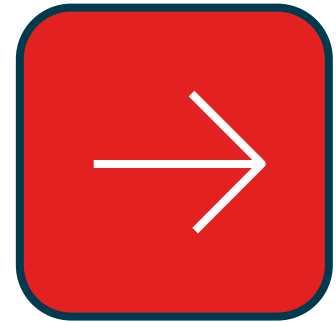


PORTAL PARKIR OTOMATIS: 'GateX'

Berpikir Komputasional

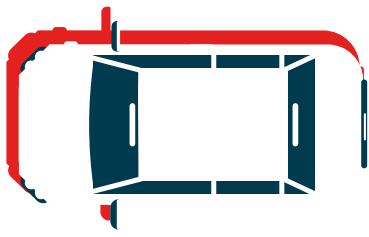
Kelompok 6 - WI1102 K31



NAMA ANGGOTA KELOMPOK



NIM	NAMA LENGKAP
19624219	Muhammad Azikra Wira Pratama
19624236	Zahran Alvan Putra Winarko
19624251	Aurelia Jennifer Gunawan
19624265	Nabilla Eka Putri Sunarto
19624286	Fayyaz Akmal Lauda

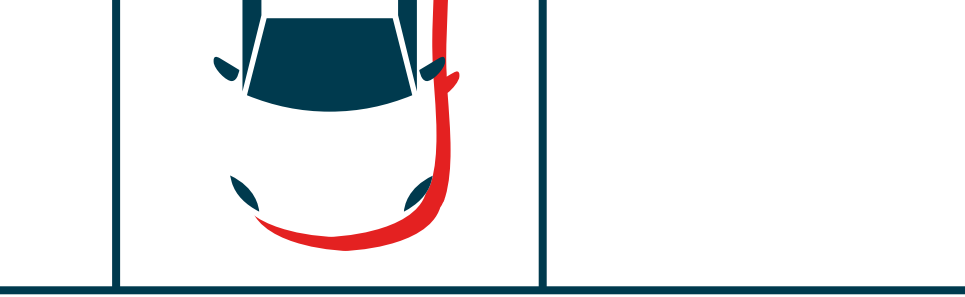


01

Identifikasi Masalah

Identifikasi masalah di lingkungan sekitar, khususnya kampus ITB Jatinangor, yang berkaitan dengan kegiatan parkir



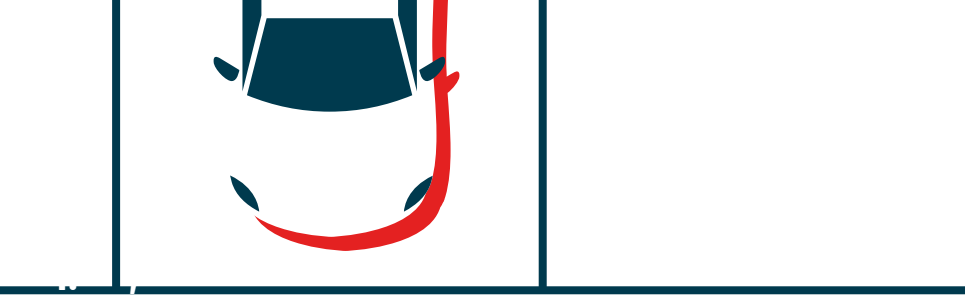


Setelah melakukan riset ke lingkungan sekitar, khususnya lingkungan kampus ITB Jatinangor, kami berhasil mengidentifikasi masalah yang berkaitan dengan sistem parkir, yaitu sebagai berikut:

Letak	Parkir	yang	Berantakan
Area parkir yang tidak tertata dengan baik menyebabkan penggunaan ruang yang tidak efisien dan banyak kendaraan yang diparkir dengan sembarangan. Selain itu, kurangnya petunjuk arah memperumit pencarian tempat parkir kosong.			

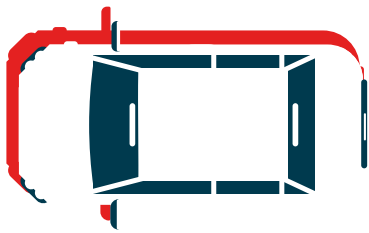
Ketidaktahuan Durasi & Letak Parkir

Kurangnya sistem yang menginformasikan durasi parkir dan lokasi kendaraan secara real-time membuat pengguna kesulitan mengatur waktu dan mencari kendaraan mereka dengan efisien.



3. Kurangnya Informasi Ketersediaan Tempat Parkir
Kurangnya sistem informasi real-time tentang ketersediaan tempat parkir dan kapasitas lokasi membuat pengguna harus berkeliling mencari tempat kosong, membuang waktu, dan menimbulkan frustrasi.

4. Proses Masuk dan Keluar yang Manual
Proses registrasi dan pembayaran parkir yang manual menyebabkan antrian panjang, memperlambat arus kendaraan, dan membuat pengelolaan parkir tidak efektif, terutama saat jam sibuk.



02

SPESIFIKASI PROGRAM

Spesifikasi Program Portal Parkir Otomatis: 'GateX'





SPESIFIKASI PROGRAM



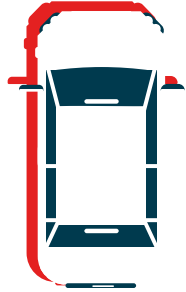
- Mengklasifikasikan tipe kendaraan sesuai untuk dimasukkan ke tempat parkir yang sesuai dan tersedia
- Menolak kendaraan masuk ketika tidak ada slot parkir
- Menampilkan status tiap slot parkir yang kosong dan terisi
- Mencatat waktu awal masuk parkir
- Menentukan biaya parkir berdasarkan durasi dan tipe kendaraan



DEKOMPOSISI

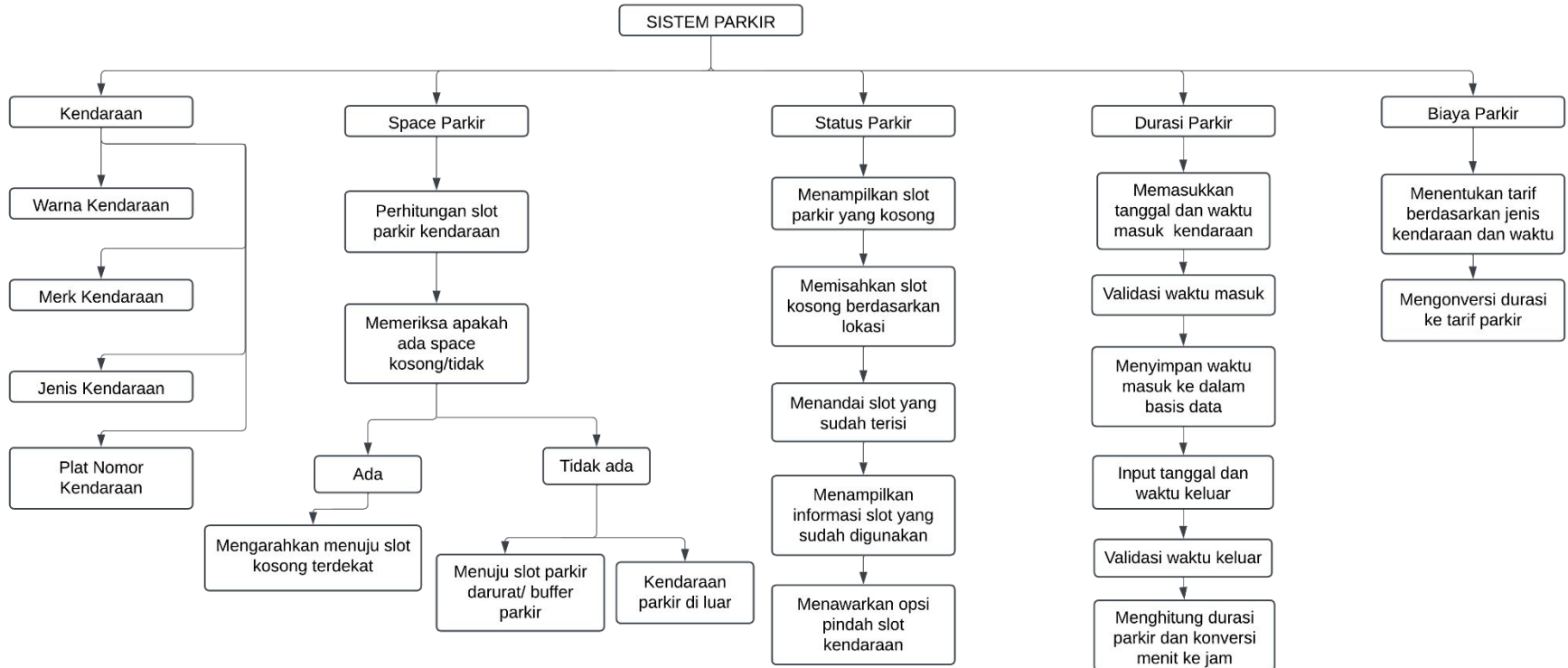
Dekomposisi Program Portal Parkir Otomatis: 'GateX'

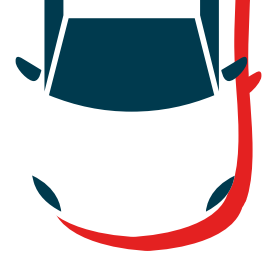
03





DEKOMPOSISI



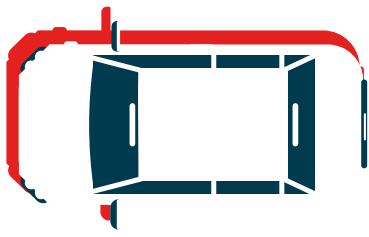


Pengenalan Pola

Pengenalan Pola Program Portal Parkir Otomatis: 'GateX'

04





Pengenalan Pola

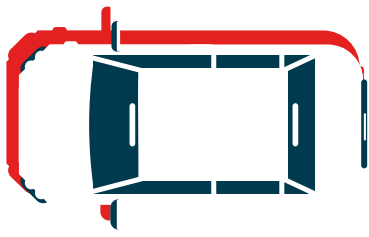


1. Pola Plat Nomor

Plat nomor terdiri atas 3 bagian utama, yaitu:

- kode huruf depan, yang terdiri atas 1-2 huruf, pada plat nomor kendaraan menjelaskan asal,
- kode untuk kendaraan penumpang terdiri atas 1-4 digit angka
- kode huruf belakang, yang terdiri atas 1-3 huruf, menerangkan lokasi kendaraan terdaftar





Pengenalan Pola

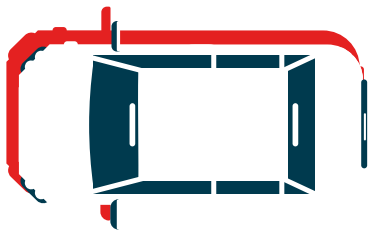


2. Pola Biaya Parkir

Biaya parkir ditentukan berdasarkan durasi parkir dan jenis kendaraan, seperti:

- a. Pola biaya parkir mobil adalah:
 - i. 1 jam pertama -> Rp 3000,00
 - ii. jam berikutnya-> Rp 1000,00/jam dengan maks. Rp 10000,00
- b. Pola biaya parkir motor
 - i. 1 jam pertama -> Rp 1000,00
 - ii. jam berikutnya -> Rp 500,00/jam dengan maks, Rp 5000,00



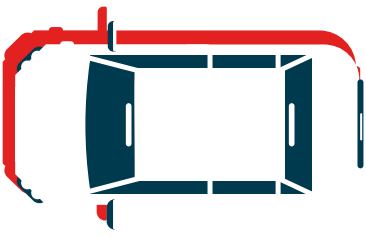


Pengenalan Pola



3. Pola Optimasi Ruang Parkir
 - a. Ruang parkir kendaraan terbagi berdasarkan jenis kendaraan,
 - b. Setiap kendaraan diparkirkan sesuai urutan ruang parkir yang tersedia untuk mengoptimalisasi ruang parkir yang ada.





05

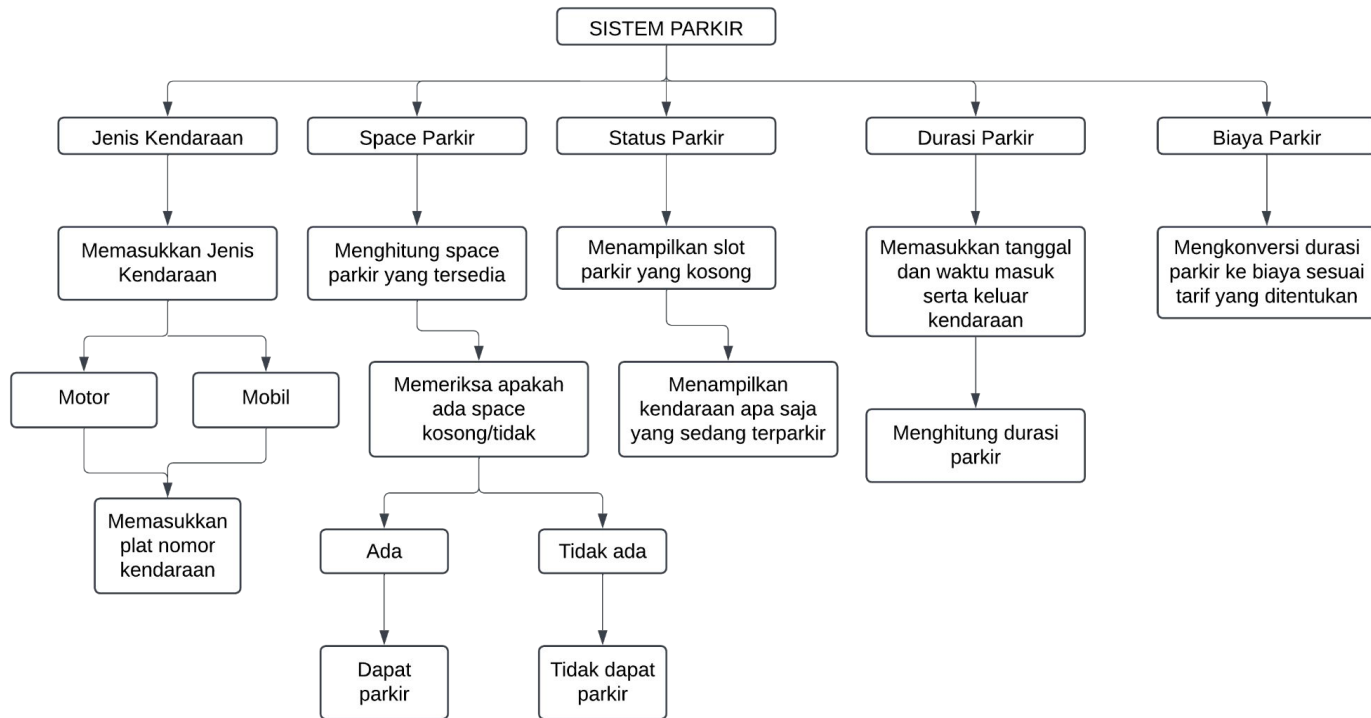
ABSTRAKSI

Abstraksi Program Portal Parkir Otomatis: 'GateX'





ABSTRAKSI



Flowchart

05

Flowchart Program Portal Parkir Otomatis: 'GateX'

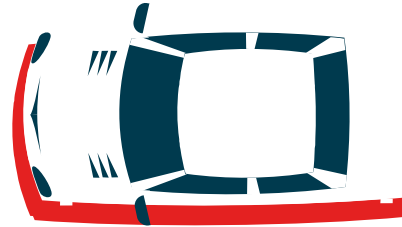
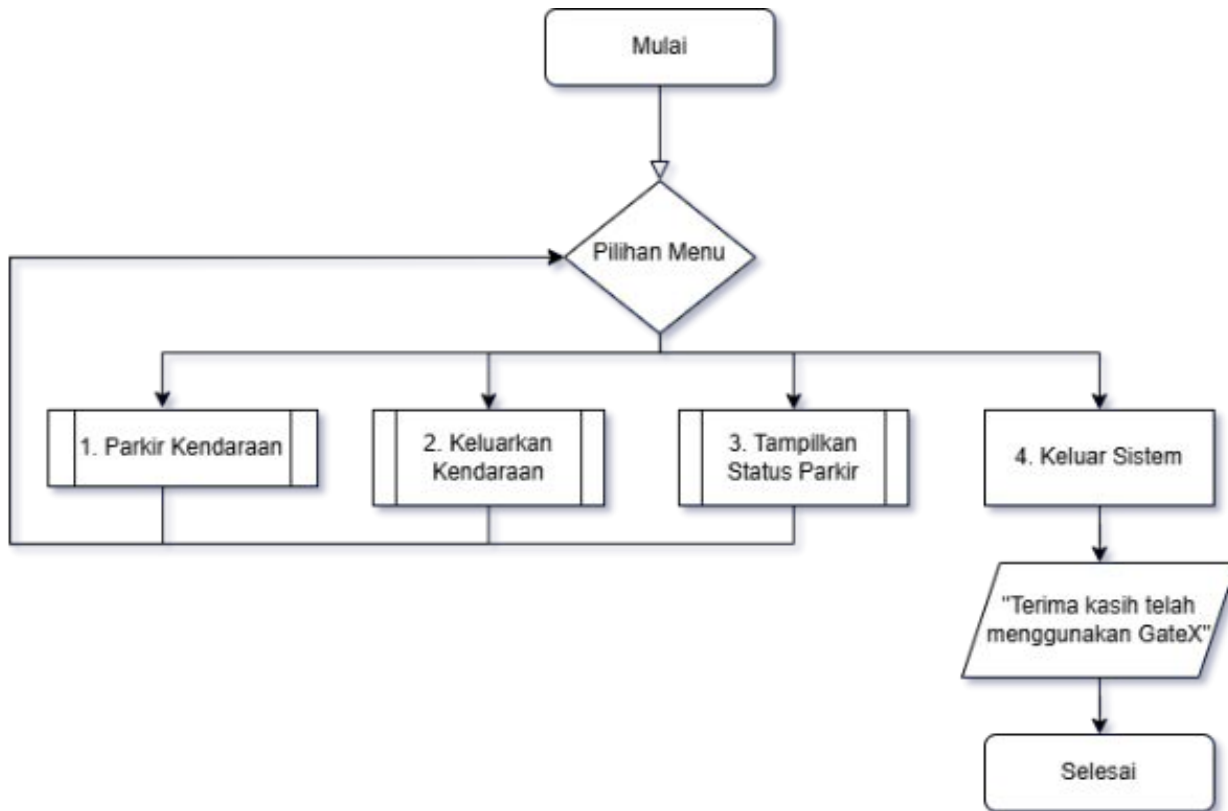


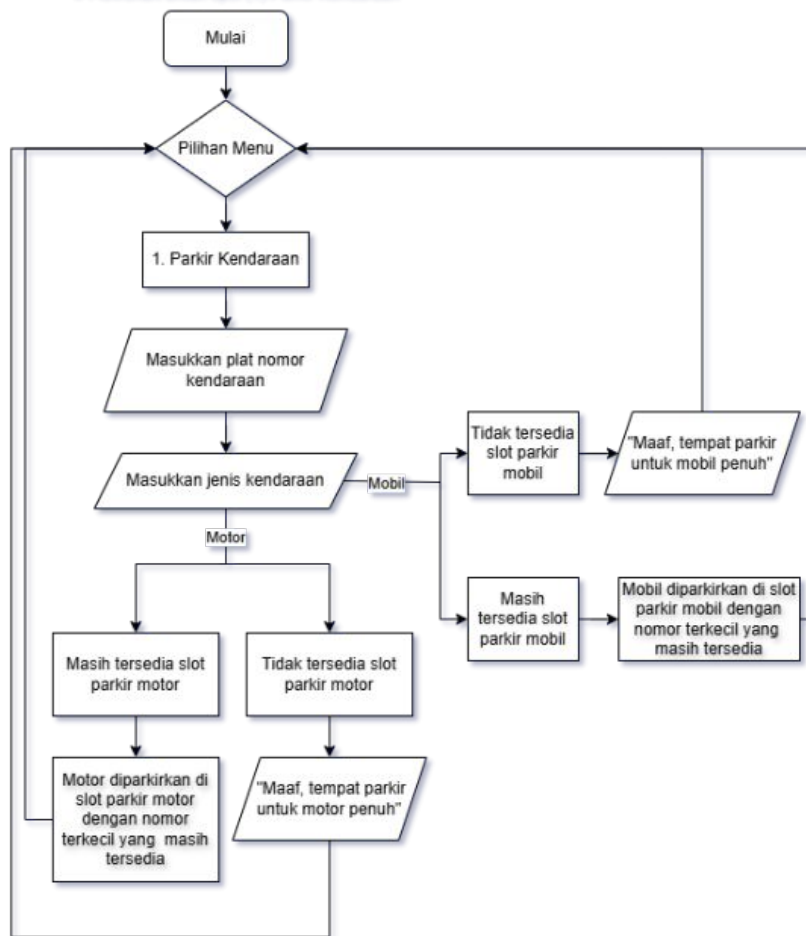


Diagram Alir/ Flowchart



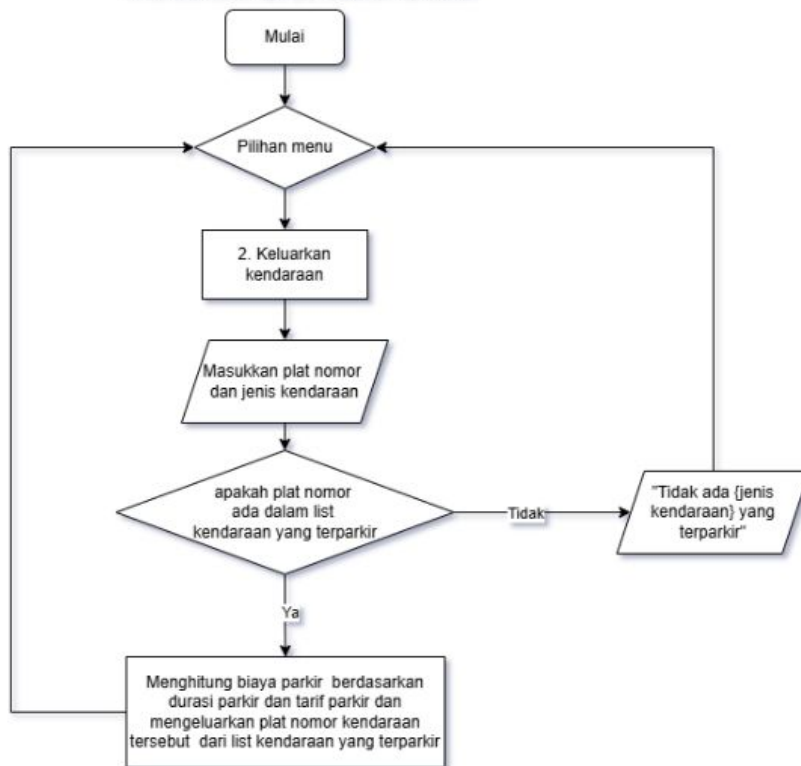


Flowchart untuk opsi (1) Parkir Kendaraan



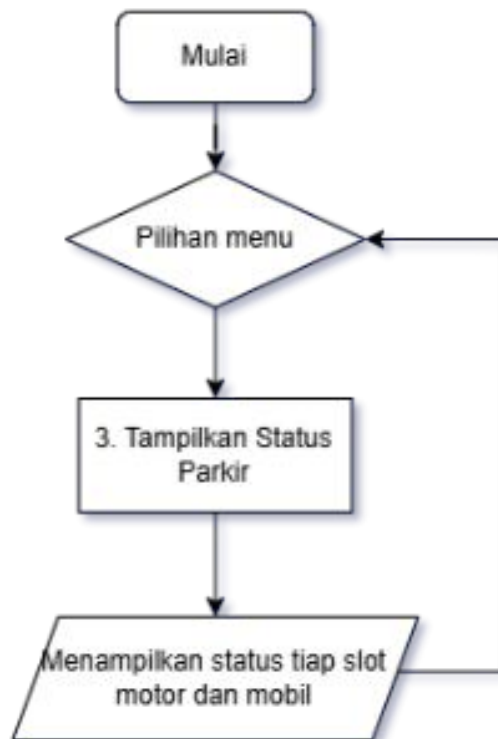


#Flowchart untuk opsi (2) Keluarkan Kendaraan





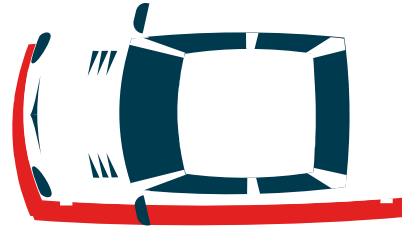
#Flowchart untuk opsi (3) Tampilkan Status Parkir



Program

06

Penjelasan program Portal Parkir Otomatis: 'GateX'





ALGORITMA: KAMUS

```
# KAMUS
#kapasitas_mobil      :int          ; kapasitas total tempat parkir untuk mobil
#kapasitas_motor      :int          ; kapasitas total tempat parkir untuk motor
#mobil_terparkir      :dict         ; dictionary yang menyimpan data mobil yang sedang terparkir,
#                               termasuk informasi nomor plat, slot parkir, dan waktu masuk
#motor_terparkir      :dict         ; dictionary yang menyimpan data motor yang sedang terparkir,
#                               termasuk informasi nomor plat, slot parkir, dan waktu masuk
#slot_tersedia_mobil  :list(int)    ; daftar yang berisi nomor slot parkir yang masih tersedia untuk mobil
#slot_tersedia_motor  :list(int)    ; daftar yang berisi nomor slot parkir yang masih tersedia untuk motor
#plat_nomor           :str          ; nomor plat kendaraan yang akan diparkirkan atau dikeluarkan
#jenis_kendaraan      :str          ; jenis kendaraan (mobil/motor)
#pilihan              :str          ; input dari pengguna yang menentukan aksi dalam sistem (1/2/3/4)
#slot                 :int          ; nomor slot parkir yang diberikan kepada kendaraan
#waktu_masuk          :datetime     ; waktu saat kendaraan masuk ke slot parkir
#waktu_keluar         :datetime     ; waktu saat kendaraan keluar dari slot parkir
#durasi_detik         :float        ; durasi parkir dalam satuan detik
#durasi_jam           :float        ; durasi parkir dalam satuan jam
#info_mobil           :dict         ; dictionary yang menyimpan informasi mobil yang akan dikeluarkan
#info_motor           :dict         ; dictionary yang menyimpan informasi motor yang akan dikeluarkan
#biaya                :float        ; biaya parkir kendaraan berdasarkan durasi parkir dan tarif parkir
#                               yang telah ditentukan
```



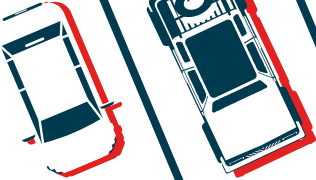
ALGORITMA

```
#ALGORITMA
from datetime import datetime, timedelta
import time

# Konfigurasi sistem parkir
kapasitas_mobil = int(input("Masukkan kapasitas tempat parkir untuk mobil: "))
kapasitas_motor = int(input("Masukkan kapasitas tempat parkir untuk motor: "))

# Variabel penyimpanan data parkir
mobil_terparkir = {}
motor_terparkir = {}
slot_tersedia_mobil = list(range(1, kapasitas_mobil + 1))
slot_tersedia_motor = list(range(1, kapasitas_motor + 1))

# Fungsi utama
while True:
    print("\n=== Sistem Parkir Otomatis ===")
    print("1. Parkir kendaraan")
    print("2. Keluarkan kendaraan")
    print("3. Tampilkan status parkir")
    print("4. Keluar Sistem")
    pilihan = input("Pilih opsi (1-4): ")
```



ALGORITMA: 1. PARKIR KENDARAAN

```
if pilihan == '1': # Parkir kendaraan
    plat_nomor = input("Masukkan nomor plat kendaraan: ")
    jenis_kendaraan = input("Masukkan jenis kendaraan (mobil/motor): ").lower()

    if jenis_kendaraan == 'mobil':
        if len(slot_tersedia_mobil) > 0:
            slot = slot_tersedia_mobil.pop(0)
            waktu_masuk = datetime.now()
            mobil_terparkir[plat_nomor] = {'slot': slot, 'waktu_masuk': waktu_masuk}
            print(f"Mobil dengan plat {plat_nomor} diparkir di slot {slot} pada {waktu_masuk}.")
        else:
            print("Maaf, tempat parkir untuk mobil penuh.")

    elif jenis_kendaraan == 'motor':
        if len(slot_tersedia_motor) > 0:
            slot = slot_tersedia_motor.pop(0)
            waktu_masuk = datetime.now()
            motor_terparkir[plat_nomor] = {'slot': slot, 'waktu_masuk': waktu_masuk}
            print(f"Motor dengan plat {plat_nomor} diparkir di slot {slot} pada {waktu_masuk}.")
        else:
            print("Maaf, tempat parkir untuk motor penuh.")

    else:
        print("Jenis kendaraan tidak dikenal. Silakan pilih 'mobil' atau 'motor'.")
```




ALGORITMA: 2. KELUARKAN KENDARAAN



```
elif pilihan == '2': # Keluarkan kendaraan
    plat_nomor = input("Masukkan nomor plat kendaraan: ")
    jenis_kendaraan = input("Masukkan jenis kendaraan (mobil/motor): ").lower()

    if jenis_kendaraan == 'mobil' and plat_nomor in mobil_terparkir:
        info_mobil = mobil_terparkir.pop(plat_nomor)
        slot = info_mobil['slot']
        waktu_masuk = info_mobil['waktu_masuk']
        waktu_keluar = datetime.now()

        # Hitung durasi parkir
        durasi_detik = (waktu_keluar - waktu_masuk).total_seconds()
        durasi_jam = durasi_detik / 4 # 4 detik = 1 jam di sistem
        if durasi_jam >= 0 and durasi_jam <= 1:
            biaya = 3000
        elif durasi_jam > 1 and durasi_jam <= 8:
            biaya = 3000 + (durasi_jam - 1) * 1000
        else:
            biaya = 10000

        # Kembalikan slot parkir
        slot_tersedia_mobil.append(slot)
        slot_tersedia_mobil.sort()

    print(f"Mobil dengan plat {plat_nomor} keluar dari slot {slot}.")
    print(f"Waktu masuk: {waktu_masuk}, Waktu keluar: {waktu_keluar}")
    print(f"Durasi parkir: {durasi_jam:.2f} jam")
    print(f"Biaya parkir: Rp{biaya:.2f}")
```



ALGORITMA: 2. KELUARKAN KENDARAAN



```
elif jenis_kendaraan == 'motor' and plat_nomor in motor_terparkir:
    info_motor = motor_terparkir.pop(plat_nomor)
    slot = info_motor['slot']
    waktu_masuk = info_motor['waktu_masuk']
    waktu_keluar = datetime.now()

    # Hitung durasi parkir
    durasi_detik = (waktu_keluar - waktu_masuk).total_seconds()
    durasi_jam = durasi_detik / 4 # 4 detik = 1 jam di sistem
    if durasi_jam >= 0 and durasi_jam <= 1:
        biaya = 1000
    elif durasi_jam > 1 and durasi_jam <= 9:
        biaya = 1000 + (durasi_jam - 1) * 500
    else:
        biaya = 5000

    # Kembalikan slot parkir
    slot_tersedia_motor.append(slot)
    slot_tersedia_motor.sort()

    print(f"Motor dengan plat {plat_nomor} keluar dari slot {slot}.")
    print(f"Waktu masuk: {waktu_masuk}, Waktu keluar: {waktu_keluar}")
    print(f"Durasi parkir: {durasi_jam:.2f} jam")
    print(f"Biaya parkir: Rp{biaya:.2f}")

else:
    print("Kendaraan dengan plat tersebut tidak ditemukan atau jenis kendaraan salah.")
```



ALGORITMA: 3. STATUS PARKIR

```
elif pilihan == '3': # Tampilkan status parkir
    print("\nStatus Tempat Parkir:")
    print("Slot mobil yang tersedia:", slot_tersedia_mobil)
    print("Slot motor yang tersedia:", slot_tersedia_motor)

    print("\nMobil yang terparkir:")
    if mobil_terparkir:
        for plat_nomor, info in mobil_terparkir.items():
            print(f"Plat {plat_nomor} di slot {info['slot']}, Waktu masuk: {info['waktu_masuk']}")
    else:
        print("Tidak ada mobil yang terparkir.")

    print("\nMotor yang terparkir:")
    if motor_terparkir:
        for plat_nomor, info in motor_terparkir.items():
            print(f"Plat {plat_nomor} di slot {info['slot']}, Waktu masuk: {info['waktu_masuk']}")
    else:
        print("Tidak ada motor yang terparkir.")
    print("\n")
```



ALGORITMA: 4. KELUAR PROGRAM

```
elif pilihan == '4': # Keluar
    print("Terima kasih telah menggunakan GateX.")
    break

else:
    print("Pilihan tidak valid. Silakan coba lagi.")
```



THANK YOU!

Do you have any questions?

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon** and infographics & images by **Freepik**