



LAPORAN

PROGRAM

MASAK ON BUDGET

KELOMPOK 18



BAB I

DESKRIPSI MASALAH

Makanan merupakan kebutuhan dasar manusia. Setiap hari manusia melakukan kegiatan memasak untuk dapat mengolah bahan makanan sehingga dapat dimakan. Namun seringkali terjadi kebingungan di masyarakat dalam menentukan makanan apa yang akan dimasak. Hal tersebut terjadi karena uang yang dimiliki (*budget*) terbatas untuk membeli bahan-bahan yang digunakan dalam memasak, sehingga terkadang rasa masakan yang dihasilkan tidak sesuai yang diinginkan. Pengetahuan akan variasi makanan juga menjadi penyebab masalah ini terjadi, masyarakat tidak tahu sepenuhnya resep-resep masakan yang bahan-bahannya sesuai dengan uang yang dimiliki. Masalah ini sering terjadi di kalangan mahasiswa atau ibu rumah tangga yang memiliki uang terbatas, sehingga mereka harus berpikir setiap harinya hanya untuk menentukan makanan apa yang akan dimasak.

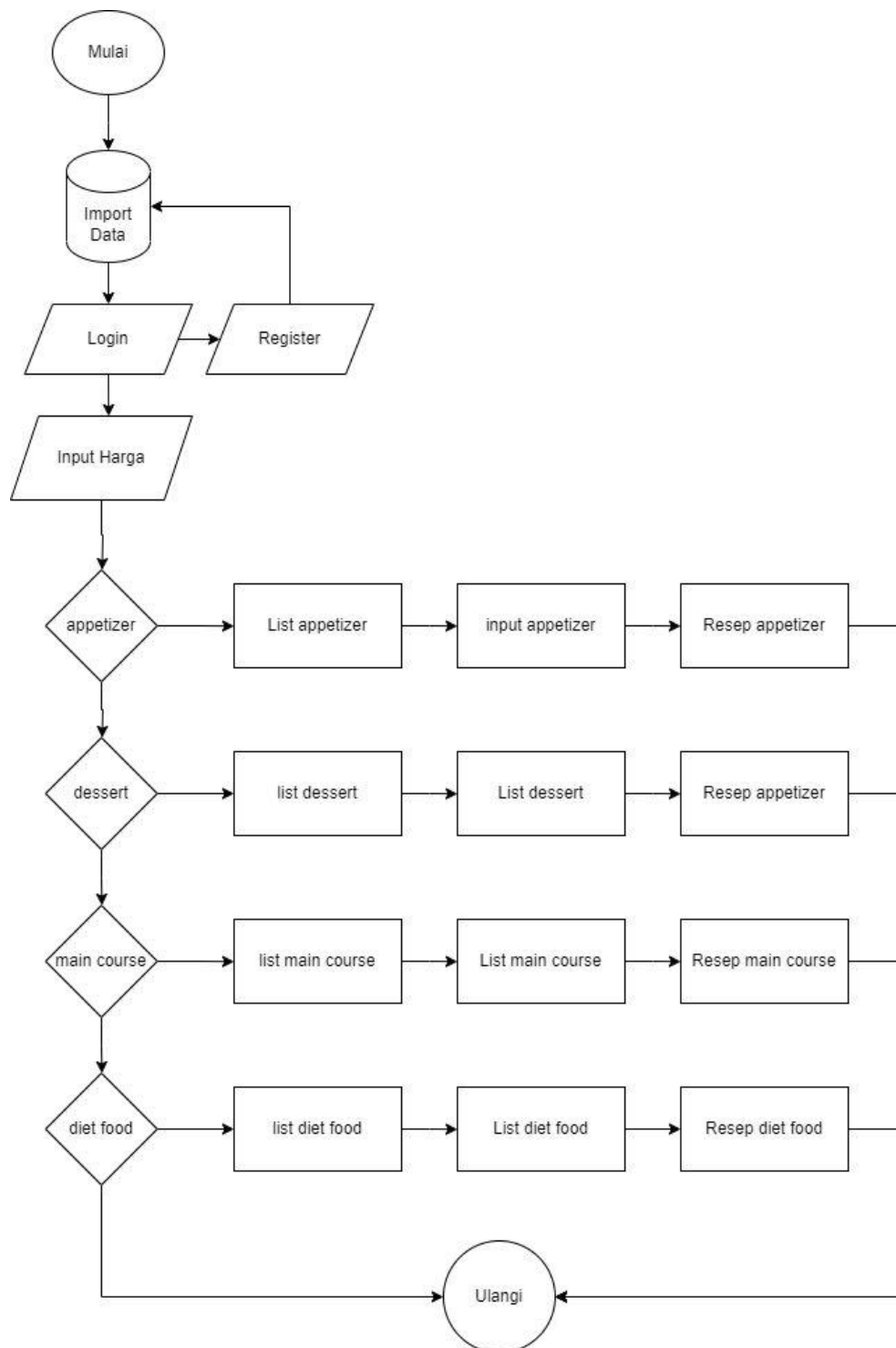
Berakar dari masalah tersebut maka diperlukan suatu perangkat untuk memudahkan masyarakat khususnya mahasiswa dan ibu rumah tangga dalam menentukan makanan apa yang akan dimasak sesuai dengan jumlah uang yang dimiliki, sehingga bahan-bahan dalam masakan dapat terpenuhi dan makanan yang dihasilkan memiliki rasa yang sesuai. Selain itu pengguna nantinya dapat juga memilih resep masakan sesuai dengan jenis masakan yang diinginkan seperti *appetizer*, *main course*, *dessert*, atau *diet food*, sehingga pilihan masakan yang tersedia akan lebih bervariasi. Dengan begitu, diharapkan pengguna khususnya mahasiswa dan ibu rumah tangga tidak lagi kesulitan dalam menentukan makanan yang dimasak, sehingga pengeluaran untuk makanan dapat lebih terkontrol.

Dalam menerapkan solusi dari masalah tersebut, dapat dibuat sebuah program dalam komputer dengan menggunakan bahasa pemrograman python dengan nama 'Masak on Budget'. Dalam program tersebut memuat kumpulan resep yang dapat dipilih oleh pengguna sesuai dengan keadaan berdasarkan uang yang dimiliki dan jenis masakan yang diinginkan. Untuk membuat tampilan program yang lebih menarik, maka program disajikan dengan menggunakan GUI (*Graphical User Interface*). Dengan begitu program dapat dengan mudah dipahami dan digunakan oleh pengguna.

BAB II

FLOWCHART DENGAN PENJELASANNYA

Bab ini akan membahas mengenai *flowchart* program bersama dengan penjelasannya. Untuk memudahkan dalam membuat program, maka terlebih dahulu membuat *flowchart* dari program yang dibuat. Berikut ini merupakan *flowchart* program 'Masak on Budget' :

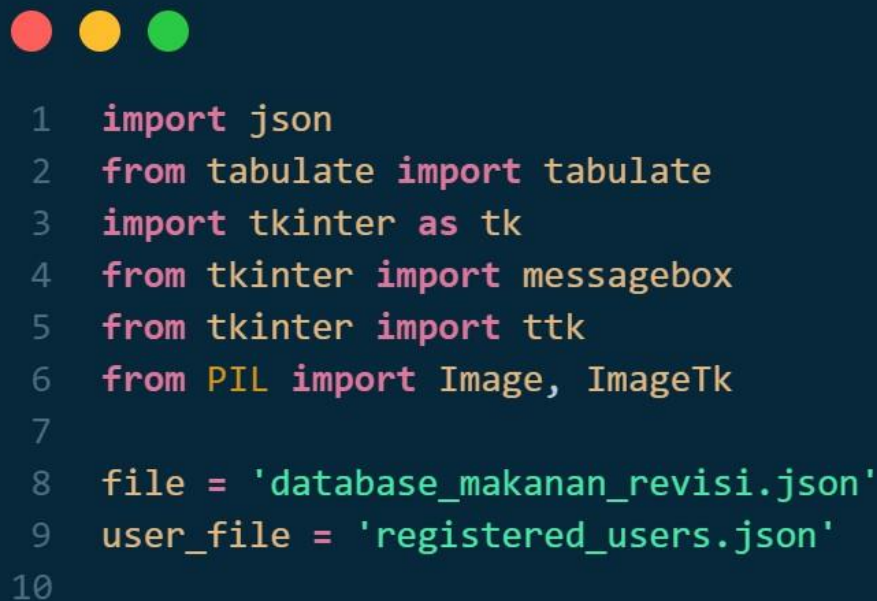


Saat program dimulai oleh pengguna, program akan terlebih dahulu mengimport seluruh data dalam database resep masakan yang dimiliki. Setelah itu pengguna harus login ke dalam program. Jika pengguna baru pertama kali menggunakan program, maka pengguna tersebut harus terlebih dahulu melakukan registrasi. Setelah pengguna teregistrasi, maka pengguna dapat login ke dalam program. Setelah itu pengguna menginput data jumlah uang yang dimiliki. Berdasarkan jumlah uang yang telah diinput, program akan menentukan resep masakan yang memiliki harga sesuai yaitu di bawah dari nilai input pengguna. Kemudian program akan meminta pengguna untuk memasukkan jenis makanan dengan memilih satu diantara empat jenis makanan yaitu *appetizer*, *dessert*, *main course*, dan *diet food*. Setelah menginput jenis makanan yang diinginkan, maka program akan menampilkan daftar makanan berdasarkan jenis makanan yang dipilih beserta harga total makanan. Kemudian pengguna akan diminta untuk memilih satu diantara daftar makanan yang tersedia. Setelah memilih makanan, maka program akan menampilkan data resep makanan antara lain nama makanan, deskripsi makanan, bahan-bahan yang dibutuhkan beserta harganya, langkah-langka memasak, serta jika tersedia akan muncul link video tutorial memasak makanan. Dengan demikian program sukses dijalankan oleh pengguna.

BAB III

KODE PEMROGRAMAN

Bab ini akan membahas mengenai kode program '*Masak On Budget*' Praktikum Program Komputer pada python.



```
1  import json
2  from tabulate import tabulate
3  import tkinter as tk
4  from tkinter import messagebox
5  from tkinter import ttk
6  from PIL import Image, ImageTk
7
8  file = 'database_makanan_revisi.json'
9  user_file = 'registered_users.json'
10
```

Gambar 3.1.Fungsi *Import*

Pada bagian awal program, terdapat fungsi *import* untuk mengatur dan mengimpor modul yang akan digunakan dalam program, serta menentukan nama file yang akan digunakan sebagai database. Modul yang digunakan seperti, *json*, *tabulate*, *tkinter*, *messagebox*, dan *Image*. Lalu dilanjutkan dengan mendefinisikan variable '*file*' dan '*user_file*'.

```

1 def register():
2     username = entry_username.get()
3     password = entry_password.get()
4
5     if username != "" and password != "":
6         user_data = {'username': username, 'password': password}
7
8         with open(user_file, 'r') as file_obj:
9             for line in file_obj:
10                 data = json.loads(line)
11                 if data['username'] == username:
12                     messagebox.showinfo("Info", "Anda sudah terdaftar! Silakan login.")
13                     return
14
15         with open(user_file, 'a') as file_obj:
16             json.dump(user_data, file_obj)
17             file_obj.write('\n')
18
19         messagebox.showinfo("Info", "Registrasi berhasil! Silakan login dengan akun yang telah didaftarkan.")
20     else:
21         messagebox.showerror("Error", "Username dan password tidak boleh kosong.")

```

Gambar 3.2. Fungsi Register

Berikut ini merupakan fungsi register, berfungsi untuk melakukan proses pendaftaran user baru dalam program. fungsi ini mengambil nilai dari dua input pengguna, yaitu username dan password. Kemudian, fungsi ini memeriksa apakah username dan password tidak kosong. Jika keduanya tidak kosong, fungsi melanjutkan dengan langkah berikutnya. Selanjutnya, fungsi membuat sebuah *user_data* yang berisi username dan password dalam bentuk dictionary. Fungsi membuka variabel *user_file* (yang merupakan database pengguna terdaftar) dalam mode membaca ('r'). Fungsi memeriksa apakah *username* dalam data tersebut sama dengan *username* yang baru diinputkan. Jika ada kecocokan, fungsi menampilkan pesan informasi bahwa pengguna sudah terdaftar dan mengembalikan (*return*) dari fungsi. Jika tidak ada kecocokan ditemukan, fungsi membuka file *user_file* dalam mode menambahkan ('a'). Fungsi menulis *user_data* ke dalam file dalam format *json* menggunakan *json.dump(user_data, file_obj)*, diikuti dengan penulisan karakter baris baru ('\n') untuk memisahkan setiap entri pengguna. Setelah berhasil menulis data pengguna baru, fungsi menampilkan pesan informasi bahwa registrasi berhasil dan menginstruksikan pengguna untuk melakukan login dengan akun yang telah didaftarkan. Jika *username* atau *password* kosong, fungsi akan menampilkan pesan error bahwa *username* dan *password* tidak boleh kosong.

```
1 def login():
2     username = entry_username.get()
3     password = entry_password.get()
4
5     with open(user_file, 'r') as file_obj:
6         users = [json.loads(line) for line in file_obj]
7
8     if any(user['username'] == username and user['password'] == password for user in users):
9         messagebox.showinfo("Info", "Login berhasil!")
10        root.deiconify()
11        login_window.withdraw()
12    else:
13        messagebox.showerror("Error", "Username atau Password salah, klik registrasi jika belum memiliki akun")
14
```

Gambar 3.3 Fungsi Login

Berikut merupakan Fungsi Login. Fungsi ini bertujuan untuk memvalidasi dan mengotentikasi pengguna yang mencoba melakukan login ke dalam program berdasarkan data pengguna yang tersimpan dalam file database. Setelah mendapatkan daftar pengguna dari file, fungsi melakukan pengecekan apakah terdapat pengguna dengan *username* dan *password* yang sesuai dengan input pengguna saat *login*. Dengan menggunakan pernyataan *if any(user['username'] == username and user['password'] == password for user in users):* , jika terdapat pengguna yang sesuai, maka pesan informasi menggunakan *messagebox.showinfo()* akan ditampilkan bahwa “login berhasil”. Selain itu, tampilan jendela *login* akan disembunyikan (*login_window.withdraw()*) dan jendela utama program akan ditampilkan (*root.deiconify()*). Jika tidak terdapat pengguna dengan *username* dan *password* yang sesuai, maka pesan kesalahan menggunakan *messagebox.showerror()* akan ditampilkan bahwa “*username* atau *password* salah”, dan disarankan untuk menekan tombol registrasi jika belum memiliki akun.


```

1  def get_masakan_by_harga():
2      global file
3      global hasil_pencarian
4      budget = int(entry_budget.get())
5      if budget > 50000:
6          messagebox.showinfo("Info", "Batas Budget Adalah 50.000")
7      if budget < 5000:
8          messagebox.showinfo("Info", "Maaf Uang Anda Tidak Cukup")
9      kategori = entry_kategori.get().lower()
10
11     with open(file, 'r') as file_obj:
12         data = json.load(file_obj)
13
14     hasil_pencarian = []
15
16     for makanan in data:
17         if makanan['Kategori'] == kategori:
18             total_harga = sum(makanan['Harga'])
19             if total_harga <= budget:
20                 makanan['Total Harga'] = total_harga
21                 hasil_pencarian.append(makanan)
22
23     table_data = print_masakan(hasil_pencarian)
24     if table_data:
25         result_text.config(state='normal')
26         result_text.delete("1.0", tk.END)
27         result_text.insert(tk.END, "Daftar masakan:\n\n")
28         result_text.insert(tk.END, table_data)
29         result_text.config(state='disabled')
30     else:
31         messagebox.showinfo("Info", "Tidak ada masakan yang sesuai dengan kriteria.")
32

```

Gambar 3.4. Fungsi Get Masakan By Harga

Bagian ini merupakan Fungsi Get Masakan By Harga, bertujuan untuk mencari dan menampilkan makanan yang sesuai dengan kriteria yang diberikan pengguna berdasarkan 'budget' dan 'kategori'. Fungsi ini mengambil input pengguna berupa 'budget' (dalam bentuk integer) dan 'kategori' (dalam bentuk string). Variabel 'file' dan 'hasil_pencarian' dijadikan variabel global. Fungsi ini melakukan beberapa validasi, jika 'budget' lebih dari 50000, fungsi menampilkan pesan informasi bahwa batas *budget* adalah 50000. Dan jika *budget* kurang dari 5000, fungsi menampilkan pesan informasi bahwa uang pengguna tidak cukup.

Fungsi membaca data dari 'file' dengan menggunakan '*json.load(file_obj)*' dan menyimpannya dalam variabel 'data'. Variabel 'hasil_pencarian' diinisialisasi sebagai list kosong. Fungsi melakukan iterasi pada setiap makanan dalam data, Jika kategori makanan sama dengan 'kategori' yang diinputkan pengguna, maka dilakukan penjumlahan harga

makanan dengan menggunakan `'sum(makanan['Harga'])'` dan disimpan dalam variabel `'total_harga'`. Jika `'total_harga'` kurang dari atau sama dengan `'budget'`, maka makanan tersebut ditambahkan ke dalam `'hasil_pencarian'` dengan menambahkan key baru `"Total Harga"` pada *dictionary* makanan.

Fungsi memanggil fungsi `'print_masakan()'` dengan parameter `'hasil_pencarian'` untuk mendapatkan data makanan yang sesuai dalam bentuk tabel. Jika `'table_data'` ada (tidak None), maka tampilan `'result_text'` akan diperbarui dengan menampilkan daftar makanan yang sesuai. Jika `'table_data'` kosong (None), maka fungsi menampilkan pesan informasi bahwa tidak ada makanan yang sesuai dengan kriteria.



```

1  def print_masakan(masakan):
2      if len(masakan) > 0:
3          table = []
4          for i, makanan in enumerate(masakan, start=1):
5              table.append([i, makanan['Nama'], makanan['Total Harga']])
6          headers = ["No.", "Nama", "Total Harga"]
7          table_data = tabulate(table, headers=headers)
8
9          return table_data
10     else:
11         return None

```

Gambar 3.5. Fungsi Print Masakan

Berikut merupakan Fungsi Print Masakan, digunakan untuk membantu menampilkan tabel yang menampilkan nomor, nama makanan, dan total harga untuk setiap makanan yang sesuai. Fungsi ini menerima masukan berupa list `'masakan'` yang berisi makanan yang sesuai dengan kriteria. Fungsi ini memeriksa jika panjang `'masakan'` lebih dari 0 (ada makanan yang sesuai), maka fungsi akan membuat sebuah list `'table'` yang berisi nomor, nama makanan, dan total harga untuk setiap makanan dalam `'masakan'`. Fungsi menggunakan modul `'tabulate'` untuk memformat data dalam bentuk tabel dengan menyediakan judul kolom (*headers*). Fungsi juga mengembalikan data tabel tersebut. Jika `'masakan'` kosong (tidak ada makanan yang sesuai), fungsi mengembalikan menjadi `'None'`.

```

1  def print_resep(hasil_pencarian):
2      masakan = hasil_pencarian
3
4      if masakan:
5          pilihan = int(entry_pilihan.get())
6          if 1 <= pilihan <= len(masakan):
7              masakan_pilihan = masakan[pilihan - 1]
8              resep = masakan_pilihan['Resep']
9
10             resep_tab = tk.Toplevel(root)
11             resep_tab.title("Resep " + masakan_pilihan['Nama'])
12             resep_tab.geometry("1280x720")
13
14             background_image_resep = Image.open("makanan.jpg")
15             background_image_resep = background_image_resep.resize((1280, 720), Image.ANTIALIAS)
16             background_photo_resep = ImageTk.PhotoImage(background_image_resep)
17
18             resep_text = tk.Text(resep_tab, height=60, width=80)
19             resep_text.pack()
20
21             resep_text.insert(tk.END, f"\n\nResep {masakan_pilihan['Nama']}:\n")
22             resep_text.insert(tk.END, "Bahan:\n")
23
24             bahan_table = [[i + 1, bahan, harga] for i, (bahan, harga) in enumerate(zip(resep['Bahan'], masakan_pilihan['Harga']))]
25             bahan_headers = ["No.", "Bahan", "Harga"]
26             bahan_table_data = tabulate(bahan_table, headers=bahan_headers)
27
28             resep_text.insert(tk.END, bahan_table_data)
29             resep_text.insert(tk.END, "\n\nInstruksi:\n")
30
31             instruksi_table = [[langkah] for langkah in resep['Instruksi']]
32             instruksi_table_data = tabulate(instruksi_table)
33
34             resep_text.insert(tk.END, instruksi_table_data)
35             resep_text.config(state='disabled')
36         else:
37             messagebox.showerror("Error", "Masukkan Angka Yang Sesuai.")
38     else:
39         messagebox.showinfo("Info", "Tidak ada masakan yang sesuai dengan kriteria.")
40

```

Gambar 3.6. Fungsi Print Resep

Berikut merupakan Fungsi Print Resep, fungsi ini bertujuan untuk mencetak resep makanan yang dipilih oleh user dalam bentuk tabel dengan menggunakan modul *'tabulate'*. Jendela baru (*resep_tab*) digunakan untuk menampilkan resep tersebut dengan bantuan objek *Text* (*resep_text*). Gambar latar belakang juga ditambahkan untuk memberikan tampilan yang lebih menarik pada jendela *resep_tab*.



Gambar 3.7. Fungsi Search Again

Berikut ini merupakan Fungsi Search Again yang bertujuan untuk mengatur ulang elemen-elemen pada interface user setelah pencarian selesai atau pengguna ingin melakukan pencarian kembali.



Gambar 3.8. Fungsi Logout

Berikut merupakan Fungsi Logout, fungsi ini bertujuan untuk melakukan proses log out dari akun *user*. *User* akan keluar dari sesi akun yang sedang digunakan. Tampilan menu login akan ditampilkan kembali, sementara tampilan menu utama akan disembunyikan. Selain itu, semua input yang ada pada elemen-elemen *interface user* akan dihapus. Hal ini memberikan *user* kesempatan untuk masuk dengan akun yang berbeda atau untuk memulai sesi baru.

```

1  root = tk.Tk()
2  root.title("Program Masak on Budget")
3  root.geometry("1280x720")
4
5  style = ttk.Style()
6  style.theme_use("clam")
7  style.configure("Biru.TButton", background="blue", foreground="white")
8  style.configure("Merah.TButton", background="red", foreground="white")
9  style.configure("Custom.TLabel", background="white", foreground="black")
10 style.configure("Green.TButton", background="green", foreground="white")
11
12 background_image = Image.open("makanan.jpg")
13 background_image = background_image.resize((1280, 720), Image.ANTIALIAS)
14 background_photo = ImageTk.PhotoImage(background_image)
15 background_label = tk.Label(root, image=background_photo)
16 background_label.place(x=0, y=0, relwidth=1, relheight=1)
17
18 login_window = tk.Toplevel(root)
19 login_window.title("Login")
20 login_window.geometry("1280x720")

```

Gambar 3.9. Kode Program Graphical User Interface (GUI)

Berikut ini merupakan implementasi (GUI) menggunakan modul Tkinter.

```

1  background_image_login = Image.open("makanan.jpg")
2  background_photo_login = ImageTk.PhotoImage(background_image_login)
3  background_label_login = tk.Label(login_window, image=background_photo_login)
4  background_label_login.place(x=0, y=0, relwidth=1, relheight=1)
5
6  frame_login = ttk.Frame(login_window)
7  frame_login.pack(pady=20)
8
9  label_title = ttk.Label(login_window, text="Program Masak On Budget", font=("Segoe UI Black", 28))
10 label_title.place(x=410, y=150)
11
12 label_username = ttk.Label(login_window, text="Username:", font=("Times New Roman", 14))
13 label_username.place(x=485, y=450)
14 entry_username = ttk.Entry(login_window, font=("Times New Roman", 14))
15 entry_username.place(x=570, y=450)
16
17 label_password = ttk.Label(login_window, text="Password:", font=("Times New Roman", 14))
18 label_password.place(x=485, y=480)
19 entry_password = ttk.Entry(login_window, font=("Times New Roman", 14), show="*")
20 entry_password.place(x=570, y=480)

```

Gambar 3.10. Kode Program Graphical User Interface (GUI)


```

1 button_login = ttk.Button(login_window, text="Login", style="Biru.TButton", command=login)
2 button_login.place(x=600,y=550)
3
4 button_register = ttk.Button(login_window, text="Register", style="Merah.TButton", command=register)
5 button_register.place(x=600,y=600)
6 root.withdraw()
7
8 frame_search = ttk.Frame(root)
9 frame_search.pack(pady=40)
10
11 label_budget = ttk.Label(frame_search, text="Masukkan budget Anda: (ex. 15000)", style="Custom.TLabel")
12 label_budget.grid(row=0, column=0, sticky=tk.W)
13 entry_budget = ttk.Entry(frame_search)
14 entry_budget.grid(row=0, column=1)
15
16 label_kategori = ttk.Label(frame_search, text="Masukkan kategori (appetizer/main course/dessert/diet food):", style="Custom.TLabel")
17 label_kategori.grid(row=1, column=0, sticky=tk.W)
18 entry_kategori = ttk.Entry(frame_search)
19 entry_kategori.grid(row=1, column=1)

```

Gambar 3.11. Kode Program Graphical User Interface (GUI)

```

1 button_cari = ttk.Button(root, text="Cari", style="Biru.TButton", command=get_masakan_by_harga)
2 button_cari.pack()
3
4 frame_resep = ttk.Frame(root)
5 frame_resep.pack(pady=40)
6
7 label_pilihan = ttk.Label(frame_resep, text="Pilih nomor masakan untuk melihat resep:", style="Custom.TLabel")
8 label_pilihan.grid(row=0, column=0, sticky=tk.W)
9 entry_pilihan = ttk.Entry(frame_resep)
10 entry_pilihan.grid(row=0, column=1)
11
12 button_print_resep = ttk.Button(frame_resep, text="Print Resep", style="Biru.TButton", command=lambda: print_resep(hasil_pencarian))
13 button_print_resep.grid(row=1, column=0, pady=10)
14
15 button_search_again = ttk.Button(frame_resep, text="Cari Lagi", style="Green.TButton", command=search_again)
16 button_search_again.grid(row=1, column=1, pady=10)

```

Gambar 3.12. Kode Program Graphical User Interface (GUI)

```

1 button_logout = ttk.Button(root, text="Log Out", style="Merah.TButton", command=log_out)
2 button_logout.pack()
3
4 result_text = tk.Text(root, height=20, width=60)
5 result_text.pack()
6 result_text.config(state='disabled')
7
8 root.mainloop()

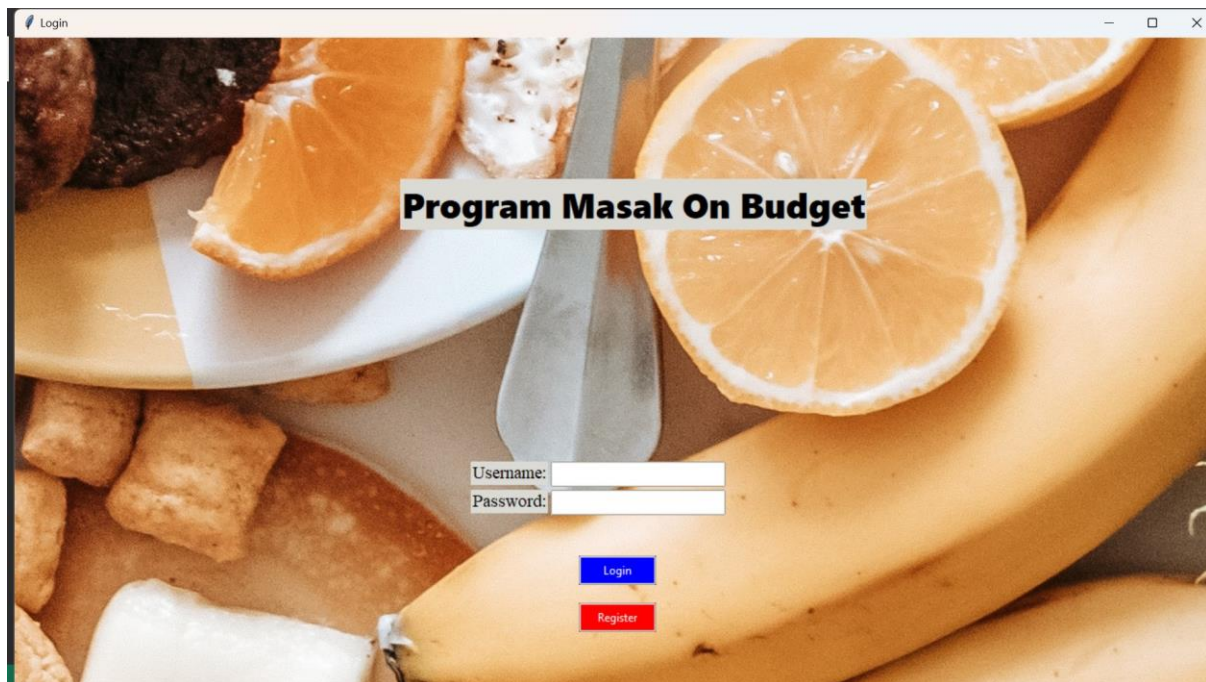
```

Gambar 3.13. Kode Program Graphical User Interface (GUI)

BAB IV

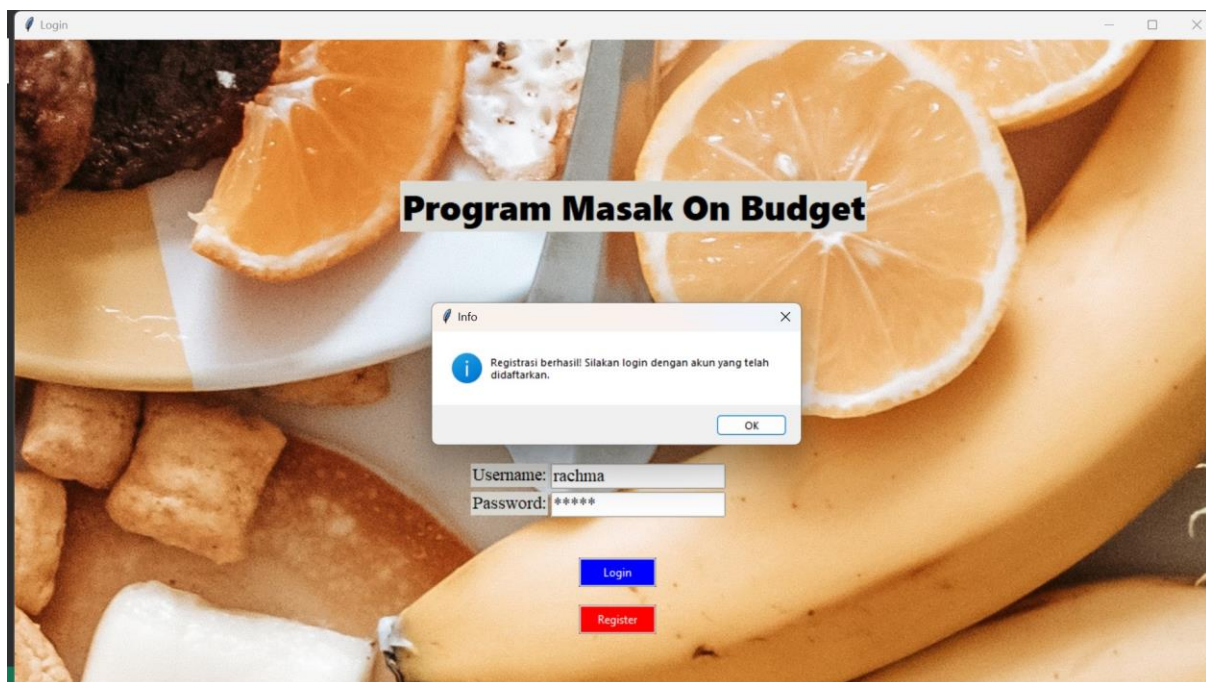
HASIL RUNNING PYTHON

Bab ini akan membahas mengenai hasil *running* dari program 'Masak On Budget' Praktikum Program Komputer pada python.



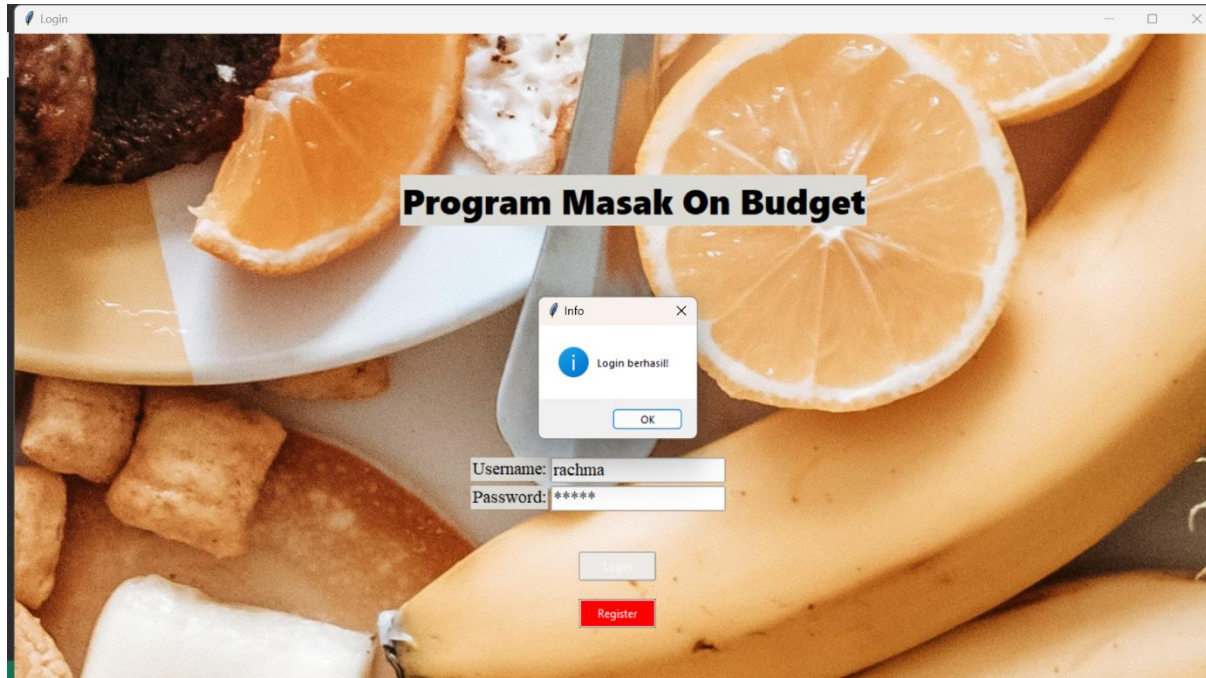
Gambar 4.1. Menu Login/Registrasi

Berikut merupakan tampilan menu login/registrasi.



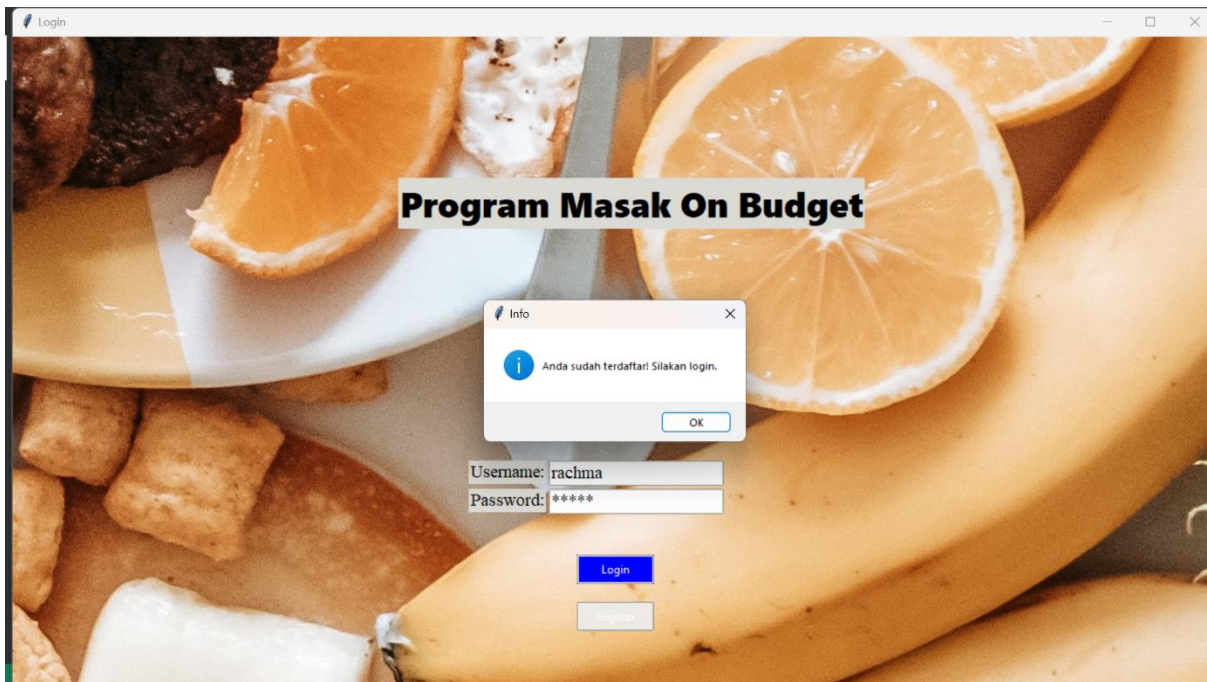
Gambar 4.2. Menu Login/Registrasi

Pada tampilan ini user diminta untuk registrasi akun terlebih dahulu, dengan mengetik username dan password lalu klik 'register', setelah itu messagebox muncul menandakan registrasi telah berhasil.



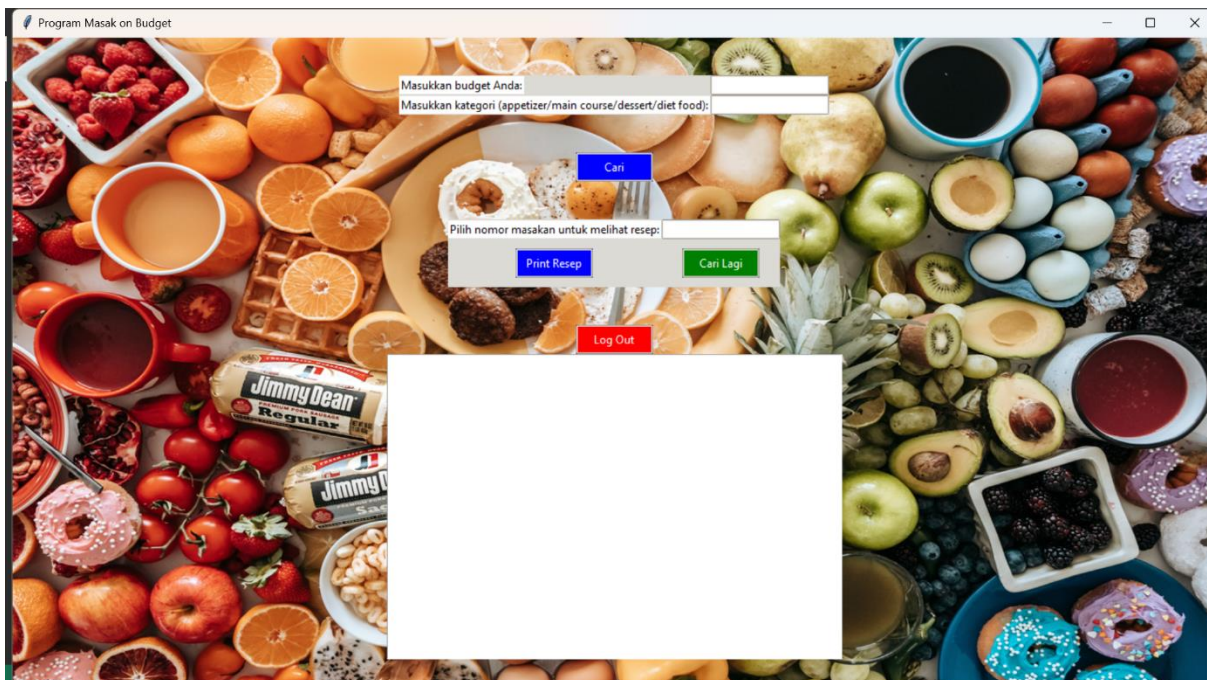
Gambar 4.3. Menu Login/Registrasi

Jika user sudah memiliki akun, user hanya diminta untuk memasukkan username dan password yang sesuai lalu klik button 'Login', maka akan muncul messagebox menandakan user berhasil login.



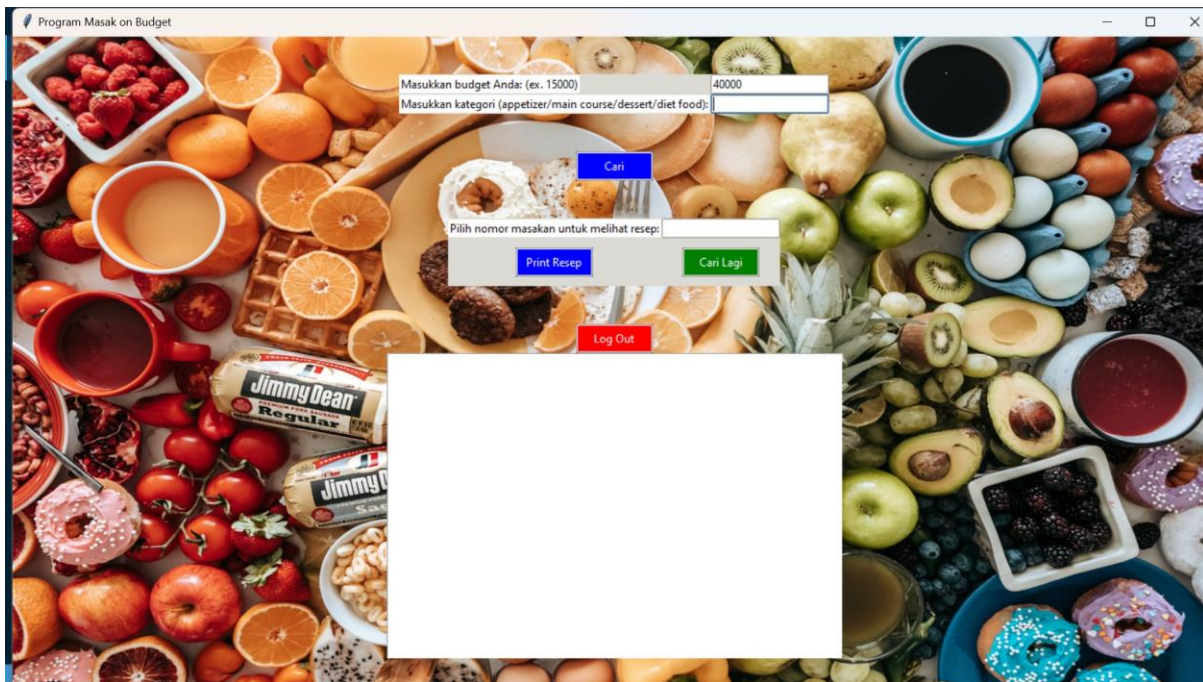
Gambar 4.4. Menu Login/Registrasi

Jika user telah memiliki akun, user tidak perlu registrasi lagi. Tampilan akan seperti ini jika user registrasi ulang dengan akun yang sudah ada.



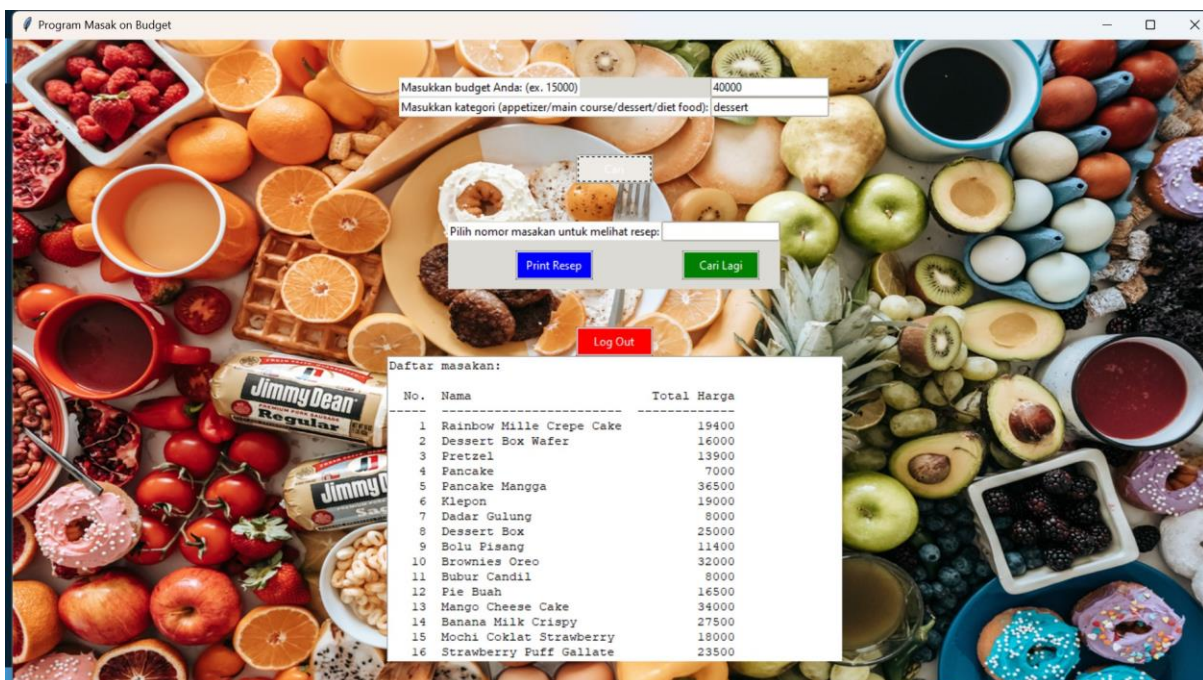
Gambar 4.5. Tampilan Menu Utama Program

Pada tampilan ini terdapat, *search box* untuk memasukkan budget, kategori makanan, pilihan menu makanan.



Gambar 4.6. Tampilan Menu Utama Program

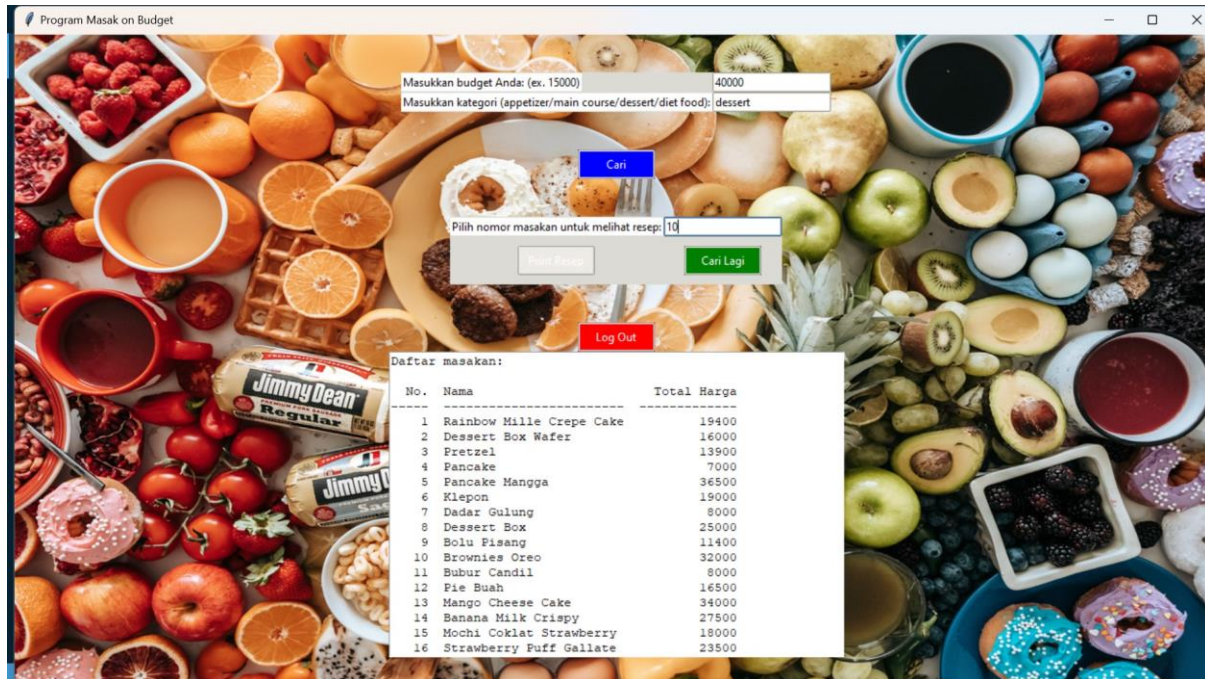
User memasukkan budget dari 5.000 rupiah hingga 50.000 rupiah dengan penulisan (contoh: 40000), lalu user memasukkan kategori makanan yang akan dipilih sesuai dengan kategori yang tersedia (contoh: *dessert*)



Gambar 4.7. Tampilan Menu Utama Program

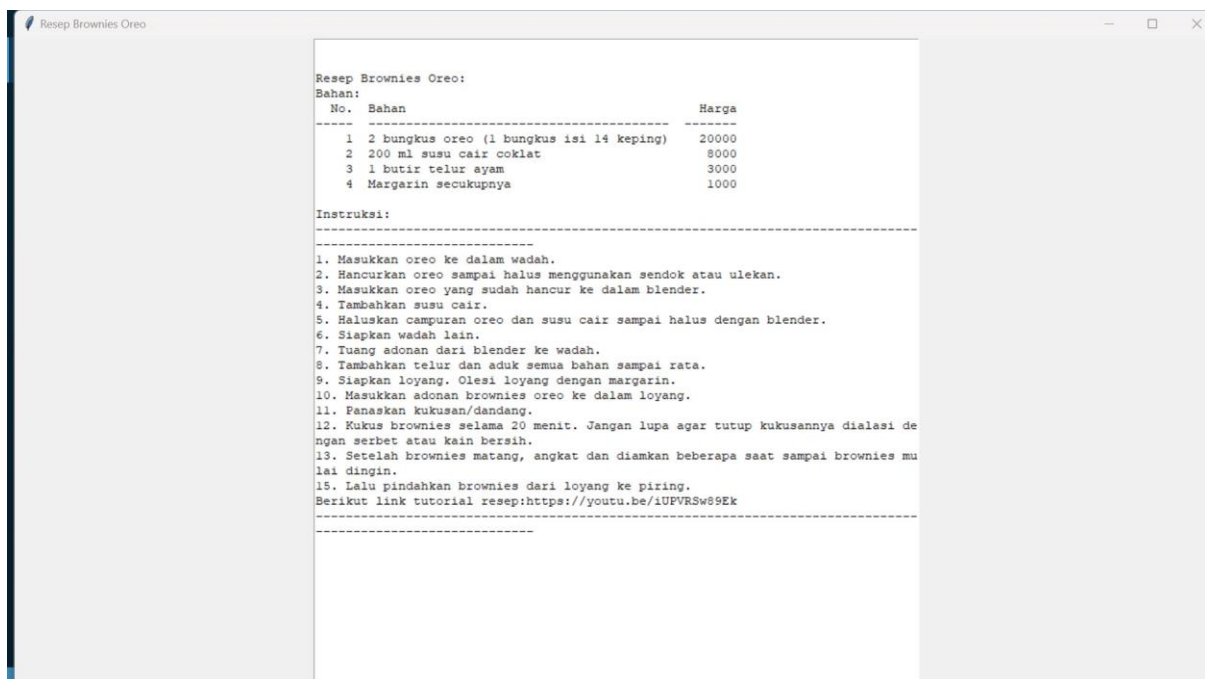
Setelah user menginputkan kategori makanan yang akan dipilih, *user* dapat menekan *button* 'Cari', lalu daftar menu makanan yang sesuai dengan kategori dan harga yang telah

diinputkan akan muncul. Menu makanan yang muncul adalah menu makanan dengan harga dibawah harga (*budget*) yang telah diinputkan.



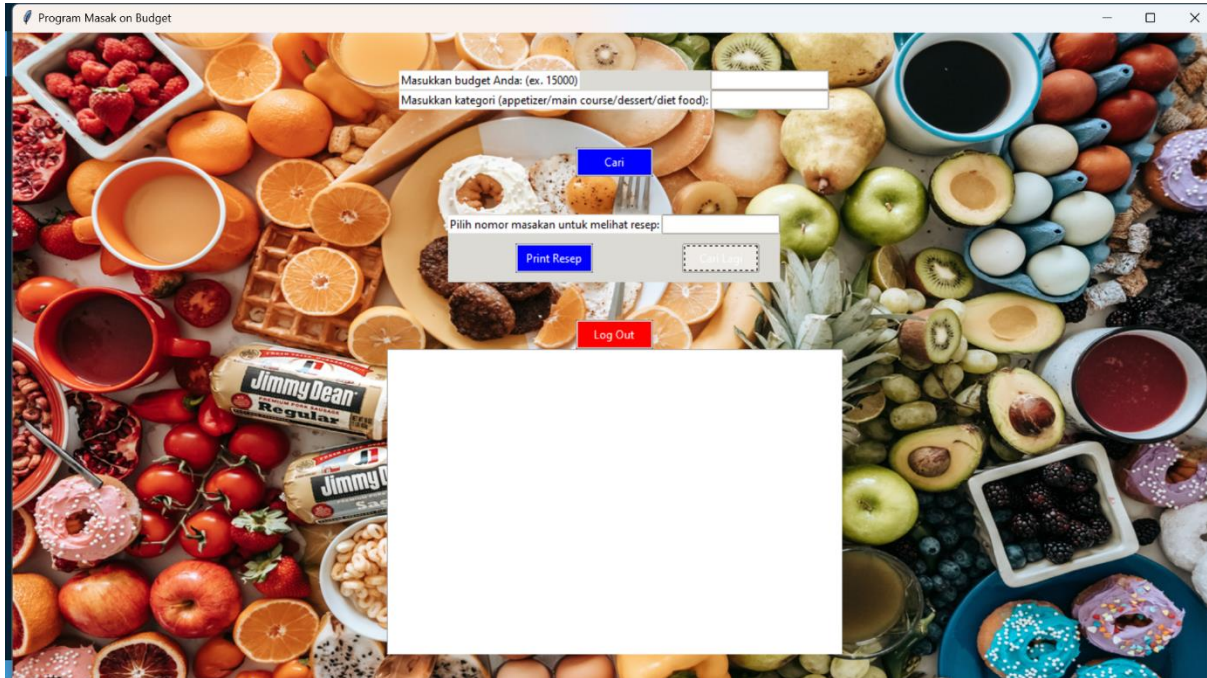
Gambar 4.8. Tampilan Menu Utama Program

Pada bagian ini *user* dapat menginputkan nomor sesuai dengan nomor menu makanan yang tersedia. Misalnya *user* ingin membuat Brownies Oreo, maka dari itu *user* harus menginputkan '10' pada *search box* yang tersedia. Lalu *user* menekan *button* 'Print Resep'



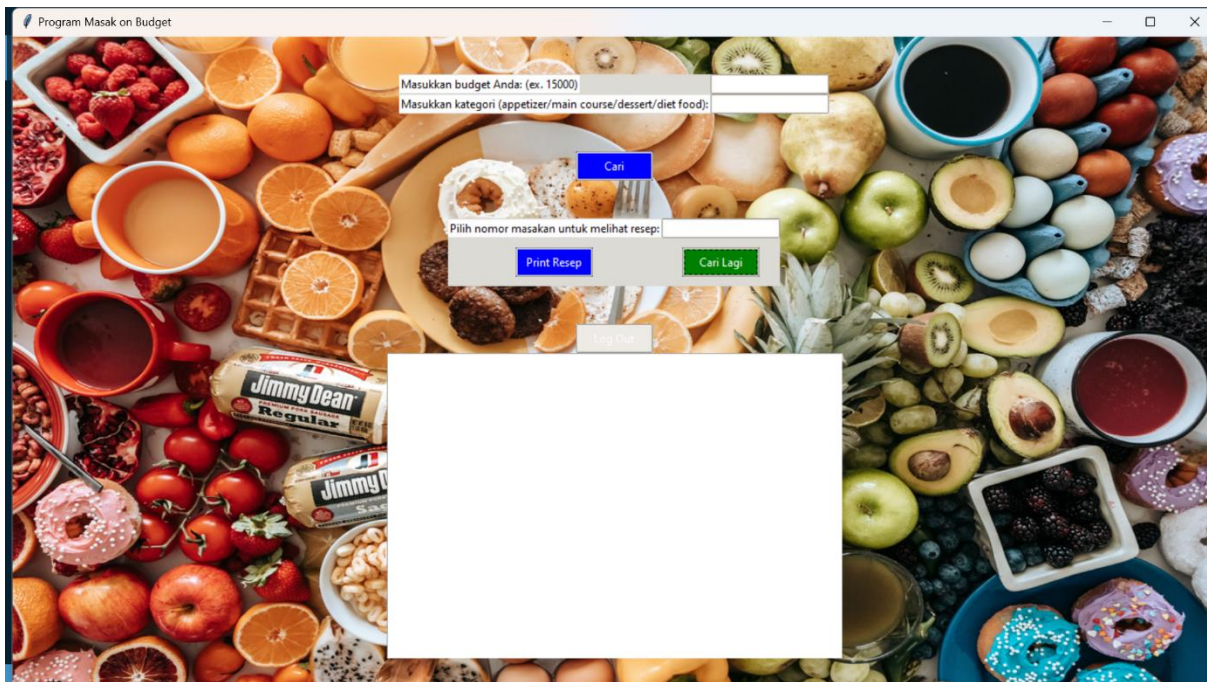
Gambar 4.9 Tampilan Resep

Berikut ini merupakan tampilan resep, dari nama menu makanan, bahan bahan yang diperlukan dengan harganya, tahap tahap memasaknya serta terdapat link video tutorial untuk melihat cara memasaknya dengan jelas. Untuk kembali ke tampilan menu utama, maka hanya perlu klik tanda (X) untuk menutup *page* resep.



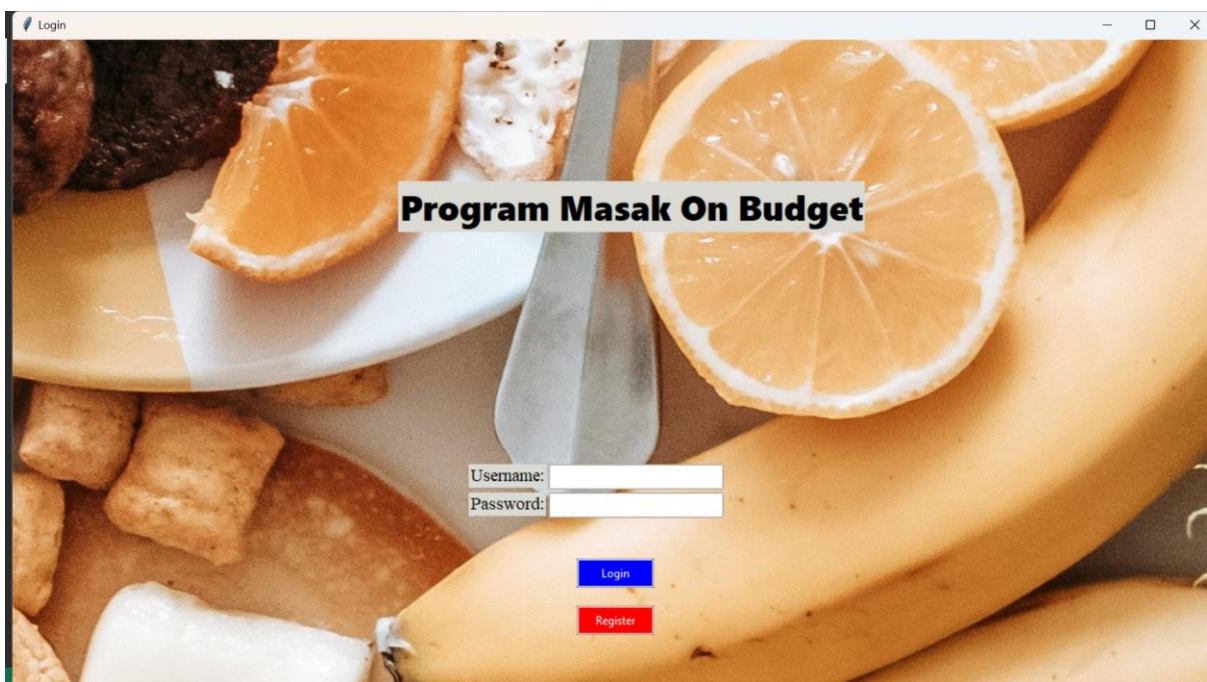
Gambar 4.10 Tampilan Menu Utama

Ketika kembali ke tampilan menu utama *user* dapat memilih untuk mencari menu lagi atau tidak. Jika mencari resep lagi, *user* dapat menekan *button* 'Cari Lagi'.



Gambar 4.11. Tampilan Menu Utama

Jika user tidak ingin mencari resep lagi, user dapat menekan button 'Logout'



Gambar 4.12 Kembali ke Tampilan Login/Registrasi

Lalu tampilan akan kembali pada tampilan login/register.