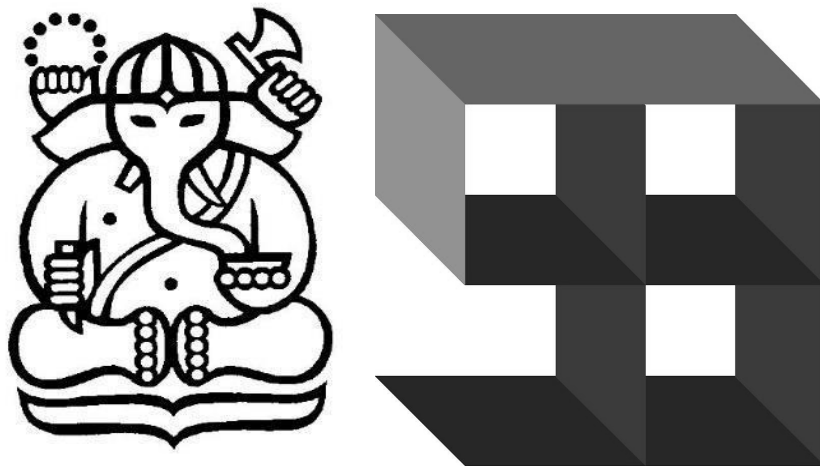


**LAPORAN**  
**KU1102 PENGENALAN KOMPUTASI**  
**SEMESTER I 2023-2024**

**“Markirr™”**

**Pengembangan Sistem Manajemen Parkir Otomatis sebagai Solusi Tata  
Letak Kendaraan di Kampus Jatinangor**



**Dibimbing oleh:**

Yuda Sukmana, S.Pd., M.T.

**Disusun oleh :**

William Anthony (16523109)  
Muhammad Jafar Fadli (16523137)  
Zulfaqqar Nayaka Athadiansyah (19623116)  
Abdullah Farhan (19623305)

**PROGRAM TAHAP PERSIAPAN BERSAMA**  
**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**  
**INSTITUT TEKNOLOGI BANDUNG**

**2023**

## KATA PENGANTAR

Dengan memanjatkan puja dan puji syukur kehadiran Tuhan Yang Maha Esa, kami dapat menyelesaikan laporan ini. Laporan ini merupakan hasil dari tugas besar KU1102 yang kami susun dengan tujuan untuk mengisi penilaian tugas besar dan memberikan solusi dalam manajemen parkir di Kampus Jatinangor.

Pelaksanaan penulisan laporan ini tidak lepas dari bantuan dan arahan dari berbagai pihak. Oleh karena itu, kami mengucapkan terima kasih kepada semua pihak yang telah mendukung penyusunan Laporan ini, di antaranya:

**Dosen Pembimbing** : Yuda Sukmana, S.Pd., M.T.

<b>Tim Penyusun</b>	: William Anthony	(16523109)
	Muhammad Jafar Fadli	(16523137)
	Zulfaqqar Nayaka Athadiansyah	(19623116)
	Abdullah Farhan	(19623305)

Akhir kata, kami berharap laporan ini dapat menjadi penunjang hasil IP KU1102 kami dan bermanfaat bagi pembaca serta dapat menjadi bahan pelajaran untuk kami selaku mahasiswa Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung.

Dengan penuh kesadaran, kami telah berupaya dengan sepenuh hati untuk mencapai kesempurnaan dalam penyusunan laporan ini. Namun, kami menyadari bahwa masih ada kekurangan dalam penulisan laporan ini. Oleh karena itu, kami sangat menghargai dan berharap untuk menerima saran serta kritik yang konstruktif untuk meningkatkan kualitas penulisan laporan kami di masa mendatang.

## DAFTAR ISI

Pengembangan Sistem Manajemen Parkir Otomatis sebagai Solusi Tata Letak Kendaraan di Kampus Jatinangor.....	1
<b>KATA PENGANTAR.....</b>	<b>2</b>
<b>DAFTAR ISI.....</b>	<b>3</b>
<b>ISI.....</b>	<b>4</b>
I Latar Belakang.....	4
I.1 Mobilisasi Letak Kendaraan yang Tidak Optimal.....	4
I.2 Kesulitan untuk Mencari Parkir di Kampus.....	5
I.3 Keingintahuan untuk Mengetahui Letak dan Durasi Kendaraan Selama dalam Area Parkir.....	5
I.4 Dekomposisi Permasalahan.....	6
II ALUR KERJA PROGRAM.....	7
II.1 Deskripsi Program.....	7
II.2 Pseudocode/Flowchart.....	8
II.3 GUI.....	9
III Penjelasan Program.....	10
III.1 Program Python yang diciptakan.....	10
III.2 Metode yang diterapkan.....	12
III.3 Presentasi.....	12
IV. Kesimpulan dan Lesson Learned.....	13
V. Pembagian Tugas dalam kelompok.....	14
<b>DAFTAR PUSTAKA.....</b>	<b>15</b>

# ISI

## I Latar Belakang

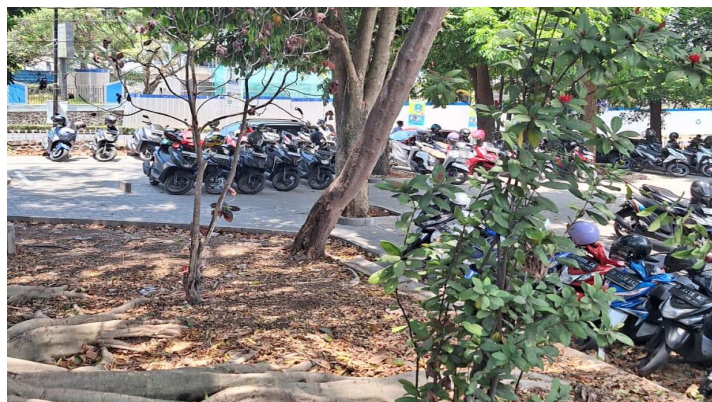


sumber: [https://jatinangor.itb.ac.id/wp-content/uploads/sites/382/2023/04/IMG\\_2948-feat.jpg](https://jatinangor.itb.ac.id/wp-content/uploads/sites/382/2023/04/IMG_2948-feat.jpg)

Kampus Institut Teknologi Bandung (ITB) Jatinangor adalah pusat pendidikan dan penelitian yang ramai. Dengan jumlah mahasiswa, dosen, dan staf yang besar, mobilisasi kendaraan menjadi tantangan tersendiri. Dengan ini kami mampu merumuskan hal-hal yang kami terapkan sebagai fitur dengan menelusuri dahulu permasalahan-permasalahan yang dialami:

### I.1 Mobilisasi Letak Kendaraan yang Tidak Optimal

Salah satu isu utama yang dihadapi adalah kendaraan yang diparkir secara acak di berbagai tempat di kampus. Dengan ketidaktahuan letak kendaraan, parkir letak kendaraan cenderung tidak sesuai. Dokumentasi terhadap kejadian ini dapat diamati di instagram @itbparkir dimana terdapat beragam kasus letak kendaraan yang menyebabkan isu kemacetan, isu kesulitan pejalan kaki, dan isu kesulitan akses menuju fasilitas penting.



## **I.2 Kesulitan untuk Mencari Parkir di Kampus**

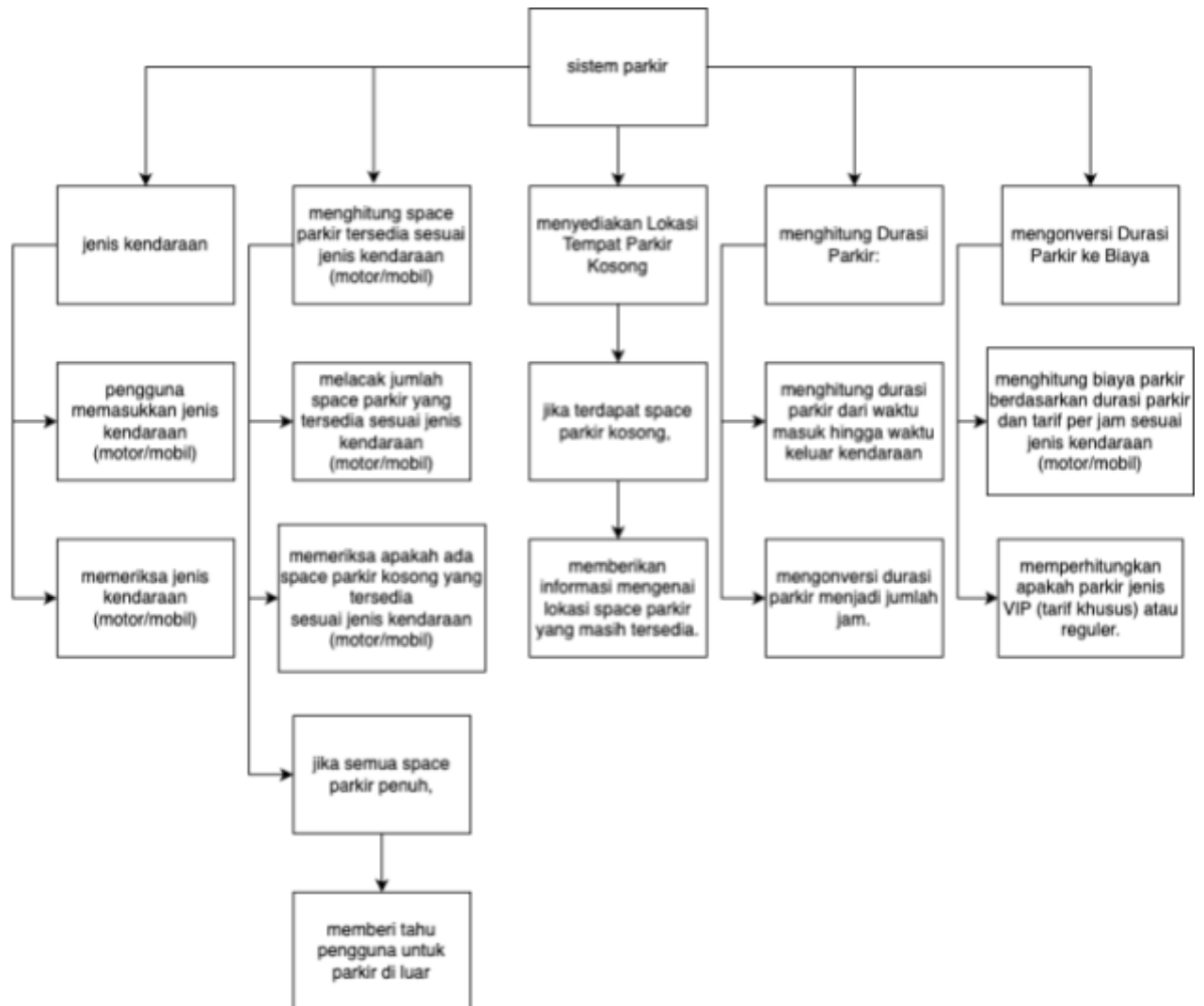
Dengan jumlah kendaraan dari massa kampus yang banyak dan ruang parkir terbatas, mencari tempat parkir bisa menjadi aktivitas yang melelahkan dan tidak efisien waktu. Proses ini dapat mengambil waktu yang dapat dimanfaatkan dalam hal-hal bermanfaat lainnya.



## **I.3 Keingintahuan untuk Mengetahui Letak dan Durasi Kendaraan Selama dalam Area Parkir**

Tanpa sistem manajemen parkir yang efektif, sulit untuk mengetahui berapa lama kendaraan telah diparkir atau di mana tepatnya kendaraan tersebut berada. Hal ini tidak hanya menyulitkan pengendara yang mencari kendaraannya, tetapi juga pengelola parkir yang perlu memantau dan mengatur ruang parkir mereka secara efisien. Keingintahuan ini muncul dari pengalaman pribadi penulis, yang mungkin telah mengalami kesulitan dalam menemukan kendaraannya di area parkir yang luas atau tidak dapat mengingat berapa lama kendaraannya telah diparkir. Situasi ini dapat menimbulkan frustrasi pada penggunaan parkir.

## I.4 Dekomposisi Permasalahan



## II ALUR KERJA PROGRAM



Dengan referensi beragam , kami berhasil merancang “Markirr™”. Program “Markirr™” adalah sistem manajemen parkir yang dirancang untuk menghitung biaya parkir berdasarkan jenis kendaraan dan durasi parkir. Program ini menggunakan struktur data seperti kamus dan matriks untuk menyimpan informasi tentang ruang parkir dan biaya parkir. Kemudian, pemanfaatan perulangan `for` dan `while`, fungsi (`def`). serta kondisional (`if-else`) untuk automasi parkir.

### II.1 Deskripsi Program

Setelah melakukan penelusuran permasalahan, kami merumuskan bahwa program harus mempunyai fitur yang mampu melakukan hal-hal berikut:

- Menentukan tipe kendaraan, apakah itu mobil atau motor.
- Mengkalkulasi kapasitas yang tersedia di area parkir sesuai dengan tipe kendaraan (jika area tertentu sudah penuh, pengguna harus mencari area lain yang masih tersedia. Jika tidak ada ruang tersedia, maka kendaraan harus parkir di luar wilayah yang ditentukan).
- Memberikan informasi tentang lokasi tempat parkir yang masih kosong.
- Menghitung durasi waktu parkir sebuah kendaraan.
- Menentukan biaya yang harus dibayarkan berdasarkan durasi waktu parkir.

Dengan ini kita merumuskan alur program ini sebagai berikut.

Program ini dimulai dengan inisialisasi data tempat parkir dan biaya parkir serta pendefinisian sejumlah fungsi dan prosedur. Kemudian, program meminta pengguna untuk memasukkan jenis kendaraan mereka. Jika ruang parkir tersedia, program akan mencari ruang parkir kosong dan memberikan informasi kepada pengguna tentang lokasi parkir mereka.

Setelah itu, program mencatat waktu masuk kendaraan dan menunggu pengguna untuk memberi tahu saat kendaraan keluar. Program kemudian menghitung durasi parkir dan biaya parkir berdasarkan tarif per jam dan durasi parkir.

Selain itu, program ini juga mencetak matriks ruang parkir untuk memberikan visualisasi tentang ruang parkir yang tersedia. Program ini terus berjalan dalam loop sampai pengguna memilih untuk berhenti.

Pada akhirnya kami berhasil merumuskan “Markirr” dengan slogan “Solusi Parkir Anti Mikir”.

## II.2 Pseudocode/Flowchart

**Judul:** Sistem Parkir ITB Pseudocode

**Deklarasi:**

1. Inisialisasi variabel parkir  
parking\_space: {'motor': 10, 'mobil': 20}
2. Inisialisasi biaya parkir per jam  
biaya\_parkir: {'motor': 2000, 'mobil': 5000}
3. Inisialisasi data parkir  
data\_parkir: {'motor': {}, 'mobil': {}}
4. Inisialisasi matriks space parkir  
matrix\_motor: Matriks 2x10 (untuk motor)  
matrix\_mobil: Matriks 2x20 (untuk mobil)

**Implementasi:**

Selama program berjalan:

1. Tampilkan matriks space parkir
2. Tampilkan pesan "Sistem Parkir ITB"
3. Identifikasi jenis kendaraan (motor/mobil)
4. Pengguna memasukkan plat nomor mereka

Jika masih ada space parkir kosong untuk jenis kendaraan:

1. Cari space parkir kosong
  - a. Jika ditemukan space parkir kosong:
    - i. Catat waktu masuk
    - ii. Tunggu hingga kendaraan keluar (tekan Enter)
    - iii. Catat waktu keluar
    - iv. Hitung durasi parkir
    - v. Hitung biaya parkir
    - vi. Tampilkan durasi parkir dan biaya
    - vii. Catat data parkir
    - ix. Tandai space parkir sebagai terisi di matriks
  - b. Jika semua space parkir sudah terisi:  
Tampilkan pesan "Maaf, semua space parkir sudah terisi."
2. Jika tidak ada space parkir kosong untuk jenis kendaraan:  
Tampilkan pesan "Maaf, tidak ada space parkir untuk kendaraan jenis ini,  
Silakan parkir di luar ITB."



### II.3 GUI

Antarmuka pengguna grafis (GUI) dari “Markiri™” didesain di Figma. Desain ini kemudian diimplementasikan dalam Python menggunakan *library* `tkinter`. Tautan dari publikasi GUI di Figma dapat diakses [di sini](#) dan [di sini](#).

### III Penjelasan Program

Program memiliki fitur-fitur untuk menyelesaikan masalah manajemen parkir. Fitur utamanya meliputi penentuan tipe kendaraan, kalkulasi kapasitas parkir, penyediaan informasi lokasi parkir, pengukuran durasi parkir, dan perhitungan biaya parkir. Program dimulai dengan inisialisasi data parkir dan biaya, kemudian meminta input dari pengguna. Jika ruang tersedia, program mencari ruang kosong dan memberikan informasi lokasi. Program mencatat waktu masuk dan keluar kendaraan, menghitung durasi dan biaya parkir. Program juga mencetak matriks ruang parkir untuk visualisasi dan berjalan dalam loop sampai pengguna memilih untuk berhenti.

#### III.1 Program Python yang diciptakan

```
import tkinter as tk
from datetime import datetime

# Membuat jendela
window = tk.Tk()
window.title("Markirr System")

# Membuat tempat dan label untuk tipe kendaraan dan plat nomor
label_type = tk.Label(window, text="Vehicle Type (e.g., Car, Motorcycle):")
label_type.pack()

entry_type = tk.Entry(window)
entry_type.pack()

label_plate = tk.Label(window, text="License Plate:")
label_plate.pack()

entry_plate = tk.Entry(window)
entry_plate.pack()

# Membuat tombol parkir
def park_vehicle():
    # Mengambil input dari pengguna
    vehicle_type = entry_type.get()
    license_plate = entry_plate.get()
    # jika kendaraan adalah mobil, gunakan ruang parkir mobil
    if vehicle_type.lower() == "car":
        available_spaces = car_spaces
    # jika kendaraan adalah motor, gunakan ruang parkir motor
    elif vehicle_type.lower() == "motorcycle":
        available_spaces = motorcycle_spaces
    else: # jika kendaraan tidak dikenal (selain motor/mobil)
        available_spaces = [] # Maka tidak menempati parkir

    if available_spaces:
        message = f"Parked {vehicle_type} with license plate {license_plate}."
        space = available_spaces.pop()

        # Mencari tempat parkir kosong terdekat
        row, col = space
        entry_time = datetime.now()
        parking_spaces[row][col].config(text=license_plate)
        parked_vehicles[row][col] = {
```

```

        "vehicle_type": vehicle_type,
        "license_plate": license_plate,
        "entry_time": entry_time,
    }
#dictionary untuk dipakai untuk fungsi yang memperlihatkan biaya parkir
dan identitas kendaraan(di akhir)
    else:
        message = f"No available parking space for {vehicle_type} with
license plate {license_plate}."

        result_label.config(text=message)

park_button = tk.Button(window, text="Park Vehicle", command=park_vehicle)
park_button.pack()

# Membuat label untuk hasil parkir
result_label = tk.Label(window, text="")
result_label.pack()

# Membuat layout tempat parkir menggunakan matriks
parking_matrix = tk.Frame(window)
parking_matrix.pack()

# Membuat tempat parkir menggunakan matriks
parking_spaces = [] # Daftar label untuk mewakili tempat parkir
parked_vehicles = [] # Daftar kendaraan yang terparkir
num_rows = 5 # Jumlah baris tempat parkir
num_cols = 10 # Jumlah kolom tempat parkir

# Mengisi matriks parkir dengan label "Kosong"
for i in range(num_rows):
    row = [] # Daftar label untuk satu baris
    vehicle_row = [] # Daftar kendaraan terparkir di satu baris
    for j in range(num_cols):
        parking_space = tk.Label(parking_matrix, text="Empty",
borderwidth=1, relief="solid", width=8)
        parking_space.grid(row=i, column=j, padx=2, pady=2)
        row.append(parking_space)
        vehicle_row.append(None)
    parking_spaces.append(row)
    parked_vehicles.append(vehicle_row)

# Membuat tempat parkir terpisah untuk kendaraan mobil dan motor
car_spaces = [(0, i) for i in range(num_cols)] # Baris 0 untuk mobil
motorcycle_spaces = [(i, j) for i in range(1, num_rows) for j in
range(num_cols)] # Baris 1-4 untuk motor

# Fungsi untuk mengeluarkan kendaraan dari tempat parkir
def exit_vehicle(row, col):
    if parked_vehicles[row][col] is not None:
        vehicle = parked_vehicles[row][col]
        exit_time = datetime.now()
        #mencatat waktu keluar kendaraan
        duration = exit_time - vehicle["entry_time"]
        duration_hours = duration.total_seconds() / 3600
        #mengkonversi detik menjadi jam(biar sesuai def hitung biayanya)
        vehicle_type = vehicle["vehicle_type"]
        license_plate = vehicle["license_plate"]
        parking_cost = calculate_parking_cost(vehicle_type, duration_hours)

```

```

        message = f"Vehicle Type: {vehicle_type}\nLicense Plate: {license_plate}\nEntry Time: {vehicle['entry_time']}\nExit Time: {exit_time}\nDuration (hours): {duration_hours:.2f}\nParking Cost: Rp {parking_cost:.2f}" #membuat label untuk menampilkan identitas kendaraan

        parking_spaces[row][col].config(text="Empty")
        parked_vehicles[row][col] = None
        result_label.config(text=message) #menampilkan "message"
        if vehicle_type.lower() == "car": # Jika merupakan mobil...
            car_spaces.append((row, col))
        else: # Jika merupakan motor...
            motorcycle_spaces.append((row, col))
    else:
        result_label.config(text="No vehicle parked in this space.")

for row in range(num_rows): # Deklarasi Barisan
    for col in range(num_cols): # Deklarasi Kolom
        parking_spaces[row][col].bind("<Button-1>", lambda event, r=row, c=col: exit_vehicle(r, c)) # Membuat parkir

# Fungsi untuk menghitung biaya parkir
def calculate_parking_cost(vehicle_type, duration_hours):
    if vehicle_type.lower() == "car":
        rate = 5000 # Biaya per jam untuk mobil
    elif vehicle_type.lower() == "motorcycle":
        rate = 2000 # Biaya per jam untuk motor
    else: # Jika tidak ada maka aman
        rate = 0

    parking_cost = rate * duration_hours
    return parking_cost

# Menjalankan perulangan utama
window.mainloop()

```

### III.2 Metode yang diterapkan

Program menerapkan metode berbasis data untuk mengelola sistem parkir. Pertama, program menginisiasi data tempat parkir yang mencakup informasi tentang ruang dan tarif parkir untuk berbagai jenis kendaraan dan pembayaran. Kemudian, program meminta pengguna untuk memasukkan jenis kendaraan dan jenis pembayaran mereka. Jika input ini valid, program akan meminta pengguna untuk memasukkan plat nomor mereka. Selanjutnya, program menghitung biaya parkir berdasarkan tarif per hari dan durasi parkir. Terakhir, program memeriksa ketersediaan ruang parkir dan memberikan informasi kepada pengguna. Jika tidak ada ruang parkir yang tersedia, program akan memberi tahu pengguna bahwa tempat parkir penuh.

### III.3 Presentasi

Untuk presentasi, diletakkan pada link berikut:

(pdf)  PRESENTASI KELOMPOK 1 PENGKOM\_KU1102.pdf

#### IV. Kesimpulan dan *Lesson Learned*



Secara keseluruhan, program “Markirr™” memberikan pelajaran berharga tentang bagaimana mengimplementasikan sistem manajemen parkir menggunakan pemrograman. Dari program ini, kita dapat belajar tentang penggunaan struktur data seperti larik (*array*), matriks, dan kamus dalam Python untuk menyimpan dan mengakses data secara efisien. Program ini juga menunjukkan bagaimana input pengguna dapat digunakan untuk mengambil data yang relevan dari struktur data tersebut.

Selain itu, kita juga dapat memahami pentingnya kontrol alur dalam pemrograman. Program ini menggunakan pernyataan kondisional untuk memvalidasi input pengguna dan memberikan umpan balik yang sesuai. Ini membantu dalam memastikan bahwa program berjalan dengan benar dan dapat menangani berbagai skenario.

Lebih lanjut lagi, pada tugas besar ini kami juga menerapkan library `tkinter` dan `datetime` serta mengeksplor seluk-beluk antarmuka pengguna grafis (*graphic user interface: GUI*) dan langsung mengimplementasikannya dalam program kami. Meskipun belum mampu menerapkan *object-oriented programming*, kami berhasil menciptakan aplikasi yang fungsional.

Terakhir, program ini juga menggunakan operasi matematika sederhana seperti perkalian untuk menghitung biaya parkir berdasarkan tarif per hari dan durasi parkir. Secara keseluruhan, program ini adalah contoh yang baik tentang bagaimana penggunaan *if/else*, perulangan *for/while*, larik (*array*), fungsi, dan pemrograman dasar secara umum dapat digunakan untuk membuat sistem yang kompleks dan fungsional seperti sistem manajemen parkir.



## **V. Pembagian Tugas dalam kelompok**

William Anthony	(16523109): Lead PPT and Editor Laporan
Muhammad Jafar Fadli	(16523137): Lead Programming structure and Editor
Zulfaqqar Nayaka Athadiansyah	(19623193): GUI Developer and Lead Simulation
Abdullah Farhan	(19623305): Lead Programmer Analyst and Editor

## DAFTAR PUSTAKA

ANALISIS SISTEM PERPARKIRAN DAN PENGEMBANGAN JARINGAN TRANSPORTASI PADA KAWASAN PANTAI LOSARI KOTA MAKASSAR. (2015). Bandung: Institut Teknologi Bandung. Diakses pada 30 Oktober 2023 pukul 19:09 melalui [be6b307236f09d3d9132c6725d579fd7.pdf](https://digilib.itb.ac.id/index.php/gdl/view/61123)

PERANCANGAN SISTEM PARKIR CERDAS MENGGUNAKAN PENCITRAAN DIGITAL BERBASIS PYTHON. (2021). Bandung: Institut Teknologi Bandung. Diakses pada 30 Oktober 2023 pukul 17:54 melalui <https://digilib.itb.ac.id/index.php/gdl/view/61123>.

Implementation of Decision Tree in Managing Parking Space in ITB. (2013). Bandung: Institut Teknologi Bandung. Diakses pada 30 Oktober 2023 pukul 18:01 melalui <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2013-2014/Makalah2013/MakalahIF2120-2013-007.pdf>