

Manajemen Kesehatan Pasien Menggunakan Struktur Data Vektor, Array, Hash Table, dan Hashing

Patient Health Management Using Vector, Array, Hash Table and Hashing Data Structures

AHMAD QAULAN SADIDA^{1*}, FARI HAFIZH NUGROHO², SULTHAN FARRAS RAZIN³, NURUL FADILLAH⁴

Abstrak

Manajemen kesehatan pasien yang efektif seringkali membutuhkan waktu dan upaya yang signifikan. Program kode untuk manajemen kesehatan pasien dapat menjadi solusi yang inovatif dan efisien dalam membantu penyedia layanan kesehatan mengelola perawatan pasien dengan mudah. Penelitian ini berfokus pada implementasi struktur data vektor, array, peta, tabel hash dan hashing. Sistem yang dikembangkan memberikan informasi kondisi kesehatan pasien dan mengelola data medis dengan baik. Akses terhadap data yang terorganisir membantu penyedia layanan kesehatan dalam merencanakan pemberian obat secara efektif dan membuat kemungkinan gejala-gejala yang muncul atau efek samping yang disebabkan oleh obat tersebut..

Kata Kunci: Kesehatan, Manajemen, Pasien, Vektor, Array, Hash Table

Abstract

Effective patient health management often requires significant time and effort. A coded program for patient health management can be an innovative and efficient solution in helping healthcare providers manage patient care with ease. This research focuses on the implementation of vector data structures, arrays, maps, hash tables and hashing. The developed system provides information on the patient's health condition and manages medical data properly. Access to organized data helps health care providers in planning the administration of drugs effectively and making possible the symptoms that appear or side effects caused by the drugs.

Keywords: Health, Management, Patient, Vector, Array, Hash Table

PENDAHULUAN

Manajemen kesehatan merupakan segala kegiatan dalam mengatur, baik para petugas kesehatan maupun non petugas kesehatan guna meningkatkan kesehatan masyarakat melalui program kesehatan. Penggunaan sumber daya yang efektif dan efisien serta ditunjang dengan penggunaan fungsi-fungsi manajemen yang baik akan mempermudah dalam mewujudkan pelayanan kesehatan masyarakat yang optimal (Saputra *et al.* 2023). Manajemen data yang sangat banyak dapat menjadi suatu tantangan. Maka dari itu, manajemen akan jauh lebih mudah jika disandingkan dengan penggunaan teknologi karena dapat mengurangi taraf *human error*. Selama ini, kasus yang sering terjadi dalam sebuah instansi kesehatan adalah sering terjadinya penggandaan nomor pasien rekam medis serta dokter salah diagnosa dengan dokter yang memeriksa pasien selama proses sistem rekam medis berlangsung (Fitri dan Putri 2022).

Manajemen kesehatan menggunakan bantuan teknologi bukanlah hal yang baru di Indonesia. Beberapa sistem manajemen kesehatan pasien sudah dilakukan oleh banyak peneliti-peneliti lain sebelumnya, salah satunya adalah yang dilakukan oleh Fitri dan Putri di tahun 2022. Sistem yang mereka bangun dilatarbelakangi oleh fakta bahwa selama ini pencatatan data rekap rekam medis pada bidan praktik mandiri masih dilakukan secara manual di mana sistem tersebut mengandung banyak kekurangan seperti memerlukan waktu yang cukup lama untuk mencari dan mendapatkan data pasien, rentan terjadi kesalahan dalam penulisan, dan berisiko untuk kehilangan data-data pasien. Sistem yang dibentuk berbasis *website* yang diperuntukkan untuk instansi bidan praktik. Tahapan-tahapan penelitian yang

¹Ahmad Qaulan Sadida, Bogor, 081293396343;

²Fari Hafizh Nugroho, Bogor, 081238552998;

³Sulthan Farras Razin, Bogor, 088214522941;

⁴Nurul Fadillah, Bogor, 085161236525;

*Penulis Korespondensi: Tel/Faks: 081293396343; Surel: ahqa24sadida@apps.ipb.ac.id

dilakukan, yaitu pengumpulan data dengan observasi dan mengidentifikasi masalah-masalah pada bidang praktik mandiri. Sistem dibangun dengan menggunakan bahasa pemrograman aplikasi PHP dan juga menggunakan database MySQL.

Mini project ini bertujuan untuk membuat sistem manajemen kesehatan pasien yang menggunakan struktur data untuk mencatat riwayat penyakit, jadwal perawatan, dan informasi kesehatan lainnya. Tujuannya adalah memberikan solusi efisien untuk melacak dan mengelola informasi kesehatan pasien secara terintegrasi.

METODE

Ruang lingkup penelitian ini dibatasi hanya pada implementasi struktur data vector, array, map, hash table, dan hashing pada program yang bertujuan membantu pengguna mencapai tujuannya.

- VECTOR

Array 1 dimensi atau yang sering disebut dengan vektor adalah suatu array yang value atau nilainya hanya ditunjukkan oleh satu indeks. Array 1 dimensi juga dideklarasikan dengan variabel yang sudah dibentuk sebelumnya dan tidak dapat diubah selama program dijalankan (Putri *et al.* 2022). Struktur data ini dipilih karena cara kerjanya yang mampu menunjang fungsi untuk pencatatan riwayat obat yang telah diberikan dan efeknya kepada pasien, fungsi untuk manajemen obat dan resep, serta fungsi untuk manajemen jadwal perawatan dan pengingat

Struktur data vektor dapat digunakan untuk mencatat riwayat obat yang telah diberikan dan efeknya kepada pasien dengan cara yang efisien dan terorganisir. Salah satu implementasi struktur data vektor dalam produk Manajemen Kesehatan Pasien yang dikembangkan adalah untuk mencatat nama-nama dokter yang terdaftar dalam sistem. Selain itu, struktur data vektor sesuai untuk diimplementasikan ke dalam fungsi manajemen obat dan resep. Dalam produk Manajemen Kesehatan Pasien yang dikembangkan, struktur data vektor dalam fungsi manajemen obat dan resep berperan dalam menyimpan data daftar penyakit. Lebih lanjut, struktur data vektor sangat efektif untuk diterapkan dalam fungsi manajemen jadwal perawatan dan pengingat. Dalam produk Manajemen Kesehatan Pasien, struktur data vektor dimanfaatkan untuk menyimpan tugas-tugas yang dapat dilakukan dalam manajemen jadwal, seperti menambahkan jadwal perawatan, mengubah jadwal perawatan, menghapus jadwal perawatan, dan melihat jadwal perawatan.

- ARRAY

Array adalah struktur penyimpanan data yang paling umum digunakan itu dibangun ke dalam sebagian besar bahasa pemrograman. Array merupakan suatu variabel yang khas dan memiliki kemampuan dalam memberi tempat yang dapat memberi tampungan nilai lebih dari satu dalam waktu bersamaan (Zein dan Eriana 2022).

Pada produk Manajemen Kesehatan Pasien yang dikembangkan, struktur data array diimplementasikan dalam fungsi pencatatan riwayat penyakit dan informasi medis. Struktur data ini digunakan untuk menambahkan dan merekam riwayat penyakit dari suatu pasien ketika dia menjalani sebuah pemeriksaan, sehingga ketika pasien tersebut ingin berkunjung kembali ke rumah sakit maka seluruh riwayat penyakit yang sebelumnya akan muncul di dalam database.

- HASH TABLE

Hash table merupakan struktur data yang menyerupai array untuk penyimpanan data dalam bentuk keys and value. Tujuan dari hash table adalah untuk mempercepat pencarian kembali dari banyak data yang disimpan (Ramdhani *et al.* 2022).

Struktur data hash table memfasilitasi manajemen obat dan resep secara efisien dan sistematis. Salah satu implementasi struktur data hash table dalam produk Manajemen

Kesehatan Pasien yang dikembangkan ada dalam fungsi pencatatan riwayat obat yang telah diberikan dan efeknya kepada pasien. Dalam produk Manajemen Kesehatan Pasien yang dikembangkan, struktur data hash table dalam fungsi pencatatan riwayat obat yang telah diberikan dan efeknya kepada pasien berperan dalam membuat *database* yang berisi nama-nama obat beserta resep dan efek sampingnya. Di samping itu, struktur data hash table cocok untuk diimplementasikan dalam fungsi manajemen obat dan resep, yakni dengan cara memanggil informasi obat yang tersimpan dalam *database*.

- **HASHING**

Algoritma hashing merupakan algoritma yang mengubah teks atau pesan menjadi rangkaian karakter acak yang memiliki karakter yang sama. Algoritma hashing mirip dengan algoritma pencarian, namun selain mencari data, hashing juga mencocokkan permintaan pencarian dengan kunci ID yang telah ditentukan sebelumnya. Fungsi hashing dapat digunakan untuk menyimpan kata sandi, integritas pesan, dan sidik jari pesan (Daulay dan Yahfizham 2023).

Pada produk Manajemen Kesehatan Pasien yang dikembangkan, struktur data hashing diimplementasikan dalam fungsi keamanan data dan akses terbatas. Struktur data ini digunakan untuk memeriksa validitas login dengan mengambil username dan password sebagai input. Jika username dan password yang diinput sesuai dengan yang ada di *database*, maka sistem akan mengembalikan nilai true yang menandakan login berhasil dan berlaku sebaliknya. Keterkaitannya dengan struktur data hashing adalah penggunaan teknik hashing untuk mempercepat pencarian data akun, sehingga proses verifikasi login dapat dilakukan menjadi lebih efisien.

HASIL DAN PEMBAHASAN

Proyek struktur data Manajemen Kesehatan Pasien melibatkan pengembangan sistem yang rumit dengan berbagai aspek penting yang memerlukan pemikiran kompleks. Salah satu aspek utama dalam proyek ini adalah manajemen data, dimana berbagai jenis data seperti pencatatan riwayat penyakit dan informasi medis, manajemen jadwal perawatan dan pengingat, manajemen obat dan resep serta efek samping, kemudian yang terakhir ada fitur keamanan dan akses terbatas untuk memastikan data pasien aman dan terlindungi. Penggunaan struktur data yang tepat sangat penting untuk pengelolaan data yang efektif. Selain itu, algoritma yang digunakan untuk memproses dan menganalisis data harus diperhatikan dengan baik. Algoritma yang efisien akan memastikan performa yang optimal dan waktu respon yang cepat dalam mengelola informasi pasien. Misalnya, untuk mencari obat dan resep yang sesuai dengan penyakit pasien, diperlukan algoritma pencarian yang cepat dan akurat agar menghasilkan informasi yang relevan. Mengetahui efek samping yang dihasilkan dari suatu obat merupakan hal yang sangat penting bagi pasien. Dengan sistem yang terintegrasi antar pencarian obat, resep, serta efek samping akan sangat memudahkan pasien dalam menjalani pengobatan dan proses menuju kesembuhan.

Sistem Manajemen Kesehatan Pasien yang dikembangkan memiliki kelebihan dan kekurangan yang perlu diperhatikan dalam evaluasi keseluruhan. Kelebihan yang pertama adalah sistem ini bisa menjalankan seluruh fungsi yang diminta dengan baik. Semua fitur yang dibutuhkan untuk operasional telah diimplementasikan dan berjalan dengan lancar. Selain itu, sistem ini dilengkapi dengan fitur untuk *register* dan *login* untuk memastikan bahwa hanya pengguna yang terotorisasi yang dapat mengakses data dan fungsi-fungsi yang ada. Fitur ini memberikan lapisan keamanan tambahan sehingga kerahasiaan dan integritas data pasien dapat terjaga. Kelebihan lain dari sistem ini adalah data pengguna yang tersimpan melalui *login* dan *register* dapat disimpan ke dalam file teks *txt*. Metode penyimpanan ini memungkinkan akses yang cepat dan efisien tanpa perlu konfigurasi yang kompleks sehingga sangat cocok untuk sistem dengan skala kecil menengah.

Di samping kelebihan-kelebihan yang dimiliki oleh sistem ini, terkandung juga beberapa kekurangan. Salah satu kekurangan dari sistem ini adalah desain interface yang terlalu sederhana sehingga kurang menarik untuk dilihat. Kekurangan lainnya adalah keterbatasan dalam hal kapasitas penyimpanan *database* di mana sistem tidak memungkinkan untuk menyimpan terlalu banyak data pasien. Informasi mengenai obat-obatan yang disimpan dalam sistem ini juga terbatas karena *database* untuk menyimpan informasi mengenai obat-obatan di-*hardcode* bersamaan dengan fungsi lain-lainnya. Karena sistem ini tidak menggunakan *database* yang terpisah, sistem menjadi kurang fleksibel.

Struktur data yang digunakan dalam pengembangan sistem ini adalah vector, array, hash table, dan hashing. Selain struktur data-struktur data tersebut, masih terdapat struktur data lain yang relevan dengan struktur data yang digunakan dalam pengembangan sistem ini. Struktur data lain yang relevan dengan struktur data vektor adalah struktur data *deque*. Tidak seperti vektor, struktur data ini mendukung operasi penambahan dan penghapusan elemen di kedua ujung. Struktur data yang relevan dengan struktur data array adalah struktur data linked list. Struktur data linked list lebih fleksibel dalam hal ukuran, insert, dan delete tetapi akses datanya lebih lambat dibandingkan jika menggunakan struktur data array. Struktur data yang relevan dengan struktur data hash table adalah struktur data binary search tree (BST). Tidak seperti struktur data hash table, struktur data binary search tree lebih cocok digunakan ketika urutan data penting sementara hash table tidak mementingkan urutan data. Struktur data lain yang relevan dengan struktur data hashing adalah hash set yang hanya menyimpan kunci tanpa nilai terkait. Sementara hashing memberikan fondasi untuk berbagai struktur data yang efisien, hash set menyediakan solusi sederhana dan efisien untuk kasus penggunaan yang memerlukan penyimpanan kumpulan elemen unik dengan akses cepat.

Metode yang digunakan pada penelitian ini melibatkan implementasi terhadap beberapa struktur data seperti vector, array, hash table, dan hashing dalam suatu program yang bertujuan untuk membantu pengguna mencapai tujuannya. Vektor satu array satu dimensi dipilih karena kemampuannya dalam mencatat riwayat obat dan efeknya pada pasien, manajemen obat dan resep, serta manajemen jadwal perawatan dan pengingat. Implementasi struktur data vektor dalam produk manajemen kesehatan pasien meliputi pencatatan nama-nama dokter, daftar penyakit, serta tugas tugas-tugas dalam manajemen jadwal perawatan. Array, yang merupakan struktur data paling umum dan fundamental, digunakan untuk mencatat riwayat penyakit dan informasi medis pasien, memastikan bahwa seluruh riwayat penyakit sebelumnya dapat diakses ketika pasien berkunjung ke rumah sakit kembali.

Hash table adalah struktur data yang sangat efisien untuk operasi insertion dan searching, kemudian diimplementasikan dalam fungsi pencatatan riwayat obat, manajemen obat, dan resep. Dalam produk manajemen kesehatan pasien, hash table digunakan untuk membuat database obat, resep, beserta efek sampingnya, memudahkan pemanggilan informasi obat dari database. Algoritma hashing mengubah teks menjadi rangkaian karakter acak dan dapat digunakan untuk memeriksa validitas login, memastikan keamanan data dan akses terbatas. Teknik hashing mempercepat pencarian data akun, sehingga proses verifikasi menjadi lebih efisien. Dengan menggunakan struktur data dan algoritma yang tepat, proyek ini bertujuan untuk meningkatkan efisiensi dan efektivitas manajemen kesehatan pasien, mengurangi kesalahan manusia, dan mempercepat respon sistem dalam menangani informasi kesehatan pasien.

Sistem manajemen kesehatan pasien yang dikembangkan terdiri atas lima fungsi utama, yakni fungsi untuk pencatatan riwayat penyakit dan informasi medis, fungsi untuk manajemen jadwal perawatan dan pengingat, fungsi untuk manajemen obat dan resep, fungsi untuk keamanan data dan akses terbatas, serta fungsi untuk pencatatan riwayat obat yang telah diberikan dan efeknya kepada pasien.

1. Fungsi untuk Pencatatan Riwayat Penyakit dan Informasi Medis

Banyaknya konsep struktur data yang diimplementasikan dalam fungsi ini adalah sebanyak satu struktur data, yakni struktur data array.

Banyaknya operasi-operasi struktur data yang digunakan dalam fungsi ini ada sebanyak tiga buah, yang terdiri dari operasi insert sebanyak dua buah dalam baris program “void tambahRiwayatPenyakit(const string& penyakit) {riwayatPenyakit.push_back(penyakit);”, yakni untuk menambah riwayat penyakit pasien dan baris program “savetofile:void saveToFile() {ofstream file("biodata.txt", ios::app);”, yakni untuk menyimpan dan menambahkan biodata pasien baru ke dalam file txt, operasi search sebanyak satu kali dalam baris program “loadbiodatafromfilePasien loadBiodataFromFile(const string& nama_pasien) { ifstream file("biodata.txt");, yakni untuk mencari biodata pasien dari sebuah file yang bernama biodata.txt.

Fungsi `tambahRiwayatPenyakit` menambahkan sebuah string yang mewakili penyakit ke dalam vector `riwayatPenyakit` menggunakan metode `push_back()`. Kompleksitas waktu dari operasi ini adalah $O(1)$ dalam kebanyakan kasus, karena penambahan elemen di akhir vector biasanya memerlukan waktu yang konstan. Namun, dalam kasus dimana vector perlu dialokasikan ulang (ketika kapasitas vector overflow), kompleksitasnya bisa menjadi $O(n)$, di mana n adalah jumlah elemen dalam vector sebelum pengalokasian ulang. Efisiensi fungsi ini tinggi untuk sebagian besar kasus karena operasi `push_back` biasanya sangat cepat, meskipun efisiensi bisa sedikit menurun ketika pengalokasian ulang terjadi.

Fungsi `saveToFile` membuka file bernama "biodata.txt" dalam mode tambahan (`ios::app`), yang dapat memungkinkan data baru untuk ditambahkan ke akhir file tanpa menghapus konten yang ada. Kompleksitas waktu untuk membuka file dan menulis data ke dalamnya tergantung pada ukuran file dan jumlah data yang ditambahkan, tetapi operasi ini biasanya dianggap memiliki kompleksitas $O(1)$ untuk operasi menulis sederhana. Efisiensi fungsi ini sangat bergantung pada performa I/O dari sistem file, namun secara umum, membuka dan menulis ke file dalam mode append (Penambahan data baru ke file lama) adalah operasi yang efisien karena tidak memerlukan pengolahan ulang seluruh file.

Fungsi `loadBiodataFromFile` membuka file "biodata.txt" untuk membaca data. Dalam fungsi ini, kompleksitas pencarian data pasien dalam file bergantung pada bagaimana data disimpan dan diakses. Jika data disimpan secara berurutan dan pencarian dilakukan secara linear, kompleksitasnya adalah $O(n)$, di mana n adalah jumlah baris dalam file. Kasus terburuknya adalah ketika seluruh file harus dibaca untuk menemukan data pasien yang dicari. Efisiensi fungsi ini dapat rendah jika file sangat besar dan data pasien berada di akhir file, karena setiap pencarian memerlukan pemrosesan seluruh file hingga data ditemukan atau file habis dibaca. Namun, jika file diindeks atau dicari dengan cara yang lebih efisien, efisiensi bisa meningkat.

Interface dari fungsi untuk pencatatan riwayat penyakit dapat dilihat pada Gambar 1 dan fungsi untuk pencatatan informasi medis dapat dilihat pada Gambar 2.

```
== MAIN MENU ==
Menu Pilihan :
1. Biodata
2. Riwayat Penyakit
3. Manajemen Obat
4. Riwayat Obat
5. Pencatatan Jadwal
6. Lihat Data Pasien
7. Simpan & Keluar
Pilih menu [1/2/3/4/5/6/7] : 2
Masukkan nama pasien: Fara Attira
Masukkan riwayat penyakit (ketik 'selesai' untuk mengakhiri):
batuk
pilek
selesai
```

Gambar 1 *Interface* fungsi pencatatan riwayat penyakit

```

== MAIN MENU ==
Menu Pilihan :
1. Biodata
2. Riwayat Penyakit
3. Manajemen Obat
4. Riwayat Obat
5. Pencatatan Jadwal
6. Lihat Data Pasien
7. Simpan & Keluar
Pilih menu [1/2/3/4/5/6/7] : 1
Masukkan nama pasien: Fara Attira
Masukkan nomor identitas pasien: Masukkan usia pasien: 541
Masukkan alamat pasien: Bogor
Masukkan kontak pasien: 0812

```

Gambar 2 *Interface* fungsi untuk pencatatan informasi medis

2. Fungsi untuk Manajemen Jadwal Perawatan dan Peningat

Banyaknya konsep struktur data yang diimplementasikan dalam fungsi ini adalah sebanyak satu struktur data, yakni struktur data vektor.

Banyaknya operasi-operasi struktur data yang digunakan dalam fungsi ini ada sebanyak 7 buah, yang terdiri dari operasi insert sebanyak 2 kali dalam baris `"vector<string> doctors = {"Dr. Tirta Saputra", "Dr. Boyke Prayoga", "Dr. Noorman Kamaru"};"`, yakni untuk menambah nama dokter ke dalam database dan baris program `"vector<string> availableSchedules = {"10:00", "12:00", "14:00", "16:00"};"`, yakni untuk menambah jam praktek dari dokter yang bersangkutan, operasi search sebanyak satu kali dalam baris program `"Pasien pasien = loadBiodataFromFile(nama_pasien);"`, yakni untuk memuat data pasien dari sebuah file, operasi time operations sebanyak dua kali dalam baris program `"due_tm.tm_mon += 1; // Tambah satu bulan"`, yakni untuk menambah satu bulan pada tanggal target dan kedua dan baris program `"time_t due_date_limit = mktime(&due_tm);"`, yakni untuk mengkonversi tanggal menjadi waktu dalam bentuk `time_t`. kemudian operasi string manipulation dalam baris program `"cout << "Masukkan tanggal (YYYY-MM-DD): ";"` yakni untuk meminta pengguna memasukkan tanggal dalam format tertentu dan baris program `"strftime(buffer, 80, "Batas Waktu: %d-%m-%Y %I:%M:%S", timeinfo);"`, yakni untuk mengatur format output yang menampilkan batas waktu.

Deklarasi dan inisialisasi dua vektor, `'doctors'` dan `'availableSchedules'`, melibatkan operasi insert pada saat vektor tersebut diinisialisasi dengan beberapa elemen. Kompleksitas waktu dari inisialisasi ini adalah $O(n)$, di mana n adalah jumlah elemen yang dimasukkan ke dalam vector karena setiap elemen harus ditempatkan dalam memori. Efisiensi dari operasi ini cukup tinggi karena inisialisasi dilakukan satu kali dan langsung dengan nilai-nilai yang diberikan, tanpa perlu alokasi dinamis tambahan setelah inisialisasi.

Operasi search dilakukan dengan memuat data pasien dari file menggunakan fungsi `'loadBiodataFromFile'`. Kompleksitas waktu dari operasi ini adalah $O(n)$, di mana n adalah jumlah baris dalam file, karena setiap baris mungkin harus dibaca hingga data yang diinginkan ditemukan. Efisiensi operasi ini bisa bervariasi tergantung pada ukuran file dan posisi data yang dicari; jika file besar dan data pasien berada di akhir, maka efisiensi akan menurun karena seluruh file harus dibaca.

Operasi waktu yang melibatkan penambahan satu bulan pada objek `'due_tm'` dan konversi struktur waktu ke `'time_t'` menggunakan `'mktime'` biasanya memiliki kompleksitas waktu $O(1)$, karena hanya melibatkan operasi aritmatika sederhana dan konversi langsung. Efisiensi operasi ini sangat tinggi karena hanya melibatkan manipulasi data dalam memori yang cepat dan tidak bergantung pada ukuran dataset.

Operasi manipulasi string termasuk menampilkan pesan dengan `'cout'` dan format tanggal dengan `'strftime'`. Kompleksitas waktu untuk kedua operasi ini

adalah $O(1)$, karena mereka hanya bekerja pada data yang sudah ada dalam memori tanpa memerlukan pemrosesan berulang. Efisiensi kedua operasi ini juga tinggi karena operasi string sederhana biasanya cepat dan tidak memerlukan banyak sumber daya komputasi.

Interface dari fungsi untuk manajemen jadwal perawatan dan pengingat dapat dilihat pada Gambar 3.

```
== MAIN MENU ==
Menu Pilihan :
1. Biodata
2. Riwayat Penyakit
3. Manajemen Obat
4. Riwayat Obat
5. Pencatatan Jadwal
6. Lihat Data Pasien
7. Simpan & Keluar
Pilih menu [1/2/3/4/5/6/7] : 5

PENYIMPANAN JADWAL PERAWATAN DAN PENGINGAT
=====

Menu Utama:
1. Tambah Jadwal Perawatan
2. Edit Jadwal Perawatan
3. Hapus Jadwal Perawatan
4. Lihat Jadwal Perawatan
5. Keluar
```

Gambar 3 *Interface* fungsi untuk manajemen jadwal perawatan dan pengingat

3. Fungsi untuk Manajemen Obat dan Resep

Banyaknya konsep struktur data yang diimplementasikan dalam fungsi ini adalah sebanyak dua struktur data, yakni struktur data vektor dan hash table.

Banyaknya operasi-operasi struktur data yang digunakan dalam fungsi ini ada sebanyak 5 buah, yang terdiri dari operasi insert sebanyak 3 kali dalam baris program “DatabaseObat[\"Dermatitis\"] = \"Clobetasol, resep : Oleskan tipis-tipis pada area yang terkena dermatitis, Efek samping: iritasi kulit, kemerahan\";”, yakni untuk memasang antara key berupa obat dengan value berupa obat, resep, dan efek samping dan baris program “unordered_map<string, string> DatabaseObat;”, yakni untuk membuat wadah dalam menyimpan data, dan baris program “DaftarPenyakit.push_back(pair.first);”, yakni untuk menyimpan nama penyakit, operasi search sebanyak satu kali dalam baris program “return DatabaseObat.at(disease);”, yakni untuk mengakses key berdasarkan value yang ada pada database, operasi traversal vector sebanyak satu buah “for (int i = 0; i < DaftarPenyakit.size(); ++i) { cout << i+1 << “. ” << DaftarPenyakit[i] << endl;”, untuk membuat nomor urut lebih user friendly (dimulai dari 0 bukan 1).

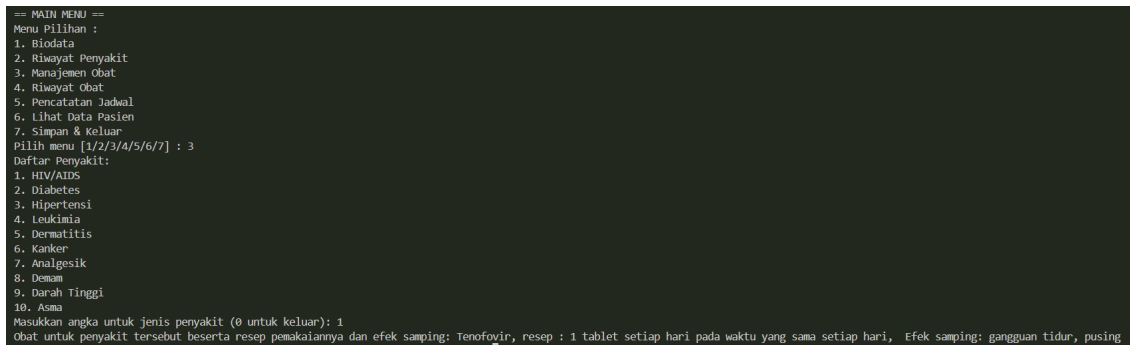
Operasi Insert melibatkan penambahan entri baru ke dalam ‘DatabaseObat’, yang merupakan sebuah ‘unordered_map’, dan menambahkan penyakit ke dalam ‘DaftarPenyakit’, yang merupakan sebuah ‘vector’. Kompleksitas waktu untuk menambahkan entri ke dalam ‘unordered_map’ adalah $O(1)$ dalam kasus rata-rata karena akses langsung ke lokasi penyimpanan. Namun, dalam skenario terburuk, kompleksitas bisa meningkat menjadi $O(n)$ karena adanya kemungkinan bentrokan hash. Penambahan elemen ke dalam ‘vector’ memiliki kompleksitas waktu $O(1)$ dalam rata-rata dan $O(n)$ dalam skenario terburuk ketika perlu dilakukan alokasi ulang memori. Secara keseluruhan, efisiensi operasi ini umumnya tinggi karena kedua struktur data mendukung penyisipan cepat dalam kasus rata-rata.

Operasi search menggunakan fungsi ‘at’ dari ‘unordered_map’ untuk mengakses data yang diinginkan berdasarkan kunci. Kompleksitas waktu dari operasi ini adalah $O(1)$ dalam kasus rata-rata karena akses langsung ke data melalui hashing. Namun, dalam skenario terburuk di mana terjadi banyak bentrokan hash, kompleksitas bisa menjadi $O(n)$. Secara umum, efisiensi operasi ini sangat baik karena

`unordered_map` dirancang untuk operasi pencarian yang cepat dalam kasus rata-rata.

Operasi Traversal Vector melibatkan iterasi melalui semua elemen dalam `DaftarPenyakit`, yang merupakan sebuah `vector`. Kompleksitas waktu dari operasi ini adalah $O(n)$, di mana n adalah jumlah elemen dalam `vector`, karena setiap elemen harus diakses satu per satu. Efisiensi operasi ini tergolong baik untuk traversal sederhana, namun kinerja dapat terpengaruh jika jumlah elemen sangat besar. Struktur data `vector` memberikan kinerja optimal untuk iterasi karena elemen-elemennya disimpan secara berurutan di dalam memori.

Interface dari fungsi untuk manajemen obat dan resep dapat dilihat pada Gambar 4.



```
== MAIN MENU ==
Menu Pilihan :
1. Biodata
2. Riwayat Penyakit
3. Manajemen Obat
4. Riwayat Obat
5. Pencatatan Jadwal
6. Lihat Data Pasien
7. Simpan & Keluar
Pilih menu [1/2/3/4/5/6/7] : 3
Daftar Penyakit:
1. HIV/AIDS
2. Diabetes
3. Hipertensi
4. Leukimia
5. Dermatitis
6. Kanker
7. Analgesik
8. Demam
9. Darah Tinggi
10. Asma
Masukkan angka untuk jenis penyakit (0 untuk keluar): 1
Obat untuk penyakit tersebut beserta resep pemakaiannya dan efek samping: Tenofovir, resep : 1 tablet setiap hari pada waktu yang sama setiap hari, Efek samping: gangguan tidur, pusing
```

Gambar 4 *Interface* fungsi untuk manajemen obat dan resep

4. Fungsi untuk Keamanan Data dan Akses Terbatas

Banyaknya konsep struktur data yang diimplementasikan dalam fungsi ini adalah sebanyak satu struktur data, yakni struktur data hashing.

Banyaknya operasi-operasi struktur data yang digunakan dalam fungsi ini ada sebanyak tiga buah, yang terdiri dari operasi insert sebanyak dua buah pada hash table dalam baris program `dataAkun[index].username = username;` dan `dataAkun[index].password = password;` untuk menyimpan user name dan password data akun dari data admin dan pada cek daftar dalam baris program `ofstream outFile("dataPasien.txt", ios::app);`, `outFile << username << " " << password << endl;`, dan `outFile.close();` yang berfungsi untuk menyimpan username dan password ke dalam file `dataAdmin.txt`. Kompleksitas dari operasi insertion struktur data hashing adalah $O(1)$.

Operasi delete ada sebanyak satu buah pada hash table cekhapusakun yang terdapat dalam baris program `dataAkun[index].username = "";` dan `dataAkun[index].password = "";` yang masing-masing digunakan untuk menghapus data akun. Kompleksitas dari operasi deletion struktur data hashing adalah $O(1)$.

Operasi search ada sebanyak dua buah pada cek login dan muatdataakundari file yang terdapat dalam baris program `if (dataAkun[index].username == username && dataAkun[index].password == password) {return true;}` dan `ifstream inFile("dataPasien.txt"); while (inFile >> fileUsername >> filePassword) {if (fileUsername == username && filePassword == password) {inFile.close();return true;}}` yang digunakan untuk mencari username dan password dalam file `dataPasien.txt`. Kompleksitas dari operasi searching struktur data hashing adalah $O(1)$.

Interface dari fungsi keamanan data dan akses terbatas dapat dilihat pada Gambar 5.

```
== SELAMAT DATANG! ==
Menu Pilihan :
1. Daftar
2. Login
3. Keluar
Pilih menu [1/2/3] : 1

== MENU DAFTAR ==
Masukkan username & password
Isi Username anda : adid
Isi Password anda : adid2403
Akun berhasil terdaftar!!

== SELAMAT DATANG! ==
Menu Pilihan :
1. Daftar
2. Login
3. Keluar
Pilih menu [1/2/3] : 2

== LOGIN ==
Masukkan username & password
Isi username anda : adid
Isi password anda : adid2403

login berhasil!!!
```

Gambar 5 *Interface* dari fungsi keamanan data dan akses terbatas

5. Fungsi untuk Pencatatan Riwayat Obat yang telah Diberikan dan Efeknya kepada Pasien

Banyaknya konsep struktur data yang diimplementasikan dalam fungsi ini adalah sebanyak dua struktur data, yakni struktur data vektor dan hash table.

Banyaknya operasi-operasi struktur data yang digunakan dalam fungsi ini ada sebanyak tiga buah, yang terdiri dari operasi insert sebanyak satu buah dalam baris program “`pasien.tambahRiwayatObat(medicine);`”, yakni untuk menambah riwayat obat pasien. Kompleksitas dari operasi insertion di akhir struktur data vektor adalah $O(1)$.

Operasi search ada sebanyak satu buah dalam baris program “`manager.getMedicineByNumber(pilihan);`”, yakni untuk mencari obat berdasarkan nomor kode obat. Kompleksitas dari operasi searching struktur data hash table adalah $O(1)$.

Operasi input/output validation ada sebanyak dua buah dalam baris program “`if (cin.fail()) {}`” dan “`if (opsiLanjut == 'a' || opsiLanjut == 'b') {}`”, yakni masing-masing untuk memastikan bahwa nomor nama penyakit yang di-*input* terdaftar dan untuk memutuskan apakah user akan melakukan pencarian lebih lanjut.

Interface dari fungsi untuk pencatatan riwayat obat yang telah diberikan dan efeknya kepada pasien dapat dilihat pada Gambar 6.

```
== MAIN MENU ==
Menu Pilihan :
1. Biodata
2. Riwayat Penyakit
3. Manajemen Obat
4. Riwayat Obat
5. Pencatatan Jadwal
6. Lihat Data Pasien
7. Simpan & Keluar
Pilih menu [1/2/3/4/5/6/7] : 4
Masukkan jumlah pasien: 1
Masukkan nama pasien yang ingin dimuat: Fara
Masukkan riwayat obat untuk Fara
Daftar Penyakit:
1. HTV/AIDS
2. Diabetes
3. Hipertensi
4. Leukimia
5. Dermatitis
6. Kanker
7. Analgesik
8. Demam
9. Darah Tinggi
10. Asma
Masukkan angka untuk jenis penyakit (0 untuk keluar): 2
Apakah ada riwayat obat lain untuk Fara (y/n)? n
Riwayat obat untuk pasien Fara:
Metformin
```

Gambar 6 *Interface* fungsi untuk pencatatan riwayat obat yang telah diberikan dan efeknya kepada pasien

SIMPULAN

Manajemen Kesehatan Pasien memberikan kemudahan kepada suatu instansi kesehatan dalam mengelola dan mengatur database pasien, merekam riwayat penyakit, manajemen obat dan resep serta efek samping, termasuk fitur keamanan dan akses terbatas. Semuanya terintegrasi menjadi sistem yang padu untuk pengelolaan data yang lebih efektif, mengurangi adanya human error, dan pengoptimalan waktu dan respon yang cepat dalam mengatur informasi pasien. Struktur data yang digunakan agar program secara keseluruhan menjadi lebih sederhana yaitu Vektor, Array, Hash Table, dan Hashing.

DAFTAR PUSTAKA

- Daulay PI, Yahfizham. 2023. Penerapan algoritma pemrograman dalam pembelajaran ilmu komputer. *Jurnal Arjuna: Publikasi Ilmu Pendidikan, Bahasa dan Matematika*. 1(6):101. <https://doi.org/10.61132/arjuna.v1i6.297>.
- Fitri DA, Putri A. 2022. Rancang bangun sistem rekam medis berbasis website. *J Computer Science and Information Technology (CoSciTech)*. 3(2):208. <https://doi.org/10.37859/coscitech.v3i2.3753>.
- ChatGPT. (n.d.). <https://openai.com/chatgpt/>.
- Putri MP, Barovich G, Azdy RA, Yuniansyah, Saputra A, Sriyeni Y, Rini A, Admojo FT. 2022. *Algoritma dan Struktur Data*. Bandung: Widina.
- Ramdhani R, Fadlil A, Sunardi. 2022. Penerapan algoritma Winnowing dan Word-Level Trigrams untuk mengidentifikasi kesamaan kata. *Jurnal Riset Komputer*. 9(2):430. 10.30865/jurikom.v9i2.4060.
- Saputra MKF, Rizqulloh L, Pati DU, Kusumawati D, Widiyastuti NE, Sihombing ESR, Putra HA, Wirdhani WA, Cahya MRF, Sidabutar S, et al.. 2023. *Manajemen Kesehatan*. Serang: Sada Kurnia Pustaka.
- Zein A, Eriana ES. 2022. *Algoritma dan Struktur Data*. Tangerang: Unpam Press.