IMPLEMENTASI ALGORITMA ANTRIAN PRIORITAS MENGGUNAKAN ARRAY DI PYTHON UNTUK SISTEM ANTRIAN LAYANAN DARURAT

Hansel Valent, Rizal Muslim Sinaga, Sovantri Putra Paskah Halawa,
Selfi Audy Priscillia, Fanny Ramadhani
Fakultas Matematika Ilmu Pengetahuan Alam / Matematika, Universitas Negeri Medan, Medan
hanseldragon557@mhs.unimed.ac.id

ABSTRAK

Manajemen layanan darurat di rumah sakit sering menghadapi tantangan dalam mengelola antrean pasien secara efisien, terutama di Instalasi Gawat Darurat (IGD). Sistem antrian prioritas berbasis algoritma dan struktur data array di Python dikembangkan untuk meningkatkan efisiensi tersebut. Penelitian ini bertujuan merancang sistem berbasis array di phyton yang mampu mendistribusikan pasien sesuai tingkat urgensi menggunakan algoritma FIFO dan algoritma prioritas. Metode yang digunakan melibatkan pembuatan kelas dan antarmuka pengguna menggunakan pustaka Python Tkinter unuk membantu memasukkan pasien. Pengujian dilakukan melalui simulasi tiga skenario: pasien dengan prioritas acak, kedatangan pasien kritis berturut-turut, dan skala antrian besar. Hasil simulasi menunjukkan bahwa sistem mampu mengelola antrean dengan efisien, memberikan prioritas pada pasien kritis, dan menampilkan antrean real-time secara terstruktur. Namun, pada skenario skala besar, ditemukan jeda pemrosesan akibat tingginya beban data. Sistem ini dapat ditingkatkan dengan algoritma atau struktur data alternatif seperti heap atau tree untuk meningkatkan performa.

Kata kunci : *Antrian prioritas, Array, FIFO, Phyton*

1. PENDAHULUAN

Manajemen rumah sakit dihadapkan pada tantangan besar dalam memastikan layanan darurat yang responsif dan efisien. Dalam lingkungan di mana keselamatan pasien bergantung pada kecepatan respons, penting bagi setiap rumah sakit untuk memiliki sistem antrian yang dapat mengelola prioritas pasien secara efektif. Antrian yang efisien di pusat layanan kesehatan mampu mengurangi waktu tunggu pasien secara signifikan, sehingga meningkatkan kualitas layanan. Sistem ini menjadi semakin penting di instalasi gawat darurat (IGD), di mana kasus kritis yang membutuhkan perhatian cepat sering terjadi [1].

Tanpa adanya sistem antrian prioritas, rumah sakit menghadapi risiko besar dalam menangani pasien yang membutuhkan tindakan segera. Sendoh mengungkapkan bahwa ketidakefisienan dalam antrian dapat meningkatkan risiko insiden keselamatan pasien akibat keterlambatan dalam respons medis. Di IGD, kurangnya sistem prioritas ini juga menambah beban kerja tenaga medis, yang harus membuat keputusan cepat tentang siapa yang paling membutuhkan perawatan segera [2].

Penelitian ini bertujuan untuk mengembangkan sistem antrian prioritas sederhana berbasis array di Python, yang akan membantu IGD dalam mengelola pasien dengan lebih efisien. Sistem ini diharapkan mampu mengurangi waktu tunggu bagi pasien kritis yang membutuhkan perawatan segera. Contoh penerapan di Rumah Sakit Syafira menunjukkan bahwa sistem antrian berbasis online dapat mempermudah proses pendaftaran dan meminimalkan waktu tunggu bagi pasien [3].

Penelitian sebelumnya telah menunjukkan bahwa algoritma FIFO dan struktur data array adalah metode yang efektif dalam sistem antrian prioritas. Array juga memungkinkan pemrosesan data yang cepat dan akurat, yang sangat dibutuhkan dalam layanan darurat berbasis prioritas. Dengan memanfaatkan array, sistem ini dapat memberikan akses cepat dan penyesuaian dinamis sesuai kebutuhan layanan di IGD.

2. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Penelitian sebelumnya oleh Gunawan] menyoroti penerapan metode FIFO (First-In First-Out) dalam sistem antrian online berbasis web di UPTD Puskesmas Sananwetan. Penelitian tersebut bertujuan untuk mengatasi masalah yang kerap terjadi pada antrian tradisional, seperti ketidakteraturan dan potensi kecurangan, termasuk manipulasi urutan oleh pasien. Dengan menggunakan algoritma FIFO, sistem dirancang untuk meningkatkan keadilan dan efisiensi pelayanan, memastikan bahwa pasien dilayani berdasarkan urutan kedatangan secara konsisten tanpa gangguan yang dapat mengacaukan antrian [4].

Penelitian yang dilakukan oleh Moch. Farid Fauzi dan Alfie Nur Rahmi bertujuan untuk mengembangkan sistem antrian berbasis web di Direktorat Administrasi Akademik dan Kemahasiswaan (DAAK) Universitas AMIKOM Yogyakarta. Sistem ini dirancang dengan menggunakan metode FIFO (First In First Out) untuk mengatasi masalah antrian panjang yang sering terjadi, terutama selama periode pengisian Kartu Rencana Studi (KRS). Penelitian ini menghasilkan sistem yang memastikan mahasiswa dilayani sesuai urutan kedatangan, menciptakan alur pelayanan yang lebih terorganisir dan efisien. Hasil implementasi menunjukkan bahwa metode FIFO dapat meningkatkan kualitas layanan dengan mengurangi kekacauan dalam antrian [5].

Puput Retno Muninggar, Lilik Linawati, dan Hanna Arini Parhusip mengevaluasi dan mensimulasikan sistem antrian di Puskesmas Cebongan, Salatiga. Penelitian ini menganalisis model antrian pada beberapa bagian layanan, seperti pendaftaran, pemeriksaan tekanan darah, poliklinik umum, dan pengambilan obat. Metode FIFO (First In First Out) digunakan untuk menganalisis disiplin antrian di masing-masing bagian. Hasil simulasi menunjukkan bahwa penerapan model antrian berbasis FIFO efektif merepresentasikan kondisi nyata, meskipun terdapat sedikit perbedaan dalam rata-rata waktu tunggu di beberapa bagian. Penelitian ini memberikan wawasan tentang cara meningkatkan efisiensi pelayanan di puskesmas [6].

Penelitian lain yang dilakukan oleh Lutfia Indah Nurmalitasari dan Muhammad Fauzan dalam jurnal Jurnal Kajian dan Terapan Matematika membahas analisis sistem antrian pada teller di Bank Mandiri Cabang Sleman, Yogyakarta. Penelitian ini bertujuan untuk mengoptimalkan kinerja pelayanan dengan menggunakan model antrian multi channel single phase (M/M/c). Dengan metode observasi selama lima hari, mereka menemukan bahwa model antrian optimal adalah (M/M/3): (FIFO/∞/∞), yang menunjukkan bahwa terdapat tiga fasilitas pelayanan dengan distribusi kedatangan Poisson dan pelayanan eksponensial. Studi ini memberikan gambaran penting tentang upaya meningkatkan efisiensi pelayanan di sektor perbankan melalui pendekatan teoritis dan praktis [7].

Pada penelitian yang dilakukan oleh H. T. Shabrina [8] dalam Bulletin of Applied Industrial Engineering Theory bertujuan untuk menganalisis dan mengoptimalkan sistem antrian pada kios minuman di sebuah food court. Dengan menggunakan model antrian multi-channel single-phase (M/M/s) dan metode observasi, penelitian ini menunjukkan bahwa penggunaan dua pegawai menghasilkan waktu tunggu rata-rata pelanggan yang lebih singkat (6,6 menit) dibandingkan dengan hanya satu pegawai (75 menit). Sistem antrian berbasis FIFO ini direkomendasikan untuk mempertahankan dua pegawai guna menjaga efisiensi layanan, bahkan jika terjadi peningkatan jumlah pelanggan di masa depan [8].

2.2. Algoritma FIFO (First In First Out)

Algoritma First In First Out (FIFO) adalah metode untuk menerapkan sistem antrian tanpa prioritas. Algoritma ini memanfaatkan struktur data tertentu dan sering digunakan dalam berbagai pemecahan masalah kehidupan serta aplikasi dan teknologi. FIFO bekerja dengan prinsip berurutan dan bergiliran, di mana elemen pertama yang masuk akan diproses terlebih dahulu, mengikuti urutan masuknya tanpa ada prioritas lain [9].

FIFO juga digunakan dalam layanan kesehatan, khususnya dalam sistem antrian pasien. Dengan sistem ini, pasien yang datang lebih awal akan dilayani terlebih dahulu, menciptakan alur pelayanan yang lebih adil dan teratur. Sebagai contoh, pasien yang mendaftar lebih dulu atau yang tiba lebih awal akan mendapatkan nomor antrean sesuai urutan kedatangan. Pendekatan ini memastikan proses pelayanan berjalan transparan dan efisien, serta mengurangi potensi konflik yang bisa terjadi jika antrian tidak teratur [10].

2.3. Algoritma Prioritas

Algoritma prioritas merupakan pendekatan penting dalam pengelolaan layanan kesehatan, terutama dalam sistem yang membutuhkan alokasi sumber daya secara efisien dan responsif terhadap tingkat urgensi pasien. Algoritma ini dirancang untuk memberikan prioritas lebih tinggi kepada pasien dengan kondisi kritis atau kebutuhan mendesak, sementara pasien dengan kondisi stabil tetap dilayani dalam kerangka waktu yang memadai. Penerapan algoritma ini bertujuan untuk mengurangi waktu tunggu yang tidak perlu, meningkatkan kepuasan pasien, serta memastikan bahwa fasilitas kesehatan dapat beroperasi secara optimal meskipun menghadapi keterbatasan sumber daya. Penelitian terbaru menunjukkan bahwa algoritma prioritas telah digunakan dalam sistem antrian berbasis web di rumah sakit. Sebagai contoh, sebuah studi menggambarkan pengembangan sistem berbasis algoritma prioritas di poli umum rumah sakit, di mana pasien dapat mendaftar secara online dan memperoleh nomor antrian yang diurutkan berdasarkan tingkat urgensi mereka. Implementasi ini berhasil mengurangi waktu tunggu pasien dan meningkatkan efisiensi alur kerja tenaga medis [11].

Lebih lanjut, algoritma prioritas sering dipadukan dengan teknologi lain, seperti kecerdasan buatan (AI) dan data mining, untuk meningkatkan proses triase di instalasi gawat darurat. Sistem berbasis AI dapat menganalisis data pasien secara otomatis, seperti tanda vital dan riwayat medis, untuk menentukan tingkat prioritas dengan lebih akurat. Sebuah penelitian menunjukkan bahwa kombinasi algoritma prioritas dan AI dapat mempercepat pengambilan keputusan di ruang gawat darurat hingga 30%, yang pada gilirannya meningkatkan keselamatan pasien dalam kondisi darurat. Selain untuk triase dan antrian, algoritma prioritas juga diterapkan dalam penjadwalan operasi dan manajemen fasilitas kesehatan lainnya. Dalam studi mengenai penjadwalan multi-channel berbasis prioritas, algoritma ini digunakan untuk menentukan urutan penggunaan ruang operasi berdasarkan tingkat keparahan kasus dan ketersediaan sumber daya, sehingga operasi dapat dilakukan tepat waktu tanpa mengabaikan pasien lainnya [12].

2.4. Array

Array adalah struktur data yang sangat penting dan banyak digunakan dalam berbagai aplikasi pemrograman. Array bisa dipahami sebagai kumpulan elemen dengan tipe data yang sama, yang disusun secara berurutan di dalam memori komputer. Konsep

ini juga bisa digambarkan sebagai elemen-elemen yang tersusun dalam bentuk tabel. Setiap elemen dalam array dapat diakses menggunakan indeks tertentu, sehingga memudahkan pengguna untuk langsung merujuk dan mengambil data dari posisi yang sudah ditentukan [13].

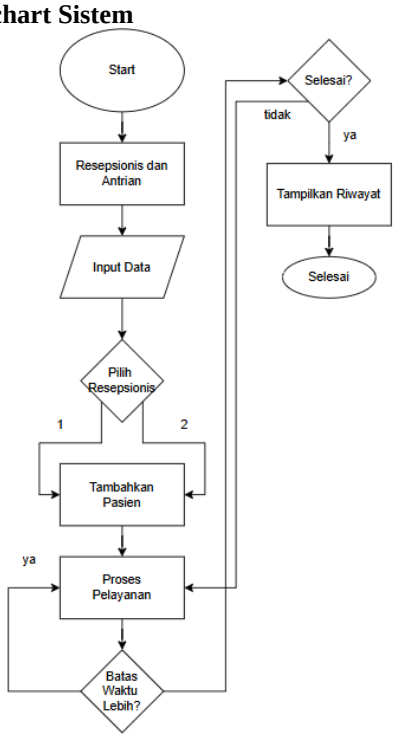
Struktur data array memiliki sifat statis terkait ukurannya, artinya jumlah elemen harus ditentukan terlebih dahulu saat deklarasi. Penggunaan array sangat bermanfaat untuk mengelola data yang terurut, seperti menyusun data pasien dalam sistem antrian rumah sakit atau mengelompokkan lagu dalam aplikasi musik. Kecepatan akses data menggunakan indeks menjadikan array pilihan yang efisien untuk operasi seperti pencarian, pengurutan, atau manipulasi data dalam jumlah besar. Dalam algoritma antrian prioritas, array bisa digunakan untuk menyimpan data pasien berdasarkan tingkat urgensi, di mana setiap elemen mewakili pasien yang diatur sesuai prioritas tertentu. Dengan demikian, array memungkinkan pemrosesan data yang cepat dan efisien, serta mendukung pengelolaan sistem antrian yang lebih teratur [14].

3. METODE PENELITIAN

3.1. Indentifikasi Masalah

Penelitian ini berfokus pada kebutuhan sistem antrian prioritas yang mampu mengelola pasien dalam situasi darurat secara efisien. Dalam layanan darurat, pasien dengan kondisi kritis membutuhkan penanganan lebih cepat dibandingkan pasien lainnya. Oleh karena itu, diperlukan sistem yang mampu mengurutkan pasien berdasarkan prioritas sehingga penanganan dapat dilakukan dengan lebih terstruktur dan efektif.

3.2. Flowchart Sistem



Gambar 1. Flowchart proses sistem

Pada gambar No. 1, program ini dimulai dengan membuat antarmuka GUI yang memungkinkan pengguna untuk memasukkan data pasien, termasuk nama, prioritas, burst time, dan pilihan resepsionis. Jika pengguna tidak menentukan resepsionis, sistem secara otomatis akan memilih berdasarkan panjang antrean. Pasien ditambahkan ke salah satu dari dua antrean berdasarkan prioritas dan beban kerja masing-masing resepsionis. Proses ini memastikan penanganan yang efisien dengan mendistribusikan pasien ke resepsionis yang antreannya lebih sedikit. Setiap resepsionis melayani pasien dalam antrean masing-masing. Kelas Server mengelola status setiap resepsionis, termasuk ketersediaan dan pasien saat ini yang dilayani. Waktu layanan untuk setiap pasien dibatasi untuk memastikan tidak ada pasien yang menghabiskan waktu resepsionis terlalu lama. Jika seorang resepsionis masih sibuk setelah waktu layanan maksimal, pasien akan di-reset dan resepsionis menjadi tersedia untuk melayani pasien berikutnya. Ini mencegah pasien tunggal memperlambat antrean secara berlebihan.

Sistem memproses semua pasien, mengelola waktu layanan dan mengalokasikan ulang pasien jika diperlukan. Setelah pemrosesan, GUI memperbarui tampilan antrean, menunjukkan status terkini dari antrean dan pasien yang menunggu. Sistem ini juga menjaga riwayat semua pasien yang dilayani, yang dapat dilihat melalui GUI. Riwayat ini mencakup waktu setiap pasien dilayani, prioritas mereka, dan resepsionis yang menangani mereka. Proses ini menutup dengan menampilkan riwayat, menandai selesainya proses layanan.

3.3. Tahapan Implementasi

a. Pastikan Python dan library tkinter telah terinstall di sistem Anda. Ini adalah langkah dasar untuk menjalankan program berbasis GUI di Python.

```
import tkinter as tk
from tkinter import ttk, messagebox
from datetime import datetime, timedelta
```

Gambar 2. Library phyton

b. Buat class Server untuk mengelola status setiap resepsionis, termasuk ketersediaan dan pasien yang sedang dilayani. Kelas ini bertanggung jawab atas logika layanan setiap resepsionis.

```
class Server:
    def __init__(self, name):
        self.name = name
        self.available_at = datetime.now()
        self.current_patient = None
        self.start_time = None

    def is_available(self):
        return datetime.now() >= self.available_at
```

Gambar 3. Class server

c. Buat class MultiplePhaseQueueSystem untuk mengelola antrean pasien, mendistribusikan pasien ke resepsionis yang tersedia, dan memproses pasien. Kelas ini menangani keseluruhan manajemen antrean.

```
class MultiplePhaseQueueSystem:
    def __init__(self):
        self.queues = {"resepsionis1": [], "resepsionis2": []}
        self.servers = {"resepsionis1": Server("Resepsionis 1"), "resepsionis2": Server("Resepsionis 2")}
        self.history = []
        self.priority_map = {"Gawat": 1, "Janji": 2, "Berobat": 3}

    def add_patient(self, name, arrival_time, burst_time, priority_str):
        priority = self.priority_map.get(priority_str, 3)
        patient = {"name": name, "arrival_time": arrival_time, "burst_time": burst_time, "priority": priority

        # Redistribusi otomatis ke resepsionis dengan antrean lebih sedikit
        target_service = "resepsionis1" if len(self.queues["resepsionis1"]) <= len(self.queues["resepsionis2"
        queue = self.queues[target_service]

        # Sisipkan pasien ke antrean berdasarkan prioritas
        index = 0
        while index < len(queue) and queue[index]["priority"] <= priority:
            index += 1
        queue.insert(index, patient)
        print(f"{name} ditambahkan ke {target_service} dengan prioritas {priority_str}.")
```

Gambar 4. Kelas MultiplePhaseQueueSystem

d. Buat kelas App untuk mengatur antarmuka GUI menggunakan tkinter. GUI ini akan berisi input untuk memasukkan data pasien dan pilihan resepsionis. Tambahkan juga frame untuk input data pasien, termasuk Entry untuk nama pasien, prioritas, burst time, dan Combobox untuk memilih resepsionis.

```
class App:
    def __init__(self, root):
        self.system = MultiplePhaseQueueSystem()
        self.root = root
        self.root.title("Sistem Antrian Resepsionis")

        # Input frame
        self.input_frame = tk.Frame(self.root)
        self.input_frame.pack(pady=10)

        tk.Label(self.input_frame, text="Nama Pasien:").grid(row=0, column=0, padx=5, pady=5)
        tk.Label(self.input_frame, text="Prioritas (Gawat, Janji, Berobat):").grid(row=1, column=0, padx=5, pady=5)
        tk.Label(self.input_frame, text="Burst Time:").grid(row=2, column=0, padx=5, pady=5)
        tk.Label(self.input_frame, text="Pilih Resepsionis:").grid(row=3, column=0, padx=5, pady=5)

        self.name_entry = tk.Entry(self.input_frame)
        self.name_entry.grid(row=0, column=1, padx=5, pady=5)

        self.priority_entry = tk.Entry(self.input_frame)
        self.priority_entry.grid(row=1, column=1, padx=5, pady=5)

        self.burst_time_entry = tk.Entry(self.input_frame)
        self.burst_time_entry.grid(row=2, column=1, padx=5, pady=5)

        self.receptionist_choice = ttk.Combobox(self.input_frame, values=["resepsionis1", "resepsionis2", "Otomatis"])
        self.receptionist_choice.grid(row=3, column=1, padx=5, pady=5)
        self.receptionist_choice.set("Otomatis")
```

Gambar 5. Class untuk GUI

e. Implementasikan method untuk menambahkan pasien ke antrean berdasarkan prioritas dan beban kerja resepsionis. Langkah ini penting untuk mendistribusikan pasien secara adil dan efisien.

```
def add_patient(self):
    name = self.name_entry.get()
    priority = self.priority_entry.get()
    burst_time = self.burst_time_entry.get()

    if not name or not priority or not burst_time.isdigit():
        messagebox.showerror("Input Error", "Semua data harus diisi dengan benar!")
        return

    priority = priority.capitalize()
    burst_time = int(burst_time)
    arrival_time = datetime.now()
    self.system.add_patient(name, arrival_time, burst_time, priority)
    self.update_queues()

def process_all_patients(self):
    self.system.process_all_patients()
    messagebox.showinfo("Proses Semua Pasien", "Semua pasien telah diproses.")
    self.update_queues()
```

Gambar 6. Code mengelola antrean

f. Buat method untuk memperbarui tampilan antrean di GUI, menampilkan pasien yang menunggu di setiap resepsionis. Ini memberikan visualisasi real-time dari status antrean. Serta Tambahkan tombol dan metode untuk menampilkan riwayat pelayanan pasien yang telah dilayani. Riwayat ini penting untuk melacak layanan yang telah diselesaikan.

```
def update_queues(self):
    self.queue_text.config(state=tk.NORMAL)
    self.queue_text.delete(1.0, tk.END)
    for service, queue in self.system.queues.items():
        self.queue_text.insert(tk.END, f"Antrean {service.capitalize()}:\n")
        for patient in queue:
            self.queue_text.insert(tk.END, f" - {patient['name']} (Priority: {patient['priority']}, "
                                           f"Burst Time: {patient['burst_time']} detik)\n")
        self.queue_text.insert(tk.END, "\n")
    self.queue_text.config(state=tk.DISABLED)

def show_history(self):
    if not self.system.history:
        messagebox.showinfo("Riwayat", "Belum ada pasien yang dilayani.")
        return

    history_text = ""
    for record in self.system.history:
        history_text += (f"{record['name']} dilayani di {record['service']} "
                         f"pada {record['service_start'].strftime('%H:%M:%S')} "
                         f"(Burst Time: {record['burst_time']} detik)\n")
    messagebox.showinfo("Riwayat Pelayanan", history_text)
```

Gambar 7. Update antrean dan riwayat

3.4. Pengujian

Simulasi dilakukan untuk menganalisis bagaimana algoritma bekerja dalam berbagai kondisi:

a. Skenario 1: Kedatangan Pasien dengan Prioritas Acak
Pasien tiba dengan tingkat prioritas yang bervariasi untuk mensimulasikan kondisi normal pada layanan darurat.

Tabel 1. Pasien Skenario 1

No	Nama Pasien	Prioritas	Burst Time
1	Hansel	2	3
2	Valent	3	4
3	Debi	3	6
4	Amanda	3	4
5	Ahmad	1	8
6	Budi	2	5

b. Skenario 2: Kondisi Kritis Berturut-turut
Pasien dengan prioritas tinggi datang berturut-turut untuk menguji kemampuan sistem dalam menangani beban kerja tinggi.

Tabel 2. Pasien Skenario 2

No	Nama Pasien	Prioritas	Burst Time
1	Ilham	3	3
2	Dika	2	4
3	Saka	1	7
4	Anisa	1	8
5	Audi	2	6
6	Reza	1	5

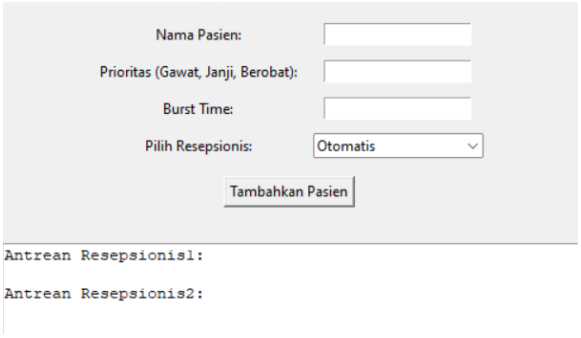
c. Skenario 3: Skala Antrian yang Lebih Besar
Antrian diuji dalam kondisi skala besar untuk menilai kinerja sistem ketika melayani jumlah pasien yang banyak.

Tabel 3. Pasien Skenario 3

No	Nama Pasien	Prioritas	Burst Time
1	Hansel	2	5
2	Budi	3	1
3	Valent	1	4
4	Dika	2	7
5	Siska	2	6
6	Ahmad	3	3
7	Sovan	3	5
8	Rizal	3	4
9	Dian	1	9
10	Clara	1	5

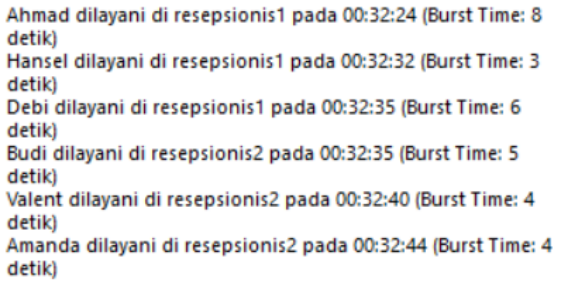
4. HASIL DAN PEMBAHASAN

Hasil pengerjaan ini menunjukkan sebuah antarmuka grafis pengguna (GUI) yang dirancang untuk sistem antrian pasien di klinik atau rumah sakit. Antarmuka ini mempermudah pencatatan dan pengelolaan pasien melalui elemen-elemen input seperti nama pasien, prioritas berdasarkan kategori tertentu seperti "Gawat", "Janji", atau "Berobat" dimana secara berturut-turut bernilai 1, 2, 3 , serta estimasi waktu layanan (Burst Time). Selain itu, terdapat menu dropdown untuk memilih resepsionis secara manual atau otomatis. Setelah semua data diisi, tombol "Tambahkan Pasien" digunakan untuk memasukkan data ke dalam sistem, dan hasilnya akan ditampilkan di bagian bawah antarmuka dengan label "Antrean Resepsionis1" dan "Antrean Resepsionis2". Sistem ini membantu mengatur antrean pasien secara lebih terorganisir dan efisien.



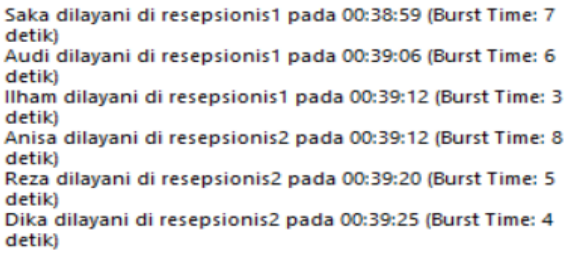
Gambar 8. Tampilan GUI

Setelah program berjalan dan menampilkan tampilan seperti diatas kita masukkan data antrian yang ada diatas . Berikut hasil dari data yang di input :



Gambar 9. Hasil skenario 1

Pada skenario pertama bisa dilihat di gambar 9, di mana pasien tiba dengan tingkat prioritas acak, sistem berhasil mengelompokkan dan mengurutkan pasien berdasarkan tingkat urgensi yang dimasukkan. Pasien dengan tingkat prioritas lebih tinggi dimana pada data nama pasien Ahmad dengan prioritas 1 mendapat perhatian lebih dahulu dibandingkan pasien dengan tingkat prioritas lebih rendah. Hasil pengelolaan ini tercermin dari antrean yang terlihat teratur di antarmuka, dengan distribusi pasien yang merata pada masing-masing resepsionis. Dengan demikian, sistem mampu mensimulasikan situasi layanan darurat sehari-hari secara efektif.



Gambar 10. Hasil skenario 2

Skenario kedua pada gambar 10 menguji kemampuan sistem dalam menangani kedatangan pasien dengan tingkat prioritas tinggi secara berturut-turut. Pada situasi ini, sistem menunjukkan efisiensi dalam merespons pasien-pasien prioritas tinggi tanpa menyebabkan penumpukan antrean pasien prioritas rendah. Dapat dilihat pada data ada 3 pasien dengan prioritas 1 dengan nama berurut Saka, Anisa, dan reza dilayani secara bergantian bisa dilihat pada gambar 10 Anisa datang duluan dan segera dilayani saat Reza datang dia akan dilayani setelah Anisa selesai. Penggunaan algoritma berbasis prioritas memungkinkan pasien kritis mendapatkan pelayanan lebih cepat, sementara pasien lainnya tetap tertangani sesuai giliran mereka.

Gambar 11. Hasil skenario 3

Skenario ketiga pada gambar 11 mensimulasikan kondisi skala besar dengan jumlah pasien yang jauh lebih banyak. Sistem ini tetap mampu mengelola distribusi pasien ke dalam dua antrean resepsionis dengan baik, meskipun terdapat variasi besar dalam prioritas dan waktu layanan setiap pasien dimana terdapat jeda waktu beberapa detik dalam memperoses data. Tampilan real-time dari status antrean melalui GUI memberikan gambaran jelas mengenai posisi setiap pasien dalam antrean, sehingga memudahkan tenaga kesehatan dalam melacak proses layanan.

Hasil penelitian ini menunjukkan bahwa sistem antrian berbasis GUI yang dikembangkan untuk manajemen pasien di fasilitas kesehatan mampu mengelola antrean menggunakan array secara terstruktur dan efisien. Sistem ini dirancang untuk memungkinkan pengguna memasukkan data pasien seperti nama, tingkat prioritas, dan estimasi waktu layanan, serta secara otomatis mendistribusikan pasien ke resepsionis yang memiliki beban kerja lebih ringan.

Hasil simulasi dalam tiga skenario menunjukkan bahwa sistem ini memberikan performa yang baik dalam situasi yang berbeda-beda.

Secara keseluruhan, sistem ini memanfaatkan algoritma prioritas yang diimplementasikan dalam struktur data array untuk memastikan pemrosesan data yang cepat dan akurat. Dengan menampilkan informasi antrean secara real-time, antarmuka sistem ini memberikan transparansi dan kemudahan penggunaan yang meningkatkan efisiensi layanan. Selain itu, riwayat layanan pasien yang disimpan oleh sistem menjadi alat evaluasi penting bagi rumah sakit untuk terus meningkatkan kualitas pelayanan di masa mendatang.

5. KESIMPULAN DAN SARAN

Dari hasil penelitian ini, dapat disimpulkan bahwa penerapan sistem antrian prioritas berbasis array di Python, meskipun secara keseluruhan berhasil mengelola antrian secara efisien, masih menghadapi kendala teknis pada skenario 3. Dalam skenario ini, yang mensimulasikan skala antrian besar dengan banyak pasien dan beragam tingkat prioritas, ditemukan jeda beberapa detik saat sistem memproses dan menampilkan hasil pada antarmuka. Kemungkinan besar, jeda tersebut terjadi akibat beban pemrosesan data yang meningkat seiring bertambahnya jumlah pasien dalam antrean, mengingat struktur data array yang bersifat statis dapat mengalami keterbatasan performa ketika menangani volume data yang besar. Meskipun demikian, sistem tetap mampu beroperasi dengan stabil dan menghasilkan keluaran yang sesuai dengan prioritas yang ditentukan, sehingga tujuan penelitian tetap tercapai. Untuk pengembangan di masa depan, diperlukan optimasi pada algoritma dan pengelolaan struktur data, seperti penggunaan struktur data alternatif yang lebih fleksibel, seperti heap atau tree, untuk mengurangi waktu pemrosesan pada skenario dengan beban besar.

DAFTAR PUSTAKA

[1] Sendoh, A., Pertiwi, J. M., & Manoppo, J. I. C. (2023). Analisis Penerapan Sasaran Keselamatan Pasien di Instalasi Gawat Darurat Rumah Sakit X Provinsi Sulawesi Utara. Medical Scope Journal, 5(1), 50-56.

[2] Wijoyo, A., Prasetiyo, A. R., Salsabila, A. A., Nife, K., & Nadapdap, P. B. (2024). Evaluasi efisiensi struktur data linked list pada implementasi sistem antrian. JRIIN: Jurnal Riset Informatika dan Inovasi, 1(12), 1244–1246.

[3] Melyanti, R., Irfan, D., Ambiyar, A., Febriani, A., & Khairana, R. (2020). Rancang bangun sistem antrian online kunjungan pasien rawat jalan pada rumah sakit syafira berbasis web. INTECOMS: Journal of Information Technology and Computer Science, 3(2), 192-198.

[4] Gunawan, R., Yuana, H., & Kirom, S. (2023). Implementasi Metode Queue Pada Sistem Antrian Online Berbasis Web Studi Kasus Uptd Puskesmas Sananwetan. JATI (Jurnal Mahasiswa Teknik Informatika), 7(3), 1538-1545.

[5] Fauzi, M. F., & Rahmi, A. N. (2021). PENERAPAN METODE FIRST IN FIRST OUT (FIFO) DALAM SISTEM ANTRIAN PELAYANAN ADMINISTRASI MAHASISWA: Studi Kasus: DAAK Universitas AMIKOM Yogyakarta. METHOMIKA: Jurnal Manajemen Informatika & Komputerisasi Akuntansi, 5(2), 183-188.

[6] Muninggar, P. R., Linawati, L., & Parhusip, H. A. (2019). Analisis Sistem Antrian dengan Simulasi di Puskesmas Cebongan Kota Salatiga. Jurnal Fourier, 8(2), 57-64.

[7] Nurmalitasari, L. I., & Fauzan, M. (2022). Analisis sistem antrian model multi channel single phase pada teller bank mandiri Sleman. Jurnal Kajian dan Terapan Matematika, 8(2), 129-137.

[8] Shabrina, H. T., Putra, R. M., Safi'i, A., Hidayat, N., Ikbal, M., & Syauqi, M. (2022). Analisis Sistem Antrian Guna Mengoptimalisasikan Pelayanan Pada Kios Minuman (Food Court). Bulletin of Applied Industrial Engineering Theory, 3(1).

[9] Fitriani, F., & Apridiansyah, Y. (2021). Aplikasi Antrian Pembayaran Uang Kuliah Berbasis Android Menggunakan Algoritma Fifo Di Universitas Muhammadiyah Bengkulu. JUSIBI (Jurnal Sistem Informasi dan Bisnis), 3(2), 91-103.

[10] Hehanussaa, S. G., Irawadia, E., & Astutia, W. (2022). Penerapan Metode RAD dan Algoritma Fifo pada Aplikasi Antrian Pasien Puskesmas. Buletin Sistem Informasi dan Teknologi Islam ISSN, 2721, 0901.

[11] Aryandi, J. A., Nugraha, M. A., Basith, Y. A. A., Pratama, M. F., 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.

[12] Widiarto, W., Maheswari, D., Sari, D. P., & Arianto, K. J. (2024). Implementasi Algoritma Round Robin dan Priority Pada Sistem Antrian Rumah Sakit. JURNAL FASILKOM, 14(2), 507-513.

[13] Sihombing, J. (2019). Penerapan stack dan queue pada array dan linked list dalam java. INFOKOM (Informatika & Komputer), 7(2), 15-24.

[14] Putri, G. M., Di Pradja, K. A., Azizi, M. B. M., Nurwahid, P., & Perdana, A. S. (2024). Implementasi Stack dan Array pada Pengurutan Lagu dengan Metode Selection Sort. Jurnal Teknologi Dan Sistem Informasi Bisnis, 6(2), 286-296.