

LAPORAN *MINI PROJECT*

Sistem Antrian Pemesanan Berbasis C++ Dengan Implementasi Fungsi CURD, Sort & Search

Diajukan untuk memenuhi tugas mata kuliah
Algoritma & Pemrograman

Dosen Pengampu: Mochammad Fadillah Abdullah S.Kom M.Kom

Disusun Oleh:

Dimas Pratama (NIM. 2512006)

Ahmad Afandi S. P (NIM. 2512011)



**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PGRI JOMBANG 2026**

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah Swt. atas rahmat dan pertolongan-Nya sehingga laporan yang berjudul “Pengelolaan Wakaf di Masjid Al-Mutaqien” dapat diselesaikan dengan baik. Shalawat serta salam semoga senantiasa tercurah kepada Nabi Muhammad Saw., beserta keluarga, sahabat, dan seluruh umatnya.

Laporan ini disusun sebagai bagian dari tugas perkuliahan sekaligus sebagai upaya penulis untuk memahami secara langsung bagaimana pengelolaan wakaf dijalankan di lingkungan masjid. Wakaf tidak hanya bernilai ibadah, tetapi juga memiliki peran penting dalam mendukung kegiatan keagamaan dan sosial masyarakat. Oleh karena itu, pengelolaan wakaf yang dilakukan secara amanah dan bertanggung jawab menjadi hal yang sangat penting agar manfaatnya dapat dirasakan secara berkelanjutan.

Dalam laporan ini, penulis membahas pengelolaan wakaf di Masjid Al-Mutaqien, mulai dari proses pengelolaan hingga pemanfaatannya dalam menunjang kegiatan masjid. Diharapkan laporan ini dapat menambah wawasan, baik bagi penulis sendiri maupun bagi pembaca, khususnya mahasiswa yang tertarik pada kajian wakaf dan pengelolaan masjid.

Penulis menyadari bahwa laporan ini masih memiliki keterbatasan. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan sebagai bahan perbaikan ke depannya. Semoga laporan ini dapat memberikan manfaat dan menjadi referensi yang berguna dalam dunia akademik.

Jombang
Senin 5 Januari 2026

Penulis

DAFTAR ISI

COVER	i
KATA PENGANTAR	ii
DAFTAR ISI	iii
BAB I	5
PENDAHULUAN	5
1.1 Latar Belakang	5
1.2 Rumusan Masalah	2
1.3 Tujuan	2
BAB II	3
PEMBAHASAN	3
2.1 Desain	3
2.1.1 Pengertian dan Fungsi Flowchart	3
2.1.2 Flowchart Sistem Antrian Pemesanan.....	3
2.1.3 Penjelasan Flowchart.....	4
2.2 Code	6
2.2.1 Penggunaan Library	6
2.2.2 Pendefinisian Function Prototype	7
2.2.3 Struktur Data Pesanan	8
2.2.4 Struktur Data Menu dan Database Menu	8
2.2.5 Fungsi Menampilkan Daftar Menu	9
2.2.6 Fungsi Tambah Pesanan (Create).....	9
2.2.7 Proses Validasi, Antrian, Pembayaran, dan Pembaruan Status Pesanan.	10
2.2.8 Fungsi Menampilkan Semua Pesanan (Read)	12
2.2.9 Fungsi Cari Pesanan (Search).....	12
2.2.10 Fungsi Batalkan Pesanan (Delete).....	13

2.2.11 Fungsi Sortir Menu Berdasarkan Harga (Sort).....	13
2.2.12 Fungsi Menu Antrian.....	14
2.2.13 Fungsi Main.....	14
2.3 Test.....	15
BAB III	21
PENUTUP	21
3.1 Kesimpulan.....	21
3.2 Link Video Presentasi	22

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi pada era digital saat ini telah memberikan pengaruh besar terhadap cara manusia mengelola data dan menjalankan aktivitas operasional di berbagai bidang. Hampir seluruh sektor pelayanan, seperti usaha kuliner, rumah sakit, perbankan, dan layanan publik, dituntut untuk mampu memberikan pelayanan yang cepat, tepat, dan efisien. Salah satu permasalahan yang sering terjadi dalam pelayanan tersebut adalah pengelolaan antrian pemesanan yang masih dilakukan secara manual.

Sistem antrian manual memiliki berbagai kelemahan, antara lain rawan terjadi kesalahan pencatatan, sulit dalam pencarian data, serta tidak teraturnya urutan pelayanan. Hal ini dapat menyebabkan keterlambatan proses, menurunnya kualitas pelayanan, dan berkurangnya kepuasan pelanggan. Oleh karena itu, diperlukan sebuah sistem antrian terkomputerisasi yang mampu mengelola data pemesanan secara terstruktur dan sistematis.

Sistem antrian pemesanan berbasis komputer merupakan salah satu bentuk penerapan Sistem Informasi, di mana data pesanan pelanggan diolah menjadi informasi yang dapat digunakan untuk mendukung proses pelayanan. Dalam makalah ini, sistem antrian dibangun menggunakan bahasa pemrograman C++, dengan menerapkan konsep CRUD (Create, Read, Update, Delete), serta fitur sorting dan searching. Pemilihan bahasa C++ bertujuan untuk melatih pemahaman logika pemrograman, algoritma dasar, dan struktur data sederhana bagi mahasiswa semester awal.

Dengan adanya sistem ini, diharapkan mahasiswa dapat memahami bagaimana konsep sistem informasi diterapkan secara nyata melalui perancangan flowchart dan implementasi program, serta mampu melihat keterkaitan antara teori dan praktik.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, maka rumusan masalah dalam makalah ini adalah sebagai berikut:

1. Bagaimana perancangan awal sistem antrian pemesanan menggunakan flowchart?
2. Bagaimana penerapan konsep CRUD dalam sistem antrian pemesanan berbasis C++?
3. Bagaimana proses sorting dan searching diterapkan dalam pengelolaan data menu dan pesanan?
4. Bagaimana hasil pengujian sistem antrian pemesanan yang telah dibuat?

1.3 Tujuan

Tujuan dari penulisan makalah ini adalah:

1. Menjelaskan perancangan sistem antrian pemesanan melalui flowchart.
2. Mengimplementasikan sistem antrian pemesanan menggunakan bahasa pemrograman C++.
3. Menerapkan fungsi CRUD, sorting, dan searching pada sistem antrian.
4. Mengetahui hasil pengujian sistem dan memastikan sistem berjalan sesuai dengan perancangan.

BAB II

PEMBAHASAN

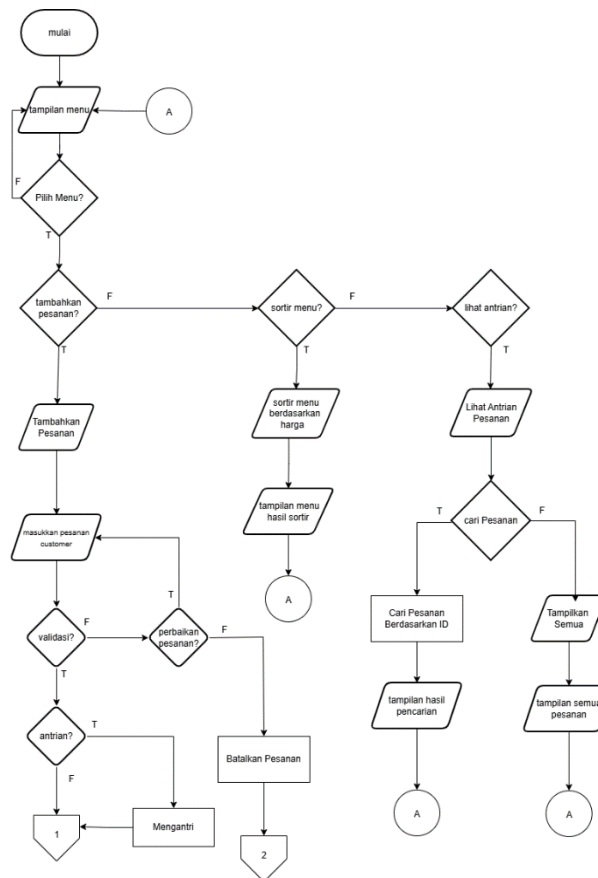
2.1 Desain

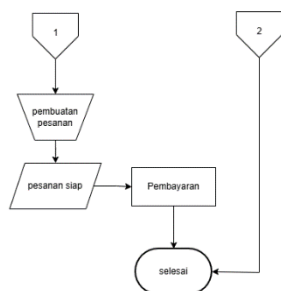
2.1.1 Pengertian dan Fungsi Flowchart

Flowchart adalah diagram alur yang digunakan untuk menggambarkan urutan langkah atau proses dalam suatu sistem secara logis dan terstruktur. Flowchart berfungsi sebagai alat bantu dalam merancang sistem agar alur kerja dapat dipahami dengan mudah sebelum diimplementasikan ke dalam kode program.

Dalam sistem antrian pemesanan ini, flowchart digunakan untuk menggambarkan alur mulai dari tampilan menu, proses pemesanan, validasi, pengelolaan antrian, hingga penyelesaian pesanan.

2.1.2 Flowchart Sistem Antrian Pemesanan





Gambar 2.1 Flowchart Sistem Antrian Pemesanan

2.1.3 Penjelasan Flowchart

Flowchart sistem antrian pemesanan pada penelitian ini dirancang untuk menggambarkan alur kerja sistem secara menyeluruh, mulai dari interaksi awal pengguna dengan sistem hingga proses akhir penyelesaian pesanan. Flowchart ini berfungsi sebagai pedoman logika utama dalam pengembangan program, sehingga setiap langkah yang terdapat di dalam sistem dapat dipahami secara jelas dan sistematis.

Proses pada flowchart diawali dengan simbol (mulai), yang menandakan bahwa sistem siap dijalankan. Setelah sistem aktif, langkah pertama yang dilakukan adalah menampilkan (daftar menu) kepada pengguna. Tampilan menu ini berisi informasi menu makanan dan minuman beserta harga masing-masing, sehingga pengguna dapat mengetahui pilihan yang tersedia sebelum melakukan pemesanan. Tahap ini penting karena menjadi dasar bagi pengguna dalam menentukan tindakan selanjutnya.

Setelah menu ditampilkan, sistem memberikan keputusan kepada pengguna melalui proses (pemilihan menu utama). Pada tahap ini, pengguna dapat memilih beberapa opsi, yaitu menambahkan pesanan, melihat antrian pesanan, atau melakukan sortir menu berdasarkan harga. Percabangan ini ditunjukkan pada flowchart menggunakan simbol keputusan (decision) yang memungkinkan sistem berjalan ke jalur yang berbeda sesuai dengan pilihan pengguna.

Apabila pengguna memilih untuk (menambahkan pesanan), sistem akan masuk ke proses pembuatan pesanan baru. Pada tahap ini, pengguna diminta untuk memasukkan data customer, seperti nama pelanggan, serta memilih menu

yang diinginkan berdasarkan ID menu yang tersedia. Sistem kemudian mencatat data pesanan tersebut sebagai data sementara sebelum dilakukan proses validasi. Proses ini bertujuan untuk memastikan bahwa data yang dimasukkan sudah sesuai dan benar.

Setelah data pesanan dimasukkan, sistem akan melakukan proses (validasi pesanan). Pada tahap validasi ini, pengguna diminta untuk mengonfirmasi apakah data pesanan sudah benar atau belum. Jika pengguna menyetujui pesanan, maka proses akan dilanjutkan ke tahap berikutnya. Namun, jika pengguna tidak menyetujui, sistem memberikan dua pilihan, yaitu memperbaiki pesanan atau membatalkan pesanan. Percabangan ini penting untuk memberikan fleksibilitas kepada pengguna dalam melakukan koreksi sebelum pesanan diproses lebih lanjut.

Jika pesanan dinyatakan valid, maka sistem akan memasukkan pesanan tersebut ke dalam (antrian pemesanan). Pesanan yang telah masuk ke antrian akan diberikan status “Sedang Diproses”, yang menunjukkan bahwa pesanan telah diterima oleh sistem dan menunggu untuk diproses lebih lanjut. Pada tahap ini, flowchart menggambarkan bahwa pesanan berada dalam kondisi mengantre sesuai dengan urutan masuknya pesanan.

Selanjutnya, sistem akan melanjutkan ke proses (pembuatan dan penyelesaian pesanan). Setelah pesanan dianggap siap, sistem mengarahkan pengguna ke tahap (pembayaran). Jika pembayaran berhasil dilakukan, maka status pesanan akan diperbarui menjadi “Selesai”. Tahap ini menandakan bahwa seluruh rangkaian proses pemesanan telah selesai dan pesanan tidak lagi berada dalam antrian aktif.

Selain proses penambahan pesanan, flowchart juga menggambarkan jalur alternatif apabila pengguna memilih menu (sortir menu berdasarkan harga). Pada jalur ini, pengguna diminta untuk memasukkan rentang harga minimum dan maksimum. Sistem kemudian akan mengurutkan menu berdasarkan harga dan menampilkan hasil sortir kepada pengguna. Setelah hasil ditampilkan, sistem akan kembali ke menu utama, sehingga pengguna dapat melanjutkan aktivitas lainnya.

Flowchart juga menunjukkan proses (melihat antrian pesanan), di mana pengguna dapat memilih untuk menampilkan seluruh data pesanan atau melakukan pencarian pesanan berdasarkan ID. Jika pengguna memilih pencarian, sistem akan mencocokkan ID yang dimasukkan dengan data yang tersimpan di dalam antrian, kemudian menampilkan hasil pencarian tersebut. Apabila ID tidak ditemukan, sistem akan memberikan informasi bahwa pesanan tidak tersedia.

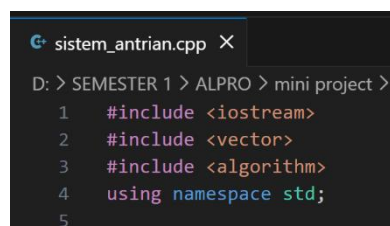
Setelah seluruh proses selesai dijalankan, baik itu penambahan pesanan, sortir menu, maupun pencarian data, sistem akan kembali ke (menu utama). Proses ini memungkinkan sistem berjalan secara berulang hingga pengguna memilih untuk keluar dari sistem. Flowchart diakhiri dengan simbol (selesai), yang menandakan bahwa program dihentikan.

Secara keseluruhan, flowchart sistem antrian pemesanan ini dirancang untuk memastikan bahwa setiap proses berjalan secara terstruktur, logis, dan mudah dipahami. Flowchart ini juga menjadi dasar utama dalam implementasi kode program C++, sehingga kesesuaian antara desain awal dan hasil implementasi dapat terjaga dengan baik.

2.2 Code

Pada bagian sub bab ini menjelaskan secara rinci implementasi sistem antrian pemesanan menggunakan bahasa pemrograman C++. Penjelasan dimulai dari penggunaan library, pendefinisian struktur data, pembuatan fungsi-fungsi utama, hingga alur program pada fungsi main() sebagai pusat pengendali sistem.

2.2.1 Penggunaan Library



```
sistem_antrian.cpp X
D: > SEMESTER 1 > ALPRO > mini project >
1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4  using namespace std;
5
```

Gambar 2.2 Code Library

Library `iostream` digunakan untuk mendukung proses input dan output, seperti menampilkan teks ke layar dan menerima input dari pengguna melalui `cin` dan `cout`. Library ini merupakan komponen utama dalam program berbasis console.

Library `vector` digunakan sebagai struktur data dinamis untuk menyimpan kumpulan data pesanan dan menu. Penggunaan `vector` memungkinkan sistem menambahkan dan menghapus data secara fleksibel tanpa harus menentukan ukuran array secara tetap.

Library `algorithm` digunakan untuk mendukung proses pengurutan data (sorting), khususnya dalam mengurutkan menu berdasarkan harga dan mengurutkan data pesanan berdasarkan ID. Library ini menyediakan fungsi bawaan seperti `sort()` yang mempermudah implementasi algoritma.

Perintah `using namespace std;` digunakan agar penulisan kode lebih sederhana tanpa harus menuliskan awalan `std::` pada setiap fungsi standar C++.

2.2.2 Pendefinisian Function Prototype

```
6 // ===== FUNCTION PROTOTYPE =====
7 void tampilkanMenu();
8 void tambahPesanan();
9 void tampilkanSemua();
10 void cariPesanan();
11 void batalkanPesanan();
12 void sortirMenuByHarga();
13 void menuAntrian();
```

Gambar 2.3 Code Function Prototype

Function prototype digunakan untuk mendeklarasikan fungsi-fungsi yang akan digunakan dalam program sebelum fungsi tersebut didefinisikan secara lengkap. Hal ini bertujuan agar fungsi dapat dipanggil pada bagian mana pun di dalam program, termasuk di dalam fungsi `main()`.

Setiap prototype memiliki peran masing-masing dalam sistem antrian, seperti menampilkan menu, menambah pesanan, mencari data, dan mengelola antrian.

2.2.3 Struktur Data Pesanan

```
15 struct Pesanan {
16     int id;
17     string nama;
18     vector<string> menu;
19     string status;
20     string jenis; // Makan di Tempat / Bawa Pulang
21 };
22
23 vector<Pesanan> antrian;
24 int idCounter = 1;
```

Gambar 2.4 Code Struktur Pesanan

Struktur Pesanan digunakan untuk merepresentasikan satu data pesanan pelanggan. Variabel id berfungsi sebagai identitas unik pesanan. Variabel nama menyimpan nama pelanggan. Variabel menu menyimpan daftar menu yang dipesan dalam bentuk vector agar dapat menyimpan lebih dari satu menu. Variabel status menunjukkan kondisi pesanan, sedangkan jenis menunjukkan jenis pesanan, yaitu makan di tempat atau bawa pulang.

2.2.4 Struktur Data Menu dan Database Menu

```
25
26 // ===== DATA BASE MENU =====
27 struct MenuItem {
28     int id;
29     string nama;
30     string kategori; // Makanan / Minuman
31     int harga;
32 };
33
34 vector<MenuItem> databaseMenu = {
35     {1, "Nasi Goreng", "Makanan", 20000},
36     {2, "Mie Ayam", "Makanan", 18000},
37     {3, "Ayam Geprek", "Makanan", 22000},
38     {4, "Sate Ayam", "Makanan", 25000},
39     {5, "Bakso", "Makanan", 20000},
40     {6, "Es Teh", "Minuman", 5000},
41     {7, "Es Jeruk", "Minuman", 7000},
42     {8, "Kopi Hitam", "Minuman", 8000},
43     {9, "Teh Hangat", "Minuman", 5000},
44     {10, "Jus Alpukat", "Minuman", 12000}
45 };
```

Gambar 2.5 Code Data Base Menu

Struktur MenuItem digunakan untuk menyimpan data menu makanan dan minuman. Setiap menu memiliki ID, nama menu, kategori, dan harga. Struktur ini membantu sistem dalam menampilkan dan mengelola data menu secara terorganisir.

Database menu disimpan dalam bentuk vector<MenuItem> yang berisi daftar menu beserta informasinya. Data ini bersifat tetap dan digunakan sebagai referensi saat pengguna melakukan pemesanan.

2.2.5 Fungsi Menampilkan Daftar Menu

```
46
47 // ===== output daftar menu =====
48 void tampilkanMenu() {
49     cout << "\n=== DAFTAR MENU ===\n";
50     cout << "ID | Nama Menu | Kategori | Harga\n";
51     cout << "-----\n";
52     for (auto m : databaseMenu) {
53         cout << m.id << " | "
54             << m.nama << " | "
55             << m.kategori << " | Rp"
56             << m.harga << endl;
57     }
58 }
59
```

Gambar 2.6 Daftar Menu

Fungsi ini berfungsi untuk menampilkan seluruh daftar menu yang tersedia kepada pengguna. Informasi yang ditampilkan meliputi ID menu, nama menu, kategori, dan harga. Fungsi ini dipanggil pada awal program dan saat pengguna ingin melihat menu sebelum melakukan pemesanan.

2.2.6 Fungsi Tambah Pesanan (Create)

```
51 void tambahPesanan() {
52     Pesanan p;
53     int jenis;
54
55     cout << "\n=== PEMBUATAN PESANAN ===\n";
56     p.id = idCounter++;
57
58     cout << "Masukkan Nama Customer : ";
59     cin >> p.nama;
60
61     // ===== INPUT & VALIDASI =====
62     bool selesai = false;
63
64     while (!selesai) {
65         p.menu.clear(); // Hapus menu lama jika perbaikan
66
67         int jumlahMenu;
68         cout << "\nMasukkan jumlah menu yang dipesan (1 - 2) : ";
69         cin >> jumlahMenu;
70
71         tampilkanMenu();
72
73         bool inputValid = true;
74
75         for (int i = 0; i < jumlahMenu; i++) {
76             int idMenu;
77             bool ditemukan = false;
78
79             cout << "Pilih ID Menu ke-" << (i + 1) << " : ";
80             cin >> idMenu;
81
82             for (auto m : databaseMenu) {
83                 if (m.id == idMenu) {
84                     p.menu.push_back(m.nama);
85                     ditemukan = true;
86                     break;
87                 }
88             }
89
90             if (!ditemukan) {
91                 cout << "ID Menu tidak valid.\n";
92                 inputValid = false;
93                 break;
94             }
95         }
96
97         if (!inputValid) continue;
98
99         cout << "\nJenis Pesanan:\n";
100         cout << "1. Makan di Tempat\n";
101         cout << "2. Sama Pulang\n";
102         cout << "Pilih: ";
103         cin >> jenis;
104
105         p.jenis = (jenis == 1) ? "Makan di Tempat" : "Sama Pulang";
106     }
107 }
```

Gambar 2.7 Code Membuat Pesanan

Fungsi ini merupakan inti dari proses pembuatan pesanan. Di dalam fungsi ini, sistem meminta pengguna untuk memasukkan nama customer, jumlah menu yang dipesan, memilih menu berdasarkan ID, serta menentukan jenis pesanan.

Fungsi ini juga melakukan proses validasi pesanan, di mana pengguna diminta untuk mengonfirmasi data pesanan. Jika validasi berhasil, pesanan akan dimasukkan ke dalam vector antrian dan status pesanan akan diatur menjadi “Sedang Diproses”.

2.2.7 Proses Validasi, Antrian, Pembayaran, dan Pembaruan Status Pesanan

```

120 // ===== VALIDASI =====
121 char konfirmasi;
122 cout << "Konfirmasi Pesanan (y/n): ";
123 cin >> konfirmasi;
124
125 if (konfirmasi == 'y' || konfirmasi == 'Y') {
126     selesai = true; // selesai loop, lanjut proses
127 } else {
128     int pilihan;
129     cout << "1. Perbaiki Menu";
130     cout << "2. Batalkan Pesanan\n";
131     cout << "Pilih: ";
132     cin >> pilihan;
133
134     if (pilihan == 1) {
135         cout << "Pesanan diperbaiki\n";
136         return;
137     }
138     // jika pilih 2 = loop ulang otomatis
139 }
140
141 // ===== ANTRIAN =====
142 p.status = "Sedang Diproses";
143 antrian.push_back(p);
144 cout << "Pesanan masuk antrian (" << p.jenis << ")!\n";
145
146 // ===== PEMBAYARAN =====
147 char bayar;
148 cout << "Apakah siap lanjut ke pembayaran (y/n)? ";
149 cin >> bayar;
150
151 if (bayar != 'y' && bayar != 'Y') {
152     cout << "Pembayaran ditunda!\n";
153     return;
154 }
155
156 cout << "Pembayaran... LUNAS!\n";
157
158 // ===== UPDATE STATUS =====
159 for (auto &p : antrian) {
160     if (p.id == p.id) {
161         p.status = "Selesai";
162         break;
163     }
164 }
165
166 cout << "Status pesanan: SELESAI!\n";
167
168 }

```

Gambar 2.8 Code Proses Validasi, Antrian, Pembayaran, dan Pembaruan Status Pesanan

Bagian kode ini merupakan lanjutan dari fungsi tambahPesanan() yang berperan penting dalam memastikan bahwa data pesanan yang dimasukkan oleh pengguna telah benar sebelum diproses lebih lanjut. Proses yang dilakukan meliputi validasi pesanan, pemasukan data ke dalam antrian, simulasi pembayaran, serta pembaruan status pesanan.

Tahap pertama adalah validasi pesanan. Sistem meminta konfirmasi kepada pengguna melalui input karakter (y/n) untuk memastikan bahwa data pesanan yang telah dimasukkan sudah sesuai. Jika pengguna memilih y, maka variabel selesai akan bernilai true sehingga perulangan berhenti dan proses dapat dilanjutkan. Sebaliknya, jika pengguna memilih selain y, sistem akan menampilkan dua opsi, yaitu memperbaiki menu atau membatalkan pesanan. Apabila pengguna memilih untuk membatalkan, fungsi akan langsung dihentikan menggunakan return, sehingga data pesanan tidak disimpan ke dalam sistem.

Setelah pesanan berhasil divalidasi, sistem memasuki tahap memasukkan pesanan ke dalam antrian. Pada tahap ini, status pesanan diatur menjadi “Sedang Diproses”, kemudian data pesanan dimasukkan ke dalam vector antrian menggunakan fungsi `push_back()`. Proses ini menandakan bahwa pesanan telah resmi tercatat dalam sistem antrian.

Tahap selanjutnya adalah pembayaran pesanan. Sistem kembali meminta konfirmasi kepada pengguna untuk melanjutkan ke proses pembayaran. Jika pengguna tidak melanjutkan pembayaran, maka sistem akan menampilkan pesan bahwa pembayaran ditunda dan fungsi akan dihentikan. Namun, jika pengguna melanjutkan pembayaran, sistem menampilkan informasi bahwa pembayaran telah berhasil atau lunas.

Setelah pembayaran berhasil, sistem melakukan pembaruan status pesanan. Proses ini dilakukan dengan melakukan perulangan pada seluruh data pesanan yang ada di dalam vector antrian. Ketika ID pesanan yang sedang diproses ditemukan, status pesanan akan diubah menjadi “Selesai”. Proses ini merupakan implementasi dari operasi Update dalam konsep CRUD, karena sistem memperbarui data yang sudah ada tanpa menambahkan atau menghapus data baru.

Secara keseluruhan, bagian kode ini memastikan bahwa setiap pesanan melalui tahapan yang terstruktur, mulai dari validasi, pencatatan ke dalam antrian, pembayaran, hingga pembaruan status. Dengan adanya alur ini, sistem antrian pemesanan menjadi lebih terkontrol, mengurangi kesalahan input, dan meningkatkan keakuratan data pesanan.

2.2.8 Fungsi Menampilkan Semua Pesanan (Read)

```
165 // ===== READ (URUT BERDASARKAN ID) =====
166 void tampilkanSemua() {
167     cout << "\n== DAFTAR PESANAN (URUT ID) ==\n";
168     if (antrian.empty()) {
169         cout << "Antrian kosong.\n";
170         return;
171     }
172     // Salin data agar urutan asli tidak berubah
173     vector<Pesanan> temp = antrian;
174
175     // Sort berdasarkan ID
176     sort(temp.begin(), temp.end(), [](Pesanan a, Pesanan b) {
177         return a.id < b.id;
178     });
179
180     for (auto p : temp) {
181         cout << "ID: " << p.id
182              << " | Nama: " << p.nama
183              << " | Menu: ";
184         for (int i = 0; i < p.menu.size(); i++) {
185             cout << p.menu[i];
186             if (i < p.menu.size() - 1) cout << ", ";
187         }
188         cout << " | Jenis: " << p.jenis
189              << " | Status: " << p.status << endl;
190         cout << "-----\n";
191     }
192 }
```

Gambar 2.9 Code Menampilkan Pesanan

Fungsi ini digunakan untuk menampilkan seluruh data pesanan yang ada di dalam sistem. Sebelum ditampilkan, data pesanan disalin ke dalam variabel sementara dan diurutkan berdasarkan ID agar tampilan lebih rapi. Fungsi ini membantu pengguna dalam melihat kondisi antrian secara keseluruhan.

2.2.9 Fungsi Cari Pesanan (Search)

```
198 // ===== SEARCH =====
199 void cariPesanan() {
200     int id;
201     cout << "\nMasukkan ID Pesanan: ";
202     cin >> id;
203
204     for (auto p : antrian) {
205         if (p.id == id) {
206             cout << "\nID      : " << p.id << endl;
207             cout << "Nama    : " << p.nama << endl;
208             cout << "Menu    : ";
209             for (int i = 0; i < p.menu.size(); i++) {
210                 cout << p.menu[i];
211                 if (i < p.menu.size() - 1) cout << ", ";
212             }
213             cout << endl;
214             cout << "Status : " << p.status << endl;
215             return;
216         }
217     }
218     cout << "Pesanan tidak ditemukan.\n";
219 }
```

Gambar 2.10 Code Search Pesanan

Fungsi ini digunakan untuk mencari data pesanan berdasarkan ID pesanan. Sistem akan membandingkan ID yang dimasukkan pengguna dengan data yang ada di dalam antrian. Jika ditemukan, sistem menampilkan detail pesanan, jika tidak maka sistem menampilkan pesan bahwa pesanan tidak ditemukan.

2.2.10 Fungsi Batalan Pesanan (Delete)

```
221 // ===== DELETE =====
222 void batalanPesanan() {
223     int id;
224     cout << "\nMasukkan ID Pesanan: ";
225     cin >> id;
226
227     for (int i = 0; i < antrian.size(); i++) {
228         if (antrian[i].id == id) {
229             antrian.erase(antrian.begin() + i);
230             cout << "Pesanan berhasil dibatalkan.\n";
231             return;
232         }
233     }
234     cout << "Pesanan tidak ditemukan.\n";
235 }
```

Gambar 2.11 Code Delete Pesanan

Fungsi ini berfungsi untuk menghapus pesanan dari sistem berdasarkan ID. Proses penghapusan dilakukan dengan menghapus elemen tertentu dari vector antrian. Fungsi ini merupakan implementasi dari operasi Delete dalam konsep CRUD.

2.2.11 Fungsi Sortir Menu Berdasarkan Harga (Sort)

```
237 // ===== SORT =====
238 void sortirMenuByHarga() {
239     int minHarga, maxHarga;
240
241     cout << "\n--- SORTIR MENU BERDASARKAN HARGA ---\n";
242     cout << "Masukkan harga minimum : Rp";
243     cin >> minHarga;
244     cout << "Masukkan harga maksimum : Rp";
245     cin >> maxHarga;
246
247     vector<Menuitem> hasil;
248
249     for (auto m : databaseMenu) {
250         if (m.harga >= minHarga && m.harga <= maxHarga) {
251             hasil.push_back(m);
252         }
253     }
254
255     if (hasil.empty()) {
256         cout << "Tidak ada menu pada rentang harga tersebut.\n";
257         return;
258     }
259
260     sort(hasil.begin(), hasil.end(), [](Menuitem a, Menuitem b) {
261         return a.harga < b.harga;
262     });
263
264     cout << "\n--- HASIL SORTIR MENU ---\n";
265     cout << "ID | Nama Menu | Kategori | Harga\n";
266     cout << "-----\n";
267
268     for (auto m : hasil) {
269         cout << m.id << " | "
270              << m.nama << " | "
271              << m.kategori << " | Rp"
272              << m.harga << endl;
273     }
274 }
```

Gambar 2.12 Code Sortir Pesanan

Fungsi ini digunakan untuk mengurutkan menu berdasarkan harga sesuai dengan rentang harga yang dimasukkan oleh pengguna. Menu yang telah disortir akan ditampilkan dari harga terendah hingga tertinggi, sehingga memudahkan pengguna dalam memilih menu.

2.2.12 Fungsi Menu Antrian

```
276 // ===== SUB MENU =====
277 void menuAntrian() {
278     int pilih;
279     do {
280         cout << "\n== MENU ANTRIAN ==\n";
281         cout << "1. Tampilkan Semua Pesanan\n";
282         cout << "2. Cari Pesanan\n";
283         cout << "0. Kembali\n";
284         cout << "Pilih: ";
285         cin >> pilih;
286
287         switch (pilih) {
288             case 1: tampilkanSemua(); break;
289             case 2: cariPesanan(); break;
290             case 0: break;
291             default: cout << "Pilihan tidak valid.\n";
292         }
293     } while (pilih != 0);
294 }
```

Gambar 2.13 Code Menu Antrian

Fungsi ini berfungsi sebagai submenu yang menyediakan fitur untuk menampilkan seluruh pesanan atau mencari pesanan. Fungsi ini membantu mengelompokkan fitur terkait antrian agar program lebih terstruktur dan mudah digunakan.

2.2.13 Fungsi Main

```
296 // ===== MAIN =====
297 int main() {
298     int menu;
299     do {
300         // ===== TAMPILKAN DATABASE MENU DI AWAL =====
301         tampilkanMenu();
302
303         cout << "\n== MENU UTAMA ==\n";
304         cout << "1. Tambah Pesanan\n";
305         cout << "2. Lihat Antrian\n";
306         cout << "3. Sortir Menu Berdasarkan Harga\n";
307         cout << "0. Keluar\n";
308
309         cout << "Pilih Menu: ";
310         cin >> menu;
311
312         switch (menu) {
313             case 1: tambahPesanan(); break;
314             case 2: menuAntrian(); break;
315             case 3: sortirMenuByHarga(); break;
316             case 0: cout << "Program selesai.\n"; break;
317             default: cout << "Pilihan tidak valid.\n";
318         }
319     } while (menu != 0);
320
321     return 0;
322 }
```

Gambar 2.14 Code Main

Fungsi main() merupakan pusat kendali dari seluruh program. Pada fungsi ini, sistem menampilkan menu utama dan menerima input pilihan dari pengguna. Berdasarkan pilihan tersebut, sistem akan memanggil fungsi yang sesuai, seperti menambah pesanan, melihat antrian, atau melakukan sortir menu.

Fungsi main() menggunakan perulangan agar program dapat berjalan terus-menerus hingga pengguna memilih untuk keluar dari sistem. Dengan demikian, fungsi main() berperan sebagai pengatur alur utama sistem antrian pemesanan.

2.3 Test

Tahap pengujian dilakukan untuk memastikan bahwa seluruh fitur pada sistem antrian pemesanan berjalan sesuai dengan perancangan dan fungsi yang diharapkan. Pengujian dilakukan dengan cara menjalankan program melalui console dan mencoba setiap menu yang tersedia berdasarkan alur penggunaan sistem.

Pada saat program dijalankan, sistem pertama kali menampilkan menu utama yang berisi beberapa pilihan fitur. Pengguna diminta untuk memasukkan pilihan menu sesuai dengan kebutuhan yang diinginkan.

1. Tampilan menu

```
PS C:\Users\USER> cd "d:\SEMESTER 1\ALPRO\min

=== DAFTAR MENU ===
ID | Nama Menu      | Kategori | Harga
-----
1  | Nasi Goreng    | Makanan | Rp20000
2  | Mie Ayam       | Makanan | Rp18000
3  | Ayam Geprek    | Makanan | Rp22000
4  | Sate Ayam      | Makanan | Rp25000
5  | Bakso          | Makanan | Rp20000
6  | Es Teh         | Minuman | Rp5000
7  | Es Jeruk       | Minuman | Rp7000
8  | Kopi Hitam     | Minuman | Rp8000
9  | Teh Hangat     | Minuman | Rp5000
10 | Jus Alpukat    | Minuman | Rp12000
```

Gambar 2.15 Output Tampilan Menu

Ketika program dijalankan, sistem secara otomatis menampilkan menu utama. Pada tahap ini, pengguna dapat melihat seluruh menu yang tersedia, seperti id menu, nama menu, kategori, dan harga menu.

2. Pilihan menu

```
=== MENU UTAMA ===
1. Tambah Pesanan
2. Lihat Antrian
3. Sortir Menu Berdasarkan Harga
0. Keluar
Pilih Menu:
```

Gambar 2.16 Output Pilihan Menu

Dibagian menu utama tak hanya semua menu yang ditampilkan tetapi terdapat juga pilihan menu, di proses ini pengguna diminta untuk memasukkan nomor pilihan menu. Sistem kemudian membaca input tersebut dan mengarahkan pengguna ke proses yang sesuai dengan pilihan yang dipilih

3. Tambahkan pesanan

```
=== MENU UTAMA ===
1. Tambah Pesanan
2. Lihat Antrian
3. Sortir Menu Berdasarkan Harga
0. Keluar
Pilih Menu: 1

=== PEMBUATAN PESANAN ===
Masukkan Nama Customer : Dimas

Masukkan jumlah menu yang dipesan (1 - 2): 1

=== DAFTAR MENU ===
ID | Nama Menu      | Kategori | Harga
-----
1  | Nasi Goreng    | Makanan  | Rp20000
2  | Mie Ayam       | Makanan  | Rp18000
3  | Ayam Geprek    | Makanan  | Rp22000
4  | Sate Ayam      | Makanan  | Rp25000
5  | Bakso          | Makanan  | Rp20000
6  | Es Teh         | Minuman  | Rp5000
7  | Es Jeruk       | Minuman  | Rp7000
8  | Kopi Hitam     | Minuman  | Rp8000
9  | Teh Hangat     | Minuman  | Rp5000
10 | Jus Alpukat    | Minuman  | Rp12000

Pilih ID Menu ke-1 : 1

Jenis Pesanan:
1. Makan di Tempat
2. Bawa Pulang
Pilih: 1
```

Gambar 2.17 Output Tambahkan Pesanan

Jika pengguna memilih menu Tambah Pesanan, maka sistem akan meminta pengguna untuk menginput nama pelanggan, jumlah menu yang dipesan, serta ID menu yang dipilih. Setelah seluruh data dimasukkan, sistem akan meminta pengguna untuk memilih apakah pesannya akan dibawa pulang atau makan ditempat.

4. Validasi

```
Validasi Pesanan (y/n)? y
Pesanan masuk antrian (Makan di Tempat).
```

Gambar 2.18 Output Validasi (ya)

Pada tahap validasi, pengguna diminta untuk memilih tindakan lanjutan. Jika pengguna memilih yes, maka pesanan akan menuju ke antrian. Namun, jika pengguna memilih no, maka pengguna akan diantarkan ke pilihan yang lain.

5. Perbaikan pesanan

```
Validasi Pesanan (y/n)? n

1. Perbaiki Menu
2. Batalkan Pesanan
Pilih: 1

Masukkan jumlah menu yang dipesan (1 - 2): 1

=== DAFTAR MENU ===
ID | Nama Menu          | Kategori | Harga
-----
1  | Nasi Goreng        | Makanan | Rp20000
2  | Mie Ayam           | Makanan | Rp18000
3  | Ayam Geprek        | Makanan | Rp22000
4  | Sate Ayam          | Makanan | Rp25000
5  | Bakso              | Makanan | Rp20000
6  | Es Teh             | Minuman | Rp5000
7  | Es Jeruk           | Minuman | Rp7000
8  | Kopi Hitam         | Minuman | Rp8000
9  | Teh Hangat         | Minuman | Rp5000
10 | Jus Alpukat        | Minuman | Rp12000
Pilih ID Menu ke-1 : 2

Jenis Pesanan:
1. Makan di Tempat
2. Bawa Pulang
Pilih: 1
```

Gambar 2.19 Output Validasi (Tidak)

Setelah memilih no, pengguna akan dikasih dua opsi yaitu perbaiki pesanan dan batalkan pesanan. Jika pengguna memilih untuk memperbaiki pesanan, maka sistem akan kembali ke proses tambahkan pemesanan dan meminta pengguna untuk mengisi kembali bagian tambahkan pemesanan

6. Batalkan pesanan

```
Validasi Pesanan (y/n)? n
Validasi ditolak.
1. Perbaiki Pesanan
2. Batalkan Pesanan
Pilih: 2
Pesanan dibatalkan.
```

Gambar 2.20 Output Batalkan pesanan

Namun, jika pengguna memilih untuk membatalkan pesanan, maka sistem akan menghentikan proses penambahan pesanan dan kembali ke menu utama.

7. Pembayaran

```
Pesanan siap. Lanjut ke pembayaran (y/n)? n
Pembayaran ditunda.

Pesanan siap. Lanjut ke pembayaran (y/n)? y
Pembayaran... [LUNAS]
Status pesanan: SELESAI
```

Gambar 2.21 Output Pembayaran

Setelah pesanan berhasil divalidasi dan dimasukkan ke dalam antrian, sistem meminta konfirmasi kepada pengguna untuk melanjutkan proses pembayaran. Jika pengguna menyetujui, sistem menampilkan informasi bahwa pembayaran berhasil dan status pesanan diperbarui menjadi “Selesai”. Namun jika pengguna tidak menyetujui pembayaran akan ditunda dan status pesanan akan menjadi sedang di proses.

8. Sortir menu

```
=== MENU UTAMA ===
1. Tambah Pesanan
2. Lihat Antrian
3. Sortir Menu Berdasarkan Harga
0. Keluar
Pilih Menu: 3

=== SORTIR MENU BERDASARKAN HARGA ===
Masukkan harga minimum : Rp5000
Masukkan harga maksimum : Rp20000

=== HASIL SORTIR MENU ===
ID | Nama Menu | Kategori | Harga
-----
6 | Es Teh | Minuman | Rp5000
9 | Teh Hangat | Minuman | Rp5000
7 | Es Jeruk | Minuman | Rp7000
8 | Kopi Hitam | Minuman | Rp8000
10 | Jus Alpukat | Minuman | Rp12000
2 | Mie Ayam | Makanan | Rp18000
1 | Nasi Goreng | Makanan | Rp20000
5 | Bakso | Makanan | Rp20000
```

Gambar 2.22 Output Sortir menu

Apabila pengguna memilih menu Sortir Menu, maka sistem meminta pengguna untuk memasukkan rentang harga minimum dan maksimum. Setelah itu, sistem mengolah data menu dan menampilkan daftar menu yang telah diurutkan berdasarkan harga sesuai dengan rentang yang dimasukkan.

9. Lihat semua antrian

```
=== MENU UTAMA ===
1. Tambah Pesanan
2. Lihat Antrian
3. Sortir Menu Berdasarkan Harga
0. Keluar
Pilih Menu: 2

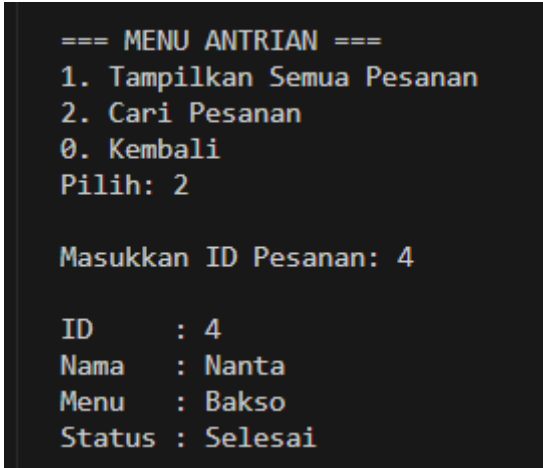
=== MENU ANTRIAN ===
1. Tampilkan Semua Pesanan
2. Cari Pesanan
0. Kembali
Pilih: 1

=== DAFTAR PESANAN (URUT ID) ===
ID: 1 | Nama: Dimas | Menu: Mie Ayam | Jenis: Makan di Tempat | Status: Sedang Diproses
-----
ID: 2 | Nama: Dimas | Menu: Nasi Goreng | Jenis: Makan di Tempat | Status: Selesai
-----
ID: 3 | Nama: Fandi | Menu: Ayam Geprek, Es Teh | Jenis: Makan di Tempat | Status: Selesai
-----
ID: 4 | Nama: Nanta | Menu: Bakso | Jenis: Bawa Pulang | Status: Selesai
-----
ID: 5 | Nama: Tegar | Menu: Mie Ayam, Kopi Hitam | Jenis: Bawa Pulang | Status: Selesai
-----
```

Gambar 2.23 Output Lihat Antrian

Jika pengguna memilih menu Lihat Semua Antrian, maka sistem akan menampilkan seluruh data pesanan yang tersimpan, lengkap dengan ID pesanan, nama pelanggan, menu yang dipesan, jenis pesanan, dan status pesanan.

10. Cari antrian berdasarkan id



```
=== MENU ANTRIAN ===
1. Tampilkan Semua Pesanan
2. Cari Pesanan
0. Kembali
Pilih: 2

Masukkan ID Pesanan: 4

ID      : 4
Nama    : Nanta
Menu    : Bakso
Status  : Selesai
```

Gambar 2.24 Output Cari Antrian

Pada menu pencarian, pengguna diminta untuk memasukkan ID pesanan.

Sistem kemudian melakukan proses pencarian dan:

- Menampilkan detail pesanan jika ID ditemukan.
- Menampilkan pesan bahwa pesanan tidak ditemukan jika ID tidak ada di dalam sistem.

Setelah setiap proses selesai dijalankan, sistem akan kembali ke menu utama, sehingga pengguna dapat melakukan proses lainnya atau memilih untuk keluar dari program. Berdasarkan hasil pengujian tersebut, dapat disimpulkan bahwa seluruh fitur sistem berjalan dengan baik dan sesuai dengan alur yang telah dirancang.

BAB III

PENUTUP

3.1 Kesimpulan

Jadi kesimpulannya berdasarkan dengan hasil perancangan, implementasi, dan pengujian sistem antrian pemesanan berbasis C++ yang telah dilakukan, dapat disimpulkan bahwa sistem ini berhasil memenuhi tujuan yang telah ditetapkan. Sistem antrian pemesanan mampu mengelola data pesanan secara terstruktur dan sistematis dengan menerapkan konsep CRUD (Create, Read, Update, Delete), serta didukung oleh fitur sorting dan searching yang memudahkan pengolahan dan pencarian data.

Penerapan flowchart sebagai desain awal terbukti membantu dalam menggambarkan alur logika sistem secara jelas dan terarah, sehingga proses implementasi ke dalam kode program C++ dapat dilakukan dengan lebih mudah dan terorganisir. Setiap fungsi yang dibangun, mulai dari penambahan pesanan, penampilan data, pencarian, penghapusan, hingga pengurutan menu berdasarkan harga, berjalan sesuai dengan perancangan dan saling terintegrasi dengan baik.

Hasil pengujian sistem menunjukkan bahwa program dapat dijalankan dengan baik melalui antarmuka berbasis console, mampu menangani input pengguna, meminimalkan kesalahan pencatatan, serta menampilkan informasi antrian secara akurat. Dengan demikian, sistem antrian pemesanan ini tidak hanya berfungsi sebagai sarana latihan pemrograman C++, tetapi juga sebagai contoh penerapan konsep sistem informasi sederhana dalam menyelesaikan permasalahan nyata terkait pengelolaan antrian.

Secara keseluruhan, mini project ini dapat dijadikan sebagai dasar pengembangan sistem yang lebih kompleks di masa mendatang, misalnya dengan penambahan fitur penyimpanan data ke database, antarmuka grafis, atau pengembangan ke platform berbasis web maupun mobile, sehingga manfaat dan efisiensinya dapat ditingkatkan lebih lanjut.

3.2 Link Video Presentasi

https://youtu.be/S_NHaZvOvbg?si=yrAPlv1LEGgowaT8