

SISTEM MANAJEMEN SERVICE BENGKEL RESMI HONDA

MUHAMMAD ALI PRATAMA PUTRA

5220411416

DAFTAR ISI

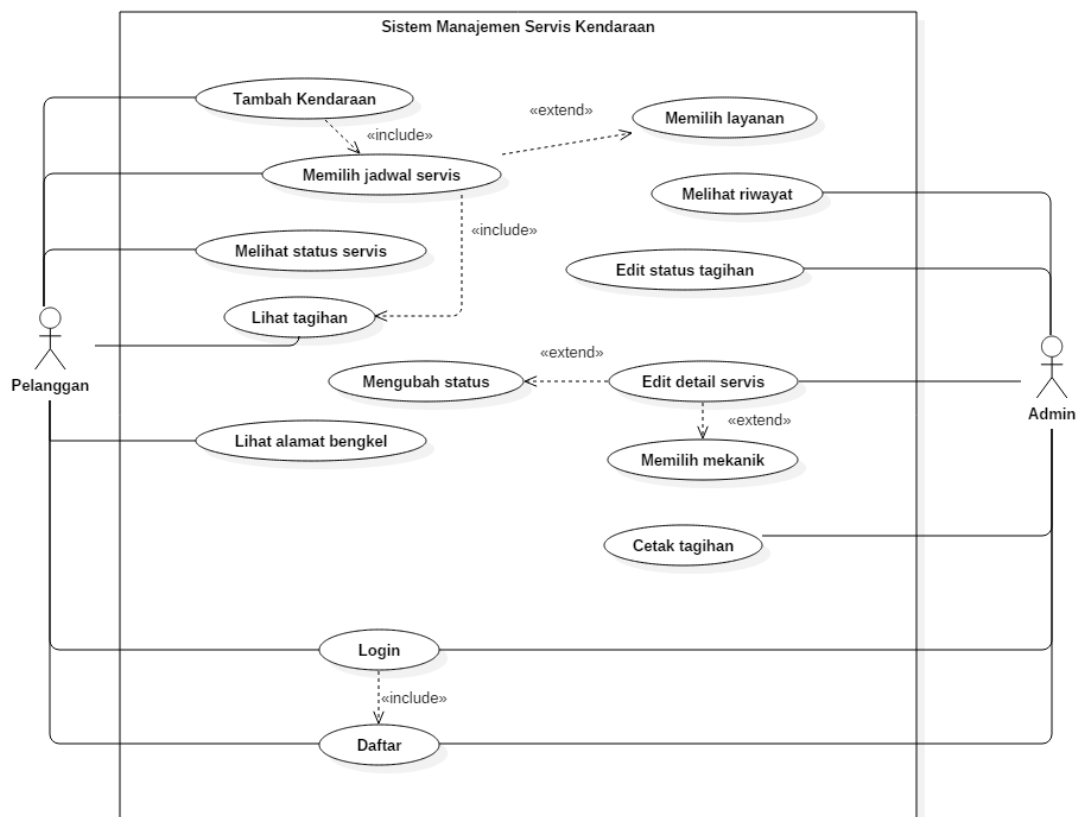
A. Latar Belakang	3
B. Detail Sistem	3
C. Library yang Digunakan	4
D. Database	5
E. Struktur File	7
F. Teknis Perancangan Aplikasi	9
G. Source Code	9
H. Output/Keluaran	32

A. Latar Belakang

Saya memilih tema ini karena menurut saya aplikasi ini akan sering digunakan terutama untuk bisnis bengkel baik resmi atau mandiri. Aplikasi ini dapat membantu sistem transaksi konsumen dan juga admin bengkel. Saya terinspirasi untuk memakai brand mobil Honda karena saya penggemar dari brand ini.

B. Detail Sistem

Saya telah membuat rancangan sistem ini menggunakan use case diagram, yang dimana diagram ini adalah standar dari UML (Unified Modelling Language) diagram.



Aplikasi ini terdiri dari 2 aktor, yaitu admin dan pelanggan. Untuk fitur pelanggan terdapat tambah data kendaraan, memilih jadwal servis, melihat status servis, melihat tagihan, dan melihat alamat bengkel. Dan fitur admin terdiri dari, melihat semua riwayat, mengubah status tagihan pelanggan, mengubah status servis pelanggan dan memilih mekanik, serta dapat mencetak tagihan dalam format pdf. Kedua aktor ini melewati tahap autentikasi (login) terlebih dahulu sebelum memasuki

tampilan utama aplikasi. Jika pelanggan belum memiliki akun, pelanggan dapat registrasi, dan mengakses sebagai akses pelanggan.

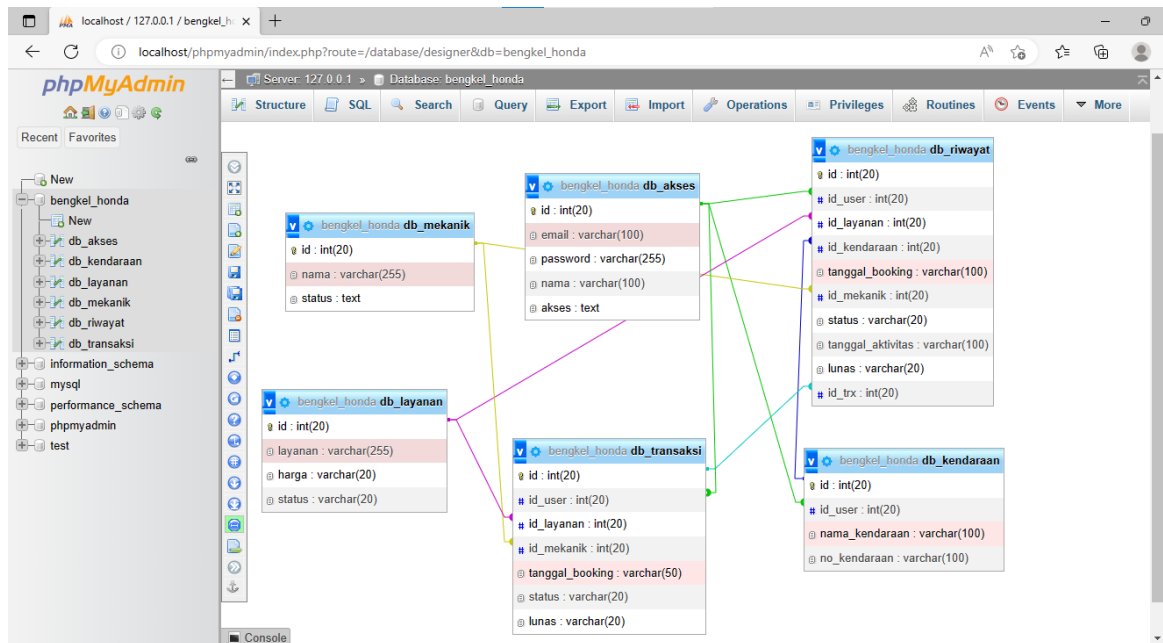
C. Library yang digunakan

- Os (built-in)
Library ini digunakan untuk merapikan screen.
Dokumentasi library ini dapat dilihat [disini](#).
- Time (built-in)
Library ini digunakan untuk memberikan delay pada screen.
Dokumentasi library ini dapat dilihat [disini](#).
- Datetime (built-in)
Library ini digunakan untuk mencetak waktu saat melakukan aktifitas pada riwayat transaksi dan membuat data tanggal untuk membuat jadwal servis.
Dokumentasi library ini dapat dilihat [disini](#).
- Json (built-in)
Library ini digunakan untuk membaca file json pada file configuration aplikasi, dengan tujuan memudahkan pengembang untuk memodifikasi aplikasi.
Dokumentasi library ini dapat dilihat [disini](#).
- Mysql.connector
Library ini digunakan untuk menghubungkan layanan mysql pada localhost dengan pemrograman python.
Dokumentasi library ini dapat dilihat [disini](#).
- Rich
Library ini digunakan untuk mengatur tampilan CLI (Command Line Interface) terlihat berwarna dan tidak membosankan.
Dokumentasi library ini dapat dilihat [disini](#).
- Questionary
Library ini digunakan untuk membuat menu interaktif, serta dapat membuat input jawaban user lebih mudah dipahami layaknya GUI (Graphical User Interface).
Dokumentasi library ini dapat dilihat [disini](#).
- Bcrypt
Library ini digunakan untuk mengenkripsi password dengan metode hashing. Sehingga data sensitif pengguna lebih aman disimpan di database.
Dokumentasi library ini dapat dilihat [disini](#).

- **FPDF**
Library ini digunakan untuk membuat struk tagihan dalam format pdf yang bisa dicetak oleh printer.
Dokumentasi library ini dapat dilihat [disini](#).

D. Database

Setelah saya mengamati sistem dan beberapa referensi, saya membuat beberapa tabel yang berhubungan dengan aplikasi ini. Saya juga sudah membuat relasi tabel data yang merupakan hal penting untuk membuat rancangan sistem.



Berikut detail penjelasan mengenai tabel-tabel:

- **db_akses**
Tabel ini bertujuan untuk menyimpan data akun pengguna, baik pelanggan maupun admin.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	int(20)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 email	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	3 password	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	4 nama	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	5 akses	text	utf8mb4_general_ci		No	None			Change Drop More

- db_layanan

Tabel ini bertujuan untuk menyimpan data layanan, harga, serta status yang disediakan oleh bengkel.

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id	int(20)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2	layanan	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	3	harga	varchar(20)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	4	status	varchar(20)	utf8mb4_general_ci		No	None			Change Drop More

- db_mekanik

Tabel ini bertujuan untuk menyimpan nama mekanik dan status yang bekerja di bengkel.

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id	int(20)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2	nama	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	3	status	text	utf8mb4_general_ci		No	None			Change Drop More

- db_kendaraan

Tabel ini bertujuan untuk menyimpan data model mobil, dan nomor kendaraan dari pelanggan. Dengan syarat, 1 pelanggan hanya dapat menyimpan 1 data mobil saja.

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id	int(20)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2	id_user	int(20)			No	None			Change Drop More
<input type="checkbox"/>	3	nama_kendaraan	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	4	no_kendaraan	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More

- db_transaksi

Tabel ini bertujuan untuk menyimpan data transaksi pengguna.

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id	int(20)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2	id_user	int(20)			No	None			Change Drop More
<input type="checkbox"/>	3	id_layanan	int(20)			No	None			Change Drop More
<input type="checkbox"/>	4	id_mekanik	int(20)			No	None			Change Drop More
<input type="checkbox"/>	5	tanggal_booking	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	6	status	varchar(20)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	7	lunas	varchar(20)	utf8mb4_general_ci		No	None			Change Drop More

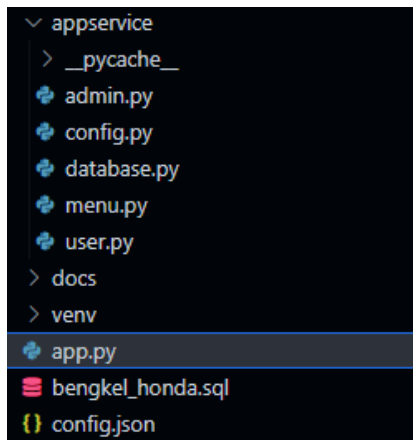
- db_riwayat

Tabel ini bertujuan untuk menyimpan pertinggal dari data transaksi dan disediakan kolom yang lebih lengkap dibanding tabel db_transaksi.

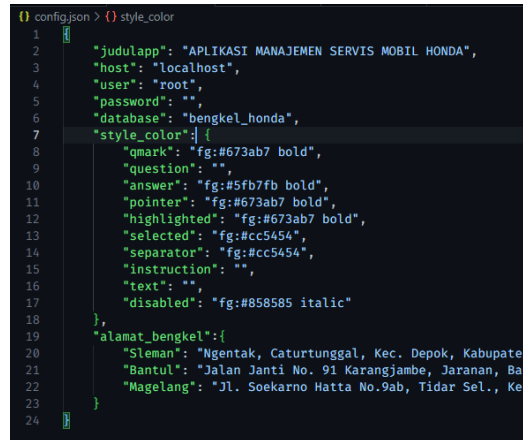
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id	int(20)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2	id_user	int(20)			No	None			Change Drop More
<input type="checkbox"/>	3	id_layanan	int(20)			No	None			Change Drop More
<input type="checkbox"/>	4	id_kendaraan	int(20)			No	None			Change Drop More
<input type="checkbox"/>	5	tanggal_booking	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	6	id_mekanik	int(20)			No	None			Change Drop More
<input type="checkbox"/>	7	status	varchar(20)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	8	tanggal_aktivitas	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	9	lunas	varchar(20)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	10	id_trx	int(20)			No	None			Change Drop More

E. Struktur File

Struktur file proyek yang rapih dapat membuat proses pengembangan menjadi lebih mudah, berikut hasil struktur yang saya buat.



Struktur file



File config.json

- config.json
File ini berfungsi sebagai file konfigurasi aplikasi ini, jadi pemilik komputer bisa mengatur judul aplikasi, konfigurasi localhost, dan tema warna tampilan.
- app.py
File ini merupakan file utama proyek.
- bengkel_honda.sql
File ini berfungsi sebagai database cadangan, ini dibutuhkan ketika menjalankan aplikasi pada komputer yang berbeda.
- appservice\config.py
File ini mengatur jalannya config.json ke fungsi-fungsi python dan menghubungkan ke sub program lain.
- appservice\database.py
File ini berisikan class session yang memiliki beberapa atribut mysql dan method yang menyambungkan ke layanan mysql database menggunakan paradigma OOP (Object Oriented Programming).
- appservice\menu.py
File ini berisikan fungsi-fungsi modular tampilan/front-end yang akan berhubungan dengan jalannya proses sistem/back-end.
- appservice\admin.py
File ini berisikan class dari actor admin. Memiliki atribut id, memiliki method yang berbeda dari pelanggan, dan berkaitan dengan class session.

- appservice\user.py
File ini berisikan class dari actor user/pelanggan. Memiliki atribut id, memiliki method yang berbeda dari admin, dan berkaitan dengan class session.

F. Teknis Perancangan Aplikasi

- Sebelum saya membuat aplikasi ini, saya membuat konsep dahulu dengan use case diagram, lalu saya membuat struktur database dan membuat relasi antar tabel data.
- Saya mencari beberapa referensi mengenai library built-in maupun library eksternal.
- Di proyek ini saya menggunakan VE (Virtual Environment) yang mana dapat membantu kelancaran aplikasi ini dan bisa digunakan di PC manapun.
- Saya merancang aplikasi ini dengan python versi 3.11.1 dan software pengolah teks Visual Studio Code.
- File source proyek dapat diunduh [disini](#).

G. Source Code

- **App.py (File Utama) [63 Baris]**

```
"""
+ -----+
| Nama Lengkap   : Muhammad Ali Pratama Putra |
| NIM            : 5220411416                 |
| Prodi          : Informatika                 |
+ -----+
"""

import time
from appservice import menu
from appservice import database
from appservice import user
from appservice import admin

session = database.Session()
session.connect()

def main_menu():
    if session.jenis_akses == 'admin':
```

```

        users = admin.Admin(session.nama_akun, session.id_akun)
    else:
        users = user.User(session.nama_akun, session.id_akun)
    menu.main_menu_akses(users.nama, session.jenis_akses)
    users.menu(session)
    del users

while True:
    menu.menu_01()
    pilihan_01 = menu.pilihan_01_01()
    if pilihan_01 == 0:
        menu.menu_01()
        login = menu.menu_02()
        data_akun =
session.get_data_where(str(login[0]).lower(), 'db_akses', 1)
        akses = session.auth(login[1], data_akun, 2)
        menu.login_load()
        menu.login_alert(akses)
        time.sleep(2)
        if akses:
            session.jenis_akses = data_akun[-1]
            session.nama_akun = data_akun[-2]
            session.id_akun = data_akun[0]
            menu.menu_akses(data_akun[-1])
            main_menu()
        elif pilihan_01 == 1:
            menu.menu_01()
            data = menu.menu_03()
            if bool(str(data[1]).find('@') == -1 and
str(data[1]).find('.') == -1) or data[2] != data[3]:
                menu.register_alert(False)
            else:
                menu.register_load()
                menu.register_alert(True) if
session.create_data('db_akses', [i for i in range(1, 5)],
[str(data[1]).lower(), data[2], data[0], 'user'], True, [1]) == 1
            else menu.register_alert(False)
                time.sleep(2)
                session.jenis_akses = 'user'
                session.nama_akun = data[0]
                data_akun = session.get_data_where(data[1], 'db_akses', 1)
                session.id_akun = data_akun[0]
                menu.menu_akses('user')
                main_menu()
        elif pilihan_01 == 2:

```

```

    menu.menu_quit()
    menu.clear_with_delay(1)
    session.quit()
    quit()

```

- **Menu.py [271 Baris]**

```

"""
+ -----+
| Nama Lengkap   : Muhammad Ali Pratama Putra |
| NIM            : 5220411416                 |
| Prodi          : Informatika                 |
+ -----+
"""

import os
import datetime
import time
import questionnaire
from questionnaire import Style
from rich.console import Console
from rich.progress import track
from rich.panel import Panel
from rich.table import Table
try:
    import config
except:
    from appservice import config

console = Console(width = 90)
tema = Style(config.getStyleColor())

def clear_with_delay(seconds):
    seconds = int(seconds)
    time.sleep(seconds)
    os.system('cls')
def menu_01():
    os.system('cls')
    console.print('Selamat Datang', style='#af1d1e blink bold',
justify='center')
    console.print(config.get_string('judulapp'),
justify='center')
    print('')
def menu_02():

```

```

        email = questionnaire.text("Masukkan Email Anda",
style=tema).ask()
        passwd = questionnaire.password("Masukkan Password Anda",
style=tema).ask()
        print('')
        return [email, passwd]
def menu_03():
    nama = questionnaire.text("Masukkan Nama Lengkap Anda",
style=tema).ask()
    email = questionnaire.text("Masukkan Email", style=tema).ask()
    pass1 = questionnaire.password("Masukkan Password",
style=tema).ask()
    pass2 = questionnaire.password("Masukkan Konfirmasi Password",
style=tema).ask()
    print('')
    return [nama, email, pass1, pass2]

def main_menu_akses(nama: str, akses: str):
    os.system('cls')
    console.print(f'Selamat Datang {nama}', style='#af1d1e blink
bold', justify='center')
    console.print(f'Anda masuk sebagai {akses}',
justify='center')
    print('')
def main_menu_user():
    pilihan = ('🚗 Cek Kendaraan', '📅 Buat Jadwal Servis', '🔍
Lihat Status Servis', '📄 Lihat Tagihan', '📍 Lihat Alamat Bengkel
Kami', '✕ Keluar')
    choise = pilihan.index(questionnaire.select(
        'Ada yang bisa kami bantu? 🙋',
        choices=pilihan,
        style=tema
    ).ask())
    return choise
def menu_user_buat_kendaraan():
    os.system('cls')
    console.print('Sepertinya anda belum menambahkan data
kendaraan', justify='center')
    print('')
    tipe = questionnaire.text("Masukkan Model Kendaraan Anda",
style=tema).ask()
    plat = questionnaire.text("Masukkan Nomor Kendaraan Anda",
style=tema).ask()
    return [tipe, plat]
def menu_user_00_create():

```

```

    os.system('cls')
    tipe = questionnaire.text("Masukkan Model Kendaraan Anda",
style=tema).ask()
    plat = questionnaire.text("Masukkan Nomor Kendaraan Anda",
style=tema).ask()
    return [tipe, plat]
def menu_user_00_check(model: str, plat: str):
    os.system('cls')
    pilihan = ('✎ Ubah', '← Kembali')
    console.print(f'🚗 Model Kendaraan\t: {model}', style='white
bold')
    console.print(f'📋 Nomor Kendaraan\t: {plat}', style='white
bold')
    print('')
    return pilihan.index(questionnaire.select(
        'Apakah anda ingin mengubah data?',
        choices=pilihan,
        style=tema
    ).ask())
def menu_user_01_sudah():
    os.system('cls')
    console.print('Anda sudah booking', style='#673ab7 blink
bold')
    print('')
    questionnaire.select('',choices=['← Kembali'],style=tema).ask()
def menu_user_01_pilih_layanan(layanan: list):
    os.system('cls')
    return [layanan.index(i) for i in questionnaire.checkbox(
        '🔗 Silahkan pilih layanan servis',
        choices=layanan,
        style=tema
    ).ask()]
def menu_user_01_pilih_hari():
    os.system('cls')
    pilih_hari = ('Hari Ini', 'Besok', 'Kustom')
    pilihan = pilih_hari.index(questionnaire.select(
        '📅 Silahkan pilih mode tanggal',
        choices=pilih_hari,
        style=tema
    ).ask())
    hari_ini = datetime.datetime.now()
    besok = hari_ini + datetime.timedelta(1)
    if pilihan == 0:
        return [hari_ini.day, hari_ini.month, hari_ini.year]
    elif pilihan == 1:

```

```

        return [besok.day, besok.month, besok.year]
    elif pilihan == 2:
        return str(questionary.text("Masukkan Tanggal dengan
format HH-BB-TTTT", style=tema).ask()).split('-')
def menu_user_02(layanan: list, tanggal_booking: list, status:
list):
    os.system('cls')
    [console.print(f"[white]{layanan[i]}{' ' if
len(layanan[i]) < 15 else
''}\t[#673ab7]{tanggal_booking[i]}\t[#af1d1e][b]{status[i]}/b]")
for i in range(len(layanan))]
    print('')
    questionary.select('',choices=['← Kembali'],style=tema).ask()
def menu_user_03(layanan: list, harga: list):
    os.system('cls')
    console.print('Rincian Tagihan Anda', style='#af1d1e blink
bold')
    print('')
    [console.print(f"[white]{layanan[i]}{' ' if
len(layanan[i]) < 15 else ''}\t[#673ab7]Rp{harga[i]}") for i in
range(len(layanan))]
    print('')
    console.print(f'Total Pembayaran: Rp{sum([int(i) for i in
harga])}', style='#af1d1e blink bold')
    print('')
    questionary.select('',choices=['← Kembali'],style=tema).ask()
def menu_user_04():
    os.system('cls')
    console.print('Alamat Bengkel Resmi Honda', style='#673ab7
blink bold')
    data = dict(config.get_string('alamat_bengkel')).items()
    [console.print(f"[#af1d1e][b]{i[0]}/b]\n[white]{i[1]}") for
i in data]
    print('')
    questionary.select('',choices=['← Kembali'],style=tema).ask()
def main_menu_admin():
    pilihan = ('✎ Edit Detail Servis','🖨 Edit Status Tagihan',
'🖨 Cetak Tagihan','📖 Lihat Riwayat Transaksi','✕ Keluar')
    chose = pilihan.index(questionary.select(
        'Silahkan pilih fitur admin 🖨',
        choices=pilihan,
        style=tema
    ).ask())
    return chose

```

```

def menu_admin_00_pilih_trx(layanan: list, status: list,
id_transaksi: list):
    pilihan = ['← Kembali']
    for i in range(len(layanan)):
        pilihan.append(f'{layanan[i]} | {status[i]}')
    choise = pilihan.index(questionary.select(
        'Silahkan pilih daftar booking 📅',
        choices=pilihan,
        style=tema
    ).ask())
    if choise > 0:
        return [id_transaksi[choise - 1], status[choise - 1]]
def menu_admin_00_pilih_mekanik(mekanik: list, id_mekanik: list):
    pilihan = ['← Kembali']
    for i in range(len(mekanik)):
        pilihan.append(mekanik[i])
    choise = pilihan.index(questionary.select(
        'Silahkan pilih mekanik 🧑🔧',
        choices=pilihan,
        style=tema
    ).ask())
    if choise > 0:
        return id_mekanik[choise - 1]
def menu_admin_00_pilih_status():
    pilihan = ('← Kembali', '🕒 Proses', '✓ Selesai')
    choise = questionary.select(
        'Silahkan pilih status 📅',
        choices=pilihan,
        style=tema
    ).ask()
    if choise == pilihan[1]:
        return 'proses'
    elif choise == pilihan[2]:
        return 'selesai'
def menu_admin_01(user: list):
    os.system('cls')
    return user.index(questionary.select(
        'Silahkan pilih user 🧑',
        choices=user,
        style=tema
    ).ask())
def menu_admin_01_ubah(layanan: list, harga: list, lunas: list,
id_transaksi: list):
    pilihan = ['← Kembali']
    for i in range(len(layanan)):

```

```

        pilihan.append(f'{layanan[i]} | {harga[i]} | {lunas[i]}')
    choise = pilihan.index(questionary.select(
        'Silahkan ubah status pelunasan 📄',
        choices=pilihan,
        style=tema
    ).ask())
    if choise > 0:
        return [id_transaksi[choise - 1], lunas[choise - 1]]
def menu_admin_02(user: list):
    os.system('cls')
    return user.index(questionary.select(
        'Silahkan pilih user 👤',
        choices=user,
        style=tema
    ).ask())
def menu_admin_04(data: list):
    os.system('cls')
    table = Table(title="Riwayat Seluruh Transaksi")
    table.add_column("Email", style="cyan", no_wrap=True)
    table.add_column("Nama", style="magenta", no_wrap=True)
    table.add_column("Layanan", style="green", no_wrap=True)
    table.add_column("Mobil", style="cyan", no_wrap=True)
    table.add_column("Tanggal Booking", style="magenta",
no_wrap=True)
    table.add_column("Mekanik", style="green", no_wrap=True)
    table.add_column("Status", style="cyan", no_wrap=True)
    table.add_column("Lunas", style="magenta", no_wrap=True)
    for i in data:
        i = str(i).split(' | ')
        table.add_row(i[0], i[1], i[2], i[3], i[4], i[5], i[6],
i[7])
    console.print(table, justify="center")
    print('')
    questionary.select('', choices=['⬅️ Kembali'], style=tema).ask()
def pilihan_01_01():
    pilihan = ('🚗 Masuk', '📄 Registrasi', '✖️ Keluar')
    choise = pilihan.index(questionary.select(
        'Ada yang bisa kami bantu? 🙋',
        choices=pilihan,
        style=tema
    ).ask())
    return choise
def error_alert(error_msg):
    os.system('cls')

```



```

error = Table.grid()
error.add_row(
    Panel.fit(
        f"[white]{error_msg}",
        title="Error!",
        border_style="#af1d1e",
        padding=(2, 2)
    ))
console.print(error)
def bool_alert(bool: bool):
    if bool:
        console.print('Berhasil', style='#673ab7 blink bold')
    else:
        console.print('Gagal', style='#af1d1e blink bold')
def login_alert(bool: bool):
    if bool:
        console.print('Berhasil', style='#673ab7 blink bold')
    else:
        console.print('Gagal', style='#af1d1e blink bold')
def register_alert(bool: bool):
    if bool:
        console.print('Berhasil Membuat Akun', style='#673ab7
blink bold')
    else:
        console.print('Gagal Membuat Akun', style='#af1d1e blink
bold')
def login_load():
    for i in track(range(20), description="Autentikasi Akun..."):
        time.sleep(0.075)
def register_load():
    for i in track(range(20), description="Membuat akun
anda..."):
        time.sleep(0.1)
def savedata_load():
    for i in track(range(20), description="Menyimpan data..."):
        time.sleep(0.05)
def pdf_alert():
    console.print('PDF sudah disimpan, silahkan print...',
style='#673ab7 blink bold')
    print('')
    questionnaire.select('',choices=['← Kembali'],style=tema).ask()
def menu_quit():
    os.system('cls')
    console.print('Terima Kasih', style='#af1d1e blink bold',
justify='center')

```

```
def menu_akses(akses):
    os.system('cls')
    console.print(f'Mengarahkan anda ke menu {akses}',
style='white blink bold', justify='center')
    time.sleep(2)
```

- **Config.py [19 Baris]**

```
"""
+ -----+
| Nama Lengkap : Muhammad Ali Pratama Putra |
| NIM          : 5220411416                 |
| Prodi        : Informatika                |
+ -----+
"""

import json
import os

dir = os.getcwd().replace('\\', '/')
with open(f'{dir}/config.json') as config_file:
    config = json.load(config_file)

def get_string(key):
    return config.get(key)
def getStyleColor():
    return list(config['style_color'].items())
```

- **Database.py [210 Baris]**

```
"""
+ -----+
| Nama Lengkap : Muhammad Ali Pratama Putra |
| NIM          : 5220411416                 |
| Prodi        : Informatika                |
+ -----+
"""

import mysql.connector as mysql
import bcrypt
try:
    import config
except:
    from appservice import config
try:
```

```

import menu
except:
from appservice import menu

class Session():
global mysql
def __init__(self):
self.host = config.config['host']
self.user = config.config['user']
self.passwd = config.config['password']
self.db = config.config['database']
self.jenis_akses = 'user'
self.nama_akun = ''
self.id_akun = 0
def connect(self):
try:
self.conn = mysql.connect(
host=self.host,
user=self.user,
passwd=self.passwd,
database=self.db
)
except:
menu.error_alert('Tidak Terkoneksi Database')
quit()
else:
self.cursor = self.conn.cursor(buffered=True)
def quit(self):
self.conn.disconnect()
def get_column(self, tabel: str):
try:
self.cursor.execute(f"""
SHOW COLUMNS FROM {tabel}
""")
except:
menu.error_alert('Ada kesalahan read database')
return ''
else:
columns = self.cursor.fetchall()
return tuple([column[0] for column in columns])
def get_data(self, tabel: str, index_kolom: list = [], isSingle:
bool = False):
tabel, isSingle, index_kolom = str(tabel), bool(isSingle),
tuple(index_kolom)
kolom_data = [self.get_column(tabel)[i] for i in index_kolom]

```

```

if len(index_kolom) < 1:
    sql = f"""
    SELECT * FROM {tabel}
    """
else:
    sql = f"""
    SELECT {str(kolom_data)[1:-1].replace("'", '')} FROM {tabel}
    """
try:
    self.cursor.execute(sql)
except:
    menu.error_alert('Ada kesalahan read database')
    return ''
else:
    if isSingle == True:
        return self.cursor.fetchone()
    else:
        return self.cursor.fetchall()
def create_data(self, tabel: str, index_kolom: list = [], values:
list = [], isEncrypt: bool = False, index_enkripsi: list = []):
    kolom = [self.get_column(tabel)[i] for i in index_kolom]
    values = [str(i) for i in values]
    if isEncrypt == True:
        for i in index_enkripsi:
            values[i] = str(bcrypt.hashpw(values[i].encode('utf-8'),
            bcrypt.gensalt()))[2:-1]
    sql = f"""INSERT INTO {tabel} ({str(kolom)[1:-1].replace("'",
    '')}) VALUES ({str(values)[1:-1]})"""
    try:
        self.cursor.execute(sql)
    except:
        menu.error_alert('Ada kesalahan write database')
        return 0
    else:
        self.conn.commit()
        return 1
def delete_data(self, tabel: str, index_kolom: list = [], values:
list = [], logical: str = 'AND'):
    kolom = [self.get_column(tabel)[i] for i in index_kolom]
    values = [str(i) for i in values]
    if len(index_kolom) > 1:
        sql = f"""DELETE FROM {tabel} WHERE
        {str([str(f'{kolom[i]}="{values[i]}"') for i, v in
        enumerate(index_kolom)])[1:-1].replace(',', ' '
        {logical}').replace("'", '')}"""

```

```

else:
sql = f"""DELETE FROM {tabel} WHERE
{kolom[index_kolom[0]]}='{values[0]}'"""
try:
self.cursor.execute(sql)
except:
menu.error_alert('Ada kesalahan delete database')
return 0
else:
self.conn.commit()
return 1
def edit_data(self, tabel: str, index_kolom: list = [], values:
list = [], index_anchor: int = 0, value_anchor: str = '',
isEncrypt: bool = False, index_enkripsi: list = []):
kolom = [self.get_column(tabel)[i] for i in index_kolom]
values = [str(i) for i in values]
if isEncrypt == True:
for i in index_enkripsi:
values[i] = str(bcrypt.hashpw(values[i].encode('utf-8'),
bcrypt.gensalt()))[2:-1]
set_statement = [f'{kolom[i]}="{values[i]}"' for i in
range(len(values))]
sql = f"""UPDATE {tabel} SET {str(set_statement)[1:-
1].replace("'", '')} WHERE
{self.get_column(tabel)[index_anchor]}="{value_anchor}""""
try:
self.cursor.execute(sql)
except:
menu.error_alert('Ada kesalahan write database')
return 0
else:
self.conn.commit()
return 1
def get_data_where(self, input_string: str, tabel: str,
index_kolom: int):
kolom = self.get_column(tabel)[index_kolom]
sql = f"""SELECT * FROM {tabel} WHERE
{kolom}="{input_string}""""
try:
self.cursor.execute(sql)
except:
menu.error_alert('Ada kesalahan read database')
return ''
else:
return self.cursor.fetchone()

```

```

def get_data_wheres(self, tabel: str, index_anchor: list,
anchor_string: list, index_target: list, isSingle: bool = False):
kolom = [self.get_column(tabel)[i] for i in index_anchor]
where_statement = [f'{kolom[i]}="{anchor_string[i]}"' for i in
range(len(index_anchor))]
if len(index_target) > 0:
select_statement = [self.get_column(tabel)[i] for i in
index_target]
sql = f"""SELECT {str(select_statement)[1:-1].replace('"', '')}
FROM {tabel} WHERE {str(where_statement)[1:-1].replace(',', 'AND
').replace('"', '')}"""
else:
sql = f"""SELECT * FROM {tabel.replace('"', '')} WHERE
{str(where_statement)[1:-1].replace(',', 'AND ').replace('"',
'')}"""
try:
self.cursor.execute(sql)
except:
menu.error_alert('Ada kesalahan read database')
return ''
else:
if isSingle:
return self.cursor.fetchone()
else:
return self.cursor.fetchall()
def auth(self, input_string: str, data: str, index_target: int):
try:
input_string = bytes(input_string, 'utf-8')
hash_value = bytes(data[index_target], 'utf-8')
except:
return False
else:
return bcrypt.checkpw(input_string, hash_value)
def get_data_sort(self, tabel: str, index_kolom: list = [],
index_anchor: int = 0, asc: bool = True, isSingle: bool = False):
tabel, isSingle, index_kolom = str(tabel), bool(isSingle),
tuple(index_kolom)
kolom_data = self.get_column(tabel)
if len(index_kolom) < 1:
sql = f"""
SELECT * FROM {tabel} ORDER BY {kolom_data[index_anchor]} {'DESC'
if asc == False else 'ASC'}
"""
else:
sql = f"""

```

```

SELECT {str(kolom_data)[1:-1].replace("'", '')} FROM {tabel}
ORDER BY {kolom_data[index_anchor]} {'DESC' if asc == False else
''}
"""
try:
self.cursor.execute(sql)
except:
menu.error_alert('Ada kesalahan read database')
return ''
else:
if isSingle == True:
return self.cursor.fetchone()
else:
return self.cursor.fetchall()
def create_dataa(self, tabel: str, list_kolom: list = [], values:
list = [], isEncrypt: bool = False, index_enkripsi: list = []):
kolom = list_kolom
values = [str(i) for i in values]
if isEncrypt == True:
for i in index_enkripsi:
values[i] = str(bcrypt.hashpw(values[i].encode('utf-8'),
bcrypt.gensalt()))[2:-1]
sql = f"""INSERT INTO {tabel} ({str(kolom)[1:-1].replace("'",
'')}) VALUES ({str(values)[1:-1]})"""
try:
self.cursor.execute(sql)
except:
menu.error_alert('Ada kesalahan write database')
return 0
else:
self.conn.commit()
return 1
def get_data_sortt(self, tabel: str, list_kolom: list = [],
index_anchor: int = 0, asc: bool = True, isSingle: bool = False):
tabel, isSingle = str(tabel), bool(isSingle)
kolom_data = list_kolom
sql = f"""
SELECT * FROM {tabel} ORDER BY {kolom_data[index_anchor]} {'DESC'
if asc == False else 'ASC'}
"""
try:
self.cursor.execute(sql)
except:
menu.error_alert('Ada kesalahan read database')
return ''

```

```

else:
if isSingle == True:
return self.cursor.fetchone()
else:
return self.cursor.fetchall()

```

- **Admin.py [156 Baris]**

```

• """
+ -----+
| Nama Lengkap      : Muhammad Ali Pratama Putra |
| NIM                : 5220411416                 |
| Prodi              : Informatika                 |
+ -----+
"""

import time
import datetime
from fpdf import FPDF
try:
    import menu
except:
    from appservice import menu
class Admin():
    def __init__(self, nama, id_akun):
        self.nama = nama
        self.id_akun = id_akun
    def menu(self, session):
        while True:
            menu.main_menu_akses(self.nama, 'admin')
            pilihan = menu.main_menu_admin()
            if pilihan == 4:
                session.nama_akun = ''
                session.id_akun = 0
                break
            elif pilihan == 3:
                akses = session.get_data('db_akses', [0,1,3])
                d_akses = [i[0] for i in akses]
                layanan = session.get_data('db_layanan', [0,1])
                d_layanan = [i[0] for i in layanan]
                kendaraan = session.get_data('db_kendaraan',
[0,2])

                d_kendaraan = [i[0] for i in kendaraan]
                mekanik = session.get_data('db_mekanik', [0,1])
                d_mekanik = [i[0] for i in mekanik]
                tabel = []

```



```

        for i in session.get_data('db_riwayat', []):
            tabel.append(f'{akses[d_akses.index(i[1])][1]} | {akses[d_akses.index(i[1])][2]} | {layanan[d_layanan.index(i[2])][1]} | {kendaraan[d_kendaraan.index(i[3])][1]} | {i[4]} | {mekanik[d_mekanik.index(i[5])][1]} | {i[6]} | {i[8]}}')
            menu.menu_admin_04(tabel)
        elif pilihan == 2:
            m_data = session.get_data_where('db_transaksi', [6], ['belum'], [1])
            m_id_akun = list(set([i[0] for i in m_data]))
            m_data_akun = session.get_data('db_akses', [])
            m_d_akun = [i[0] for i in m_data_akun]
            m_n_akun = [i[1] for i in m_data_akun]
            m_nn_akun = [i[3] for i in m_data_akun]
            m_nama_akun = [f'{m_n_akun[m_d_akun.index(m_id_akun[i])]} | {m_nn_akun[m_d_akun.index(m_id_akun[i])]} ' for i in range(len(m_id_akun))]
            id_akun_pilih = m_d_akun[menu.menu_admin_02(m_nama_akun)]
            # print(f'{id_akun_pilih}')
            # import os
            # os.system('pause')
            data_h = session.get_data('db_layanan', [])
            id_layanan = [i[0] for i in data_h]
            d_layanan = [i[1] for i in data_h]
            d_harga = [i[2] for i in data_h]
            layanan = []
            harga = []
            data = session.get_data_where('db_transaksi', [1, 6], [id_akun_pilih, 'belum'], [])
            for i in data:
                layanan.append(d_layanan[id_layanan.index(i[2])])
                harga.append(d_harga[id_layanan.index(i[2])])
            tghn = FPDF()
            tghn.add_page()
            tghn.set_font('Arial', 'B', 16)
            tghn.ln(20)
            tghn.cell(40, 10, 'Tagihan Servis Mobil')
            tghn.ln(7)
            tghn.set_font('Arial', 'I', 13)
            tghn.cell(40, 10, 'Servis & Dealer Resmi Honda Yogyakarta')

```

```

        tghn.ln(20)
        for i in range(len(layanan)):
            tghn.set_font('Arial', '', 11)
            tghn.cell(40, 10, f"{layanan[i]}{' ' if
len(layanan[i]) < 15 else ''}\tRp{harga[i]}")
            tghn.ln(7)
        tghn.ln(15)
        tghn.set_font('Arial', 'B', 11)
        tghn.cell(40, 10, f"Total Pembayaran:
Rp{sum([int(i) for i in harga])}")
        tghn.output(f'tagihan_{id_akun_pilih}.pdf', 'F')
        menu.savedata_load()
        menu.bool_alert(True)
        menu.pdf_alert()
    elif pilihan == 1:
        m_data = session.get_data('db_transaksi', [1])
        m_id_akun = list(set([i[0] for i in m_data]))
        m_data_akun = session.get_data('db_akses', [])
        m_d_akun = [i[0] for i in m_data_akun]
        m_n_akun = [i[1] for i in m_data_akun]
        m_nn_akun = [i[3] for i in m_data_akun]
        m_nama_akun =
[f'{m_n_akun[m_d_akun.index(m_id_akun[i])]} |
{m_nn_akun[m_d_akun.index(m_id_akun[i])]} ' for i in
range(len(m_id_akun))]
        id_akun_pilih =
m_d_akun[menu.menu_admin_01(m_nama_akun)]
        data_h = session.get_data('db_layanan', [])
        id_layanan = [i[0] for i in data_h]
        d_layanan = [i[1] for i in data_h]
        d_harga = [i[2] for i in data_h]
        layanan = []
        harga = []
        lunas = []
        id_transaksi = []
        data = session.get_data_where('db_transaksi',
[1], [id_akun_pilih], [])
        for i in data:
            layanan.append(d_layanan[id_layanan.index(i[2
]))))
            harga.append(d_harga[id_layanan.index(i[2]))
            lunas.append(i[6])
            id_transaksi.append(i[0])
        pilih_trx = menu.menu_admin_01_ubah(layanan,
harga, lunas, id_transaksi)

```

```

        if pilih_trx != None:
            data_riwayat =
session.get_data_where('db_riwayat', [9], [pilih_trx[0]],
[0])[0][0]
            if pilih_trx[1] == 'belum':
                session.edit_data('db_transaksi', [6],
['sudah'], 0, pilih_trx[0])
                session.edit_data('db_riwayat', [7, 8],
[datetime.datetime.now(), 'sudah'], 0, data_riwayat)
            else:
                session.edit_data('db_transaksi', [6],
['belum'], 0, pilih_trx[0])
                session.edit_data('db_riwayat', [7, 8],
[datetime.datetime.now(), 'belum'], 0, data_riwayat)
                menu.savedata_load()
                menu.bool_alert(True)
                time.sleep(2)
        elif pilihan == 0:
            m_data = session.get_data('db_transaksi', [1])
            m_id_akun = list(set([i[0] for i in m_data]))
            m_data_akun = session.get_data('db_akses', [])
            m_d_akun = [i[0] for i in m_data_akun]
            m_n_akun = [i[1] for i in m_data_akun]
            m_nn_akun = [i[3] for i in m_data_akun]
            m_nama_akun =
[f'{m_n_akun[m_d_akun.index(m_id_akun[i])]} |
{m_nn_akun[m_d_akun.index(m_id_akun[i])}]' for i in
range(len(m_id_akun))]
            id_akun_pilih =
m_d_akun[menu.menu_admin_01(m_nama_akun)]
            data_h = session.get_data('db_layanan', [])
            id_layanan = [i[0] for i in data_h]
            d_layanan = [i[1] for i in data_h]
            layanan = []
            status = []
            id_transaksi = []
            data = session.get_data_where('db_transaksi',
[1], [id_akun_pilih], [])
            for i in data:
                layanan.append(d_layanan[id_layanan.index(i[2
]))))
                status.append(i[5])
                id_transaksi.append(i[0])
            pilih_trx = menu.menu_admin_00_pilih_trx(layanan,
status, id_transaksi)

```

```

        if pilih_trx != None:
            data_riwayat =
session.get_data_wheres('db_riwayat', [9], [pilih_trx[0]],
[0])[0][0]

            if pilih_trx[1] == 'baru':
                data_mekanik =
session.get_data_wheres('db_mekanik', [2], ['ready'], [])
                id_mekanik = [i[0] for i in data_mekanik]
                nama_mekanik = [i[1] for i in
data_mekanik]

                id_mekanik.remove(1)
                nama_mekanik.remove('null')
                pilih_mekanik =
menu.menu_admin_00_pilih_mekanik(nama_mekanik, id_mekanik)
                session.edit_data('db_transaksi', [3, 5],
[pilih_mekanik, 'proses'], 0, pilih_trx[0])
                session.edit_data('db_riwayat', [5, 6,
7], [pilih_mekanik, 'proses', datetime.datetime.now()], 0,
data_riwayat)

            else:
                pilih_status =
menu.menu_admin_00_pilih_status()
                session.edit_data('db_transaksi', [5],
[pilih_status], 0, pilih_trx[0])
                session.edit_data('db_riwayat', [6, 7],
[pilih_status, datetime.datetime.now()], 0, data_riwayat)
                menu.savedata_load()
                menu.bool_alert(True)
                time.sleep(2)

```

- **User.py [120 Baris]**

```

"""
+ -----+
| Nama Lengkap      : Muhammad Ali Pratama Putra |
| NIM                : 5220411416                 |
| Prodi              : Informatika                 |
+ -----+
"""

import time
import datetime
import mysql.connector as mysql
try:
    import menu

```

```

except:
    from appservice import menu
class User():
    def __init__(self, nama, id_akun):
        self.nama = nama
        self.id_akun = id_akun
        self.status_cek_kendaraan = False
        self.id_kendaraan = 0
        self.model_kendaraan = ''
        self.nomor_kendaraan = ''
        self.status_booking = False
        self.id_booking = []
    def cek_kendaraan(self, session):
        data = session.get_data_where(self.id_akun,
'db_kendaraan', 1)
        if data == None:
            pengisian = menu.menu_user_buat_kendaraan()
            menu.savedata_load()
            session.create_data('db_kendaraan', [i for i in
range(1, 4)], [self.id_akun, pengisian[0], pengisian[1]])
            menu.bool_alert(True)
            time.sleep(2)
        else:
            self.status_cek_kendaraan = True
            self.id_kendaraan = int(data[0])
            self.model_kendaraan = data[2]
            self.nomor_kendaraan = data[3]
    def cek_booking(self, session):
        if self.status_booking == False:
            data_sudah_booking =
session.get_data_wheres('db_transaksi', [1], [self.id_akun], [0])
            if len(data_sudah_booking) > 0:
                self.status_booking = True
                for i in data_sudah_booking:
                    self.id_booking.append(i[0])
    def tambah_data_riwayat(self, session, id_layanan, tanggal,
id_trx):
        # session.create_data('db_riwayat', [i for i in range(1,
10)], [self.id_akun, id_layanan, self.id_kendaraan, tanggal, '1',
'baru', datetime.datetime.now(), 'belum', id_trx])
        session.cursor.execute("""
                                INSERT INTO db_riwayat (id_user,
id_layanan, id_kendaraan, tanggal_booking, id_mekanik, status,
tanggal_aktivitas, lunas, id_trx) VALUES (%s, %s, %s, %s, %s, %s,
%s, %s, %s)

```

```

        """', (self.id_akun, id_layanan,
self.id_kendaraan, tanggal, '1', 'baru', datetime.datetime.now(),
'belum', id_trx))
        session.conn.commit()
    def menu(self, session):
        while self.status_cek_kendaraan == False:
            self.cek_kendaraan(session)
        while True:
            menu.main_menu_akses(self.nama, 'user')
            pilihan = menu.main_menu_user()
            if pilihan == 5:
                session.nama_akun = ''
                session.id_akun = 0
                break
            elif pilihan == 4:
                menu.menu_user_04()
            elif pilihan == 0:
                if menu.menu_user_00_check(self.model_kendaraan,
self.nomor_kendaraan) == 0:
                    data = menu.menu_user_00_create()
                    session.edit_data('db_kendaraan', [2, 3],
[data[0], data[1]], 0, self.id_kendaraan)
                    self.model_kendaraan = data[0]
                    self.nomor_kendaraan = data[1]
                    menu.savedata_load()
                    menu.bool_alert(True)
                    time.sleep(2)
                elif pilihan == 1:
                    self.cek_booking(session)
                    if self.status_booking == True:
                        menu.menu_user_01_sudah()
                    else:
                        data = session.get_data('db_layanan', [])
                        layanan = [i[1] for i in data]
                        layanan.insert(0, 'Kembali')
                        index_layanan =
menu.menu_user_01_pilih_layanan(layanan)
                        if len(index_layanan) > 0:
                            tanggal =
str(menu.menu_user_01_pilih_hari())[1:-1].replace(', ', '-')
                            kolom_trx =
session.get_column('db_transaksi')
                            kolom_riw =
session.get_column('db_riwayat')
                            for il in index_layanan:

```

```

        il = int(il)
        session.create_dataa('db_transaksi',
kolom_trx[1:], [self.id_akun, data[il][0], '1', tanggal, 'baru',
'belum'])

        id_trx =
session.get_data_sortt('db_transaksi', kolom_trx, 0, False, True)
        session.create_dataa('db_riwayat',
kolom_riw[1:], [self.id_akun, data[il][0], self.id_kendaraan,
tanggal, '1', 'baru', datetime.datetime.now(), 'belum',
id_trx[0]])

        menu.savedata_load()
        menu.bool_alert(True)
        time.sleep(2)
    elif pilihan == 2:
        self.cek_booking(session)
        data_l = session.get_data('db_layanan', [])
        id_layanan = [i[0] for i in data_l]
        d_layanan = [i[1] for i in data_l]
        layanan = []
        tanggal_booking = []
        status = []
        for i in self.id_booking:
            data = session.get_data_where(i,
'db_transaksi', 0)
            layanan.append(d_layanan[id_layanan.index(dat
a[2])])

            tanggal_booking.append(data[4])
            status.append(data[5])
        menu.menu_user_02(layanan, tanggal_booking,
status)
    elif pilihan == 3:
        self.cek_booking(session)
        data_h = session.get_data('db_layanan', [])
        id_layanan = [i[0] for i in data_h]
        d_layanan = [i[1] for i in data_h]
        d_harga = [i[2] for i in data_h]
        layanan = []
        harga = []
        data = session.get_data_wheres('db_transaksi',
[1, 6], [self.id_akun, 'belum'], [])
        for i in data:
            layanan.append(d_layanan[id_layanan.index(i[2
])])

            harga.append(d_harga[id_layanan.index(i[2])])
        menu.menu_user_03(layanan, harga)

```


6) Data berhasil masuk pada database

-T→		id	email	password	nama	akses
<input type="checkbox"/>	Edit Copy Delete	13	user2@honda.id	\$2b\$12\$R5uJcxVc9KasV9yZ28qdU9eOCWZ0ImDUASJ0SK7UaU2...	user2	user
<input type="checkbox"/>	Edit Copy Delete	15	user1@honda.id	\$2b\$12\$Adile8ooHfjMwyepPSP7W0B4y8j0BU9a6.1LPsS2...	user1	user
<input type="checkbox"/>	Edit Copy Delete	16	admin@honda.id	\$2b\$12\$ZnjgUjJAC1GLbKtAXOzh0lInlPgzAbj7Gz5yEgPS...	admin	admin
<input type="checkbox"/>	Edit Copy Delete	20	user3@honda.id	\$2b\$12\$84dVptfYKjgJ2h8ucmXcjeFmaApotXbz15n7LilzYUuX...	user3	user
<input type="checkbox"/>	Edit Copy Delete	21	allepratama2@gmail.com	\$2b\$12\$UX5FcGw3kjrjms5034h.q3n1DYRULy5ILL1eBWgu...	Alie Pratama	user

7) Tampilan jika gagal membuat akun

```

Selamat Datang
APLIKASI MANAJEMEN SERVIS MOBIL HONDA

? Masukkan Nama Lengkap Anda Alie Pratama
? Masukkan Email aliepratama@honda.id
? Masukkan Password ***
? Masukkan Konfirmasi Password ****

Gagal Membuat Akun

```

b) Login

1) Menu utama

```

        Selamat Datang
        APLIKASI MANAJEMEN SERVIS MOBIL HONDA

        ? Ada yang bisa kami bantu? 🙋 (Gunakan tombol arah panah)
        » 🚗 Masuk
        📄 Registrasi
        ✖ Keluar
    
```

2) Masukkan email yang sudah terdaftar

```

Selamat Datang
APLIKASI MANAJEMEN SERVIS MOBIL HONDA

? Masukkan Email Anda aliepratama2@gmail.com

```

3) Masukkan password

```
Selamat Datang
APLIKASI MANAJEMEN SERVIS MOBIL HONDA

? Masukkan Email Anda aliepratama2@gmail.com
? Masukkan Password Anda ****
```

4) Proses autentikasi

```

          Selamat Datang
        APLIKASI MANAJEMEN SERVIS MOBIL HONDA

? Masukkan Email Anda aliepratama2@gmail.com
? Masukkan Password Anda ***

Autentikasi Akun... 35% 0:00:01

```

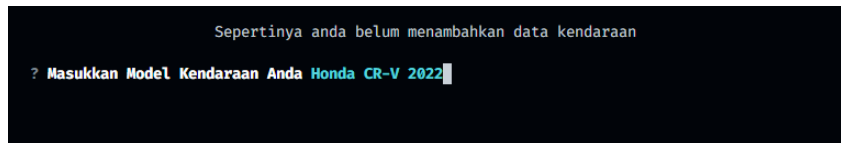
5) Tampilan jika pengguna gagal autentikasi



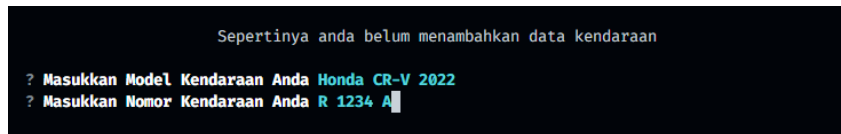
2. Akses Pelanggan

a) Tampilan menu akun baru

1) Meminta informasi model mobil



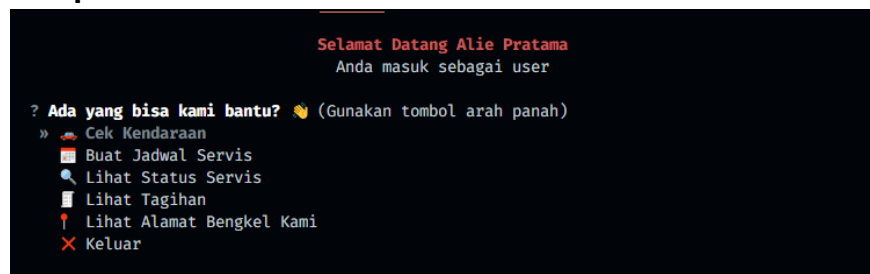
2) Meminta informasi nomor kendaraan



3) Data berhasil disimpan pada database

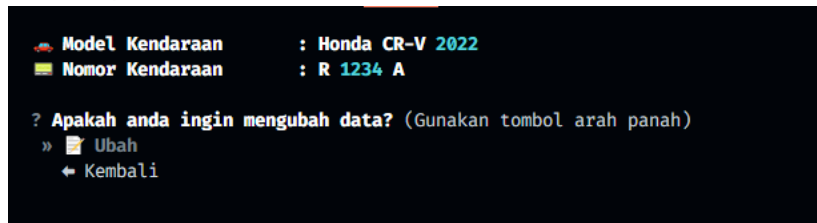


b) Tampilan menu utama

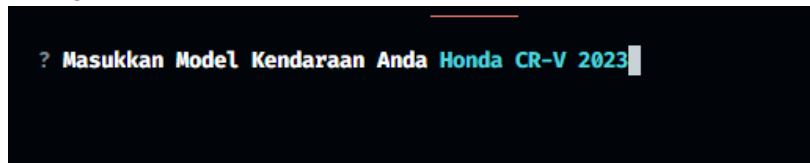


c) Fitur cek kendaraan

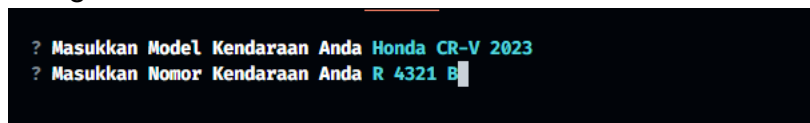
- 1) Menampilkan data kendaraan



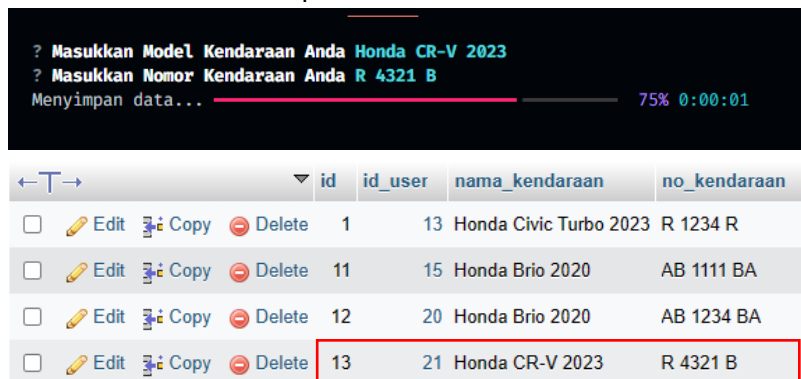
- 2) Mengubah informasi model mobil



- 3) Mengubah informasi nomor kendaraan

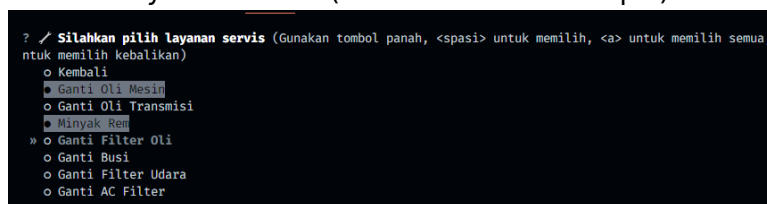


- 4) Data berhasil diubah pada database

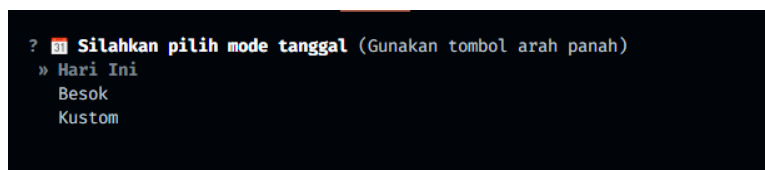


d) Fitur buat jadwal servis

- 1) Memilih layanan servis (bisa memilih dari 1 opsi)



- 2) Memilih waktu servis



3) Data berhasil disimpan pada database

```
? 📄 Silahkan pilih mode tanggal Hari Ini
Menyimpan data... 100% 0:00:00
Berhasil
```

		id	id_user	id_layanan	id_mekanik	tanggal_booking	status	lunas
<input type="checkbox"/>	Edit Copy Delete	13	13	4	2	17-1-2023	selesai	sudah
<input type="checkbox"/>	Edit Copy Delete	14	13	5	2	17-1-2023	selesai	sudah
<input type="checkbox"/>	Edit Copy Delete	15	13	6	2	17-1-2023	selesai	sudah
<input type="checkbox"/>	Edit Copy Delete	46	21	4	1	18-1-2023	baru	belum
<input type="checkbox"/>	Edit Copy Delete	47	21	6	1	18-1-2023	baru	belum

db_transaksi

		id	id_user	id_layanan	id_kendaraan	tanggal_booking	id_mekanik	status	tanggal_aktivitas	lunas	id_trx
<input type="checkbox"/>	Edit Copy Delete	5	13	4	1	17-1-2023	1	baru	2023-01-17 19:28:59.681915	sudah	13
<input type="checkbox"/>	Edit Copy Delete	6	13	5	1	17-1-2023	1	selesai	2023-01-17 19:29:06.081773	sudah	14
<input type="checkbox"/>	Edit Copy Delete	7	13	6	1	17-1-2023	2	selesai	2023-01-17 19:29:35.603039	sudah	15
<input type="checkbox"/>	Edit Copy Delete	15	21	4	13	18-1-2023	1	baru	2023-01-18 07:26:46.158325	belum	46
<input type="checkbox"/>	Edit Copy Delete	16	21	6	13	18-1-2023	1	baru	2023-01-18 07:26:46.167321	belum	47

db_riwayat

e) Fitur lihat status servis

```
Ganti Oli Transmisi      18-1-2023      baru
Ganti Filter Oli        18-1-2023      baru

? (Gunakan tombol arah panah)
» ⬅ Kembali
```

f) Fitur lihat tagihan

```
Rincian Tagihan Anda

Ganti Oli Transmisi      Rp125000
Ganti Filter Oli        Rp45000

Total Pembayaran: Rp170000

? (Gunakan tombol arah panah)
» ⬅ Kembali
```

g) Fitur lihat alamat bengkel

```
Alamat Bengkel Resmi Honda
Sleman
Ngentak, Caturtunggal, Kec. Depok, Kabupaten Sleman, Daerah Istimewa Yogyakarta 55281
Bantul
Jalan Janti No. 91 Karangjambe, Jaranan, Banguntapan, Bantul, Kabupaten Sleman, Daerah
Istimewa Yogyakarta 55198
Magelang
Jl. Soekarno Hatta No.9ab, Tidar Sel., Kec. Magelang Sel., Kota Magelang, Jawa Tengah
56125

? (Gunakan tombol arah panah)
» ◀ Kembali
```

h) Keluar/Logout

```
                Selamat Datang
            APLIKASI MANAJEMEN SERVIS MOBIL HONDA

? Ada yang bisa kami bantu? 🙋 (Gunakan tombol arah panah)
» 🚗 Masuk
  📄 Registrasi
  ✖ Keluar
```

3. Akses Admin

a) Tampilan menu utama

```
                Selamat Datang admin
            Anda masuk sebagai admin

? Silahkan pilih fitur admin 📄 (Gunakan tombol arah panah)
» 🛠 Edit Detail Servis
  📄 Edit Status Tagihan
  📄 Cetak Tagihan
  📄 Lihat Riwayat Transaksi
  ✖ Keluar
```

b) Fitur edit detail servis

1) Memilih data user/pelanggan

```
? Silahkan pilih user 👤 (Gunakan tombol arah panah)
» aliepratama2@gmail.com | Alie Pratama
  user2@honda.id | user2
```

2) Memilih data booking

```
? Silahkan pilih user 👤 aliepratama2@gmail.com | Alie Pratama
? Silahkan pilih daftar booking 📄 (Gunakan tombol arah panah)
  ◀ Kembali
» Ganti Oli Transmisi | baru
  Ganti Filter Oli | baru
```

- 3) Jika data booking masih status 'baru', maka admin dapat memilih mekaniknya terlebih dahulu. Status akan otomatis menjadi 'proses'

```
? Silahkan pilih user 🧑 aliepratama2@gmail.com | Alie Pratama
? Silahkan pilih daftar booking 🛠️ Ganti Oli Transmisi | baru
? Silahkan pilih mekanik 🧑 (Gunakan tombol arah panah)
  ← Kembali
    Pratama
  » Alie
```

```
? Silahkan pilih user 🧑 aliepratama2@gmail.com | Alie Pratama
? Silahkan pilih daftar booking 🛠️ Ganti Oli Transmisi | baru
? Silahkan pilih mekanik 🧑 Alie
Menyimpan data... 100% 0:00:00
Berhasil
```

	id	id_user	id_layanan	id_mekanik	tanggal_booking	status	lunas
<input type="checkbox"/> Edit Copy Delete	46	21	4	1	18-1-2023	baru	belum
<input type="checkbox"/> Edit Copy Delete	47	21	6	1	18-1-2023	baru	belum
<input type="checkbox"/> Edit Copy Delete	48	13	4	4	18-1-2023	proses	belum
<input type="checkbox"/> Edit Copy Delete	49	13	6	1	18-1-2023	baru	belum

db_transaksi

	id	id_user	id_layanan	id_kendaraan	tanggal_booking	id_mekanik	status	tanggal_aktivitas	lunas	id_trx
<input type="checkbox"/> Edit Copy Delete	15	21	4	13	18-1-2023	1	baru	2023-01-18 07:26:46.158325	belum	46
<input type="checkbox"/> Edit Copy Delete	16	21	6	13	18-1-2023	1	baru	2023-01-18 07:26:46.167321	belum	47
<input type="checkbox"/> Edit Copy Delete	17	13	4	1	18-1-2023	4	proses	2023-01-18 07:51:18.651101	belum	48
<input type="checkbox"/> Edit Copy Delete	18	13	6	1	18-1-2023	1	baru	2023-01-18 07:45:24.131979	belum	49

db_riwayat

- 4) Jika data booking status selain 'baru', maka admin dapat memilih status menjadi 'proses' atau 'selesai'

```
? Silahkan pilih user 🧑 aliepratama2@gmail.com | Alie Pratama
? Silahkan pilih daftar booking 🛠️ Ganti Oli Transmisi | proses
? Silahkan pilih status 📁 (Gunakan tombol arah panah)
  ← Kembali
    Proses
  » ✓ Selesai
```

- 5) Data berhasil diubah pada database

```
? Silahkan pilih user 🧑 aliepratama2@gmail.com | Alie Pratama
? Silahkan pilih daftar booking 🛠️ Ganti Oli Transmisi | proses
? Silahkan pilih status 📁 ✓ Selesai
Menyimpan data... 100% 0:00:00
Berhasil
```

		id	id_user	id_layanan	id_mekanik	tanggal_booking	status	lunas
<input type="checkbox"/>	Edit Copy Delete	46	21	4	1	18-1-2023	baru	belum
<input type="checkbox"/>	Edit Copy Delete	47	21	6	1	18-1-2023	baru	belum
<input type="checkbox"/>	Edit Copy Delete	48	13	4	4	18-1-2023	selesai	belum
<input type="checkbox"/>	Edit Copy Delete	49	13	6	1	18-1-2023	baru	belum

db_transaksi

		id	id_user	id_layanan	id_kendaraan	tanggal_booking	id_mekanik	status	tanggal_aktivitas	lunas	id_trx
<input type="checkbox"/>	Edit Copy Delete	15	21	4	13	18-1-2023	1	baru	2023-01-18 07:26:46.158325	belum	46
<input type="checkbox"/>	Edit Copy Delete	16	21	6	13	18-1-2023	1	baru	2023-01-18 07:26:46.167321	belum	47
<input type="checkbox"/>	Edit Copy Delete	17	13	4	1	18-1-2023	4	selesai	2023-01-18 07:53:47.072771	belum	48
<input type="checkbox"/>	Edit Copy Delete	18	13	6	1	18-1-2023	1	baru	2023-01-18 07:45:24.131979	belum	49

db_riwayat

c) Fitur edit status tagihan

a) Memilih data user/pelanggan

```
? Silahkan pilih user (Gunakan tombol arah panah)
» aliepratama2@gmail.com | Alie Pratama
  user2@honda.id | user2
```

b) Memilih data booking (Status lunas akan berubah menjadi 'sudah', jika status sebelumnya 'belum', berlaku sebaliknya)

```
? Silahkan pilih user aliepratama2@gmail.com | Alie Pratama
? Silahkan ubah status pelunasan (Gunakan tombol arah panah)
« Kembali
» Ganti Oli Transmisi | 125000 | belum
  Ganti Filter Oli | 45000 | belum
```

c) Data berhasil diubah pada database

```
? Silahkan pilih user aliepratama2@gmail.com | Alie Pratama
? Silahkan ubah status pelunasan Ganti Oli Transmisi | 125000 | belum
Menyimpan data... 100% 0:00:00
Berhasil
```

		id	id_user	id_layanan	id_mekanik	tanggal_booking	status	lunas
<input type="checkbox"/>	Edit Copy Delete	46	21	4	1	18-1-2023	baru	belum
<input type="checkbox"/>	Edit Copy Delete	47	21	6	1	18-1-2023	baru	belum
<input type="checkbox"/>	Edit Copy Delete	48	13	4	4	18-1-2023	selesai	sudah
<input type="checkbox"/>	Edit Copy Delete	49	13	6	1	18-1-2023	baru	belum

db_transaksi

		id	id_user	id_layanan	id_kendaraan	tanggal_booking	id_mekanik	status	tanggal_aktivitas	lunas	id_trx
<input type="checkbox"/>	Edit Copy Delete	15	21	4	13	18-1-2023	1	baru	2023-01-18 07:26:46.158325	belum	46
<input type="checkbox"/>	Edit Copy Delete	16	21	6	13	18-1-2023	1	baru	2023-01-18 07:26:46.167321	belum	47
<input type="checkbox"/>	Edit Copy Delete	17	13	4	1	18-1-2023	4	selesai	2023-01-18 07:56:37.022825	sudah	48
<input type="checkbox"/>	Edit Copy Delete	18	13	6	1	18-1-2023	1	baru	2023-01-18 07:45:24.131979	belum	49

db_riwayat

d) Fitur cetak tagihan

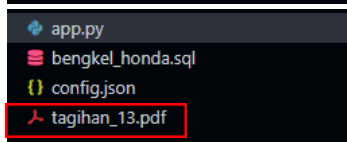
a) Memilih data user/pelanggan

```
? Silahkan pilih user 🧑 (Gunakan tombol arah panah)
» user2@honda.id | user2
  aliepratama2@gmail.com | Alie Pratama
```

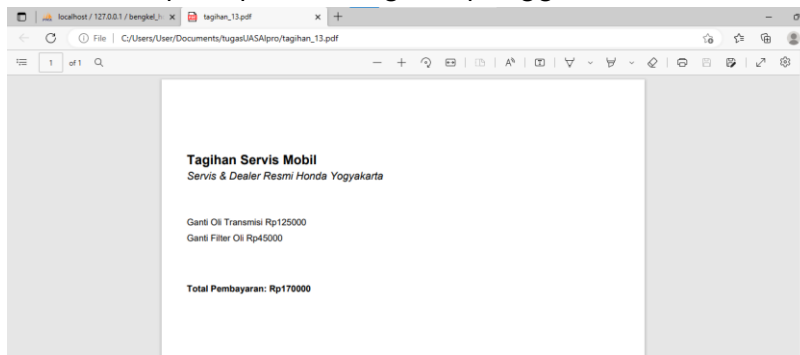
b) Hasil tagihan akan muncul di folder dengan format pdf

```
? Silahkan pilih user 🧑 user2@honda.id | user2
Menyimpan data... 100% 0:00:00
Berhasil
PDF sudah disimpan, silahkan print...

? (Gunakan tombol arah panah)
» ⬅️ Kembali
```



c) Berikut tampilan pdf dari tagihan pengguna



e) Fitur lihat riwayat transaksi

Riwayat Seluruh Transaksi

Email	Nama	Layanan	Mobil	Tanggal ...	M...	-	
user2@honda.id	user2	Ganti Oli Tr...	Honda Civic Turb...	17-1-2023	n...	...	
user2@honda.id	user2	Minyak Rem	Honda Civic Turb...	17-1-2023	n...	...	
user2@honda.id	user2	Ganti Filter...	Honda Civic Turb...	17-1-2023	P...	...	
aliepratama2@gmail...	Alie P...	Ganti Oli Tr...	Honda CR-V 2023	18-1-2023	n...	...	
aliepratama2@gmail...	Alie P...	Ganti Filter...	Honda CR-V 2023	18-1-2023	n...	...	

? (Gunakan tombol arah panah)
» ⬅️ Kembali

f) Keluar/Logout

```
Selamat Datang
APLIKASI MANAJEMEN SERVIS MOBIL HONDA

? Ada yang bisa kami bantu? 🧑 (Gunakan tombol arah panah)
» 🏠 Masuk
  📄 Registrasi
  ✖️ Keluar
```