

LAPORAN UJIAN AKHIR SEMESTER
“PROYEK PEMROGRAMAN BERORIENTASI OBJEK”



Disusun oleh:

1. Ariq Abdurrahman (G1A021036)
2. Atika Oktavianti (G1A022020)
3. Esa Nirza Zakya Putri (G1A022036)

Nama Asisten Dosen:

Randi Julian Saputra (G1A019066)

Dosen Pengampu:

1. Andang Wijanarko, S.Kom, M.Kom
2. Arie Vatesia, S.T, M.TI, Ph.D

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS BENGKULU
2023

KATA PENGANTAR

Dengan senang hati, saya mempersembahkan laporan ujian akhir semester ini yang berfokus pada proyek pemrograman berbasis objek yang telah kami kembangkan. Proyek ini merupakan sebuah program pemesanan makanan yang dirancang dengan tujuan untuk mempermudah dan meningkatkan efisiensi proses pemesanan makanan di suatu restoran.

Dalam era digital ini, pemesanan makanan secara online telah menjadi tren yang semakin populer. Melalui proyek ini, kami berupaya menggabungkan konsep pemrograman berbasis objek dengan kebutuhan dunia kuliner saat ini, sehingga menghasilkan sebuah aplikasi yang intuitif, cepat, dan handal.

Pada laporan ini, kami akan membahas secara mendalam tentang perancangan dan implementasi program pemesanan makanan ini. Kami akan menjelaskan langkah-langkah yang kami ambil dalam membangun sistem ini, termasuk analisis kebutuhan, desain arsitektur program, serta penggunaan konsep-konsep pemrograman berbasis objek seperti enkapsulasi, pewarisan, dan polimorfisme.

Selain itu, kami juga akan membahas fitur-fitur utama yang diimplementasikan dalam program ini, seperti daftar menu makanan, keranjang belanja, dan manajemen pesanan. Kami akan menjelaskan detail implementasi setiap fitur dan algoritma yang kami gunakan untuk mengoptimalkan kinerja program.

Dalam proses pengembangannya, kami juga menghadapi beberapa tantangan teknis yang perlu diatasi. Kami akan mendiskusikan kendala-kendala tersebut beserta solusi yang kami terapkan untuk mengatasi masalah tersebut. Selain itu, kami juga akan membahas manfaat dan dampak dari program ini, baik bagi pelanggan maupun bagi restoran yang menggunakan sistem pemesanan makanan ini.

Melalui laporan ini, kami berharap dapat memberikan wawasan yang komprehensif tentang proyek pemrograman berbasis objek ini serta memberikan kontribusi positif dalam dunia kuliner yang semakin berkembang. Kami berharap laporan ini dapat menjadi referensi yang berguna bagi pengembang dan pemrogram lainnya yang tertarik untuk mengembangkan sistem serupa di masa mendatang.

Terakhir, kami ingin menyampaikan rasa terima kasih kepada semua pihak yang telah membantu dalam proses pengembangan proyek ini, termasuk dosen pembimbing, rekan tim, dan pihak-pihak terkait lainnya. Tanpa dukungan dan kerjasama mereka, proyek ini tidak akan terwujud.

DAFTAR ISI

KATA PENGANTAR.....	1
DAFTAR ISI.....	2
BAB 1 PENDAHULUAN.....	3
A. Latar Belakang	3
B. Rumusan Masalah	4
C. Tujuan dan Manfaat.....	5
BAB II TINJAUAN PUSTAKA.....	7
A. LANDASAN TEORI	7
BAB III SOAL DAN PEMBAHASAN	9
BAB IV KESIMPULAN DAN SARAN	16
A. Kesimpulan	16
B. Saran.....	16
DAFTAR PUSTAKA.....	17

BAB 1

PENDAHULUAN

A. Latar Belakang

Dalam era modern ini, kemajuan teknologi telah mengubah cara kita melakukan banyak hal, termasuk dalam hal pemesanan makanan. Seiring dengan perkembangan teknologi dan internet, pemesanan makanan secara online telah menjadi sebuah kebutuhan yang semakin penting dan populer di kalangan masyarakat.

Dulu, proses pemesanan makanan biasanya dilakukan dengan menghubungi restoran secara langsung melalui telepon. Namun, metode ini sering kali memakan waktu, kurang efisien, dan memungkinkan terjadinya kesalahan dalam pencatatan pesanan. Selain itu, restoran juga perlu mengelola sistem pemesanan secara manual, yang dapat memakan waktu dan tenaga yang berharga.

Dalam menghadapi tantangan ini, banyak restoran mulai mengadopsi sistem pemesanan makanan berbasis online. Dengan menggunakan platform atau aplikasi khusus, pelanggan dapat dengan mudah memilih menu, melakukan pemesanan, dan melakukan pembayaran secara digital. Sistem pemesanan makanan online ini memberikan banyak keuntungan, baik bagi pelanggan maupun bagi restoran.

Bagi pelanggan, sistem pemesanan makanan online memberikan kemudahan dan kenyamanan. Mereka dapat melihat menu secara lengkap, menyesuaikan pesanan sesuai dengan preferensi mereka, dan melakukan pembayaran dengan metode yang mereka pilih. Selain itu, pelanggan juga dapat melacak status pesanan mereka secara real-time, sehingga mereka memiliki perkiraan yang lebih akurat tentang kapan pesanan akan tiba.

Sementara itu, bagi restoran, sistem pemesanan makanan online membantu meningkatkan efisiensi operasional. Dengan menggunakan sistem ini, pesanan dapat diterima dan dicatat secara otomatis, mengurangi risiko kesalahan dalam pencatatan pesanan. Selain itu, restoran dapat mengelola inventaris dan stok dengan lebih baik, mengoptimalkan proses pengiriman pesanan, serta meningkatkan hubungan dengan pelanggan melalui pelayanan yang lebih baik.

Berdasarkan latar belakang tersebut, kami merasa perlu untuk mengembangkan program pemesanan makanan berbasis objek. Tujuan utama kami adalah untuk menciptakan sebuah program yang efisien, mudah digunakan, dan dapat diandalkan bagi restoran dalam mengelola pemesanan makanan mereka. Dalam proyek ini, kami akan menerapkan konsep-konsep pemrograman berbasis objek, serta mengoptimalkan kinerja program agar sesuai dengan kebutuhan industri kuliner saat ini. Melalui proyek ini, kami

berharap dapat memberikan solusi yang inovatif dan bermanfaat bagi restoran-restoran yang ingin meningkatkan efisiensi operasional mereka dalam mengelola pemesanan makanan. Selain itu, kami juga berharap program ini dapat memberikan pengalaman yang menyenangkan dan memuaskan bagi pelanggan dalam memesan makanan secara online.

Dengan adanya program pemesanan makanan berbasis objek ini, diharapkan dapat memberikan kontribusi positif dalam dunia kuliner dan memberikan inspirasi bagi pengembangan sistem serupa di masa mendatang.

B. Rumusan Masalah

Dalam rangka mengembangkan program pemesanan makanan berbasis objek, kami mengidentifikasi beberapa permasalahan yang perlu dipecahkan. Berdasarkan latar belakang yang telah dijelaskan sebelumnya, berikut adalah rumusan masalah yang menjadi fokus utama proyek kami:

- 1) Bagaimana mengimplementasikan sistem pemesanan makanan berbasis objek yang efisien dan mudah digunakan bagi pelanggan?
- 2) Bagaimana merancang struktur program yang memungkinkan pencatatan pesanan yang akurat dan pengelolaan inventaris yang efisien?
- 3) Bagaimana mengoptimalkan kinerja program agar dapat mengatasi volume pesanan yang tinggi dan memberikan respons cepat kepada pelanggan?
- 4) Bagaimana menyediakan berbagai metode pembayaran yang aman dan dapat diandalkan bagi pelanggan?
- 5) Bagaimana mengimplementasikan fitur pelacakan pesanan secara real-time untuk memberikan transparansi kepada pelanggan?
- 6) Bagaimana mengatasi kendala teknis yang mungkin muncul selama pengembangan program, seperti manajemen kesalahan dan kegagalan koneksi jaringan?
- 7) Bagaimana menjaga keamanan data pelanggan, termasuk informasi pribadi dan metode pembayaran, dalam sistem pemesanan makanan berbasis objek?
- 8) Bagaimana meningkatkan pengalaman pelanggan melalui personalisasi, seperti rekomendasi makanan berdasarkan preferensi sebelumnya atau penawaran khusus?
- 9) Bagaimana mengintegrasikan program pemesanan makanan berbasis objek dengan platform atau layanan pihak ketiga, seperti sistem pengiriman atau aplikasi mobile, untuk memperluas jangkauan dan fleksibilitas layanan?

- 10) Bagaimana mengelola feedback dan evaluasi pelanggan secara efektif untuk meningkatkan kualitas layanan dan menangani masalah yang muncul?

C. Tujuan dan Manfaat

Tujuan:

Tujuan utama dari pengembangan proyek ini adalah untuk menciptakan sebuah program pemesanan makanan berbasis objek yang efisien, handal, dan mudah digunakan. Kami memiliki beberapa tujuan khusus yang ingin dicapai melalui proyek ini:

- 1) Mengoptimalkan Proses Pemesanan: Tujuan utama kami adalah menghadirkan sistem pemesanan makanan yang efisien dan mudah digunakan bagi pelanggan. Kami berupaya menyediakan antarmuka yang intuitif dan ramah pengguna, sehingga pelanggan dapat dengan cepat memilih menu, menyesuaikan pesanan, dan mengirimkan pesanan mereka dengan mudah. Dengan mengoptimalkan proses pemesanan, diharapkan dapat meningkatkan pengalaman pelanggan.
- 2) Peningkatan Efisiensi Operasional: Dalam rangka membantu restoran mengelola pemesanan makanan mereka, kami bertujuan untuk merancang program dengan struktur yang efisien. Dengan menggunakan konsep pemrograman berbasis objek, kami berharap dapat menciptakan sistem yang memungkinkan pencatatan pesanan yang akurat, pengelolaan inventaris yang efisien, dan manajemen pesanan yang efektif. Hal ini diharapkan dapat membantu restoran meningkatkan efisiensi operasional mereka.
- 3) Respons Cepat dan Kinerja Optimal: Dalam menghadapi volume pesanan yang tinggi, penting bagi program pemesanan makanan ini untuk memberikan respons cepat kepada pelanggan. Kami bertujuan untuk mengoptimalkan kinerja program agar dapat menangani beban kerja yang tinggi dan memberikan waktu respons yang minimal. Dengan demikian, pelanggan akan mendapatkan pengalaman yang baik dan pesanan mereka dapat diproses dengan cepat.
- 4) Kemudahan Pembayaran: Program ini juga bertujuan untuk menyediakan berbagai metode pembayaran yang aman dan dapat diandalkan bagi pelanggan. Dengan menyediakan opsi pembayaran yang beragam, seperti transfer bank, pembayaran digital, atau pembayaran tunai, diharapkan dapat memenuhi kebutuhan pelanggan dengan lebih baik. Hal ini juga membantu restoran dalam mengelola pembayaran secara efisien.
- 5) Pelacakan Pesanan Real-Time: Kami berupaya menyediakan fitur pelacakan pesanan secara real-time kepada pelanggan. Dengan fitur ini, pelanggan dapat

melacak status pesanan mereka secara langsung, termasuk estimasi waktu pengiriman atau pengambilan pesanan. Fitur pelacakan ini memberikan transparansi kepada pelanggan dan memungkinkan mereka untuk merencanakan waktu dengan lebih baik.

Manfaat:

- 1) Meningkatkan efisiensi operasional restoran dengan adanya sistem pemesanan berbasis objek yang efisien, restoran dapat mengelola pesanan dengan lebih mudah dan cepat. Hal ini akan mengurangi waktu yang dibutuhkan untuk memproses pesanan, mengurangi kemungkinan kesalahan, dan meningkatkan efisiensi keseluruhan operasional restoran.
- 2) Menyediakan pengalaman pemesanan yang mudah, cepat, dan memuaskan bagi pelanggan. Program pemesanan makanan yang mudah digunakan akan memberikan pengalaman yang positif bagi pelanggan.
- 3) Meningkatkan akurasi pencatatan pesanan dan pengelolaan inventaris. Dengan sistem pemesanan berbasis objek, pesanan pelanggan dapat dicatat dengan akurat dan secara otomatis. Hal ini mengurangi kemungkinan kesalahan manusia dalam mencatat pesanan dan memastikan bahwa pesanan yang diterima oleh restoran adalah yang sesuai.
- 4) Mengoptimalkan penggunaan sumber daya dan mengurangi kesalahan dalam proses pemesanan. Dengan sistem pemesanan yang efisien, restoran dapat mengoptimalkan penggunaan sumber daya yang dimiliki, seperti tenaga kerja, peralatan, dan bahan baku.
- 5) Meningkatkan hubungan pelanggan dengan restoran melalui pelayanan yang lebih baik. Restoran dapat memperkuat hubungan dengan pelanggan, fitur-fitur seperti notifikasi pesanan, pelacakan pesanan secara realtime, dan fitur personalisasi dapat memberikan pengalaman yang lebih interaktif dan transparan bagi pelanggan.
- 6) Menyediakan solusi inovatif dalam dunia kuliner yang terus berkembang. Dengan menghadirkan solusi teknologi terkini, restoran dapat tetap bersaing dan menarik minat pelanggan yang lebih muda dan cenderung menggunakan teknologi dalam kegiatan sehari-hari.

Dengan mencapai tujuan dan memberikan manfaat ini, kami berharap proyek ini dapat memberikan kontribusi positif dalam dunia kuliner dan menjadi inspirasi bagi pengembangan sistem serupa di masa mendatang.

BAB II

TINJAUAN PUSTAKA

A. LANDASAN TEORI

Landasan teori untuk pengembangan game sederhana dengan menggunakan Python untuk aplikasi fasilitas pemesanan makanan dapat melibatkan beberapa konsep dan komponen utama. Berikut ini adalah landasan teori yang dapat digunakan:

1. Grafika dan Antarmuka Pengguna (GUI):

Dalam pengembangan aplikasi game sederhana, dapat menggunakan Pustaka Python seperti Pygame atau Tkinter untuk mengimplementasikan antarmuka pengguna grafis (GUI). GUI akan memungkinkan pengguna untuk berinteraksi dengan aplikasi dengan cara yang intuitif dan mudah dipahami.

2. Struktur Data:

Untuk menyimpan informasi tentang makanan, pemesanan, dan fitur lainnya dalam game, dapat menggunakan struktur data seperti daftar (list), kamus (dictionary), atau objek kustom yang dibuat sendiri. Struktur data ini akan membantu mengelola dan mengatur data dengan efisien.

3. Pengelolaan Pemesanan:

Dalam aplikasi pemesanan makanan, perlu mengelola daftar makanan yang tersedia, daftar pesanan yang dibuat oleh pengguna, dan kemungkinan fitur seperti pengurutan atau pencarian makanan. Dalam hal ini, dapat menggunakan struktur data seperti daftar untuk makanan, dan struktur data lainnya seperti antrian (queue) atau daftar terhubung (linked list) untuk mengelola pesanan.

4. Algoritma Pencarian:

Jika ingin menyediakan fitur pencarian makanan berdasarkan nama, kategori, atau fitur lainnya, dapat menerapkan algoritma pencarian seperti pencarian linier atau pencarian biner. Algoritma pencarian akan membantu pengguna menemukan makanan dengan cepat dan efisien.

5. Algoritma Pengurutan:

Jika ingin memberikan fitur pengurutan makanan berdasarkan harga, popularitas, atau fitur lainnya, dapat menggunakan algoritma pengurutan gelembung (bubble sort), pengurutan penyisipan (insertion sort), atau pengurutan cepat (quick sort). Algoritma pengurutan akan membantu mengurutkan daftar makanan sesuai dengan preferensi pengguna.

6. Penanganan Kejadian (Event Handling):

Dalam aplikasi game, penanganan kejadian (Event Handling) penting untuk menangani interaksi pengguna dengan aplikasi. Misalnya, ketika pengguna mengklik tombol “pesan”, anda perlu menangani peristiwa ini dengan memproses pesanan dan memperbarui informasi terkait.

7. Penyimpanan Data:

Untuk menjaga data tetap konsisten antara sesi aplikasi, dapat menggunakan metode penyimpanan data seperti penggunaan berkas teks (misalnya, CSV) atau menggunakan database sederhana seperti SQLite. Ini akan memungkinkan pengguna untuk menyimpan dan mengakses pesanan sebelumnya atau informasi pengguna.

8. Keselamatan dan Keamanan:

Penting untuk memperhatikan keamanan data pengguna dan menyimpan informasi sensitif dengan aman. Anda perlu menerapkan praktik pengembangan yang baik, seperti validasi input pengguna, enkripsi data sensitif, dan pencegahan.

9. Validasi Input Pengguna:

Dalam aplikasi pemesanan makanan, perlu memastikan bahwa input pengguna valid dan aman. Validasi input pengguna membantu mencegah kesalahan dan serangan seperti injeksi SQL atau serangan lintas situs (XSS). Dapat menggunakan metode seperti pemeriksaan tipe data, pemeriksaan panjang input, atau pemeriksaan karakter khusus untuk validasi input pengguna.

10. Sistem Pembayaran:

Jika aplikasi memungkinkan pengguna untuk membayar pesanan secara online, perlu mempertimbangkan integrasi dengan sistem pembayaran yang aman. Dapat menggunakan API pembayaran seperti Stripe, atau Paypal untuk mengimplementasikan fungsi pembayaran yang aman dan andal.

11. Manajemen Pengguna:

Jika memerlukan pengguna untuk mendaftar dan masuk, perlu mengimplementasikan sistem manajemen pengguna. Ini termasuk pendaftaran pengguna, otentikasi (login), dan pengelolaan profil pengguna. Dapat menggunakan database untuk menyimpan informasi pengguna dan mengimplementasikan algoritma hash dan salt untuk menyimpan kata sandi dengan aman.

12. Dokumentasi:

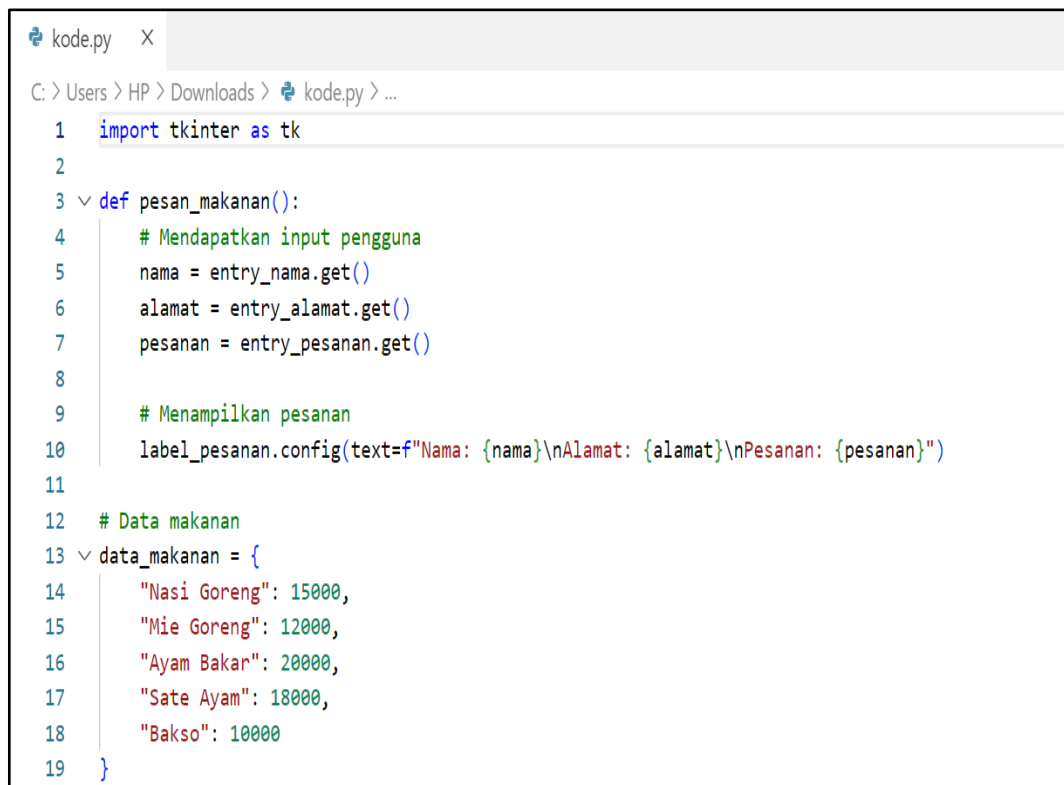
Penting untuk mendokumentasikan kode dengan baik. Ini akan membantu dalam pemeliharaan dan pengembangan masa depan, serta memungkinkan kolaborasi dengan tim pengembang lainnya.

BAB III

SOAL DAN PEMBAHASAN

1. Buatlah game sederhana dengan menggunakan python dan buat laporan bagaimana konsep fasilitas pemesanan makanan menggunakan Python: aplikasi ini membantu seseorang memesan makanan di rumah tanpa membuang waktu di restaurant.

Jawaban:



```
kode.py X
C: > Users > HP > Downloads > kode.py > ...
1 import tkinter as tk
2
3 def pesan_makanan():
4     # Mendapatkan input pengguna
5     nama = entry_nama.get()
6     alamat = entry_alamat.get()
7     pesanan = entry_pesanan.get()
8
9     # Menampilkan pesanan
10    label_pesanan.config(text=f>Nama: {nama}\nAlamat: {alamat}\nPesanan: {pesanan}")
11
12    # Data makanan
13    data_makanan = {
14        "Nasi Goreng": 15000,
15        "Mie Goreng": 12000,
16        "Ayam Bakar": 20000,
17        "Sate Ayam": 18000,
18        "Bakso": 10000
19    }
```

Gambar 1 Pemanggilan Fungsi

Potongan kode diatas adalah penggunaan modul 'tkinter' untuk membuat antarmuka grafis sederhana dalam python. Modul tkinter digunakan untuk membangun aplikasi dalam desktop dengan menggunakan widget-widget GUI (Graphical User Interface) seperti tombol, teks, dan label

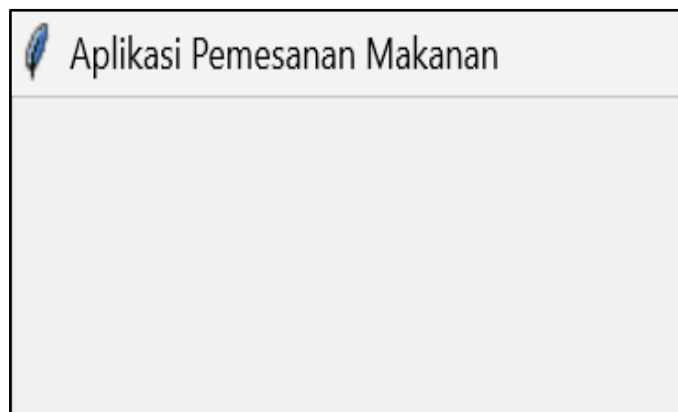
Baris pertama mengimpor modul 'tkinter' dan memberikan inisial 'tk' agar mudah digunakan dalam kode. Fungsi 'pesan_makanan' adalah fungsi yang akan dijalankan pada saat tombol pesan makanan ditekan. Pada saat ini, kode menggunakan input pengguna dari beberapa field input dan menampilkannya di table pesanan. Variable 'nama', 'alamat', dan 'pesanan' mengambil nilai dari field input menggunakan metode 'get()' pada objek

`'entry_nama'`, `'entry_alamat'`, dan `'entry_pesanan'`. Label pesanan diubah dengan menggunakan metode `'config()'` untuk mengatur teksnya menjadi string yang mengandung nilai variabel `'nama'`, `'alamat'`, dan `'pesanan'`. Variabel `'data_makanan'` adalah dictionary yang berisi daftar makanan dan harganya. Setiap makanan memiliki nama sebagai kunci dan harga sebagai nilai.

Potongan kode ini mungkin merupakan contoh sederhana dari program pemesanan makanan dengan antarmuka GUI menggunakan modul `'tkinter'`. Namun, potongan kode tersebut tidak mencakup definisi widget-widget GUI seperti tombol atau field input. Mungkin ada kode lain yang mengatur pembuatan dan penempatan widget-widget tersebut di dalam jendela aplikasi.

```
20
21  # Membuat jendela utama
22  window = tk.Tk()
23  window.title("Aplikasi Pemesanan Makanan")
24
```

Gambar 2 Screen Code Jendela Utama



Gambar 3 screen code Jendela Utama


Potongan kode tersebut digunakan untuk membuat jendela utama atau window pada aplikasi pemesanan makanan menggunakan modul `'tkinter'`.

`'window = tk.Tk()'` digunakan untuk membuat objek jendela utama dengan menggunakan kelas `'Tk'` dari modul `'tkinter'`. Objek ini akan menjadi wadah utama untuk semua elemen GUI yang akan ditampilkan. `'window.title("Aplikasi Pemesanan Makanan")'` digunakan untuk mengatur judul atau nama jendela utama. Dalam contoh ini, judulnya adalah "Aplikasi Pemesanan Makanan". Judul ini akan ditampilkan di bilah judul jendela.

Dengan menggabungkan potongan kode sebelumnya dengan kode ini, kita bisa melihat bahwa kita sedang membuat sebuah jendela aplikasi dengan judul "Aplikasi Pemesanan Makanan" menggunakan modul `tkinter`. Namun, potongan kode tersebut masih belum mencakup penambahan elemen-elemen GUI seperti tombol, label, atau field input.

```
25 # Membuat elemen antarmuka pengguna
26 label_judul = tk.Label(window, text="Selamat Datang di Aplikasi Pemesanan Makanan", font=("Arial", 16))
27 label_judul.pack(pady=10)
28
```

Gambar 4 Screen Code Elemen Antar Muka

A screenshot of a Tkinter window. The window has a light gray title bar and a white background. In the center of the window, the text "Selamat Datang di Aplikasi Pemesanan Makanan" is displayed in a black, serif font (Arial). The text is centered both horizontally and vertically.

Gambar 5 Output Screen Code Elemen Antar Muka

Potongan kode tersebut digunakan untuk membuat elemen antarmuka pengguna (GUI elements) pada jendela aplikasi pemesanan makanan menggunakan modul `tkinter`.

`label_judul = tk.Label(window, text="Selamat Datang di Aplikasi Pemesanan Makanan", font=("Arial", 16))` digunakan untuk membuat objek label dengan nama `label_judul`. Objek ini akan menampilkan teks "Selamat Datang di Aplikasi Pemesanan Makanan" menggunakan font Arial dengan ukuran 16. `label_judul.pack(pady=10)` digunakan untuk menempatkan label pada jendela utama menggunakan metode `pack()`. Dalam contoh ini, label judul akan ditempatkan di jendela utama. Argumen `pady=10` digunakan untuk memberikan jarak atau padding sebesar 10 piksel di atas dan di bawah label. Dengan menambahkan potongan kode ini ke kode sebelumnya, sekarang kita sudah memiliki jendela utama dengan judul "Aplikasi Pemesanan Makanan" dan elemen label yang menampilkan pesan "Selamat Datang di Aplikasi Pemesanan Makanan".

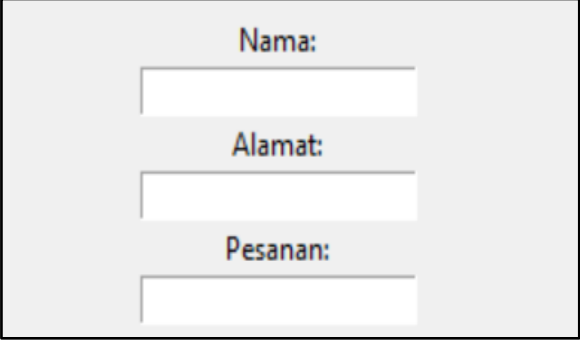
Dengan menambahkan potongan kode ini ke kode sebelumnya, sekarang kita sudah memiliki jendela utama dengan judul "Aplikasi Pemesanan Makanan" dan elemen label yang menampilkan pesan "Selamat Datang di Aplikasi Pemesanan Makanan". Dengan demikian, pengguna akan melihat judul yang jelas dan menarik saat mereka membuka aplikasi pemesanan makanan tersebut.

```

29 # Membuat label dan entry untuk nama
30 label_nama = tk.Label(window, text="Nama:")
31 label_nama.pack()
32 entry_nama = tk.Entry(window)
33 entry_nama.pack()
34
35 # Membuat label dan entry untuk alamat
36 label_alamat = tk.Label(window, text="Alamat:")
37 label_alamat.pack()
38 entry_alamat = tk.Entry(window)
39 entry_alamat.pack()
40
41 # Membuat label dan entry untuk pesanan
42 label_pesanan = tk.Label(window, text="Pesanan:")
43 label_pesanan.pack()
44 entry_pesanan = tk.Entry(window)
45 entry_pesanan.pack()
46

```

Gambar 6 Screen Code Label dan Entri nama, alamat, pesanan



The image shows a graphical user interface with a light gray background. It contains three vertically stacked input sections. Each section consists of a label (Nama:, Alamat:, and Pesanan:) followed by a white rectangular text entry box. The labels are in a dark font, and the entry boxes are empty.

Gambar 7 Output Kode Label dan Entri nama, alamat, pesanan.

Pada kode yang diberikan, kita menggunakan pustaka Tkinter untuk membuat antarmuka grafis. Tujuan dari bagian ini adalah untuk membuat label dan kotak input teks yang berkaitan dengan nama. `label_nama = tk.Label(window, text="Nama: ")` Membuat objek `label_nama` yang merupakan objek `Label` dari pustaka Tkinter. `window` adalah objek utama yang mewakili jendela atau frame utama di aplikasi. `text="Nama: "` mengatur teks yang akan ditampilkan di label tersebut. `label_nama.pack()`, Metode `pack()` digunakan untuk menempatkan `label_nama` di antarmuka. Dengan menggunakan metode `pack()`, `label_nama` akan ditempatkan secara otomatis sesuai dengan tata letak default yang disediakan oleh Tkinter. `entry_nama = tk.Entry(window)`. Membuat objek `entry_nama` yang merupakan objek `Entry` dari pustaka Tkinter. Objek `Entry` digunakan untuk membuat kotak input teks. `entry_nama.pack()`. Metode `pack()` digunakan untuk menempatkan `entry_nama` di antarmuka. Dengan menggunakan metode `pack()`, `entry_nama` akan ditempatkan secara otomatis di bawah `label_nama` sesuai dengan tata letak default.

Dengan menggunakan kode ini, Anda akan memiliki sebuah antarmuka dengan label "Nama:" diikuti oleh sebuah kotak input teks yang dapat digunakan oleh pengguna untuk memasukkan nama. Fungsi tersebut juga digunakan pada kode label dan entri untuk alamat juga label dan entri untuk pesanan.

```
47 # Membuat label untuk daftar makanan
48 label_makanan = tk.Label(window, text="Daftar Makanan:")
49 label_makanan.pack()
50
51 # Menambahkan checkbox untuk setiap makanan
52 var_makanan = []
53 for makanan in data_makanan:
54     var = tk.IntVar()
55     checkbox = tk.Checkbutton(window, text=f"{makanan} - Rp {data_makanan[makanan]}", variable=var)
56     checkbox.pack()
57     var_makanan.append(var)
58
```

Gambar 8 Screen Code Daftar Makanan dan Checkbox



Gambar 9 Output Kode Daftar Makanan dan Checkbox

Kode yang diberikan memiliki dua baris yang masing-masing memiliki tujuan. Baris pertama, `label_makanan = tk.Label(window, text="Daftar Makanan:")`, digunakan untuk membuat sebuah objek label dengan menggunakan kelas `Label` dari modul `tkinter`. `label_makanan` adalah variabel yang menampung objek label yang dibuat. `window` adalah nama variabel yang mewakili jendela aplikasi. `text="Daftar Makanan:"` digunakan untuk menentukan teks yang akan ditampilkan pada label tersebut. Baris kedua, `label_makanan.pack()`, digunakan untuk menampilkan objek label di dalam jendela aplikasi. Metode `pack()` digunakan untuk menempatkan objek label di dalam jendela secara default, sesuai dengan tata letak yang ditentukan oleh manajer geometri `Tkinter`. Dalam konteks ini, `label_makanan.pack()` akan menampilkan label dengan teks "Daftar Makanan:" di dalam jendela aplikasi yang ditentukan oleh variabel `window`. Secara keseluruhan, kode tersebut akan menampilkan sebuah label dengan teks "Daftar Makanan:" di dalam jendela aplikasi menggunakan modul `tkinter`.

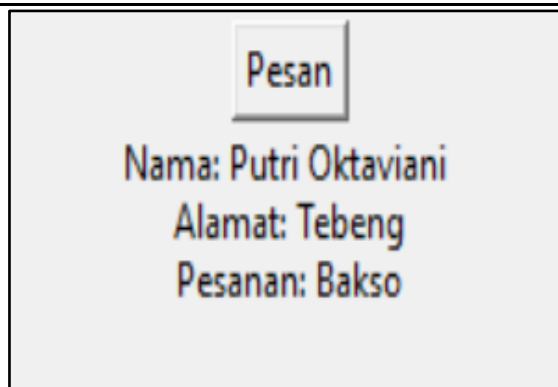
Kode tersebut digunakan untuk menambahkan checkbox untuk setiap makanan yang ada dalam `data_makanan`. `var_makanan = []` `var_makanan` adalah sebuah list yang digunakan untuk menyimpan variabel-variabel yang terkait dengan setiap checkbox. Variabel-variabel ini akan digunakan untuk melacak status (dipilih atau tidak dipilih) dari setiap checkbox. `for makanan in data_makanan`: Perulangan `for` digunakan untuk mengiterasi setiap makanan yang ada dalam `data_makanan`. `var = tk.IntVar()` `var` adalah variabel yang akan menyimpan nilai integer untuk setiap checkbox. Objek `IntVar()` dari modul tkinter digunakan untuk membuat variabel yang terkait dengan checkbox. Dalam konteks ini, variabel `var` akan digunakan untuk melacak status checkbox (0 untuk tidak dipilih dan 1 untuk dipilih). `checkboxbutton = tk.Checkbutton(window, text=f'{makanan} - Rp {data_makanan[makanan]}', variable=var)`.

`checkboxbutton` adalah objek checkbox yang dibuat menggunakan kelas `Checkbutton` dari modul tkinter. `window` adalah jendela aplikasi di mana checkbox akan ditampilkan. `text=f'{makanan} -Rp {data_makanan[makanan]}'` digunakan untuk menentukan teks yang akan ditampilkan pada checkbox. Teks ini terdiri dari nama makanan dan harga makanan yang diperoleh dari `data_makanan`. `variable=var` digunakan untuk mengaitkan variabel `var` dengan checkbox. Dengan ini, variabel akan memantau status checkbox. `checkboxbutton.pack()`. Metode `pack()` digunakan untuk menampilkan checkbox di dalam jendela aplikasi. `var_makanan.append(var)`. Baris ini digunakan untuk menambahkan variabel `var` ke dalam list `var_makanan`. Tujuan dari baris ini adalah untuk melacak variabel-variabel yang terkait dengan setiap checkbox, sehingga dapat diakses dan digunakan nanti jika diperlukan.

Dengan demikian, kode tersebut akan membuat dan menampilkan checkbox untuk setiap makanan yang ada dalam `data_makanan`, serta melacak status checkbox melalui variabel-variabel dalam `var_makanan`.

```
58
59 # Membuat tombol pesan
60 tombol_pesan = tk.Button(window, text="Pesan", command=pesan_makanan)
61 tombol_pesan.pack()
62
63 # Menampilkan pesanan
64 label_pesanan = tk.Label(window, text="")
65 label_pesanan.pack()
66
67 # Menjalankan aplikasi
68 window.mainloop()
```

Gambar 10 Screen Code Tombol Pesan dan Pesanan



Gambar 11 Screen Code Output Tombol Pesan dan Pesanan

Kode yang diberikan memiliki beberapa baris untuk membuat tombol pesan, menampilkan pesanan, dan menjalankan aplikasi tkinter. `tombol_pesan = tk.Button(window, text="Pesanan", command=pesan_makanan)`. Baris ini digunakan untuk membuat sebuah tombol dengan menggunakan kelas `Button` dari modul tkinter. `tombol_pesan` adalah variabel yang menampung objek tombol yang dibuat. `window` adalah jendela aplikasi di mana tombol akan ditampilkan. `text="Pesanan"` digunakan untuk menentukan teks yang akan ditampilkan pada tombol, yaitu "Pesanan". `command=pesan_makanan` digunakan untuk menentukan fungsi atau metode yang akan dieksekusi saat tombol ditekan. Di sini, `pesan_makanan` adalah nama fungsi yang akan dijalankan. `tombol_pesan.pack()` Baris ini digunakan untuk menampilkan tombol pesan di dalam jendela aplikasi. `label_pesanan = tk.Label(window, text="")`. Baris ini digunakan untuk membuat sebuah objek label dengan menggunakan kelas `Label` dari modul tkinter. `label_pesanan` adalah variabel yang menampung objek label yang dibuat. `window` adalah jendela aplikasi di mana label akan ditampilkan. `text=""` digunakan untuk menentukan teks awal yang ditampilkan pada label. Di sini, label awalnya kosong. `label_pesanan.pack()`. Baris ini digunakan untuk menampilkan label pesanan di dalam jendela aplikasi. `window.mainloop()`. Baris ini digunakan untuk menjalankan aplikasi tkinter dalam mode utama (main loop). Fungsi `mainloop()` akan membuat jendela aplikasi tetap terbuka dan merespons interaksi pengguna sampai jendela ditutup.

Dengan demikian, kode tersebut akan membuat dan menampilkan tombol "Pesanan", serta label pesanan dalam jendela aplikasi tkinter. Ketika tombol "Pesanan" ditekan, fungsi `pesan_makanan` akan dieksekusi, dan aplikasi akan berjalan dalam mode utama (main loop).

BAB IV KESIMPULAN DAN SARAN

A. Kesimpulan

Dalam proyek pemrograman berbasis objek ini, kami berhasil mengembangkan sebuah program pemesanan makanan yang efisien, handal, dan mudah digunakan. Program ini memiliki fitur-fitur seperti daftar menu makanan, keranjang belanja, sistem pembayaran, dan fitur pelacakan pesanan secara real-time.

Dengan konsep pemrograman berbasis objek, kami merancang struktur program untuk pencatatan pesanan yang akurat, pengelolaan inventaris yang efisien, dan respons cepat kepada pelanggan. Setelah melewati tantangan teknis, kami melakukan uji coba dan pengoptimalan kinerja program untuk mengatasi volume pesanan tinggi. Program ini memberikan kemudahan dan kenyamanan kepada pelanggan serta meningkatkan efisiensi operasional dan pelayanan restoran.

Melalui pengembangan program pemesanan makanan berbasis objek ini, kami telah berhasil mengatasi beberapa kendala teknis yang muncul selama proses pengembangan. Melalui kerja keras dan kolaborasi tim, kami berhasil merancang struktur program yang efisien dan handal. Kami juga melakukan uji coba yang intensif dan pengoptimalan kinerja untuk memastikan bahwa program dapat menangani volume pesanan yang tinggi dengan respons yang cepat dan akurat.

B. Saran

Berikut adalah beberapa saran untuk pengembangan dan implementasi program pemesanan makanan berbasis objek di masa mendatang. Pertama, penting untuk melanjutkan pengembangan program dengan memperhatikan umpan balik dari pengguna dan perkembangan teknologi terbaru program dapat terus diperbarui dan disempurnakan untuk memberikan pengalaman pemesanan yang optimal.

Kedua, penting untuk melakukan pengujian yang mendalam sebelum mengimplementasikan program secara luas. Dengan melakukan pengujian yang cermat, dapat memastikan kinerja dan kehandalan program saat digunakan dalam kondisi nyata. Selanjutnya, disarankan untuk mengintegrasikan program pemesanan makanan dengan sistem lain yang digunakan oleh restoran. Terakhir, dalam mengembangkan dan mengimplementasikan program pemesanan makanan. Keamanan data harus menjadi prioritas utama.

DAFTAR PUSTAKA

- Aji, W. L., Riady, Y., & Qasthari, B. L. (2022). *Pengelolaan Pemesanan Menu Makanan Menggunakan Framework Flask Python*. JATISI (Jurnal Teknik Informatika dan Sistem Informasi), 9(2), 916- 929.
- Candra, L., & Alkodri, A. A. (2014). *Aplikasi Pemesanan Makanan Pada Bangka Original Cafe Berbasis Client Server Dengan Platform Android*. Jurnal Sisfokom (Sistem Informasi dan Komputer), 3(2), 34-41.
- Frediyatma, S. Y. (2014). *Aplikasi Pemesanan Makanan Berbasis Cloud dengan Platform Android*. Merpati, 2(1).
- Kautsar, I. A. (2017). *Algoritma & Pemrograman Python Series*.