

# **LAPORAN PROJEK PEMROGRAMAN BERBASIS OBJEK**

## **Sistem Parkir Online**



**Disusun oleh:**

**KELOMPOK 6**

**Anggota:**

**5210411213 Venus Al-Fatah**

**5210411215 William Kessler Suryanto**

**5210411231 Putri Wahyuni**

**PROGRAM STUDI INFORMATIKA  
FAKULTAS SAINS & TEKNOLOGI  
UNIVERSITAS TEKNOLOGI YOGYAKARTA  
YOGYAKARTA  
2022**

### **Pembagian Tugas Kelompok 6:**

- 5210411213 Venus Al-Fatah  
Tugas : Mengimplementasikan ke dalam program
- 5210411215 William Kessler Suryanto  
Tugas : Membuat rancangan class diagram
- 5210411231 Putri Wahyuni  
Tugas : Membuat laporan proyek

# **BAB I**

## **PENDAHULUAN**

### **A. Latar Belakang Masalah**

Pada saat ini sistem keamanan parkir sudah mengalami banyak perkembangan akibat kemajuan teknologi yang sangat pesat dan adanya inovasiinovasi terbaru yang diciptakan pada sistem keamanan parkir. Kendaraan yang di parkir pada area parkir yang rendah sistem keamanannya dapat mengakibatkan terjadinya kasus kriminalitas seperti hilangnya kendaraan bermotor. Sebaiknya sistem keamanan parkir menjadi salah satu prioritas utama dalam pembangunan tempat-tempat umum agar dapat mengurangi tingginya tingkat pencurian atau kehilangan kendaraan bermotor sehingga tidak ada pihak yang merasa dirugikan saat mengunjungi tempat tersebut. Saat ingin memasuki area parkir umumnya pemilik kendaraan bermotor akan mengambil karcis atau tiket parkir terlebih dahulu sebagai tanda bukti bahwa kendaraan akan parkir di tempat tersebut dan akan dilakukan pengambilan gambar secara otomatis menggunakan Web Camera.

Hal ini dapat menyebabkan persediaan kertas untuk tiket parkir akan habis sehingga beberapa pemilik yang akan parkir di tempat tersebut tidak mendapatkan tiket parkir karena persediaan kertas tidak dapat diganti secara otomatis dan gambar yang dihasilkan oleh Web Camera juga memiliki resolusi yang tidak terlalu tinggi. Dengan dibuatnya sistem keamanan parkir tersebut, maka kami merasa bahwa tindakan pencurian kendaraan bermotor di lokasi parkir dapat dicegah sehingga tidak ada pihak yang merasa dirugikan.

### **B. Rumusan Masalah**

Dari latar belakang masalah diatas kami dapat mengambil rumusan masalah sebagai berikut :

- a. Bagaimana merancang aplikasi parkir yang dapat menentukan lokasi parkir kendaraan?
- b. Bagaimana kinerja aplikasi parkir dalam menentukan lokasi parkir kendaraan?
- c. Bagaimana merancang sistem untuk menampilkan data saat pemilik kendaraan bermotor akan keluar dari lokasi parkir ?

## **C. Tujuan**

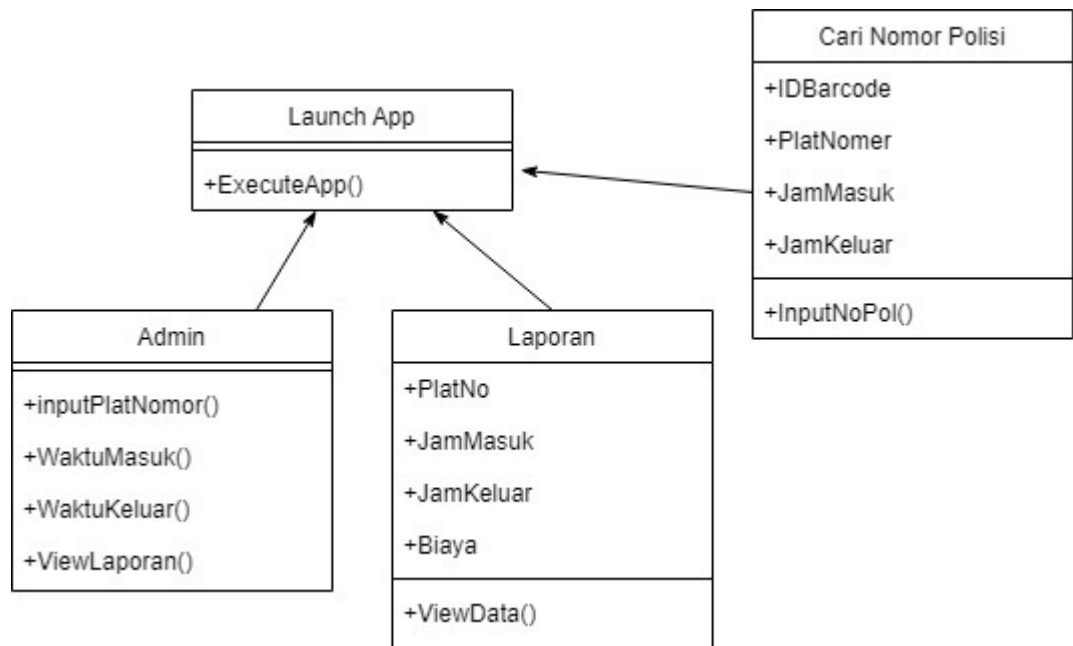
Adapun yang menjadi tujuan dari penelitian yang akan dilakukan adalah :

- a. Menciptakan aplikasi parkir yang mampu menentukan lokasi parkir.
- b. Mengetahui kinerja aplikasi parkir dalam menentukan lokasi parkir kendaraan.
- c. Memenuhi tugas proyek mata kuliah Pemograman Berorientasi Objek  
Praktik

## BAB II PEMBAHASAN

### A. Class Diagram

Class diagram adalah jenis diagram struktur statis dalam UML yang menggambarkan struktur sistem dengan menunjukkan sistem *class*, atributnya, metode, dan hubungan antar objek. Class Diagram merupakan suatu diagram yang berfungsi untuk memperlihatkan atau menampilkan struktur dari sebuah sistem, yang dimana sistem tersebut akan menampilkan sistem kelas, atribut, dan juga hubungan anatar kelas yang berada didalam suatu sistem.



Class diagram diatas adalah class diagram dari sistem parker online diaman class diagram tersebut mempunyai satu induk yaitu bernama Launch App. Class diagram tersebut mempunyai sub class 3 yaitu terdiri dari Admin, Cari nomor polisi dan Laporan.

## B. Implementasi

Berikut ini adalah implementasi program sistem parkir online dari kelompok 6:

```
parkir.py X
D: > COOLEYAH > PBO VII > TUGAS KELOMPOK > program > parkir.py > ...

1 from tkinter import *
2 from tkinter import ttk, messagebox
3 from datetime import datetime
4
5 class Parkir():
6     def __init__(self):
7         self.root = Tk()
8         self.datas = []
9         self.root.title("Tiket Parkir by Kelompok 6")
10        self.root.geometry("664x400+450+250")
11        self.root.resizable(width=False, height=False)
12        self.createHome()
13        self.root.mainloop()
14
15    def createHome(self):
16        Label(self.root, text="Aplikasi Parkir Kelompok 6", font=('Comic Sans MS', 15)).place(x=25, y=10)
17        #row 1
18        Label(self.root, text="No Polisi", font=('Comic Sans MS', 10)).place(x=17, y=60)
19        self.nopol = StringVar()
20        Entry(self.root, textvariable=self.nopol).place(x=110, y=62)
21        Button(self.root, text="Cari", font=('Comic Sans MS', 7), width=3, height=1, command=self.searchData).place(x=250, y=60)
22        #row 2
23        Label(self.root, text="No Plat Polisi", font=('Comic Sans MS', 10)).place(x=17, y=100)
24        self.noplat = StringVar()
25        Entry(self.root, textvariable=self.noplat).place(x=110, y=102)
26        #row 3
27        Label(self.root, text="Waktu Masuk", font=('Comic Sans MS', 10)).place(x=17, y=130)
28        self.masuk = StringVar()
29        Entry(self.root, textvariable=self.masuk).place(x=110, y=132)
30        self.masuk.trace("w", self.updateMasuk)
31        #row 4
32        Label(self.root, text="Waktu Keluar", font=('Comic Sans MS', 10)).place(x=17, y=160)
33        self.keluar = StringVar()
34        Entry(self.root, textvariable=self.keluar).place(x=110, y=162)
35        self.keluar.trace("w", self.updateKeluar)
36        #row 5
37        Label(self.root, text="Biaya", font=('Comic Sans MS', 10)).place(x=17, y=190)
38        self.biaya = IntVar()
39        Entry(self.root, textvariable=self.biaya).place(x=110, y=192)
40        Button(self.root, text="Button", font=('Comic Sans MS', 7), width=5, height=1, command=self.insertDataTreeView1).place(x=250, y=190)
41        #biaya per jam 2000
42        Label(self.root, text="Biaya Per Jam", font=('Comic Sans MS', 20), fg="red").place(x=350, y=68)
43        Label(self.root, text="Rp. 2.000", font=('Comic Sans MS', 35), fg="red").place(x=350, y=105)
44        #----- treeview section -----#
45        #Label
46        Label(self.root, text="List Pelanggan Urut Terakhir Keluar", font=('Comic Sans MS', 10), fg="blue").place(x=17, y=240)
47        Label(self.root, text="List Pelanggan Banyak Bayar", font=('Comic Sans MS', 10), fg="blue").place(x=347, y=240)
48        self.DataTreeView1()
49        self.DataTreeView2()
50
51    def DataTreeView1(self):
52        self.treev = ttk.Treeview(selectmode='browse', height=3)
53        self.treev.place(x=19, y=270)
54        verscribar = ttk.Scrollbar(
55            self.root,
56            orient="vertical",
57            command=self.treev.yview
58        )
59        verscribar.place(x=312, y=287)
60        self.treev.configure(xscrollcommand=verscribar.set)
61        self.treev["columns"] = ("1", "2", "3", "4")
62        self.treev["show"] = 'headings'
63        self.treev.column("1", width=80, anchor='c')
64        self.treev.column("2", width=70, anchor='se')
65        self.treev.column("3", width=70, anchor='se')
66        self.treev.column("4", width=70, anchor='se')
67        self.treev.heading("1", text="No Plat Polisi")
68        self.treev.heading("2", text="Masuk")
69        self.treev.heading("3", text="Keluar")
70        self.treev.heading("4", text="Biaya")
```

Pada baris program dari 5 hingga 13 merupakan kelas induk program bernama Parkir. Kemudian pada baris ke 15 hingga baris ke 49 merupakan method dari interface aplikasi parkir online. Sedangkan baris ke 51 hingga baris ke 70 merupakan method inputan data dari aplikasi parkir online

```

72 def DataTreeView2(self):
73     self.treev2 = ttk.Treeview(selectmode="browse", height=3)
74     self.treev2.place(x=350, y=270)
75     verscrbar = ttk.Scrollbar(
76         self.root,
77         orient="vertical",
78         command=self.treev2.yview
79     )
80     verscrbar.place(x=643, y=287)
81     self.treev2.configure(xscrollcommand=verscrbar.set)
82     self.treev2["columns"] = ("1", "2", "3", "4")
83     self.treev2['show'] = 'headings'
84     self.treev2.column("1", width=80, anchor='c')
85     self.treev2.column("2", width=70, anchor='se')
86     self.treev2.column("3", width=70, anchor='se')
87     self.treev2.column("4", width=70, anchor='se')
88     self.treev2.heading("1", text="No Plat Polisi")
89     self.treev2.heading("2", text="Masuk")
90     self.treev2.heading("3", text="Keluar")
91     self.treev2.heading("4", text="Biaya")
92
93 def insertDataTreeView1(self):
94     self.datas.append(
95         {
96             'nopolat': str(self.nopolat.get()),
97             'masuk': str(self.masuk.get()),
98             'keluar': str(self.keluar.get()),
99             'biaya': self.biaya.get()
100         }
101     )
102     self.treev.insert("", 0,
103         values=(str(self.nopolat.get()), str(self.masuk.get()), str(self.keluar.get()), self.biaya.get()))
104     self.clearDataTreeView2()
105     self.insertDataTreeView2()
106

```

Pada baris ke 72 hingga 91 merupakan method inputan data dari program parkir online. Sedangkan pada baris ke 93 hingga baris ke 105 merupakan method untuk menambahkan data seperti no plat, masuk, keluar dan biaya.

```

107 def insertDataTreeView2(self):
108     self.datas.sort(key=self.bSort)
109     for x in self.datas:
110         self.treev2.insert("", 0,
111             values=(x['nopolat'], x['masuk'], x['keluar'], x['biaya']))
112
113 def clearDataTreeView2(self):
114     try:
115         for i in self.treev2.get_children():
116             self.treev2.delete(i)
117     except:
118         print("error")
119
120 def searchData(self):
121     found = 0
122     text = ""
123     for z in self.datas:
124         if str(z['nopolat']).lower().strip() == str(self.nopol.get()).lower().strip():
125             text += "No. {} | No Plat : {} | Jam Masuk : {} | Jam Keluar : {} | Biaya : Rp. {}".format(str(found + 1), str(z['nopolat']), str(z['masuk']), str(z
126             found += 1
127     if found > 0:
128         messagebox.showinfo("Hasil Pencarian : "+str(self.nopol.get()), message=text)
129     else:
130         messagebox.showerror("Tidak ditemukan", "Hasil pencarian : {} tidak ditemukan.".format(str(self.nopol.get())))
131
132 def updateMasuk(self, *args):
133     if len(str(self.masuk.get())) >= 8:
134         if len(str(self.keluar.get())) >= 8:
135             biaya = self.hitungParkir(self.masuk.get(), self.keluar.get())
136             self.biaya.set(biaya)
137             print(biaya)
138         else:
139             self.biaya.set("input jam masuk/keluar")
140
141 def updateKeluar(self, *args):
142     if len(str(self.keluar.get())) >= 8:
143         if len(str(self.masuk.get())) >= 8:
144             biaya = self.hitungParkir(self.masuk.get(), self.keluar.get())
145             self.biaya.set(biaya)
146             print(biaya)
147         else:
148             self.biaya.set("input jam masuk/keluar")
149
150 def hitungParkir(self, masuk, keluar):
151     jam_masuk = str(masuk)
152     jam_keluar = str(keluar)
153     get_hour = (datetime.strptime(jam_keluar, "%H:%M:%S") - datetime.strptime(jam_masuk, "%H:%M:%S")).seconds
154     if (get_hour//3600) > 0 and (get_hour//60)%60 < 50:
155         return get_hour//3600 * 2000
156     elif ((get_hour//60)%60) > 50: # lebih dari 50 menit terhitung 60 menit
157         return ((get_hour//3600) + 1) * 2000
158     else:
159         return 2000
160
161 def bSort(self, e):
162     return e['biaya']
163
164 Parkir()

```

Pada baris ke 107 hingga baris ke 118 adalah method inputan data untuk dimasukan de dalam list. Baris ke 120 hingga baris ke 130 adalah method untuk mencari data kendaraan. Baris ke 132 hingga baris ke 139 adalah method untuk mengupdate masuknya kendaraan. Sedangkan baris ke 141 hingga baris ke 148 adalah method untuk update keluarnya kendaraan. Pada baris ke 150 hingga baris ke 164 merupakan method untuk menghitung biaya kendaraan selama diparkiran.

### **C. Dokumentasi**

