

Penerapan Konsep TreeMap pada Software Sistem Administrasi Informasi Pasien Rawat Inap (SAPIR)

Raihan Sultan Pasha Basuki
Ilmu Komputer 2022 | Computer Science
ESQ Business School

Kampus Menara 165, Jl. Tb. Simatupang Kav. 1, Cilandak Timur, Pasar Minggu, RT.3/RW.3, Cilandak Tim.,
Ps. Minggu, Kota Jakarta Selatan, Daerah Khusus Ibukota Jakarta 12560, Indonesia
raihansultan.pashabasuki@students.esqbs.ac.id

Abstrak:

Dalam struktur data, khususnya Bahasa pemrograman Java, terdapat banyak konsep untuk pengelolaan administrasi data, terutama pada Bahasa pemrograman Java yang memiliki banyak library maupun method untuk membangun struktur data tersebut. Pada aplikasi SAPIR, beberapa konsep algoritma dan struktur data digunakan untuk memaksimalkan kinerja administrasi data pasien rawat inap rumah sakit, seperti TreeMap.

Kata Kunci: Algoritma, Struktur Data, TreeMap, Tree, Administrasi, Data Pasien.

1 Pendahuluan

1.1 Latar Belakang

Sistem administrasi informasi pasien rawat inap (SAPIR) merupakan sistem yang digunakan untuk mengelola informasi pasien yang dirawat inap di rumah sakit. Sistem ini memiliki fungsi yang penting, yaitu untuk menyimpan dan mengelola data pasien, seperti data identitas, data medis, dan data perawatan. Salah satu aspek penting dalam pengembangan SAPIR adalah penerapan ilmu algoritma dan struktur data. Algoritma dan struktur data merupakan fondasi dasar dalam ilmu komputer yang berperan penting dalam merancang dan mengimplementasikan sistem informasi yang efisien dan efektif. Pada umumnya, SAPIR menggunakan struktur data tree untuk menyimpan data pasien, di dalam sebuah objek berisi detail data pribadi. Struktur data tree memiliki kelebihan dalam hal efisiensi penyimpanan data, tetapi memiliki kekurangan dalam hal fleksibilitas. Ketidakefleksibilitas struktur data tabel dapat menjadi masalah, terutama jika sistem SAPIR perlu melakukan operasi pencarian, penyisipan, atau penghapusan data secara dinamis. Untuk mengatasi masalah tersebut, dapat diterapkan konsep linkedlist dan BST pada SAPIR. Linkedlist dan BST merupakan struktur data yang lebih fleksibel dibandingkan tabel. Linkedlist memiliki kelebihan dalam hal efisiensi pencarian data, sedangkan BST memiliki kelebihan dalam hal efisiensi penyisipan dan penghapusan data. Penerapan konsep linkedlist dan BST pada SAPIR dapat meningkatkan efisiensi dan efektivitas sistem. Sistem SAPIR akan menjadi lebih cepat dan responsif dalam melakukan operasi pencarian, penyisipan, dan penghapusan data.

1.2 Deskripsi

Proyek yang diangkat dalam problematika ini, adalah sebuah perangkat lunak berbasis Bahasa pemrograman Java bernama SAPIR (Sistem Administrasi Pasien Rawat Inap). Perangkat lunak ini mengusung konsep administrasi data pasien yang sedang di rawat inap di dalam sebuah instalasi medis seperti rumah sakit, yang pada umumnya memiliki banyak Kasur rawat dan tersebar di banyak kamar, koridor, dan bangsal rumah sakit. SAPIR hadir dengan menerapkan konsep Algoritma dan Struktur Data, lebih spesifiknya adalah konsep TreeMap yang digunakan untuk menyimpan maupun mengoperasikan seperti input dan output data pasien yang akan dan/atau sedang dirawat.

1.3 Penjelasan

Dengan adanya fasilitas yang diberikan oleh Perangkat Lunak SAPIR ini, diharapkan pengurusan administrasi yang dilakukan di rumah sakit akan menjadi lebih praktis dan efektif.

1.4 Tujuan

Memaksimalkan administrasi pasien rumah sakit, terutama pasien rawat inap yang harus diolah datanya. Dengan begitu, efektivitas pada pengolahan data administrasi pasien rawat inap di rumah sakit, dapat dilakukan dengan lebih cepat, serta lebih efisien.

2 Dasar Teori Struktur Data yang Diaplikasikan

TreeMap adalah struktur data terurut yang menyimpan data dalam bentuk pohon biner. TreeMap menggunakan prinsip pengurutan alami untuk menyimpan data, sehingga data dapat diakses dengan cepat dan efisien.

2.1 Struktur TreeMap

TreeMap terdiri dari dua komponen utama, yaitu:

a. Node: Node adalah elemen dasar dari TreeMap. Setiap node memiliki dua elemen, yaitu:

b. Key: Key adalah kunci yang digunakan untuk mengurutkan data.

Value: Value adalah nilai yang disimpan pada node.

Root: Root adalah node tertinggi dalam TreeMap. Root digunakan untuk mengakses data dalam TreeMap.

2.2 Prinsip Penyimpanan Data

Data dalam TreeMap disimpan dalam urutan alami. Urutan alami ini ditentukan berdasarkan tipe data key. Misalnya, jika key adalah tipe data integer, maka urutan alaminya adalah urutan angka.

2.3 Operasi TreeMap

TreeMap menyediakan berbagai operasi untuk mengakses dan memanipulasi data, yaitu:

Insert: Menambahkan data baru ke TreeMap.

Search: Mencari data dalam TreeMap.

Remove: Menghapus data dari TreeMap.

First: Mengembalikan data pertama dalam TreeMap.

Last: Mengembalikan data terakhir dalam TreeMap.

Ceiling: Mengembalikan data tertinggi yang lebih kecil atau sama dengan key yang diberikan.

Floor: Mengembalikan data terendah yang lebih besar atau sama dengan key yang diberikan.

2.4 Keuntungan TreeMap

TreeMap memiliki beberapa keuntungan, yaitu:

Data terurut: Data dalam TreeMap selalu terurut, sehingga data dapat diakses dengan cepat dan efisien.

Efisiensi: TreeMap relatif efisien dalam hal ruang dan waktu.

Fleksibilitas: TreeMap dapat digunakan untuk menyimpan data dengan tipe data yang berbeda.

2.5 Kekurangan TreeMap

TreeMap juga memiliki beberapa kekurangan, yaitu:

Kompleksitas: TreeMap memiliki kompleksitas yang relatif tinggi dalam hal operasi pencarian dan penghapusan.

Membutuhkan ruang lebih: TreeMap membutuhkan ruang lebih untuk menyimpan data dibandingkan struktur data lain, seperti array atau linked list.

3. Implementasi Struktur Data

Dalam program SAPIR, implementasi konsep struktur data yang digunakan adalah konsep TreeMap. Dalam program tersebut, TreeMap digunakan untuk menyimpan data pasien. Berikut adalah tempat-tempat di mana TreeMap digunakan:

3.1 Deklarasi

```
public static void main(String[] args) {  
    TreeMap<Integer, DataPasien> dataPasienMap = new TreeMap<>();
```

Pada bagian ini, TreeMap dideklarasikan untuk menyimpan data pasien. Kunci (key) dari TreeMap adalah nomor pasien (NoPasien), dan nilai (value) adalah objek DataPasien.

3.2 Tambah Data Pasien Baru

```
DataPasien newPasien = new DataPasien(newNamaPasien, newNoPasien, newDiagnosa, newLokasiKasur);  
dataPasienMap.put(newNoPasien, newPasien);
```

Setiap kali pengguna memilih opsi untuk menambahkan data pasien baru, objek DataPasien dibuat dan disimpan dalam TreeMap dengan nomor pasien sebagai kunci.

3.3 Cari Data Pasien Menggunakan NoPasien (ID)

```
System.out.print(s:"Masukkan NoPasien: ");  
int searchNoPasien = scanner.nextInt();  
DataPasien foundPasien = dataPasienMap.get(searchNoPasien);
```

Pada bagian ini, nomor pasien yang dimasukkan oleh pengguna digunakan sebagai kunci untuk melakukan pencarian dalam TreeMap.

3.4 Print Seluruh Data Pasien

```
System.out.println(x:"Seluruh Data Pasien:");  
for (DataPasien pasien : dataPasienMap.values()) {  
    printDataPasien(pasien);
```

Pada bagian ini, TreeMap digunakan untuk mendapatkan semua nilai (values) yang berupa objek DataPasien. Kemudian, setiap objek DataPasien dicetak.

3.5 Print Data Pasien dan Seluruh Data yang ada di atasnya

```
for (Map.Entry<Integer, DataPasien> entry : dataPasienMap.entrySet()) {  
    if (entry.getKey() >= noPasien) {  
        printDataPasien(entry.getValue());  
        dataFound = true;
```

Pada bagian ini, TreeMap digunakan untuk mendapatkan setiap pasangan kunci-nilai (entry) yang memenuhi kondisi tertentu (nomor pasien di atas nilai tertentu).

3.6 Hapus Data Pasien

```
System.out.print(s:"Masukkan NoPasien untuk menghapus: ");  
int deleteNoPasien = scanner.nextInt();  
DataPasien deletedPasien = dataPasienMap.remove(deleteNoPasien);
```

Pada bagian ini, nomor pasien yang dimasukkan oleh pengguna digunakan sebagai kunci untuk menghapus data pasien dari TreeMap.

4. Fungsionalitas Aplikasi

Sistem Administrasi Pasien Rawat Inap memiliki sejumlah fungsionalitas kunci yang mendukung pengelolaan data pasien secara efektif. Pertama, aplikasi memungkinkan pengguna untuk mencari data pasien berdasarkan nomor pasien atau nama pasien, memudahkan akses cepat dan akurat terhadap informasi spesifik. Selanjutnya, pengguna dapat menambahkan data pasien baru ke dalam sistem dengan mengisi informasi seperti nama, nomor pasien, diagnosa, dan lokasi kasur. Aplikasi juga menyediakan opsi untuk menghapus data pasien yang tidak diperlukan. Seluruh data pasien dapat dicetak untuk melihatnya secara keseluruhan, sementara opsi pencetakan berdasarkan nomor pasien memungkinkan pengguna untuk mendapatkan informasi pasien dengan nomor pasien di atas nilai tertentu. Fungsionalitas ini memberikan kemudahan dalam manajemen informasi pasien, sambil memastikan bahwa data selalu tersusun secara teratur berdasarkan nomor pasien untuk mempermudah pencarian dan pemrosesan data. Dengan demikian, aplikasi ini menciptakan suatu lingkungan administrasi yang terstruktur dan efisien untuk pemantauan pasien rawat inap.

5. Kesimpulan dan Saran

5.1 Kesimpulan:

Program Sistem Administrasi Pasien Rawat Inap yang telah dibuat berhasil memberikan solusi dalam pengelolaan data pasien secara efektif. Dengan menggunakan struktur data TreeMap, program dapat menyimpan dan mengakses informasi pasien dengan cara yang terstruktur, memudahkan pencarian dan manipulasi data. Fungsionalitas aplikasi mencakup pencarian berdasarkan nomor pasien atau nama pasien, penambahan data pasien baru, penghapusan data, serta pencetakan data untuk keperluan administrasi. Penerapan menu dengan switch-case memberikan antarmuka yang sederhana dan mudah digunakan bagi pengguna.

5.2 Saran:

Meskipun program ini telah memberikan solusi untuk pengelolaan data pasien, beberapa peningkatan dapat dipertimbangkan untuk meningkatkan fungsionalitas dan pengalaman pengguna. Pertama, penambahan fitur validasi input dapat membantu mencegah kesalahan data yang tidak diinginkan. Selanjutnya, pemberian pesan atau notifikasi yang lebih informatif saat interaksi dengan program dapat membantu pengguna memahami tindakan yang diambil oleh sistem. Selain itu, pengembangan lebih lanjut pada antarmuka grafis pengguna (GUI) dapat memberikan pengalaman yang lebih intuitif. Terakhir, mempertimbangkan integrasi dengan database untuk penyimpanan persisten dapat meningkatkan kehandalan data jangka panjang. Dengan merespons umpan balik pengguna dan terus melakukan pembaruan, program ini dapat menjadi alat administrasi yang lebih efisien dan ramah pengguna.

6. Referensi

- [1] Oktavia, Mega., Afrizal, Thomas., Sonny, Michael. (2020). "PERANCANGAN APLIKASI SISTEM ADMINISTRASI RAWAT JALAN KLINIK SUKAMAJUD DEPOK". Diakses 25 Desember 2023.
- [2] Anonim. (2023, Oktober 12). Treemap dalam Java. Diakses pada 25 Desember 2023 dari <https://www.geeksforgeeks.org/treemap-in-java/>.
- [3] Anonim. (2022, Agustus 25). Treemap dalam Java. Diakses pada 26 Desember 2023 dari <https://www.javatpoint.com/java-treemap>.
- [4] Anonim. (2022, Maret 28). Treemaps: Mengapa dan Bagaimana. Diakses pada 26 Desember 2023 dari <https://towardsdatascience.com/treemaps-why-and-how-cfb1e1c863e8>.
- [5] Ridha, M., Affandes, M., Cynthia, Eka P., Pizaini, Pizaini. (2022). "Implementasi TreeMap untuk Visualisasi Data Angka Kesakita (Morbiditas)(Studi Kasus: Dinas Kesehatan Indragiri Hilir)". Diakses pada 27 Desember 2023 dari <https://ojs.serambimekkah.ac.id/jnkti/article/view/4147>.
- [6] Pradesan, Lia., Hermanto, Dedy. (2017). "Perancangan dan Implementasi Sistem Informasi Layanan Kesehatan Masyarakat". Diakses pada 31 Desember 2023 dari <https://ojs.serambimekkah.ac.id/jnkti/article/download/4147/pdf>.