

TUGAS PEMOGRAMAN BERBASIS OBJEK “IMPLEMENTASI PROGRAM PEMESANAN BENGKEL PYTHON”



Disusun oleh:

Muhammad Naufal Arhab – 5190411053

**PRODI INFORMATIKA
FAKULTAS SAINS & TEKNOLOGI
UNIVERSITAS TEKNOLOGI YOGYAKARTA
2024**

KATA PENGANTAR

Dalam era teknologi yang terus berkembang, pelayanan yang efisien dan mudah diakses menjadi kunci utama keberhasilan suatu bisnis. Program "Otofast" hadir sebagai solusi inovatif untuk mempermudah proses pemesanan layanan bengkel, memastikan bahwa pengalaman pelanggan dalam merawat kendaraan mereka menjadi lebih efisien dan menyenangkan.

Program ini dirancang dengan fokus pada kenyamanan pelanggan dalam melakukan pemesanan layanan untuk kendaraan mereka, baik itu motor maupun mobil. Dengan antarmuka yang sederhana dan fitur-fitur yang mudah digunakan, Otofast memberikan solusi yang handal dan efektif dalam mengelola antrian layanan bengkel serta menyediakan informasi yang dibutuhkan oleh pelanggan.

Dalam laporan ini, kami akan menyajikan detail implementasi, fitur utama, dan desain dari program Otofast. Kami juga akan menjelaskan struktur kelas dalam program ini melalui Class Diagram untuk memberikan gambaran yang lebih jelas tentang hubungan antar-komponen.

Semoga program Otofast dapat memberikan manfaat dan memberikan pengalaman yang memuaskan bagi pelanggan. Kami berharap masukan dan saran untuk terus meningkatkan kualitas layanan kami.

Yogyakarta, 11 Januari 2024

Penulis

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI.....	ii
DAFTAR GAMBAR	ii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Tujuan Pembuatan Program	1
1.3 Lingkup dan Batasan	1
1.4 Manfaat Program	1
1.5 Metode Pengembangan.....	1
1.6 Sistematika Laporan	2
BAB II DESKRIPSI PROGRAM	3
2.1 Konsep Dasar Program.....	3
2.2 Struktur Kode Program.....	3
2.3 Fungsi Utama Program.....	10
2.4 Penggunaan Inheritance.....	10
2.5 Penggunaan Modul Eksternal.....	10
2.6 Class Diagram.....	11
2.7 Contoh Penggunaan Program	12
KESIMPULAN	13

DAFTAR GAMBAR

Gambar 2. 1 Class Diagram	8
---------------------------------	---

BAB I PENDAHULUAN

1.1 Latar Belakang

Pertumbuhan industri otomotif dan bengkel telah menjadi bagian integral dalam mobilitas masyarakat modern. Dalam upaya memberikan layanan yang efisien dan berkualitas, pemanfaatan teknologi menjadi sebuah kebutuhan. Program pemesanan bengkel hadir sebagai solusi untuk mempermudah proses pendaftaran dan layanan kendaraan, meningkatkan pengalaman pelanggan, dan membantu pengelolaan data secara lebih terstruktur.

1.2 Tujuan Pembuatan Program

Program ini dirancang dengan tujuan utama untuk memberikan solusi dalam pengelolaan pemesanan layanan bengkel secara efektif dan efisien. Adapun tujuan khusus meliputi:

- Memfasilitasi proses pendaftaran pelanggan dan kendaraan.
- Menawarkan pilihan layanan yang sesuai dengan jenis kendaraan.
- Menciptakan struktur data yang terorganisir untuk memudahkan pengelolaan antrian bengkel.
- Memberikan informasi status kendaraan dan pembayaran kepada pelanggan.

1.3 Lingkup dan Batasan

Program ini difokuskan pada pemesanan layanan bengkel untuk kendaraan bermotor, baik motor maupun mobil. Namun, program ini tidak mencakup aspek-aspek tertentu seperti manajemen stok suku cadang dan mekanisme pembayaran elektronik. Pengembangan selanjutnya dapat mengakomodasi fitur-fitur tambahan sesuai kebutuhan.

1.4 Manfaat Program

Penerapan program ini diharapkan dapat memberikan manfaat yang signifikan, antara lain:

- Meningkatkan efisiensi pendaftaran pelanggan dan kendaraan.
- Meminimalisir kesalahan administrasi yang mungkin terjadi.
- Menyediakan informasi yang cepat dan akurat mengenai status kendaraan.
- Meningkatkan kepuasan pelanggan dengan layanan yang terintegrasi.

1.5 Metode Pengembangan

Program ini dikembangkan menggunakan bahasa pemrograman Python dengan konsep pemrograman berorientasi objek (OOP) untuk memastikan struktur yang terorganisir dan

mudah dipahami. Penggunaan metode pengembangan ini diharapkan dapat mendukung fleksibilitas dan pemeliharaan kode program.

1.6 Sistematika Laporan

Laporan ini terstruktur dalam beberapa bab, meliputi Pendahuluan, Deskripsi Program, Implementasi Kode, Uji Coba dan Evaluasi, Kesimpulan, Lampiran, dan Daftar Pustaka. Setiap bab memberikan informasi terperinci terkait aspek-aspek kunci dari pengembangan program.

BAB II DESKRIPSI PROGRAM

2.1 Konsep Dasar Program

Program pemesanan bengkel "Otofast" merupakan implementasi dari konsep pemrograman berorientasi objek (OOP) dengan menggunakan bahasa pemrograman Python. Konsep OOP digunakan untuk menciptakan struktur yang terorganisir, modular, dan mudah dipahami, dengan pembagian fungsi-fungsi utama ke dalam kelas-kelas yang mewakili objek-objek dalam domain bengkel.

2.2 Struktur Kode Program

Kode program terdiri dari beberapa kelas utama, yaitu:

- A. Kendaraan: Kelas induk yang mencakup atribut umum untuk motor dan mobil seperti merk, jenis, tipe, dan tahun pembuatan.

```
from datetime import date
import numpy
import sys
import os

class Kendaraan:
    def __init__(self, merk, jenis, tipe,
tahun_pembuatan):
        self.merk = merk
        self.jenis = jenis
        self.__tipe = tipe
        self.__tahun_pembuatan = tahun_pembuatan

    def get_tipe(self):
        return self.__tipe

    def get_tahun(self):
        return self.__tahun_pembuatan
```

- B. Motor dan Mobil: Kelas turunan dari kelas Kendaraan yang menambahkan atribut dan metode khusus untuk masing-masing jenis kendaraan.

```
class Motor(Kendaraan):
    def __init__(self, merk, tipe, tahun_pembuatan, tgl,
keluhan):
        super().__init__(merk, "Motor", tipe,
tahun_pembuatan)
        self.__tgl = tgl
        self.__keluhan = keluhan

    def get_tgl(self):
        return self.__tgl

    def get_keluhan(self):
        return self.__keluhan

    def layanan(self):
        print("=== Layanan Service Motor ===")
        print("1. Ganti Oli      Rp. 50.000")
        print("2. Service Rutin Rp. 150.000 ")
        print("3. Keluhan Lain  Rp. 35000 (Biaya Jasa)")
        print("0. Kembali")

class Mobil(Kendaraan):
    def __init__(self, merk, tipe, tahun_pembuatan, tgl,
keluhan):
        super().__init__(merk, "Mobil", tipe,
tahun_pembuatan)
        self.__tgl = tgl
        self.__keluhan = keluhan

    def get_tgl(self):
        return self.__tgl

    def get_keluhan(self):
        return self.__keluhan

    def layanan(self):
        print("=== Layanan Service Motor ===")
        print("1. Ganti Oli      Rp. 200.000")
        print("2. Service Rutin Rp. 500.000 ")
        print("3. Keluhan Lain  Rp. 100.000 (Biaya Jasa)")
        print("0. Kembali")
```

- C. Customer: Kelas yang menyimpan informasi pelanggan seperti nama, alamat, dan nomor handphone.

```
class Customer:
    def __init__(self, nama, alamat, nohp):
        self.nama = nama
        self.__alamat = alamat
        self.__nohnp = nohp

    def get_alamat(self):
        return self.__alamat

    def get_nohp(self):
        return self.__nohnp
```

- D. Bengkel: Kelas utama yang mengelola proses pemesanan, layanan, dan cek status kendaraan.

```
class Bengkel:
    def __init__(self, nama_bengkel):
        self.nama_bengkel = nama_bengkel
        self.antrian_motor = {}
        self.antrian_mobil = {}

    def print_motor(self):
        if len(self.antrian_motor) == 0:
            print("\nTidak Ada Antrian Motor.")

        else:
            print("Atas Nama : ",
self.antrian_motor['nama'])
            print("Tipe Kendaraan: ",
self.antrian_motor['tipe'])
            print("Tanggal Masuk : ",
self.antrian_motor['tanggal masuk'])
            print("Keluhan : ",
self.antrian_motor['keluhan'])
            print("Total Bayar : ", f"Rp.
{self.antrian_motor['bayar']}")

    def print_mobil(self):
        if len(self.antrian_mobil) == 0:
            print("\nTidak Ada Antrian Mobil.")

        else:
```



```

        print("Atas Nama : ",
self.antrian_mobil['nama'])
        print("Tipe Kendaraan: ",
self.antrian_mobil['tipe'])
        print("Tanggal Masuk : ",
self.antrian_mobil['tanggal masuk'])
        print("Keluhan : ",
self.antrian_mobil['keluhan'])
        print("Total Bayar : ", f"Rp.
{self.antrian_mobil['bayar']}")

```

```

def menu(self):
    while True:
        print("=== Bengkel Otofast ===")
        print("1. Daftar Service Kendaraan")
        print("2. Cek Kendaraan")
        print("0. Keluar")
        mn = int(input("Pilihan : "))

        if mn == 1:
            print("=== Masukan Data Diri dan Data
Kendaraan ===")
            nama = str(input("Nama : "))
            nama = nama.upper()
            alamat = str(input("Alamat : "))
            nohp = str(input("No Hp : "))
            print("="*10)
            jenis = int(input("Kendaraan (1 = Motor
| 2 = Mobil): "))
            merk = str(input("Merk Kendaraan : "))
            tipe = str(input("Tipe Kendaraan : "))
            tgl = date.today()
            thn = str(input("Tahun Pembuatan : "))

            if jenis == 1:
                a = 1
                bayar_motor = []
                while a == 1:
                    data = Customer(nama, alamat,
nohp)
                    motor = Motor(merk, tipe, thn,
tgl, "None")
                    motor.layanan()
                    layanan = int(input("Jasa yg di
inginkan : "))

                    if layanan == 1:
                        motor = motor = Motor(

```

```

                                merk, tipe, thn, tgl,
"Ganti Oli")
                                keluhan = "Ganti Oli"
                                bayar1 = 50000
                                ulang = str(input("Apaka Ada
Tambahan ?(y/n) : "))
                                if ulang == "y" or "Y":
                                    bayar_motor.append(bayar1)
                                    continue

                                elif ulang == "n" or "N":
                                    a = 0
                                    break

elif layanan == 2:
    motor = motor = Motor(
                                merk, tipe, thn, tgl,
"Ganti Oli")
                                keluhan = "Ganti Oli"
                                bayar1 = 50000
                                ulang = str(input("Apaka Ada
Tambahan ?(y/n) : "))
                                if ulang == "y" or "Y":
                                    bayar_motor.append(bayar1)
                                    continue

                                elif ulang == "n" or "N":
                                    a = 0
                                    break

elif layanan == 2:
    motor = motor = Motor(
                                merk, tipe, thn, tgl,
"Ganti Oli")
                                keluhan = "Service Rutin"
                                bayar2 = 150000
                                ulang = str(input("Apaka Ada
Tambahan ?(y/n) : "))
                                if ulang == "y" or "Y":
                                    bayar_motor.append(bayar2)
                                    continue

                                elif ulang == "n" or "N":
                                    a = 0
                                    break

elif layanan == 3:
    keluhan = str(input("Masukan
Keluhan"))
                                motor = motor = Motor(
                                merk, tipe, thn, tgl,
keluhan)
                                bayar3 = 35000
                                ulang = str(input("Apaka Ada
Tambahan ?(y/n) : "))
                                if ulang == "y" or "Y":
                                    bayar_motor.append(bayar3)
                                    continue

                                elif ulang == "n" or "N":
                                    a = 0
                                    break

elif layanan == 0:
    break

```

```

else:
    print("Input Salah! Silakan
Ulangi.")
    continue

total = numpy.sum(bayar_motor)
keys = ['nama', 'tipe', 'tanggal
masuk', 'keluhan', 'bayar']
values = [nama, tipe, tgl, keluhan,
total]

self.antrian_motor = dict(zip(keys,
values))

bayar_motor.clear()
os.system('cls')

if jenis == 2:
    a = 1
    while a == 1:
        data = Customer(nama, alamat,
nohp)

        motor = Mobil(merk, tipe, thn,
tgl, "None")

        motor.layanan()
        layanan = int(input("Jasa yg di
inginkan : "))

        bayar_mobil = []
        if layanan == 1:
            motor = motor = Mobil(
                merk, tipe, thn, tgl,
"Ganti Oli")

            keluhan = "Ganti Oli"
            bayar1 = 200000
            ulang = str(input("Apakah Ada
Tambahan ?(y/n)"))

            if ulang == "y" or "Y":
                bayar_mobil.append(bayar1)
                continue

            elif ulang == "n" or "N":
                a = a + 1
                break

        elif layanan == 2:
            motor = motor = Mobil(
                merk, tipe, thn, tgl,
"Service Rutin")

            keluhan = "Service Rutin"
            bayar2 = 500000

```

```

        ulang = str(input("Apakah Ada
Tambahan ?(y/n)"))

        if ulang == "y" or "Y":
            bayar_mobil.append(bayar2)
            continue
        elif ulang == "n" or "N":
            a = a + 1
            break

    elif layanan == 3:
        keluhan = str(input("Masukan
Keluhan"))

        motor = motor = Mobil(
            merk, tipe, thn, tgl,
            keluhan)

        bayar3 = 100000
        ulang = str(input("Apakah Ada
Tambahan ?(y/n)"))

        if ulang == "y" or "Y":
            bayar_mobil.append(bayar3)
            continue
        elif ulang == "n" or "N":
            a = a + 1
            break

    elif layanan == 0:
        break

    else:
        print("Input Salah! Silakan
Ulangi.")
        continue

    total = numpy.sum(bayar_mobil)
    keys = ['nama', 'tipe', 'tanggal
masuk', 'keluhan', 'bayar']
    values = [nama, tipe, tgl, keluhan,
total]

    self.antrian_mobil = dict(zip(keys,
values))

    bayar_mobil.clear()
    os.system('cls')

elif mn == 2:
    print("=== Cek Kendaraan")
    nama = str(input("Atas Nama : "))
    nama = nama.upper()

```

```

        jenis = int(input("Kendaraan (1 = Motor
| 2 = Mobil): "))
        if jenis == 1 and nama ==
self.antrian_motor['nama']:
            self.print_motor()

            elif jenis == 2 and nama ==
self.antrian_mobil['nama']:
            self.print_mobil()

        elif mn == 0:
            sys.exit()

bengkel = Bengkel("Otofast")
bengkel.menu()

```

2.3 Fungsi Utama Program

Program ini memiliki beberapa fungsi utama, di antaranya:

- **Input Data Pelanggan dan Kendaraan:** Program memungkinkan pengguna untuk memasukkan informasi pelanggan dan kendaraan secara interaktif.
- **Pilihan Layanan:** Pengguna dapat memilih layanan yang diinginkan sesuai dengan jenis kendaraan (motor/mobil) dan jenis layanan yang tersedia.
- **Perhitungan Biaya:** Program menghitung total biaya layanan berdasarkan pilihan yang dibuat oleh pengguna.
- **Cek Status Kendaraan:** Pelanggan dapat memeriksa status dan biaya layanan kendaraan yang sedang diproses di bengkel.

2.4 Penggunaan Inheritance

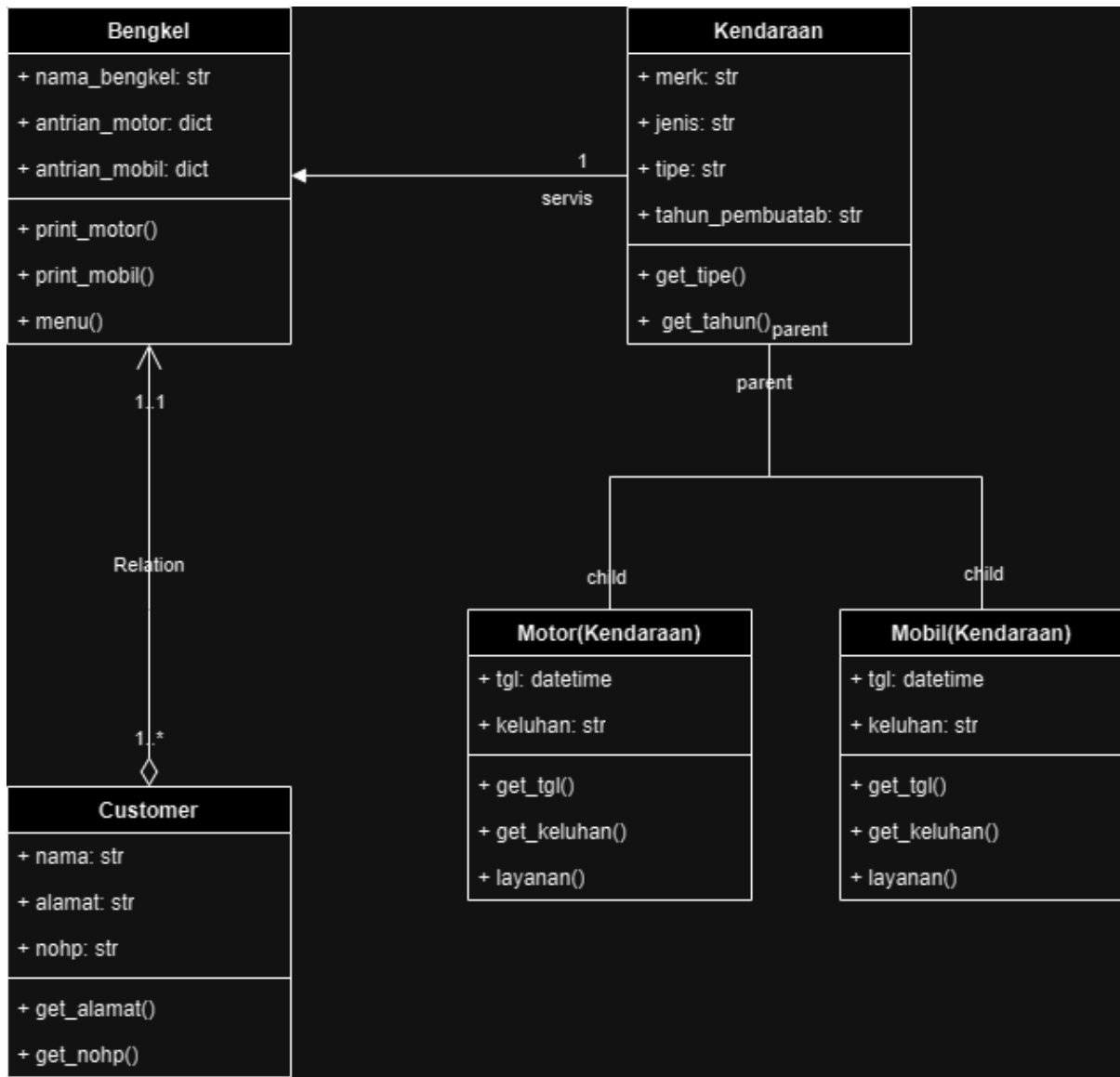
Konsep inheritance digunakan dalam pembuatan kelas Motor dan Mobil yang merupakan turunan dari kelas Kendaraan. Hal ini memungkinkan penggunaan atribut dan metode umum dari kelas Kendaraan, sementara tetap memperbolehkan penambahan fitur khusus pada kelas turunannya.

2.5 Penggunaan Modul Eksternal

Program menggunakan modul eksternal seperti 'numpy' untuk melakukan perhitungan total biaya layanan secara efisien. Penggunaan modul eksternal ini membantu dalam mengoptimalkan kinerja dan fungsionalitas program.

2.6 Class Diagram

Untuk memberikan gambaran visual yang lebih jelas mengenai struktur kelas dan hubungan antar kelas dalam program ini, kami menyertakan Class Diagram sebagai berikut:



Gambar 2. 1 Class Diagram

Pada Class Diagram di atas, dapat dilihat bahwa:

- Kelas Motor dan Mobil merupakan turunan dari kelas Kendaraan, mewarisi atribut dan metode dari kelas induk tersebut.
- Kelas Customer memiliki atribut rahasia (private) untuk alamat dan nomor handphone, dan menyediakan metode untuk mendapatkan informasi tersebut.

- Kelas Bengkel memiliki metode untuk mencetak informasi mengenai antrian kendaraan baik motor maupun mobil, serta metode menu untuk mengelola interaksi dengan pengguna.

2.7 Contoh Penggunaan Program

Program memandu pengguna melalui langkah-langkah pendaftaran, pemilihan layanan, dan cek status kendaraan. Pengguna dapat memasukkan data, memilih layanan yang diinginkan, dan mendapatkan informasi mengenai status kendaraan mereka.

1. Running Program

```
=== Bengkel Otofast ===
1. Daftar Service Kendaraan
2. Cek Kendaraan
0. Keluar
Pilihan : █
```

2. Mendaftarkan Kendaraan

```
=== Bengkel Otofast ===
1. Daftar Service Kendaraan
2. Cek Kendaraan
0. Keluar
Pilihan : 1
=== Masukkan Data Diri dan Data Kendaraan ===
Nama      : Naufal Arhab
Alamat    : Yogyakarta
No Hp     : 5190411053
=====
Kendaraan (1 = Motor | 2 = Mobil): 1
Merk Kendaraan : Kawasaki
Tipe Kendaraan : ZX25r
Tahun Pembuatan : 2020
=== Layanan Service Motor ===
1. Ganti Oli      Rp. 50.000
2. Service Rutin  Rp. 150.000
3. Keluhan Lain   Rp. 35000 (Biaya Jasa)
0. Kembali
Jasa yg di inginkan : 1
```

3. Cek Data Kendaraan

```
=== Cek Kendaraan
Atas Nama : naufal arhab
Kendaraan (1 = Motor | 2 = Mobil): 1
Atas Nama : NAUFAL ARHAB
Tipe Kendaraan: ZX25r
Tanggal Masuk : 2024-01-11
Keluhan      : Ganti Oli
Total Bayar  : Rp. 50000
```

KESIMPULAN

Dalam implementasi program pemesanan bengkel "Otofast," berhasil mencapai tujuan utama dengan sukses. Program ini, yang dikembangkan menggunakan bahasa pemrograman Python dan menerapkan prinsip pemrograman berorientasi objek (OOP), menyajikan solusi yang efektif untuk mempermudah proses pendaftaran dan pelayanan kendaraan di bengkel. Dalam pencapaian tujuan tersebut, beberapa hal dapat disoroti. Pertama, antarmuka program dirancang agar intuitif bagi pengguna, memastikan pengalaman pemesanan yang menyenangkan dan efisien. Kedua, penerapan konsep OOP memberikan struktur kode yang terorganisir, modular, dan mudah dipahami, mendukung pemeliharaan dan pengembangan lebih lanjut. Terakhir, penyediaan informasi status kendaraan dan perhitungan biaya layanan secara cepat dan akurat menjadi fokus utama yang terpenuhi.

Namun, selama proses pengembangan, berbagai tantangan dihadapi, termasuk manajemen antrian dan pemahaman mendalam terkait konsep OOP. Tantangan tersebut, meskipun memerlukan upaya dan penyesuaian, akhirnya diatasi, memberikan pengalaman berharga bagi tim pengembang. Pembelajaran dari setiap tantangan tersebut dijadikan bekal untuk peningkatan lebih lanjut.

Sebagai saran dan rekomendasi untuk pengembangan berikutnya, pertimbangan melibatkan penambahan fitur manajemen stok suku cadang guna mendukung kelengkapan operasional bengkel. Selain itu, penyempurnaan antarmuka pengguna untuk memberikan pengalaman yang lebih intuitif dan responsif dapat menjadi fokus perbaikan.

Program ini diharapkan memberikan dampak positif dalam meningkatkan efisiensi operasional bengkel dan pengalaman pelanggan. Penerapan layanan yang terintegrasi dan efisien dapat mengoptimalkan proses kerja bengkel secara menyeluruh. Dengan tekad untuk terus meningkatkan dan mengembangkan program ini, tim pengembang mengucapkan terima kasih kepada semua pihak yang terlibat dalam perjalanan ini.

Demikianlah, program pemesanan bengkel "Otofast" bukan hanya menjadi solusi praktis untuk kebutuhan bengkel, tetapi juga merupakan implementasi konkret dari konsep OOP dalam aplikasi dunia nyata. Melalui perjalanan ini, diharapkan program ini dapat terus berkembang dan memberikan kontribusi positif dalam dunia layanan bengkel.