

Sistemi Informasi Pengingat Obat: Implementasi dengan Flask dan SQLite

Dibuat oleh : Bintang Bimantara Putra(3260230019)

Abstrak

Makalah ini mengampu perancangan dan implementasi sistem informasi pengingat obat berbasis web menggunakan kerangka kerja Flask dan database SQLite. Sistem ini memungkinkan pengguna untuk menambahkan, melihat, mengedit, dan menghapus data obat-obatan, termasuk nama, dosis, waktu, dan frekuensi. Tujuannya adalah untuk membantu pengguna dalam mengelola jadwal konsumsi obat secara efektif.

Bab 1 Pendahuluan

1.1 Latar Belakang

Kepatuhan terhadap jadwal pengobatan adalah faktor krusial dalam efektivitas terapi medis. Banyak individu, terutama mereka yang memiliki jadwal padat atau harus mengonsumsi banyak jenis obat, kesulitan untuk mengingat dosis dan waktu minum obat yang tepat.

1.2 Tujuan

Tujuan dari proyek ini adalah untuk mengembangkan sistem informasi pengingat obat yang:

- Memungkinkan pengguna untuk mencatat detail obat (nama, dosis, waktu, frekuensi).
- Menyediakan antarmuka yang intuitif untuk menambah, mengubah, dan menghapus data obat..
- Dibangun menggunakan teknologi web yang ringan dan mudah diimplementasikan.

Bab 2 Tinjauan Pustaka

2.1 Flask

Flask adalah kerangka kerja aplikasi web mikro untuk Python. Disebut "mikro" karena Flask tidak membutuhkan alat atau pustaka tertentu. Ini menyediakan dasar-dasar yang diperlukan untuk membangun aplikasi web, memberikan fleksibilitas kepada pengembang untuk memilih alat dan pustaka lain sesuai kebutuhan.

2.2 SQLite

SQLite adalah sistem manajemen basis data relasional (RDBMS) yang mandiri, nir-server, nir-konfigurasi, dan nir-transaksional. SQLite adalah mesin basis data paling banyak digunakan di

dunia. Basis data SQLite disimpan dalam satu berkas di sistem file, menjadikannya sangat portabel dan cocok untuk aplikasi kecil hingga menengah, serta sebagai basis data lokal.

2.3 Sistem Pengingat Obat

Sistem pengingat obat adalah aplikasi atau perangkat yang dirancang untuk membantu individu mengingat kapan harus minum obat mereka. Sistem ini dapat berupa aplikasi seluler, perangkat keras khusus, atau aplikasi web. Fitur umum meliputi penjadwalan dosis, notifikasi, dan pencatatan riwayat konsumsi obat.

Bab 3 Perancangan Sistem

3.1 Arsitektur Sistem

Sistem ini mengadopsi arsitektur client-server sederhana. Bagian *backend* dibangun dengan Flask yang berinteraksi dengan database SQLite. Bagian *frontend* adalah antarmuka web yang diakses melalui *browser*.

3.2 Desain Database

Database terdiri dari satu tabel utama bernama `medications`. Struktur tabel ini dirancang untuk menyimpan semua informasi penting mengenai setiap obat yang perlu diingat.

Tabel 3.1 Struktur Tabel `medications`

Kolom	Tipe Data	Keterangan
<code>id</code>	INTEGER PRIMARY KEY AUTOINCREMENT	ID unik untuk setiap entri obat
<code>name</code>	TEXT NOT NULL	Nama obat
<code>dosage</code>	TEXT NOT NULL	Dosis obat (contoh: "1 tablet", "5 ml")
<code>time</code>	TEXT NOT NULL	Waktu konsumsi obat (contoh: "08:00", "20:00")
<code>frequency</code>	TEXT NOT NULL	Frekuensi konsumsi obat (contoh: "Setiap hari", "Dua kali sehari")

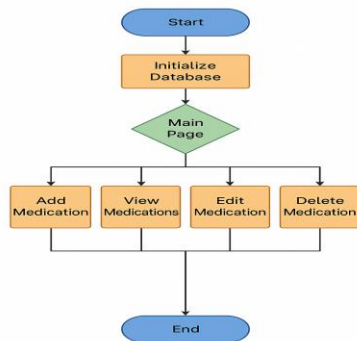
Export to Sheets.

3.3 Alur Kerja Aplikasi

1. **Inisialisasi Database:** Saat aplikasi pertama kali dijalankan, fungsi `init_db()` akan memeriksa keberadaan file database `database.db`. Jika tidak ada, database dan tabel `medications` akan dibuat.
2. **Halaman Utama (Index):** Ketika pengguna mengakses halaman utama (`/`), aplikasi akan mengambil semua data obat dari database, mengurutkannya berdasarkan waktu, dan menampilkannya di halaman `index.html`.

3. **Tambah Obat:** Pengguna dapat mengisi formulir di halaman utama untuk menambahkan obat baru. Data yang dimasukkan (nama, dosis, waktu, frekuensi) akan disimpan ke database melalui rute `/add`.
4. **Edit Obat:** Setiap entri obat memiliki opsi edit. Ketika pengguna mengklik edit, mereka akan diarahkan ke halaman `/edit/<id>` yang menampilkan detail obat yang dipilih. Pengguna dapat memperbarui informasi dan menyimpannya kembali ke database.
5. **Hapus Obat:** Pengguna dapat menghapus entri obat melalui rute `/delete/<id>`. Data obat yang sesuai akan dihapus dari database.

Berikut flowchart alur kerja aplikasi:



Gambar 3.3 Flowchart Alur Kerja Aplikasi

Bab 4 Implementasi Sistem

4.1 Persyaratan Sistem

Untuk menjalankan aplikasi ini, lingkungan pengembangan harus memiliki:

- Python 3.x
- Flask
- SQLite (biasanya sudah termasuk dalam Python)
- Docker dan Docker Compose (opsional, untuk deployment)

4.2 Struktur Direktori Proyek

```
reminder-obat/
├── app.py
├── templates/
│   ├── index.html
│   └── edit.html
├── requirements.txt
├── Dockerfile
├── docker-compose.yml
├── database.db
└── static/
```

└─ alarm.mp3

4.3 Penjelasan Kode

1. **reminder-obat/Dockerfile:** Berkas ini mendefinisikan cara membangun *image* Docker aplikasi. Ini memulai dari *image* Python 3.9, mengatur direktori kerja menjadi `/app`, menyalin kode aplikasi, menginstal dependensi dari `requirements.txt`, mengekspos *port* 5000, dan menentukan perintah awal kontainer (`init_db.py` diikuti `app.py`).
2. **reminder-obat/app.py:** Ini adalah inti aplikasi Flask.
 - Mengimpor modul Flask, SQLite, dan OS.
 - Mengatur `DB_NAME` untuk menunjuk ke `database.db`.
 - `get_db_connection()`: Membuat koneksi ke SQLite dengan `row_factory` untuk akses data mudah.
 - `init_db()`: Fungsi untuk membuat tabel `medications` jika belum ada (tidak dipanggil langsung dari `main` aplikasi saat menggunakan Docker).
 - `@app.route('/')`: Menampilkan semua data obat dari database di `index.html`.
 - `@app.route('/add')`: Menangani penambahan data obat dari formulir ke database.
 - `@app.route('/edit/<int:id>')`: Menampilkan formulir edit dan memperbarui data obat di database.
 - `@app.route('/delete/<int:id>')`: Menghapus data obat dari database.
 - `app.run()`: Menjalankan aplikasi Flask di *debug mode* pada *port* 5000.
3. **reminder-obat/templates/index.html:** Berkas HTML ini adalah halaman utama yang menampilkan daftar obat yang ada dan formulir untuk menambahkan obat baru. Juga berisi tautan untuk mengedit dan menghapus obat.
4. **reminder-obat/templates/edit.html:** Berkas HTML ini adalah halaman untuk mengedit detail obat yang dipilih, mengisi formulir dengan data obat yang sudah ada.
5. **reminder-obat/requirements.txt:** Berkas ini mencantumkan semua pustaka Python yang dibutuhkan oleh proyek, yaitu `flask`.
6. **reminder-obat/docker-compose.yml:** Berkas konfigurasi Docker Compose ini mendefinisikan layanan `reminder-obat`. Ini memerintahkan Docker untuk membangun *image* dari `Dockerfile`, memetakan *port* 5000 *host* ke *port* 5000 kontainer, me-mount direktori proyek ke `/app` di kontainer untuk pengembangan, dan menjalankan `init_db.py` kemudian `app.py`.
7. **reminder-obat/init_db.py:** Skrip Python terpisah yang berfungsi untuk membuat file database `database.db` dan skema tabel `medications` di dalamnya jika belum ada.
8. **reminder-obat/database.db:** File database SQLite ini akan dibuat secara otomatis di direktori utama `reminder-obat/` saat `init_db.py` pertama kali dijalankan oleh Docker Compose, dan akan menyimpan semua data obat.
9. **reminder-obat/static/alarm.mp3 (opsional):** Direktori `static/` adalah tempat untuk menyimpan aset statis seperti file audio, yang dapat digunakan untuk fitur alarm di masa mendatang.

Struktur ini memungkinkan aplikasi berjalan dalam kontainer Docker yang terisolasi, memudahkan pengembangan dan *deployment* yang konsisten.

4.4 Cara Menjalankan Aplikasi

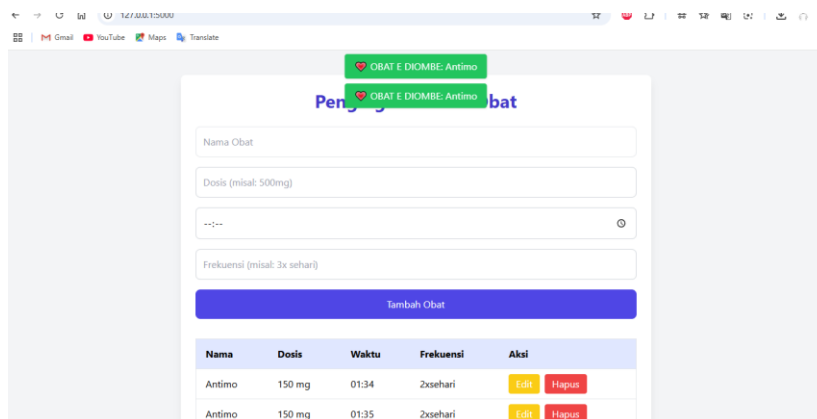
Menggunakan Python Langsung:

1. Pastikan Python dan Flask terinstal.
2. Instal dependensi: `pip install -r requirements.txt`
3. Jalankan skrip inisialisasi database: `python init_db.py`
4. Jalankan aplikasi Flask: `python app.py`
5. Akses aplikasi di *browser* Anda: `http://127.0.0.1:5000/`

Menggunakan Docker Compose:

1. Pastikan Docker dan Docker Compose terinstal.
2. Dari direktori proyek, jalankan: `docker-compose up --build`
3. Akses aplikasi di *browser* Anda: <http://localhost:5000/5>

Akses Di github : <https://github.com/PoppaGwopPPa/reminder-obat.git>



Gambar 4.5 aplikasi pengingat obat

Bab 5 Kesimpulan

5.1 Kesimpulan

Sistem informasi pengingat obat berbasis web ini berhasil diimplementasikan menggunakan Flask dan SQLite, menyediakan fungsionalitas dasar untuk pengelolaan jadwal obat. Sistem ini ringan, mudah digunakan, dan portabel, menjadikannya solusi yang baik untuk kebutuhan pengingat obat pribadi.