

IMPLEMENTASI SISTEM ANTRIAN RUMAH SAKIT BERBASIS PRIORITAS MENGGUNAKAN METODE QUEUE DAN BUBBLE SHORT

Azril Arfansyah¹, Melika Debiyana Putri², Calvin Sahputra Buulolo³, Triwanti Andini Hutasoit⁴

azrilarfansyah403@gmail.com¹, melikadebiana@gmail.com², calvinsahputra4@gmail.com³,
triwantiandinihutasoit@gmail.com⁴

Universitas Negeri Medan

ABSTRAK

Sistem antrian pada rumah sakit memiliki peran penting dalam meningkatkan kualitas pelayanan kesehatan. Antrian yang tidak teratur dapat menimbulkan keterlambatan penanganan, terutama pada pasien dengan kondisi darurat. Penelitian ini membahas implementasi sistem antrian rumah sakit berbasis prioritas menggunakan metode queue dan algoritma bubble sort. Metode queue digunakan untuk mengelola aliran pasien secara terstruktur, sementara algoritma bubble sort dimanfaatkan untuk mengurutkan pasien berdasarkan tingkat prioritas medis. Dengan pendekatan ini, pasien dengan kondisi lebih kritis dapat memperoleh pelayanan terlebih dahulu dibandingkan pasien dengan tingkat prioritas lebih rendah. Hasil implementasi menunjukkan bahwa sistem mampu mengatur antrian secara efektif dan adil sesuai prioritas pasien, sehingga dapat mendukung peningkatan efisiensi serta kualitas pelayanan rumah sakit.

Kata Kunci: Sistem Antrian, Prioritas Pasien, Queue, Bubble Sort, Rumah Sakit.

ABSTRACT

The queuing system in hospitals plays a crucial role in improving the quality of healthcare services. Unorganized queues may cause delays in treatment, especially for patients in emergency conditions. This study discusses the implementation of a hospital queuing system based on priority using the queue method and the bubble sort algorithm. The queue method is applied to manage patient flow in a structured manner, while the bubble sort algorithm is used to arrange patients according to their medical priority levels. With this approach, patients with more critical conditions receive treatment earlier than those with lower priority levels. The implementation results show that the system can manage queues effectively and fairly based on patient priorities, thereby supporting improved efficiency and quality of hospital services.

Keywords: Queuing System, Patient Priority, Queue, Bubble Sort, Hospital.

PENDAHULUAN

Rumah sakit merupakan salah satu institusi pelayanan kesehatan yang berfungsi memberikan penanganan medis bagi masyarakat dengan berbagai tingkat kondisi kesehatan. Dalam praktiknya, rumah sakit sering menghadapi permasalahan antrian pasien, terutama pada bagian rawat jalan dan instalasi gawat darurat. Antrian yang tidak terkelola dengan baik dapat menyebabkan keterlambatan pelayanan, penumpukan pasien, hingga menurunnya kualitas layanan kesehatan.

Salah satu aspek penting dalam sistem antrian rumah sakit adalah penentuan prioritas pasien. Pasien dengan kondisi darurat seharusnya memperoleh pelayanan terlebih dahulu dibandingkan pasien dengan kondisi ringan. Namun, pada sistem manual, penentuan prioritas ini seringkali tidak berjalan optimal karena keterbatasan tenaga dan metode pengelolaan. Oleh karena itu, dibutuhkan sebuah sistem antrian berbasis prioritas yang mampu mengatur alur pasien secara lebih efisien, adil, dan sesuai tingkat urgensi.

Dalam penelitian ini, metode queue digunakan untuk mengelola aliran pasien secara terstruktur, sedangkan algoritma bubble sort digunakan untuk mengurutkan pasien

berdasarkan tingkat prioritas medis. Kombinasi kedua metode ini diharapkan dapat menghasilkan sistem antrian yang mampu memberikan pelayanan lebih cepat bagi pasien kritis tanpa mengabaikan pasien lainnya. Implementasi sistem antrian berbasis prioritas ini diharapkan mampu meningkatkan efisiensi operasional rumah sakit serta kualitas pelayanan kepada masyarakat.

TINJAUAN PUSTAKA

Konsep Antrian (Queue)

Queue adalah salah satu struktur data linear yang menggunakan prinsip First In First Out (FIFO), artinya elemen yang pertama masuk akan dilayani terlebih dahulu. Operasi utama pada queue meliputi enqueue untuk menambahkan data dan dequeue untuk menghapus data dari antrian (Zein & Tristianto, 2025). Modul pembelajaran struktur data juga menegaskan bahwa queue sangat cocok digunakan untuk memodelkan antrian nyata seperti pelayanan di rumah sakit, bank, maupun kasir (Universitas BSI, 2018).

Penelitian yang dilakukan oleh Pratama dkk. (2022) menunjukkan penerapan algoritma queue pada sistem penjualan minuman berbasis web. Sistem tersebut mampu mengatur pesanan pelanggan secara teratur sehingga mengurangi kekacauan dalam antrian. Dalam konteks pelayanan kesehatan, Aryandi dkk. (2023) menerapkan priority queue pada sistem antrian rumah sakit berbasis web, di mana pasien dengan prioritas lebih tinggi seperti lansia dan pasien darurat memperoleh pelayanan lebih dahulu dibanding pasien umum.

Algoritma Bubble Sort

Bubble sort adalah algoritma pengurutan sederhana yang bekerja dengan membandingkan dua elemen berurutan dan menukar jika tidak sesuai, hingga seluruh data terurut. Algoritma ini termasuk kategori brute force, dengan kompleksitas waktu relatif tinggi tetapi mudah diimplementasikan (Zein & Tristianto, 2025).

Penelitian Kusuma Jaya dkk. (2020) menjelaskan bahwa bubble sort dikategorikan sebagai metode brute force yang mudah diterapkan meskipun tidak efisien untuk data dalam jumlah besar. Sementara itu, algoritma lain seperti merge sort yang berbasis divide and conquer lebih cepat, namun bubble sort tetap relevan sebagai algoritma dasar untuk pengurutan sederhana.

Implementasi Queue dan Bubble Sort dalam Sistem Antrian

Penerapan queue dan algoritma sorting sering dipadukan dalam sistem antrian. Pada penelitian Pratama dkk. (2022), queue digunakan untuk mengatur antrian pelanggan berdasarkan urutan kedatangan. Sedangkan Aryandi dkk. (2023) menekankan bahwa priority queue sangat penting dalam sistem rumah sakit untuk memastikan pasien prioritas lebih cepat dilayani.

Dalam penelitian ini, metode queue digunakan untuk mengelola alur pasien secara terstruktur, sedangkan bubble sort digunakan untuk mengurutkan pasien berdasarkan tingkat prioritas medis (Biji dkk., 2023). Dengan kombinasi tersebut, pasien dengan kondisi darurat dapat segera memperoleh pelayanan tanpa mengabaikan pasien dengan kondisi lebih ringan. Pendekatan ini terbukti meningkatkan efisiensi serta kualitas pelayanan rumah sakit.

METODOLOGI

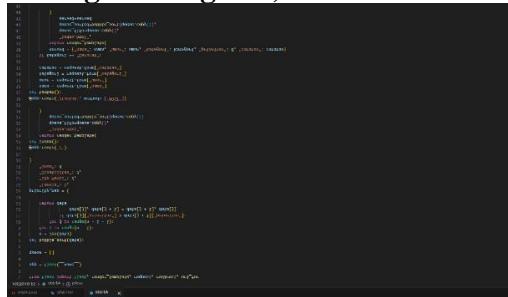
Penelitian ini menggunakan pendekatan terapan (applied research) dengan metode implementasi sistem dan eksperimen algoritmik. Fokus utama penelitian adalah penerapan struktur data Queue (antrian) dan algoritma Bubble Sort (pengurutan prioritas) untuk mengatur antrian pasien rumah sakit berdasarkan tingkat urgensi medis.

Implementasi Kode

Sistem antrian yang dikembangkan diberi nama MedLine (Sistem Antrian Pasien) dan diimplementasikan menggunakan bahasa pemrograman Python dengan framework Flask.

Tujuannya adalah menciptakan aplikasi berbasis web yang mampu menampilkan proses antrian secara interaktif. Struktur kode utama dibagi menjadi tiga komponen:

1. File "app.py" (Backend/Logika Program)



```
from queue import Queue
import os
import time

class Antrian:
    def __init__(self):
        self.queue = Queue()

    def tambah_antrian(self, pasien):
        self.queue.put(pasien)

    def layani_pasien(self):
        if not self.queue.empty():
            pasien = self.queue.get()
            print(f"Menyajikan pasien: {pasien['id']} {pasien['nama']} {pasien['prioritas']}")

            # Menghitung total waktu antrean
            total_waktu = 0
            for pasien in self.queue.queue:
                total_waktu += pasien['waktu']

            print(f"Total Waktu Antrean: {total_waktu} menit")

            # Mengurutkan pasien berdasarkan prioritas
            self.queue.queue.sort(key=lambda pasien: pasien['prioritas'])

    def cetak_antrian(self):
        if not self.queue.empty():
            print("Antrian saat ini:")
            for pasien in self.queue.queue:
                print(f"ID: {pasien['id']}, Nama: {pasien['nama']}, Prioritas: {pasien['prioritas']}, Waktu: {pasien['waktu']} menit")

    def selesai(self):
        print("Antrian selesai.")

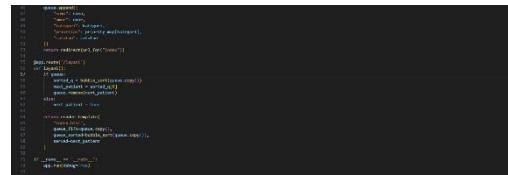

def main():
    antrian = Antrian()

    antrian.tambah_antrian({'id': 1, 'nama': 'Andi', 'prioritas': 1, 'waktu': 10})
    antrian.tambah_antrian({'id': 2, 'nama': 'Budi', 'prioritas': 2, 'waktu': 15})
    antrian.tambah_antrian({'id': 3, 'nama': 'Catur', 'prioritas': 3, 'waktu': 20})
    antrian.tambah_antrian({'id': 4, 'nama': 'Dedi', 'prioritas': 4, 'waktu': 25})

    antrian.layani_pasien()
    antrian.cetak_antrian()

    antrian.selesai()

if __name__ == "__main__":
    main()
```



```
<html>
<head>
    <title>Antrian Pasien</title>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
</head>
<body>
    <h2>Formulir Pasien</h2>
    <form>
        <label>ID:</label> <input type="text" name="id" value="1" />
        <label>Nama:</label> <input type="text" name="nama" value="Andi" />
        <label>Prioritas:</label> <input type="text" name="prioritas" value="1" />
        <label>Waktu:</label> <input type="text" name="waktu" value="10" />
        <input type="button" value="Layani Pasien" />
    </form>

    <table border="1">
        <thead>
            <tr>
                <th>ID</th>
                <th>Nama</th>
                <th>Prioritas</th>
                <th>Waktu</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>1</td>
                <td>Andi</td>
                <td>1</td>
                <td>10</td>
            </tr>
            <tr>
                <td>2</td>
                <td>Budi</td>
                <td>2</td>
                <td>15</td>
            </tr>
            <tr>
                <td>3</td>
                <td>Catur</td>
                <td>3</td>
                <td>20</td>
            </tr>
            <tr>
                <td>4</td>
                <td>Dedi</td>
                <td>4</td>
                <td>25</td>
            </tr>
        </tbody>
    </table>

    <table border="1">
        <thead>
            <tr>
                <th>ID</th>
                <th>Nama</th>
                <th>Prioritas</th>
                <th>Waktu</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>1</td>
                <td>Andi</td>
                <td>1</td>
                <td>10</td>
            </tr>
            <tr>
                <td>2</td>
                <td>Budi</td>
                <td>2</td>
                <td>15</td>
            </tr>
            <tr>
                <td>3</td>
                <td>Catur</td>
                <td>3</td>
                <td>20</td>
            </tr>
            <tr>
                <td>4</td>
                <td>Dedi</td>
                <td>4</td>
                <td>25</td>
            </tr>
        </tbody>
    </table>
</body>
</html>
```

- Mengatur proses input pasien, penyimpanan data antrian, dan pengurutan prioritas.
- Menggunakan dua konsep utama:
- Queue (FIFO) → menyimpan urutan kedatangan pasien.
- Bubble Sort → mengurutkan pasien berdasarkan prioritas medis.

Pemetaan kategori pasien:

- Darurat langsung dilayani, tidak masuk antrian.
- Lansia prioritas 1
- Ibu Hamil prioritas 2
- Disabilitas prioritas 3
- Umum prioritas 4

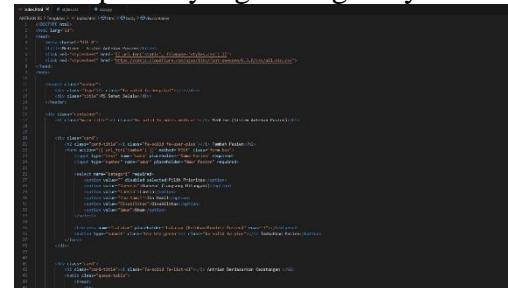
Logika kerja:

- Jika kategori pasien Darurat, pasien langsung masuk ke daftar Pasien yang Sedang Dilayani tanpa dimasukkan ke antrian.
- Pasien kategori lain dimasukkan ke dalam queue dan diurutkan menggunakan Bubble Sort.
- Tombol “Layani Pasien” akan memanggil pasien dengan prioritas tertinggi dan menghapusnya dari antrian (dequeue).

2. File "index.html" (Antarmuka Pengguna)

- Menyediakan form untuk menambah pasien baru.
- Menampilkan dua tabel:

Antrian Berdasarkan Kedatangan (FIFO), Antrian Berdasarkan Prioritas (Hasil Bubble Sort) Menampilkan data pasien yang sedang dilayani.



```
<html>
<head>
    <title>Antrian Pasien</title>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
</head>
<body>
    <h2>Formulir Pasien</h2>
    <form>
        <label>ID:</label> <input type="text" name="id" value="1" />
        <label>Nama:</label> <input type="text" name="nama" value="Andi" />
        <label>Prioritas:</label> <input type="text" name="prioritas" value="1" />
        <label>Waktu:</label> <input type="text" name="waktu" value="10" />
        <input type="button" value="Layani Pasien" />
    </form>

    <table border="1">
        <thead>
            <tr>
                <th>ID</th>
                <th>Nama</th>
                <th>Prioritas</th>
                <th>Waktu</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>1</td>
                <td>Andi</td>
                <td>1</td>
                <td>10</td>
            </tr>
            <tr>
                <td>2</td>
                <td>Budi</td>
                <td>2</td>
                <td>15</td>
            </tr>
            <tr>
                <td>3</td>
                <td>Catur</td>
                <td>3</td>
                <td>20</td>
            </tr>
            <tr>
                <td>4</td>
                <td>Dedi</td>
                <td>4</td>
                <td>25</td>
            </tr>
        </tbody>
    </table>

    <table border="1">
        <thead>
            <tr>
                <th>ID</th>
                <th>Nama</th>
                <th>Prioritas</th>
                <th>Waktu</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>1</td>
                <td>Andi</td>
                <td>1</td>
                <td>10</td>
            </tr>
            <tr>
                <td>2</td>
                <td>Budi</td>
                <td>2</td>
                <td>15</td>
            </tr>
            <tr>
                <td>3</td>
                <td>Catur</td>
                <td>3</td>
                <td>20</td>
            </tr>
            <tr>
                <td>4</td>
                <td>Dedi</td>
                <td>4</td>
                <td>25</td>
            </tr>
        </tbody>
    </table>
</body>
</html>
```



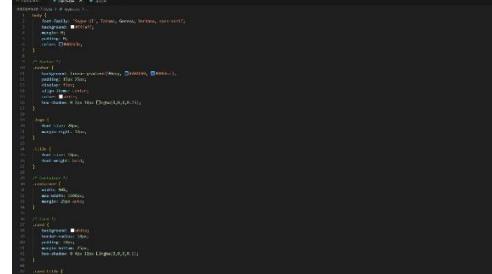
```
1 // Header file for Queue
2
3 #ifndef QUEUE_H
4 #define QUEUE_H
5
6 #include <cs50.h>
7 #include <stdio.h>
8 #include <stdlib.h>
9
10 #define MAX 100
11
12 struct Queue
13 {
14     int front;
15     int rear;
16     int size;
17     int queue[MAX];
18 };
19
20 void enqueue(struct Queue*, int);
21 int dequeue(struct Queue*, int);
22 int peek(struct Queue*, int);
23 int isFull(struct Queue*, int);
24 int isEmpty(struct Queue*, int);
25
26 #endif
```



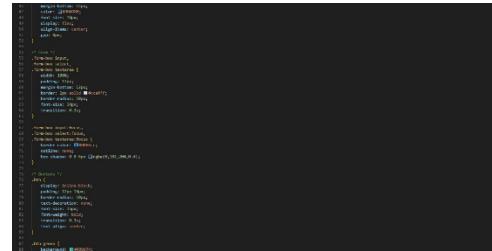
```
1 // Implementation of Queue
2
3 #include "queue.h"
4
5 void enqueue(struct Queue* q, int value)
6 {
7     if (isFull(q))
8     {
9         printf("Queue is full\n");
10    }
11    else
12    {
13        q->queue[q->rear] = value;
14        q->rear++;
15    }
16 }
17
18 int dequeue(struct Queue* q, int* value)
19 {
20     if (isEmpty(q))
21     {
22         printf("Queue is empty\n");
23        return 0;
24    }
25    else
26    {
27        *value = q->queue[q->front];
28        q->front++;
29        return 1;
30    }
31 }
32
33 int peek(struct Queue* q, int value)
34 {
35     if (isEmpty(q))
36     {
37         printf("Queue is empty\n");
38        return 0;
39    }
40    else
41    {
42        value = q->queue[q->front];
43        return 1;
44    }
45 }
46
47 int isFull(struct Queue* q)
48 {
49     if (q->rear == MAX)
50     {
51         return 1;
52     }
53     else
54     {
55         return 0;
56     }
57 }
58
59 int isEmpty(struct Queue* q)
60 {
61     if (q->front == q->rear)
62     {
63         return 1;
64     }
65     else
66     {
67         return 0;
68     }
69 }
```

3. File "styles.css" (Tampilan Sistem)

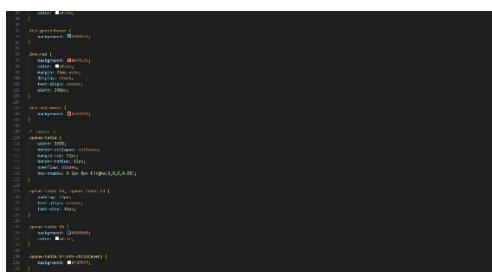
Mengatur tampilan antarmuka agar responsif, profesional, dan mudah digunakan oleh pengguna.



```
1 /* Header */
2
3 header {
4     background-color: #333;
5     color: white;
6     padding: 10px;
7     text-align: center;
8 }
9
10 /* Navigation */
11
12 nav {
13     background-color: #333;
14     color: white;
15     display: flex;
16     justify-content: space-around;
17     padding: 10px;
18 }
19
20 nav a {
21     color: inherit;
22     text-decoration: none;
23 }
24
25 /* Main Content */
26
27 main {
28     background-color: #f0f0f0;
29     padding: 20px;
30 }
31
32 /* Footer */
33
34 footer {
35     background-color: #333;
36     color: white;
37     padding: 10px;
38     text-align: center;
39 }
```



```
1 /* Main Content */
2
3 main {
4     padding: 20px;
5 }
6
7 table {
8     border-collapse: collapse;
9     width: 100%;
10 }
11
12 table tr {
13     border-top: 1px solid black;
14     border-bottom: 1px solid black;
15     padding: 5px;
16 }
17
18 table tr td {
19     padding: 10px;
20 }
21
22 table tr td:first-child {
23     text-align: left;
24 }
25
26 table tr td:last-child {
27     text-align: right;
28 }
```



```
1 /* Main Content */
2
3 main {
4     padding: 20px;
5 }
6
7 table {
8     border-collapse: collapse;
9     width: 100%;
10 }
11
12 table tr {
13     border-top: 1px solid black;
14     border-bottom: 1px solid black;
15     padding: 5px;
16 }
17
18 table tr td {
19     padding: 10px;
20 }
21
22 table tr td:first-child {
23     text-align: left;
24 }
25
26 table tr td:last-child {
27     text-align: right;
28 }
```



```
1 /* Main Content */
2
3 main {
4     padding: 20px;
5 }
6
7 table {
8     border-collapse: collapse;
9     width: 100%;
10 }
11
12 table tr {
13     border-top: 1px solid black;
14     border-bottom: 1px solid black;
15     padding: 5px;
16 }
17
18 table tr td {
19     padding: 10px;
20 }
21
22 table tr td:first-child {
23     text-align: left;
24 }
25
26 table tr td:last-child {
27     text-align: right;
28 }
```

HASIL DAN PEMBAHASAN

Hasil implementasi menunjukkan bahwa sistem dapat menjalankan logika Queue dan Bubble Sort dengan baik. Berikut pembahasan berdasarkan dua tampilan utama sistem.

The screenshot shows the MedLine patient management system. At the top, there is a header with the system name and a search bar. Below the header, the 'Tambah Pasien' (Add Patient) form is displayed, containing fields for 'Nama' (Name), 'Umur' (Age), 'Kategori' (Category), and 'Catatan' (Notes). A 'Simpan' (Save) button is at the bottom of this form. Below the form is a table titled 'Antrian Berdasarkan Kedatangan' (Arrival Queue). The table has columns for 'Nama' (Name), 'Umur' (Age), 'Kategori' (Category), 'Nomor' (Number), and 'Catatan' (Notes). The data in the table is as follows:

Nama	Umur	Kategori	Nomor	Catatan
Citra	20	Umum	1	Citra dari umur 20
Budi	60	Lansia	2	Budi dari umur 60
Andi	35	Darurat	3	Andi dari umur 35
Eka	40	Disabilitas	4	Eka dari umur 40
Farhan	25	Umum	5	Farhan dari umur 25

Gambar 1. Tampilan Awal Sistem dan Penambahan Pasien

Pada gambar ini terlihat halaman utama MedLine, yang terdiri dari tiga bagian utama:

1. Form Tambah Pasien

Digunakan untuk memasukkan data pasien berupa nama, umur, kategori, dan catatan kondisi. Setelah menekan tombol “Tambahkan Pasien”, sistem akan:

1. Memasukkan pasien ke dalam antrian FIFO.
2. Melakukan pengurutan antrian berdasarkan prioritas medis menggunakan Bubble Sort.

Jika kategori pasien adalah Darurat, maka pasien langsung masuk ke daftar Pasien yang Sedang Dilayani.

2. Tabel Antrian Berdasarkan Kedatangan (FIFO)

Menampilkan urutan pasien sesuai waktu pendaftaran:

- Citra (Umum), Budi (Lansia), Dewi (Ibu Hamil), Eka (Disabilitas), dan Farhan (Umum). Urutan ini mencerminkan prinsip First In, First Out tanpa memperhatikan prioritas medis.

3. Tabel Antrian Berdasarkan Prioritas

Menampilkan urutan pasien setelah proses pengurutan dengan Bubble Sort:

- Budi (Lansia → prioritas 1)
- Dewi (Ibu Hamil → prioritas 2)
- Eka (Disabilitas → prioritas 3)
- Citra dan Farhan (Umum → prioritas 4)

Urutan ini membuktikan bahwa algoritma Bubble Sort berjalan dengan benar dalam mengelompokkan pasien sesuai tingkat urgensi medis.

The screenshot shows a list of patients currently being treated. The table has columns for 'Nama' (Name), 'Umur' (Age), 'Kategori' (Category), 'Nomor' (Number), and 'Catatan' (Notes). The data is as follows:

Nama	Umur	Kategori	Nomor	Catatan
Andi	35	Darurat	1	Luka parah akibat kecelakaan motor

Gambar 2. Tampilan Pasien yang Sedang Dilayani

Tampilan ini menunjukkan bahwa sistem dapat mengenali pasien Darurat dan menanganiinya secara langsung tanpa masuk antrian. Pada contoh gambar, pasien bernama Andi (35 tahun) dengan catatan “Luka parah akibat kecelakaan motor” langsung muncul di bagian “Pasien yang Sedang Dilayani”.

Hal ini membuktikan bahwa:

- Pasien dengan prioritas 0 (Darurat) tidak dimasukkan ke dalam struktur Queue.
- Sistem otomatis menampilkan pasien tersebut di bagian layanan utama, mendahului pasien lain dalam antrian.

Selain itu, tombol “Layani Pasien” digunakan untuk memanggil pasien berikutnya dari daftar prioritas setelah pasien sebelumnya selesai dilayani. Sistem akan:

- Mengambil pasien dengan prioritas tertinggi (angka terkecil).
- Menghapus pasien tersebut dari antrian (dequeue).
- Memperbarui tampilan tabel antrian secara otomatis.

KESIMPULAN

Sistem antrian rumah sakit berbasis prioritas dengan metode Queue dan algoritma Bubble Sort mampu mengatur pasien secara efisien dan adil. Pasien darurat dilayani terlebih dahulu, sedangkan pasien lain diurutkan sesuai tingkat urgensi medis. Implementasi sistem MedLine menunjukkan peningkatan efisiensi dan kualitas pelayanan rumah sakit.

DAFTAR PUSTAKA

- Aryandi, J. A., Nugraha, M. A., Basith, Y. A. A., Pradeka, D., & Anggraini, D. (2023). Implementasi Algoritma Queue untuk Menentukan Prioritas Pelayanan Umum di Rumah Sakit. *JIKO (Jurnal Informatika dan Komputer)*, 7(2), 218–228.
- Biji, B., dkk. (2023). Implementasi Sistem Antrian Rumah Sakit Berbasis Prioritas Menggunakan Metode Queue dan Bubble Short. *Universitas Negeri Medan*.
- Kusuma Jaya, I. G. W., Manuaba, I. B. N. W., Wijaya, K. R., Wardhana, I. P. S. P., Saputra, I. M. A., & Gunadi, I. G. A. (2020). Analisis Komparasi Algoritma Sorting Antara Metode Brute Force dengan Divide and Conquer. *Jurnal Ilmu Komputer Indonesia*, 5(2), 45–52.
- Pratama, N. A., Kurnia, R., Setiawan, M. W., Jayanta, & Widiastiwi, Y. (2022). Pemanfaatan Algoritma Queue dalam Pengaturan Penjualan untuk Produk Minuman JC-JELLY CORNER. *Syntax Literate: Jurnal Ilmiah Indonesia*, 7(6).
- Universitas Bina Sarana Informatika (2018). *Modul Struktur Data*. Fakultas Teknik dan Informatika, Program Studi Sistem Informasi Akuntansi.
- Zein, A., & Tristanto, C. (2025). *Algoritma dan Struktur Data Menggunakan Pemrogramman Python*. CV. Eureka Media Aksara.