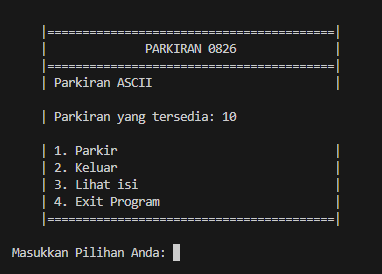
**Nama : Danang Tri Atmaja**

**NIM : 22.83.0826**

**Kelas : 22-TK02**

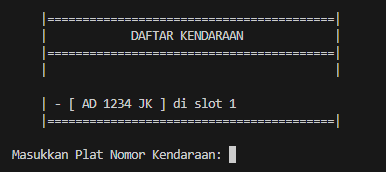
**PROGRAM PYTHON PARKIRAN**



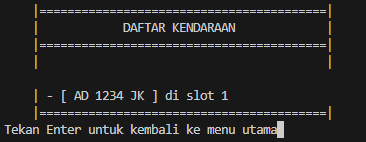
**Home Screen**



**Pilihan 1**



**Pilihan 2**



**Pilihan 3**

1. **Penjelasan Program**

Program ini ditujukan pada pintu masuk dan pintu keluar parkiran agar kendaraan tetap terdata dengan baik, ditambah dengan UI ASCII untuk meningkatkan user experience.

1. **Source Code**

import datetime

import time

import os

import numpy as np

max\_parkiran = 10

parking\_lot = np.zeros(max\_parkiran, dtype=int) #numpy array lahan parkir

parked\_vehicles = {}

log = {}

log\_counter = 0

def main():

    while True:

        render\_ui()

        choice = input("Masukkan Pilihan Anda: ")

        if choice == '1':

            park()

        elif choice == '2':

            render\_parking\_list()

            print()

            exit\_parking()

        elif choice == '3':

            render\_parking\_list()

            input("Tekan Enter untuk kembali ke menu utama")

        elif choice == '4':

            exit\_program()

        else:

            render\_error("Inputan tidak Valid!")

            time.sleep(2)

def park():

    global max\_parkiran

    if len(parked\_vehicles) == max\_parkiran:

        render\_error("Parkiran Penuh!")

        time.sleep(2)

    else:

        plate\_number = input("Masukkan Plat Nomor Kendaraan: ")

        print()

        plate\_number = plate\_number.upper()

        if plate\_number in parked\_vehicles:

            render\_error("Kendaraan dengan Plat Nomor tersebut sudah terparkir!")

            time.sleep(2)

            return

        parking\_slot = np.where(parking\_lot == 0)[0][0]

        parking\_lot[parking\_slot] = 1

        parked\_vehicles[plate\_number] = parking\_slot

        print("Kendaraan", plate\_number, "telah diparkir di slot", parking\_slot + 1)

        max\_parkiran -=1

        time.sleep(2)

def exit\_parking():

    global log\_entry, max\_parkiran

    if len(parked\_vehicles) == 0:

        render\_error("Tidak ada kendaraan yang terparkir di parkiran")

        time.sleep(2)

    else:

        plate\_number = input("Masukkan Plat Nomor Kendaraan: ")

        print()

        plate\_number = plate\_number.upper()

        if plate\_number not in parked\_vehicles:

            render\_error("Kendaraan tidak terparkir di parkiran")

            time.sleep(2)

            return

        parking\_slot = parked\_vehicles[plate\_number]

        parking\_lot[parking\_slot] = 0

        del parked\_vehicles[plate\_number]

        print("Kendaraan", plate\_number, "telah keluar dari parkiran")

        print()

        log\_parking\_event(plate\_number)

        log\_exit\_event(plate\_number)

        print('Biaya : Rp.',log\_entry['Biaya'])

        print()

        input ("Tekan enter untuk lanjut.")

        max\_parkiran +=1

def get\_cost(enter\_time, exit\_time):

    parking\_duration = exit\_time - enter\_time

    parking\_hours = parking\_duration.total\_seconds() / 3600

    parking\_rate\_per\_hour = 3000

    parking\_rate\_much = 5000

    if parking\_hours <= 1:

        parking\_cost = parking\_rate\_per\_hour

    elif parking\_hours > 1:

        parking\_cost = parking\_rate\_much

    return parking\_cost

def render\_ui():

    clear\_screen()

    menu = '''

    |=========================================|

    |              PARKIRAN 0826              |

    |=========================================|

    | Parkiran ASCII                          |

    '''

    print(menu)

    print("    | Parkiran yang tersedia:", max\_parkiran)

    lanjut = '''

    | 1. Parkir                               |

    | 2. Keluar                               |

    | 3. Lihat isi                            |

    | 4. Exit Program                         |

    |=========================================|

    '''

    print(lanjut)

def render\_error(message):

    clear\_screen()

    error = '''

    |=========================================|

    |                 ERROR                   |

    |=========================================|

    |      \_\_\_\_\_\_                             |

    |     |  \_\_\_\_|                            |

    |     | |\_\_    \_ \_\_  \_ \_\_  \_\_\_   \_ \_\_     |

    |     |  \_\_|  | '\_\_|| '\_\_/ \_ \ | '\_\_|     |

    |     | |\_\_\_\_ | |   | | | (\_) || |        |

    |     |\_\_\_\_\_\_||\_|   |\_|  \\_\_\_/ |\_|        |

    |=========================================|

    '''

    print(error)

    print(message)

def render\_exit():

    clear\_screen()

    exit = '''

    |=========================================|

    |             TERIMA KASIH                |

    |=========================================|

    |                                         |

    |    Terima kasih atas kunjungan Anda     |

    |=========================================|

    '''

    print(exit)

def render\_parking\_list():#fungsi menampilkan list kendaraan yang terparkir

    clear\_screen()

    list\_park = '''

    |=========================================|

    |            DAFTAR KENDARAAN             |

    |=========================================|

    |                                         |

    '''

    print(list\_park)

    if len(parked\_vehicles) == 0:

        print("    |            Parkiran Kosong !            |")

    else:

        for plate\_number, parking\_slot in parked\_vehicles.items():

            print("    | - [", plate\_number, "] di slot", parking\_slot + 1)

    print('    |=========================================|')

def exit\_program(): #fungsi keluar program dan menampilkan data data yang telah tersimpan

    render\_exit()

    time.sleep(3)

    clear\_screen()

    for log\_entry in log.values():

        print('Plat:', log\_entry['Plat'],'=====> Rp.', log\_entry['Biaya'])

        print()

    input("Tekan tombol apapun untuk mengakhiri")

    exit()

def clear\_screen(): #Fungsi memperbarui CLI

    os.system('cls' if os.name == 'nt' else 'clear')

def log\_parking\_event(plate\_number):

    global log\_counter, log\_entry

    log\_counter += 1

    log\_entry = { #Dictionary menampung log keluar & masuk kendaraan

        'Plat': plate\_number,

        'Waktu Masuk': datetime.datetime.now()

    }

    log[log\_counter] = log\_entry

def log\_exit\_event(plate\_number):

    global log\_counter

    log\_entry = log[log\_counter]

    log\_entry['Waktu Keluar'] = datetime.datetime.now()

    masuk = log\_entry['Waktu Masuk']

    keluar = log\_entry['Waktu Keluar']

    log\_entry['Biaya'] = get\_cost(masuk, keluar)

main()

1. **Referensi**
2. <https://www.anandanesia.com/menghitung-waktu-dan-biaya-parkir-dengan-python/>
3. <https://srirahayuuu.blogspot.com/2016/10/python_11.html>
4. <https://www.erikadielsson.com/program-menghitung-biaya-parkir-python/>