

MedQ - Aplikasi Simulasi Sistem Antrean Klinik Berbasis Web

Disusun Untuk Memenuhi Tugas Mata Kuliah
Cloud Computing

Dosen Pengampu :
Sam Farisa CH, ST, M.Kom.



Disusun Oleh
Alief Dzulfikar (32602300087)

FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM SULTAN AGUNG SEMARANG
TAHUN PELAJARAN 2024/2025

DAFTAR ISI

DAFTAR ISI.....	1
DAFTAR GAMBAR.....	1
BAB I PENDAHULUAN.....	1
1.1 TUJUAN	1
BAB II PEMBAHASAN.....	2
1.2 Fitur Aplikasi	2
2.1 Setup Docker.....	2
BAB III PENUTUP.....	3
Kesimpulan	3

DAFTAR GAMBAR

Gambar 1. 1 Pengisian Jadwal	2
Gambar 1. 2 Hasil Simulasi	2
Gambar 1. 3 Riwayat Simulasi	2
Gambar 1. 4 GitHub.....	2

BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Perkembangan teknologi cloud computing memungkinkan pengembangan aplikasi yang ringan, fleksibel, dan mudah di-deploy. Di bidang layanan kesehatan, sistem

antrean sangat penting untuk meningkatkan efisiensi pelayanan pasien.

Oleh karena itu dibuatlah aplikasi MedQ, yaitu simulasi antrean klinik berbasis web menggunakan Flask (Python) yang dapat dijalankan dengan mudah melalui Docker.

1.1 TUJUAN

1. Membuat aplikasi web sederhana untuk simulasi antrean klinik.
2. Menerapkan konsep containerization menggunakan Docker.
3. Memberikan contoh implementasi cloud computing sederhana yang dapat di-deploy dimana saja.

BAB II PEMBAHASAN

1.2 Fitur Aplikasi

1. Input jumlah pasien, durasi pemeriksaan, jam mulai, dan jumlah layanan (dokter/poli).

The screenshot shows the 'Pengisian Jadwal' (Schedule Filling) form. It includes input fields for 'Poli Poli Umum', 'Poli Poli Gigi', and 'Poli Poli Anak'. There are also fields for 'Durasi Pemeriksaan (menit)' and 'Jam Mulai Klinik (default 08:00)'. A 'Jumlah Layanan (Dokter/Poli)' field is at the bottom. A blue 'Simulasi' button is on the right.

Gambar 1. 1 Pengisian Jadwal

2. Simulasi antrean dengan jadwal mulai dan selesai otomatis.

The screenshot shows the 'Hasil Simulasi' (Simulation Results) table. It has columns: No, Mulai, Selesai, Status, and Layanan. The table lists 12 simulation runs with their respective start and end times and the service provided.

No	Mulai	Selesai	Status	Layanan
1	08:00	08:10	Diperiksa	Poli Umum - Dokter A
2	08:00	08:10	Diperiksa	Poli Umum - Dokter B
6	08:00	08:10	Diperiksa	Poli Gigi - Dokter A
7	08:00	08:10	Diperiksa	Poli Gigi - Dokter B
13	08:00	08:10	Diperiksa	Poli Anak - Dokter A
14	08:00	08:10	Diperiksa	Poli Anak - Dokter B
3	08:10	08:20	Dalam Antrean	Poli Umum - Dokter A
4	08:10	08:20	Dalam Antrean	Poli Umum - Dokter B
8	08:10	08:20	Dalam Antrean	Poli Gigi - Dokter A
9	08:10	08:20	Dalam Antrean	Poli Gigi - Dokter B
15	08:10	08:20	Dalam Antrean	Poli Anak - Dokter A
5	08:20	08:30	Dalam Antrean	Poli Umum - Dokter A
10	08:20	08:30	Dalam Antrean	Poli Gigi - Dokter A
11	08:20	08:30	Dalam Antrean	Poli Gigi - Dokter B
12	08:30	08:40	Dalam Antrean	Poli Gigi - Dokter A

Gambar 1. 2 Hasil Simulasi

3. Dukungan multi layanan (Poli A, Poli B, dst).
4. Tombol Reset, Cetak Jadwal, dan Riwayat Simulasi.

The screenshot shows the 'Riwayat Simulasi' (Simulation History) table. It has columns: #, Jumlah Pasien, Jumlah Poli, Durasi / Pasien, and Jenis Poli. The table lists 9 simulation runs with their respective patient counts, number of doctors, duration, and service type.

#	Jumlah Pasien	Jumlah Poli	Durasi / Pasien	Jenis Poli
1	10	2	10 menit	Poli Umum
2	5	2	10 menit	Poli Gigi
3	12	2	10 menit	Poli Anak
4	5	2	10 menit	Poli Umum
5	7	2	10 menit	Poli Gigi
6	3	2	10 menit	Poli Anak
7	10	1	10 menit	Poli Umum
8	5	1	10 menit	Poli Gigi
9	12	1	10 menit	Poli Anak

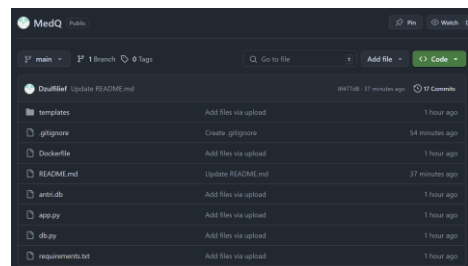
Gambar 1. 3 Riwayat Simulasi

5. Tampilan menggunakan HTML + Bootstrap 5.

2.1 Setup Docker

1. Install Docker Dekstop
2. Buka folder MedQ pada terminal
`cd MedQ`
3. Build Docker Image
`docker build -t medq .`
4. Jalankan Container
`docker run -p 5000:5000 medq`
5. Akses di browser
<http://localhost:5000>

Seluruh file diunggah melalui GitHub dengan link :
<https://github.com/Dzulfilief/MedQ>



Gambar 1. 4 GitHub

Repositori ini berisi seluruh source code aplikasi MedQ, file Dockerfile, requirements.txt, serta template HTML untuk tugas Cloud Computing.

BAB III

PENUTUP

3.1 Kesimpulan

Aplikasi **MedQ** berhasil dibuat sebagai simulasi antrean klinik berbasis web menggunakan Flask (Python) dan dijalankan dengan Docker.

Dengan fitur input jumlah pasien, durasi pemeriksaan, jam mulai, serta dukungan multi layanan, aplikasi ini mampu menghasilkan jadwal antrean otomatis yang dapat di-reset, ditambah pasien baru, maupun dicetak untuk dokumentasi.

Melalui proses containerization menggunakan Docker, aplikasi dapat dijalankan dengan mudah di berbagai environment tanpa perlu konfigurasi ulang.

Proyek ini sekaligus menjadi contoh penerapan sederhana cloud computing yang mendukung pengembangan dan distribusi aplikasi secara fleksibel.