id Pembeli' (012), 'Id Barang' (002), 'Tanggal Transaksi' (15-5-2024), 'Jumlah' (2), and 'Id karyawan' (001). A 'Tambah' button is next to the 'Jumlah' field. Underneath, a table titled 'Detail pembelian' lists items: Oli (Rp 10.000, 2 units, sub Total Rp 20.000) and Kampas Rem (Rp 20.000, 1 unit, sub Total Rp 20.000). A '+ simpan' button is at the bottom right of the table.

Id Barang	Nama Barang	Harga	Jumlah	Sub Total
001	Oli	Rp 10.000	2	Rp 20.000
002	Kampas Rem	Rp 20.000	1	Rp 20.000
				Total Harga : Rp 40.000

Query:

```
1 INSERT INTO transaksi (id_pembeli, tanggal_transaksi, id_karyawan)
2 VALUES (012, CURRENT_DATE, 001);
3
4 SET @last_id = LAST_INSERT_ID();
5
6 INSERT INTO detail_transaksi (id_transaksi, id_barang, jumlah_beli)
7 VALUES
8 (@last_id, 001, 2),
9 (@last_id, 002, 1);
10
11 UPDATE transaksi
12 SET total_harga = 40000
13 WHERE id_transaksi = @last_id;
14
15 SELECT
16     b.id_barang,
17     b.nama_barang,
18     b.harga_beli,
19     dt.jumlah_beli
20 FROM
21 barang b
22 JOIN
23     detail_transaksi dt ON b.id_barang = dt.id_barang
24 WHERE dt.id_transaksi = @last_id;
```

Halaman Riwayat Transaksi

The screenshot shows the ATTAYA MOTOR application interface. The left sidebar has a dark theme with white text and icons. It includes a user profile for 'Jaenal Abidin Admin', navigation links for 'Dashboard', 'Barang', 'Transaksi' (which is highlighted in teal), 'Laporan Stok', 'Laporan Keuangan', 'Kontak', and 'Pengaturan'. The main content area has a light background. At the top, there are buttons for 'Logout' and a back arrow. Below that, tabs for 'Riwayat' and 'Transaksi' are visible, with 'Riwayat' being the active tab. A date range selector shows 'Periode Transaksi : 10-5-2024 s/d 20-5-2024'. The 'Riwayat Transaksi' section displays a table with one row of data:

ID Transaksi	Tanggal Transaksi	Nama Pembeli	Nama Barang	Total Harga
001	15-5-2024	kiki	oli,kampas rem	40000,00

A small 'Detail Transaksi' button is located at the bottom right of the table.

Query:

```
1  SELECT
2      t.id_transaksi,
3      t.tanggal_transaksi,
4      p.nama_pembeli,
5      GROUP_CONCAT(b.nama_barang SEPARATOR ', ') AS nama_barang,
6      t.total_harga
7  FROM
8      transaksi t
9  JOIN
10     pembeli p ON t.id_pembeli = p.id_pembeli
11  JOIN
12     detail_transaksi dt ON t.id_transaksi = dt.id_transaksi
13  JOIN
14     barang b ON dt.id_barang = b.id_barang
15 WHERE
16     t.tanggal_transaksi BETWEEN '2024-01-01' AND '2024-05-31'
17 GROUP BY
18     t.id_transaksi
19 ORDER BY
20     t.tanggal_transaksi DESC;
```

Halaman Detail Transaksi

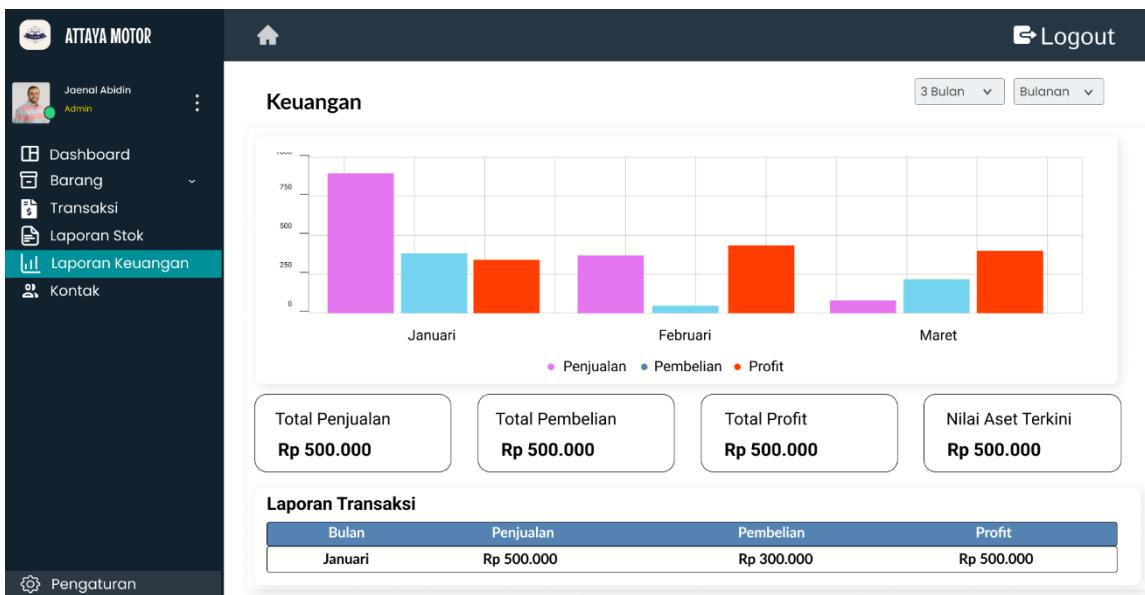
The screenshot shows the ATTAYA MOTOR application interface. The top navigation bar includes a logo, user profile (Jenail Abidin Admin), a home icon, and a Logout button. The left sidebar menu has items: Dashboard, Barang (selected), Transaksi (highlighted in blue), Laporan Stok, Laporan Keuangan, and Kontak. At the bottom of the sidebar is a Pengaturan (Settings) button. The main content area displays a table titled "Detail Transaksi" with the following data:

ID Detail	ID Transaksi	Tanggal Transaksi	Nama Pembeli	Nama Admin	Nama Barang	Harga	Jumlah	Total Harga
001	001	15-5-2024	kiki	Jumali	oli	10000,00	2	20000,00
002	001	15-5-2024	kiki	Jumali	kampas rem	20000,00	1	20000,00

Query:

```
1  SELECT
2      dt.id_detail,
3      t.id_transaksi,
4      t.tanggal_transaksi,
5      p.nama_pembeli,
6      k.nama_karyawan AS nama_admin,
7      b.nama_barang,
8      b.harga_jual AS harga,
9      dt.jumlah_beli,
10     (b.harga_jual * dt.jumlah_beli) AS total_harga
11    FROM
12        detail_transaksi dt
13    JOIN
14        transaksi t ON dt.id_transaksi = t.id_transaksi
15    JOIN
16        pembeli p ON t.id_pembeli = p.id_pembeli
17    JOIN
18        karyawan k ON t.id_karyawan = k.id_karyawan
19    JOIN
20        barang b ON dt.id_barang = b.id_barang
21    WHERE
22        t.id_transaksi = 1
23    ORDER BY
24        t.tanggal_transaksi DESC;
```

Halaman Laporan Keuangan



Query:

```
1 SELECT
2     -- Total Penjualan dalam 3 Bulan Terakhir
3     SUM(transaksi.total_harga) AS total_penjualan,
4
5     -- Total Pembelian dalam 3 Bulan Terakhir
6     (SELECT SUM(barang_masuk.jumlah_barang_masuk * barang.harga_beli)
7      FROM barang_masuk
8      JOIN barang ON barang_masuk.id_barang = barang.id_barang
9      WHERE barang_masuk.tanggal_masuk >= DATE_SUB(CURDATE(), INTERVAL 3 MONTH)) AS total_pembelian,
10
11     -- Total Profit dalam 3 Bulan Terakhir
12     (SELECT SUM((barang.harga_jual - barang.harga_beli) * detail_transaksi.jumlah_beli)
13      FROM detail_transaksi
14      JOIN transaksi ON detail_transaksi.id_transaksi = transaksi.id_transaksi
15      JOIN barang ON detail_transaksi.id_barang = barang.id_barang
16      WHERE transaksi.tanggal_transaksi >= DATE_SUB(CURDATE(), INTERVAL 3 MONTH)) AS total_profit,
17
18     -- Nilai Aset Terkini
19     (SELECT SUM(barang.stok_barang * barang.harga_beli)
20      FROM barang) AS nilai_aset_terkini
21
22     FROM transaksi
23     WHERE transaksi.tanggal_transaksi >= DATE_SUB(CURDATE(), INTERVAL 3 MONTH);
```

```

1  SELECT
2      DATE_FORMAT(transaksi.tanggal_transaksi, '%Y-%m') AS bulan,
3
4      -- Total Penjualan per Bulan
5      SUM(transaksi.total_harga) AS total_penjualan,
6
7      -- Total Pembelian per Bulan
8      (SELECT SUM(barang_masuk.jumlah_barang_masuk * barang.harga_beli)
9       FROM barang_masuk
10      JOIN barang ON barang_masuk.id_barang = barang.id_barang
11      WHERE DATE_FORMAT(barang_masuk.tanggal_masuk, '%Y-%m') = DATE_FORMAT(transaksi.tanggal_transaksi, '%Y-%m')) AS total_pembelian,
12
13      -- Total Profit per Bulan
14      (SELECT SUM((barang.harga_jual - barang.harga_beli) * detail_transaksi.jumlah_beli)
15       FROM detail_transaksi
16      JOIN transaksi t2 ON detail_transaksi.id_transaksi = t2.id_transaksi
17      JOIN barang ON detail_transaksi.id_barang = barang.id_barang
18      WHERE DATE_FORMAT(t2.tanggal_transaksi, '%Y-%m') = DATE_FORMAT(transaksi.tanggal_transaksi, '%Y-%m')) AS total_profit
19
20  FROM transaksi
21  GROUP BY bulan
22  ORDER BY bulan;

```

Halaman Laporan Stok Barang

The screenshot shows the ATTAYA MOTOR application interface. On the left is a sidebar with a user profile (Jaenal Abidin, Admin), navigation links (Dashboard, Barang, Transaksi, Laporan Stok, Laporan Keuangan, Kontak), and a settings link (Pengaturan). The main content area has a header with a home icon, Logout, and date filters (10-5-2024 s/d 20-5-2024). Below the header, there are two boxes: 'Jenis Barang Terkini' showing '1' and 'Total Barang Terkini' showing '45'. A table below lists items: Id Barang, Nama Barang, Harga, Stok Masuk, Terjual, Sisa. The first item is Oli Mesin with ID 001.

ID Barang	Nama Barang	Harga	Stok Masuk	Terjual	Sisa
001	Oli Mesin	kiki	20	10	10

Query:

```

1  SELECT
2      COUNT(DISTINCT bm.id_barang) AS jumlah_jenis_barang_terbaru,
3      SUM(bm.jumlah_barang_masuk) AS total_barang_masuk
4  FROM
5      barang_masuk bm
6  WHERE
7      DATE(bm.tanggal_masuk) BETWEEN '2024-05-10' AND '2024-05-20';

```

```

1 SELECT
2     bm.id_barang,
3     b.nama_barang,
4     b.harga_beli AS harga,
5     bm.jumlah_barang_masuk AS stok_masuk,
6     COALESCE(SUM(dt.jumlah_beli), 0) AS jumlah_terjual,
7     (bm.jumlah_barang_masuk - COALESCE(SUM(dt.jumlah_beli), 0)) AS sisa_stok
8 FROM
9     barang_masuk bm
10 JOIN
11     barang b ON bm.id_barang = b.id_barang
12 LEFT JOIN
13     detail_transaksi dt ON bm.id_barang = dt.id_barang
14 LEFT JOIN
15     transaksi t ON dt.id_transaksi = t.id_transaksi AND DATE(t.tanggal_transaksi) BETWEEN '2024-05-10' AND '2024-05-20'
16 WHERE
17     DATE(bm.tanggal_masuk) BETWEEN '2024-05-10' AND '2024-05-20'
18 GROUP BY
19     bm.id_barang, b.nama_barang, b.harga_beli, bm.jumlah_barang_masuk
20 ORDER BY
21     bm.id_barang;

```

Halaman Data Barang

ID Barang	Nama Barang	Kategori	Stok	Harga Beli	Harga Jual	Aksi
001	Oli	Part Pelumas	5	Rp 8.000	Rp 10.000	
002	Kampas Rem	Part pegereman	10	Rp 16.000	Rp 20.000	

Query :

```

1 SELECT
2     b.id_barang,
3     b.nama_barang,
4     b.kategori_barang,
5     b.stok_barang,
6     b.harga_jual,
7     bm.harga_beli
8 FROM
9     barang_masuk bm
10 JOIN
11     detail_barang_masuk dbm ON bm.id_barang_masuk = dbm.id_barang_masuk
12 JOIN
13     barang b ON dbm.id_barang = b.id_barang

```

Halaman Tambah Data Barang

The screenshot shows the 'Tambah Data Barang' (Add Item Data) page. The page title is 'Tambah Data Barang'. There are four input fields: 'Nama Barang', 'Kategori Barang', 'Stok Awal', and 'Harga Jual'. Below the input fields is a blue 'Tambah' (Add) button. On the left side, there is a sidebar with the 'ATTAYA MOTOR' logo and a user profile for 'Jaenal Abidin Admin'. The sidebar includes links for Dashboard, Barang (which is highlighted in teal), Barang Masuk, Return Barang, Transaksi, Laporan Stok, Laporan Keuangan, Kontak, and Pengaturan.

Query:

```
1 INSERT INTO barang
2 (nama_barang,kategori_barang,stok_barang,harga_jual)
3 VALUES ("%","%",%,%);
```

Halaman Search Data Barang

ID Barang	Nama Barang	Kategori	Stok	Harga	Aksi
001	Tank Bag	Part Motor	5	Rp. 1.800.000	

Query:

```
1 SELECT * FROM barang
2 WHERE id_barang LIKE '' OR
3     nama_barang LIKE 'Tank Bag' OR
4     kategori_barang LIKE '' OR
5     stok_barang LIKE '' OR
6     harga LIKE '';
```

Halaman Data Supplier

The screenshot shows the ATTAYA MOTOR application interface. On the left is a sidebar with a user profile for 'Joenal Abidin Admin' and a navigation menu including 'Dashboard', 'Barang', 'Transaksi', 'Laporan Stok', 'Laporan Keuangan', 'Kontak', 'Supplier' (which is highlighted in blue), 'Pembeli', and 'Karyawan'. Below the sidebar is a 'Pengaturan' button. The main content area is titled 'Data Supplier' and contains a table with the following data:

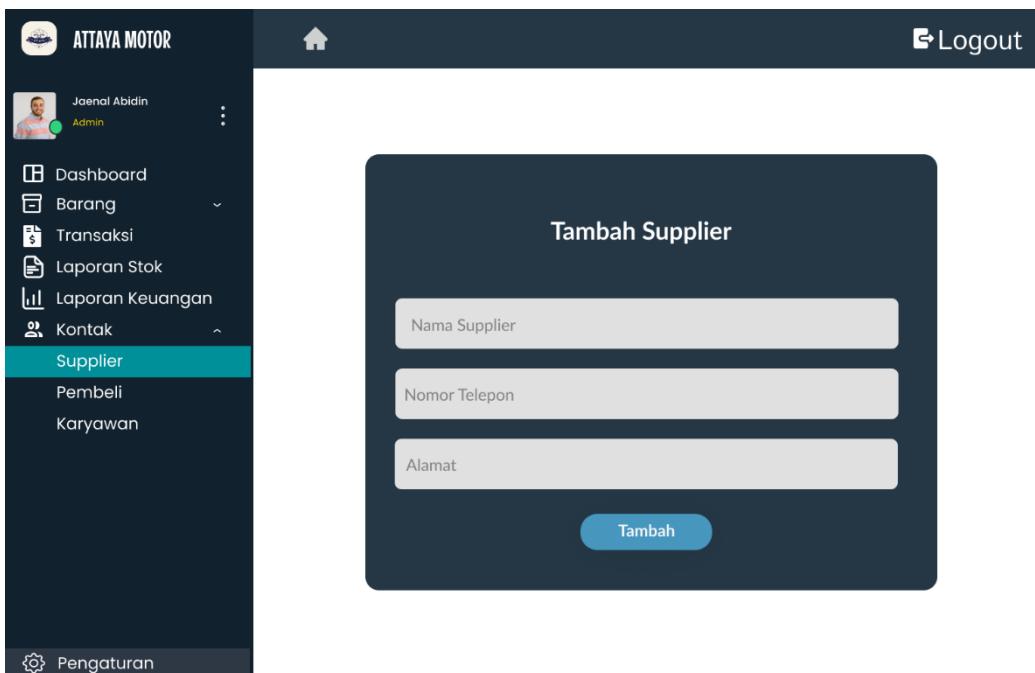
ID Supplier	Nama	Nomor Telepon	Alamat	Aksi
001	Fahreza Syahrul Ramadhan	083100156725	Jombor	
002	Muhammad Daffa Yudis Averil	083100156726	Malioboro	
003	Hery Setiawan Jordi	083100156727	Solo	
004	Adinesa Mukharam	083100156728	Klaten	
005	Muhammad Raihan	083100156759	Grobogan	

Query:

```
1  SELECT *FROM supplier;
```

```
1  DELETE FROM supplier WHERE id_supplier = %;
```

Halaman Tambah Data Supplier



Query:

```
1 INSERT INTO supplier(nama_supplier, no_telepon_supplier, alamat_supplier)
2 VALUES ("%","%","%");
```

Halaman Search Data Supplier

The screenshot shows a user interface for managing suppliers. On the left is a sidebar with a logo, user info (Joenaf Abidin Admin), and a navigation menu including Dashboard, Barang, Transaksi, Laporan Stok, Laporan Keuangan, Kontak, Supplier (selected), Pembeli (selected), Karyawan, and Pengaturan. The main content area has a header with a home icon, 'Setting', and 'Logout'. It displays a table titled 'Data Supplier' with one row of data:

ID Supplier	Nama Supplier	Nomor Telepon	Alamat	Aksi
005	Muhammad Raihan	083100156759	Grobogan	

Buttons for '+ Tambah Supplier' and '001' are also visible.

Query:

```
1 SELECT * FROM supplier
2 WHERE id_supplier LIKE 005 OR
3 nama_supplier LIKE "" OR
4 no_telepon_supplier LIKE "" OR
5 alamat_supplier LIKE "" ;
```

Halaman Data Pembeli

The screenshot shows a user interface for managing buyers. The sidebar and header are identical to the supplier search page. The main content area displays a table titled 'Data Pembeli' with five rows of data:

ID Pelanggan	Nama	Nomor Telepon	Aksi
001	Fahreza Syahrul Ramadhan	083100156725	
002	Muhammad Daffa Yudis Averil	083100156726	
003	Hery Setiawan Jordi	083100156727	
004	Adinesa Mukharam	083100156728	
005	Muhammad Raihan	083100156729	

Query:

```
1  SELECT * FROM pembeli;
```

```
1  DELETE FROM pembeli WHERE id_pembeli = %;
```

Halaman Tambah Data Pembeli

The screenshot shows a dark-themed web application interface. On the left is a sidebar with a user profile picture of a man, the name 'Jaenal Abidin', and the title 'Admin'. Below the profile are several menu items: Dashboard, Barang, Transaksi, Laporan Stok, Laporan Keuangan, Kontak, Supplier, **Pembeli** (which is highlighted in blue), Karyawan, and Pengaturan. At the top right are 'Logout' and a home icon. The main content area has a dark background and contains a form titled 'Tambah Pembeli'. It has two input fields: 'Nama' (Name) and 'Nomor Telepon' (Phone Number). A blue 'Tambah' (Add) button is at the bottom of the form.

Query:

```
● ● ●  
1  INSERT INTO pembeli(nama_pembeli, no_telepon_pembeli)  
2  VALUES("%",%);
```

Halaman Edit Data Pembeli

The screenshot shows a dark-themed web application. On the left is a sidebar with a user profile for 'Jaenal Abidin Admin'. The sidebar includes links for Dashboard, Barang, Transaksi, Laporan Stok, Laporan Keuangan, Kontak, Supplier, **Pembeli** (which is highlighted in teal), Karyawan, and Pengaturan. The main content area has a title 'Edit Pembeli' and two input fields: 'Nama' and 'Nomor Telepon'. A blue 'Simpan' button is at the bottom right of the form.

Query:

```
● ● ●  
1  UPDATE pembeli  
2  SET nama_pembeli = 'Kurnia', no_telepon_pembeli = 08564234252  
3  WHERE id_pembeli = 1;
```

Halaman Search Data Pembeli

ID Pembeli	Nama Pembeli	Nomor Telepon	Aksi
001	Fahreza Syahrul Ramadhan	083100156727	

Query:

```
1 SELECT* FROM pembeli
2 WHERE id_pembeli LIKE "001" OR
3 nama_pembeli LIKE "" OR
4 no_telepon_pembeli LIKE "";
```

Halaman Data Karyawan

ID Karyawan	Nama	Nomor Telepon	Email	password	Aksi
001	Jordi	083100156727	Jordi@gmail.com	1234	
002	Yudis	083100156728	Yudis@gmail.com	4234	
003	Muhammad	083100156729	Muhammad@gmail.com	6436	

Query:

```
1  SELECT *FROM karyawan;
```

```
1  DELETE FROM karyawan WHERE id_karyawan = %;
```

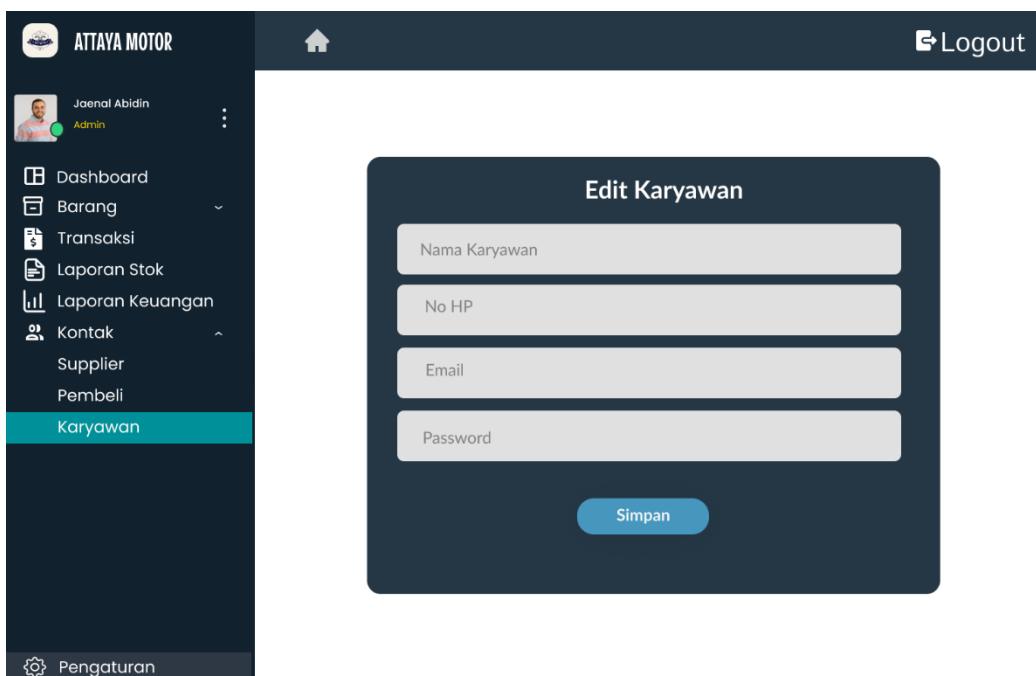
Halaman Tambah Data Karyawan

The screenshot shows the ATTAYA MOTOR application interface. On the left is a sidebar menu with the following items: Dashboard, Barang, Transaksi, Laporan Stok, Laporan Keuangan, Kontak, Supplier, Pembeli, and Karyawan (which is highlighted in teal). At the bottom of the sidebar is Pengaturan. The main content area has a dark header with a home icon, Logout button, and a back arrow. Below the header is a form titled "Tambah Karyawan" with four input fields: Nama Karyawan, No HP, Email, and Password. A blue "Tambah" button is at the bottom of the form.

Query:

```
1 INSERT INTO karyawan(nama_karyawan, email, PASSWORD,no_telepon)
2 VALUES ("%","%","%","%");
```

Halaman Edit Data Karyawan



Query:

```
1 UPDATE karyawan
2 SET nama_karyawan = CASE WHEN 'daffa' IS NOT NULL THEN 'daffa' ELSE nama_karyawan END,
3     email = CASE WHEN 'daffa@gmail.com' IS NOT NULL THEN 'daffa@gmail.com' ELSE email END,
4     PASSWORD = CASE WHEN "" IS NOT NULL THEN "" ELSE PASSWORD END,
5     no_telepon = CASE WHEN "" IS NOT NULL THEN "" ELSE no_telepon END
6 WHERE id_karyawan = '1';
```

Halaman Search Data Karyawan

The screenshot shows the 'Data Karyawan' section of the ATTAYA MOTOR application. The sidebar on the left has a user profile for 'Jenail Abidin Admin' and a navigation menu with items like Dashboard, Barang, Transaksi, Laporan Stok, Laporan Keuangan, Kontak, Supplier, Pembeli, and Karyawan (which is highlighted in blue). The main content area has a search bar with 'Jordi' and a magnifying glass icon. A table lists the data for one employee:

ID Karyawan	Nama Karyawan	Email	Nomor Telepon	Aksi
001	Jordi	Jordi@gmail.com	083100156727	

Query:

```
1 SELECT * FROM karyawan
2 WHERE id_karyawan LIKE "" OR
3 nama_karyawan LIKE "Jordi" OR
4 email LIKE "" OR
5 PASSWORD LIKE "" OR
6 no_telepon LIKE "";
```

Halaman Barang Masuk

The screenshot shows the 'Barang Masuk' (Item Purchase) page of the ATTAYA MOTOR application. On the left is a sidebar with user information (Jesual Abidin, Admin) and navigation links: Dashboard, Barang, Barang Masuk (highlighted in green), Return Barang, Transaksi, Laporan Stok, Laporan Keuangan, and Kontak. Below the sidebar are 'Pengaturan' and 'Logout' buttons.

The main content area has two tabs: 'Barang Masuk' and 'Data'. The 'Barang Masuk' tab is active, showing a form with fields: Id Supplier (05), Tanggal Masuk (15-5-2024), Id karyawan (01), Id Barang (002), Harga (20000), and Jumlah (30). A 'Tambah' button is at the bottom right of the form. Below the form is a table titled 'Detail Barang Masuk' with columns: Id Barang, Nama Barang, Harga, and Jumlah. It contains two rows: row 001 (Oli, Rp 10.000, 20) and row 002 (Kampas Rem, Rp 20.000, 30). A 'Total barang' field shows 50, and a 'simpan' button is at the bottom right of the table.

Query:

```
1 INSERT INTO barang_masuk (id_supplier, id_karyawan, harga_beli, tanggal_barang_masuk)
2 VALUES (05, 01, 20000, CURRENT_DATE, );
3
4 SET @last_id = LAST_INSERT_ID();
5
6 INSERT INTO detail_barang_masuk (id_barang_masuk, id_barang, jumlah_barang)
7 VALUES (@last_id, 001, 20),
8      (@last_id, 002, 30);
9
10 UPDATE barang_masuk
11 SET jumlah_barang_masuk = 2
12 WHERE id_barang_masuk = @last_id;
13
14 SELECT
15     b.id_barang,
16     b.nama_barang,
17     bm.harga_beli,
18     dbm.jumlah_barang
19 FROM
20     barang_masuk bm
21 JOIN
22     detail_barang_masuk dbm ON bm.id_barang_masuk = dbm.id_barang_masuk
23 JOIN
24     barang b ON dbm.id_barang = b.id_barang
25 WHERE
26     bm.id_barang_masuk = @last_id;
```

Halaman Riwayat barang masuk

The screenshot shows the 'ATTAYA MOTOR' application interface. On the left is a sidebar with a user profile for 'Jenail Abidin Admin'. The sidebar includes links for Dashboard, Barang, Barang Masuk (which is highlighted in blue), Return Barang, Transaksi, Laporan Stok, Laporan Keuangan, Kontak, and Pengaturan. The main content area has a header with a home icon, 'Logout', and a search bar. Below the header is a table titled 'Data Barang Masuk' with columns: Id Barang Masuk, Nama Barang, Nama Supplier, and Tanggal Transaksi. A single row is displayed: '001 oli,kampas rem', 'kiki', '15-5-2024', and a 'Detail Masuk' button.

Query:

```
1 SELECT
2     bm.id_barang_masuk,
3     GROUP_CONCAT(b.nama_barang) AS nama_barang,
4     s.nama_supplier,
5     bm.tanggal_barang_masuk
6 FROM
7     barang_masuk bm
8 JOIN
9     detail_barang_masuk dbm ON bm.id_barang_masuk = dbm.id_barang_masuk
10 JOIN
11     barang b ON dbm.id_barang = b.id_barang
12 JOIN
13     supplier s ON bm.id_supplier = s.id_supplier
14 GROUP BY
15     bm.id_barang_masuk,
16     s.nama_supplier,
17     bm.tanggal_barang_masuk;
```

Halaman Detail barang masuk

The screenshot shows the 'ATTAYA MOTOR' application interface. On the left is a sidebar with a user profile for 'Joenal Abidin Admin'. The sidebar includes links for Dashboard, Barang, Barang Masuk (which is highlighted in teal), Return Barang, Transaksi, Laporan Stok, Laporan Keuangan, and Kontak. At the bottom of the sidebar is a 'Pengaturan' link. The main content area has a header with a home icon and a 'Logout' button. Below the header is a title 'Detail Barang Masuk'. A table displays two rows of data:

ID Barang Masuk	ID Detail Barang Masuk	Nama Barang	Nama Supplier	Tanggal Masuk	Harga	Jumlah
001	001	Oli	kiki	15-5-2024	10000,00	20
001	002	Kampas rem	kiki	15-5-2024	20000,00	30

Query

```
1  SELECT
2      bm.id_barang_masuk,
3      dbm.id_detail_barang_masuk,
4      b.nama_barang,
5      s.nama_supplier,
6      bm.tanggal_barang_masuk,
7      bm.harga_beli,
8      dbm.jumlah_barang
9  FROM
10     barang_masuk bm
11 JOIN
12     detail_barang_masuk dbm ON bm.id_barang_masuk = dbm.id_barang_masuk
13 JOIN
14     barang b ON dbm.id_barang = b.id_barang
15 JOIN
16     supplier s ON bm.id_supplier = s.id_supplier
17 WHERE
18     bm.id_barang_masuk = 1;
19
```

Halaman Edit barang masuk

The screenshot shows the 'Edit Barang Masuk' form. The sidebar on the left has a dark theme with white icons and text. The 'Barang Masuk' link is highlighted with a teal background. The main form has a light gray background with four input fields: 'ID Barang', 'ID Supplier', 'Harga', and 'Jumlah Barang'. A large blue 'Simpan' button is at the bottom. The top right corner has a 'Logout' button.

Query:

```
1 UPDATE barang_masuk
2 SET
3     id_supplier = CASE WHEN '2' = '' THEN id_supplier ELSE '2' END,
4     harga_beli = CASE WHEN '10' = '' THEN harga_beli ELSE '10' END
5 WHERE id_barang_masuk = '1';
6
7 UPDATE detail_barang_masuk
8 SET
9     id_barang = CASE WHEN '1' = '' THEN id_barang ELSE '1' END
10 WHERE id_detail_barang_masuk = '1';
```

Halaman Search Barang Masuk

The screenshot shows a web application interface for managing motor inventory. On the left is a dark sidebar with user information (Jaenal Abidin, Admin) and navigation links: Dashboard, Barang, Barang Masuk (selected), Return Barang, Transaksi, Laporan Stok, Laporan Keuangan, and Kontak. Below these are Pengaturan and Logout buttons. The main content area has a header 'Detail Barang Masuk' and a search bar with placeholder 'Data Barang Masuk'. A table displays a single row of data:

Id Barang Masuk	Id Detail Barang Masuk	Nama Barang	Nama Supplier	Tanggal Masuk	Harga	Jumlah	Aksi
001	001	Oli	kiki	15-5-2024	10000,00	20	

Query;

```
1 SELECT
2     bm.id_barang_masuk,
3     dbm.id_detail_barang_masuk,
4     b.nama_barang,
5     bm.tanggal_barang_masuk,
6     bm.harga_beli,
7     dbm.jumlah_barang AS jumlah_barang_masuk
8 FROM
9     barang_masuk bm
10 JOIN
11     detail_barang_masuk dbm ON bm.id_barang_masuk = dbm.id_barang_masuk
12 JOIN
13     barang b ON dbm.id_barang = b.id_barang
14 WHERE bm.id_barang_masuk = 1;
```

Halaman Return Penjualan

The screenshot shows the 'Return Penjualan' section of the ATTAYA MOTOR application. On the left, there's a sidebar with user information (Jenal Abidin, Admin) and navigation links: Dashboard, Barang, Barang Masuk, Return Barang, Penjualan (which is selected), Pembelian, Laporan Stok, Laporan Keuangan, and Kontak. Below these are Pengaturan and Log Out buttons.

The main content area has a header 'Return Penjualan' and a 'Data' tab. It contains a form with fields: Id Pembeli (05), Id Karyawan (05), Id Transaksi (0005), Tanggal Return (15-5-2024), Id Barang (002), and Jumlah (2). A 'Tambah' button is at the bottom of the form. To the right, a table titled 'Detail Return Penjualan' lists items: Oli (3 units) and Kampas Rem (2 units).

Id Barang	Nama Barang	Jumlah
001	Oli	3
002	Kampas Rem	2

[+ simpan](#)

Query:

```
1  INSERT INTO return_penjualan (id_pembeli, id_transaksi, id_karyawan, tanggal_return)
2  VALUES (0005,05, 05, CURRENT_DATE);
3
4  SET @last_id = LAST_INSERT_ID();
5
6  INSERT INTO detail_return_penjualan (id_barang, id_return_penjualan, jumlah_return)
7  VALUES
8  (001, @last_id, 3),
9  (002, @last_id, 2);
10
11 SELECT
12     b.id_barang,
13     b.nama_barang,
14     drp.jumlah_return
15 FROM
16     barang b
17 JOIN
18     detail_return_penjualan drp ON b.id_barang = drp.id_barang
19 WHERE
20     drp.id_return_penjualan = @last_id;
```

Halaman Riwayat Return Penjualan

The screenshot shows the ATTAYA MOTOR application interface. On the left is a sidebar with a user profile (Joenal Abidin, Admin), navigation links (Dashboard, Barang, Barang Masuk, Return Barang, Penjualan, Pembelian, Laporan Stok, Laporan Keuangan, Kontak), and Pengaturan. The main content area has a header with a home icon, Logout button, and search bar ('Cari Data Return'). Below is a table titled 'Data Return Penjualan' with columns: Id Return Penjualan, Id Transaksi, Nama Barang, Nama Pembeli, Tanggal Retrun. A single row is displayed: 001, 0005, oli,kampas rem, kiki, 15-5-2024. Buttons '+ Tambah Data Return' and 'Detail Return' are also present.

Query:

```
1 SELECT
2     rp.id_return_penjualan,
3     rp.id_transaksi,
4     GROUP_CONCAT(b.nama_barang) AS nama_barang,
5     p.nama_pembeli,
6     rp.tanggal_return
7 FROM
8     return_penjualan rp
9 JOIN
10    detail_return_penjualan drp ON rp.id_return_penjualan = drp.id_return_penjualan
11 JOIN
12    barang b ON drp.id_barang = b.id_barang
13 JOIN
14    pembeli p ON rp.id_pembeli = p.id_pembeli
15 GROUP BY
16     rp.id_return_penjualan,
17     rp.id_transaksi,
18     p.nama_pembeli,
19     rp.tanggal_return;
```

Halaman Detail Return Penjualan

Id Return Penjualan	Id Detail Return Penjualan	Id Transaksi	Nama Barang	Nama Pembeli	Tanggal Return	Jumlah	Aksi
001	001	0005	Oli	kiki	15-5-2024	3	
001	002	0005	Kampas rem	kiki	15-5-2024	2	

Query:

```
1 SELECT
2     rp.id_return_penjualan,
3     drp.id_detail_return_penjualan,
4     rp.id_transaksi,
5     b.nama_barang,
6     p.nama_pembeli,
7     rp.tanggal_return,
8     drp.jumlah_return
9 FROM
10     return_penjualan rp
11 JOIN
12     detail_return_penjualan drp ON rp.id_return_penjualan = drp.id_return_penjualan
13 JOIN
14     barang b ON drp.id_barang = b.id_barang
15 JOIN
16     pembeli p ON rp.id_pembeli = p.id_pembeli
17 WHERE
18     rp.id_return_penjualan = 'id_return_penjualan_yang_dicari'
19     OR drp.id_detail_return_penjualan = 'id_detail_return_penjualan_yang_dicari'
20     OR rp.id_transaksi = 'id_transaksi_yang_dicari'
21     OR b.nama_barang LIKE '%nama_barang_yang_dicari%'
22     OR p.nama_pembeli LIKE '%nama_pembeli_yang_dicari%'
23     OR rp.tanggal_return = 'tanggal_return_yang_dicari'
24     OR drp.jumlah_return = jumlah_return_yang_dicari;
```

Halaman Edit Return penjualan

The screenshot shows the 'Edit Return penjualan' page of the ATTAYA MOTOR application. The page has a header with the logo and user information ('Logout'). On the left is a sidebar with various menu items. The 'Penjualan' item is currently selected. The main content area contains a form titled 'Edit Return Penjualan' with four input fields: 'ID Barang', 'ID Pembeli', 'ID Transaksi', and 'Jumlah Return'. A blue 'Simpan' button is at the bottom of the form.

Query;

```
1 UPDATE return_penjualan
2 SET
3     id_pembeli = CASE WHEN '2' = '' THEN id_pembeli ELSE '2' END,
4     id_transaksi = CASE WHEN '10' = '' THEN id_transaksi ELSE '10' END
5 WHERE id_return_penjualan = '1';
6
7 UPDATE detail_return_penjualan
8 SET
9     id_barang = CASE WHEN '1' = '' THEN id_barang ELSE '1' END,
10    jumlah_return = CASE WHEN '10' = '' THEN jumlah_return ELSE '10' END
11 WHERE id_detail_return_penjualan = '1';
```

Halaman Search return penjualan

The screenshot shows the ATTAYA MOTOR software interface. On the left is a sidebar with a user profile (Joenal Abidin, Admin) and navigation links: Dashboard, Barang, Barang Masuk, Return Barang, Penjualan (selected), Pembelian, Laporan Stok, Laporan Keuangan, Kontak, and Pengaturan. The main content area has a header 'Detail Return Penjualan' and a search bar with placeholder 'Data Return Penjualan' and a search icon. A table titled 'Data Return Penjualan' displays two rows of data:

ID Return Penjualan	ID Detail Return Penjualan	ID Transaksi	Nama Barang	Nama Pembeli	Tanggal Return	Jumlah	Aksi
001	001	0005	Oli	kiki	15-5-2024	3	
001	002	0005	Kampas rem	kiki	15-5-2024	2	

Query:

```
1 SELECT
2     rp.id_return_pembelian,
3     drp.id_detail_return_pembelian,
4     b.nama_barang,
5     s.nama_supplier,
6     rp.tanggal_return,
7     drp.jumlah_return
8 FROM
9     return_pembelian rp
10 JOIN
11     detail_return_pembelian drp ON rp.id_return_pembelian = drp.id_return_pembelian
12 JOIN
13     barang b ON rp.id_barang = b.id_barang
14 JOIN
15     supplier s ON drp.id_supplier = s.id_supplier
16 WHERE rp.id_return_pembelian=1;
```

Halaman return pembelian

The screenshot shows the 'Return Pembelian' section of the ATTAYA MOTOR application. On the left is a sidebar with user information (Jaenot Abidin, Admin) and navigation links: Dashboard, Barang, Barang Masuk, Return Barang, Penjualan, Pembelian (which is selected), Laporan Stok, Laporan Keuangan, and Kontak. The main area has a header with a home icon and 'Logout'. Below the header are two tabs: 'Return Pembelian' and 'Riwayat'. The 'Return Pembelian' tab is active. It contains a form with fields: Id Barang (05), Id Karyawan (01), Id Supplier (01), Tanggal Return (16-5-2024). To the right is a table titled 'Detail Return Pembelian' with two rows:

Id Barang	Nama Barang	Jumlah
001	Oli	10
002	Kampas Rem	10

Below the table are buttons '+ simpan' and 'Tambah'.

Query;

```
1 INSERT INTO return_pembelian (id_barang, id_karyawan, tanggal_return)
2 VALUES (05, 01, CURRENT_DATE);
3
4 SET @last_id = LAST_INSERT_ID();
5
6 INSERT INTO detail_return_pembelian (id_supplier, id_return_pembelian, jumlah_return)
7 VALUES
8 (001, @last_id, 10),
9 (002, @last_id, 10);
10
11 SELECT
12     rp.id_barang,
13     b.nama_barang,
14     drp.jumlah_return
15 FROM
16     return_pembelian rp
17 JOIN
18     detail_return_pembelian drp ON rp.id_return_pembelian = drp.id_return_pembelian
19 JOIN
20     barang b ON rp.id_barang = b.id_barang
21 WHERE
22     rp.id_return_pembelian = @last_id;
23
```

Halaman Riwayat return pembelian

Id Return Penjualan	Nama Barang	Nama Pembeli	Tanggal Retrun
001	oll,kampas rem	kiki	15-5-2024

Query:

```
1  SELECT
2    rp.id_return_pembelian,
3    GROUP_CONCAT(b.nama_barang SEPARATOR ', ') AS nama_barang,
4    p.nama_pembeli,
5    rp.tanggal_return
6  FROM
7    return_pembelian rp
8  JOIN
9    detail_return_pembelian drp ON rp.id_return_pembelian = drp.id_return_pembelian
10 JOIN
11   barang b ON rp.id_barang = b.id_barang
12 JOIN
13   transaksi t ON rp.id_return_pembelian = t.id_transaksi
14 JOIN
15   pembeli p ON t.id_pembeli = p.id_pembeli
16 GROUP BY
17   rp.id_return_pembelian,
18   p.nama_pembeli,
19   rp.tanggal_return;
```

Halaman Detail return pembelian

ID Return Pembelian	ID Detail Return Pembelian	Nama Barang	Nama Supplier	Tanggal Return	Jumlah	Aksi
001	001	Oli	kiki	15-5-2024	10	
001	002	Kampas rem	kiki	15-5-2024	10	

Query;

```
1 SELECT
2     rp.id_return_pembelian,
3     drp.id_detail_return_pembelian,
4     b.nama_barang,
5     s.nama_supplier,
6     rp.tanggal_return,
7     drp.jumlah_return
8 FROM
9     return_pembelian rp
10 JOIN
11     detail_return_pembelian drp ON rp.id_return_pembelian = drp.id_return_pembelian
12 JOIN
13     barang b ON rp.id_barang = b.id_barang
14 JOIN
15     supplier s ON drp.id_supplier = s.id_supplier
16 WHERE rp.id_return_pembelian=1;
```

Halaman update return pembelian

The screenshot shows a software interface for managing returns. On the left is a sidebar with a user profile for 'Jaenal Abidin Admin' and a navigation menu including 'Dashboard', 'Barang', 'Barang Masuk', 'Return Barang', 'Penjualan', 'Pembelian' (which is highlighted in teal), 'Laporan Stok', 'Laporan Keuangan', and 'Kontak'. At the bottom of the sidebar is a 'Pengaturan' button. The main area has a dark header with a home icon and a 'Logout' button. Below the header is a title 'Edit Return Pembelian'. A sub-section titled 'Return Pembelian' contains three input fields: 'ID Barang', 'ID Supplier', and 'Jumlah Return'. A blue 'Simpan' button is positioned below these fields.

Query:

```
1 UPDATE detail_return_pembelian
2 SET
3     id_supplier = CASE WHEN '2' = '' THEN id_supplier ELSE '2' END,
4     jumlah_return = CASE WHEN '10' = '' THEN jumlah_return ELSE '10' END
5 WHERE id_return_pembelian = '1';
6
7 UPDATE return_pembelian
8 SET
9     id_barang = CASE WHEN '1' = '' THEN id_barang ELSE '1' END
10 WHERE id_return_pembelian = '1';
```

The screenshot shows the ATTAYA MOTOR application interface. On the left is a dark sidebar menu with the following items:

- Dashboard
- Barang
- Barang Masuk
- Return Barang
- Penjualan
- Pembelian (selected)
- Laporan Stok
- Laporan Keuangan
- Kontak

On the right, the main content area displays the "Detail Return Pembelian" page. At the top, there is a search bar labeled "Data Return Pembelian" with the value "001" and a magnifying glass icon. Below the search bar is a table with the following data:

ID Return Pembelian	ID Detail Return Pembelian	Nama Barang	Nama Supplier	Tanggal Return	Jumlah	Aksi
001	001	Oli	kiki	15-5-2024	10	

Query;

```
1 SELECT
2     rp.id_return_pembelian,
3     drp.id_detail_return_pembelian,
4     b.nama_barang,
5     s.nama_supplier,
6     rp.tanggal_return,
7     drp.jumlah_return
8 FROM
9     return_pembelian rp
10 JOIN
11     detail_return_pembelian drp ON rp.id_return_pembelian = drp.id_return_pembelian
12 JOIN
13     barang b ON drp.id_barang = b.id_barang
14 JOIN
15     supplier s ON b.id_supplier = s.id_supplier
16 WHERE
17     rp.id_return_pembelian = 'id_return_pembelian_yang_dicari'
18     OR drp.id_detail_return_pembelian = 'id_detail_return_pembelian_yang_dicari'
19     OR b.nama_barang LIKE '%nama_barang_yang_dicari%'
20     OR s.nama_supplier LIKE '%nama_supplier_yang_dicari%'
21     OR rp.tanggal_return = 'tanggal_return_yang_dicari'
22     OR drp.jumlah_return = jumlah_return_yang_dicari;
```

20

PROYEK APLIKASI PERANCANGAN & IMPLEMENTASI DATABASE

Laporan ini memuat 20 judul aplikasi yang disertai dengan deskripsi, permasalahan yang dihadapi, dan solusi aplikasi yang dirancang untuk mengatasinya. Setiap aplikasi dilengkapi dengan detail pengguna, desain Entity-Relationship Diagram (ERD), desain relasi basis data, query Data Definition Language (DDL), serta desain interface/wireframe/mockup yang menarik beserta query-nya.

- **Solusi Praktis dan Siap Pakai.** Setiap contoh aplikasi dirancang untuk diimplementasikan langsung ke dalam proyek.
- **Desain yang Mudah Dipahami.** Diagram, relasi, dan wireframe melalui setiap langkah dengan mudah.
- **Panduan Step by Step** Dari identifikasi masalah hingga solusi, membantu memahami keseluruhan proses perancangan database.
- **Kumpulan query.** mulai dari DDL, DML yang efektif untuk membangun dan mengelola basis data.

Laporan dapat dijadikan sebagai referensi bagi mahasiswa, pengembang, dan profesional IT, dan akan membantu keterampilan perancangan database ke tingkat lanjut.

JIKA ADA SUATU PELUANG PENTING,
KAMU HARUS MENCOBANYA
MESKIPUN KEMUNGKINAN KAMU AKAN GAGAL”
- ELON MUSK