

PROYEK AKHIR
ALGORITMA DAN PEMROGRAMAN II



APLIKASI “MakeOvers”

**PROGRAM PENCATATAN RESERVASI NEAN SALON 2 DI JEMBER DENGAN
MENGUNAKAN ALGORITMA SORTING INSERTION SORT PADA STOK
BARANG DAN SEARCHING INTERPOLASI PADA JADWAL LAYANAN
RESERVASI**

TIM PENYUSUN:

KELOMPOK 9

- | | |
|-------------------------------|----------------|
| 1. Ariski Ade Raharjo | (232410101015) |
| 2. Rendy Nayogi Pramudya | (232410101016) |
| 3. Azzahro Alfi Nabila | (232410101024) |
| 4. Fachrio Ferdinandz Firdaus | (232410101043) |
| 5. Muhammad Nabil Baihaqi | (232410101046) |

FAKULTAS ILMU KOMPUTER

TAHUN AKADEMIK
2024

DAFTAR ISI

BAB I

PENDAHULUAN

1.1 Latar Belakang

Nean Salon merupakan sebuah salon di Surabaya yang menyediakan berbagai macam jasa untuk perawatan badan. Nean salon baru saja memiliki cabang yaitu Nean salon 2 yang berada di Kabupaten Jember, Jawa Timur Alamat : JL.Basuki Rahmat No.220, Muktisari, Tegal Besar, Kec. Kaliwates, Kabupaten Jember, Jawa Timur 68131. Nean salon 2 menawarkan berbagai macam layanan seperti potong rambut, *creambath*, *smoothing*, *coloring*, *facial whitening*, *eyelash*, *nail art*, serta bekam. Dengan berbagai jenis pelayanan yang tersedia pada Nean Salon 2, pegawai yang ada harus melayani banyak pelanggan setiap harinya.

Nean Salon 2 yang baru-baru ini beroperasi masih belum memiliki sistem penyimpanan data yang baik. Beberapa permasalahan yang dialami selama salon beroperasi adalah pencatatan transaksi masih secara manual dan belum terstruktur, sistem suplai produk yang tidak pasti, serta sistem reservasi yang tidak terkoordinir dengan baik. Dampak dari permasalahan tersebut menyebabkan transaksi yang telah dilakukan tidak tercatat dengan baik, terjadi pembelian ulang terhadap stok yang masih ada, serta terjadinya jadwal reservasi yang tidak tercatat. Dari masalah tersebut, maka diperlukan sebuah sistem informasi yang dapat menampung data-data penting yang terjadi pada salon sehingga segala peristiwa yang terjadi di salon dapat dicatat dan disimpan dengan baik.

Oleh karena itu, kami membuat sebuah aplikasi bernama “MakeOvers” yang akan memberikan solusi dari permasalahan-permasalahan yang ada pada nean salon 2, yaitu Sistem Informasi Manajemen Transaksi Nean Salon 2. Dengan sistem informasi manajemen transaksi ini, berbagai kegiatan pencatatan dan transaksi yang ada akan lebih terstruktur dan mudah dijalankan. Sistem ini memiliki beberapa fitur seperti,

fitur login-register, fitur reservasi, fitur riwayat pembayaran fitur stok produk, serta fitur harga layanan salon.

1.2 Tujuan

Tujuan dari dibuatnya Aplikasi MakeOvers ini adalah:

1. Menyediakan jasa layanan salon dan jadwal layanannya yang dapat dipeservasi oleh Customer guna mempermudah mendapatkan layanan yang tepat dan waktu yang diinginkan.
2. Bentuk implementasi perkembangan teknologi untuk jasa salon khususnya di Nean Salon 2 Jember.
3. Mengganti pencatatan manual yang tidak terstruktur dengan sistem yang lebih rapi dan efisien (pembayaran).
4. Membantu para Admin dan User dalam melakukan transaksi yang lebih efisien lewat Aplikasi MakeOvers.
5. Mempermudah operasional harian salon dengan fitur login-register, reservasi, riwayat pembayaran, stok produk, dan harga layanan.

1.3 Batasan Masalah

1. Aplikasi ini merupakan aplikasi berbasis mobile (handphone).
2. Aplikasi ini memerlukan akses internet dan dapat digunakan pada *system* operasi Android dan Windows
3. Aplikasi ini mencakup seluruh wilayah Indonesia khususnya di Jawa Timur pada daerah Jember yang ingin mereservasi layanan pada Nean Salon 2.
4. Aplikasi ini memberikan informasi standarisasi pada layanan Nean Salon 2 terhadap jasa layanan serta jadwal yang tersedia.

BAB II

RANCANGAN APLIKASI

2.1 Kebutuhan Sistem

User

1. Aplikasi ini memiliki fitur “*Register*” sebagai autentikasi pengguna.
2. Aplikasi ini memiliki fitur “*Login*” sebagai autentikasi pengguna.
3. Aplikasi ini memiliki fitur “*Reservasi*” untuk memesan, melihat daftar pelayanan dan jadwal layanan yang dipesan di salon pada “*Menu Reservasi*” .
4. Aplikasi ini memiliki fitur “*Reservasi*” untuk melakukan pembayaran layanan salon pada “*Menu Pembayaran*” berdasarkan jenis pembayaran yang dipilih user.
5. Aplikasi ini memiliki fitur pembayaran untuk layanan yang telah dipesan
6. Aplikasi ini memiliki fitur “*Riwayat Pembayaran*” untuk melihat pemesanan

Admin

1. Aplikasi ini memiliki fitur “*Login*” sebagai autentikasi pengguna
2. Aplikasi ini memiliki fitur “*Reservasi*” untuk melihat layanan dan jadwal yang dipesan oleh user pada *Menu Reservasi* di user tersebut.
3. Aplikasi ini memiliki fitur “*Reservasi*” untuk menambah dan menghapus jadwal reservasi.
4. Aplikasi ini memiliki fitur “*Stok*” untuk melihat dan mengedit stok produk pada salon.
5. Aplikasi ini memiliki fitur “*Pemesanan*” untuk menginputkan pesanan pada user yang membayar dengan metode cash
6. Aplikasi ini memiliki fitur “*Riwayat Pembayaran*” untuk melihat seluruh riwayat pembayaran pada user.

2.2 Struktur Data

Di bawah ini merupakan struktur dari semua data dan tipe data dari kode yang digunakan dalam pembuatan program MAKEOVERS :

USER

No	Fitur (User)	Data	Tipe Data
1.	Register	Nama No.Hp Email Username Password	String String String String String
2.	Login	Username Password	String String
3.	Reservasi	Menu Layanan - No - Nama layanan - Harga Layanan Menu Reservasi - No - Pukul - Hari - Tanggal	String Integer String String String Integer Time String String
4.	Pembayaran	Menu Pembayaran - Metode Bank 1. BRI 2. BNI 3. BCA Detail Pembayaran - Nama - No.Rek - Nama Bank - Layanan - Harga - Metode Ewallet 1. DANA 2. OVO 3. SHOPEEPAY Detail Pembayaran - Nama	String String String String String String String Integer String String String String String String String String

		- No.Hp - Nama E-Wallet - Layanan - Harga	String String String Integer
5.	Riwayat Pembayaran	Jenis Pembayaran Layanan Total Harga Tanggal Pembayaran	String String Integer Timestamp

ADMIN

No	Fitur (User)	Data	Tipe Data
1.	Login (admin)	Username Password	String String
2.	Reservasi	Cek Reservasi - Nama - Pukul - Hari - Tanggal Reservasi Edit Jadwal Reservasi - Tambah - Hapus	String Integer String Integer String Integer String String
3.	Stok Barang	Edit Menu Stok - Tambah - Hapus	String String String
4.	Pemesanan (Cash)	Menu Layanan - No - Nama layanan - Harga Layanan Menu Reservasi - No - Pukul - Hari - Tanggal	String Integer String String String Integer Time String String
5.	Riwayat Pembayaran	Jenis Pembayaran Layanan Total Harga Tanggal Pembayaran	String String Integer Integer

2.3 Rancangan Langkah Penggunaan Aplikasi

2.3.1 Program

Pada rancangan pengguna aplikasi, dibutuhkan langkah-langkah yang harus dilakukan oleh user dan admin dalam mengoperasikan program aplikasi “MakeOvers” ini. Berikut adalah rancangan penggunaan aplikasinya.

USER

1. Ketika program dijalankan pertama kali, program akan menampilkan 3 menu yaitu Login, Register dan Admin. Jika User belum memiliki akun untuk login, maka diarahkan untuk memilih menu Register terlebih dahulu pada no 2.
2. Setelah user memilih menu pada no 2 register maka akan diarahkan untuk mengisi data-data yakni nama, no.hp, email, username dan password kemudian mengkonfirmasi apakah data-data yang diinputkan sudah benar apa tidak. Jika sudah benar maka pilih no 1 yaitu konfirmasi dan akun sudah berhasil ‘register’. Apabila pilih no 2 yaitu kembali maka user akan kembali ke menu awal tadi. *Noted* untuk menghindari kesamaan username dan password maka terdapat pengecualian juga ‘Username telah digunakan! Enter untuk Kembali’ yang artinya menginputkan data kembali sesuai data yang sudah di register sebelumnya dengan benar. (data akan disimpan pada database di tabel akun)
3. Setelah selesai register, user akan diarahkan lagi ke fitur login yang dimana menginputkan data-data tadi yang sudah teregister yaitu “username dan password” sebagai autentikasi pengguna. Jika username dan password yang dimasukkan salah, maka ada pemberitahuan ‘Login gagal!’, periksa kembali username dan password dan meminta untuk opsi ulangi atau kembali. Namun sebaliknya apabila berhasil maka akan lanjut ke menu layanan pada fitur reservasi.
4. Kemudian user diarahkan kepada **menu layanan**, dimana user akan melanjutkan untuk memilih no 1 yaitu ‘pesan’ atau no 2 ‘kembali. Ketika user memilih no 1 maka akan diarahkan ke **menu pesan layanan** yang berisikan list daftar layanan seperti: No, Nama layanan, dan Harga Layanan tersebut. Dan user akan memilih layanan apa yang

ingin dia pesan di Aplikasi MakeOvers ini. Sebagai contoh no 2, nama layanannya creambath dan untuk harganya yaitu Rp.60.000.

5. Lanjut setelah memilih jenis layanan, user akan diarahkan ke **menu reservasi** yang dimana akan menampilkan jadwal layanan yang telah user pilih berdasarkan list jadwal layanan seperti: No, Pukul, Hari, dan Tanggal. Sebagai contoh pada no 3, pukul 10:00 wib, hari selasa, pada tanggal 28 Mei 2024. Maka terdapat persetujuan no 1 yakni konfirmasi reservasi dan no 2 kembali, jika user mengkonfirmasi maka data akan tersimpan pada database tabel layanan dan reservasi jika kembali maka reservasi tersebut dibatalkan dan kembali ke menu awal.
6. Setelah reservasi dilakukan, kemudian user akan diarahkan ke fitur pembayaran pada **menu pembayaran** untuk melanjutkan transaksi yang sudah di reservasi tadi sebelumnya pada fitur reservasi. Ketika dalam menu ini user akan memilih menggunakan metode apa dia akan membayar. Pada aplikasi MakeOvers ini menyediakan 2 metode yaitu 1. Metode Bank, dan 2. Metode E-Wallet, nah ketika semisal user ingin membayar menggunakan metode bank maka dia memilih no 1 dan selanjutnya akan diarahkan ke metode bank dan memilih jenisnya yaitu BRI, BNI, atau BCA. Ketika user memilih menggunakan bank BRI, maka diarahkan untuk menginputkan no rekeningnya dan nama layanan tadi. Jika sudah tercetak detail pembayarannya maka user diminta untuk mengkonfirmasi pada pilihan 1 atau batal pada pilihan 2. Ketika ingin melanjutkan memilih no 1 maka akan muncul status 'pembayaran berhasil' dan data akan disimpan di database pada tabel pembayaran.
7. Kemudian yang terakhir user bisa melihat transaksi reservasi tadi pada fitur riwayat pembayaran yang terdiri dari data: jenis pembayaran, layanan, total harga dan tanggal pembayaran yang disimpan pada database pada tabel riwayat. Program telah selesai dan user bisa kembali ke menu awal.

ADMIN

1. Berikutnya untuk admin, Ketika program dijalankan pertama kali, program akan menampilkan 3 menu yaitu Login, Register dan Admin. maka diarahkan untuk memilih menu 'admin' terlebih dahulu pada no 3 untuk melakukan login.
2. Kemudian admin diminta memasukan username dan password yang telah didapat pada instansi nya tersebut dan mengkonfirmasi lagi. Ketika sudah login maka admin akan diarahkan kepada fitur reservasi.
3. Pada fitur reservasi terdapat 2 menu yaitu **menu pesan layanan (jenis-jenis layanan) dan menu reservasi (jadwal layanannya)**. Admin bisa melihat dan memantau pada kedua menu tersebut dan juga dapat mengedit reservasi nya berdasarkan jadwal atau layanan jika terdapat admin(pegawai) yang sedang berhalangan untuk melakukan layanan tersebut dan mengganti jadwal serta layanannya. (data akan disimpan di database pada tabel detail layanan).
4. Kemudian terdapat fitur stok yang dimana admin dapat melihat stok barang berdasarkan layanannya serta bisa mengedit stok tersebut (menambah, mengurangi dan hapus). (data akan disimpan di database pada tabel stok).
5. Kemudian terdapat fitur pemesanan yang dimana admin bisa menginputkan data-data user yang ingin memesan layanan salon dan membayar dengan metode cash.
6. Terakhir terdapat fitur riwayat pembayaran pada admin untuk melihat data-data transaksi dari user yang akan disimpan pada database detail reservasi.

2.3.2 Flowchart

Untuk Diagram alir pada aplikasi "MakeOvers"

USER

ADMIN

BAB III

IMPLEMENTASI RANCANGAN

Berikut ini adalah penjelasan kode program yang akan kami gunakan untuk menyusun aplikasi “MakeOvers” yang kami rancang :

USER

3.1 Fitur Register

A screenshot of a code editor with a dark background and light-colored text. The code is written in Python and defines a function named 'menu()'. The function starts with a 'while' loop that continues as long as a variable 'kondisi' is True. Inside the loop, it clears the terminal screen, prints a welcome message, and displays a menu with four options: LOGIN, REGISTER, ADMIN, and KELUAR. It then prompts the user to make a choice. A 'match' statement handles the user's input: '1' leads to 'login()', '2' leads to 'register()', and '3' leads to 'admin()'. After each of these actions, 'kondisi' is set to False. Input '4' leads to clearing the screen, printing a thank-you message, and setting 'kondisi' to False. Input for any other value does not change 'kondisi', so the loop repeats.

```
1 def menu():
2     kondisi = True
3     while kondisi :
4         os.system('cls')
5         print("="*100)
6         print("WELCOME TO MAKEOVERS".center(100))
7         print("="*100)
8         print("[1] LOGIN\n[2] REGISTER\n[3] ADMIN\n[4] KELUAR ")
9         pilihan = input('Pilian : ')
10        match pilihan :
11            case '1' :
12                login()
13                kondisi = False
14            case '2':
15                register()
16            case '3' :
17                admin()
18                kondisi = False
19            case '4':
20                os.system('cls')
21                print("="*100)
22                print("TERIMA KASIH TELAH MENGGUNAKAN MAKESOVERS".center(100))
23                print("="*100)
24                kondisi = False
```

Gambar 3.1.1 *Source Code* Pilihan Menu untuk User dan Admin

```

1 def register():
2     kondisi = True
3     while kondisi:
4         os.system('cls')
5         print("="*100)
6         print("REGISTER".center(100))
7         print("="*100)
8         nama = input("Nama : ")
9         nomer_hp = input("Nomer HP : ")
10        email = input("Email : ")
11        username = input("Username : ")
12        password = input("Password : ")
13        print("Tekan Enter Untuk Mengisi Ulang Data".center(100))
14        print("[1] Konfirmasi \n[2] Kembali")
15        opsi = input("Pilihan : ")
16        match opsi:
17            case '1':
18                if login_register.register(nama, nomer_hp, email, username, password) == True:
19                    print("="*100)
20                    print("REGISTER BERHASIL, ENTER UNTUL KEMBALI".center(100))
21                    print("="*100)
22                    input()
23                    kondisi = False
24            else:
25                print("="*100)
26                print("USERNAME TELAH DIGUNAKAN! ENTER UNTUK KEMBALI".center(100))
27                print("="*100)
28                input()
29            case '2':
30                break
31

```

Gambar 3.1.2 *Source Code Akses Register User*

```

1 def register(nama, nomorhp, email, username, password):
2     conn = psycopg2.connect(dbname=database, user=userdatabase, password=passworddatabase)
3     cur = conn.cursor()
4
5     check_sql = "SELECT 1 FROM akuncus WHERE username = %s"
6     cur.execute(check_sql, [username])
7     cek = cur.fetchone()
8     if cek:
9         cur.close()
10        conn.close()
11        return False
12
13    insert_sql = "INSERT INTO akuncus (nama, no_hp, email, username, password) VALUES (%s, %s, %s, %s, %s)"
14    cur.execute(insert_sql, (nama, nomorhp, email, username, password))
15    conn.commit()
16    cur.close()
17    conn.close()
18    return True

```

Function (Register User)

Di atas merupakan *source code* dari Pilihan Menu untuk User dan Admin yang berfungsi untuk autentikasi login pada aplikasi “MakeOvers”. Jadi pertama dari user sebelum login jika tidak memiliki akun maka diarahkan ke menu register untuk bisa mendapatkan data-data (username dan password) sebagai syarat login pada aplikasi ini.

3.2 Fitur Login

```

1 def login():
2     os.system('cls')
3     print("="*100)
4     print("LOGIN".center(100))
5     print("="*100)
6     username = input('Username : ')
7     password = input('Password : ')
8     kondisi = True
9     while kondisi:
10        if login_register.login(username,password) == True :
11            id_user = login_register.user(username,password)
12            pembayaran.riwayat_pembayaran_user(id_user[0])
13            os.system('cls')
14            print("="*100)
15            print("MENU MAKEOVERS".center(100))
16            print("="*100)
17            print('[1] Reservasi\n[2] Riwayat Pemabayaran\n[3] Keluar')
18            pilihan_menu = input('Pilih : ')
19            match pilihan_menu :
20                case '1' :
21                    reservasi_user(username,password)
22                case '2':
23                    riwayat_transaksi(id_user[0])
24                case '3':
25                    menu()
26            else :
27                os.system('cls')
28                print("="*100)
29                print("LOGIN".center(100))
30                print("="*100)
31                print('Login gagal!, periksa Kembali username dan password'.center(100))
32                print("[1] Ulangi\n[2] Kembali")
33                opsi = input("Pilihan : ")
34                os.system("cls")
35                if opsi == '1' :
36                    login()
37                    kondisi = False
38                elif opsi == '2' :
39                    menu()
40                    kondisi = False
41                else :
42                    login()

```

Gambar 3.2.1 *Source Code* Akses Login User

```

1 def login(username:str, password:str):
2     conn = psycopg2.connect(dbname=database, user=userdatabase, password=passworddatabase)
3     cur = conn.cursor()
4     sql = f"SELECT * FROM akuncus WHERE username = '{username}' AND password = '{password}'"
5     cur.execute(sql)
6     login = cur.fetchone()
7     if login :
8         cur.close()
9         conn.close()
10        return True
11    else :
12        return False

```

Function (Login User)

Kode di atas merupakan tampilan akses login untuk user dari fitur kedua yaitu login, yakni fitur ini bertujuan untuk user agar bisa mengakses aplikasi MakeOvers yang telah register tadi pada fitur pertama. Pada *Source Code* ini juga ketika user salah menginputkan (username serta password) maka dia akan ada pemberitahuan ‘ulangi’ atau ‘kembali’ guna memastikan kebenarannya.

3.3 Fitur Reservasi

```

1 def reservasi_user(username,password) :
2     kondisi = True
3     while kondisi :
4         os.system('cls')
5         print("="*100)
6         print("MENU LAYANAN".center(100))
7         print("="*100)
8         print(tabulate(layanan.layanan_user(), headers=['No', ' Nama Layanan', 'Harga Layanan'],tablefmt="double_grid"))
9         print("[1] Pesan\n[2] Kembali")
10        pilih = input('Pilih : ')
11        match pilih :
12            case 1 :
13                os.system('cls')
14                print("="*100)
15                print("MENU PESAN LAYANAN".center(100))
16                print("="*100)
17                print(tabulate(layanan.layanan_user(), headers=['No', ' Nama Layanan', 'Harga Layanan'],tablefmt="double_grid"))
18                pilih_layanan = input('Pilih Berdasarkan No : ')
19                while kondisi :
20                    os.system('cls')
21                    print("="*100)
22                    print("MENU RESERVASI".center(100))
23                    print("="*100)
24                    print (tabulate(reservasi.menampilkan_reservasi(), headers=['No','Pukul','Hari','Tanggal'],tablefmt="double_grid"))
25                    Pili_reservasi = input('Pilih ')
26                    print("[1] konfirmasi reservasi | [2] kembali".center(100))
27                    inputan = input('Pilih : ')

```

Gambar 3.3.1 *Source Code* Reservasi (Menu Layanan dan Menu Reservasi)

```

1 def layanan_user ():
2     conn = psycopg2.connect(dbname=database, user=userdatabase, password=passworddatabase)
3     cur = conn.cursor()
4     sql = f"SELECT * FROM layanan"
5     cur.execute(sql)
6     layanan_user = cur.fetchall()
7     cur.close()
8     conn.close()
9     return layanan_user

```

Function Reservasi - Menu Layanan

```

1 def menampilkan_reservasi ():
2     conn = psycopg2.connect(dbname=database, user=userdatabase, password=passworddatabase)
3     cur = conn.cursor()
4     sql = f"select id_reservasi, pukul,hari,tanggal from reservasi where id_akuncus is null"
5     cur.execute(sql)
6     reservasi_user = cur.fetchall()
7     cur.close()
8     conn.close()
9     return reservasi_user

```

Function Reservasi - Menu Reservasi

Di atas merupakan *Source Code* dari fitur reservasi pada user. Dengan menginputkan pesan pada menu layanan maka user akan diarahkan ke “Menu Pesan Layanan” untuk memilih jenis layanan salon apa yang ingin di reservasi (pada ini menampilkan tabel daftar layanan beserta harganya). Kemudian user akan diarahkan lagi ke “Menu Reservasi” untuk menentukan jadwal layanan yang tersedia pada aplikasi ini.

3.4 **Fitur Pembayaran**

```

1 match inputan :
2
3     case '1' :
4         while kondisi :
5             os.system('cls')
6             print("="*100)
7             print("MENU PEMBAYARAN".center(100))
8             print("="*100)
9             print('[1] Metode Bank\n[2] Metode E-wallet\n[3] Kembali')
10            metode = input('Pilih Metode : ')
11            match metode :
12                case '1' :
13                    while kondisi :
14                        os.system('cls')
15                        print("="*100)
16                        print("METODE BANK".center(100))
17                        print("="*100)
18                        id_user = login_register.user(username,password)
19                        print('[1] BRI\n[2] BNI\n[3] BCA\n[4] KEMBALI')
20                        jenis_metode = input ('pilih : ')
21                        match jenis_metode :
22                            case '1':
23                                nama_bank = 'BRI'
24                                pembayaran_bank_user(id_user,nama_bank,pili_layanan,Pili_reservasi)
25                                break
26                            case '2' :
27                                nama_bank = 'BNI'
28                                pembayaran_bank_user(id_user,nama_bank,pili_layanan,Pili_reservasi)
29                                break
30                            case '3' :
31                                nama_bank = 'BCA'
32                                pembayaran_bank_user(id_user,nama_bank,pili_layanan,Pili_reservasi)
33                                break
34                            case '4' :
35                                break

```

```

1 case '2':
2
3     while kondisi :
4         os.system('cls')
5         print("="*100)
6         print("METODE E-WALLET".center(100))
7         print("="*100)
8         id_user = login_register.user(username,password)
9         print('[1] DANA\n[2] OVO\n[3] SHOPEEPAY\n[4] KEMBALI')
10        jenis_metode = input ('pilih : ')
11        match jenis_metode :
12            case '1':
13                nama_ewallet = 'DANA'
14                pembayaran_ewallet_user(id_user,nama_ewallet,pili_layanan,Pili_reservasi)
15                break
16            case '2' :
17                nama_ewallet = 'OVO'
18                pembayaran_ewallet_user(id_user,nama_ewallet,pili_layanan,Pili_reservasi)
19                break
20            case '3' :
21                nama_ewallet = 'SHOPEEPAY'
22                pembayaran_ewallet_user(id_user,nama_ewallet,pili_layanan,Pili_reservasi)
23                break
24            case '4' :
25                break
26        break
27    case '3' :
28        break
29
30    break
31
32
33    case '2':
34        break
35
36    case '2':
37        kondisi = False

```

Gambar 3.4.1 *Source Code* Fitur Pembayaran (Menu dan Metode Pembayaran)

```

1 def pembayaran_bank_user(id_user,nama_bank,pili_Layanan,Pili_reservasi):
2     kondisi = True
3     while kondisi:
4         os.system('cls')
5         print("="*100)
6         print("METODE BANK".center(100))
7         print("="*100)
8         rekening = input('Rekening : ')
9         nama_layanan = layanan.detail_layanan(pili_Layanan)[1]
10        os.system('cls')
11        print("="*100)
12        print("DETAIL PEMBAYARAN".center(100))
13        print("="*100)
14        print(f'NAMA           : {id_user[1]}')
15        print(f'NO REKENING      : {rekening}')
16        print(f'NAMA BANK        : {nama_bank}')
17        print(f'LAYANAN          : {nama_layanan}')
18        print(f'HARGA           : {pembayaran.nominal(pili_Layanan)[2]}')
19        print('[1] konfirmasi | [2] Batal | [Enter] Ulangi'.center(100))
20        pilihan = input('Pilih : ')
21        match pilihan :
22            case '1':
23                nominal = pembayaran.nominal(pili_Layanan)[2]
24                pembayaran.pembayaran_bank(nama_bank,rekening,id_user[0],nominal)
25                reservasi.detail_reservsi(id_user[0],Pili_reservasi)
26                pembayaran.tanggal_pembayaran_bank(id_user[0])
27                pembayaran.data_riwayat(nama_bank,id_user[0],nominal,nama_layanan)
28                pembayaran.tanggal_riwayat_pembayaran(id_user[0])
29                print("="*100)
30                print('PEMBAYARAN BERHASIL'.center(100))
31                print("="*100)
32                input()
33                break
34            case '2' :
35                kondisi = False

```

Gambar 3.4.2 *Source Code* Fitur Pembayaran Metode Bank

```

1 def pembayaran_ewallet_user(id_user,nama_ewallet,pili_Layanan,Pili_reservasi):
2     kondisi = True
3     while kondisi:
4         os.system('cls')
5         print("="*100)
6         print("METODE BANK".center(100))
7         print("="*100)
8         nomor = input('Nomor : ')
9         nama_layanan = layanan.detail_layanan(pili_Layanan)[1]
10        os.system('cls')
11        print("="*100)
12        print("DETAIL PEMBAYARAN".center(100))
13        print("="*100)
14        print(f'NAMA           : {id_user[1]}')
15        print(f'NOMOR          : {nomor}')
16        print(f'NAMA E-WALLET   : {nama_ewallet}')
17        print(f'LAYANAN          : {nama_layanan}')
18        print(f'HARGA           : {pembayaran.nominal(pili_Layanan)[2]}')
19        print('[1] konfirmasi | [2] Batal | [Enter] Ulangi'.center(100))
20        pilihan = input('Pilih : ')
21        match pilihan :
22            case '1':
23                nominal = pembayaran.nominal(pili_Layanan)[2]
24                pembayaran.pembayaran_ewallet(nama_ewallet,nomor,id_user[0],nominal)
25                reservasi.detail_reservsi(id_user[0],Pili_reservasi)
26                pembayaran.tanggal_pembayaran_ewallet(id_user[0])
27                pembayaran.data_riwayat(nama_ewallet,id_user[0],nominal,nama_layanan)
28                pembayaran.tanggal_riwayat_pembayaran(id_user[0])
29                print("="*100)
30                print('PEMBAYARAN BERHASIL'.center(100))
31                print("="*100)
32                input ()
33                break
34            case '2' :
35                kondisi = False

```

Gambar 3.4.3 *Source Code* Fitur Pembayaran Metode E-Wallet


```

1 def detail_layanan (id_Layanan):
2     conn = psycopg2.connect(dbname=database, user=userdatabase, password=passworddatabase)
3     cur = conn.cursor()
4     sql = f"SELECT * FROM layanan where id_layanan = {id_Layanan}"
5     cur.execute(sql)
6     detail_layanan_user = cur.fetchone()
7     cur.close()
8     conn.close()
9     return detail_layanan_user

```

Function Pembayaran (Detail Layanan)

```

1 def nominal (layanan):
2     conn = psycopg2.connect(dbname=database, user=userdatabase, password=passworddatabase)
3     cur = conn.cursor()
4     sql = f"SELECT * FROM layanan where id_layanan = '{layanan}' "
5     cur.execute(sql)
6     riwayat_user = cur.fetchone()
7     cur.close()
8     conn.close()
9     return riwayat_user

```

Function Pembayaran (Nominal)

```

1 def pembayaran_bank(nama_bank:str, rekening:str, id_user:int, total_harga:int) :
2     conn = psycopg2.connect(dbname=database, user=userdatabase, password=passworddatabase)
3     cur = conn.cursor()
4     sql = f"INSERT INTO bank(nama_bank, no_rekening,id_akuncus,nominal) VALUES('{nama_bank}','{rekening}','{id_user}','{total_harga}) "
5     cur.execute(sql)
6     conn.commit()
7     cur.close()
8     conn.close()

```

Function Pembayaran (Metode Bank)

```

1 def pembayaran_ewallet(nama_ewallet:str,nomor:str, id_user:int, total_harga:int) :
2     conn = psycopg2.connect(dbname=database, user=userdatabase, password=passworddatabase)
3     cur = conn.cursor()
4     sql = f"INSERT INTO ewallet(nama_ewallet, nomor,id_akuncus,nominal) VALUES('{nama_ewallet}','{nomor}','{id_user}','{total_harga}) "
5     cur.execute(sql)
6     conn.commit()
7     cur.close()
8     conn.close()

```

Function Pembayaran (Metode E-Wallet)

```

1 def tanggal_pembayaran_bank (id_user):
2     conn = psycopg2.connect(dbname=database, user=userdatabase, password=passworddatabase)
3     cur = conn.cursor()
4     sql = f"UPDATE bank SET tanggal_pembayaran = NOW() WHERE id_akuncus = {id_user}"
5     cur.execute(sql)
6     conn.commit()
7     cur.close()
8     conn.close()

```

Function Pembayaran (Tanggal Pembayaran Bank)

```
1 def tanggal_pembayaran_ewallet (id_user):
2     conn = psycopg2.connect(dbname=database, user=userdatabase, password=passworddatabase)
3     cur = conn.cursor()
4     sql = f"UPDATE ewallet SET tanggal_pembayaran = NOW() WHERE id_akuncus = {id_user}"
5     cur.execute(sql)
6     conn.commit()
7     cur.close()
8     conn.close()
9
```

Function Pembayaran (Tanggal Pembayaran E-Wallet)

Kode di atas merupakan tampilan dari fitur keempat yaitu pembayaran. Jadi pada source code ini pada **Gambar 3.4.1** itu user akan memilih jenis pembayaran apa yang diinginkan dan terdapat dua jenis yaitu Metode Bank dan Metode E-Wallet. Kemudian pada **Gambar 3.4.2** dan **Gambar 3.4.3** user akan memilih jika misal metode pembayaran menggunakan Bank, maka data yang sudah direservasi tersebut akan tersimpan di riwayat pembayaran jika transaksi sudah berjalan dengan lancar.

3.5 Fitur Riwayat Pembayaran

```
1 def riwayat_transaksi(id_user):
2     os.system('cls')
3     print("="*100)
4     print("RIWAYAT PEMBAYARAN".center(100))
5     print("="*100)
6     print(tabulate(pembayaran.riwayat_pembayaran_user(id_user), headers=['Jenis pembayaran', 'Layanan', 'Total Harga', 'Tanggal Pembayaran'], tablefmt='double_grid'))
7     input('\nENTER UNTUK KEMBALI !')
```

Gambar 3.5.1 *Source Code* Fitur Riwayat Pembayaran User

```

1 def riwayat_pembayaran_user(id_pengguna):
2     conn = psycopg2.connect(dbname=database, user=userdatabase, password=passworddatabase)
3     cur = conn.cursor()
4     sql = f"select jenis_pembayaran,layanan,nominal, TO_CHAR(tanggal, 'DD-MM-YYYY') as Tanggal from riwayat_pembayaran where id_akuncus = '{id_pengguna}' "
5     cur.execute(sql)
6     riwayat_user = cur.fetchall()
7     cur.close()
8     conn.close()
9     return riwayat_user
10
11 def tanggal_riwayat_pembayaran (id_user):
12     conn = psycopg2.connect(dbname=database, user=userdatabase, password=passworddatabase)
13     cur = conn.cursor()
14     sql = f"UPDATE riwayat_pembayaran SET tanggal = NOW() WHERE id_akuncus = {id_user}"
15     cur.execute(sql)
16     conn.commit()
17     cur.close()
18     conn.close()
19
20 def data_riwayat (nama_metode,id_user,nominal,nama_layanan):
21     conn = psycopg2.connect(dbname=database, user=userdatabase, password=passworddatabase)
22     cur = conn.cursor()
23     sql = f"insert into riwayat_pembayaran (jenis_pembayaran,id_akuncus,nominal,layanan) values ('{nama_metode}',{id_user},{nominal},{nama_layanan})"
24     cur.execute(sql)
25     conn.commit()
26     cur.close()
27     conn.close()
28

```

Function Riwayat Pembayaran, Tanggal Pembayaran, dan data Riwayat

Kode di atas merupakan tampilan *Source Code* dari fitur kelima yaitu Riwayat Pembayaran. Jadi pada fitur ini dari hasil transaksi pada fitur ke 3 dan 4 maka hasil detail transaksi user tersebut akan tersimpan pada tabel “Riwayat Pembayaran” untuk melihat transaksi yang sudah terjadi. Nantinya rincian data tersebut akan masuk ke database pada tabel riwayat dan program selesai dan kembali ke menu awal (login).

ADMIN

- 4.6 Fitur Login (Admin)**
- 4.7 Fitur Reservasi (Admin)**
- 4.8 Fitur Stok Barang (Admin)**
- 4.9 Fitur Pemesanan (Admin)**
- 4.10 Fitur Riwayat Pembayaran (Admin)**

BAB IV

HASIL DAN PEMBAHASAN

USER

- 4.1 Output Register (User)**
- 4.2 Output Login (User)**
- 4.3 Output Reservasi (User)**
- 4.4 Output Pembayaran (User)**
- 4.5 Output Riwayat Pembayaran (User)**

ADMIN

- 4.6 Output Login (Admin)**
- 4.7 Output Reservasi (Admin)**
- 4.8 Output Stok Barang (Admin)**
- 4.9 Output Pemesanan (Admin)**
- 4.10 Output Riwayat Pembayaran (Admin)**

LAMPIRAN

DAFTAR PUSTAKA