

**TUGAS 8 PBOP MENGENAI IMPELEMENTASI GUI PADA STUDY
KASUS**



DI SUSUN OLEH :

BRYAN ALBERD LEONARDO SIHOMBING (5230411253)

**FAKULTAS SAINS & TEKNOLOGI
UNIVERSITAS TEKNOLOGI YOGYAKARTA**

28/11/2024

DAFTAR ISI

HALAMAN JUDUL	i
DAFTAR ISI	ii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	1
1.3. Tujuan	1
BAB II PEMBAHASAN	2
2.1. Activity Diagram	2
2.2. Codingan	2
2.3. Penjelasan	3
2.4. Dokumentasi	4
BAB III PENUTUP	5
3.1. Kesimpulan	5

BAB I

PENDAHULUAN

1.1. Latar Belakang

Daftar akun game online adalah sebuah akun yang digunakan untuk mendaftarkan akun ke dalam sebuah game tersebut agar kita menyimpan data game tersebut ke dalam akun yang sudah kita buat. Dalam era digital yang semakin maju, game online telah menjadi salah satu hiburan utama bagi banyak orang hampir di seluruh dunia. Sebagai bagian dari pengalaman bermain game, pengelolaan akun pengguna menjadi elemen sangat penting untuk bermain game agar semua data yang kita punya bisa tersimpan ke dalam akun yang sudah ada. Untuk itu GUI (*Graphical User Interface*) dalam fitur daftar akun game online memegang peran krusial sebagai penghubung antara pemain dengan game.

1.2. Rumusan Masalah

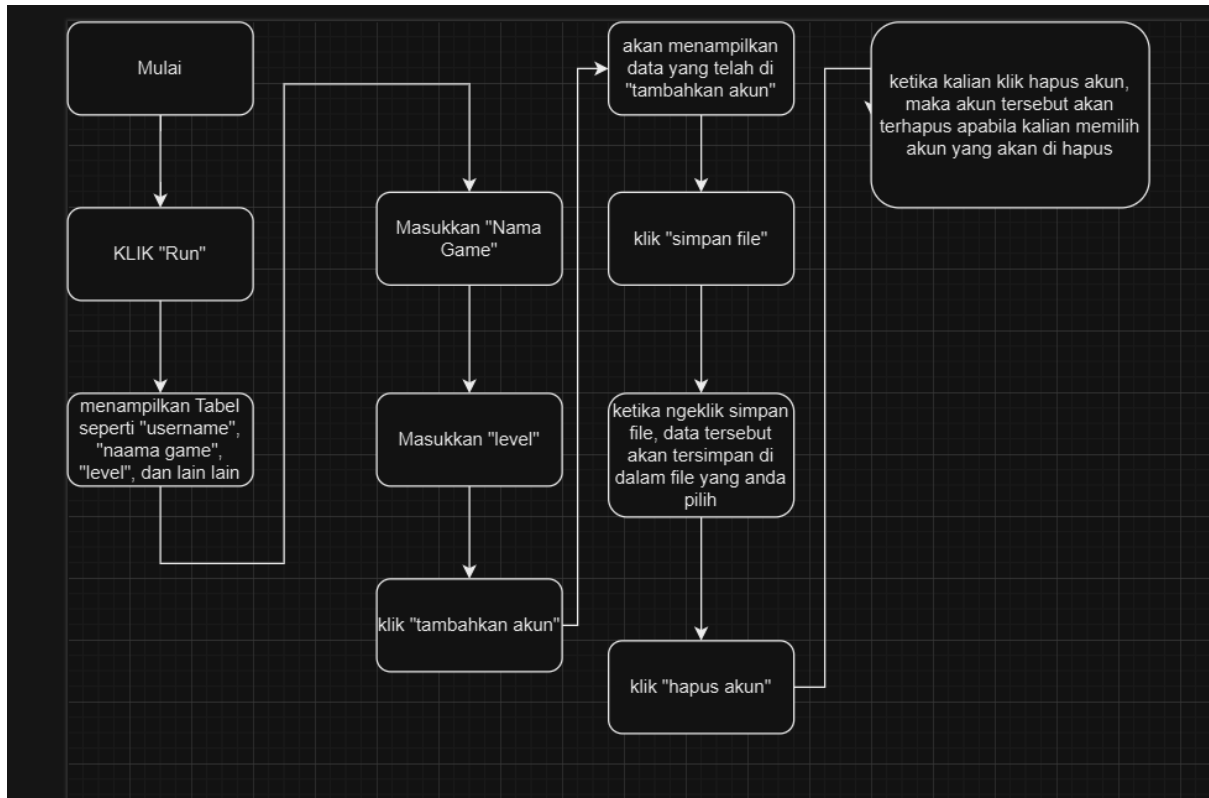
1. Bagaimana cara membuat daftar isi akun game online sederhana menggunakan Tkinter?
2. Bagaimana cara menambahkan username dan nama game pada aplikasi?
3. Bagaimana cara menambahkan akun dan menghapus akun dalam aplikasi?

1.3. Tujuan

1. Membuat daftar akun game online menggunakan python dan Tkinter.
2. Menyediakan fitur penghapusan akun dan menambahkan akun agar si pengguna dapat menggunakannya dengan mudah.
3. Menyediakan penyimpanan file akun agar akun yang sudah di tambahkan akan tersimpan ke dalam file.

BAB II PEMBAHASAN

2.1. Activity Diagram



1. Mulai : memulai program
2. Klik Run : ketika mengklik akan menampilkan sebuah aplikasi sederhana yang berisi Username, Nama game, level, Tambah akun, Simpan ke file, Hapus akun.
3. Username : ketika kalian mengisi username kalian, maka username tersebut akan di tampilkan ke dalam table.
4. Nama game : sama seperti Username, maka akan memunculkan nama game yang kalian masukkan ke dalam table.
5. Level : masukkan level yang kalian inginkan, maka akan muncul sama seperti yang lainnya.
6. Tambah akun : ketika kalian klik nambah akun maka (username, nama game, dan level) akan muncul di dalam table dan akan memunculkan notifikasi seperti “akun berhasil ditambahkan” apabila kalian tidak mengisi semuanya dan kalian klik maka akan muncul notifikasi seperti “semua kolom harus di isi”
7. Simpan ke file ; ketika table yang sudah kalian isi dan mau menyimpan ke file maka data tersebut akan tersimpan di file yang anda simpan.
8. Hapus akun : ketika kalian klik hapus akun maka akan menampilkan notifikasi seperti “ pilih akun yang akan di hapus” jadi jika kalian ingin menghapus akun maka pilih nama akun yang akan di hapus apabila kalian sudah memilih akun yang akan di hapus maka akan menampilkan notifikasi seperti “apakah anda yakin ingin menghapus akun

yang dipilih?” maka ketika kalian klik “ya” akun tersebut akan terhapus dan memunculkan notifikasi seperti “akun berhasil dihapus”.

2.2. Codingan

```
1 import tkinter as tk
2 from tkinter import messagebox
3 from tkinter import ttk
4
5 class DaftarAkunGameOnline:
6     def __init__(self, root):
7         self.root = root
8         self.root.title("Daftar Akun Game Online")
9         self.root.geometry("700x600")
10
11         self.akun_list = []
12
13         frame_input = tk.Frame(self.root)
14         frame_input.pack(pady=10)
15
16         tk.Label(frame_input, text="Username").grid(row=0, column=0, padx=5, pady=5)
17         self.entry_username = tk.Entry(frame_input)
18         self.entry_username.grid(row=0, column=1, padx=5, pady=5)
19
20         tk.Label(frame_input, text="Nama Game").grid(row=1, column=0, padx=5, pady=5)
21         self.entry_game = tk.Entry(frame_input)
22         self.entry_game.grid(row=1, column=1, padx=5, pady=5)
23
24         tk.Label(frame_input, text="Level").grid(row=2, column=0, padx=5, pady=5)
25         self.entry_level = tk.Entry(frame_input)
26         self.entry_level.grid(row=2, column=1, padx=5, pady=5)
27
28         btn_tambah = tk.Button(frame_input, text="Tambah Akun", command=self.tambah_akun)
29         btn_tambah.grid(row=3, column=0, columnspan=2, pady=10)
30
31         frame_tree = tk.Frame(self.root)
32         frame_tree.pack(pady=10)
33
34         columns = ("No", "Username", "Game", "Level")
35         self.tree = ttk.Treeview(frame_tree, columns=columns, show="headings")
36         self.tree.heading("No", text="No")
37         self.tree.heading("Username", text="Username")
```

```

38     self.tree.heading("Game", text="Game")
39     self.tree.heading("Level", text="Level")
40     self.tree.column("No", width=50, anchor="center")
41     self.tree.column("Level", width=80, anchor="center")
42     self.tree.pack()
43
44     btn_simpan = tk.Button(self.root, text="Simpan ke File", command=self.simpan_ke_file)
45     btn_simpan.pack(pady=5)
46
47     btn_hapus = tk.Button(self.root, text="Hapus Akun", command=self.hapus_akun)
48     btn_hapus.pack(pady=5)
49
50     def tambah_akun(self):
51         username = self.entry_username.get()
52         game = self.entry_game.get()
53         level = self.entry_level.get()
54
55         if username and game and level:
56             try:
57                 level = int(level)
58                 self.akun_list.append({'username': username, 'game': game, 'level': level})
59                 self.update_treeview()
60                 self.entry_username.delete(0, tk.END)
61                 self.entry_game.delete(0, tk.END)
62                 self.entry_level.delete(0, tk.END)
63                 messagebox.showinfo("Sukses", "Akun berhasil ditambahkan!")
64             except ValueError:
65                 messagebox.showerror("Error", "Level harus berupa angka.")
66         else:
67             messagebox.showwarning("Peringatan", "Semua kolom harus diisi!")

```

```

68
69     def update_treeview(self):
70         for row in self.tree.get_children():
71             self.tree.delete(row)
72         for idx, akun in enumerate(self.akun_list):
73             self.tree.insert("", "end", values=(idx + 1, akun['username'], akun['game'], akun['level']))
74
75     def simpan_ke_file(self):
76         try:
77             with open("daftar_akun_game.txt", "w") as file:
78                 for akun in self.akun_list:
79                     file.write(f"{akun['username']} - {akun['game']} - Level {akun['level']}\n")
80                 messagebox.showinfo("Sukses", "Data berhasil disimpan ke file!")
81         except Exception as e:
82             messagebox.showerror("Error", f"Gagal menyimpan file: {e}")
83
84     def hapus_akun(self):
85         selected_item = self.tree.selection()
86         if selected_item:
87             confirm = messagebox.askyesno("Konfirmasi", "Apakah Anda yakin ingin menghapus akun yang dipilih?")
88             if confirm:
89                 for item in selected_item:
90                     item_index = self.tree.index(item)
91                     self.akun_list.pop(item_index)
92                     self.tree.delete(item)
93                 self.update_treeview()
94                 messagebox.showinfo("Sukses", "Akun berhasil dihapus!")
95             else:
96                 messagebox.showwarning("Peringatan", "Pilih akun yang ingin dihapus!")
97
98 if __name__ == "__main__":
99     root = tk.Tk()
100     app = DaftarAkunGameOnline(root)
101     root.mainloop()
102

```

2.3. Penjelasan

2.3.1. inisialisasi Program (__init__):

1. self.root : menampilkan jendela dengan ukuran 700x600 pixel
2. self.akun_list[] : menampilkan semua data yang sudah di isi
3. columns = ("no, Username, game, level") : menampilkan status di dalam table
4. messagebox error/warning : ketika kalian mengisi level kalian bukan dengan angka maka output tersebut akan eror, ketika kalian tidak mengisi kolom lalu kalian klik tambah akun maka akan ada peringatan seperti "kolom harus di isi"

2.3.2. def simpan_ke_file(self):

1. ketika sudah mengisi semua data dan ketika klik simpan maka file yang sudah terisi akan tersimpan.

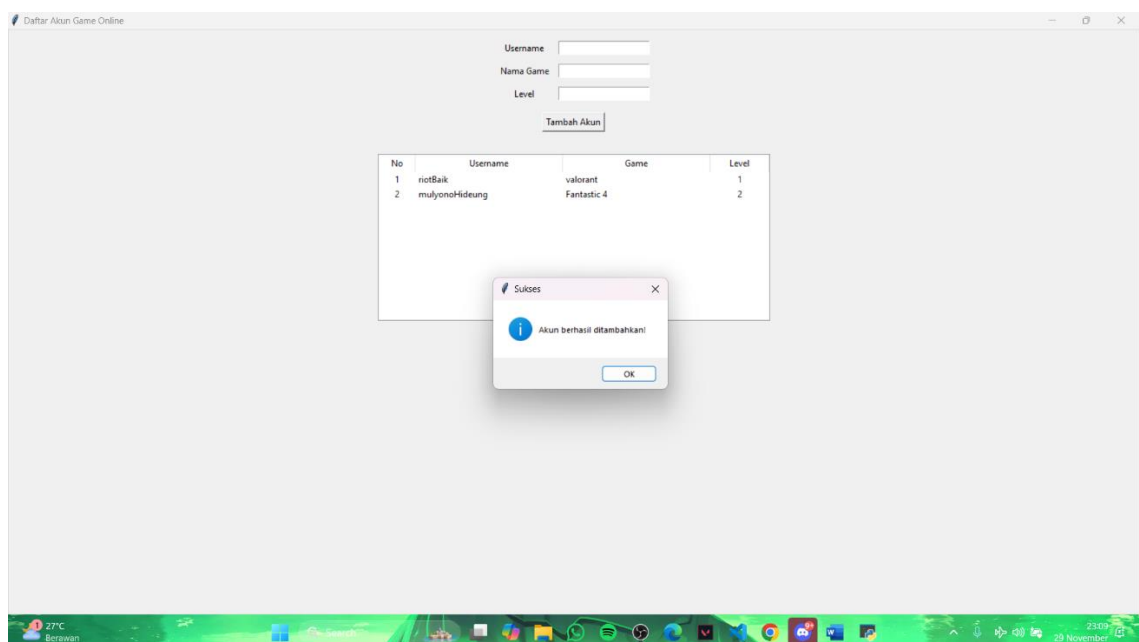
2.3.3. def hapus_akun(self):

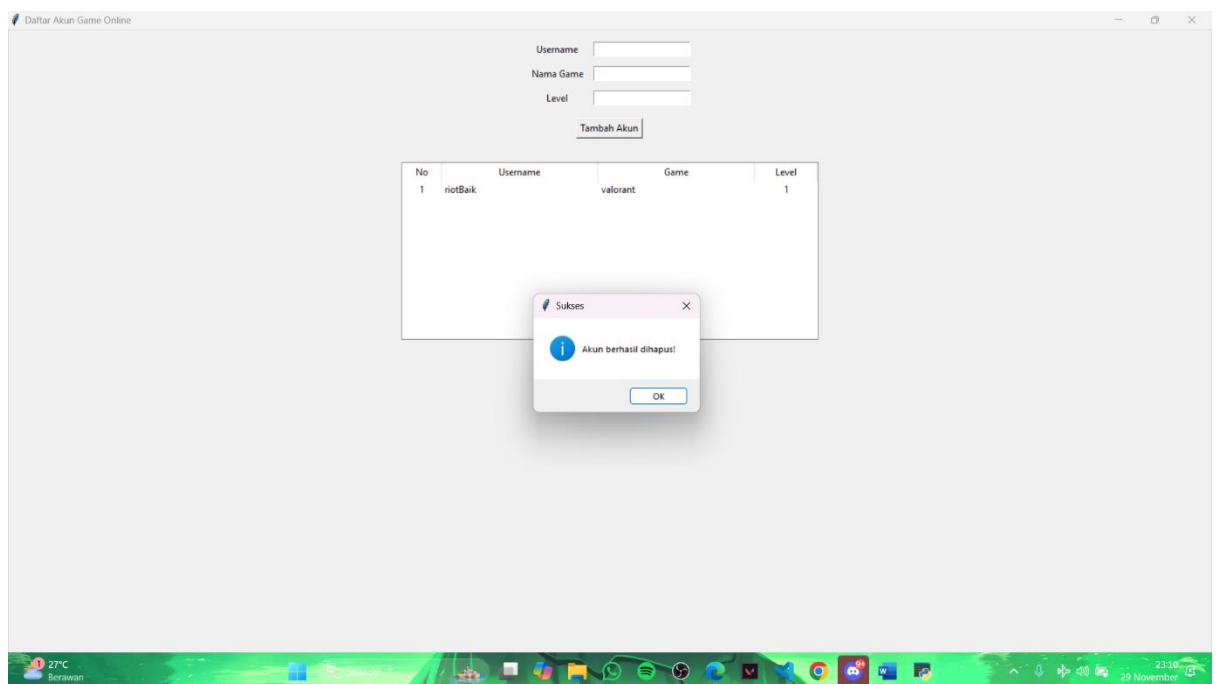
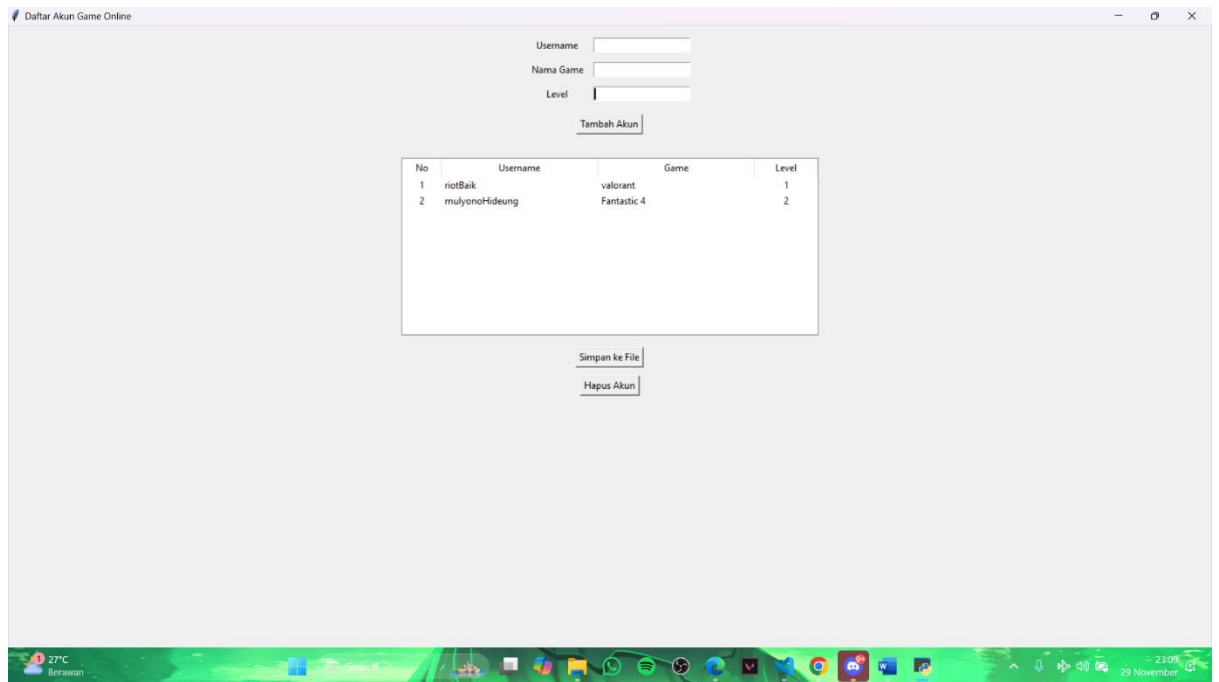
1. selected_item = self.tree.selection() : ketika kalian menghapus akun tanpa memilih akun yang akan di hapus maka tidak akan bisa menghapus akun, jadi kalian pilih dulu akun yang akan di hapus lalu kalian klik hapus akun maka akun tersebut berhasil dihapus.

2.3.4. Bagian utama (if __name__ == "__main__":)

1. Menampilkan dan menjalankan aplikasi Daftar Akun Game Online dengan root.mainloop()

2.4. Dokumentasi





BAB III

PENUTUPAN

3.1. Kesimpulan

Aplikasi daftar akun game online yang dibuat menggunakan Python dan Tkinter berhasil dijalankan dan juga berhasil mengimplementasikan fungsi dasar seperti Tambah akun, simpan ke file, dan hapus akun. Dalam daftar akun game online menggunakan sistem GUI *Graphical User Interface* memiliki peran yang sangat signifikan dalam memberikan pengalaman pengguna yang optimal, dan juga dalam program ini menggunakan library Tkinter, termasuk penggunaan widget seperti label dan tombol. Aplikasi ini juga memberikan penyimpanan file agar akun yang telah di isi tidak akan hilang.

Aplikasi akun game online ini adalah contoh aplikasi sederhana yang dibuat secara efektif untuk bisa memahami konsep dasar dalam pembuatan aplikasi menggunakan GUI dan juga library Tkinter.