

**LAPORAN PROYEK AKHIR PRAKTIKUM
MATA KULIAH ALGORITMA DAN PEMROGRAMAN DASAR**



[SISTEM MANAJEMEN SERVIS MOTOR]

Oleh:

Kelompok 4

MIFTAHUL FAUZAN	2409106048
MUHAMMAD NAUFAL ADI	2409106049
BRATA P.S.P	
FAHCLEVI MUHAMMAD	2409106059

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MULAWARMAN
SAMARINDA 2024**

KATA PENGANTAR

Puji syukur kami panjatkan kepada Allah SWT, yang telah memberikan rahmat, hidayah, dan kekuatan kepada kami sehingga laporan ini dapat diselesaikan dengan baik. Laporan ini disusun sebagai salah satu syarat untuk menyelesaikan proyek akhir pada mata kuliah Algoritma Pemrograman Dasar dan diharapkan dapat memberikan gambaran yang jelas mengenai pengembangan sistem manajemen servis motor yang telah kami buat.

Program yang kami buat adalah Sistem Manajemen Servis Motor, yang dirancang untuk memudahkan dalam mengelola data pelanggan, transaksi servis, serta inventaris suku cadang di bengkel motor. Sistem ini dibangun dengan menggunakan bahasa pemrograman python, dengan tujuan untuk meningkatkan efisiensi dan akurasi dalam pengelolaan data servis motor, baik untuk pelanggan maupun pihak bengkel.

Selama proses penyusunan laporan dan pembuatan program ini, kami menghadapi beberapa kendala, terutama dalam hal alur flowchart, error code, dan lain lain. Meskipun demikian, kami berhasil mengatasinya dengan bantuan dari berbagai pihak. Oleh karena itu, kami ingin mengucapkan terima kasih yang sebesar-besarnya kepada bang ifnu dan bang adi, selaku aslab algoritma pemrograman dasar yang telah mengajarkan kami selama ini. Kami juga mengucapkan terima kasih kepada bang Tito selaku aslab pembimbing proyek akhir ini yang turut membantu dalam proses ini, baik dalam hal teknis maupun ide-ide yang sangat membantu. Tanpa dukungan dan bantuan mereka, laporan ini tidak akan dapat diselesaikan dengan baik.

Akhir kata, kami berharap laporan ini dapat bermanfaat dan memberikan kontribusi positif dalam pengembangan sistem manajemen servis motor. Kami juga menyadari bahwa laporan ini masih jauh dari sempurna, oleh karena itu kami sangat mengharapkan kritik dan saran yang membangun demi perbaikan di masa yang akan datang.

Penulis

Samarinda, 19 November 2024

TAKARIR

Daftar padanan kata bahasa asing dalam bahasa Indonesia yang digunakan adalah sebagai berikut:

<i>Database</i>	Basis Data
<i>Managemen</i>	Mengatur
<i>Input</i>	Memasukkan
<i>Update</i>	Memperbarui
<i>Delete</i>	Menghapus
<i>Service</i>	Layanan
<i>Status</i>	Keadaan
<i>Process</i>	Proses
<i>Admin</i>	Pengelola
<i>System</i>	Sistem
<i>User</i>	Pengguna
<i>Interface</i>	Antarmuka
<i>Data</i>	Data
<i>Record</i>	Catatan
<i>error</i>	Kesalahan
<i>Display</i>	Tampilan
<i>Output</i>	Keluaran
<i>Report</i>	Laporan
<i>Menu</i>	Pilihan
<i>Sparepart</i>	Suku Cadang
<i>Save</i>	Menyimpan
<i>Header</i>	Judul
<i>Break</i>	Berhenti
<i>Clear</i>	Membersihkan

DAFTAR ISI

KATA PENGANTAR	2
TAKARIR	3
DAFTAR ISI.....	4
DAFTAR GAMBAR	5
BAB I	
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Kebutuhan Fungsional	1
1.3 Rumusan Masalah	2
1.4 Batasan Masalah.....	2
1.5 Tujuan	3
BAB II	
PERANCANGAN	4
2.1 Analisis Program.....	4
2.2 Flowchart	6
2.3 Konsep/Materi Praktikum yang dipakai	28
BAB III	
HASIL DAN PEMBAHASAN.....	33
3.1 Tampilan Program.....	33
1. Menu Utama.....	33
2. Register	33
3. Login	34
4. Exit	35
5. Tampilan Menu User.....	35
6. Menu Servis Motor	36
7. Update Keluhan.....	37
8. Histori Servis.....	38
9. Tampilan Menu Admin	38
10. Menu Update Status	38
11. Menu Delete Data Motor	39
12. Tampilan Menu Sparepart.....	40
13. Menu Tambah Sparepart	41
14. Menu Update Sparepart.....	42
15. Menu Delete Sparepart.....	43

3.2 Source Code	44
1. Main	44
2. Menu Admin	50
3. Menu User	59
4. Util.....	66
BAB IV	
PENUTUP	71
4.1 Kesimpulan	71
4.2 Saran.....	71
DAFTAR PUSTAKA	73
LAMPIRAN	74

DAFTAR GAMBAR

Gambar 2.2.1 Alur Awal Menu.....	5
Gambar 2.2.2 Alur Menu Pilihan 1	6
Gambar 2.2.3 Alur Memverifikasi	7
Gambar 2.2.4 Alur Jika Username / Password Salah.....	8
Gambar 2.2.5 Alur Menu Pilihan 2.....	9
Gambar 2.2.6 Alur Memasukkan Data Ke Database	10
Gambar 2.2.7 Alur Menu Pilihan 3.....	10
Gambar 2.2.8 Alur Menuju End.....	10
Gambar 2.2.9 Alur Menu User.....	11
Gambar 2.2.10 Alur Isi Menu User.....	12
Gambar 2.2.11 Alur Pilihan 1 Menu User	13
Gambar 2.2.12 Alur Memasukkan Data User Ke Database.....	14
Gambar 2.2.13 Alur Pilihan 2 Menu User	14
Gambar 2.2.14 Alur Jika User Ingin Ada Keluhan Baru	15
Gambar 2.2.15 Alur Pilihan 3 Menu User	15
Gambar 2.2.16 Alur Pilihan 4 Menu User	16
Gambar 2.2.17 Alur Kembali Ke Menu Utama	16
Gambar 2.2.18 Alur Menu Admin	17
Gambar 2.2.19 Alur Isi Menu Admin	18
Gambar 2.2.20 Alur Pilihan 1 Menu Admin.....	18
Gambar 2.2.21 Alur Mengubah Status Servis.....	19
Gambar 2.2.22 Alur Pilihan 2 Menu Admin.....	19
Gambar 2.2.23 Alur Pilihan 3 Menu Admin.....	20
Gambar 2.2.24 Alur Pilihan 4 Menu Admin.....	20
Gambar 2.2.25 Alur Pilihan Menu Sparepart.....	22
Gambar 2.2.26 Alur Pilihan 1 Menu Sparepart.....	23
Gambar 2.2.27 Alur Pilihan 2 Menu Sparepart.....	23
Gambar 2.2.28 Alur Pilihan 3 Menu Sparepart.....	24
Gambar 2.2.29 Alur Pilihan 4 menu Sparepart	25
Gambar 3.1.1 Tampilan Menu Awal.....	30
Gambar 3.1.2 Tampilan Register Berhasil	30
Gambar 3.1.3 Tampilan Register Jika Username/Password Kurang Dari 5 Karakter.....	31

Gambar 3.1.4 Tampilan Register Jika Username Sudah Ada.....	31
Gambar 3.1.5 Tampilan Login Gagal Jika Memasukkan Username / Password Yang Salah.....	31
Gambar 3.1.6 Tampilan Jika Memilih Exit.....	32
Gambar 3.1.7 Tampilan Menu User.....	32
Gambar 3.1.8 Tampilan Menu Service Motor	33
Gambar 3.1.9 Tampilan Sesudah	33
Gambar 3.1.10 Tampilan Update Keluhan	34
Gambar 3.1.11 Tampilan Setelah di Update	34
Gambar 3.1.12 Tampilan Histori Servis	35
Gambar 3.1.13 Tampilan Menu Admin	35
Gambar 3.1.14 Tampilan Menu Update Status.....	35
Gambar 3.1.15 Tampilan Setelah Mengupdate Status	36
Gambar 3.1.16 Tampilan Menu Delete Data Motor	36
Gambar 3.1.17 Tampilan Setelah Delete Data Motor.....	37
Gambar 3.1.18 Tampilan Menu Sparepart.....	37
Gambar 3.1.19 Tampilan Menu Tambah Sparepart.....	38
Gambar 3.1.20 Tampilan Ketika Sudah Ditambahkan	38
Gambar 3.1.21 Tampilan Menu Update Sparepart	39
Gambar 3.1.22 Tampilan Setelah Mengupdate Sparepart	39
Gambar 3.1.23 Tampilan Menu Delete Sparepart	40
Gambar 3.1.24 Tampilan Setelah Di Delete	40
Gambar 3.2.1 Source Code Import	41
Gambar 3.2.2 Source Utama	42
Gambar 3.2.3 Source Code Menu Login	43
Gambar 3.2.4 Source Code Lanjutan Menu Login	44
Gambar 3.2.5 Source Code Menu Register.....	45
Gambar 3.2.6 Source Code Lanjutan Menu Register	46
Gambar 3.2.7 Source Code Import Menu Admin.....	47
Gambar 3.2.8 Source Code Menu Admin.....	48
Gambar 3.2.9 Source Code Update Status Services	49
Gambar 3.2.10 Source Code Delete Status Services.....	50
Gambar 3.2.11 Source Code Menu Sparepart Main	51
Gambar 3.2.12 Source Code Menu Sparepart.....	52
Gambar 3.2.13 Source Code Add Sparepart	53

Gambar 3.2.14 Source Code Update Sparepart	54
Gambar 3.2.15 Source Code Delete Sparepart	55
Gambar 3.2.16 Source Code Import Menu User	56
Gambar 3.2.17 Source Code Menu User	57
Gambar 3.2.18 Source Code Service Motor	58
Gambar 3.2.19 Source Code Update Keluhan	59
Gambar 3.2.20 Source Code Histori Service	60
Gambar 3.2.21 Source Code Import Util	61
Gambar 3.2.22 Source Code Clear.....	62
Gambar 3.2.23 Source Code Write Data & Read Data.....	63
Gambar 3.2.24 Source Code Print Header	63
Gambar 3.2.25 Source Code Print Data Motorcycles	64
Gambar 3.2.26 Source Code Print Motorcycles User.....	65
Kartu Konsul	69
Konsul Pertama	70
Konsul Kedua.....	70

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada era globalisasi seperti sekarang banyak terdapat teknologi yang semakin maju dan semakin berkembang. Teknologi otomotif menjadi salah satu yang mengalami peningkatan yang pesat. Salah satu alasan peningkatan teknologi ini adalah meningkatnya kebutuhan konsumen akan mode transportasi yang Aman, Nyaman, Efektif dan Efisien yang beragam. Hal inilah yang menjadikan semakin bermunculan variasi jenis dan tipe kendaraan yang memicu peningkatan jumlah unit kendaraan setiap tahunnya tentunya dengan kebutuhan dan keinginan masing-masing. Sangat erat kaitannya dengan fenomena tersebut, jasa service yang dibutuhkan pun harus mengikuti perkembangan teknologi dan tren kendaraan pada era sekarang ini. Peningkatan teknologi akan mempengaruhi aspek perawatan yang diperlukan (Dwifa, et al., 2021)

Motor menjadi salah satu alat transportasi utama yang digunakan oleh masyarakat karena efisiensinya dalam mobilitas harian. Namun, seiring pemakaian yang terus menerus, motor memerlukan perawatan berkala agar tetap berfungsi optimal. Permasalahan yang sering muncul adalah kurangnya pengelolaan data servis motor, baik dari pihak pengguna maupun bengkel. Catatan servis yang dilakukan secara manual cenderung sulit dilacak, tidak terorganisir, dan rawan hilang.

Kondisi ini mengakibatkan ketidakefisienan dalam pengelolaan riwayat servis, seperti jadwal servis berikutnya atau riwayat penggantian suku cadang. Oleh karena itu, diperlukan sebuah sistem yang mampu mengelola data servis motor secara terstruktur dan mudah diakses. Program ini diharapkan dapat menjadi solusi untuk mempermudah pihak bengkel maupun pengguna dalam mencatat, mengakses, dan mengelola informasi servis motor.

1.2 Kebutuhan Fungsional

Agar sistem ini dapat berfungsi dengan baik sesuai tujuan, berikut adalah kebutuhan fungsionalnya:

1. Pencatatan Servis Motor: Sistem harus dapat mencatat data pelanggan, data motor, serta jenis servis yang dilakukan.
2. Riwayat Servis: Memungkinkan pengguna untuk melihat catatan lengkap servis motor berdasarkan tanggal atau jenis servis.
3. Manajemen Data: Administrator dapat menambah, mengedit, atau menghapus data pelanggan dan motor.

1.3 Rumusan Masalah

Berdasarkan latar belakang di atas, masalah yang dapat dirumuskan adalah:

1. Bagaimana merancang sistem manajemen servis motor yang mampu mencatat riwayat servis secara efektif?
2. Bagaimana sistem dapat membantu mempermudah pengelolaan data bagi pihak bengkel?
3. Bagaimana sistem dapat memberikan pengingat jadwal servis kepada pengguna secara otomatis?

1.4 Batasan Masalah

Agar pembahasan lebih fokus dan terarah, berikut adalah batasan-batasan yang diterapkan dalam perancangan sistem ini:

1. Sistem hanya digunakan untuk manajemen servis motor dan tidak mencakup kendaraan lain.
2. Data yang dikelola terbatas pada data pelanggan, motor, jenis servis, dan jadwal servis.

3. Sistem ini dirancang untuk digunakan pada komputer dengan jaringan lokal, tidak berbasis cloud.

1.5 Tujuan

Tujuan dari pembuatan sistem manajemen servis motor ini adalah :

1. Mempermudah pihak bengkel dalam mencatat dan mengelola data servis motor secara terstruktur.
2. Meningkatkan kepuasan pengguna dengan menyediakan riwayat servis yang mudah diakses.
3. Mengurangi risiko kehilangan data dengan sistem pencatatan yang terpusat.

BAB II

PERANCANGAN

2.1 Analisis Program

Program “Sistem Manajemen Servis Motor” adalah program yang dibuat untuk mengelola sebuah bengkel motor. Program ini dirancang untuk memberikan solusi pengelolaan servis motor secara terstruktur dengan membedakan peran pengguna menjadi Admin dan User. Setiap peran pengguna memiliki fitur-fitur tersendiri yang hanya dapat diaplikasikan oleh pengguna tersebut.

Program dimulai dengan meminta pengguna untuk memasukkan username dan password. Sistem akan melakukan pengecekan untuk memastikan akses hanya diberikan kepada pengguna yang sudah melakukan registrasi atau pengguna yang sudah terdaftar. Jika data tidak sesuai, maka program akan menampilkan pesan kesalahan dan meminta pengguna mencoba kembali. Setelah berhasil login, sistem menentukan jenis pengguna berdasarkan data akun: apakah sebagai Admin atau User. Untuk pengguna yang belum melakukan registrasi/belum terdaftar, pengguna bisa melakukan registrasi terlebih dahulu dengan memasukkan beberapa data seperti nama, username, dan password.

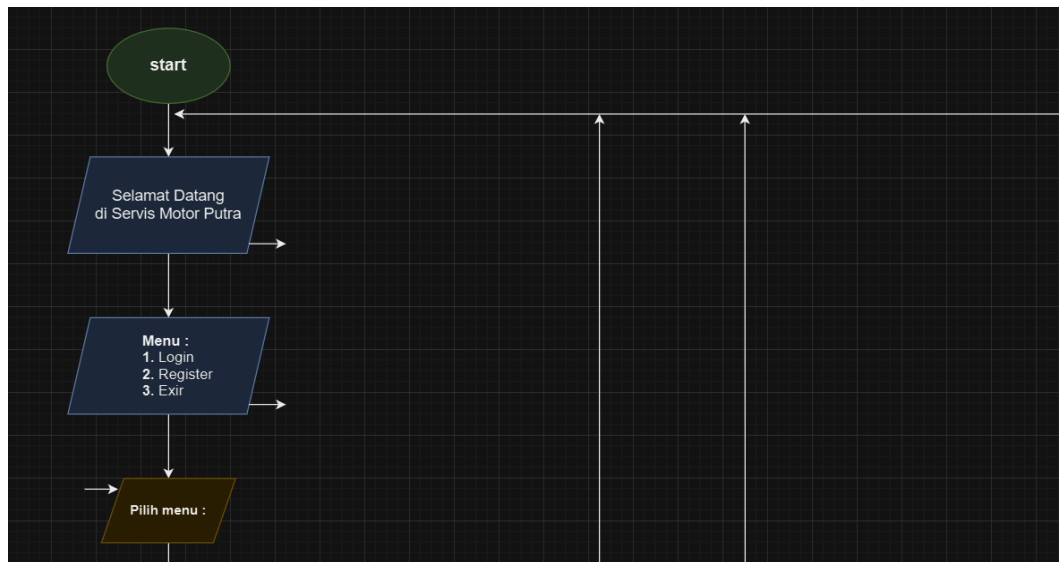
Setelah melakukan proses login pengguna dialihkan ke menu yang sesuai dengan perannya masing-masing. Di dalam menu Admin, Admin dapat melakukan update terhadap data servis motor. Seorang admin dapat mengupdate status servis motor. Tidak hanya itu, seorang Admin juga dapat melakukan delete data servis motor apabila data tersebut tidak dibutuhkan lagi atau dengan kata lain proses servis telah selesai. Seorang admin juga dapat melakukan beberapa manipulasi data sparepart motor, seperti menambah, mengupdate dan menghapus data sparepart motor yang dibutuhkan.

Di dalam menu User, User dapat melakukan servis motor. Apabila User ingin melakukan servis motor, ia dapat masuk ke menu servis motor. Sistem akan meminta beberapa data terkait motor yang akan diservis, seperti nama motor, kilometer motor dan keluhan/servis apa yang ingin dilakukan oleh User. Selain itu, user dapat melakukan update keluhan apabila terjadi kesalahan dalam memasukkan data keluhan. Di dalam menu User juga terdapat fitur histori servis, fitur ini dapat digunakan oleh User untuk melihat riwayat servis motor yang pernah dilakukan oleh user.

Terdapat fitur exit untuk setiap peran baik Admin maupun User yang ingin keluar dari program. Jika Admin atau User memilih fitur exit, maka mereka akan kembali ke menu utama.

2.2 Flowchart

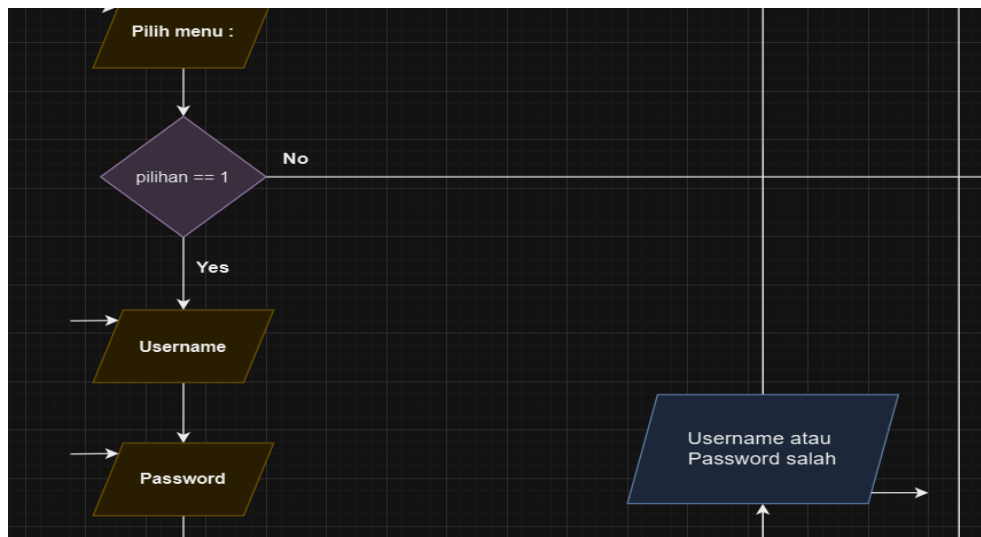
Main



Gambar 2.2.1 alur awal menu

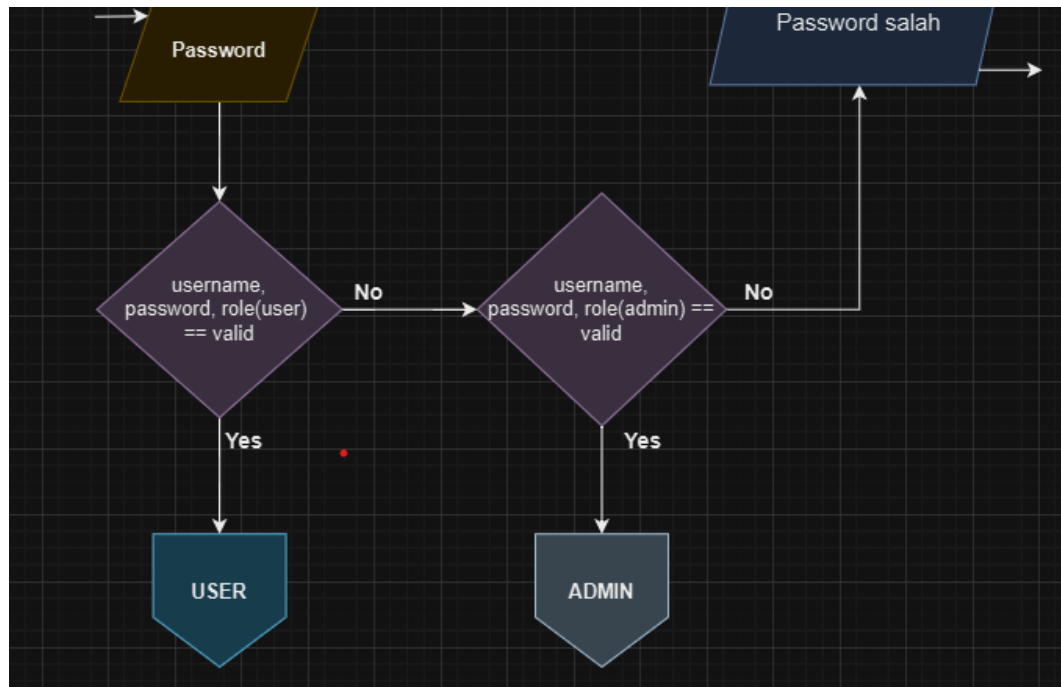
Program dimulai dengan menampilkan beberapa menu, diantaranya ada menu login, menu register, dan menu exit untuk keluar dari program. Menu login

adalah menu yang dapat dipilih oleh pengguna apabila pengguna tersebut telah melakukan registrasi sebelumnya sedangkan menu register dapat dipilih apabila pengguna sebelumnya belum pernah melakukan registrasi sama sekali. Program meminta pengguna untuk memilih satu opsi sesuai angka yang terdapat dalam menu.



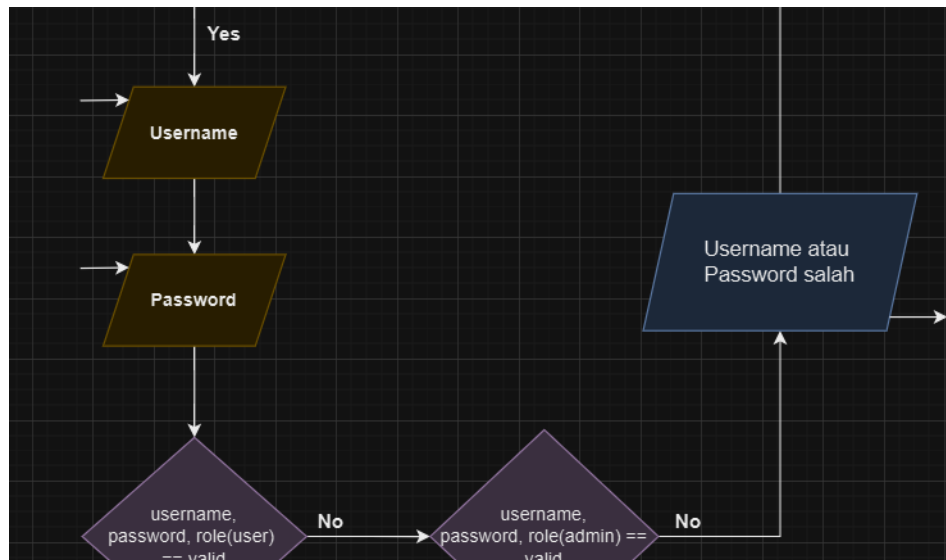
Gambar 2.2.2 Alur menu pilihan 1

Apabila pengguna memilih opsi “1”, maka akan masuk ke menu login. Program akan meminta pengguna untuk menginputkan username dan password.



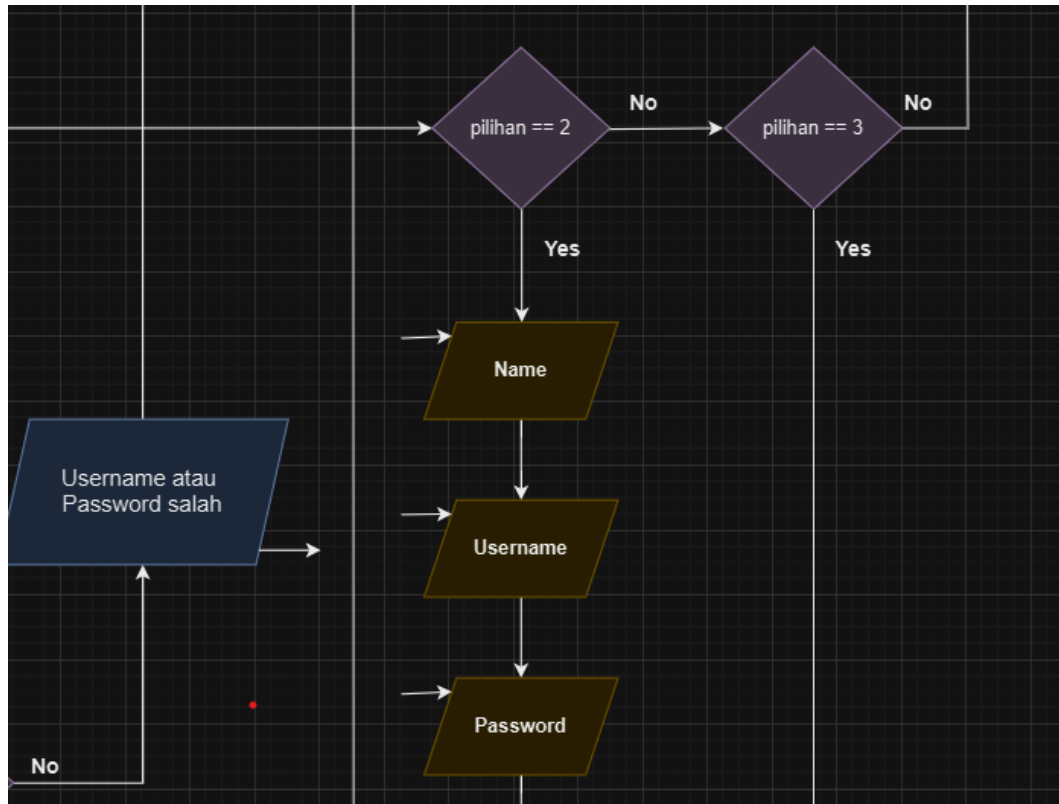
Gambar 2.2.3 Alur memverifikasi

Kemudian program akan memverifikasi apakah username dan password yang diinputkan oleh pengguna valid dan memverifikasi role dari pengguna tadi. Jika username dan password pengguna valid serta pengguna tersebut memiliki role “user”, maka pengguna tadi akan masuk ke menu user. Jika username dan password pengguna valid serta pengguna tersebut memiliki role “admin”, maka pengguna tadi akan masuk ke menu admin.



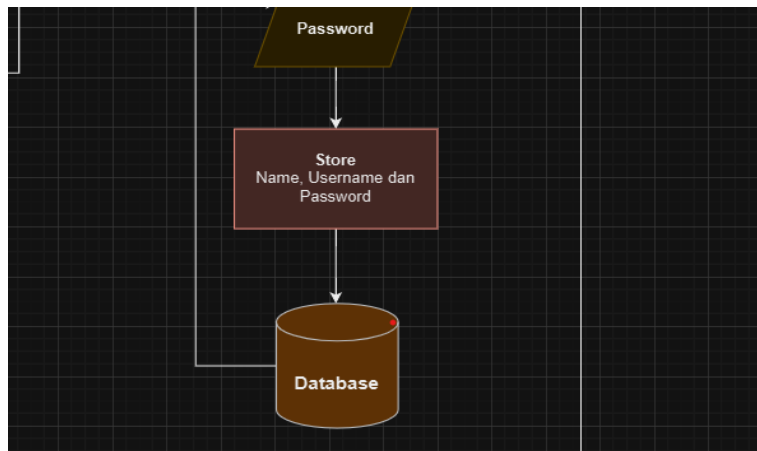
Gambar 2.2.4 Alur jika username / password salah

Jika username atau password salah, maka program akan menampilkan peringatan bahwa username atau password salah, setelah itu program akan balik menu awal.



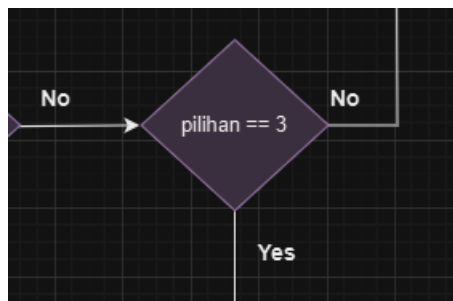
Gambar 2.2.5 Alur menu pilihan 2

Apabila pengguna memilih opsi “2”, maka pengguna akan masuk ke menu register. Program akan meminta pengguna untuk menginputkan nama, username dan password.

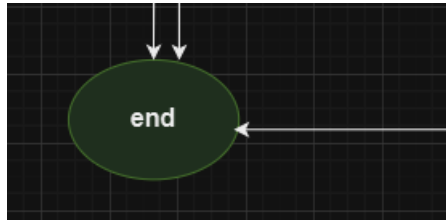


Gambar 2.2.6 Alur memasukkan data ke database

Kemudian program akan memproses data yang telah diinputkan tadi dan memasukkan data tadi ke sebuah database, setelah itu program akan kembali ke menu awal.



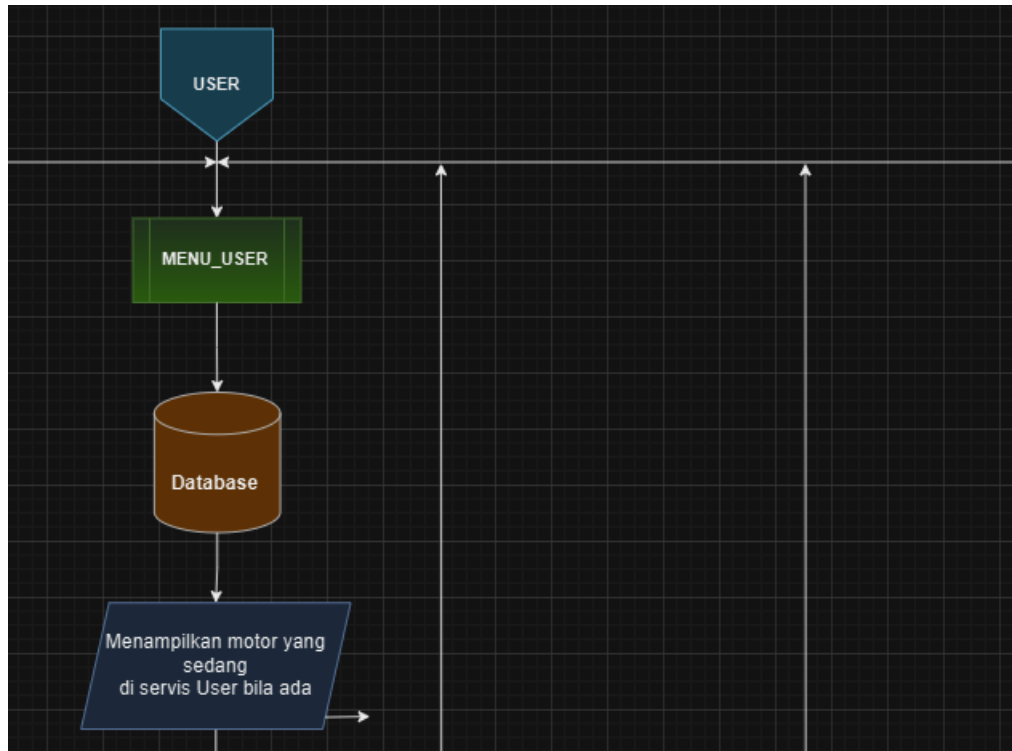
Gambar 2.2.7 Alur menu pilihan 3



Gambar 2.2.8 Alur menuju end

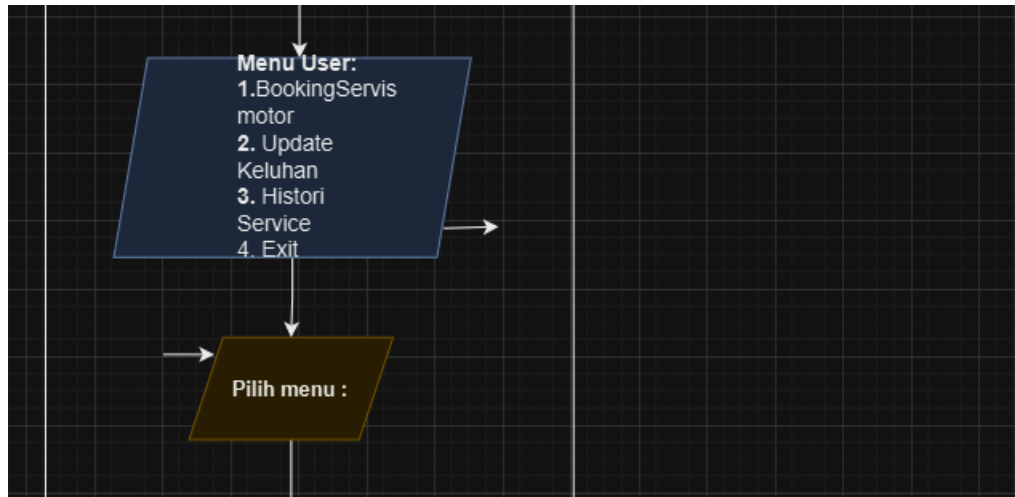
Apabila pengguna memilih opsi “3”, maka pengguna akan keluar dari program/program akan terhenti. Jika pengguna memasukkan opsi selain yang tersedia di menu, maka program akan menampilkan pesan kesalahan dan program diulang ke menu utama

Menu User



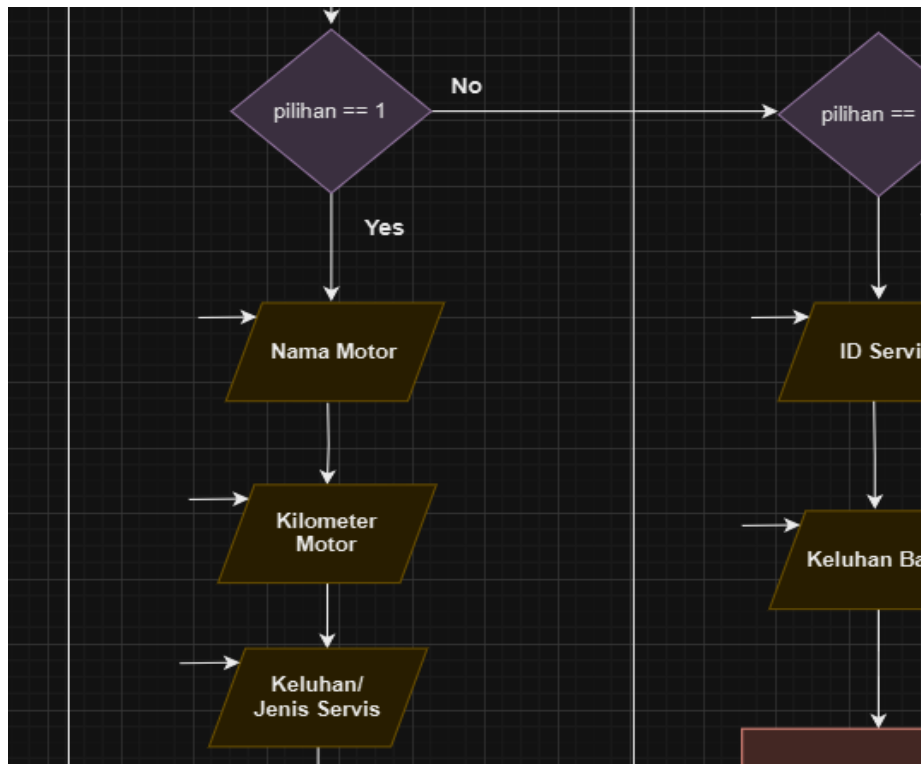
Gambar 2.2.9 Alur menu user

Pada menu user, program akan menampilkan data motor yang sedang diservice oleh pengguna jika sebelumnya pengguna melakukan servis motor.



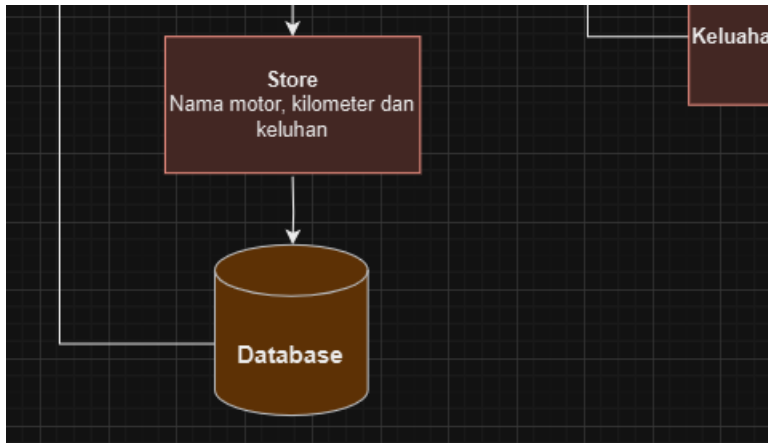
Gambar 2.2.10 Alur isi menu user

Kemudian program akan menampilkan beberapa menu, diantaranya ada menu booking servis motor, update keluhan, histori service dan menu exit. Menu booking servis motor dapat dipilih apabila pengguna akan melakukan service motor. Menu update keluhan dapat dipilih apabila pengguna ingin mengupdate keluhan service. Menu histori service dapat dipilih apabila pengguna ingin melihat hsitori/track record motor yang pernah diservis dan exit untuk keluar dari menu user. Program meminta pengguna untuk memilih satu opsi sesuai angka yang terdapat dalam menu.



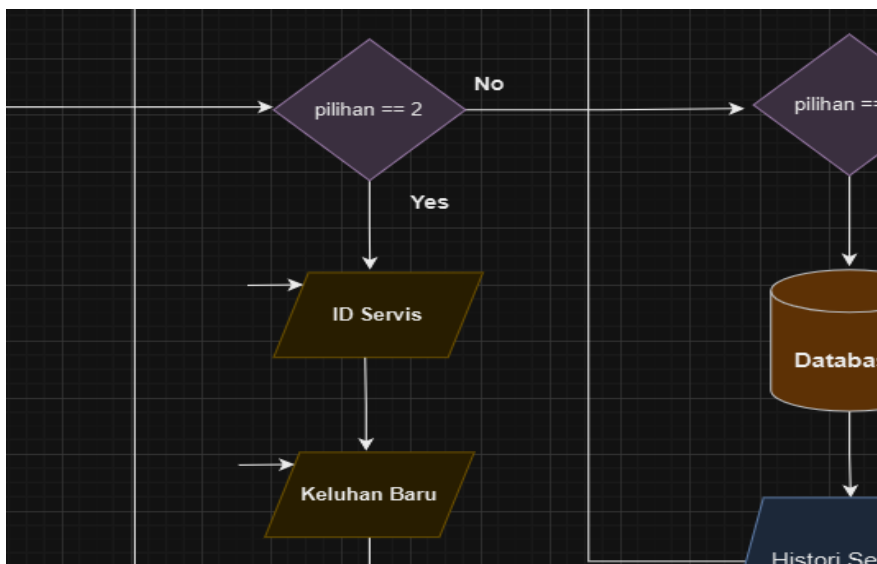
Gambar 2.2.11 Alur pilihan 1 menu user

Apabila pengguna memilih opsi “1”, maka pengguna akan masuk ke menu booking sservis. Program akan meminta pengguna untuk menginputkan nama motor, kilometer motor dan keluhan motor.



Gambar 2.2.12 Alur memasukkan data user ke database

Kemudian program akan memproses data yang telah diinputkan tadi dan memasukkan data tadi ke sebuah database, setelah itu program akan kembali ke menu awal user.



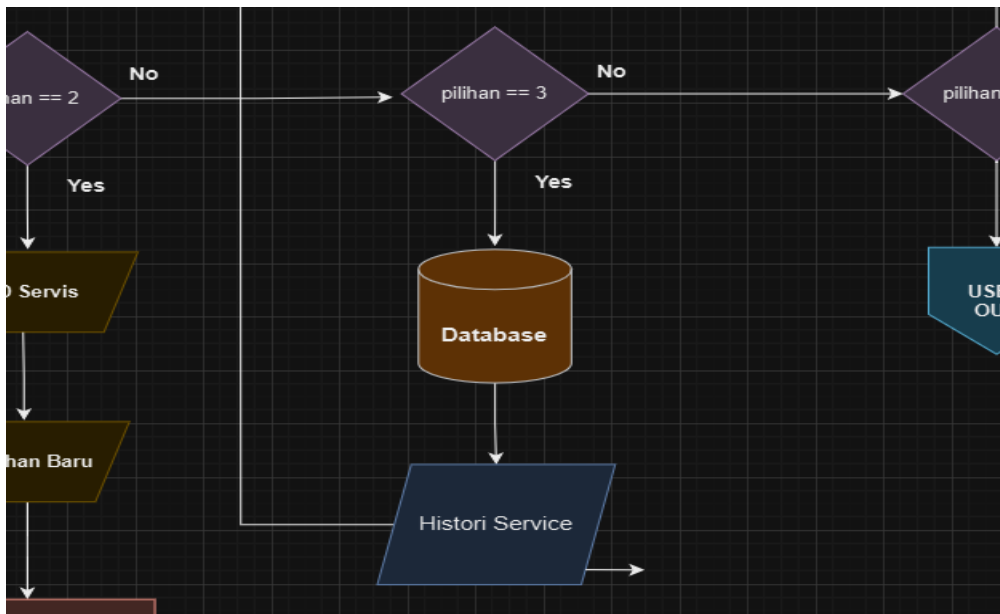
Gambar 2.2.13 Alur pilihan 2 menu user

Apabila pengguna memilih opsi “2”, maka pengguna akan masuk ke menu update keluhan. Program akan meminta pengguna untuk menginputkan ID servis dan keluhan baru.



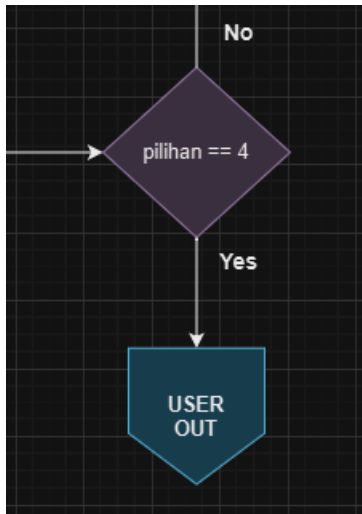
Gambar 2.2.14 Alur jika user ingin ada keluhan baru

Program akan mengubah keluhan yang lama dengan keluhan yang baru, setelah itu program akan kembali ke menu awal user.

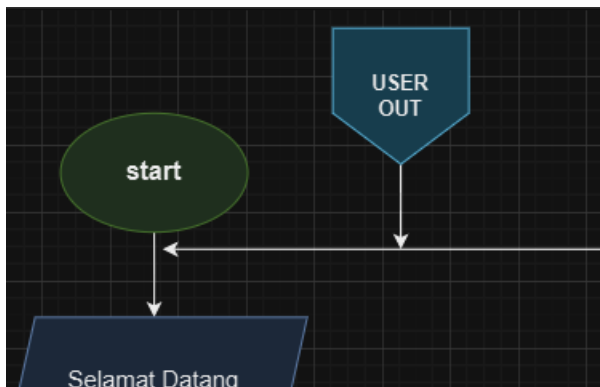


Gambar 2.2.15 Alur pilihan 3 menu user

Apabila pengguna memilih opsi “3”, maka pengguna akan masuk ke menu histori service. Program akan menampilkan histori servis motor pengguna. Setelah itu program akan kembali ke menu utama user.



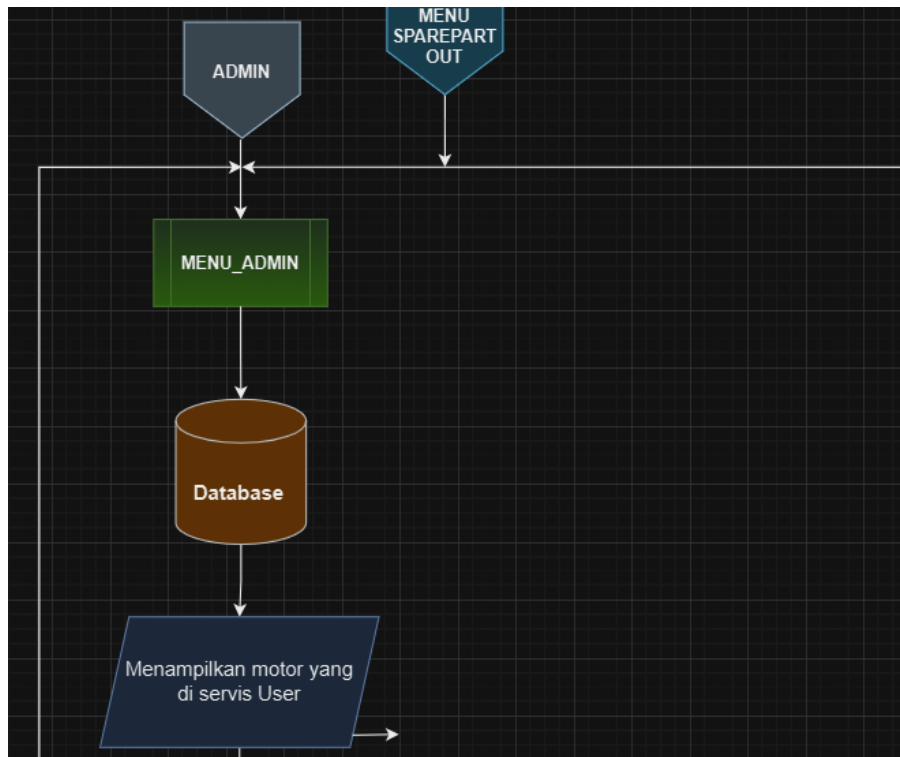
Gambar 2.2.16 Alur pilihan 4 menu user



Gambar 2.2.17 Alur kembali ke menu utama

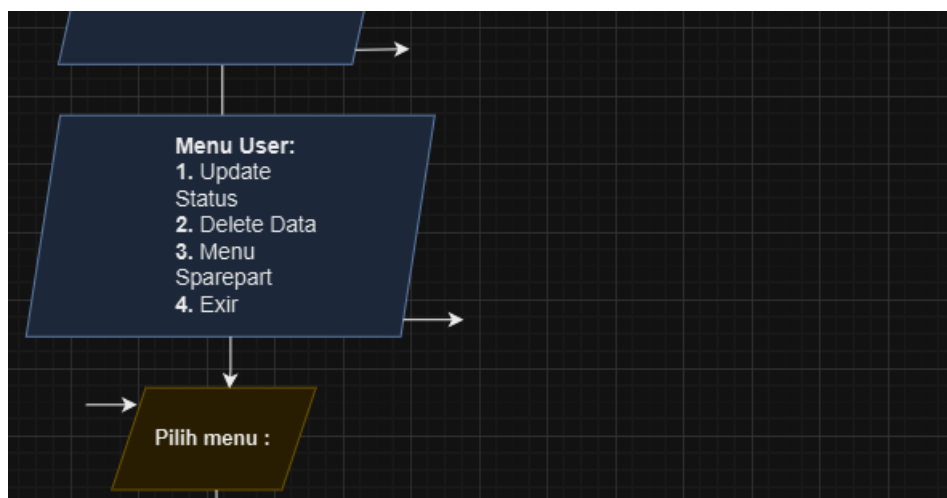
Apabila pengguna memilih opsi “4”, maka pengguna akan keluar dari menu user dan masuk ke menu utama program. Jika pengguna memasukkan opsi selain yang tersedia di menu, maka program akan menampilkan pesan kesalahan dan program diulang ke menu user.

Menu Admin



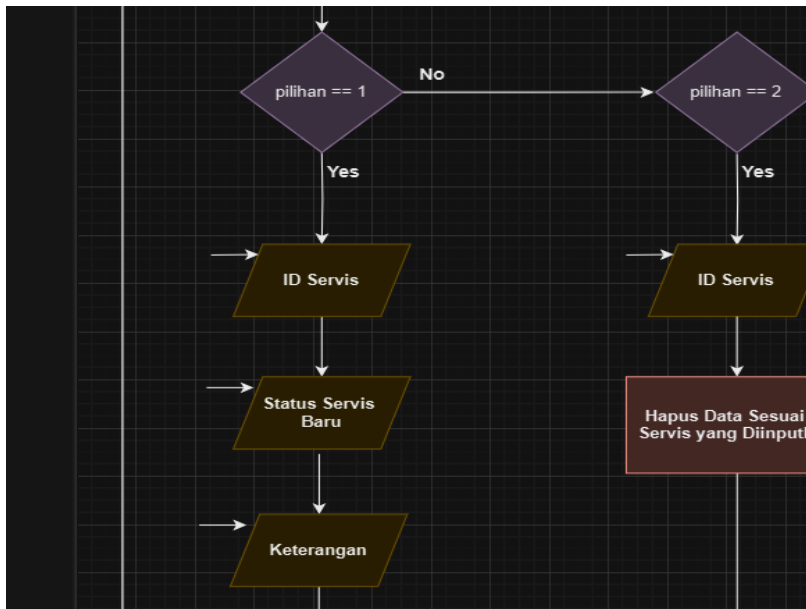
Gambar 2.2.18 Alur menu admin

Pada menu admin, program akan menampilkan data motor yang sedang diservice oleh user jika sebelumnya pengguna melakukan servis motor.



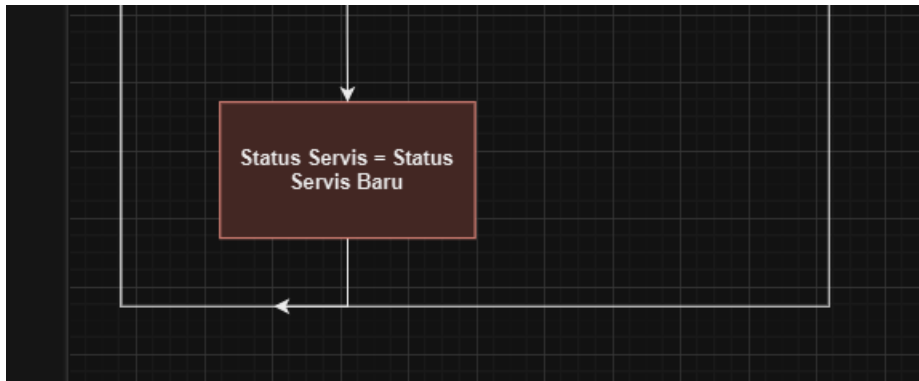
Gambar 2.2.19 Alur isi menu admin

Kemudian program akan menampilkan beberapa menu, diantaranya ada menu update status, delete data motor, menu sparepart dan menu exit. Menu update status dapat dipilih admin apabila ingin mengupdate status servis motor. Menu delete data dapat dipilih admin apabila ingin menghapus data motor dari tampilan karena motor telah selesai diservis. Menu sparepart dapat dipilih admin apabila ingin menambah, mengupdate, menghapus sparepart motor. Menu exit dapat dipilih apabila ingin keluar dari menu admin. Program meminta pengguna untuk memilih satu opsi sesuai angka yang terdapat dalam menu.



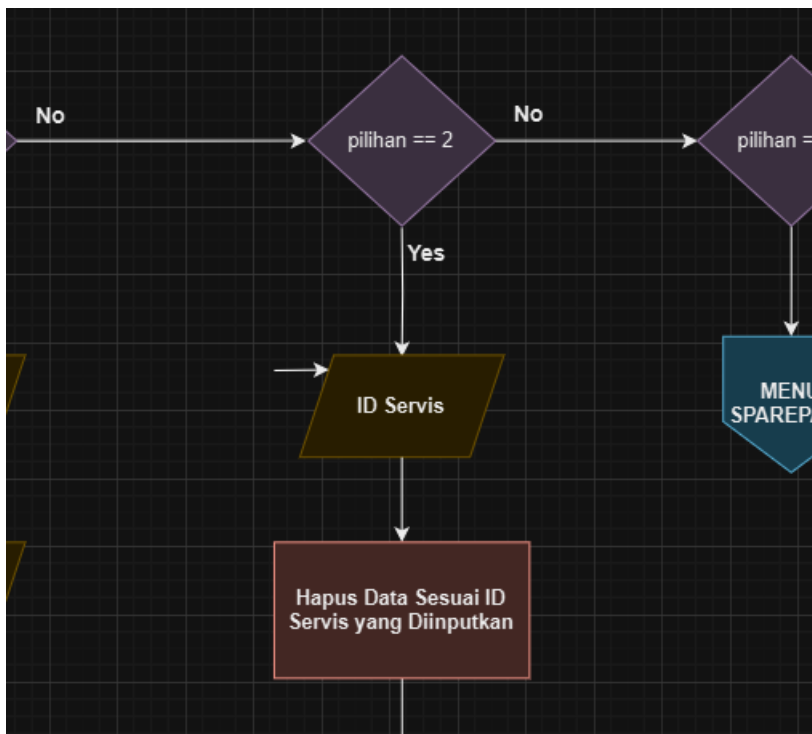
Gambar 2.2.20 Alur pilihan 1 menu admin

Apabila pengguna memilih opsi “1”, maka pengguna akan masuk ke menu update status. Program akan meminta pengguna untuk menginputkan ID servis, status servis yang baru dan keterangan.



Gambar 2.2.21 Alur mengubah status servis

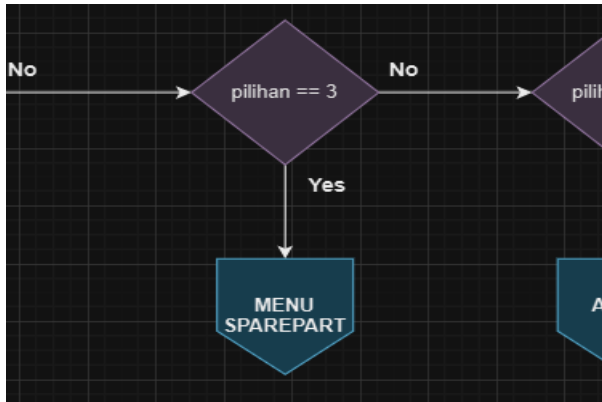
Program akan mengubah status yang lama dengan status yang baru, setelah itu program akan kembali ke menu awal admin.



Gambar 2.2.22 Alur pilihan 2 menu admin

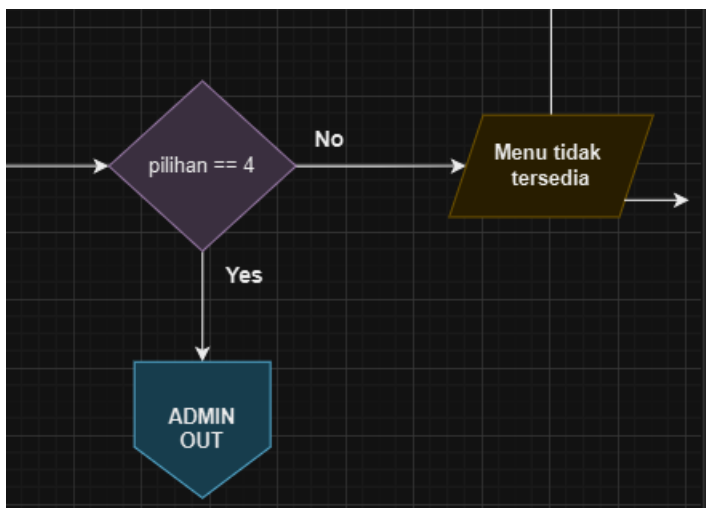
Apabila pengguna memilih opsi “2”, maka pengguna akan masuk ke menu delete data. Program akan meminta pengguna untuk menginputkan ID servis yang

sesuai dengan data, setelah itu program akan menghapus data sesuai dengan ID servis yang diberikan. Lanjut, program akan kembali ke menu awal admin.



Gambar 2.2.23 Alur pilihan 3 menu admin

Apabila pengguna memilih opsi “3”, maka pengguna akan masuk ke menu sparepart. Didalam menu sparepart pengguna dapat menambah, mengubah dan menghapus sparepart motor sesuai dengan keinginan pengguna.

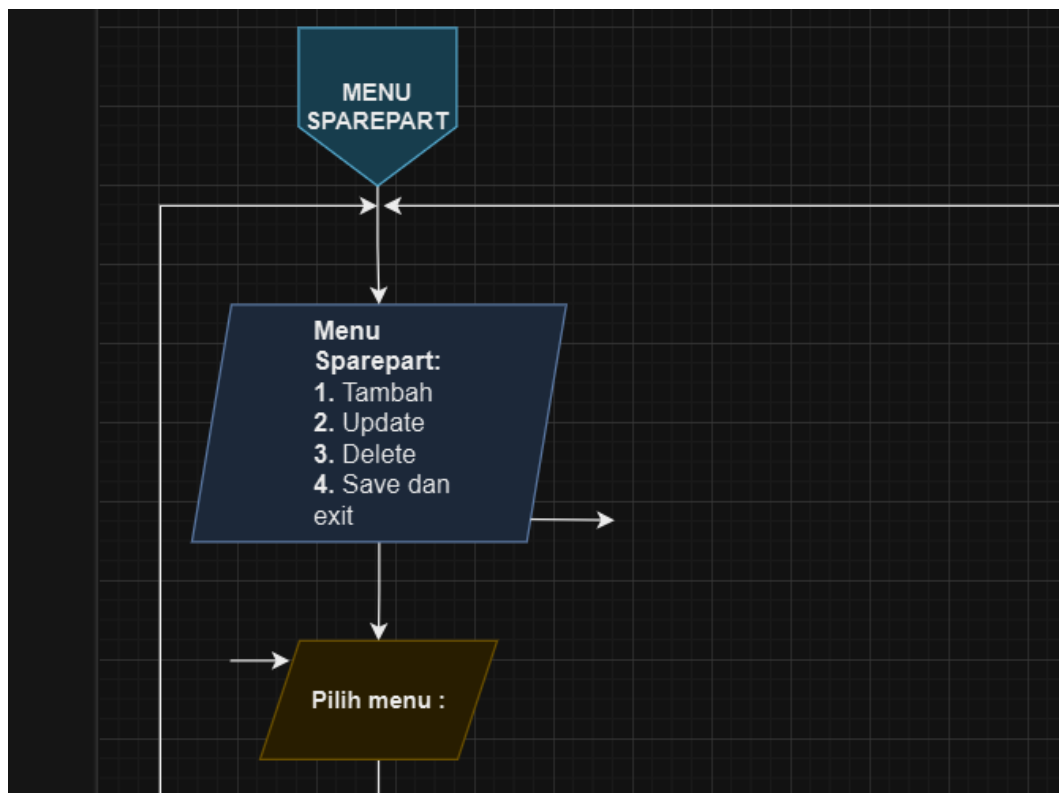


Gambar 2.2.24 Alur pilihan 4 menu admin

Apabila pengguna memilih opsi “4”, maka pengguna akan keluar dari menu admin dan masuk ke menu utama program. Jika pengguna memasukkan opsi selain

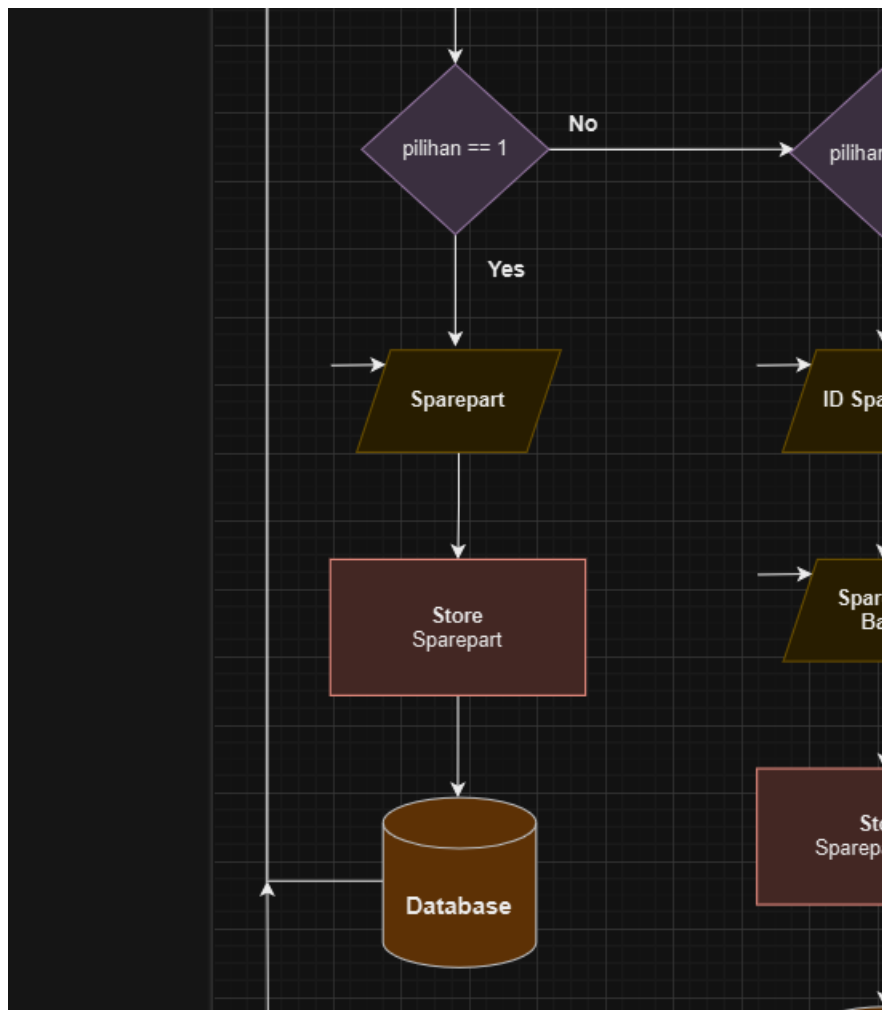
yang tersedia di menu, maka program akan menampilkan pesan kesalahan dan program diulang ke menu admin

Menu Sparepart



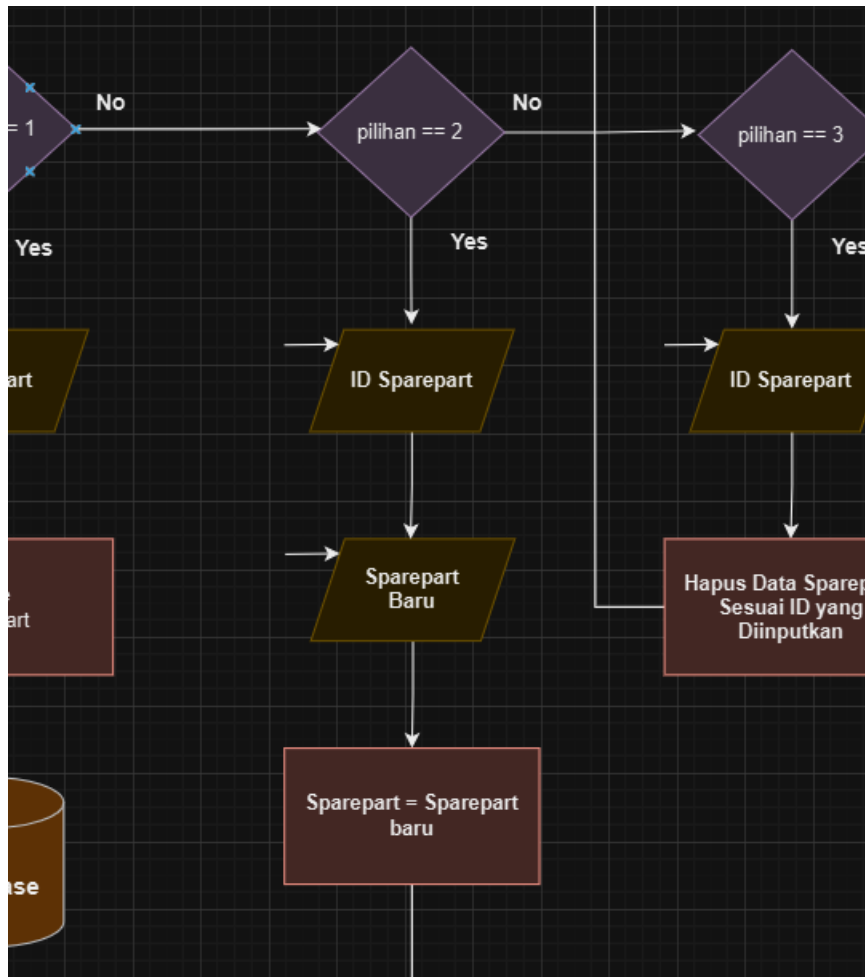
Gambar 2.2.25 Alur plihan menu sparepart

Program akan menampilkan beberapa menu, diantaranya ada menu tambah, update dan delete sparepart. Menu tambah dapat dipilih admin apabila ingin menmabah sparepart yang dibutuhkan motor. Menu update dapat dipilih apabila ingin mengubah sparepart motor. Menu delete dapat dipilih admin apabila ingin menghapus sparepart motor. Menu exit dapat dipilih admin apabila ingin keluar dari menu sparepart dan beralih ke menu utama admin. Program meminta pengguna untuk memilih satu opsi sesuai angka yang terdapat dalam menu.



Gambar 2.2.26 Alur pilihan 1 menu sparepart

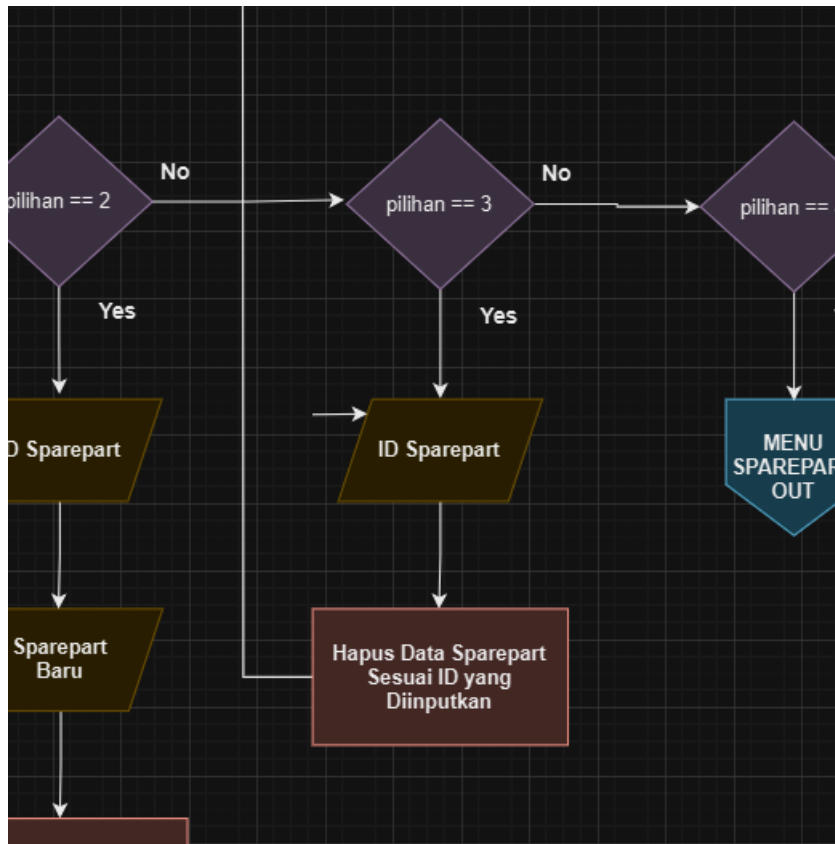
Apabila pengguna memilih opsi “1”, maka pengguna akan masuk ke menu tambah sparepart. Program akan meminta pengguna untuk menginputkan sparepart. Kemudian program akan memproses data yang telah diinputkan tadi dan memasukkan data tadi ke sebuah database, setelah itu program akan kembali ke menu awal sparepart.



Gambar 2.2.27 Alur pilihan 2 menu sparepart

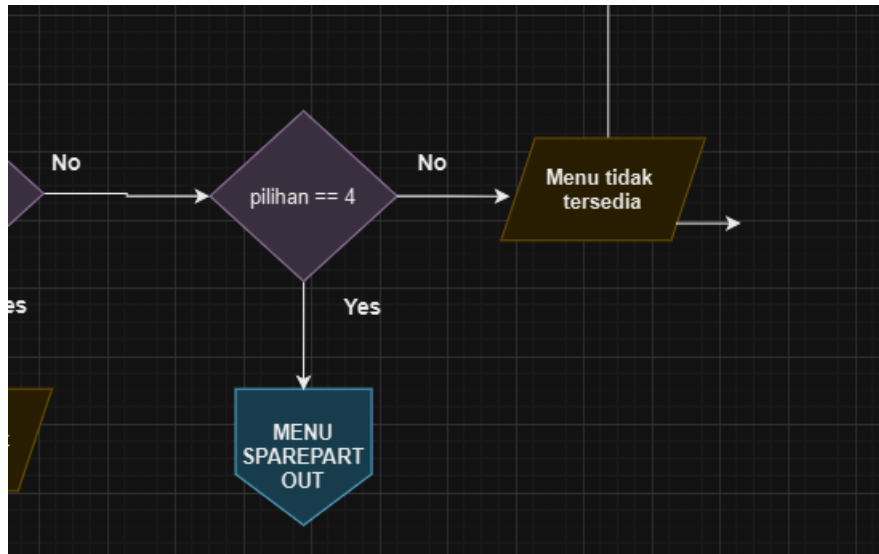
Apabila pengguna memilih opsi “2”, maka pengguna akan masuk ke menu delete sparepart. Program akan meminta pengguna untuk menginputkan ID

sparepart dan sparepart baru. Kemudian program akan mengubah sparepart yang lama dengan yang baru, setelah itu program akan kembali ke menu awal sparepart.



Gambar 2.2.28 Alur pilihan 3 menu sparepart

Apabila pengguna memilih opsi “3”, maka pengguna akan masuk ke menu delete sparepart. Program akan meminta pengguna untuk menginputkan ID sparepart. Kemudian program akan menghapus sparepart sesuai dengan ID sparepart, setelah itu program akan kembali ke menu awal sparepart.



Gambar 2.2.29 Alur pilihan 4 menu sparepart

Apabila pengguna memilih opsi “4”, maka pengguna akan keluar dari menu sparepart dan masuk ke menu admin. Jika pengguna memasukkan opsi selain yang tersedia di menu, maka program akan menampilkan pesan kesalahan dan program diulang ke menu awal sparepart.

2.3 Konsep/Materi Praktikum yang dipakai

Konsep yang kami gunakan dalam pembuatan laporan ini sesuai dengan persyaratan yang terdapat pada modul diantaranya:

Dalam menyusun laporan ini, kami menerapkan konsep-konsep berikut sesuai dengan

modul praktikum :

1. Built-in Function Python/Fungsi bawaan

- input() dan print()

Fungsi `input()` memungkinkan pengguna memberikan instruksi atau data ke program dengan mengetikkannya. Fungsi `print()` mengirimkan output ke pengguna dengan menampilkan pesan atau hasil di layar.

- `len()`

Fungsi `len()` adalah fungsi yang dapat digunakan untuk menghitung jumlah karakter dalam string (termasuk spasi). Fungsi `len()` akan mengembalikan nilai berupa angka/integer (Dr. Joseph Teguh Santoso, 2022).

- `strip()`

Fungsi `strip()` adalah fungsi yang dapat digunakan untuk menghapus spasi awal dan spasi akhir. Fungsi ini memudahkan kita untuk menghilangkan spasi putih atau white space (Dr. Joseph Teguh Santoso S. M., 2023).

2. Looping/Perulangan

Perulangan atau disebut Looping adalah algoritma yang kegunaannya untuk mengulang sebuah perintah/intruksi yang dibuat dalam script sesuai dengan jumlah yang telah ditentukan. Perulangan juga bisa diartikan sebagai suatu proses dimana kita dapat mengulang satu atau beberapa perintah atau eksekusi blok kode tanpa henti, selama kondisi yang dijadikan acuan terpenuhi. Biasanya disiapkan variabel untuk iterasi atau variabel penanda kapan perulangan akan diberhentikan (Rahmat, A'zizah, & Mulyani, 2022).

- Perulangan FOR

Perulangan FOR merupakan sebuah pernyataan mengulang suatu proses yang telah diketahui jumlah perulangannya. Ciri utama

penggunaan pernyataan perulangan FOR itu terdapat nilai awal dan nilai akhir yang menunjukkan syarat yang harus terpenuhi.

- Perulangan WHILE

Perulangan WHILE merupakan sebuah pernyataan mengulang suatu proses yang belum diketahui jumlah perulangannya. Ciri utama penggunaan pernyataan perulangan WHILE adalah akan diuji syarat yang ditentukan terlebih dahulu. Sehingga jika syarat tidak terpenuhi maka pernyataan pada perulangan akan berhenti.

3. Conditional/Percabangan

Percabangan digunakan untuk memecahkan masalah yang mempunyai pernyataan yang banyak dan mengambil Keputusan di antara banyaknya pernyataan yang ada. Dengan adanya struktur percabangan ini kita dapat menyeleksi antara pernyataan yang satu dengan lain dan mengambil Keputusan dalam menyelesaikan suatu masalah dalam pemrograman.

Pemilihan atau percabangan menggunakan statemen If, If/Else, If/Elif/Else. Statemen If digunakan saat terdapat satu pilihan, statemen If/Else digunakan saat terdapat dua pilihan, statemen If/Elif/Else digunakan apabila terdapat lebih dari dua pilihan (Romzi & Kurniawan, 2022).

4. List

List adalah tipe data kolektif yang paling umum digunakan di Python untuk menyimpan kumpulan item yang diurutkan dan dapat diubah. Untuk mendefinisikan sebuah list Anda cukup buat sebuah variabel dengan nama yang diinginkan, kemudian isi variabel tersebut dengan list yang akan dibuat. List diawali dengan tanda '[' dan diakhiri dengan tanda ']'. Isinya dapat beragam, dapat string, number, object, bahkan list lagi. Pemisah antara data yang satu dengan yang lainnya digunakan tanda koma. List dapat ditambah isinya, dirubah data pada elemennya, hapus elemen, dan hapus seluruh list.

5. Dictionary

Dictionary adalah sebuah tipe data bawaan Python yang dapat menyimpan key-value dengan value dapat berupa tipe data apapun termasuk dictionary itu sendiri. Beda halnya dengan list, pada dictionary setiap data akan memiliki pengenalnya masing – masing. Pengenal tersebut dinamakan dengan key dan datanya dinamakan dengan value. Dictionary diawali dengan tanda '{' dan diakhiri dengan tanda '}'. Khusus untuk key pada dictionary, nilainya harus berupa tipe data yang tidak dapat diganti seperti tuple, string dan number. Tapi umumnya key berisi number dan string.

6. Fungsi

Function (fungsi) adalah blok kode yang dirancang untuk melakukan tugas tertentu dan dapat dipanggil berkali-kali dalam program. Function memungkinan programmer membagi kode menjadi segmen-segmen kecil yang lebih terkelola, masing-masing melakukan bagian tertentu dari tugas yang lebih besar. Tidak hanya membantu dalam mengorganisasi kode secara lebih efisien, hal ini juga memudahkan pemeliharaan dan pengujian kode. Setiap function biasanya memiliki nama unik dan bisa menerima *input* dalam bentuk parameter, sehingga dapat bekerja dengan data yang berbeda-beda setiap kali dipanggil. Function juga bisa mengembalikan nilai sebagai output. Fungsi yang tidak mengembalikan nilai disebut dengan prosedur. Dalam bahasa pemrograman Python fungsi direpresentasikan dengan keyword `def`.

7. Error Handling

Error Handling adalah satu penanganan kesalahan (error) pada berbagai macam keadaan dalam pemrograman. Setiap ada kesalahan, maka eksekusi program tidak akan dihentikan secara tiba tiba, tetapi akan diteruskan ke baris program yang terdapat script penanganan kesalahan. Error handling membantu menjaga kelancaran jalannya eksekusi program.

8. File Eksternal

File eksternal adalah file yang disimpan permanen dalam media seperti hard drive atau SSD. Biasanya jika kita tidak menggunakan file, hasil pemrosesan data hanya akan disimpan di main memory. Setelah program dihentikan atau tiba – tiba computer mati, semua data akan hilang. Untuk menyimpan data agar bisa diproses di kesempatan selanjutnya, misal komputer dimatikan kemudian dinyalakan lagi hari esoknya. Kita butuh sebuah penyimpanan yang bersifat resident dan disimpan di secondary storage seperti hard disk. File eksternal diperlukan untuk menyimpan data yang perlu diakses kembali setelah program ditutup

BAB III

HASIL DAN PEMBAHASAN

3.1 Tampilan Program

1. Menu Utama

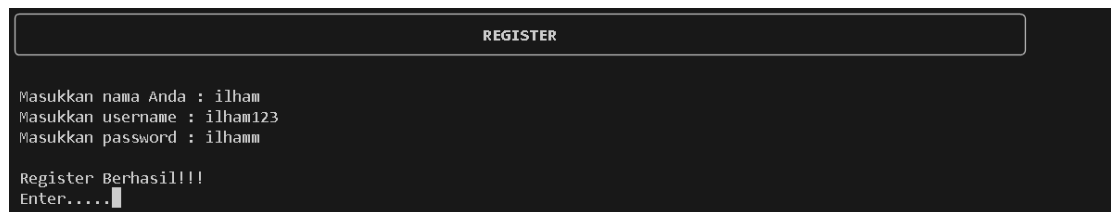


Gambar 3.1.1 Tampilan Menu Awal

Di Awal program kita ada tampilan menu utama, di menu utama ini ada 3 pilihan yaitu

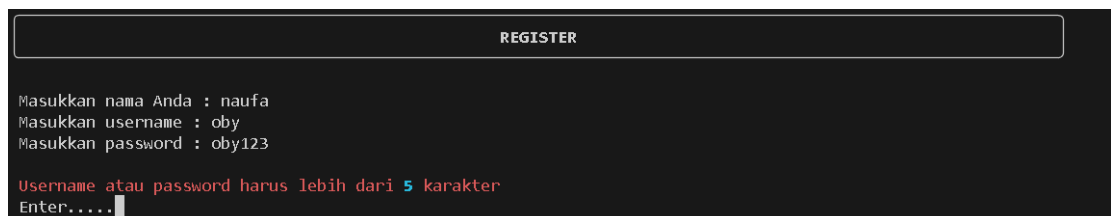
login, register, dan exit.

2. Register

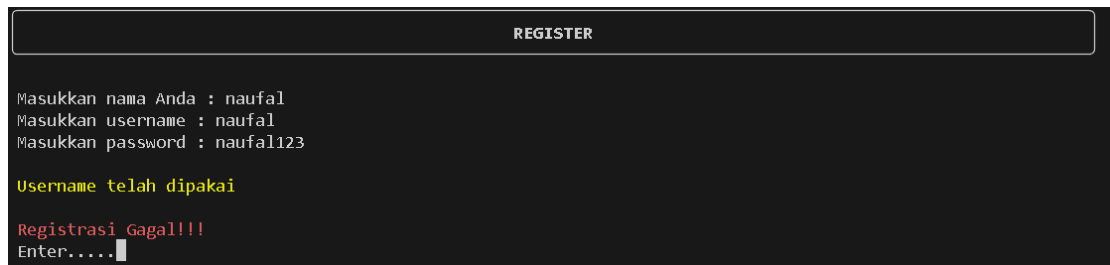


Gambar 3.1.2 Tampilan Register Berhasil

disini kita diminta memasukkan nama, username, dan password

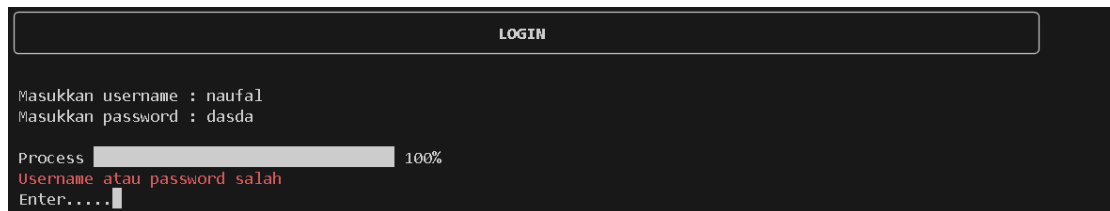


Gambar 3.1.3 Tampilan Register jika username / password kurang dari 5 karakter di menu register jika kita memasukkan username atau password kurang dari 5 karakter maka akan muncul notif seperti diatas.



Gambar 3.1.4 Tampilan Register Jika Username Sudah Ada kemudian ini tampilan jika kita memasukkan username yang sama maka akan muncul notif seperti diatas.

3. Login



Gambar 3.1.5 Tampilan Login Gagal Jika Memasukkan Username / Password Yang Salah

ini tampilan ketika jika kita memasukkan username atau password yang salah. maka akan muncul notif seperti diatas.

4. Exit

```
SERVIS MOTOR

LOGIN    >> 1
REGISTER >> 2
EXIT     >> 3

Masukkan pilihan Anda : 3

Anda telah keluar dari Program
```

Gambar 3.1.6 Tampilan Jika Memilih Exit

Ini tampilan ketika kita exit. akan muncul notif seperti diatas.

5. Tampilan Menu User

```
MENU USER
```

IDservis	name_of_motorcycle	owner	servis	kilometer	sparepart	status	keterangan

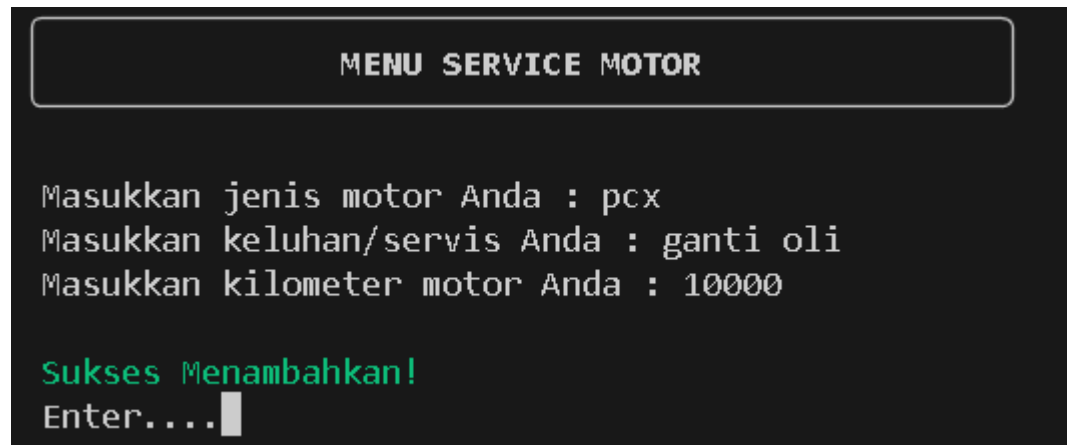
```
SERVICE MOTOR    >> 1
UPDATE KELUHAN    >> 2
HISTORI SERVIS    >> 3
EXIT              >> 4

Masukkan pilihan Anda : █
```

Gambar 3.1.7 Tampilan Menu User

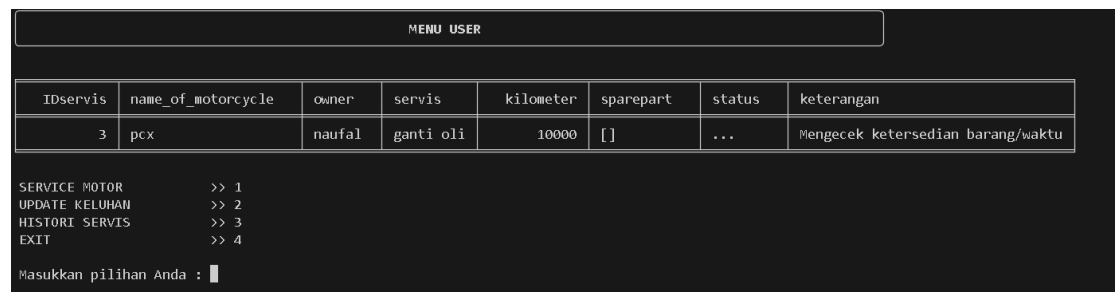
ini adalah tampilan utama menu user, sebagai user kita bisa melakukan service motor, mengupdate keluhan, dan melihat histori servis.

6. Menu Servis Motor



Gambar 3.1.8 Tampilan Menu Service Motor

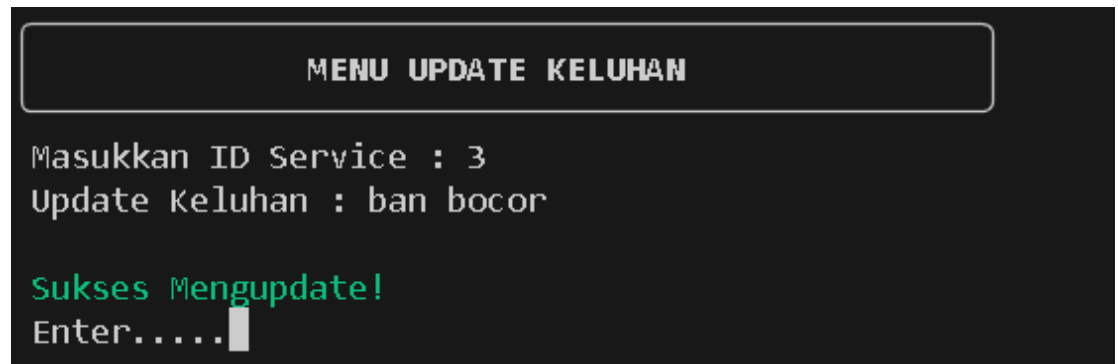
Di menu ini kita diminta memasukkan jenis motor, keluhan / servis yang ingin dilakukan, dan kilometer dari motor kita. dan ketika selesai memasukkan maka akan muncul notif seperti diatas.



Gambar 3.1.9 Tampilan Sesudah

ini tampilan ketika sudah memasukkan apa yang diminta sebelumnya. tampilan akan berubah sesuai yang kita masukkan.

7. Update Keluhan



Gambar 3.1.10 Tampilan Update Keluhan

Ini tampilan ketika kita ingin mengupdate keluhan, kita diminta memasukkan id service agar bisa tau id mana yang mau di update. kemudian kita diminta memasukkan keluhan yang baru. ketika sudah selesai memasukkan maka akan muncul notif seperti diatas.



Gambar 3.1.11 Tampilan Setelah di Update

ini tampilan saat sesudah di update, berubah di bagian servis.

8. Histori Servis

HISTORI SERVICE							
IDservis	name_of_motorcycle	owner	servis	kilometer	sparepart	status	keterangan
3	pcx	naufal	ban bocor	10000	[]	diterima	dalam antrian

Enter....

Gambar 3.1.12 Tampilan Histori Servis

Ini tampilan histori service, akan bertambah ketika admin ingin menghapus service.

9. Tampilan Menu Admin

MENU ADMIN							
IDservis	name_of_motorcycle	owner	servis	kilometer	sparepart	status	keterangan
3	pcx	naufal	ban bocor	10000	[]	...	Mengecek ketersediaan barang/waktu

UPDATE STATUS SERVICE MOTOR >> 1
 DELETE DATA MOTOR >> 2
 MENU SPAREPART >> 3
 EXIT >> 4

Masukkan pilihan Anda :

Gambar 3.1.13 Tampilan Menu Admin

Ini tampilan menu admin, di menu admin kita bisa mengupdate status servis, menghapus data motor, ada menu sparepart, dan exit.

10. Menu Update Status

MENU UPDATE STATUS
Masukkan ID Service : 3 Update status (diterima, ditolak, proses, finish, sudah di ambil) : diterima Update Keterangan : dalam antrian Sukses Mengupdate! Enter.....

Gambar 3.1.14 Tampilan Menu Update Status

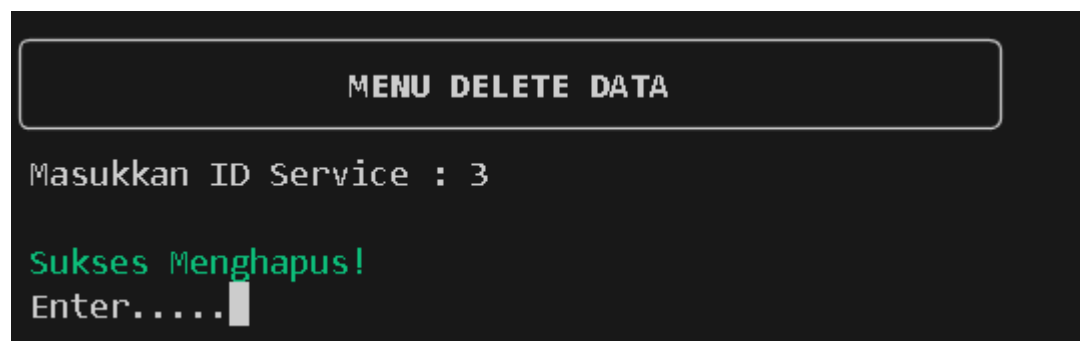
Ini tampilan ketika kita ingin mengupdate status servis, kita diminta memasukkan id yang ingin diupdate, kemudian memasukkan status yang diinginkan, dan memasukkan keterangan. ketika sudah memasukkan semua maka akan muncul notif seperti diatas.



Gambar 3.1.15 Tampilan Setelah Mengupdate Status

Ini tampilan ketika berhasil mengupdate status, bagian status dan keterangan di tabel berubah.

11. Menu Delete Data Motor



Gambar 3.1.16 Tampilan Menu Delete Data Motor

Ini tampilan ketika kita ingin menghapus data motor, kita diminta memasukkan id yang ingin dihapus. dan akan muncul notif seperti diatas.



Gambar 3.1.17 Tampilan Setelah Delete Data Motor

Ini tampilan setelah menghapus data motor, seluruh bagian tabel kosong.

12. Tampilan Menu Sparepart

MENU SPAREPART							
IDservis	name_of_motorcycle	owner	servis	kilometer	sparepart	status	keterangan
4	pcx	naufal	ganti oli	100000	[]	proses	sedang di prosesS
TAMBAH SPAREPART >> 1							
UPDATE SPAREPART >> 2							
DELETE SPAREPART >> 3							
SAVE AND EXIT FROM SPAREPART MENU >> 4							
Masukkan pilihan Anda : █							

Gambar 3.1.18 Tampilan Menu Sparepart

ini tampilan menu sparepart. di menu ini kita bisa menambah sparepart, mengupdate, dan menghapus.

13. Menu Tambah Sparepart

Menu Add Sparepart

Masukkan Sparepart : oli spx 1

Sukses Menambahkan!

Enter.... █

Gambar 3.1.19 Tampilan Menu Tambah Sparepart

ini tampilan ketika ingin menambah sparepart. kita diminta memasukkan sparepart yang diinginkan.

MENU SPAREPART							
IDservis	name_of_motorcycle	owner	servis	kilometer	sparepart	status	keterangan
4	pcx	naufal	ganti oli	100000	['oli spx 1']	proses	sedang di proses

TAMBAH SPAREPART

UPDATE SPAREPART

DELETE SPAREPART

SAVE AND EXIT FROM SPAREPART MENU

Masukkan pilihan Anda :

>> 1

>> 2

>> 3

>> 4

Gambar 3.1.20 Tampilan Ketika Sudah Ditambahkan

ini tampilan sesudah ditambahkan

14. Menu Update Sparepart

Menu Update Sparepart

1. oli spx 1

Masukkan No Sparepart : 1

Masukkan Sparepart : oli cpx 2

Sukses Mengupdate!

Enter....

Gambar 3.1.21 Tampilan Menu Update Sparepart

Ini tampilan menu update sparepart, kita diminta memasukkan no sparepart yang ingin diubah kemudian memasukkan sparepart yang baru.

MENU SPAREPART							
IDservis	name_of_motorcycle	owner	servis	kilometer	sparepart	status	keterangan
4	pcx	naufal	ganti oli	100000	['oli cpx 2']	proses	sedang di proses

TAMBAH SPAREPART >> 1
 UPDATE SPAREPART >> 2
 DELETE SPAREPART >> 3
 SAVE AND EXIT FROM SPAREPART MENU >> 4
 Masukkan pilihan Anda :

Gambar 3.1.22 Tampilan Setelah Mengupdate Sparepart

Ini tampilan ketika sudah mengupdate sparepart

15. Menu Delete Sparepart



Gambar 3.1.23 Tampilan Menu Delete Sparepart

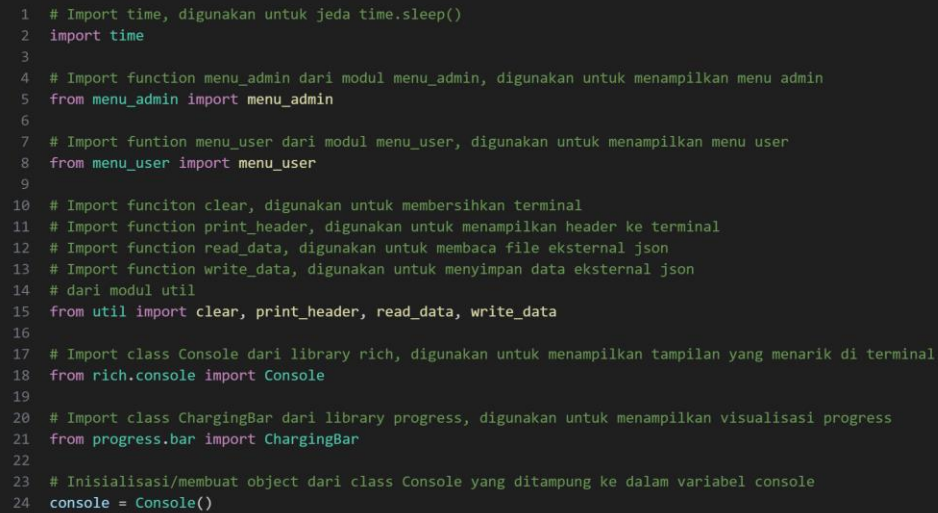
Ini tampilan menu delete sparepart, kita diminta untuk memasukkan nomor sparepart yang ingin dihapus. kemudian akan muncul notif seperti diatas



Gambar 3.1.24 Tampilan Setelah Di Delete

3.2 Source Code

1. Main



```
1 # Import time, digunakan untuk jeda time.sleep()
2 import time
3
4 # Import function menu_admin dari modul menu_admin, digunakan untuk menampilkan menu admin
5 from menu_admin import menu_admin
6
7 # Import function menu_user dari modul menu_user, digunakan untuk menampilkan menu user
8 from menu_user import menu_user
9
10 # Import function clear, digunakan untuk membersihkan terminal
11 # Import function print_header, digunakan untuk menampilkan header ke terminal
12 # Import function read_data, digunakan untuk membaca file eksternal json
13 # Import function write_data, digunakan untuk menyimpan data eksternal json
14 # dari modul util
15 from util import clear, print_header, read_data, write_data
16
17 # Import class Console dari library rich, digunakan untuk menampilkan tampilan yang menarik di terminal
18 from rich.console import Console
19
20 # Import class ChargingBar dari library progress, digunakan untuk menampilkan visualisasi progress
21 from progress.bar import ChargingBar
22
23 # Inisialisasi/membuat object dari class Console yang ditampung ke dalam variabel console
24 console = Console()
```

Gambar 3.2.1 Source Code Import

Gambar di atas adalah source code import beberapa module dan inisialisasi object. Module module dan object tersebut nantinya akan digunakan dalam source code Main.

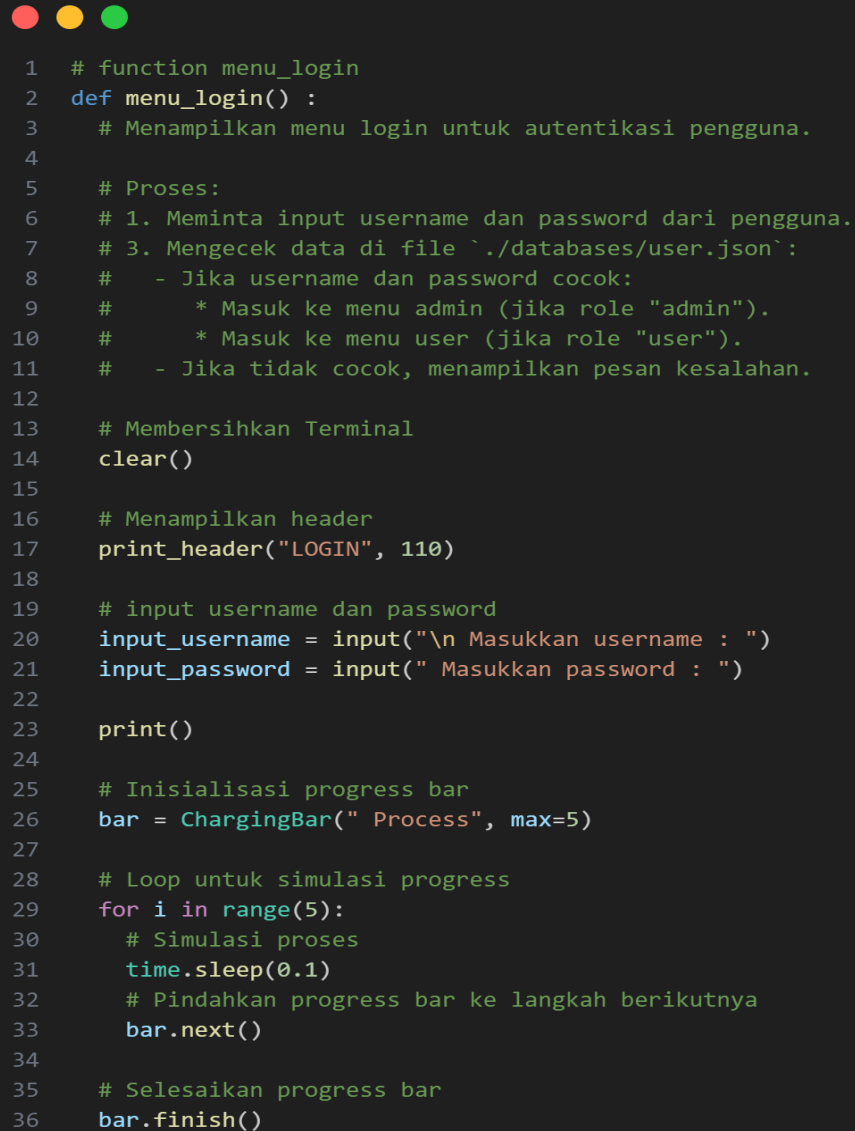
```

1  # Menu utama
2  while True :
3      # Membersihkan Terminal
4      clear()
5
6      # Menampilkan header dan beberapa menu
7      print_header("SERVIS MOTOR", 110)
8      print("\n LOGIN    >> 1")
9      print(" REGISTER >> 2")
10     print(" EXIT     >> 3")
11     print()
12
13     # Input pengguna sesuai menu yang telah tersedia
14     input_user = input(" Masukkan pilihan Anda : ")
15
16     if input_user == "1" :
17
18         # Masuk ke menu login
19         menu_login()
20
21     elif input_user == "2" :
22
23         # Masuk ke menu register
24         menu_register()
25
26     # Keluar dari program
27     elif input_user == "3" :
28         print()
29         console.print(" Anda telah keluar dari Program", style="red")
30         break
31
32     # Kode di bawah ini akan dieksekusi apabila input dari pengguna tidak sesuai dengan menu yang tersedia
33     else :
34         print()
35         console.print("\n Mohon maaf, menu tidak tersedia!!!", style="red")
36         input(" Enter.....")
37         clear()

```

Gambar 3.2.2 Source Utama

Ini adalah Source code utama dari keseluruhan program. Source code tersebut menampilkan menu utama. Ada tiga menu utama dari program diantaranya menu login, register dan exit.



```

1  # function menu_login
2  def menu_login() :
3      # Menampilkan menu login untuk autentikasi pengguna.
4
5      # Proses:
6      # 1. Meminta input username dan password dari pengguna.
7      # 3. Mengecek data di file `./databases/user.json`:
8      #     - Jika username dan password cocok:
9      #         * Masuk ke menu admin (jika role "admin").
10     #         * Masuk ke menu user (jika role "user").
11     #     - Jika tidak cocok, menampilkan pesan kesalahan.
12
13     # Membersihkan Terminal
14     clear()
15
16     # Menampilkan header
17     print_header("LOGIN", 110)
18
19     # input username dan password
20     input_username = input("\n Masukkan username : ")
21     input_password = input(" Masukkan password : ")
22
23     print()
24
25     # Inisialisasi progress bar
26     bar = ChargingBar(" Process", max=5)
27
28     # Loop untuk simulasi progress
29     for i in range(5):
30         # Simulasi proses
31         time.sleep(0.1)
32         # Pindahkan progress bar ke langkah berikutnya
33         bar.next()
34
35     # Selesaikan progress bar
36     bar.finish()

```

Gambar 3.2.3 Source Code Menu Login

Ini adalah source code function menu_login() yang akan dijalankan jika pengguna menginputkan “1” pada menu utama. Dalam function ini pengguna diminta untuk menginputkan username dan password.

```

1  # Selesaikan progress bar
2  bar.finish()
3
4  is_wrong = True
5
6  # Melakukan iterasi dan membaca data pengguna dari file JSON
7  for data in read_data("./databases/user.json") :
8
9      # Melakukan pengecekan apakah username dan password yang diinputkan user sesuai dengan data dari file
10     if data["username"] == input_username and data["password"] == input_password and data["role"] == "admin":
11         console.print("\n BERHASIL LOGIN", style="bold green")
12         time.sleep(1)
13
14         # Masuk ke menu admin
15         menu_admin()
16
17         # Reassign variabel is_wrong dengan nilai boolean "False" karena username dan password sesuai
18         is_wrong = False
19
20         # Memberhentikan perulangan
21         break
22
23     # Melakukan pengecekan apakah username dan password yang diinputkan user sesuai dengan data dari file
24     if data["username"] == input_username and data["password"] == input_password and data["role"] == "user":
25         console.print("\n BERHASIL LOGIN", style="green")
26         time.sleep(1)
27
28         # Masuk ke menu user
29         menu_user(data["name"])
30
31         # Reassign variabel is_wrong dengan nilai boolean "False" karena username dan password sesuai
32         is_wrong = False
33
34         # Memberhentikan perulangan
35         break
36
37     # Kode di bawah ini akan dieksekusi apabila pengguna salah memasukkan username atau password
38     if is_wrong:
39         console.print(" Username atau password salah", style="red")
40         input(" Enter.....")

```

Gambar 3.2.4 Source Code Lanjutan Menu Login

Ini adalah source code lanjutan menu_login() yang akan memverifikasi data yang diinputkan oleh pengguna tadi. Di dalam source code ini sistem akan mengverifikasi username, password dan role dari pengguna tadi. Jika pengguna adalah seorang admin, maka function menu_admin() akan dieksekusi. Jika pengguna adalah seorang user, maka function menu_user() akan dieksekusi.

```

1  def menu_register() :
2      # Menampilkan menu registrasi untuk pendaftaran pengguna baru.
3
4      # Proses:
5      # 1. Meminta input nama, username, dan password dari pengguna.
6      # 2. Mengecek apakah username sudah digunakan.
7      #    - Jika belum, tambahkan data ke file `./databases/user.json`.
8      #    - Jika sudah, tampilkan pesan kesalahan.
9
10     while True :
11         # Membersihkan Terminal
12         clear()
13
14         # Menampilkan header
15         print_header("REGISTER", 110)
16         # Input nama, username, password
17         input_name_register = input("\n Masukkan nama Anda : ").strip()
18         input_username_register = input(" Masukkan username : ").strip()
19         input_password_register = input(" Masukkan password : ").strip()
20
21         # Mengecek apakah input string kosong atau kurang dari 5 karakter
22         if input_username_register == "" or input_password_register == "" or input_name_register == "":
23             console.print("\n Username atau password tidak boleh kosong", style="red")
24             input(" Enter.....")
25         elif len(input_username_register) <= 5 or len(input_password_register) <= 5 :
26             console.print("\n Username atau password harus lebih dari 5 karakter", style="red")
27             input(" Enter.....")
28         else :
29             break

```

Gambar 3.2.5 Source Code Menu Register

Ini adalah source code function menu_register() yang akan dijalankan jika pengguna menginputkan “2” pada menu utama. Dalam function ini pengguna diminta untuk menginputkan nama, username, dan password. Source code ini akan melakukan pengecekan data yang diinputkan pengguna. Jika nama, username dan password adalah sebuah whitespace maka input dari pengguna tadi ditolak. Jika username dan password tidak lebih dari 5 karakter, maka input dari pengguna tadi juga akan ditolak.

```

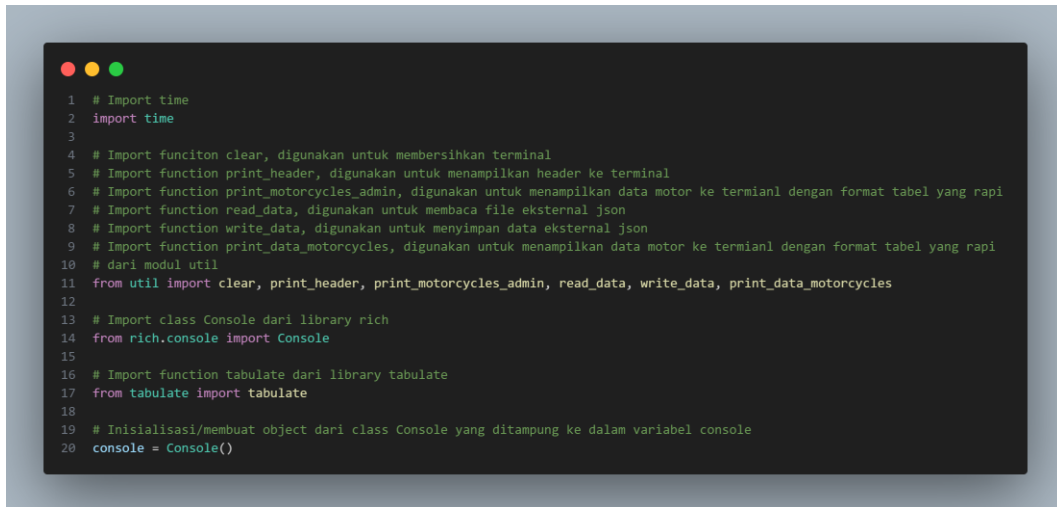
1 elif len(input_username_register) <= 5 or len(input_password_register) <= 5 :
2     console.print("\n Username atau password harus lebih dari 5 karakter", style="red")
3     input(" Enter.....")
4 else :
5     break
6 # Membaca data pengguna dari file external JSON
7 data_users = read_data("./databases/user.json")
8 is_used = False
9 # Mengecek apakah username yang diinputkan pengguna telah digunakan
10 for data in data_users :
11     if data["username"] == input_username_register :
12         is_used = True
13 if is_used == False :
14     # Menambahkan data pengguna baru
15     data_users.append(
16     {
17         "name" : input_name_register,
18         "username" : input_username_register,
19         "password" : input_password_register,
20         "role" : "user"
21     })
22 # Menyimpan data ke file eksternal JSON
23 write_data("./databases/user.json", data_users)
24 print()
25 print(" Register Berhasil!!!")
26 input(" Enter.....")
27 # Kode di bawah ini akan dieksekusi apabila username telah digunakan
28 else :
29     console.print("\n Username telah dipakai", style="yellow")
30     console.print("\n Registrasi Gagal!!!", style="red")
31     input(" Enter.....")

```

Gambar 3.2.6 Source Code Lanjutan Menu Register

Ini adalah source code lanjutan dari menu_register(). Source ini berisi perintah untuk membaca file yang nanti dengannya data akan disimpan. Setelah berhasil membaca file, kemudian file tadi dimanipulasi dengan ditambahkan data baru (data yang diinputkan pengguna tadi). Source code ini juga melakukan pengecekan apakah username yang diinputkan pengguna telah terpakai oleh pengguna lain. Jika iya, maka data gagal untuk ditambahkan ke dalam file. Langkah terakhir yang dilakukan source code ini adalah menyimpan data yang telah dimanipulasi tadi ke dalam sebuah file eksternal.

2. Menu Admin

A screenshot of a code editor with a dark background and light-colored text. The code is Python source code for importing modules and initializing objects for a menu admin. It includes comments in Indonesian explaining the purpose of each import. The code is numbered from 1 to 20 on the left side of the editor.

```
1 # Import time
2 import time
3
4 # Import function clear, digunakan untuk membersihkan terminal
5 # Import function print_header, digunakan untuk menampilkan header ke terminal
6 # Import function print_motorcycles_admin, digunakan untuk menampilkan data motor ke terminal dengan format tabel yang rapi
7 # Import function read_data, digunakan untuk membaca file eksternal json
8 # Import function write_data, digunakan untuk menyimpan data eksternal json
9 # Import function print_data_motorcycles, digunakan untuk menampilkan data motor ke terminal dengan format tabel yang rapi
10 # dari modul util
11 from util import clear, print_header, print_motorcycles_admin, read_data, write_data, print_data_motorcycles
12
13 # Import class Console dari library rich
14 from rich.console import Console
15
16 # Import function tabulate dari library tabulate
17 from tabulate import tabulate
18
19 # Inisialisasi/membuat object dari class Console yang ditampung ke dalam variabel console
20 console = Console()
```

Gambar 3.2.7 Source Code Import Menu Admin

Gambar di atas adalah source code import beberapa module dan inisialisasi object. Module module dan object tersebut nantinya akan digunakan dalam source code menu_admin.

```

1 def menu_admin() :
2     while True :
3         # Membersihkan Terminal
4         clear()
5
6         # Menampilkan header
7         print_header("MENU ADMIN", 110)
8         print()
9
10        # Menampilkan data motor
11        print_motorcycles_admin()
12
13        # Menampilkan menu
14        print("\n UPDATE STATUS SERVICE MOTOR >> 1")
15        print(" DELETE DATA MOTOR          >> 2")
16        print(" MENU SPAREPART              >> 3")
17        print(" EXIT                                >> 4")
18        print()
19
20        # Input pengguna sesuai menu yang telah tersedia
21        input_user = input(" Masukkan pilihan Anda : ")
22        if input_user == "1" :
23
24            # Masuk ke menu update status service
25            update_status_service()
26
27        elif input_user == "2" :
28
29            # Masuk ke menu delete data motor
30            delete_data_motor()
31
32        elif input_user == "3" :
33
34            # Masuk ke menu sparepart
35            menu_sparepart_main()
36
37        # Keluar dari akun
38        elif input_user == "4" :
39            console.print(" LOG OUT...", style="yellow")
40            time.sleep(2)
41            break
42
43        # Kode di bawah ini akan dieksekusi apabila input dari pengguna tidak sesuai dengan menu yang tersedia
44        else :
45            print()
46            console.print("\n Mohon maaf, menu tidak tersedia!!!", style="red")
47            input(" Enter.....")
48            clear()

```

Gambar 3.2.8 Source Code Menu Admin

Ini adalah source code function menu_admin() yang akan dijalankan jika sistem telah memverifikasi bahwa pengguna adalah admin. Di dalam function ini terdapat empat menu, diantaranya menu update status servis motor, delete data motor, menu sparepart, dan exit. Di dalam function ini pengguna diminta untuk menginputkan pilihan berdasarkan menu yang tersedia.

```

1 def update_status_service() :
2     try :
3         print()
4
5         # Menampilkan header
6         print_header("MENU UPDATE STATUS", 50)
7
8         # Input ID service
9         input_user_ID = int(input(" Masukkan ID Service : "))
10
11        # Membaca data motor dari file JSON
12        data_motorcycles = read_data("./databases/motorcycle.json")
13        is_not_match = True
14
15        # Mengecek apakah ID service yang diinputkan pengguna sesuai
16        for item in data_motorcycles :
17            if item["IDservis"] == input_user_ID and item["display"] == "on":
18                # Input status dan keterangan baru
19                input_admin_update_status = input(" Update status (diterima, ditolak, proses, finish, sudah di ambil) : ")
20                input_admin_update_keterangan = input(" Update Keterangan : ")
21
22                # Update status dan keterangan dengan status dan keterangan yang baru
23                item["status"] = input_admin_update_status
24                item["keterangan"] = input_admin_update_keterangan
25
26                # Menyimpan data ke file eksternal JSON
27                write_data("./databases/motorcycle.json", data_motorcycles)
28
29                console.print("\n Sukses Mengupdate!", style="green")
30                input(" Enter.....")
31
32                # Reassign variabel is_not_match dengan nilai boolean "False" karena ID service sesuai
33                is_not_match = False
34                break
35
36        # Kode di bawah ini akan dieksekusi apabila ID service tidak sesuai
37        if is_not_match :
38            console.print("\n ID Servis tidak ditemukan", style="red")
39            input(" Enter.....")
40        # Kode di bawah ini akan dieksekusi apabila pengguna menginputkan karakter selain angka
41        except :
42            console.print("\n Harap masukkan angka", style="red")
43            input(" Enter.....")

```

Gambar 3.2.9 Source Code Update Status Services

Ini adalah source code function `update_status_service()` yang akan dijalankan jika pengguna menginputkan "1". Dalam function ini pengguna diminta untuk menginputkan ID service. Source code ini akan melakukan pengecekan data yang diinputkan pengguna. Sistem akan membaca file data motor terlebih dahulu, kemudian melakukan pengecekan ID service dengan data motor. Jika ID servis dengan data sesuai, maka sistem akan meminta pengguna menginputkan status baru dan keterangannya. Jika tidak, maka update gagal.

```

1 def delete_data_motor() :
2     try :
3         print()
4
5         # Menampilkan header
6         print_header("MENU DELETE DATA", 50)
7
8         # Input ID service
9         input_user_ID = int(input(" Masukkan ID Service : "))
10
11        # Membaca data motor dari file JSON
12        data_motorcycles = read_data("./databases/motorcycle.json")
13        is_not_match = True
14
15        # Mengecek apakah ID service yang diinputkan pengguna sesuai
16        for index, item in enumerate(data_motorcycles) :
17            if item["IDservis"] == input_user_ID and item["display"] == "on":
18
19                # Mengubah nilai properti display dengan nilai "False"
20                data_motorcycles[index]["display"] = "off"
21
22                # Menyimpan data ke file eksternal JSON
23                write_data("./databases/motorcycle.json", data_motorcycles)
24
25                console.print("\n Sukses Menghapus!", style="green")
26                input(" Enter.....")
27
28                # Reassign variabel is_not_match dengan nilai boolean "False" karena ID service sesuai
29                is_not_match = False
30                break
31
32        # Kode di bawah ini akan dieksekusi apabila ID service tidak sesuai
33        if is_not_match :
34            console.print("\n ID Servis tidak ditemukan", style="red")
35            input(" Enter....")
36
37        # Kode di bawah ini akan dieksekusi apabila pengguna menginputkan karakter selain angka
38        except :
39            console.print("\n Harap masukkan angka", style="red")
40            input(" Enter....")

```

Gambar 3.2.10 Source Code Delete Status Services

Ini adalah source code function `delete_status_service()` yang akan dijalankan jika pengguna menginputkan "2". Dalam function ini pengguna diminta untuk menginputkan ID service. Source code ini akan melakukan pengecekan data yang diinputkan pengguna. Sistem akan membaca file data motor terlebih dahulu, kemudian melakukan pengecekan ID service dengan data motor. Jika ID servis dengan data sesuai, maka sistem akan menghapus data motor sesuai ID servis yang diinputkan. Jika tidak, maka delete gagal.

```

1  def menu_sparepart_main() :
2      try :
3          print()
4
5          # Menampilkan header
6          print_header("MENU SPAREPART", 50)
7
8          # Input ID service
9          input_user_ID = int(input(" Masukkan ID Service : "))
10
11         # Membaca data motor dari file JSON
12         data_motorcycles = read_data("./databases/motorcycle.json")
13         is_not_match = True
14
15         # Mengecek apakah ID service yang diinputkan pengguna sesuai
16         for item in data_motorcycles :
17             if item["IDservis"] == input_user_ID and item["display"] == "on":
18
19                 # Masuk ke menu sparepart
20                 menu_sparepart(item)
21
22                 # Reassign variabel is_not_match dengan nilai boolean "False" karena ID service sesuai
23                 is_not_match = False
24                 break
25         if is_not_match :
26             console.print("\n ID Servis tidak ditemukan", style="red")
27             input(" Enter...")
28             write_data("./databases/motorcycle.json", data_motorcycles)
29         # Kode di bawah ini akan dieksekusi apabila pengguna menginputkan karakter selain angka
30         except :
31             print(" Harap masukkan angka")
32             input(" Enter...")

```

Gambar 3.2.11 Source Code Menu Sparepart Main

Ini adalah source code function menu_sparepart_main() yang akan dijalankan jika pengguna menginputkan “3” . Dalam function ini pengguna diminta untuk menginputkan ID service. Source code ini akan melakukan pengecekan data yang diinputkan pengguna. Sistem akan membaca file data motor terlebih dahulu, kemudian melakukan pengecekan ID service dengan data motor. Jika ID servis dengan data sesuai, maka sistem akan mengalihkan tampilan ke menu sparepart.

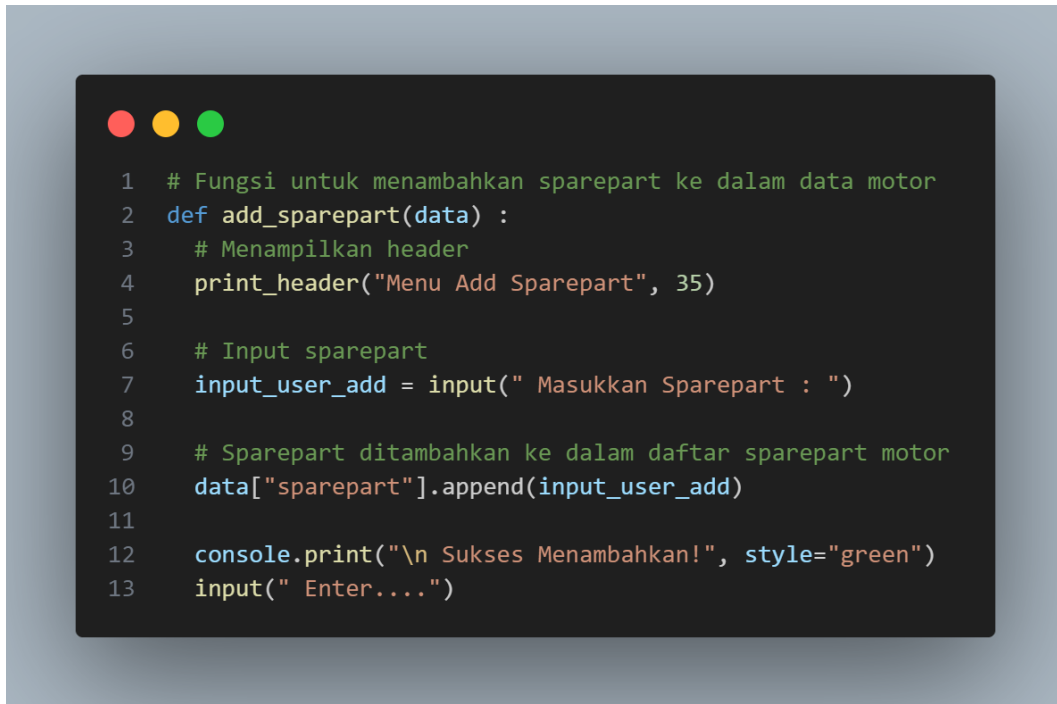
```

1 def menu_sparepart(data) :
2
3     # Menu untuk mengelola sparepart dalam data servis motor.
4
5     # Proses:
6     # 1. Menampilkan sparepart terkait ID servis yang dipilih.
7     # 2. Menyediakan pilihan:
8     #     - Menambah sparepart.
9     #     - Memperbarui sparepart.
10    #     - Menghapus sparepart.
11    #     - Menyimpan perubahan dan keluar.
12    # 3. Memperbarui data sparepart sesuai dengan input pengguna.
13    # Args:
14    #     data (dict): Data servis motor yang dipilih.
15
16    while True :
17        # Membersihkan Terminal
18        clear()
19
20        # Menampilkan header
21        print_header("MENU SPAREPART", 50)
22        print()
23
24        # Menampilkan data motor sesuai dengan data yang dipilih
25        print_data_motorcycles([data])
26        print()
27
28        # Menampilkan menu
29        print("\n TAMBAH SPAREPART                >> 1")
30        print(" UPDATE SPAREPART                >> 2")
31        print(" DELETE SPAREPART                >> 3")
32        print(" SAVE AND EXIT FROM SPAREPART MENU  >> 4")
33
34        # Input pengguna sesuai menu yang telah tersedia
35        input_user = input(" Masukkan pilihan Anda : ")
36
37        if input_user == "1" :
38
39            # Masuk ke menu add sparepart
40            add_sparepart(data)
41        elif input_user == "2" :
42
43            # Masuk ke menu update Sparepart
44            update_sparepart(data)
45
46        elif input_user == "3" :
47
48            # Masuk ke menu delete sparepart
49            delete_sparepart(data)
50
51        # Menyimpan dan keluar dari menu sparepart
52        elif input_user == "4" :
53            console.print(" EXIT FROM Sparepart MENU", style="yellow")
54            time.sleep(2)
55            break
56
57        # Kode di bawah ini akan dieksekusi apabila input dari pengguna tidak sesuai dengan menu yang tersedia
58        else :
59            print()
60            console.print("\n Mohon maaf, menu tidak tersedia!!!", style="red")
61            input(" Enter.....")
62            clear()

```

Gambar 3.2.12 Source Code Menu Sparepart

Ini adalah source code function menu_sparepart() yang akan dijalankan jika pengguna berhasil meinputkan ID service yang benar. Source code ini akan menampilkan 4 menu, diantaranya tambah sparepart, update, delete dan exit.



Gambar 3.2.13 Source Code Add Sparepart

Ini adalah source code function `add_sparepart ()` yang akan dijalankan jika pengguna menginputkan “1” pada menu sparepart. Source code akan meminta pengguna menginputkan sparepart baru, kemudian sistem akan menyimpan sparepart baru tersebut.

```

1  # Fungsi untuk memperbarui sparepart yang sudah ada
2  def update_sparepart(data) :
3      try :
4
5          # Menampilkan header
6          print_header("Menu Update Sparepart", 35)
7          print()
8
9          # Menampilkan sparepart motor
10         for index, item in enumerate(data["sparepart"]) :
11             print(f"{index + 1}. {item}")
12         print()
13
14         # Input no sparepart dan sparepart yang baru
15         input_user_update_no = int(input(" Masukkan No Sparepart : "))
16         input_user_update = input(" Masukkan Sparepart : ")
17         is_not_match = True
18
19         # Mengecek apakah no sparepart sesuai
20         for index, item in enumerate(data["sparepart"]) :
21             if (input_user_update_no - 1) == index:
22
23                 # Mengupdate sparepart
24                 data["sparepart"][index] = input_user_update
25
26                 # Reassign variabel is_not_match dengan nilai boolean "False" karena no sparepart sesuai
27                 is_not_match = False
28                 break
29
30         # Kode di bawah ini akan dieksekusi apabila no sparepart tidak sesuai
31         if is_not_match :
32             console.print("\n ID Servis tidak ditemukan", style="red")
33             input(" Enter...")
34
35             console.print("\n Sukses Mengupdate!", style="green")
36             input(" Enter...")
37         # Kode di bawah ini akan dieksekusi apabila pengguna memasukkan karakter selain angka
38         except :
39             console.print("\n Harap masukkan angka", style="red")
40             input(" Enter...")

```

Gambar 3.2.14 Source Code Update Sparepart

Ini adalah source code function `update_sparepart()` yang akan dijalankan jika pengguna memasukkan “2” pada menu sparepart. Source code akan meminta pengguna memasukkan no sparepart. Kemudian, sistem akan mengecek apakah no yang diinputkan sesuai dengan data. Jika no yang diinputkan sesuai dengan data, maka sistem akan meminta pengguna memasukkan sparepart yang baru. Jika tidak sesuai, maka update sparepart gagal


```

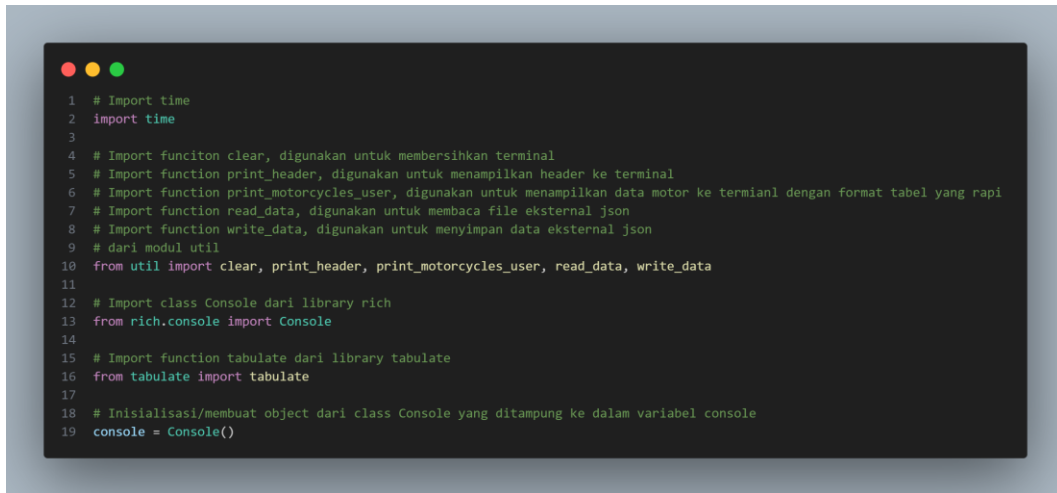
1 def delete_sparepart(data) :
2     try :
3         # Menampilkan header
4         print_header("Menu Delete Sparepart", 35)
5         print()
6
7         # Menampilkan sparepart motor
8         for index, item in enumerate(data["sparepart"]) :
9             print(f"{index + 1}. {item}")
10        print()
11
12        # Input no sparepart
13        input_user_delete_no = int(input(" Masukkan No Sparepart : "))
14        is_not_match = True
15
16        # Mengecek apakah no sparepart sesuai
17        for index, item in enumerate(data["sparepart"]) :
18            if index == (input_user_delete_no - 1) :
19                # Menghapus sparepart sesuai dengan no
20                del data["sparepart"][index]
21
22                # Reassign variabel is_not_match dengan nilai boolean "False" karena no sparepart sesuai
23                is_not_match = False
24                break
25
26        # Kode di bawah ini akan dieksekusi apabila no sparepart tidak sesuai
27        if is_not_match :
28            console.print("\n ID Servis tidak ditemukan", style="red")
29            input(" Enter...")
30
31            console.print("\n Sukses Menghapus!", style="green")
32            input(" Enter...")
33        # Kode di bawah ini akan dieksekusi apabila pengguna memasukkan karakter selain angka
34        except :
35            console.print("\n Harap masukkan angka", style="red")
36            input(" Enter...")

```

Gambar 3.2.15 Source Code Delete Sparepart

Ini adalah source code function delete_sparepart() yang akan dijalankan jika pengguna memasukkan “3” pada menu sparepart. Source code akan meminta pengguna memasukkan no sparepart. Kemudian, sistem akan mengecek apakah no yang diinputkan sesuai dengan data. Jika no yang diinputkan sesuai dengan data, maka sistem akan menghapus data sparepart sesuai dengan no sparepart. Jika tidak sesuai, maka delete sparepart gagal

3. Menu User



```
1 # Import time
2 import time
3
4 # Import function clear, digunakan untuk membersihkan terminal
5 # Import function print_header, digunakan untuk menampilkan header ke terminal
6 # Import function print_motorcycles_user, digunakan untuk menampilkan data motor ke terminal dengan format tabel yang rapi
7 # Import function read_data, digunakan untuk membaca file eksternal json
8 # Import function write_data, digunakan untuk menyimpan data eksternal json
9 # dari modul util
10 from util import clear, print_header, print_motorcycles_user, read_data, write_data
11
12 # Import class Console dari library rich
13 from rich.console import Console
14
15 # Import function tabulate dari library tabulate
16 from tabulate import tabulate
17
18 # Inisialisasi/membuat object dari class Console yang ditampilkan ke dalam variabel console
19 console = Console()
```

Gambar 3.2.16 Source Code Import Menu User

Gambar di atas adalah source code import beberapa module dan inisialisasi object. Module module dan object tersebut nantinya akan digunakan dalam source code menu_user.

```

1  def menu_user(name) :
2
3      # Menampilkan menu utama untuk pengguna.
4
5      # Proses:
6      # 1. Menampilkan daftar opsi menu:
7      #     - SERVICE MOTOR: Untuk mendaftarkan motor baru.
8      #     - UPDATE KELUHAN: Untuk memperbarui keluhan.
9      #     - HISTORI SERVIS: Untuk melihat riwayat servis selesai.
10     #     - EXIT: Untuk keluar dari akun.
11     # 2. Meminta pengguna memilih opsi.
12     # 3. Menjalankan fungsi berdasarkan pilihan pengguna.
13     # Args:
14     #     name (str): Nama pengguna yang sedang login.
15
16     while True :
17         # Membersihkan Terminal
18         clear()
19
20         # Menampilkan header
21         print_header("MENU USER", 110)
22         print()
23
24         # Menampilkan motor sesuai dengan kepemilikan pengguna
25         print_motorcycles_user(name)
26
27         # Menampilkan menu
28         print("\n SERVICE MOTOR          >> 1")
29         print(" UPDATE KELUHAN              >> 2")
30         print(" HISTORI SERVIS                  >> 3")
31         print(" EXIT                            >> 4")
32
33         # Input pilihan pengguna
34         input_user = input("\n Masukkan pilihan Anda : ")
35         if input_user == "1" :
36
37             # Masuk ke menu service motor
38             service_motor(name)
39
40         elif input_user == "2" :
41
42             # Masuk ke menu update keluhan
43             update_keluhan()
44
45         elif input_user == "3" :
46
47             # Masuk ke menu histori service
48             histori_service(name)
49
50         # Keluar dari akun
51         elif input_user == "4" :
52             console.print(" LOG OUT...", style="yellow")
53             time.sleep(2)
54             break
55
56         # Kode di bawah ini akan dieksekusi apabila input dari pengguna tidak sesuai dengan menu yang tersedia
57         else :
58             print()
59             console.print("\n Mohon maaf, menu tidak tersedia!!!", style="red")
60             input(" Enter.....")
61             clear()

```

Gambar 3.2.17 Source Code Menu User

Ini adalah source code function menu_user() yang akan dijalankan jika sistem telah memverifikasi bahwa pengguna adalah user. Di dalam function ini terdapat empat menu, diantaranya menu service motor, update keluhan, histori service. Di dalam

function ini pengguna diminta untuk menginputkan pilihan berdasarkan menu yang tersedia.

```
1 def service_motor(name) :
2     # Mendaftarkan motor untuk diservis.
3     # Proses:
4     # 1. Meminta pengguna memasukkan jenis motor dan keluhan.
5     # 2. Membaca ID servis terakhir dari file 'id_service.json' dan increment nilainya.
6     # 3. Menambahkan data baru ke file 'motorcycle.json' dengan informasi:
7     #     - ID servis.
8     #     - Jenis motor.
9     #     - Nama pemilik.
10    #     - Keluhan.
11    #     - Status awal servis.
12    #     - Keterangan awal.
13    # 4. Menyimpan data ID servis terbaru ke 'id_service.json'.
14    # Argumen:
15    #     name (str): Nama pengguna yang sedang login.
16
17    # Membersihkan Terminal
18    clear()
19
20    print()
21
22    # Menampilkan header
23    print_header("MENU SERVICE MOTOR", 50)
24
25    # Input Nama motor, keluhan/servis dan kilometer motor
26    input_motorcycle = input("\n Masukkan jenis motor Anda : ")
27    input_keluhan = input(" Masukkan keluhan/servis Anda : ")
28    input_kilometer = input(" Masukkan kilometer motor Anda : ")
29
30    # Membaca data ID dari file JSON
31    data_id = read_data("./databases/id_service.json")
32
33    # Membaca data motor dari file JSON
34    data_motorcycles = read_data("./databases/motorcycle.json")
35
36    # Menambahkan data motor baru
37    data_motorcycles.append(
38        {
39            "IDServis" : data_id["IDServis"],
40            "name_of_motorcycle" : input_motorcycle,
41            "owner" : name,
42            "servis" : input_keluhan,
43            "kilometer" : input_kilometer,
44            "sparepart" : [],
45            "status" : "...",
46            "display" : "on",
47            "keterangan" : "Mengecek ketersediaan barang/waktu"
48        }
49    )
50
51    # Mengubah data ID
52    data_id = {
53        "IDServis" : data_id["IDServis"] + 1
54    }
55
56    # Menyimpan data ke file eksternal JSON
57    write_data("./databases/id_service.json", data_id)
58    write_data("./databases/motorcycle.json", data_motorcycles)
59
60    console.print("\n Sukses Menambahkan!", style="green")
61    input(" Enter...")
```

Gambar 3.2.18 Source Code Service Motor

Ini adalah source code function `service_motor ()` yang akan dijalankan jika pengguna menginputkan "1" pada menu user. Source code akan meminta pengguna

menginputkan nama motor, kilometer motor dan keluhan/servis yang ingin dilakukan. Source ini berisi perintah untuk membaca file yang nanti dengannya data akan disimpan. Setelah berhasil membaca file, kemudian file tadi dimanipulasi dengan ditambahkan data baru (data yang diinputkan pengguna tadi). Langkah terakhir yang dilakukan source code ini adalah menyimpan data yang telah dimanipulasi tadi ke dalam sebuah file eksternal.

```

1  def update_keluhan() :
2      try :
3          print()
4
5          # Menampilkan header
6          print_header("MENU UPDATE KELUHAN", 50)
7
8          # Input ID service
9          input_user_ID = int(input(" Masukkan ID Service : "))
10
11         # Membaca data motor dari file JSON
12         data_motorcycles = read_data("./databases/motorcycle.json")
13
14         is_not_match = True
15
16         # Mengecek apakah ID service yang diinputkan pengguna sesuai
17         for item in data_motorcycles :
18             if item["IDservis"] == input_user_ID and item["display"] == "on":
19
20                 # Input keluhan baru
21                 input_user_update_keluhan = input(" Update Keluhan : ")
22
23                 # Reassign value dengan keluhan yang baru
24                 item["servis"] = input_user_update_keluhan
25
26                 # Menyimpan data ke file eksternal JSON
27                 write_data("./databases/motorcycle.json", data_motorcycles)
28                 console.print("\n Sukses Mengupdate!", style="green")
29                 input(" Enter....")
30
31                 # Reassign variabel is_wrong dengan nilai boolean "False" karena ID service sesuai
32                 is_not_match = False
33                 break
34
35         # Kode di bawah ini akan dieksekusi apabila ID service tidak sesuai
36         if is_not_match :
37             console.print("\n ID Servis tidak ditemukan", style="red")
38             input(" Enter....")
39         # Kode di bawah ini akan dieksekusi apabila pengguna menginputkan karakter selain angka
40         except :
41             console.print("\n Harap masukkan angka", style="red")
42             input(" Enter....")

```

Gambar 3.2.19 Source Code Update Keluhan

Ini adalah source code function `update_keluhan ()` yang akan dijalankan jika pengguna menginputkan "2". Dalam function ini pengguna diminta untuk menginputkan ID service. Source code ini akan melakukan pengecekan data yang diinputkan pengguna. Sistem akan membaca file data motor terlebih dahulu, kemudian melakukan pengecekan ID service dengan data motor. Jika ID servis dengan data sesuai, maka sistem akan meminta pengguna menginputkan keluhan baru. Jika tidak, maka update gagal.

```

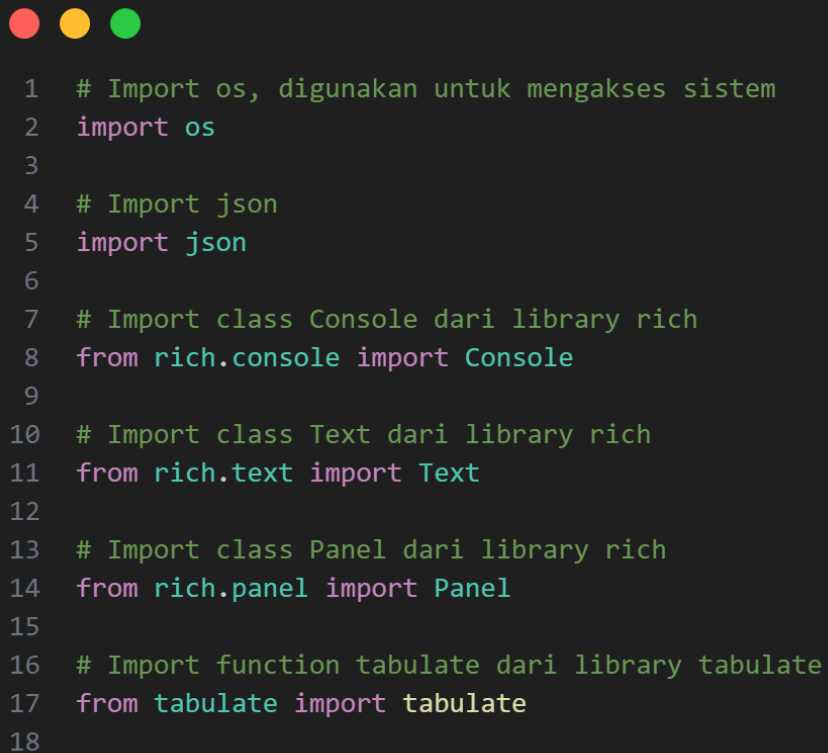
1  def histori_service(name) :
2      # Menampilkan histori servis motor yang telah selesai.
3
4      # Proses:
5      # 1. Membaca data dari file `motorcycle.json`.
6      # 2. Memfilter data berdasarkan:
7      #     - Status `display` yang bernilai `False` (servis selesai).
8      #     - Pemilik motor sesuai dengan nama pengguna.
9      # 3. Menampilkan hasil dengan format tabel.
10     # Args:
11     #     name (str): Nama pengguna yang sedang login.
12
13     # Membersihkan Terminal
14     clear()
15
16     # Menampilkan header
17     print_header("HISTORI SERVICE", 110)
18     print()
19
20
21     data_after_filter = {
22         "IDservis": [],
23         "name_of_motorcycle": [],
24         "owner": [],
25         "servis": [],
26         "kilometer": [],
27         "sparepart": [],
28         "status": [],
29         "keterangan": []
30     }
31     # Filter data motor yang telah diservis
32     for motorcycle in read_data("../databases/motorcycle.json") :
33
34         # Mengecek apakah data tersebut memiliki properti display bernilai "False" dan memiliki
35         # properti owner sesuai dengan nama pengguna
36         if motorcycle["display"] == "off" and motorcycle["owner"] == name:
37             data_after_filter["IDservis"].append(motorcycle["IDservis"])
38             data_after_filter["name_of_motorcycle"].append(motorcycle["name_of_motorcycle"])
39             data_after_filter["owner"].append(motorcycle["owner"])
40             data_after_filter["servis"].append(motorcycle["servis"])
41             data_after_filter["kilometer"].append(motorcycle["kilometer"])
42             data_after_filter["sparepart"].append(motorcycle["sparepart"])
43             data_after_filter["status"].append(motorcycle["status"])
44             data_after_filter["keterangan"].append(motorcycle["keterangan"])
45
46     print(tabulate(data_after_filter, headers="keys", tablefmt="fancy_grid"))
47     # Menampilkan histori servis
48
49     input(" Enter...")

```

Gambar 3.2.20 Source Code Histori Service

Ini adalah source code function `histori_service()` yang akan dijalankan jika pengguna menginputkan "3". Source code ini akan menampilkan histori servis dari pengguna jika pengguna sebelumnya pernah melakukan servis.

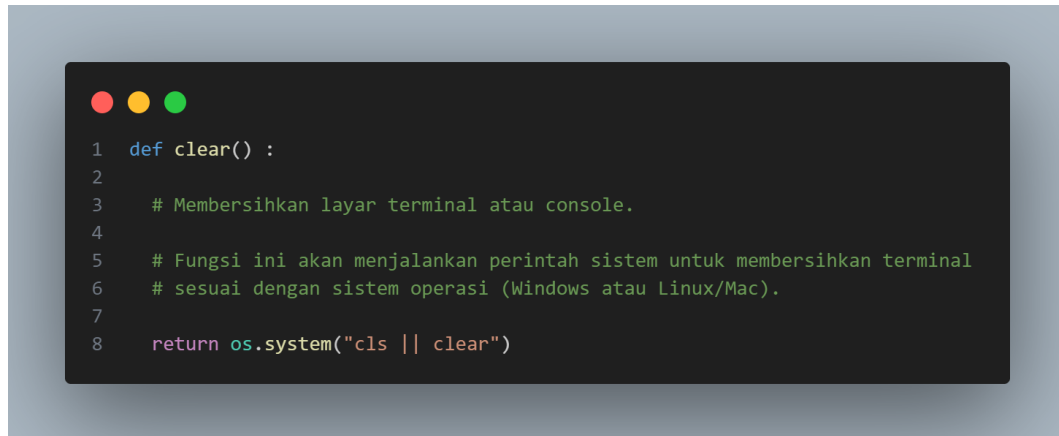
4. Util



```
1  # Import os, digunakan untuk mengakses sistem
2  import os
3
4  # Import json
5  import json
6
7  # Import class Console dari library rich
8  from rich.console import Console
9
10 # Import class Text dari library rich
11 from rich.text import Text
12
13 # Import class Panel dari library rich
14 from rich.panel import Panel
15
16 # Import function tabulate dari library tabulate
17 from tabulate import tabulate
18
```

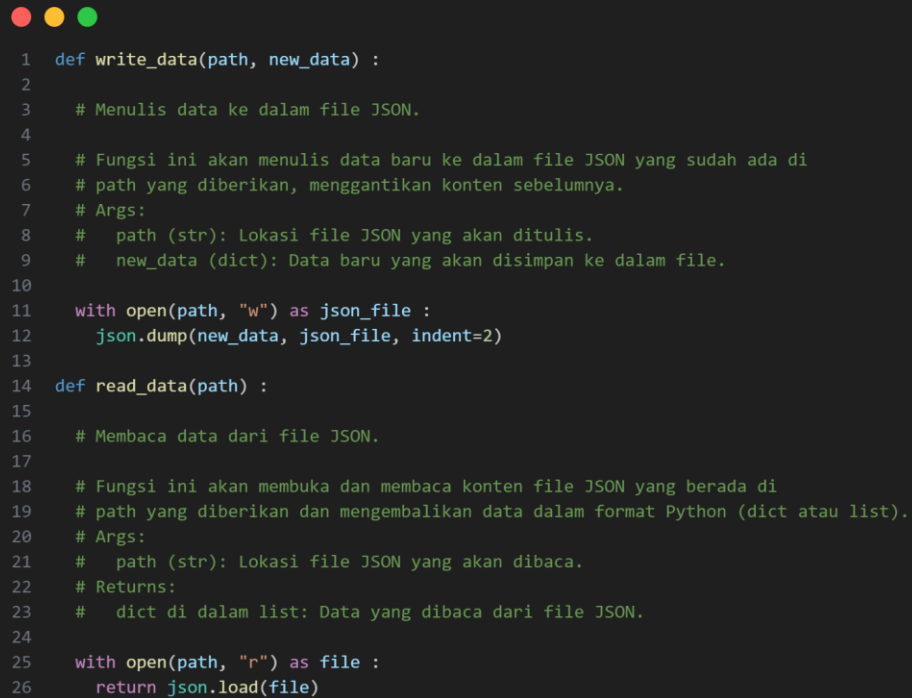
Gambar 3.2.21 Source Code Import Util

Gambar di atas adalah source code import beberapa module dan inisialisasi object. Module module dan object tersebut nantinya akan digunakan dalam source code util.



Gambar 3.2.22 Source Code Clear

Source code yang digunakan untuk mengakses sistme lalur memberisihkan terminal



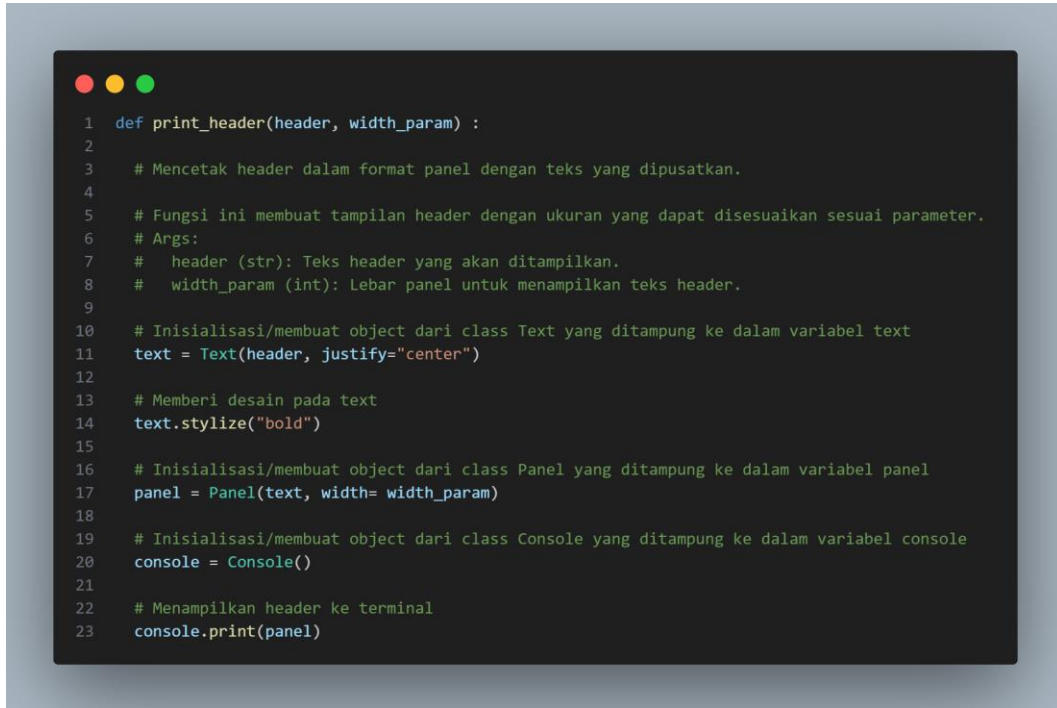
```

1  def write_data(path, new_data) :
2
3      # Menulis data ke dalam file JSON.
4
5      # Fungsi ini akan menulis data baru ke dalam file JSON yang sudah ada di
6      # path yang diberikan, menggantikan konten sebelumnya.
7      # Args:
8      #   path (str): Lokasi file JSON yang akan ditulis.
9      #   new_data (dict): Data baru yang akan disimpan ke dalam file.
10
11     with open(path, "w") as json_file :
12         json.dump(new_data, json_file, indent=2)
13
14     def read_data(path) :
15
16         # Membaca data dari file JSON.
17
18         # Fungsi ini akan membuka dan membaca konten file JSON yang berada di
19         # path yang diberikan dan mengembalikan data dalam format Python (dict atau list).
20         # Args:
21         #   path (str): Lokasi file JSON yang akan dibaca.
22         # Returns:
23         #   dict di dalam list: Data yang dibaca dari file JSON.
24
25         with open(path, "r") as file :
26             return json.load(file)

```

Gambar 3.2.23 Source Code Write Data & Read Data

Source code yang digunakan untuk membaca file eksternal JSON dan menyimpan eksternal JSON.



```
1 def print_header(header, width_param) :
2
3     # Mencetak header dalam format panel dengan teks yang dipusatkan.
4
5     # Fungsi ini membuat tampilan header dengan ukuran yang dapat disesuaikan sesuai parameter.
6     # Args:
7     #   header (str): Teks header yang akan ditampilkan.
8     #   width_param (int): Lebar panel untuk menampilkan teks header.
9
10    # Inisialisasi/membuat object dari class Text yang ditampung ke dalam variabel text
11    text = Text(header, justify="center")
12
13    # Memberi desain pada text
14    text.stylize("bold")
15
16    # Inisialisasi/membuat object dari class Panel yang ditampung ke dalam variabel panel
17    panel = Panel(text, width= width_param)
18
19    # Inisialisasi/membuat object dari class Console yang ditampung ke dalam variabel console
20    console = Console()
21
22    # Menampilkan header ke terminal
23    console.print(panel)
```

Gambar 3.2.24 Source Code Print Header

Source code yang digunakan untuk menampilkan tampilan header yang menarik di terminal

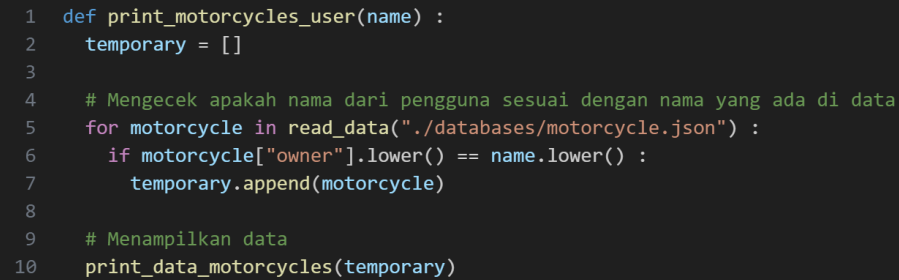
```

1 def print_data_motorcycles(data) :
2
3     # Mmembuat dictionary untuk menampung data
4     temp_data = {
5         "IDservis": [],
6         "name_of_motorcycle": [],
7         "owner": [],
8         "servis": [],
9         "kilometer" : [],
10        "sparepart" : [],
11        "status": [],
12        "keterangan": []
13    }
14
15    for motorcycle in data :
16        # Mengecek apakah property display bernilai "True"
17        if motorcycle["display"] == "on" :
18            temp_data["IDservis"].append(motorcycle["IDservis"])
19            temp_data["name_of_motorcycle"].append(motorcycle["name_of_motorcycle"])
20            temp_data["owner"].append(motorcycle["owner"])
21            temp_data["servis"].append(motorcycle["servis"])
22            temp_data["kilometer"].append(motorcycle["kilometer"])
23            temp_data["sparepart"].append(motorcycle["sparepart"])
24            temp_data["status"].append(motorcycle["status"])
25            temp_data["keterangan"].append(motorcycle["keterangan"])
26
27    # Menampilkan tabel
28    print(tabulate(temp_data, headers="keys", tablefmt="fancy_grid"))

```

Gambar 3.2.25 Source Code Print Data Motorcycles

Source code yang digunakan untuk menampilkan data motor dengan format table yang menarik di terminal.



```
1 def print_motorcycles_user(name) :  
2     temporary = []  
3  
4     # Mengecek apakah nama dari pengguna sesuai dengan nama yang ada di data  
5     for motorcycle in read_data("./databases/motorcycle.json") :  
6         if motorcycle["owner"].lower() == name.lower() :  
7             temporary.append(motorcycle)  
8  
9     # Menampilkan data  
10    print_data_motorcycles(temporary)
```

Gambar 3.2.26 Source Code Print Motorcycles User

Source code yang digunakan untuk menampilkan data motor dengan format table yang menarik di terminal. Di dalam source ini data akan difilter terlebih dahulu.

BAB IV

PENUTUP

4.1 Kesimpulan

Jumlah pengguna sepeda motor di Indonesia meningkat secara signifikan, sehingga menimbulkan kebutuhan yang besar akan layanan perawatan dan servis kendaraan yang efisien. Namun banyak bengkel yang masih mengandalkan cara manual dalam mengelola data layanan, sehingga seringkali menimbulkan berbagai permasalahan, antara lain: Kehilangan data, kesalahan pencatatan, dan kesulitan pelacakan riwayat layanan. Digitalisasi adalah solusi yang tepat untuk mengatasi tantangan ini.

Dengan menerapkan sistem manajemen servis kendaraan roda dua berbasis digital, Anda dapat mengelola data pelanggan, perencanaan servis, inventaris suku cadang, dan riwayat servis dengan lebih efisien dan terorganisir. Selain itu, sistem ini memberikan kemudahan akses informasi kepada pelanggan, sehingga meningkatkan produktivitas dan kualitas layanan bengkel.

Penerapan sistem ini diharapkan tidak hanya meningkatkan efisiensi operasional bengkel perawatan, tetapi juga meningkatkan kepuasan pelanggan, serta berdampak positif terhadap keberlangsungan operasional bengkel dan keselamatan sepeda motor. Oleh karena itu, sistem ini merupakan langkah penting menuju modernisasi layanan otomotif di era digital.

4.2 Saran

1. Peningkatan User Interface (UI) dan User Experience (UX)

Disarankan untuk meningkatkan antarmuka sistem agar lebih mudah dipahami dan ramah pengguna, baik untuk operator bengkel maupun pelanggan.

Tampilan yang intuitif akan membantu meningkatkan adopsi dan kepuasan pengguna.

2. Fokus pada Keamanan Data
Untuk menjaga privasi dan kepercayaan pelanggan, sistem harus memiliki fitur keamanan data yang kuat, seperti enkripsi data, otentikasi dua faktor, dan backup data otomatis.

DAFTAR PUSTAKA

Dr. Joseph Teguh Santoso, S. M. (2022). *PROYEK CODING dengan PYTHON*. Semarang: Yayasan Prima Agus Teknik bekerja sama dengan Universitas Sains & Teknologi Komputer (Universitas STEKOM).

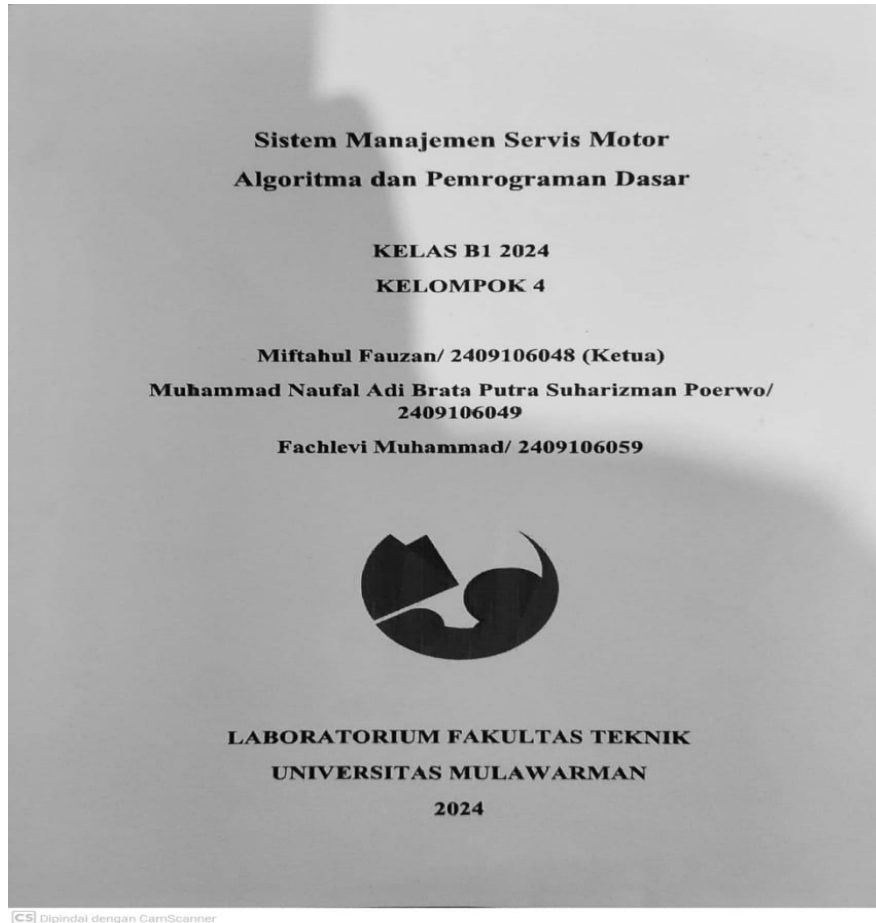
Dr. Joseph Teguh Santoso, S. M. (2023). *ILMU DATA (Data Science)*. Semarang: Yayasan Prima Agus Teknik bekerja sama dengan Universitas Sains & Teknologi Komputer (Universitas STEKOM).

Dwifa, M. B., Sasue, R. R., Dwipayana, A. D., Ahmad, R., Nitiyasa, G. B., Fitasari, Y., & Suraharta, I. M. (2021). *SERVIS SEPEDA MOTOR MURAH SEBAGAI UPAYA PENINGKATAN SKILLS TARUNA POLITEKNIK TRANSPORTASI DARAT BALI DI DESA BATUBULAN*. *Jurnal Pengabdian Kepada Masyarakat Semangat Nyata untuk*



Rahmat, D., A'zizah, S., & Mulyani, S. (2022). *PERKALIAN MENGGUNAKAN BAHASA PEMROGRAMAN PERULANGAN (LOOPING) BERBANTU SUBLIME TEXT DAN XAMPP*. *Djtechno: Journal of Information Technology Research Vol. 3, No. 2 Desember 2022*.

Romzi, M., & Kurniawan, B. (2022). *PEMBELAJARAN PEMROGRAMAN PYTHON DENGAN PENDEKATAN LOGIKA ALGORITMA*. *Jurnal Teknik Informatika Mahakarya (JTIM), Vol. 3, No. 2, Desember 2020, 37- 44*.



LAMPIRAN



Kartu Konsul

Tanggal Konsultasi : 1 November 2024	
Uraian / Pembahasan : 1. Memperbaiki Alur Flowchart 2. Menambah Fitur Delete untuk admin 3. Menambah Fitur Update untuk user	
Asisten Lab  Nama: Tito Darmawan	Ketua Kelompok  Nama: Miftahul Fauzan

Konsul Pertama

Tanggal Konsultasi : 14 November 2024	
Uraian / Pembahasan : 1. Mempercepat Visualisasi loading login 2. Menambah Fitur keterangan untuk admin.	
Asisten Lab  Nama: Tito Darmawan	Ketua Kelompok  Nama: Miftahul Fauzan

Konsul Kedua