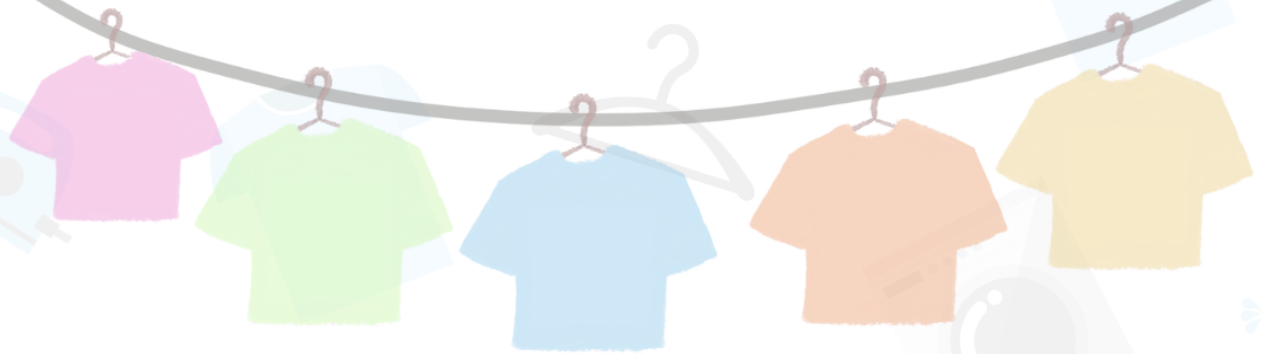


# SILAU



*sistem laundry*



## Kelompok 20

Miranti Sukma Dewi

10322076

Muchammad Wildan Eko Kusuma

10322078

Muhammad Daffa'ul Haqqi Murti

10322083

Neerzalla Atha Nafisah

10322094

Pradipta Daniswara

10322097

## BAB I

### DESKRIPSI MASALAH

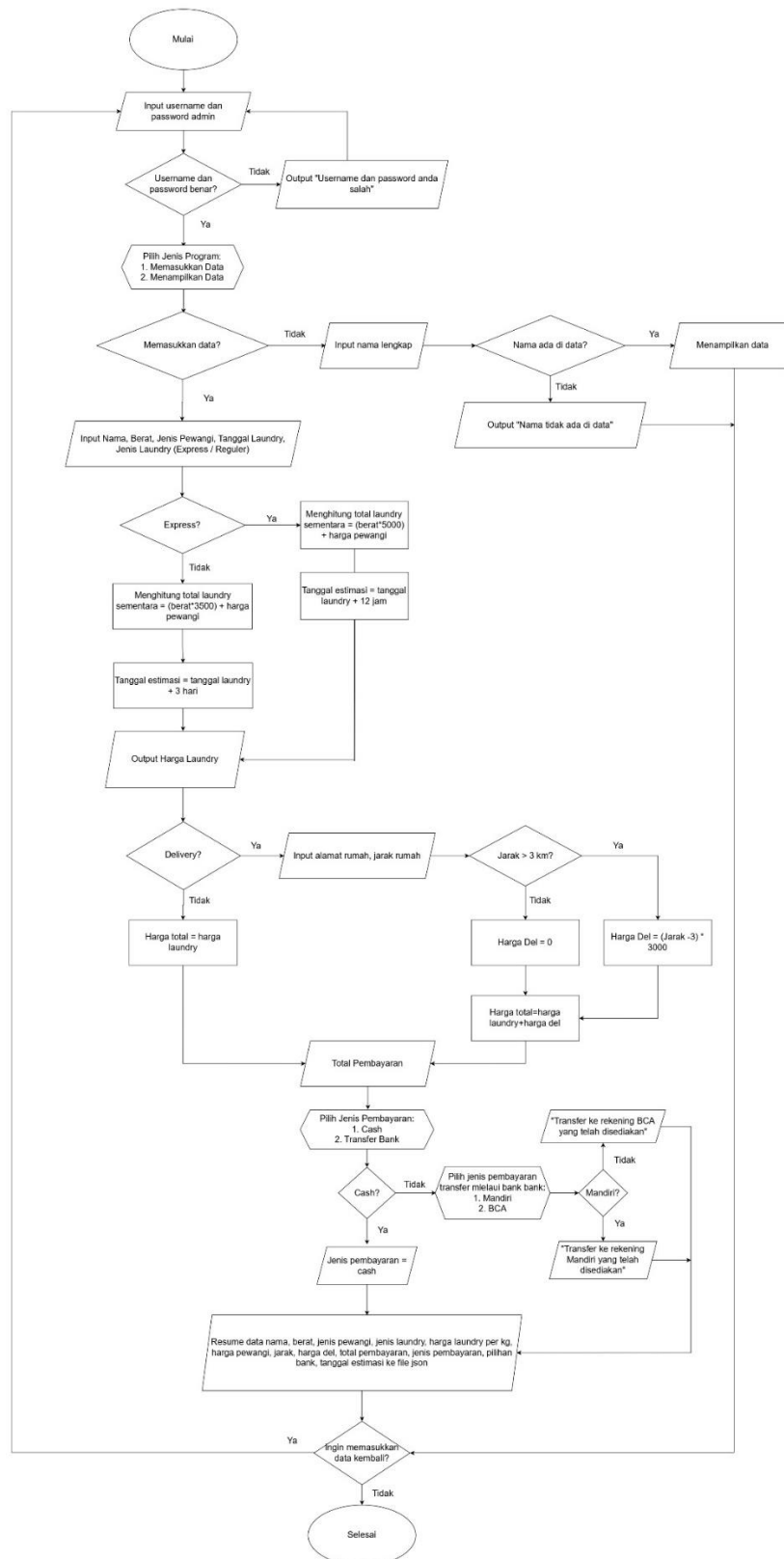
Di zaman yang sudah modern ini, sudah banyak teknologi yang mampu memudahkan manusia dalam mengerjakan dan menyelesaikan masalahnya. Dengan adanya teknologi, pekerjaan manusia bisa mengurangi waktu yang banyak digunakan dan memudahkan manusia untuk melakukan pekerjaannya dengan lebih efektif dan efisien. Akan tetapi, tidak semua teknologi tersedia dalam membantu pekerjaan manusia, salah satunya adalah pengelolaan pesanan *laundry*. Pengelolaan pesanan *laundry* yang kurang efektif sehingga dapat terjadinya pesanan yang tertinggal dan menyebabkan pihak *laundry* terlambat dalam pengiriman cucian ke pelanggan. Selain itu, masih banyak tempat *laundry* yang menggunakan kertas dalam mendaftarkan pesanan dan ketika kertas tersebut sudah tidak digunakan, maka akan menjadi timbunan sampah. Maka dari itu, dibutuhkan sebuah teknologi yang mampu menangani semua masalah tersebut. Teknologi yang dimaksud adalah program sistem manajemen *laundry* atau "Silau".

Program "Silau" dapat menyelesaikan masalah yang ada. Pemilik *laundry* bisa menggunakan program ini untuk memasukkan data-data pelanggan, di mana data-data yang diperlukan adalah nama pelanggan, berat *laundry*, opsi penggunaan iya atau tidaknya *delivery system*, alamat pelanggan, jenis pewangi, dan status pesanan. Dengan menggunakan program ini, pemilik *laundry* dapat melihat dan mengecek pesanan agar tidak ada pesanan yang tertinggal.

Semoga dengan adanya program ini, para pemilik usaha *laundry* bisa melakukan pengelolaan pesanan dengan lebih efektif dan efisien.

## BAB II

### FLOWCHART PROGRAM



**Gambar 2.1. Flowchart Program Silau**

## 2.1. Penjelasan *Flowchart*

Dalam menjalankan program “Silau” diperlukan untuk menginput username serta password milik admin yang telah ditentukan. Jika username dan password salah, user akan dikembalikan ke tampilan awal untuk memasukkan username dan password kembali. Jika username dan password benar maka user akan dilanjutkan ke menu pemilihan jenis program. Jenis program yang tersedia adalah memasukkan data dan menampilkan data.

Apabila user memilih jenis program menampilkan data maka user harus input nama lengkap. Ketika nama yang diinput ada di dalam program maka program akan menampilkan data dari nama tersebut, sedangkan jika nama tidak ada di dalam data maka user akan diarahkan ke bagian pemilihan apakah ingin memasukkan data atau tidak. Ketika user memilih tidak ingin memasukkan data maka program selesai. Namun, apabila user ingin memasukkan data maka user akan diarahkan untuk input username dan password kembali.

Apabila user memilih jenis program untuk memasukkan data, user diminta untuk menginput nama, berat laundry, jenis pewangi yang digunakan, tanggal laundry, dan jenis laundry (express/reguler). Jenis laundry express akan dilanjutkan ke program untuk menghitung total laundry sementara dengan rumus berat laundry dikali dengan harga laundry express per satu kilo yakni Rp5.000,00 lalu ditambah dengan harga pewangi sesuai jenis pewangi yang dipilih. Selanjutnya, program akan memproses estimasi laundry selesai dengan rumus tanggal laundry masuk ditambah 12 jam lalu dilanjutkan ke bagian output harga laundry.

Berbeda dengan jenis laundry express, laundry reguler akan dilanjutkan ke program menghitung total laundry sementara dengan rumus berat laundry dikali dengan harga laundry reguler per satu kilo yakni Rp3.500,00 lalu ditambah dengan harga pewangi sesuai dengan jenis pewangi yang dipilih. Selanjutnya, program akan memproses tanggal estimasi laundry dengan rumus tanggal laundry masuk ditambah 3 hari lalu dilanjutkan dengan output harga laundry.

User akan diminta untuk memilih apakah ingin menggunakan jasa delivery atau tidak. Jika user ingin menggunakan jasa delivery maka akan dilanjutkan ke program input alamat rumah dan jarak antara rumah dengan tempat laundry. Apabila jarak rumah lebih dari 3 km maka program akan memproses harga delivery dengan rumus banyaknya jarak dikurang 3 dan dikalikan dengan harga delivery per 1 km yakni Rp. 3.000,00. Sedangkan jika jarak rumah kurang dari 3 km, maka tidak dikenakan harga delivery. Setelah mendapatkan harga delivery, program akan memproses harga total dengan rumus harga laundry ditambah dengan harga

delivery. Selanjutnya, jika user tidak menggunakan jasa delivery maka harga total sama dengan harga laundry.

Apabila telah mendapatkan harga total maka program akan melanjutkan ke tampilan total pembayaran. Pada tampilan total pembayaran, user diminta untuk memilih jenis pembayaran. Jenis pembayaran yang ditampilkan yaitu cash dan transfer bank. Ketika user memilih pembayaran transfer bank, maka program akan menampilkan jenis bank yaitu bank Mandiri dan bank BCA. Jika user memilih bank Mandiri maka program akan menampilkan output bank Mandiri yang digunakan untuk melakukan pembayaran dan apabila user tidak memilih bank Mandiri maka program akan menampilkan output bank BCA yang akan digunakan untuk melakukan pembayaran. Namun, jika user memilih jenis pembayaran cash maka program akan menghasilkan output jenis pembayaran adalah cash. Setelah user telah selesai menentukan jenis pembayaran, maka program akan memasukan data nama, berat, jenis pewangi, jenis laundry, harga laundry per kg, harga pewangi, jarak, harga delivery, total pembayaran, jenis pembayaran, pilihan bank, dan tanggal estimasi ke file json. Selanjutnya, user diminta untuk memilih apakah ingin memasukan data kembali atau tidak. Jika user ingin memasukan data kembali maka program akan meminta user untuk menginput username dan password yang kemudian dilanjutkan pengisian data seperti pada langkah-langkah yang diatas. Namun, apabila user tidak ingin memasukan data kembali, maka program telah selesai.

## BAB III

### KODE PROGRAM

```

main.py > login
1  import tkinter as tk
2  from tkinter import *
3  import json
4  from tkinter import ttk
5  from tkinter.messagebox import showinfo
6  from PIL import ImageTk, Image
7  from tkinter.messagebox import *
8  import opsi
9
10 def login():
11     username = usernam.get()
12     password = passwor.get()
13     with open('data.json', 'r') as r:
14         a = json.load(r)
15         if username == a["admin"]["username"] and password == a["admin"]["password"]:
16             print("dani")
17             loginwindow.destroy()
18             showinfo(title="Login", message="Login Berhasil")
19             opsi.opsi()
20         else:
21             showwarning(title="Login", message='Username atau Password Salah')
22
23 loginwindow = tk.Tk()
24
25 #variable yang dibutuhkan
26 passwor = tk.StringVar()
27 usernam = tk.StringVar()
28
29
main.py > login
30 loginwindow.title("Login")
31 loginwindow.state('zoomed')
32 screen_width = 1280
33 screen_height = 720
34 loginwindow.geometry("%dx%d" % (screen_width, screen_height))
35 loginwindow.resizable(False, False)
36
37 bg_image = tk.PhotoImage(file='./BAGIAN PERTAMA/username.png')
38 bg_label = tk.Label(loginwindow, image=bg_image)
39 bg_label.pack()
40
41 #username entry
42 usernameEntry=Entry(bg_label,width=25,font=('Courier'), textvariable=usernam)
43 usernameEntry.configure(borderwidth=0, relief="solid", foreground="black")
44 usernameEntry.place(x=screen_width/2, y=screen_height/2+25, anchor="center")
45
46 #password entry
47 passEntry=Entry(bg_label,width=25,font=('Courier'), textvariable=passwor)
48 passEntry.configure(borderwidth=0, relief="flat", foreground="black")
49 passEntry.place(x=screen_width/2, y=screen_height/2+90, anchor="center")
50
51 #gambar button
52 button = PhotoImage(file='./BAGIAN PERTAMA/Button.png')
53
54 #button
55 button1 = tk.Button(bg_label, cursor="hand2", image=button, borderwidth=0, highlightthickness=0, command=login)
56 button1.place(x=screen_width/2 - 100, y=screen_height/2 + 150)
57
58 loginwindow.mainloop()

```

**Gambar 3.1 main.py**

```

opsi.py > ...
1  import tkinter as tk
2  from tkinter import *
3  import json
4  import datetime
5  from tkinter import ttk
6  from PIL import ImageTk, Image
7  from tkinter.messagebox import showinfo
8  from fungsi import *
9
10 #State window
11 with open("data.json", "r") as f:
12     read = json.load(f)
13 def open_input():
14     InputData.tampilan_input(self=NONE)
15
16 def opsi():
17     menuwindow = tk.Tk()
18     menuwindow.title("Menu")
19     screen_width = 1280
20     screen_height = 720
21     menuwindow.geometry("%dx%d" % (screen_width, screen_height))
22     menuwindow.resizable(False, False)
23
24     #gambar menu
25     bg = tk.PhotoImage(file= "./BAGIAN MENU/bg.png")
26     label = Label(menuwindow, image=bg)
27     label.pack(fill="both", expand=True)
28
29     tombol1 = PhotoImage(file= "./BAGIAN MENU/memasukkan.png")
30     tombol2 = PhotoImage(file= "./BAGIAN MENU/tampil.png")
31     tombol3 = PhotoImage(file= "./BAGIAN MENU/hapus.png")
32
33     button1 = tk.Button(label, image=tombol1, cursor="hand2", borderwidth=0, highlightthickness=0, bg="white", command= InputData)
34     button1.place(x=screen_width/2-245, y=screen_height/2.5-100)
35     button2 = tk.Button(label, image=tombol2, cursor="hand2", borderwidth=0, highlightthickness=0, bg="white", command= OutputData)
36     button2.place(x= screen_width/2-245, y=screen_height/2.5+50)
37     button3 = tk.Button(label, image=tombol3, cursor="hand2", borderwidth=0, highlightthickness=0, bg="white", command= DelData)
38     button3.place(x= screen_width/2-245, y=screen_height/2.5+200)
39     menuwindow.mainloop()
40
41 if __name__ == "__main__":
42     opsi()

```

**Gambar 3.2 opsi.py**



```

fungsi.py x
fungsi.py > InputData > __init__
1 import json
2 import datetime
3 import tkinter as tk
4 from tkinter import *
5 from PIL import ImageTk, Image
6 import PIL
7 from tkinter.messagebox import *
8
9 with open("data.json", "r") as f:
10     read = json.load(f)
11
12 class InputData:
13     def __init__(self):
14         input_window = tk.Toplevel()
15         self.nama = tk.StringVar()
16         self.berat_laundry = tk.DoubleVar()
17         self.jenis_laundry = tk.StringVar()
18         self.jenis_pewangi = tk.StringVar()
19         self.alamat = tk.StringVar()
20         self.jarak = tk.DoubleVar()
21         self.jenis_payment = tk.StringVar()
22         self.jenis_bank = tk.StringVar()
23
24         pewangi = ["lavender", "jeruk", "peppermint", "strawberry", "vanilla", "mango"]
25         self.jenis_pewangi.set(pewangi[0])
26         laundry = ["express", "reguler"]
27         self.jenis_laundry.set(laundry[0])
28         payment = ["Cash", "Transfer Bank"]
29         self.jenis_payment.set(payment[0])
30         bank = ["-", "Mandiri", "BCA"]
31         self.jenis_bank.set(bank[0])
32
33         input_window.title("Input Data")
34         xw = 1280
35         yw = 720
36         input_window.geometry("%dx%d" % (xw, yw))
37         input_window.resizable(False, False)
38
39
40         bg = PhotoImage(file='./inputdata/bg.png')
41         label = Label(input_window, image=bg)
42         label.pack()
43
44         nama_entry = Entry(label, width=30, font=('Courier'), bg="grey", textvariable=self.nama)
45         nama_entry.configure(borderwidth=0, relief="solid", foreground="black")
46         nama_entry.place(x=xw/2, y=yw/2-110)
47
48         berat_entry = Entry(label, width=30, font=('Courier'), bg="grey", textvariable=self.berat_laundry, show="")
49         berat_entry.configure(borderwidth=0, relief="solid", foreground="black")
50         berat_entry.place(x=xw/2, y=yw/2-60)
51
52         jenis_pewangi_option = OptionMenu(label, self.jenis_pewangi, *pewangi)
53         jenis_pewangi_option.configure(borderwidth=0, relief="solid", foreground="black", compound='left')
54         jenis_pewangi_option.place(x=xw/2, y=yw/2-10)
55
56         jenis_laundry_option = OptionMenu(label, self.jenis_laundry, *laundry)
57         jenis_laundry_option.configure(borderwidth=0, relief="solid", foreground="black", compound='left')
58         jenis_laundry_option.place(x=xw/2, y=yw/2+40)
59
60         alamat_entry = Entry(label, width=30, font=('Courier'), bg="grey", textvariable=self.alamat)
61         alamat_entry.configure(borderwidth=0, relief="solid", foreground="black")
62         alamat_entry.place(x=xw/2, y=yw/2+80)
63
64         jarak_entry = Entry(label, width=30, font=('Courier'), bg="grey", textvariable=self.jarak, show="")
65         jarak_entry.configure(borderwidth=0, relief="solid", foreground="black")
66         jarak_entry.place(x=xw/2, y=yw/2+130)
67
68         jenis_payment_option = OptionMenu(label, self.jenis_payment, *payment)
69         jenis_payment_option.configure(borderwidth=0, relief="solid", foreground="black", compound='left')
70         jenis_payment_option.place(x=xw/2, y=yw/2+180)
71
72         jenis_bank_option = OptionMenu(label, self.jenis_bank, *bank)
73         jenis_bank_option.configure(borderwidth=0, relief="solid", foreground="black", compound='left')
74         jenis_bank_option.place(x=xw/2, y=yw/2+220)

```



```

74
75 button = PhotoImage(file= "./inputdata/submit.png")
76 submit = Button(label, image=button, bg="white", cursor="hand2", borderwidth=0, highlightthickness=0, command=self.input_data)
77 submit.place(x=xw-250,y=yw-150)
78
79 input_window.mainloop()
80
81 def input_data(self):
82     x = read[['pewangi']]
83
84     jarak = self.jarak.get()
85
86     jenis_pewangi = self.jenispewangi.get()
87     tanggal = datetime.datetime.now()
88     if jarak <= 3:
89         harga_delivery = 0
90     elif jarak > 3:
91         harga_delivery = (jarak-3)*3000
92
93     if self.jenis_laundry.get() == "reguler":
94         waktu_selesai = datetime.datetime.now() + datetime.timedelta(days=3)
95         harga_laundry = 3500
96     else:
97         waktu_selesai = datetime.datetime.now() + datetime.timedelta(hours=12)
98         harga_laundry = 5000
99
100     if not jarak:
101         showerror("Error", "Jarak Tidak Boleh Kosong!")
102
103     koma = self.berat_laundry.get()
104     if not koma:
105         showerror("Error", "Berat Laundry Tidak Boleh Kosong!")
106
107     total_biaya = (jarak * harga_laundry) + x[jenis_pewangi] + harga_delivery
108     nama = self.nama.get()
109     berat = koma,
110     jenis = self.jenis_laundry.get()
111
112     pewangi = self.jenispewangi.get()
113     alamat = self.alamat.get()
114     bank = self.jenis_bank.get()
115     pay = self.jenis_payment.get()
116
117     if not nama:
118         showerror("Error", "Nama tidak boleh kosong")
119     if not berat:
120         showerror("Error", "Berat tidak boleh kosong")
121     if not jenis:
122         showerror("Error", "Jenis laundry tidak boleh kosong")
123     if not pewangi:
124         showerror("Error", "Pewangi tidak boleh kosong")
125     if not alamat:
126         showerror("Error", "Alamat tidak boleh kosong")
127     if not bank:
128         showerror("Error", "Bank tidak boleh kosong")
129     if not pay:
130         showerror("Error", "Jenis pembayaran tidak boleh kosong")
131
132     data_pelanggan = {
133         "nama": self.nama.get(),
134         "berat_laundry": koma,
135         "jenis_laundry": self.jenis_laundry.get(),
136         "jenis_pewangi": self.jenispewangi.get(),
137         "harga_laundry_per_kg": harga_laundry,
138         "harga_pewangi": x[jenis_pewangi],
139         "tanggal_laundry": tanggal.strftime("%Y-%m-%d %H:%M:%S"),
140         "jarak": jarak,
141         "jenis_payment": self.jenis_payment.get(),
142         "jenis_bank": self.jenis_bank.get(),
143         "alamat": self.alamat.get(),
144         "harga_delivery": harga_delivery,
145         "total_biaya": total_biaya,
146         "estimasi_waktu_selesai": waktu_selesai.strftime("%Y-%m-%d %H:%M:%S")
147     }

```

```

148     print(json.dumps(data_pelanggan, indent=4))
149     with open("data.json", 'r') as f:
150         jsongdata = f.read()
151         data = json.loads(jsongdata)
152         data["pelanggan"].append(data_pelanggan)
153         a = json.dumps(data["pelanggan"], indent=4)
154
155     with open("data.json", "w") as f:
156         f.write(json.dumps(data, indent=4))
157     showinfo(title="Input Data", message="Data Berhasil Dimasukkan!")

```

fungsi.py > InputData > \_\_init\_\_

```

203 class OutputData:
204     def __init__(self):
205         self.output_window = tk.Toplevel()
206         self.namaout = tk.StringVar()
207         self.xw = 1280
208         self.yw = 720
209         self.output_window.geometry("%dx%d" % (self.xw, self.yw))
210         self.output_window.title("Output Data")
211         self.output_window.resizable(False, False)
212
213         gambar = tk.PhotoImage(file=("./BAGIAN MENAMPILKAN DATA/bg.png"))
214         self.label = tk.Label(self.output_window, image=gambar)
215         self.label.pack(fill="both", expand=True)
216
217         self.nama_entry = tk.Entry(self.label, width=30, font=('Courier'), bg="grey", textvariable=self.namaout)
218         self.nama_entry.configure(borderwidth=0, relief="solid", foreground="black")
219         self.nama_entry.place(x=self.xw/2-100,y=self.yw/4-10)
220
221         button = tk.PhotoImage(file=("./BAGIAN MENAMPILKAN DATA/button.png"))
222         submit = tk.Button(self.label, image=button,bg="white", cursor="hand2", borderwidth=0, highlightthickness=0, command=self.output)
223         submit.place(x=self.xw/2-50,y=self.yw/4+50)
224
225         self.output_window.mainloop()
226
227     def output(self):
228         with open("data.json", "r") as f:
229             data = json.load(f)
230             found = False
231
232             for pelanggan in data["pelanggan"]:
233                 if pelanggan["nama"] == self.namaout.get():
234                     namap = pelanggan["nama"]
235                     tanggal = pelanggan["tanggal_laundry"]
236                     berat = pelanggan["berat_laundry"]
237                     jenis = pelanggan["jenis_laundry"]
238                     pewangi = pelanggan["jenis_pewangi"]
239                     hargalaundry = pelanggan["harga_laundry_per_kg"]
240                     hargapewa = pelanggan["harga_pewangi"]
241                     pembayaran = pelanggan["jenis_payment"]
242                     bank = pelanggan["jenis_bank"]
243                     total = pelanggan["total_biaya"]
244                     estimasi = pelanggan["estimasi_waktu_selesai"]
245                     found = True
246                     break
247                 else:
248                     if not found:
249                         showinfo(title='Output Data', message='Data Tidak Ditemukan!')
250
251             frame = tk.Frame(self.label)
252             frame.place(x= self.xw/2-250, y= self.yw/2-30)
253
254             nama_label = tk.Label(frame, text="Nama: " + namap, justify='left', font="Courier")
255             nama_label.pack(anchor='w')
256
257             tanggal_label = tk.Label(frame, text="Tanggal: " + tanggal, justify='left', font="Courier")
258             tanggal_label.pack(anchor='w')

```

```

218 berat_label = tk.Label(frame, text="Berat: " + str(berat), justify='left', font="Courier")
219 berat_label.pack(anchor='w')
220
221 jenis_label = tk.Label(frame, text="Jenis: " + jenis, justify='left', font="Courier")
222 jenis_label.pack(anchor='w')
223
224 pewangi_label = tk.Label(frame, text="Pewangi: " + pewangi, justify='left', font="Courier")
225 pewangi_label.pack(anchor='w')
226
227 harga_laundry_label = tk.Label(frame, text="Harga Laundry per Kg: Rp" + str(hargalaundry), justify='left', font="Courier")
228 harga_laundry_label.pack(anchor='w')
229
230 harga_pewangi_label = tk.Label(frame, text="Harga Pewangi: Rp" + str(hargapewa), justify='left', font="Courier")
231 harga_pewangi_label.pack(anchor='w')
232
233 alamat_label = tk.Label(frame, text="Alamat: " + alamat, justify='left', font="Courier")
234 alamat_label.pack(anchor='w')
235
236 pembayaran_label = tk.Label(frame, text="Jenis Pembayaran: " + pembayaran, justify='left', font="Courier")
237 pembayaran_label.pack(anchor='w')
238
239 bank_label = tk.Label(frame, text="Jenis Bank: " + bank, justify='left', font="Courier")
240 bank_label.pack(anchor='w')
241
242 total_label = tk.Label(frame, text="Total Biaya: Rp" + str(total), justify='left', font="Courier")
243 total_label.pack(anchor='w')
244
245 estimasi_label = tk.Label(frame, text="Estimasi Waktu Selesai: " + estimasi, justify='left', font="Courier")
246 estimasi_label.pack(anchor='w')
247
248
249 self.output_window.update()
250
251 class DelData:
252     def __init__(self):
253         self.del_window = tk.Toplevel()
254         self.namaout = tk.StringVar()
255         xw = 1280
256         yw = 720
257         self.del_window.geometry("%dx%d" % (xw, yw))
258         self.del_window.title("Input Data")
259         self.del_window.config(bg="white")
260         self.del_window.resizable(False, False)
261
262         bg = tk.PhotoImage(file="./BAGIAN MENGHAPUS DATA/bg.png")
263         self.label = tk.Label(self.del_window, image=bg)
264         self.label.pack(fill="both", expand=True)
265
266         self.nama_entry = tk.Entry(self.label, width=30, font=('Courier'), bg="grey", textvariable=self.namaout)
267         self.nama_entry.configure(borderwidth=0, relief="solid", foreground="black")
268         self.nama_entry.place(x=xw/2-100, y=yw/4+25)
269
270         button = tk.PhotoImage(file="./BAGIAN MENAMPILKAN DATA/button.png")
271         submit = tk.Button(self.label, image=button, bg="white", cursor="hand2", borderwidth=0, highlightthickness=0, command=self.delete)
272         submit.place(x=xw/2-50, y=yw/4+90)
273
274         self.del_window.mainloop()
275
276
277 def delete(self):
278     with open("data.json", "r") as f:
279         data = json.load(f)
280         found = False
281         for pelanggan in data["pelanggan"]:
282             if pelanggan["nama"] == self.namaout.get():
283                 data["pelanggan"].remove(pelanggan)
284                 a = json.dumps(data["pelanggan"], indent=4)
285                 with open("data.json", "w") as f:
286                     f.write(json.dumps(data, indent=4))
287                 showinfo(title='Delete Data', message='Data Berhasil Dihapus!')
288                 found = True
289             else:
290                 if not found:
291                     showinfo(title='Delete Data', message='Data Tidak Ditemukan!')

```

**Gambar 3.3 fungsi.py**

```
{} data.json M ×
{} data.json > {} admin
1 {
2   "admin": {
3     "username": "admin",
4     "password": "dani"
5   },
6   "pelanggan": [],
7   "pewangi": {
8     "lavender": 1000,
9     "jeruk": 1500,
10    "peppermint": 2000,
11    "strawberry": 1800,
12    "vanilla": 1200,
13    "mango": 1700
14  }
15 }
```

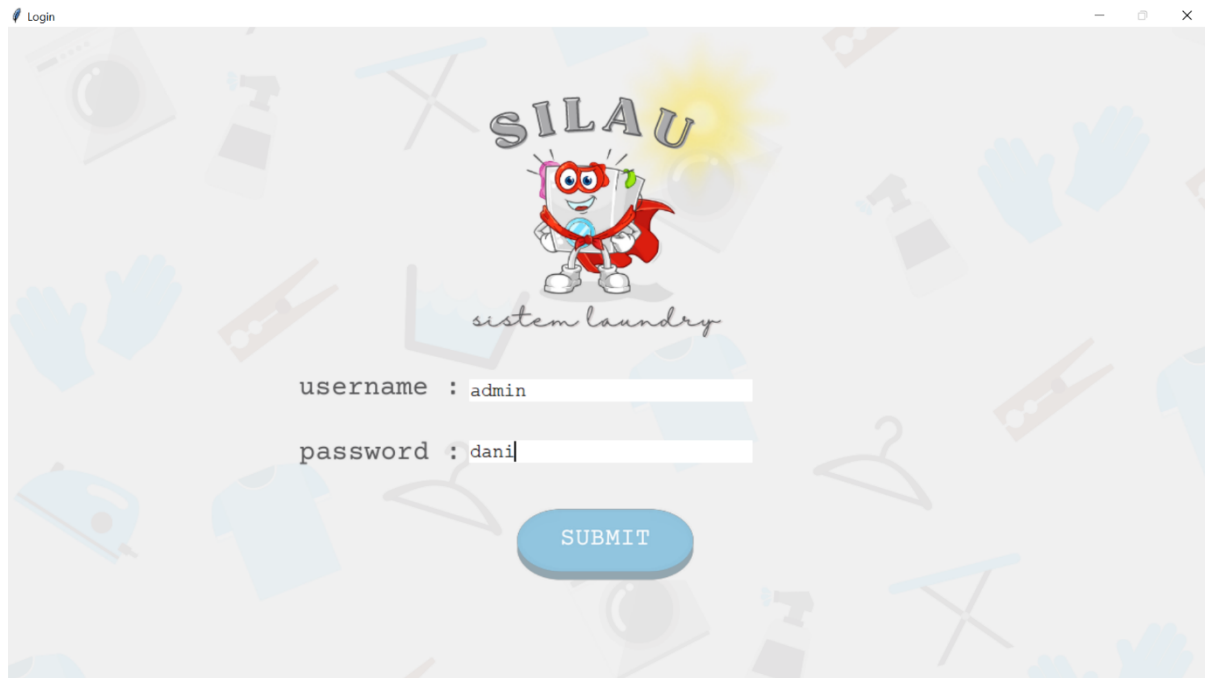
Gambar 3.4 data.json

```
code > {} settings.json > ...
1 {
2   "[python]": {
3     "editor.defaultFormatter": "ms-python.black-formatter"
4   },
5   "python.formatting.provider": "none"
6 }
```

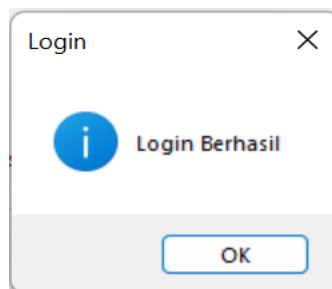
Gambar 3.5 settings.json

## BAB IV

### HASIL *RUNNING* PROGRAM



**Gambar 4.1 Tampilan Awal Program**



**Gambar 4.2 Tampilan Berhasil Login**



**Gambar 4.3 Tampilan Menu Program**

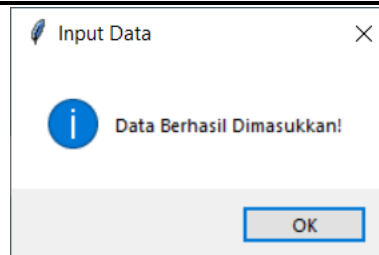
Input Data

## MEMASUKKAN DATA

NAMA	:	dafa
BERAT LAUNDRY	:	4
JENIS PEWANGI	:	lavender
JENIS LAUNDRY	:	express
ALAMAT	:	solo
JARAK	:	4
JENIS PEMBAYARAN	:	Transfer Bank
		Mandiri

SUBMIT

**Gambar 4.4 Tampilan Program Memasukkan Data**

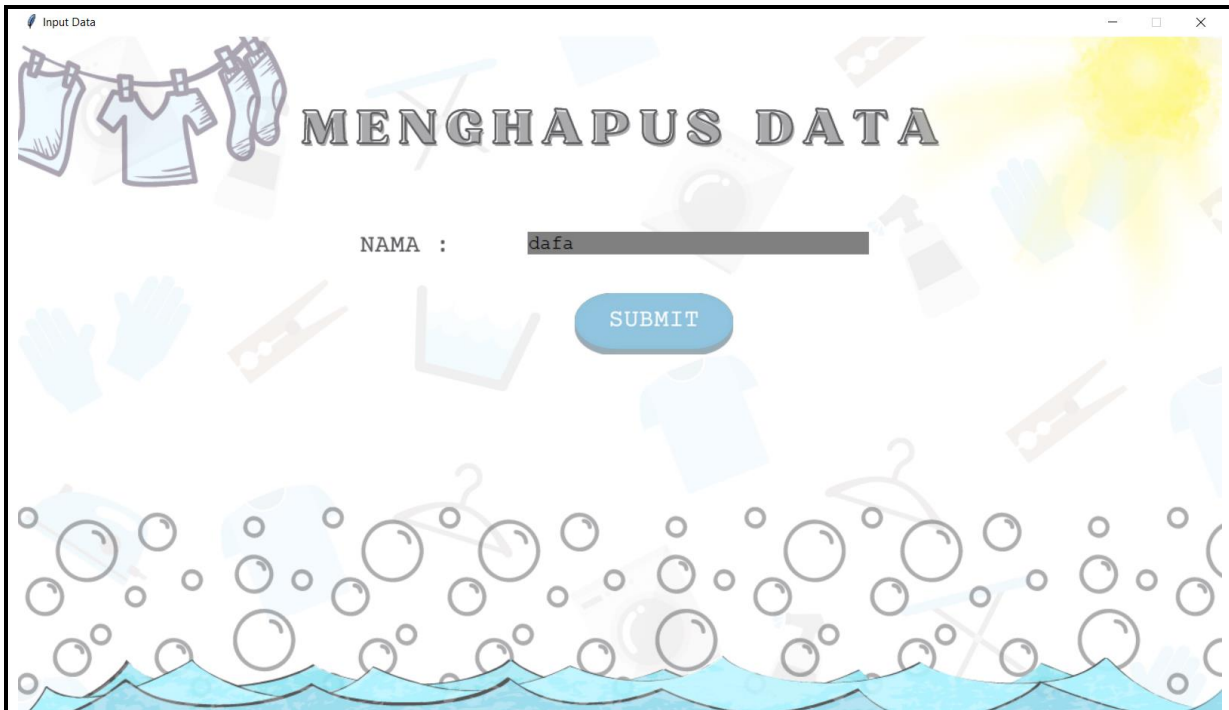


**Gambar 4.5 Tampilan Data Berhasil Dimasukkan**



**Gambar 4.6 Tampilan Program Menampilkan Data**





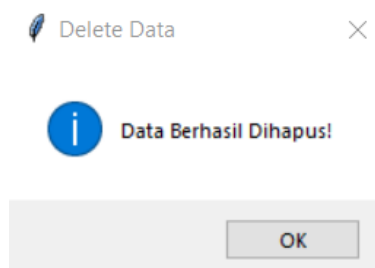
Input Data

## MENGHAPUS DATA

NAMA :

SUBMIT

**Gambar 4.7 Tampilan Program Menghapus Data**



**Gambar 4.8 Tampilan Data Berhasil Dihapus**