

Aplikasi Laundry Berbasis Desktop

LAPORAN

Diajukan Untuk Memenuhi Syarat Mengikuti Ujian Kompetensi Keahlian

Program Keahlian : Teknik Komputer dan Informatika

Kompetensi Keahlian : Rekayasa Perangkat Lunak



Disusun oleh :

Raditya Kanahaya

202118553

XII RPL 1



**PEMERINTAH DAERAH PROVINSI JAWA BARAT
DINAS PENDIDIKAN
SMK NEGERI 4 BANDUNG
TAHUN PELAJARAN 2022/2023**

Kata Pengantar

Puji syukur penulis panjatkan kehadiran Allah SWT, yang telah memberikan karunia nikmat dan kesehatan, sehingga kami dapat menyelesaikan laporan ini dapat diselesaikan dengan baik.

Penulisan laporan ini merupakan sebuah tugas untuk mengikuti Ujian Kompetensi Keahlian. Adapun tujuan penulisan laporan ini adalah untuk memberikan pemahaman tentang alur kerja aplikasi laundry pada pihak-pihak yang terkait. Kami ucapkan terima kasih kepada semua pihak yang telah membantu dalam penyusunan laporan ini.

Karena kebaikan semua pihak yang telah kami sebutkan tadi maka kami bisa menyelesaikan laporan hasil UKK ini dengan sebaik-baiknya. Laporan ini memang masih jauh dari kesempurnaan, sehingga kami mengharapkan kritik dan saran yang bersifat membangun demi penyempurnaan laporan ini. Sekali lagi terima kasih. Semoga laporan ini bermanfaat bagi kita semua.

Daftar Isi

Kata Pengantar	i
Daftar Isi	ii
A. Deskripsi Awal Perancangan Aplikasi.....	1
B. Perancangan Sistem Aplikasi.....	2
Entity Relationship Diagram (ERD)	2
Rancangan Tabel Relasional.....	2
C. Perancangan Pengembangan Software.....	3
UML	3
D. Perancangan User Interface.....	6
Wireframe.....	6
Mockup	8
E. Implementasi Aplikasi	11
Dokumentasi Front-End.....	11
Dokumentasi Back-End.....	13
Dokumentasi dan Manual Singkat Penggunaan Aplikasi.....	23
Pengembangan Aplikasi	27

A. Deskripsi Awal Perancangan Aplikasi

Aplikasi laundry dalam platform desktop merupakan sebuah program komputer yang dirancang untuk membantu pengelolaan bisnis laundry. Aplikasi ini akan memungkinkan pengguna untuk mengelola berbagai aspek bisnis laundry, seperti pelanggan, outlet, dan transaksi.

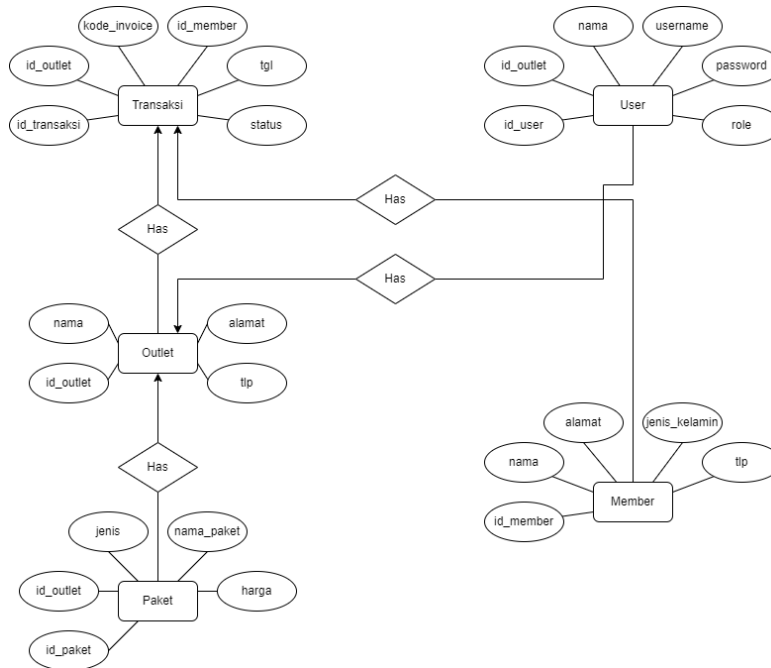
Aplikasi laundry dalam platform desktop akan memiliki beberapa fitur utama, seperti sistem manajemen pelanggan yang memungkinkan pengguna untuk mengelola daftar pelanggan dan riwayat transaksi pelanggan.

Untuk mempermudah pengguna dalam mengelola aplikasi, aplikasi laundry dalam platform desktop akan memiliki antarmuka pengguna yang intuitif dan mudah dipahami. Dengan aplikasi laundry dalam platform desktop, pengguna dapat menghemat waktu dan tenaga dalam mengelola bisnis laundry mereka.

B. Perancangan Sistem Aplikasi

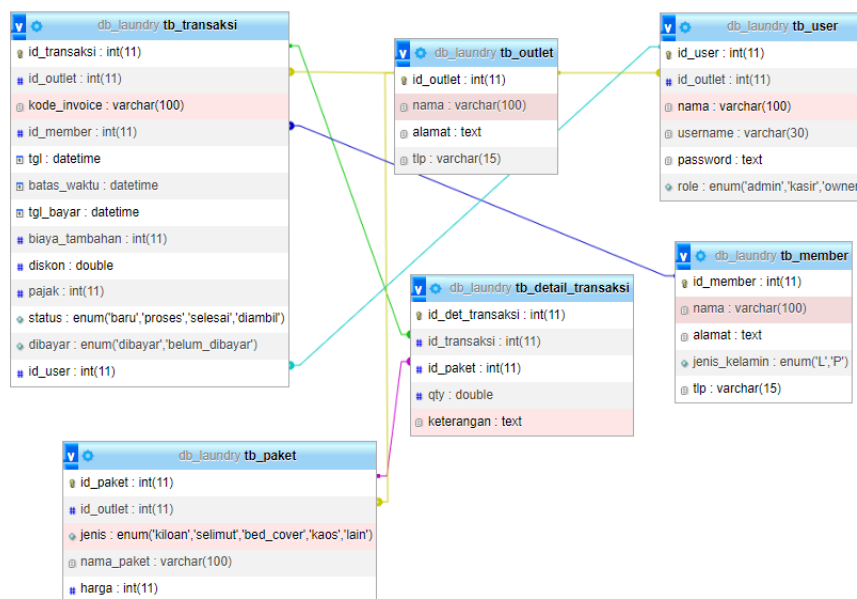
Entity Relationship Diagram (ERD)

ERD Aplikasi :



Rancangan Tabel Relasional

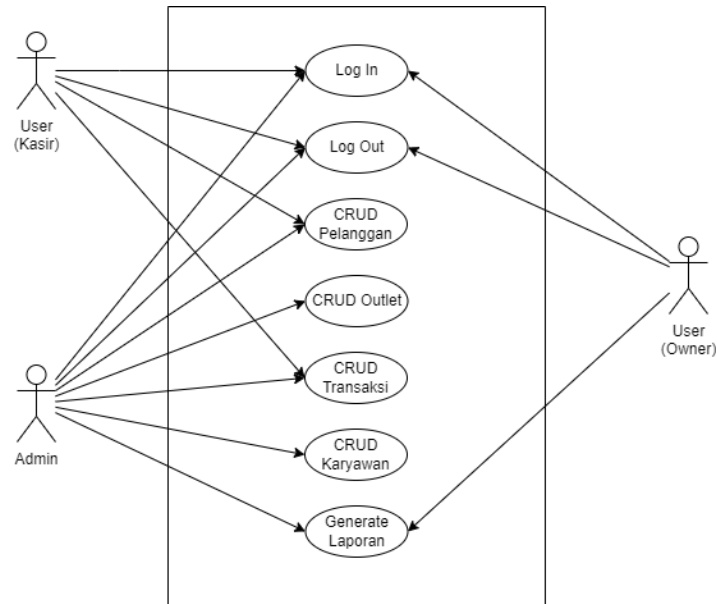
Rancangan Relasi Tabel Aplikasi :



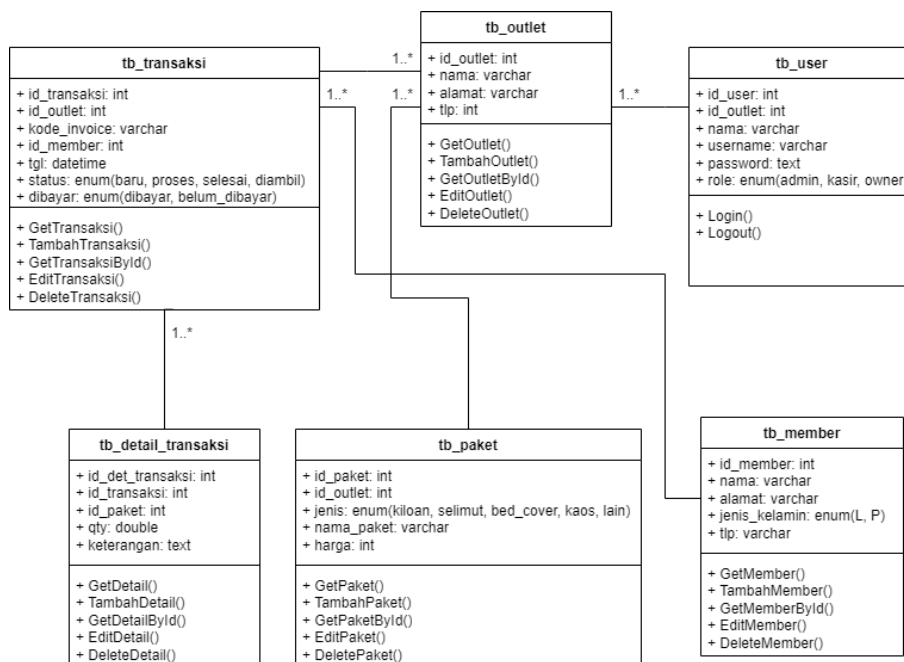
C. Perancangan Pengembangan Software

UML

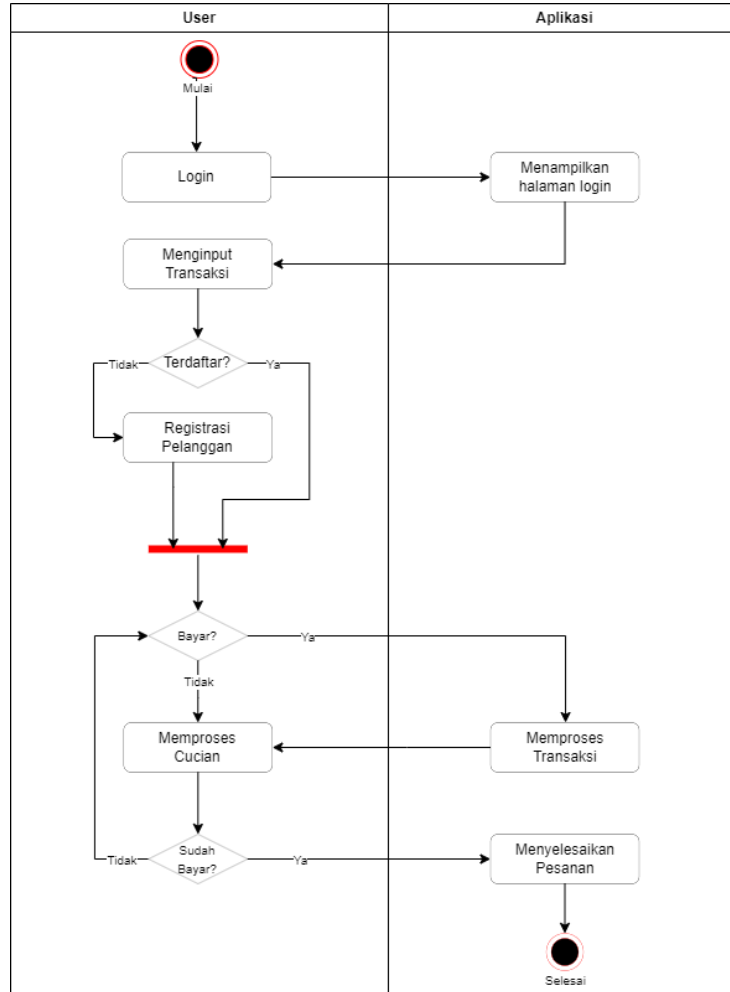
Use Case Aplikasi :

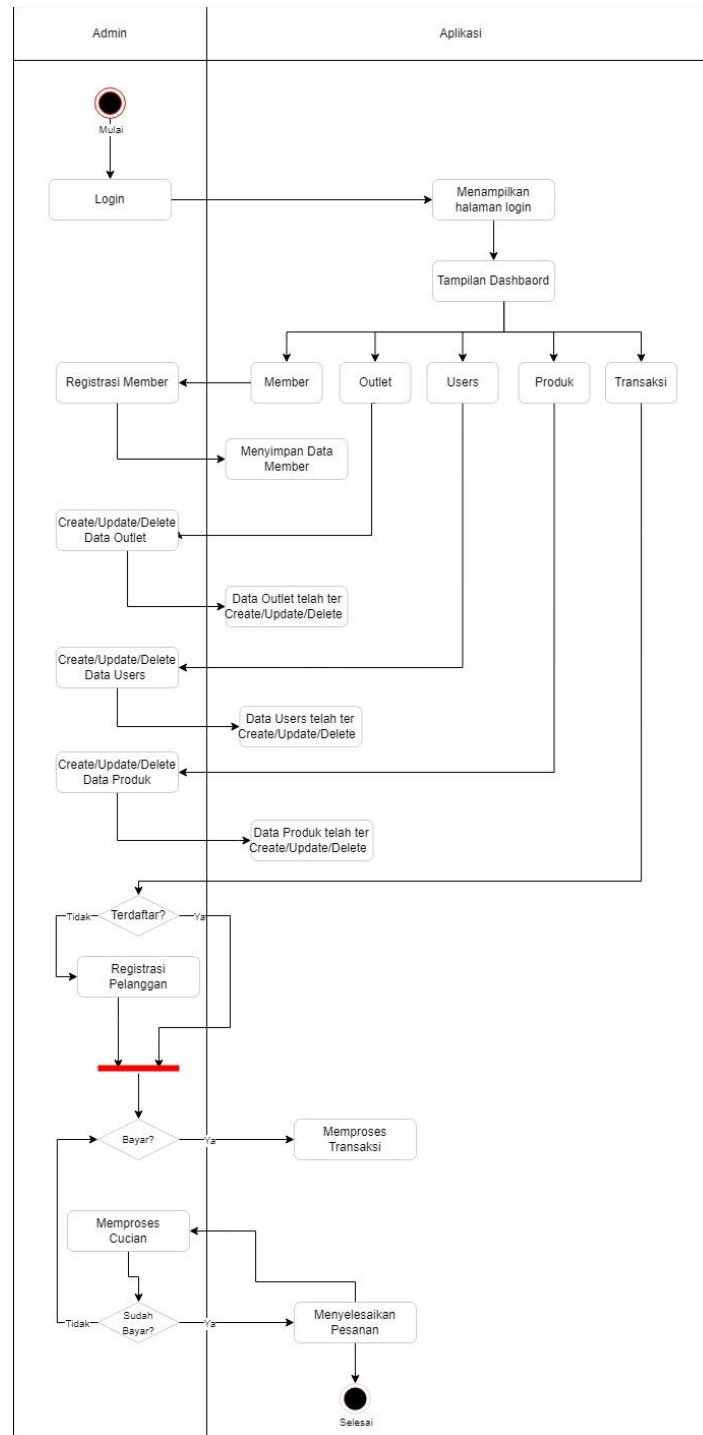


Class Diagram Aplikasi :



Activity Diagram Aplikasi :





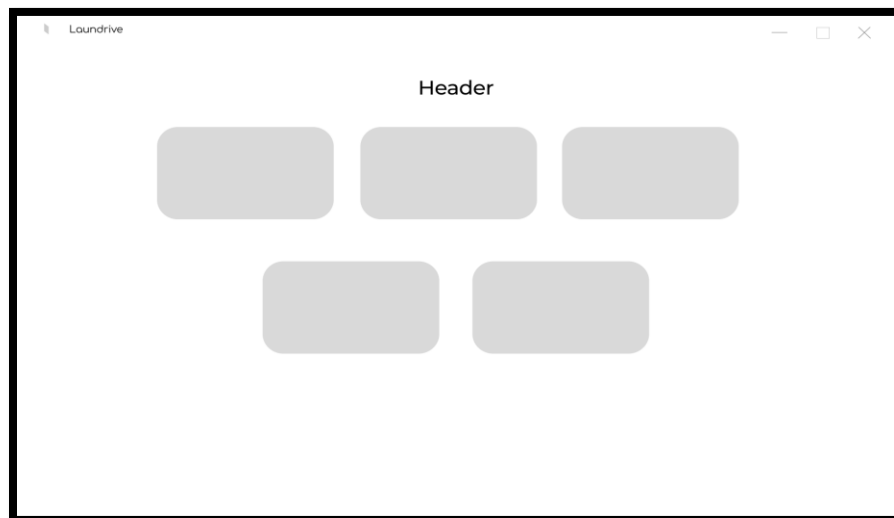
D. Perancangan User Interface

Wireframe



A wireframe of a sign-in form. The form is titled "Sign in" and contains two input fields labeled "Username" and "Password". Below the input fields is a "Login" button. The form is centered within a light gray rectangular frame, which is itself centered within a larger black rectangular frame.

Login



A wireframe of a main dashboard. The dashboard has a header bar at the top with the text "Loundrive" on the left and window control icons (minimize, maximize, close) on the right. Below the header, the word "Header" is centered. The main content area contains five gray rounded rectangular boxes: three in the top row and two in the bottom row, arranged in a grid-like fashion.

Main Dashboard

Laundrive

Header

Tambah Data

Edit Data

Detail Data

Table Data

Laundrive

Header

Field

Field

Field

Field

Field

Submit

Form

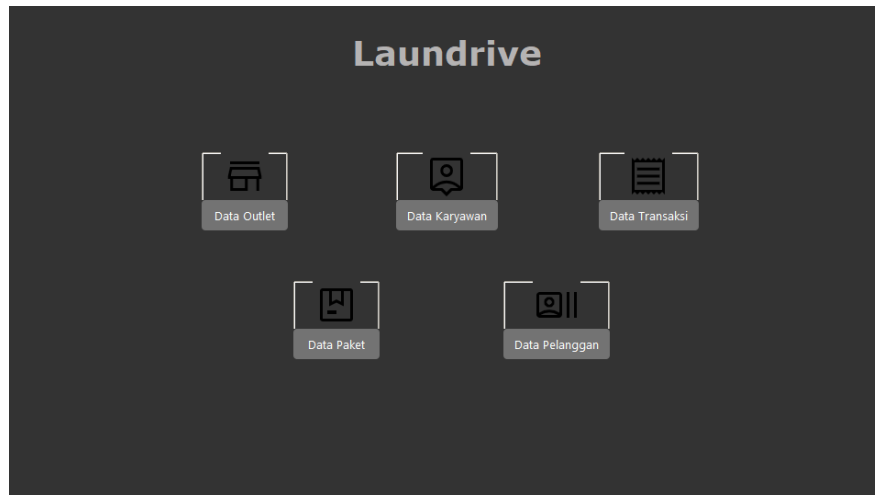
A mockup of a web form titled "Header". The form is contained within a window with a title bar that says "Loundrive". Inside the window, the word "Header" is centered at the top. Below it, there are four lines of text, each consisting of the word "Field" followed by a colon and another "Field". At the bottom center of the form is a button labeled "Submit".

Detail

Mockup

A mockup of a login form titled "Login". The form is centered on a dark gray background. It features the word "Login" in a large, bold, white font. Below the title, there are two input fields. The first is labeled "Username" in white text, and the second is labeled "Password" in white text. Both fields have a light gray border and a small "Username" or "Password" label inside the top right corner. Below the input fields is a button labeled "Login" in white text on a dark gray background.

Login



Main Dashboard

Data Transaksi									
Export as CSV		Tambah Data		Edit Data		Detail Data			
ID	Kode Invoice	Outlet	Karyawan	Pelanggan	Tanggal	Batas Waktu	Waktu Bayar	Status	Dibayar
46	L-46-2023-03-22	UKK	Rei	Kita	2023-03-22	2023-03-22	2023-03-22	baru	dibayar

Table Data

Tambah Transaksi

Outlet	<input type="text"/>
Karyawan	<input type="text"/>
Pelanggan	<input type="text"/>
Tanggal	<input type="text" value="3/23/23"/>
Batas Waktu	<input type="text" value="3/23/23"/>
Waktu Bayar	<input type="text" value="3/23/23"/>
Biaya Tambah	<input type="text"/>
Diskon	<input type="text"/>
Status	<input type="text"/>
Dibayar	<input type="text"/>

Form

L-46-2023-03-22

Id Paket : 2
Kuantitas : 1
Keterangan : Kilos
Total Harga : Rp. 9901

Detail / Invoice

E. Implementasi Aplikasi

Dokumentasi Front-End

```
1  # Copyright © 2021 rdbende <rdbende@gmail.com>
2
3  source [file join [file dirname [info script]] theme light.tcl]
4  source [file join [file dirname [info script]] theme dark.tcl]
5
6  option add *tearOff 0
7
8  proc set_theme {mode} {
9      if {$mode == "dark"} {
10         ttk::style theme use "azure-dark"
11
12         array set colors {
13             -fg          "#ffffff"
14             -bg          "#333333"
15             -disabledfg  "#ffffff"
16             -disabledbg  "#737373"
17             -selectfg    "#ffffff"
18             -selectbg    "#007fff"
19         }
20
21         ttk::style configure . \
22             -background $colors(-bg) \
23             -foreground $colors(-fg) \
24             -troughcolor $colors(-bg) \
```

Tema

```
9  def backgroundimg(self):
10     # Image
11     bg = os.getenv("bg_path")
12     bg = f"{bg}"
13     self.image = PhotoImage(file=bg)
14
15  def btnimg(self):
16     ot = os.getenv("ic_outlet")
17     self.otimg = PhotoImage(file=ot)
18     pkt = os.getenv("ic_paket")
19     self.pkting = PhotoImage(file=pkt)
20     kar = os.getenv("ic_karyawan")
21     self.karimg = PhotoImage(file=kar)
22     tr = os.getenv("ic_transaksi")
23     self.trimg = PhotoImage(file=tr)
24     plg = os.getenv("ic_pelanggan")
25     self.plgimg = PhotoImage(file=plg)
26
```

Image Untuk Background dan Display Text

```

43 def create_card(
44     self,
45     x: int,
46     y: int,
47     width: int = 150,
48     height: int = 200,
49 ) -> tk.Frame:
50     card_frame = tk.Frame(
51         self.main,
52         width=width,
53         height=height,
54         bd=2,
55         relief="groove",
56         highlightthickness=2,
57         highlightbackground="black"
58     )
59     self.canvas.create_window(x, y, anchor='n', window=card_frame)
60     return card_frame

```

Card

```

62 def create_button(
63     self,
64     frame: tk.Frame,
65     image: PhotoImage,
66     text: str,
67     command: callable,
68     font_size: int = 10,
69     font_weight: str = 'bold',
70     width: int = 20,
71     height: int = 2,
72 ):
73     label = tk.Label(frame, image=image)
74     button = tk.Button(
75         frame,
76         text=text,
77         command=command,
78         font=font.Font(size=font_size, weight=font_weight),
79         width=width,
80         height=height,
81     )
82     label.pack()
83     button.pack()
84

```

Button dan Label

Dokumentasi Back-End

```
27 def database(self):
28     # Connect to the database
29     self.db_host = os.getenv("host")
30     self.db_user = os.getenv("user")
31     self.db_password = os.getenv("password")
32     self.db_database = os.getenv("database")
33     self.db = mysql.connector.connect(
34         host=self.db_host,
35         user=self.db_user,
36         password=self.db_password,
37         database=self.db_database
38     )
39
40     # Create Cursor
41     self.cursor = self.db.cursor()
```

Koneksi ke Database

```
77 def create_crud_button(
78     self,
79     frame: ttk.Frame,
80     x: int,
81     y: int,
82     text: str,
83     command: callable,
84     disabled: bool = False,
85     font_size: int = 14,
86     font_weight: str = 'bold',
87 ):
88     self.button = ttk.Button(
89         frame,
90         text=text,
91         command=command,
92     )
93     self.button["state"] = "disabled" if disabled else "normal"
94
95     self.canvas.create_window(x, y, window=self.button, anchor='center')
96     return self.button
97
98 def switch(buttons: list, selection: Tuple):
99     for button in buttons:
100         button["state"] = "normal" if button["state"] == "disabled" else "normal"
101
```

Buat Button CRUD


```

102 def create_laporan_button(
103     self,
104     frame: ttk.Frame,
105     x: int,
106     y: int,
107     text: str,
108     command: callable,
109     font_size: int = 10,
110     font_weight: str = 'bold',
111 ):
112     self.button = ttk.Button(
113         frame,
114         text=text,
115         command=command,
116     )
117
118     self.canvas.create_window(x, y, window=self.button, anchor='center')
119     return self.button

```

Buat Button Generate Laporan

```

121 def create_treeview(
122     self,
123     frame: ttk.Frame,
124     x: int,
125     y: int,
126     proc: str,
127     columns: tuple,
128     headings: tuple,
129     texts: tuple,
130     width: int = 90,
131     minwidth: int = 90,
132 ):
133     self.treeview = ttk.Treeview(frame)
134     self.treeview.pack(padx=20, pady=20)
135     self.treeview['columns'] = columns
136
137     self.treeview.column("#0", width=0, stretch=False)
138     for each in columns:
139         self.treeview.column(each, anchor='center', width=width, minwidth=minwidth, stretch=True)
140
141     self.treeview.heading("#0", text="", anchor="center")
142     for heading, text in zip(headings, texts):
143         self.treeview.heading(heading, text=text, anchor='center')
144
145     self.cursor.callproc(proc)
146     result = self.cursor.stored_results()
147
148     for each in result:
149         i = each.fetchall()
150         for row in i:
151             self.treeview.insert(parent='', index='end', values=row)
152
153     self.canvas.create_window(x, y, anchor='n', window=self.treeview)
154     return self.treeview

```

Buat Tabel Data

```

156 def create_tambah_entry(
157     self,
158     frame: ttk.Frame,
159     x: int,
160     y: int,
161 ):
162     entry = ttk.Entry(frame)
163     entry.pack()
164     self.canvas.create_window(x, y, window=entry)
165     return entry

```

Buat Entry Tambah Data

```

167 def create_edit_entry(
168     self,
169     frame: ttk.Frame,
170     x: int,
171     y: int,
172     index: int,
173     state: str,
174     treeview: ttk.Treeview,
175     procid: str,
176 ):
177     editing = treeview.selection()[0]
178     values = treeview.item(editing, 'values')
179
180     self.cursor.callproc(procid, (values[0],))
181     result = self.cursor.stored_results()
182
183     data = None
184     for each in result:
185         # print(each)
186         data = each.fetchone()
187
188     entry = ttk.Entry(frame, state=state)
189     entry.insert(0, data[index])
190     entry.pack()
191     self.canvas.create_window(x, y, window=entry)
192     return entry

```

Buat Entry Edit Data

```

195 def create_tambah_dropdown(
196     self,
197     frame: tk.Frame,
198     x: int,
199     y: int,
200     procdrop: str,
201 ):
202     self.cursor.callproc(procdrop)
203     result = self.cursor.stored_results()
204
205     for values in result:
206         values = values.fetchall()
207
208     dropdown = ttk.Combobox(frame, values=values, state="readonly")
209     dropdown.configure(width=17)
210     dropdown.pack()
211     self.canvas.create_window(x, y, window=dropdown)
212     return dropdown

```

Buat Dropdown Berdasarkan Value Database

```

213 def create_edit_dropdown(
214     self,
215     frame: ttk.Frame,
216     x: int,
217     y: int,
218     index: int,
219     target_index: int,
220     state: str,
221     treeview: ttk.Treeview,
222     procid: str,
223     procdrop: str,
224 ):
225     editing = treeview.selection()[0]
226     values = treeview.item(editing, 'values')
227     self.cursor.callproc(procid, (values[0],))
228     result = self.cursor.stored_results()
229
230     data = None
231     for each in result:
232         data = each.fetchone()
233
234     self.cursor.callproc(procdrop)
235     drop = self.cursor.stored_results()
236
237     for values in drop:
238         values = values.fetchall()
239
240     selected = None
241     for each in values:
242         # print(each, data)
243         if data[index] == each[target_index]:
244             selected = each
245             break
246
247     dropdown = ttk.Combobox(frame, values=values, state="readonly")
248     dropdown.configure(width=17, state=state)
249     dropdown.current(values.index(selected))
250     dropdown.pack()
251     self.canvas.create_window(x, y, window=dropdown)
252     return dropdown

```

Buat Dropdown Edit Berdasarkan Value Database

```

254 def create_tambah_enumdropdown(
255     self,
256     frame: tk.Frame,
257     x: int,
258     y: int,
259     procenum: str,
260 ):
261     self.cursor.callproc(procenum)
262     result = self.cursor.stored_results()
263
264     for value in result:
265         values = value.fetchone()[1][5:-1].split(',')
266         values = [i.replace('"', '') for i in values]
267
268     dropdown = ttk.Combobox(frame, values=values, state="readonly")
269     dropdown.configure(width=17)
270     dropdown.pack()
271     self.canvas.create_window(x, y, window=dropdown)
272     return dropdown

```

Buat Dropdown Berdasarkan Value Enum Database

```

274 def create_edit_enumdropdown(
275     self,
276     frame: ttk.Frame,
277     x: int,
278     y: int,
279     index: int,
280     state: str,
281     treeview: ttk.Treeview,
282     procid: str,
283     procenum: str,
284 ):
285     editing = treeview.selection()[0]
286     values = treeview.item(editing, 'values')
287     self.cursor.callproc(procid, (values[0],))
288     result = self.cursor.stored_results()
289
290     data = None
291     for each in result:
292         data = each.fetchone()
293
294     self.cursor.callproc(procenum)
295     drop = self.cursor.stored_results()
296
297     for value in drop:
298         values = value.fetchone()[1][5:-1].split(',')
299         values = [i.replace('"', '') for i in values]
300
301     dropdown = ttk.Combobox(frame, values=values, state="readonly")
302     dropdown.configure(width=17, state=state)
303     dropdown.current(values.index(data[index]))
304     dropdown.pack()
305     self.canvas.create_window(x, y, window=dropdown)
306     return dropdown

```

Buat Dropdown Edit Berdasarkan Value Enum Database

```

308 def create_tambah_date(
309     self,
310     frame: ttk.Frame,
311     x: int,
312     y: int,
313     width: int = 12,
314     background: str = 'darkblue',
315     foreground: str = 'white',
316     borderwidth: int = 2,
317 ):
318     tanggal = DateEntry(frame, width=width, background=background, foreground=foreground, borderwidth=borderwidth)
319     tanggal.configure(width=17)
320     tanggal.pack()
321     self.canvas.create_window(x, y, window=tanggal)
322     return tanggal

```

Buat Entry Tambah Tanggal

```

324 def create_edit_date(
325     self,
326     frame: ttk.Frame,
327     x: int,
328     y: int,
329     index: int,
330     state: str,
331     treeview: ttk.Treeview,
332     procid: str,
333     width: int = 12,
334     background: str = 'darkblue',
335     foreground: str = 'white',
336     borderwidth: int = 2,
337 ):
338     editing = treeview.selection()[0]
339     values = treeview.item(editing, 'values')
340     self.cursor.callproc(procid, (values[0],))
341     result = self.cursor.stored_results()
342
343     data = None
344     for each in result:
345         data = each.fetchone()
346
347     tanggal = DateEntry(frame, width=width, background=background, foreground=foreground, borderwidth=borderwidth)
348     tanggal.configure(width=17, state=state)
349     tanggal.set_date(data[index])
350     tanggal.pack()
351     self.canvas.create_window(x, y, window=tanggal)
352     return tanggal

```

Buat Entry Edit Tanggal

```

354 def create_submit_button(
355     self,
356     frame: ttk.Frame,
357     x: int,
358     y: int,
359     command: callable,
360     text: str = 'Submit',
361     font_size: int = 10,
362     font_weight: str = 'bold',
363 ):
364     self.button = ttk.Button(frame, text=text, command=command)
365     self.canvas.create_window(x, y, window=self.button, anchor="center")
366
367 def validate_number(
368     values: Tuple,
369 ) -> bool:
370     for each in values:
371         if each.isdigit():
372             return True
373         else:
374             return False

```

Buat Button Submit Dan Validasi Angka Untuk Entry

```

376 def get_id(
377     self,
378     treeview: ttk.Treeview,
379     procid: str
380 ):
381     selecting = treeview.selection()[0]
382     values = treeview.item(selecting, 'values')
383
384     self.cursor.callproc(procid, (values[0],))
385     result = self.cursor.stored_results()
386
387     data = None
388     for each in result:
389         data = each.fetchall()[0]
390
391     return data[0]
392

```

Function Mengambil ID Suatu Baris Tabel

```

393 def tambah(
394     self,
395     frame: ttk.Frame,
396     destroy: tk.Toplevel,
397     redirect: callable,
398     entries: Tuple,
399     proc: str,
400 ):
401     values = entries
402     self.cursor.callproc(proc, values)
403     self.db.commit()
404     self.cursor.close()
405     frame.destroy()
406     destroy.destroy()
407     redirect()
408
409 def edit(
410     self,
411     frame: ttk.Frame,
412     destroy: tk.Toplevel,
413     redirect: callable,
414     entries: Tuple,
415     proccedit: str
416 ):
417     values = entries
418     self.cursor.callproc(proccedit, values)
419     self.db.commit()
420     frame.destroy()
421     destroy.destroy()
422     redirect()

```

Function Tambah dan Edit

```

424 def delete(
425     self,
426     treeview: ttk.Treeview,
427     proc: str,
428 ):
429     deleting = treeview.selection()[0]
430     values = treeview.item(deleting, 'values')
431
432     self.cursor.callproc(proc, (values[0],))
433     self.db.commit()
434     treeview.delete(deleting)

```

Function Delete


```

436 def importcsv(
437     filename: str,
438     treeview: ttk.Treeview,
439 ):
440     with open(filename, 'w', newline='') as csvfile:
441         writer = csv.writer(csvfile)
442         # Write header row
443         header = []
444         for col in treeview["columns"]:
445             header.append(treeview.heading(col)["text"])
446         writer.writerow(header)
447         # Write data rows
448         for item in treeview.get_children():
449             row = []
450             for col in treeview["columns"]:
451                 row.append(treeview.set(item, col))
452             writer.writerow(row)

```

Function Export ke CSV

```

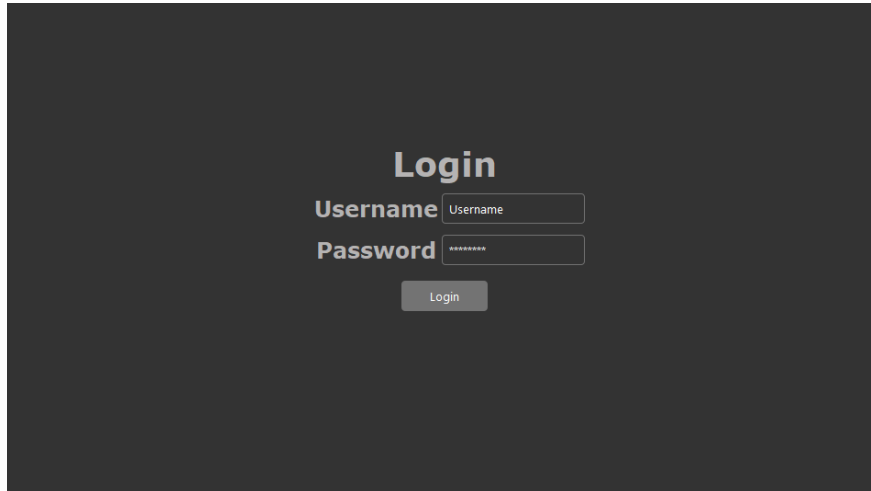
505 def exportxls(
506     filename: str,
507     treeview: ttk.Treeview,
508 ):
509     # Get the data from the Treeview
510     data = []
511     for item in treeview.get_children():
512         values = treeview.item(item, 'values')
513         data.append(values)
514
515     # Convert the data to a Pandas DataFrame
516     df = pd.DataFrame(data, columns=[col for col in treeview['columns']])
517
518     # Write the DataFrame to an Excel file
519     df.to_excel(filename, index=False)

```

Function Export ke Excel

Dokumentasi dan Manual Singkat Penggunaan Aplikasi

1. User Login menggunakan akun yang sudah ada didalam Database



The login screen features a dark gray background. At the top center, the word "Login" is displayed in a bold, white, sans-serif font. Below it, the labels "Username" and "Password" are shown in a smaller white font. Each label is followed by a white rectangular input field. The "Username" field contains the placeholder text "Username", and the "Password" field contains "password". Centered below the input fields is a gray rectangular button with the word "Login" in white text.

2. Pilih bagian Data Transaksi (sebagai contoh)



2.4 Tambah Data Transaksi

Data Transaksi									
Export as CSV		Tambah Data		Edit Data		Detail Data			
ID	Kode Invoice	Outlet	Karyawan	Pelanggan	Tanggal	Batas Waktu	Waktu Bayar	Status	Dibayar
46	L-46-2023-03-22	UKK	Rei	Kita	2023-03-22	2023-03-22	2023-03-22	baru	dibayar

2.2 Isi Data Transaksi

Tambah Transaksi	
Outlet	1 UKK
Karyawan	1 Rei
Pelanggan	1 Kita
Tanggal	3/24/23
Batas Waktu	3/24/23
Waktu Bayar	3/24/23
Biaya Tambah	1
Diskon	1
Status	Baru
Dibayar	Dibayar
Submit	

2.3 Setelah berhasil menambah Data Transaksi akan otomatis diarahkan untuk mengisi Detail Transaksi dari Data Transaksi yang baru dibuat

Tambah Detail Transaksi

Paket

Keterangan

Kuantitas

3.1 Pilih 1 baris Data Transaksi lalu pilih Edit Data untuk mengedit Data Transaksi

Data Transaksi

ID	Kode Invoice	Outlet	Karyawan	Pelanggan	Tanggal	Batas Waktu	Waktu Bayar	Status	Dibayar
46	L-46-2023-03-22	UKK	Rei	Kita	2023-03-22	2023-03-22	2023-03-22	Baru	Dibayar

3.2 User dapat mengubah (hanya) Status dan Sudah atau Belum Dibayarnya suatu Transaksi

Tambah Transaksi

Outlet

Karyawan

Pelanggan

Tanggal

Batas Waktu

Waktu Bayar

Biaya Tambah

Diskon

Status

Dibayar

4.1 Pilih 1 baris Data Transaksi lalu pilih Detail Data untuk melihat Informasi lebih lanjut

Data Transaksi

ID	Kode Invoice	Outlet	Karyawan	Pelanggan	Tanggal	Batas Waktu	Waktu Bayar	Status	Dibayar
46	L-46-2023-03-22	UKK	Rei	Kita	2023-03-22	2023-03-22	2023-03-22	Baru	Dibayar

4.2 Informasi lebih lanjut mengenai Data Transaksi

PDF		Transaksi	
CSV	Laundrive		Data
IE	Kode Invoice L-46-2023-03-22		Detail Data
4	Klien		Tanggal
4	Outlet		Batas Waktu
	Kita		Waktu Bayar
	Jl. Kitaaaaa No.1		Status
	081234567890		Dibayar
	UKK		2023-03-22
	Jl. UKK.jpg		2023-03-24
	081234567890		2023-03-24
			Proses
			Belum_Dibayar
	Paket		
	Keterangan		
	Kuantitas		
	Total Harga		
	Kilos		
	Kilos		
	1		
	Rp. 9901		

Pengembangan Aplikasi

Aplikasi laundry berbasis desktop menggunakan python dengan database MySQL akan membantu mengelola dan mempermudah operasional bisnis laundry. Aplikasi ini memiliki fitur CRUD outlet, paket pembelian, karyawan, pelanggan, dan data transaksi, serta dapat menghasilkan laporan untuk data transaksi. Dengan fitur CRUD, pengguna dapat melakukan operasi dasar seperti membuat, membaca, memperbarui, dan menghapus data outlet, paket pembelian, karyawan, dan pelanggan. Sementara itu, data transaksi akan dicatat dan dapat digunakan untuk menghasilkan laporan transaksi. Aplikasi ini dapat membantu mempercepat proses bisnis laundry serta memungkinkan pengguna untuk mengambil keputusan yang tepat berdasarkan data transaksi yang dihasilkan.