

Implementasi Sistem Pengelolaan Pesanan Menu Restoran Berbasis Stack dan Queue

Agata Putri Handayani Simbolon^{1)*}, Khairul Fahmi Sagala²⁾, Muhammad Raffi Akbar Tanjung³⁾, Tri Saptia Warman Zai⁴⁾, Fanny Ramadhani⁵⁾

¹⁾²⁾³⁾⁴⁾⁵⁾ Universitas Negeri Medan

Jl. William Iskandar Ps. V, Kenangan Baru, Kec. Percut Sei Tuan, Kabupaten Deli Serdang, Sumatera Utara, Indonesia

¹⁾agataputri.4233550042@mhs.unimed.ac.id

²⁾fhmisagala.4231250020@mhs.unimed.ac.id

³⁾mraffiakbar2005.4233550002@mhs.unimed.ac.id

⁴⁾muridholmeswatson@mhs.unimed.ac.id

⁵⁾fannyr@unimed.ac.id

Article history:

Received xx xxx 20xx;
Revised xx xxx 20xx;
Accepted xx xxx 20xx;
Available online xx xxx 20xx

Keywords:

FIFO
LIFO
Queue
Sistem Pengelolaan Pesanan
Stack

Abstrak

Di industri restoran, memiliki sistem pengelolaan pesanan yang efisien sangat penting untuk meningkatkan kualitas layanan dan kepuasan pelanggan. Namun, sebagian besar restoran masih bergantung pada metode manual yang sering kali menyebabkan masalah, seperti antrian yang tidak teratur, kesalahan pencatatan, dan kesulitan dalam melacak riwayat pesanan, terutama pada volume pelanggan yang tinggi. Penelitian ini bertujuan untuk merancang dan mengembangkan sistem pengelolaan pesanan yang berbasis struktur data stack dan queue. Sistem ini menggunakan metode FIFO pada queue untuk memastikan pesanan diproses secara berurutan sesuai urutan kedatangan, sementara stack berfungsi untuk menyimpan dan menampilkan riwayat pesanan yang sudah selesai diproses. Pengembangan sistem dilakukan dengan pendekatan waterfall yang meliputi analisis kebutuhan, desain sistem menggunakan diagram UML, implementasi dengan Python, dan pengujian sistem untuk memastikan semua fungsionalitas bekerja dengan baik. Hasil penelitian menunjukkan bahwa sistem ini dapat mengelola alur pemesanan dengan lebih terorganisir, mempercepat proses layanan, mengurangi kesalahan, dan meningkatkan kepuasan pelanggan. Selain itu, sistem ini memungkinkan staf restoran untuk dengan mudah mengakses riwayat pesanan yang sudah diproses. Secara keseluruhan, penelitian ini menyimpulkan bahwa penerapan struktur data stack dan queue menawarkan solusi yang efektif dan efisien dalam pengelolaan pesanan restoran. Penelitian lanjutan dapat fokus pada integrasi sistem dengan database eksternal untuk meningkatkan skalabilitas dan mendukung operasi restoran dengan lebih banyak pelanggan.

I. PENDAHULUAN

Manajemen pemesanan menu di restoran memiliki peranan penting dalam memberikan pengalaman yang optimal bagi pelanggan. Jika tidak dikelola dengan baik, berbagai masalah dapat muncul, seperti kesalahan dalam pencatatan pesanan, antrian yang tidak tertata, hingga kesulitan melacak riwayat pemesanan. Masalah ini dapat memengaruhi kelancaran operasional restoran dan menurunkan kepuasan pelanggan, yang merupakan elemen vital dalam industri kuliner. Oleh karena itu, kebutuhan akan sistem pemesanan yang efisien, akurat, dan terintegrasi semakin mendesak, terutama di era digital ini.

Kemajuan teknologi informasi telah menjadi faktor kunci dalam mempercepat dan menyederhanakan berbagai aktivitas, termasuk di sektor kuliner. Restoran yang mengadopsi teknologi ini dapat meningkatkan produktivitas, meminimalkan kesalahan operasional, dan menyediakan layanan yang lebih baik bagi pelanggan. Sistem berbasis struktur data seperti stack dan queue dapat menjadi solusi efektif untuk mengelola alur pemesanan secara terstruktur dan efisien. Dengan memanfaatkan teknologi ini, restoran dapat memperbaiki pengelolaan data pemesanan, memberikan kenyamanan lebih kepada pelanggan, serta memperkuat kemampuan analitik terhadap data pemesanan yang dihasilkan.

* Corresponding author

Meski demikian, banyak restoran saat ini masih mengandalkan sistem manual dalam pengelolaan pemesanan. Sebagai contoh, pelanggan sering kali diberikan selembar kertas untuk mencatat pesannya secara manual [1]. Pendekatan ini memiliki risiko tinggi terhadap kesalahan operasional, seperti pesanan yang tertukar, tumpang tindih, atau tidak sesuai, terutama saat menghadapi lonjakan pelanggan. Selain itu, pengelola restoran sering kali kesulitan dalam menyusun data pemesanan secara terorganisir, yang menghambat efisiensi operasional. Sistem manual semacam ini mengakibatkan waktu tunggu yang lama dan berkurangnya kenyamanan pelanggan [2].

Restoran, sebagai tempat makan, dan kafe, sebagai tempat bersantai, menghadapi tantangan serupa dalam memastikan pelayanan yang cepat dan akurat kepada pelanggan. Ketidakmampuan untuk memenuhi pesanan dengan cepat sering kali membuat pelanggan merasa tidak puas, yang dapat memengaruhi reputasi bisnis tersebut. Oleh sebab itu, implementasi teknologi pengelolaan pemesanan berbasis struktur data menjadi salah satu solusi penting untuk meningkatkan kualitas layanan dan daya saing.

Struktur data stack dan queue menawarkan pendekatan yang berbeda untuk mengelola alur pemesanan. Stack, yang berprinsip kerja LIFO (Last In First Out), memungkinkan data terakhir yang masuk menjadi data pertama yang diproses. Di sisi lain, queue, dengan prinsip FIFO (First In First Out), memproses data sesuai dengan urutan masuknya. Selain itu, ada deque, struktur data yang menggabungkan operasi stack dan queue, memberikan fleksibilitas lebih dalam manipulasi data [3] [4]. Karakteristik unik dari struktur data ini memberikan peluang besar dalam menciptakan sistem yang terorganisir untuk pengelolaan pemesanan restoran.

Dalam penelitian ini, dikembangkan sistem pengelolaan pemesanan berbasis stack dan queue untuk memanfaatkan keunggulan masing-masing struktur data tersebut. Queue digunakan untuk mengatur antrian pemesanan secara efisien, memastikan setiap pesanan diproses berdasarkan urutan masuknya. Sementara itu, stack dimanfaatkan untuk mencatat riwayat pemesanan yang telah selesai diproses, sehingga data tersebut dapat diakses kapan saja untuk analisis atau evaluasi layanan. Pendekatan ini diharapkan dapat mengatasi berbagai tantangan operasional yang dihadapi restoran, sekaligus meningkatkan efisiensi dan kepuasan pelanggan.

Penelitian ini bertujuan untuk mengimplementasikan struktur data queue dan stack dalam pengelolaan pemesanan restoran, sehingga memungkinkan sistem yang lebih terorganisir dalam mengelola antrian, mengurangi kesalahan dalam proses pemesanan, dan menyediakan data riwayat pesanan yang terstruktur. Dengan sistem ini, restoran dapat meningkatkan produktivitas operasional sekaligus memberikan pengalaman pelanggan yang lebih baik. Solusi ini menjadi langkah inovatif dalam mengintegrasikan teknologi berbasis struktur data untuk mendukung perkembangan di industri kuliner.

II. TINJAUAN PUSTAKA

Dalam buku Konsep Sistem Informasi, Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang memiliki hubungan, berkumpul bersama untuk melakukan kegiatan atau melakukan sasaran tertentu [5]. Layanan pemesanan makan di tempat yang harus menggunakan pencatatan di kertas masih banyak dilakukan di restoran baik yang di tempat umum maupun hotel. Sistem pemesanan manual ini, pembeli harus menulis pesanan dalam kertas dan harus datang lebih awal hanya untuk memesan makanan. Pencatatan yang dilakukan pada menu pesanan secara manual dengan menggunakan kertas tentunya kurang efisien baik dari sisi waktu maupun layanan yang diberikan kepada pelanggan. Upaya mewujudkan kepuasan konsumen dalam pemesanan makanan dapat dikembangkan aplikasi pemesanan makanan berbasis teknologi informasi [6].

Stack atau tumpukan adalah suatu struktur data yang penting dalam pemrograman dengan metode pemrosesan yang bersifat LIFO (Last In First Out) dimana objek/benda yang terakhir masuk ke dalam stack akan menjadi benda pertama yang dikeluarkan dari stack. Adapun operasi-operasi/fungsi yang dapat dilakukan pada stack adalah sebagai berikut : Push (menambah item pada stack pada tumpukan paling atas), Pop (mengambil item pada stack pada tumpukan paling atas), Clear (mengosongkan stack), IsEmpty (mengecek apakah stack sudah kosong), dan IsFull (mengecek apakah stack sudah penuh).

Kebalikan dari stack, queue (antrian) adalah jenis struktur data yang dapat diproses dengan sifat FIFO (First In First Out), dimana elemen yang pertama kali masuk ke antrian akan keluar pertama kalinya. Ada dua jenis operasi yang bias dilakukan di antrian : enqueue (memasukkan elemen baru ke dalam elemen) dan dequeue (adalah mengeluarkan satu elemen dari suatu antrian) [3].

Flowchart adalah representasi secara simbolik dari suatu algoritma atau prosedur untuk menyelesaikan suatu masalah. Pada penelitian ini, flowchart digunakan untuk menggambarkan bagaimana jalannya sistem yang akan dibangun secara garis besar. Pembuatan diagram alur pada tahap perancangan, dapat memudahkan programmer ketika pada tahap implementasi, pengujian awal maupun penerapan logika sistem yang akan dibangun [7].

Metode waterfall merupakan salah satu paradigma untuk membuat sistem rekayasa perangkat lunak dimana klien dan pengembang dapat berinteraksi untuk mencapai persyaratan sistem. Unified Modelling Language (UML) merupakan sebuah standar bahasa yang banyak digunakan didunia industri guna mendefinisikan requirement, membuat analisa & desain, serta menggambarkan desain arsitektur yang terdapat dalam pemrograman berorientasi objek [8]. Use case diagram adalah sebuah diagram yang menunjukkan hubungan antara actors dan use cases. Activity diagram menurut adalah suatu diagram yang menggambarkan konsep aliran data/kontrol, aksi terstruktur serta dirancang dengan baik dalam suatu sistem [9]. Sequence Diagram adalah tool yang sangat populer dalam pengembangan sistem informasi secara object oriented untuk

menampilkan interaksi antar objek. Selain itu Sequence Diagram dapat digunakan sebagai perkakas dalam perancangan antarmuka pemakai. Ada dua hal yang dapat dilakukan dengan Sequence Diagram, pertama untuk menguraikan sebuah proses bisnis menjadi aktivitas-aktivitas yang lebih kecil untuk mengidentifikasi kebutuhan interaksi pemakai pada masing-masing aktivitas tersebut [10]. Class diagram adalah sebuah diagram yang menunjukkan hubungan antar class yang didalamnya terdapat atribut dan fungsi dari suatu objek [9].

Visual Studio Code adalah kode editor sumber yang dikembangkan oleh Microsoft untuk Windows, Linux dan mac OS. Visual Code memudahkan dalam penulisan code yang mendukung beberapa jenis bahasa pemrograman yang digunakan dan memberi variasi warna sesuai dengan fungsi dalam rangkaian code tersebut. Selain itu, fitur lainnya adalah kemampuan untuk menambah ekstensi dimana para pengembang dapat menambah ekstensi untuk menambah fitur yang tidak ada di Visual Studio Code [11].

Draw.io adalah aplikasi berbasis web yang digunakan untuk membuat diagram dan grafik. Ini adalah alat yang populer untuk membuat berbagai jenis diagram, termasuk diagram alir, diagram organisasi, diagram alur kerja, diagram jaringan, diagram UML (Unified Modeling Language), dan banyak lagi. Draw.io sering digunakan oleh pengembang perangkat lunak, perencana proyek, analis bisnis, dan orang-orang dari berbagai bidang lainnya yang memerlukan visualisasi data atau informasi [12].

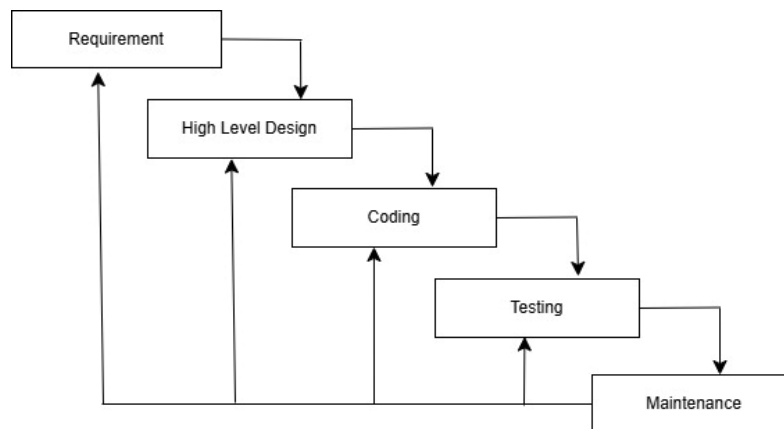
Berdasarkan berbagai penelitian sebelumnya, terlihat bahwa struktur data stack dan queue memiliki potensi besar untuk meningkatkan efisiensi dalam berbagai aplikasi. Penelitian oleh Nugroho et al. menunjukkan penggunaan queue dalam sistem panggilan wisudawan, di mana proses enqueue dan dequeue memastikan urutan panggilan berjalan terorganisir [13]. Begitupun Hadi et al. meneliti penerapan queue dalam sistem gudang, yang relevan dengan sistem restoran dalam memastikan pesanan diproses sesuai urutan kedatangan [14].

Metode FIFO (First In First Out) telah diterapkan secara luas dalam sistem antrian untuk memastikan bahwa layanan diberikan sesuai dengan urutan kedatangan. Dalam penelitian yang dilakukan oleh Moch. Farid Fauzi dan Alfie Nur Rahmi, penerapan metode FIFO pada sistem antrian administrasi mahasiswa berhasil mengurangi waktu tunggu dan meningkatkan kepuasan pengguna [15]. Implementasi metode yang sama dalam pengelolaan pesanan restoran juga dapat memastikan bahwa pesanan diproses berdasarkan urutan masuknya, sehingga mengurangi kemungkinan kesalahan dan mempercepat pelayanan kepada pelanggan.

Secara keseluruhan, berbagai penelitian yang telah dijelaskan menunjukkan bahwa penerapan system berbasis Queue dan Stack dalam pengelolaan pesanan restoran dapat memberikan solusi efektif untuk meningkatkan efisiensi dan kecepatan pemrosesan pesanan. Selain itu, penggunaan metode FIFO dalam antrian juga terbukti dapat mengurangi waktu tunggu dan meningkatkan kepuasan pelanggan, yang sangat penting dalam meningkatkan kualitas layanan di sektor restoran.

III. METODE

Penulis menggunakan metode Waterfall sebagai pendekatan penelitian sistem [16]:

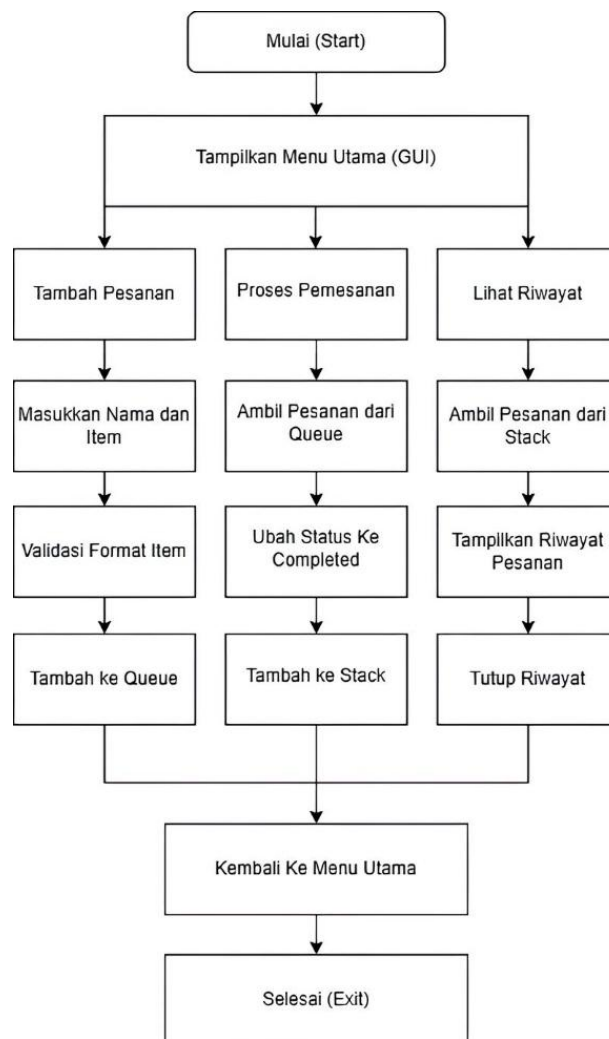


Gambar 1 Tahapan metode waterfall

Tahapan-tahapan kegiatan penelitian sistem oleh peneliti berdasarkan metode waterfall yaitu sebagai berikut requirement(kebutuhan sistem), high level design(desain sistem), coding(pengkodean sistem), testing(pengujian sistem), maintenance(pemeliharaan sistem).

1. Tahap Requirement (Kebutuhan Sistem): Pada tahap ini, pengembang sistem perlu berkomunikasi secara mendalam untuk memahami kebutuhan perangkat lunak yang diinginkan pengguna serta batasan-batasannya. Informasi tersebut dapat diperoleh melalui wawancara, diskusi, atau survei langsung. Setelah informasi terkumpul, data dianalisis untuk mengidentifikasi kebutuhan spesifik yang harus dipenuhi.
2. Tahap High-Level Design (Desain Rancangan Sistem): Tahap ini melibatkan pembuatan rancangan sistem yang bertujuan untuk menentukan kebutuhan perangkat keras (*hardware*) dan perangkat lunak (*software*). Selain itu, desain ini juga digunakan untuk mendefinisikan struktur dan arsitektur sistem secara keseluruhan, sehingga dapat menjadi pedoman dalam proses pengembangan.

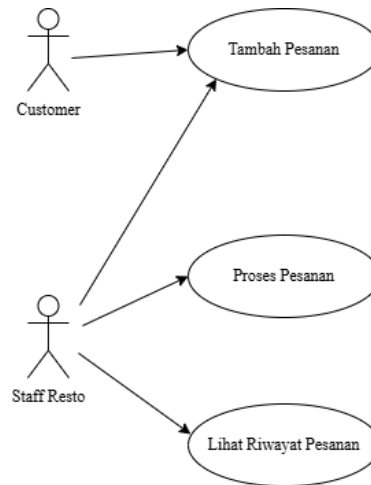
3. Coding (Implementasi Pengkodean Sistem): Tahap ini adalah proses pengkodean atau pembuatan program berdasarkan desain yang telah dibuat. Sistem dikembangkan dalam unit-unit kecil yang berdiri sendiri. Setiap unit ini diuji terlebih dahulu melalui *unit testing* untuk memastikan fungsionalitasnya sebelum digabungkan dalam tahap selanjutnya.
4. Testing (Pengujian Sistem): Tahap ini berfokus pada pengujian sistem untuk memastikan bahwa sistem yang dikembangkan sudah memenuhi kebutuhan yang ditentukan. Pengujian dilakukan dalam beberapa kategori, yaitu *unit testing* (untuk memvalidasi modul tertentu), pengujian sistem (untuk mengevaluasi interaksi antar modul setelah integrasi), dan *acceptance testing* (melibatkan pengguna atau pelanggan untuk memastikan semua kebutuhan telah terpenuhi).
5. Maintenance: Tahap ini merupakan fase terakhir dalam model *waterfall*. Perangkat lunak yang telah selesai dikembangkan mulai dijalankan, dan proses pemeliharaan dilakukan untuk memastikan kinerjanya tetap optimal. Pemeliharaan ini juga mencakup perbaikan kesalahan atau bug yang mungkin tidak terdeteksi pada tahap sebelumnya.



Gambar 2 Flowchart Sistem Pengelolaan Pemesanan Menu

A. Rancangan Pemodelan Data

1. Use Case Diagram



Gambar 3 Use Case Diagram Admin Restoran

Pada gambar 3 merupakan pemodelan data *use case diagram* dimana ada dua aktor utama yang terlibat dalam sistem pengelolaan pesanan restoran yaitu *customer* dan *staff resto*. *customer* memberikan nama dan item pesanan untuk dipesan. Proses ini diwakili oleh *use case tambah pesanan*. Setelah itu, *staff resto* akan memasukkan pesanan pelanggan ke dalam sistem dan memprosesnya. Dua langkah utama yang dilakukan *staff resto* adalah *proses pesanan* (mengambil dan memproses pesanan) serta *lihat riwayat pesanan* untuk memeriksa pesanan yang sudah selesai. Dengan diagram ini, terlihat bagaimana sistem mengelola pesanan dengan jelas dan siapa saja yang terlibat dalam setiap tahapannya.

2. Activity Diagram

Pemodelan data *activity diagram* melibatkan *customer* dan *staff resto*. Keduanya memulai dengan memilih *tambah pesanan* di *menu utama*. *Customer* mengisi nama dan item yang ingin dipesan, tetapi *staff resto* juga bisa menambahkan pesanan jika pelanggan belum melakukannya. Lalu sistem akan memeriksa format pesanan. Jika ada kesalahan, sistem akan meminta input ulang. Setelah pesanan valid, item akan dimasukkan ke dalam *queue (antrian)*. Jika memilih *proses pesanan*, *staff resto* akan mengambil pesanan dari antrian, memprosesnya, dan memindahkannya ke *stack (riwayat)*. Terakhir, jika memilih *lihat riwayat pesanan*, *staff resto* bisa melihat daftar pesanan yang sudah selesai. Setelah selesai, *staff resto* bisa kembali ke *menu utama* atau keluar dari sistem. Proses ini memastikan pesanan diproses dengan baik, dari awal hingga akhir.

3. Sequence Diagram

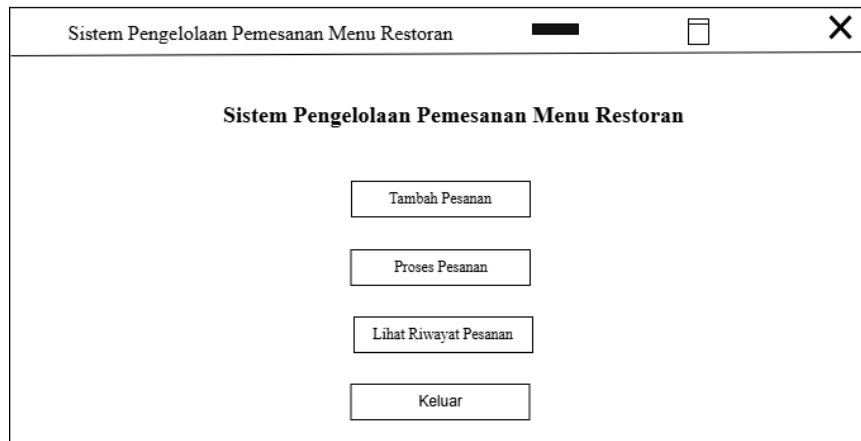
Pemodelan data *sequence diagram* menunjukkan bagaimana interaksi antara pengguna dan sistem berlangsung dalam pengelolaan pesanan restoran. Proses dimulai saat *customer* memberikan pesanan kepada *staff resto*, yang kemudian memasukkan pesanan tersebut ke dalam sistem dan memeriksa formatnya. Jika formatnya benar, pesanan ditambahkan ke *queue*. Kemudian, *staff resto* mengambil pesanan dari antrian untuk diproses dan memindahkannya ke *stack* setelah selesai. Jika ada yang ingin melihat riwayat pesanan, *staff resto* akan menampilkannya dari *stack*. Setelah itu, mereka dapat kembali ke *menu utama* atau keluar dari sistem. Alur ini memastikan setiap langkah pesanan dikelola dengan jelas dan efisien.

4. Class Diagram

Pemodelan data *class diagram* menjelaskan hubungan antara elemen-elemen utama dalam sistem. Terdapat lima kelas utama yang saling berinteraksi. *Customer* bertanggung jawab menyimpan informasi pelanggan, seperti nama dan daftar pesanan. *Order* merepresentasikan detail pesanan, mencakup nama item dan statusnya, apakah masih dalam antrian atau telah selesai. *Queue* digunakan untuk mengelola pesanan yang masih menunggu, dengan menyediakan fungsi untuk menambah (*enqueue*) dan mengambil (*dequeue*) pesanan. *Stack* menyimpan riwayat pesanan yang telah selesai melalui metode *push* dan *pop*. Sementara itu, *Staff* memiliki peran memproses pesanan dari antrian ke riwayat, sekaligus melihat pesanan yang telah selesai. Hubungan antar kelas ini mencerminkan alur kerja sistem, mulai dari pelanggan memesan hingga pesanan diselesaikan dan dicatat dalam riwayat, tanpa memanfaatkan database karena menggunakan memori program untuk pengelolaan datanya.

B. Rancangan Pemodelan Desain Interface

Dalam design interface ini terdapat button tambah pesanan, proses pesanan, lihat riwayat pesanan dan button keluar.



Gambar 4 Desain Interface Sistem Pengelolaan Pemesanan Menu

Gambar 4 menunjukkan antarmuka sederhana dari sistem pemesanan menu restoran. Di bagian atas terdapat judul "*Sistem Pengelolaan Pemesanan Menu Restoran*" yang menunjukkan fungsi utama aplikasi. *Ada empat tombol utama: Tambah Pesanan* untuk menambah pesanan baru, *Proses Pesanan* untuk memproses pesanan yang sudah masuk, *Lihat Riwayat Pesanan* untuk melihat daftar pesanan sebelumnya, dan *Keluar* untuk menutup aplikasi. Desainnya dibuat minimalis agar mudah digunakan oleh staf restoran dalam mengelola pesanan secara cepat dan efisien.

IV. HASIL

A. Requirement

Proses pemesanan di restoran sangat penting untuk menjaga efisiensi dan kepuasan pelanggan. Saat ini, banyak restoran masih menggunakan cara manual yang bisa menyebabkan masalah seperti pesanan tertukar atau sulitnya melacak riwayat pesanan, terutama ketika jumlah pelanggan banyak. Sistem manual ini juga berisiko menurunkan kualitas pelayanan dan meningkatkan ketidakpuasan pelanggan.

Untuk itu, solusi berbasis teknologi seperti sistem dengan stack dan queue menjadi pilihan tepat. Teknologi ini dapat mengatur pemesanan dengan lebih terstruktur, mengurangi kesalahan, dan meningkatkan kecepatan serta akurasi layanan. Dengan menggunakan sistem otomatis, restoran dapat lebih mudah melacak pesanan, mengurangi kesalahan pengiriman, dan meningkatkan pengalaman pelanggan.

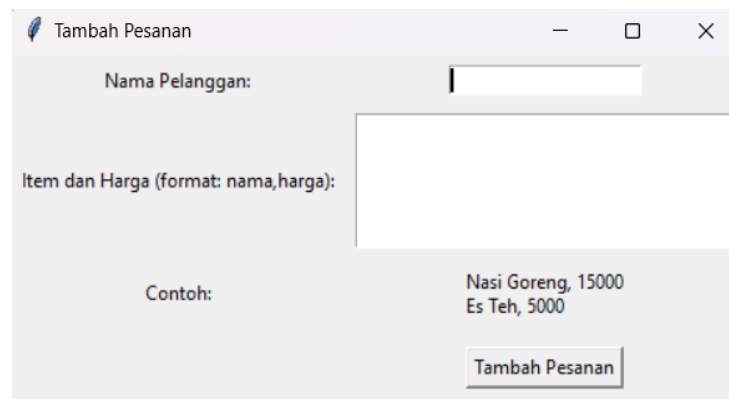
Sistem ini dirancang untuk menggantikan proses manual yang kurang efisien, memberikan kemudahan bagi pelayan dan kasir, serta meningkatkan produktivitas restoran. Dengan implementasi sistem ini, diharapkan restoran dapat meningkatkan kepuasan pelanggan dan membuat alur pemesanan lebih cepat dan terorganisir.

B. High Level Design (Desain Rancangan Sistem)



Gambar 5 Tampilan Utama

Gambar 5 merupakan tampilan utama Sistem Pengelolaan Pesanan Restoran. Dalam tampilan terdapat *button tambah pesanan, proses pesanan, lihat riwayat pesanan, dan keluar.*



Gambar 6 Tampilan *Tambah Pesanan*

Gambar 6 merupakan tampilan tambah pesanan. Dalam tampilan terdapat text nama pelanggan, item dan harga, dan contoh format pengetikan yang digunakan untuk memasukkan nama customer dan menu yang ingin dipesan beserta harganya. Lalu terdapat juga button tambah pesanan guna menginput data yang sudah diketik secara lengkap agar masuk ke tahap memvalidasi apakah sudah sesuai contoh format atau tidak. Secara teori pesanan pertama akan diproses pertama pula.



Gambar 7 Tampilan *Riwayat Pesanan*

Gambar 7 merupakan tampilan riwayat pesanan. Terlihat ada 4 riwayat pesanan yang dilakukan oleh Atta, Adam, Agata dan juga Zai. Dapat dilihat bahwa pesanan Atta berada paling atas karena pesanan Atta adalah pesanan yang terakhir diproses.

C. Coding (Implementasi Pengkodean Sistem)

Pada tahap implementasi, skrip kode yang digunakan untuk membangun sistem pengelolaan pesanan restoran memanfaatkan beberapa modul utama, yaitu *TKInter* untuk antarmuka grafis pengguna (GUI), *deque* sebagai struktur data *queue*, dan *datetime* untuk mencatat waktu pesanan (*timestamp*). Di dalam skrip kode, fungsi-fungsi seperti *queue* dan *stack* telah dideklarasikan untuk mendukung proses pengelolaan data pesanan. Fungsi *enqueue* dibuat untuk menambahkan pesanan baru ke dalam antrian, sedangkan *dequeue* berfungsi memproses pesanan yang telah masuk dan menandainya sebagai selesai.

Dalam fungsi *enqueue*, beberapa variabel penting didefinisikan, seperti customer (nama pelanggan), item (nama pesanan), timestamp (waktu pemesanan), dan status pesanan. Ketika status pesanan berubah menjadi "Completed", pesanan tersebut akan dipindahkan ke dalam *stack* melalui proses *push_to_history*. *Stack* ini digunakan untuk menyimpan dan menampilkan riwayat pesanan, di mana pesanan terakhir yang diproses akan berada di bagian atas, sesuai dengan konsep LIFO (Last In, First Out).

Selain itu, terdapat skrip kode yang mengatur tombol untuk menambah pesanan dan elemen-elemen di dalamnya. Ketika tombol ditekan, sebuah kotak pesan (*display_message*) akan ditampilkan jika terjadi peristiwa tertentu, seperti ketiadaan riwayat pesanan. Skrip ini juga mencakup pengaturan format isian untuk pesanan baru dan memberikan contoh format yang benar agar pengguna tidak melakukan kesalahan. Format pengisian pesanan yang valid adalah seperti berikut: "Nasi Goreng, 5000". Jika format isian tidak sesuai, sistem akan melakukan validasi dan memberikan notifikasi kepada pengguna.

Skrip kode untuk antarmuka pengguna (GUI) menggunakan *TKInter* untuk mengatur elemen seperti nama tombol dan tampilan layar utama. Antarmuka ini dirancang agar mudah digunakan dan mendukung pengelolaan pesanan secara efisien.

D. Testing (Pengujian Sistem)

Pada tahap ini, pengujian dilakukan untuk memastikan bahwa setiap fungsi dalam sistem berjalan sesuai dengan kebutuhan dan spesifikasi yang telah dirancang. Pengujian dilakukan dalam beberapa kategori, yaitu:

1. Unit Testing
Setiap modul atau unit kode diuji secara terpisah untuk memastikan bahwa fungsi-fungsi dasar, seperti *enqueue*, *dequeue*, *push*, dan *pop*, bekerja dengan benar. Seperti, menambahkan pesanan baru ke dalam *queue* dengan data lengkap (nama pelanggan, item pesanan, harga), mengambil pesanan dari *queue* dan memindahkannya ke *stack* setelah selesai diproses, serta memastikan *stack* menyimpan riwayat pesanan secara urut sesuai prinsip LIFO.
2. Integration Testing
Setelah semua unit diuji, modul-modul yang saling terhubung diuji untuk memastikan bahwa alur data antar modul berjalan tanpa kendala. Seperti, mengintegrasikan modul *queue* dengan GUI, sehingga pesanan yang ditambahkan dari antarmuka pengguna langsung masuk ke dalam *queue* dan memastikan bahwa status pesanan berubah menjadi "Completed" secara otomatis setelah diproses dan dipindahkan ke *stack*.
3. System Testing
Pengujian menyeluruh dilakukan untuk memastikan sistem dapat menangani berbagai skenario penggunaan. Seperti, menambahkan beberapa pesanan secara berurutan, kemudian memprosesnya untuk memastikan urutan *queue* sesuai dengan prinsip FIFO, memeriksa apakah riwayat pesanan dapat ditampilkan dengan benar dalam antarmuka GUI dan menguji skenario *edge case*, seperti input data yang tidak valid (misalnya format pesanan yang salah), untuk memastikan sistem memberikan peringatan dan tidak memproses data tersebut.
4. Acceptance Testing
Pengujian ini melibatkan simulasi langsung dengan pengguna (staf restoran) untuk memastikan bahwa sistem memenuhi kebutuhan operasional. Feedback dari pengguna digunakan untuk melakukan perbaikan tambahan. Seperti, memastikan bahwa pengguna dapat dengan mudah menambah, memproses, dan melihat riwayat pesanan melalui antarmuka, serta mengukur waktu pemrosesan pesanan untuk memastikan sistem membantu mempercepat layanan.

Hasil Pengujian:

Pengujian menunjukkan bahwa semua fungsi berjalan sesuai harapan. Sistem berhasil dalam mengelola pesanan dengan urutan yang benar (FIFO untuk *queue*), menyimpan dan menampilkan riwayat pesanan secara akurat (LIFO untuk *stack*), serta memberikan notifikasi jika terjadi kesalahan input atau ketika tidak ada pesanan yang dapat diproses.

E. Maintenance (Pemeliharaan Sistem)

Pemeliharaan sistem adalah langkah penting setelah sistem diimplementasikan untuk memastikan kinerjanya tetap optimal. Proses ini melibatkan pemantauan rutin, identifikasi dan perbaikan masalah teknis, serta pembaruan sistem jika diperlukan. Tim pengembang bertanggung jawab untuk menjaga agar sistem tetap berjalan lancar, melakukan perbaikan bug, dan menyesuaikan dengan perubahan kebutuhan atau teknologi. Dengan pemeliharaan yang baik, sistem akan tetap efisien dan dapat diandalkan dalam pengelolaan pesanan restoran.

V. PEMBAHASAN

Sistem pengelolaan pesanan berbasis *stack* dan *queue* yang telah dikembangkan menunjukkan hasil yang signifikan dalam meningkatkan efisiensi dan akurasi layanan di restoran. Dengan memanfaatkan prinsip FIFO pada *queue* untuk antrian pesanan dan LIFO pada *stack* untuk menyimpan riwayat, sistem ini berhasil mengurangi potensi kesalahan dalam pengelolaan pesanan serta mempersingkat waktu pelayanan. Hasil pengujian membuktikan bahwa sistem ini mampu mengelola alur pemesanan dengan terstruktur, memberikan pengalaman pengguna yang lebih baik, dan memudahkan pelacakan riwayat pesanan tanpa menggunakan database eksternal.

Meskipun demikian, penelitian ini memiliki beberapa kekurangan yang perlu dicermati. Pertama, sistem ini belum mendukung integrasi dengan database, sehingga penyimpanan data sepenuhnya bergantung pada memori program. Hal ini membatasi kapasitas penyimpanan dan kurang ideal untuk restoran dengan volume pesanan yang tinggi. Kedua, antarmuka pengguna (GUI) yang dirancang menggunakan *TKInter* masih sederhana dan mungkin memerlukan pengembangan lebih lanjut agar lebih intuitif dan menarik secara visual. Selain itu, pengujian sistem hanya dilakukan pada lingkungan simulasi dengan volume data yang terbatas, sehingga belum sepenuhnya merepresentasikan kondisi nyata di restoran dengan beban kerja yang kompleks.

Penelitian ini juga memiliki keterbatasan dalam hal fleksibilitas fungsi. Sistem hanya dirancang untuk menangani alur pesanan dasar tanpa mempertimbangkan fitur tambahan, seperti modifikasi pesanan atau pembatalan pesanan oleh pelanggan. Untuk penggunaan di dunia nyata, fitur-fitur ini sangat dibutuhkan agar sistem dapat memenuhi kebutuhan operasional restoran secara menyeluruh.

Ke depan, pengembangan sistem dapat difokuskan pada integrasi dengan database eksternal untuk meningkatkan skalabilitas dan efisiensi dalam menangani data pesanan dalam jumlah besar. Selain itu, pengujian

sistem dalam lingkungan nyata dengan melibatkan restoran sebagai pengguna langsung akan memberikan gambaran yang lebih komprehensif tentang keefektifan sistem ini. Dengan demikian, sistem ini tidak hanya akan menjadi solusi teknis yang fungsional, tetapi juga mampu memberikan dampak positif yang signifikan dalam operasional restoran modern.

VI. KESIMPULAN

Penelitian ini menghasilkan sistem pengelolaan pesanan restoran yang memanfaatkan struktur data stack dan queue untuk meningkatkan efisiensi operasional. Dengan prinsip FIFO pada queue, sistem mampu memproses pesanan sesuai urutan masuk, sementara stack digunakan untuk menyimpan dan mempermudah akses riwayat pesanan menggunakan metode LIFO. Berdasarkan hasil pengujian, sistem ini terbukti dapat mempercepat layanan, mengurangi kesalahan dalam pencatatan, serta memberikan pengalaman pengguna yang lebih baik. Desain sederhana tanpa memanfaatkan database eksternal memungkinkan implementasi yang efisien, khususnya untuk restoran dengan volume pesanan yang tidak terlalu besar.

Namun, penelitian ini belum sepenuhnya bebas dari keterbatasan. Penyimpanan data hanya berbasis memori program, sehingga sulit diterapkan pada restoran dengan volume pesanan tinggi. Selain itu, antarmuka pengguna berbasis GUI masih sederhana dan membutuhkan pengembangan agar lebih ramah pengguna. Ke depan, integrasi dengan database eksternal dan uji coba langsung di restoran akan menjadi langkah penting untuk mengatasi kendala ini.

Penerapan struktur data stack dan queue dalam penelitian ini menunjukkan efektivitasnya sebagai solusi teknologi untuk mengelola pesanan restoran. Dengan peningkatan fitur dan uji coba lebih lanjut, sistem ini memiliki peluang besar untuk mendukung digitalisasi dalam industri kuliner modern.

DAFTAR PUSTAKA

- [1] D. M. R. L. Finsa Nurpandi, "Perancangan dan Pembangunan Aplikasi Electronic Menu Restoran," *Media Jurnal Informatika*, vol. XIII, nr 1, pp. 1-11, 2021.
- [2] M. R. Malik Fajar Siddiq, "Sistem Informasi Pemesanan Menu Makanan dan Minuman pada Kafe Lori Berbasis Web," *Kajian Ilmiah Informatika dan Komputer*, vol. IV, nr 3, pp. 1255-1263, 2023.
- [3] J. Sihombing, "PENERAPAN STACK DAN QUEUE PADA ARRAY DAN LINKED LIST DALAM JAVA," *INFOKOM*, vol. VII, nr 2, pp. 15-24, 2019.
- [4] R. Selamat, "IMPLEMENTASI STRUKTUR DATA LIST, QUEUE DAN STACK DALAM JAVA," *Media Informatika*, vol. XV, nr 3, pp. 18-25, 2016.
- [5] J. Hutahaean, *Konsep Sistem Informasi*, Yogyakarta: Deepublish, 2019.
- [6] W. L. A. Y. R. B. L. Q. Muhammad Sholeh, "Pengelolaan Pemesanan Menu Makanan Menggunakan Framework Flask Python," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. IX, nr 2, pp. 916-929, 2022.
- [7] H. Y. S. K. Rizaldy Gunawan, "Implementasi Metode Queue Pada Sistem Antrian Online Berbasis Web Studi Kasus Uptd Puskesmas Sananwetan," *Jurnal Mahasiswa Teknik Informatika*, vol. VII, nr 3, pp. 1538-1545, 2023.
- [8] R. A.S och M. Shalahuddin, *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*, Bandung: Informatika, 2019.
- [9] W. R. H. T. W. Mochammad Hasymi Somaída, "Perancangan Sistem Informasi Perpustakaan Menggunakan Diagram UML (Unified Modelling Language)," *Jurnal Ilmiah Komputer Terapan dan Informasi*, vol. I, nr 1, pp. 19-25, 2022.
- [10] N. Nurdam, "Sequence Diagram Sebagai Perkakas Perancangan Antarmuka Pemakai," *Ultimatics*, vol. VI, nr 1, pp. 21-25, 2014.
- [11] D. A. N. Nur Ariesanto Ramdhan, "Rancang Bangun dan Implementasi Sistem Informasi Skripsi Online Berbasis Web," *Information Technology Journal of UMUS*, vol. I, nr 2, pp. 1-12, 2019.
- [12] M. Y. A. Mahendra, "THE USE OF DRAW IO AS DIGITAL MIND MAP TO IMPROVE STUDENTS' CREATIVITY AND STUDENTS' CONCEPT MASTERY IN LEARNING HUMAN INFLUENCE ON ECOSYSTEM," 2021.
- [13] Y. S. T. R. H. L. Zulia Putri Tanjung, "PENERAPAN METODE FIFO DAN METODE LIFO DALAM MENJAGA EFEKTIVITAS PERSEDIAAN PUPUK (STUDI KASUS PT. CAHAYA PELITA ANDHIKA) KABUPATEN TAPANULI TENGAH," *Jurnal Ekonomi Bisnis dan Manajemen*, vol. I, nr 1, pp. 1-8, 2023.

- [14] T. D. S. M Zaky Hadi1, "PEMODELAN ANTRIAN SISTEM PENGAMBILAN PESANAN PRODUK PADA GUDANG MINUMAN RINGAN DENGAN SISTEM RAK DRIVE-IN," *Jurnal Teknologi Industri Pertanian*, vol. XXVII, nr 3, pp. 298-309, 2017.
- [15] A. N. R. Moch. Farid Fauzi, "PENERAPAN METODE FIRST IN FIRST OUT (FIFO) DALAM SISTEM ANTRIAN PELAYANAN ADMINISTRASI MAHASISWA (STUDI KASUS: DAAK UNIVERSITAS AMIKOM YOGYAKARTA)," *Jurnal Manajemen Informatika & Komputerisasi Akuntansi*, vol. V, nr 2, pp. 183-188, 2021.
- [16] A. A. Wahid, "Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi," *Jurnal Ilmu-ilmu Informatika dan Manajemen STMIK*, pp. 1-5, 2020.