

STRUKTUR DATA DAN ALGORITMA

“Sistem Apotek”



DISUSUN OLEH KELOMPOK 5 :

- 1. Raidha Qatrunnada (2011522006)**
- 2. Delicia Syifa Maghfira (2011522007)**
- 3. Septia Amanda (2111521008)**
- 4. Gilang Kharisma (2111522002)**
- 5. Alvino Albas (2111522016)**
- 6. Husna Afiqah Yossyafra (2111522030)**
- 7. Rasyid Nugrahesa Riqua (2111523012)**
- 8. Khairin Nisa (2111523024)**

Kelas : B (02)

Dosen Pengampu : Jefril Rahmadoni, M.Kom

PROGRAM STUDI S1 SISTEM INFORMASI
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS
PADANG 2021

KATA PENGANTAR

Alhamdulillahirabbil'alamin, Puji syukur kehadiran Allah SWT. Atas segala rahmat dan hidayah-Nya. Shalawat serta salam senantiasa tercurahkan kepada junjungan Nabi Muhammad SAW yang selalu kita nantikan syafa'atnya di akhirat nanti. Puji syukur penulis panjatkan kehadiran Allah SWT, karena atas berkat dan rahmat-Nya penulis dapat menyelesaikan penyusunan Tugas Besar Struktur Data dan Algoritma ini dengan baik.

Penyusunan laporan tugas besar ini dilakukan untuk menjadi syarat penyelesaian Tugas Besar Mata Kuliah Struktur Data dan Algoritma Jurusan Sistem Informasi Universitas. Dalam penyusunan tugas ini, penulis banyak memperoleh bantuan dan masukan dari berbagai pihak, baik secara langsung maupun tidak langsung. Oleh karena itu, dalam kesempatan ini penulis mengucapkan terima kasih kepada:

1. Tuhan Yang Maha Esa
2. Bapak Jefril Rahmadoni, M.Komselaku dosen pengampu
3. Rekan-rekan mahasiswa dan sumber lain yang menjadi referensi penulis dalam menyusun laporan ini.

Penulis menyadari bahwa Laporan Tugas Besar Struktur Data & Algoritma ini masih belum sempurna, oleh karena itu segala kritik dan saran yang bersifat membangun sangat penulis harapkan. Semoga laporan ini dapat berguna bagi pembaca maupun untuk penulis sendiri.

Padang, 27 Maret 2022

Penulis,

Kelompok 5

DAFTAR ISI

BAB I.....	4
PENDAHULUAN	4
1.1 Latar Belakang	4
1.2 Rumusan Masalah	4
1.3 Tujuan.....	5
1.4 Manfaat.....	5
BAB II.....	6
LANDASAN MATERI	6
2.1 Array.....	6
2.2 Linkedlist.....	7
2.3 Stack	9
2.4 Queue	10
2.5 Binary Tree.....	12
BAB III	14
PEMBAHASAN	14
3.1 Profil Studi Kasus.....	14
3.1.1 Skenario	14
3.1.2 Flowchart	15
3.1.3 Algoritma	16
3.2 Source Code	16
3.3 Penjelasan Program	22
3.3.1 Syntax Program.....	22
3.3.2 Penjelasan Output Program.....	29
BAB IV	34
PENUTUP.....	34
4.1 Kesimpulan.....	34
4.2 Saran.....	34

BAB I

PENDAHULUAN

1.1 Latar Belakang

Seiring dengan perkembangan zaman, pemanfaatan teknologi informasi dalam pengelolaan data saat ini sangat diperlukan. Dengan berbagai keunggulan tersebut tentunya dapat meningkatkan efisiensi operasional suatu perusahaan, dimana data yang diolah menjadi informasi yang berguna bagi yang membutuhkan.

Kemajuan teknologi komputer dapat digunakan di berbagai bidang, termasuk sistem informasi yang berperan penting dalam kinerja kegiatan operasi perusahaan untuk memproses, mengumpulkan, dan menyediakan informasi untuk tujuan mendukung efisiensi dan akurasi hasil data .

Apotek merupakan salah satu usaha perdagangan dibidang kesehatan yang juga memerlukan adanya sebuah sistem pengolahan data untuk dapat mempermudah dan memperlancar kinerjanya. Oleh karena itu diperlukannya pengelolaan apotek yang baik, salah satunya dalam hal pengolahan data penjualan obat demi kelancaran usaha tersebut.

Di apotek terdapat banyak sekali data obat dan data transaksi yang tidak semua apoteker atau asisten apoteker simpan. Oleh karena itu, diperlukan sistem informasi untuk menyimpan data tersebut dengan mengelola sistem pengolahan data. Salah satunya adalah sistem pengolahan data. yang dibutuhkan yaitu sistem informasi penjualan yang berbasis komputer baik dari segi pendataan persediaan barang, transaksi penjualan serta proses laporan. Namun, masih terdapat Apotek yang masih menggunakan cara manual dalam melaksanakan pelayanan kefarmasiannya,

1.2 Rumusan Masalah

1. Bagaimana cara pengaplikasian Array dalam pembuatan program Sistem Apotek menggunakan javascript?
2. Bagaimana cara pengaplikasian Linkedlist dalam pembuatan program Sistem Apotek?
3. Bagaimana cara pengaplikasian stack dalam pembuatan program Sistem Apotek?
4. Bagaimana cara pengaplikasian Queue dalam pembuatan program Sistem Apotek?

5. Bagaimana cara pengaplikasian Binary Tree dalam pembuatan program Sistem Apotek?
6. Bagaimana cara kerja Program Sistem Apotek?
7. Bagaimana Laporan yang dapat dihasilkan dari program Sistem Apotek?

1.3 Tujuan

Adapun tujuan dari penelitian sebagai berikut:

1. Merancang sebuah sistem informasi penjualan obat di Apotek
2. Mengimplementasikan perancangan sistem informasi penjualan obat di Apotek
3. Memudahkan serta memaksimalkan kinerja Apotek dalam pengelolaan data penjualan obat.

1.4 Manfaat

Manfaat yang ingin dicapai dalam penulisan laporan ini yaitu:

1. Dapat melatih diri untuk mencari algoritma yang lebih efektif dan efisien.
2. Dapat menambah pengetahuan mengenai Struktur Data dan Algoritma.
3. Dapat menerapkan materi Struktur Data dan Algoritma

BAB II

LANDASAN MATERI

2.1 Array

Array merupakan struktur data yang digunakan untuk menyimpan sekumpulan data dalam satu tempat. Setiap data dalam Array memiliki indeks, sehingga kita akan mudah memprosesnya. Indeks array selalu dimulai dari angka nol (0). Pada teori struktur data ukuran array akan bergantung dari banyaknya data yang disimpan di dalamnya. Cara membuat array hampir sama seperti cara membuat variabel biasa. Bedanya pada array harus menentukan panjangnya.

Pembagian array berdasarkan dimensinya:

1. Array Satu Dimensi

Array satu dimensi adalah suatu array yang terdiri dari 1 subskrip array, yaitu jumlah data maksimum.

Bentuk umum:

Tipe_data nama_array[jumlah_maks_data];

2. Array Dua Dimensi

Array dua dimensi adalah suatu array yang terdiri dari 2 subskrip, yaitu jumlah elemen baris dan jumlah elemen kolom. Bentuknya dapat berupa matriks atau tabel.

Deklarasi array:

Tipe_array nama_array[baris][kolom];

Cara mengakses array:

Untuk mengakses array, misalnya kita ingin mengisi elemen array baris 2 kolom 3 dengan 10 maka perintahnya adalah sbb:

$X[1][2] = 10;$

Untuk mengisi dan menampilkan isi elemen array ada dua cara yaitu:

- a. Row Major Order (secara baris per baris),
- b. Column Major Order (secara kolom per kolom)

3. Array Multidimensi

Array multidimensi adalah suatu array yang mempunyai subskrip lebih dari dua. Bentuk pendeklarasian array sama saja dengan array dimensi satu maupun array dimensi dua.

Bentuk umumnya yaitu:

tipe_array nama_array[ukuran1][ukuran2]...[ukuranN];

2.2 Linkedlist

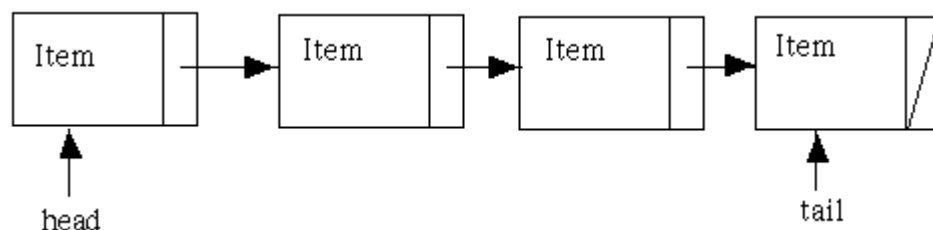
Linked List atau dikenal juga dengan sebutan senarai berantai adalah struktur data yang terdiri dari urutan record data dimana setiap record memiliki field yang menyimpan alamat/referensi dari record selanjutnya (dalam urutan). Elemen data yang dihubungkan dengan link pada Linked List disebut Node.

Ada beberapa macam Linked List, yaitu :

1. Single Linked List

Single Linked List merupakan suatu linked list yang hanya memiliki satu variabel pointer saja. Dimana pointer tersebut menunjuk ke node selanjutnya. Biasanya field pada tail menunjuk ke NULL.

contoh :



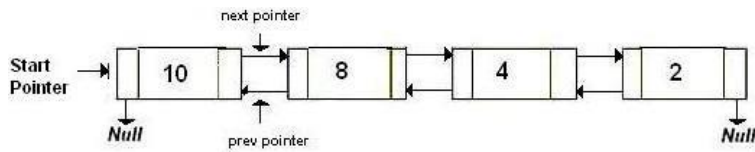
contoh codingannya :

```
struct Mahasiswa{
    char nama[25];
    int usia;
    struct Mahasiswa *next;
}*head,*tail;
```

2. Double Linked List

Double Linked List merupakan suatu linked list yang memiliki dua variabel pointer yaitu pointer yang menunjuk ke node selanjutnya dan pointer yang menunjuk ke node sebelumnya. Setiap head dan tailnya juga menunjuk ke NULL.

contoh :



contoh codingannya :

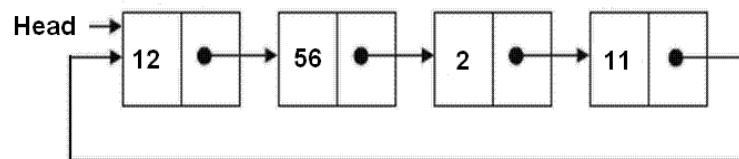
```
struct Mahasiswa{
    char nama[25];
    int usia;
    struct Mahasiswa *next,*prev;
}*head,*tail;
```

3. Circular Linked List

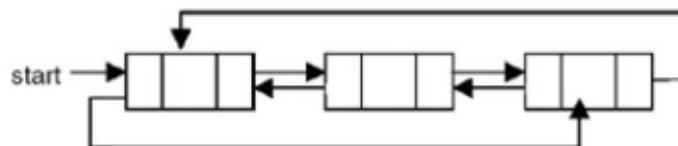
Circular Linked List merupakan suatu linked list dimana tail (node terakhir) menunjuk ke head (node pertama). Jadi tidak ada pointer yang menunjuk NULL.

Ada 2 jenis Circular Linked List, yaitu :

- Circular Single Linked List

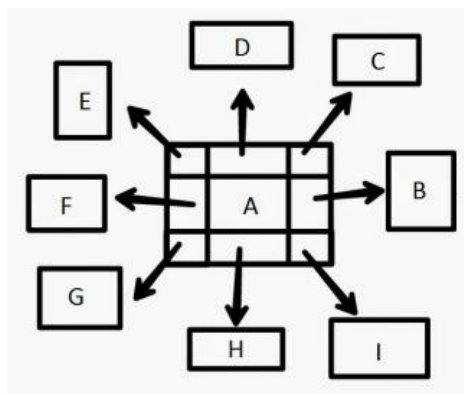


- Circular Double Linked List



4. Multiple Linked List

Multiple Linked List merupakan suatu linked list yang memiliki lebih dar 2 buah variabel pointer. contoh :



5. Priority Queue

Priority Queue mirip dengan queue biasa yang telah dijelaskan pada Array, Pointer dan Struktur Data yang dipost sebelumnya. Hanya saja queue ini di urutkan berdasarkan prioritasnya. Misalnya kita ingin membuat queue berdasarkan umur yang paling muda ke tua. Maka umur menjadi prioritas. Penyusunan node ini mungkin mirip seperti sorting.

2.3 Stack

1. Tumpukan (stack) pada Array

Stack adalah suatu tumpukan dari benda. Konsep utamanya adalah LIFO (Last In First Out), benda yang terakhir masuk dalam stack akan menjadi benda pertama yang dikeluarkan dari stack.

Operasi-operasi pada Stack dengan Array antara lain:

a. IsFull

Fungsi ini memeriksa apakah stack yang ada sudah penuh. Stack penuh jika stack penuh jika puncak stack terdapat tepat dibawah jumlah maksimum yang dapat ditampung stack atau dengan kata lain $Top = MAX_STACK - 1$.

b. Push

Fungsi ini menambahkan sebuah elemen ke dalam stack dan tidak bias dilakukan lagi jika stack sudah penuh.

c. IsEmpty

Fungsi menentukan apakah stack kosong atau tidak. Tanda bahwa stack kosong adalah Top bernilai kurang dari nol.

d. Pop

Fungsi ini mengambil elemen teratas dari stack dengan syarat stack tidak boleh kosong.

2. Tumpukan (Stack) pada linked list

Selain menggunakan array, tumpukan (stack) juga dapat dibuat dengan linked list.

Operasi-operasi Stack dengan Linked List

a. Create

Merupakan operator yang berfungsi untuk membuat sebuah tumpukan kosong.

b. IsEmpty

Merupakan operator yang berfungsi untuk menentukan apakah suatu tumpukan merupakan tumpukan kosong.

c. IsFull

Merupakan operator yang digunakan untuk memeriksa apakah tumpukan yang ada sudah penuh. Tumpukan akan penuh jika puncak tumpukan terletak tepat dibawah jumlah maksimum yang dapat ditampung.

d. Push

Merupakan operator yang berfungsi untuk menambahkan satu elemen ke dalam tumpukan dan tidak dapat dilakukan jika tumpukan dalam keadaan penuh.

e. Pop

Merupakan operator yang berfungsi untuk mengeluarkan satu elemen teratas dari dalam tumpukan dengan syarat tumpukan tidak dalam kondisi kosong.

f. Clear

Fungsi yang digunakan untuk mengosongkan tumpukan.

g. Retrieve

Fungsi yang digunakan untuk melihat nilai yang berada pada posisi tumpukan teratas.

2.4 Queue

1. Antrian (Queue) pada Array

Queue atau antrian merupakan suatu kumpulan data yang memiliki head/front dimana data dikeluarkan (dequeue) dan tail/rear dimana data dimasukkan (enqueue) ke antrian. Seperti halnya pada antrian yang biasa kita lakukan sehari-hari, di manapun. Antrian dimulai dari depan ke belakang, jika didepan belum pergi meninggalkan antrian maka antrian terus bertambah dari belakang dan antrian paling belakang disini dinamakan rear/tail. Jadi selama antrian terus bertambah (enqueue) maka antrian yang paling akhir adalah tail/rear. Jika ada yang keluar dari antrian (dequeue) maka data tersebut adalah yang paling depan (head/front), dan data berikutnya setelah data yang keluar berubah menjadi yang paling depan (head/front).

Queue menggunakan metode FIFO atau *First In First Out*, dimana data yang masuk pertama kali akan keluar pertama kali juga.

Operasi – operasi Queue pada Array :

a. IsEmpty

Fungsi IsEmpty berguna untuk mengecek apakah queue masih kosong atau sudah berisi data. Hal ini dilakukan dengan mengecek apakah tail bernilai -1 atau tidak. Nilai -1 menandakan bahwa queue masih kosong.

b. IsFull

Fungsi IsFull berguna untuk mengecek apakah queue sudah penuh atau masih bisa menampung data dengan cara mengecek apakah nilai tail sudah sama dengan jumlah maksimal queue. Jika nilai keduanya sama, berarti queue sudah penuh.

c. EnQueue

Fungsi EnQueue berguna untuk memasukkan sebuah elemen dalam queue.

d. DeQueue

Fungsi DeQueue berguna untuk mengambil sebuah elemen dari queue. Operasi ini sering disebut juga serve. Hal ini dilakukan dengan cara memindahkan sejauh satu langkah ke posisi di depannya sehingga otomatis elemen yang paling depan akan tertimpa dengan elemen yang terletak di belakangnya.

e. Clear

Fungsi Clear berguna untuk menghapus semua elemen dalam queue dengan jalan mengeluarkan semua elemen tersebut satu per satu hingga queue kosong dengan memanfaatkan fungsi DeQueue.

2. Antrian (Queue) pada linked list

Selain menggunakan array, Queue juga dapat dibuat dengan linked list. Metode linked list yang digunakan adalah double linked list.

Operasi-operasi Queue dengan Double Linked List antara lain :

a. IsEmpty

Fungsi IsEmpty berguna untuk mengecek apakah queue masih kosong atau sudah berisi data. Hal ini dilakukan dengan mengecek apakah head masih menunjukkan pada Null atau tidak. Jika benar berarti queue masih kosong.

b. IsFull

Fungsi IsFull berguna untuk mengecek apakah queue sudah penuh atau masih bisa menampung data dengan cara mengecek apakah Jumlah Queue sudah sama dengan MAX_QUEUE atau belum. Jika benar maka queue sudah penuh.

c. EnQueue

Fungsi EnQueue berguna untuk memasukkan sebuah elemen ke dalam queue (head dan tail mula-mula meunjukkan ke NULL). Penambahan data dilakukan di belakang. namun pada saat pertama kali data langsung ditunjuk pada head-nya. Penambahan di belakang lebih sulit karena kita membutuhkan pointer bantu untuk mengetahui data terbelakang, kemudian dikaitkan dengan data baru. Untuk mengetahui data terbelakang perlu digunakan perulangan.

d. DeQueue

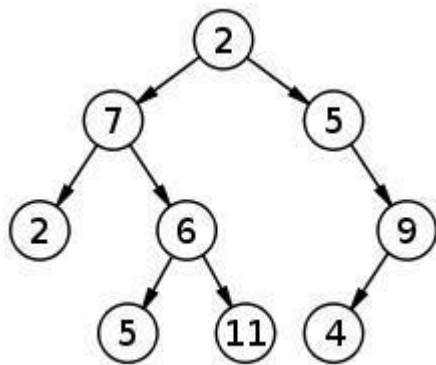
Prosedur DeQueue berguna untuk mengambil sebuah elemen dari queue. Hal ini dilakukan dengan cara menghapus satu simpul yang terletak paling depan (head).

2.5 Binary Tree

Binary tree adalah sebuah struktur data yang menyerupai pohon dan setiap simpulnya memiliki cabang maksimal 2. Pada setiap pohon biner memiliki Root dan Leaf. Root adalah simpul utama yang merupakan simpul awal pada suatu pohon biner.

Sedangkan Leaf adalah adalah simpul terakhir yang tidak memiliki cabang lagi.

contohnya :



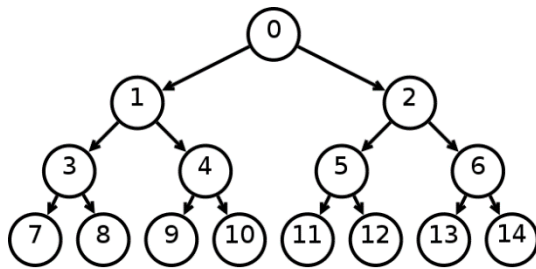
Pada pohon biner diatas, rootnya adalah 2 dan leafnya adalah 2, 5, 11, 4.

Tipe-Tipe Pohon Biner

1. Perfect Binary Tree

PBT adalah suatu pohon biner yang setiap levelnya memiliki kedalaman yang sama. Terkadang PBT juga termasuk CBT (complete binary tree).

Contoh :



2. Complete Binary Tree

CBT adalah suatu pohon biner yang kedalamannya sebesar n atau $n-1$ untuk beberapa n . Jadi tidak seperti PBT yang harus sama semuanya, melainkan boleh sama ataupun tidak (namun pada simpul kedua dari terakhir saja). Dan dalam penempatan simpulnya diutamakan yang sebelah kiri yang terpenuhi.

3. Skewed Binary Tree

SBT merupakan suatu pohon biner yang setiap simpulnya hanya memiliki satu anak atau satu cabang, sehingga membuatnya tidak seimbang.

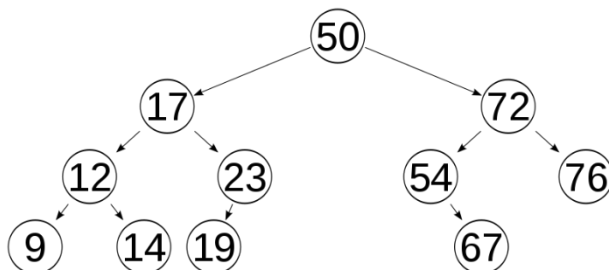
Contoh :



4. Balanced Binary Tree

BBT adalah suatu pohon biner yang tinggi antara anak sebelah kiri dan kanannya hanya berselisih maksimal satu. PBT dan CBT juga merupakan binary tree yang seimbang.

Contoh :



BAB III

PEMBAHASAN

3.1 Profil Studi Kasus

3.1.1 Skenario

Program Sistem Apotek merupakan program yang dibuat untuk membantu penjualan obat pada sebuah apotek dengan tujuan agar transaksi jual beli dapat dengan lebih mudah dilakukan dan lebih terstruktur dalam penyimpanannya.

Pada program ini terdapat daftar nama obat apa saja yang tersedia di apotek. Pada apotek ini terdapat 4 jenis obat yang tersedia dan juga 1 alat kesehatan, yaitu

1. Paracetamol dengan harga per itemnya 5.000
2. Amoxcillin dengan harga per itemnya 6.000
3. Asam mefenamat dengan harga per itemnya 3.000
4. Bodrex dengan harga per itemnya 4.000
5. Masker dengan harga per itemnya 15.000

Pada program ini terdapat beberapa menu dengan tahapannya, yaitu pengecekan stok obat atau item, penambahan stok obat atau item, pembelian obat atau item dan pengecekan antrian pembelian obat yang dilakukan oleh pelanggan yang datang pada apotek, lalu menu tutup untuk menutup program. Untuk menjalankan perintah program tersebut dapat dilakukan dengan menginputkan indeks angka sesuai urutan pada program oleh user.

Setiap pelanggan yang datang dan melakukan pemesanan maka namanya akan diinputkan pada program dan akan tersimpan pada array. Dengan memberikan ketengan antrian dan nomor tempat duduk yang akan ditempati pelanggan. Lalu, dapat melakukan pemesanan obat yang diinginkan. Setelah memesan maka akan muncul output total harga yang harus dibayarkan, silahkan inputkan jumlah uang yang akan dibayarkan, jika jumlah uang yang diinputkan lebih dari total harga, maka akan muncul output jumlah kembalian uang pelanggan.

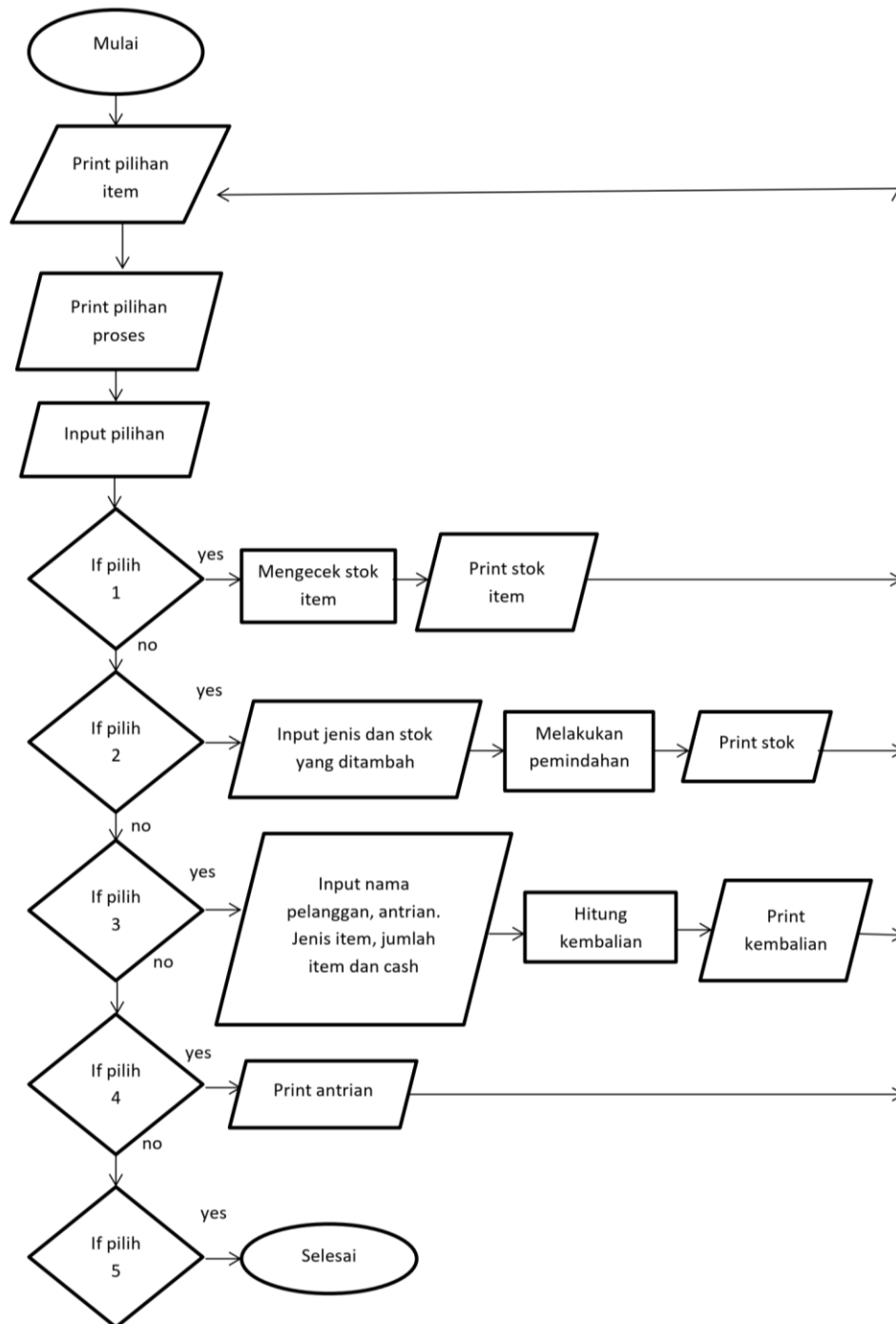
Karna setiap transaksi disimpan dalam array, maka dapat dilakukan pencarian daftar nama pelanggan yang sudah melakukan transaksi dengan mengecek antrian pembelian.

Jika ingin melakukan pengecekan stok obat maka inputkan indeks angka sesuai urutan menu yang mana akan muncul output stok obat yang tersedia pada apotek. Begitupun dengan penambahan stok, dapat dilakukan dengan menginputkan indeks

angka sesuai urutan menu lalu silahkan inputkan indeks angka jenis obat yang ingin ditambahkan stoknya. Maka program akan berjalan sesuai perintah.

Jika tidak ada lagi pelanggan yang melakukan transaksi atau jika toko ingin tutup, maka user dapat memilih menu tutup pada program dan program selesai.

3.1.2 Flowchart



3.1.3 Algoritma

1. Memulai Program dimana sudah terdapat 5 jenis obat dan 6 menu pilihan
2. Menu I untuk pengecekan Stok obat. Memilih jenis obat dan program akan menampilkan semua Stock jenis obatnya.
3. Menu ke II untuk menambah Stok obat jika obat tersisa sedikit atau sudah habis. Pilih jenis obat yang ingin ditambahkan stoknya dan inputkan jumlah obat yang akan ditambahkan
4. Menu III pelanggan datang, lalu user menginputkan nama pelanggan lanjut ke nomor antrian, lalu tampilan tempat duduk yang akan di tempati oleh pelanggan. Barulah penginputan jenis obat yang akan dibeli pelanggan beserta jumlah yang diinginkan. Maka akan muncul total harga, silahkan inputkan uang yang dibayarkan pelanggan. Jika uang berlebih dari total harga maka akan muncul output kembalian pada program.
5. Menu ke IV Menampilkan urutan antrian pelanggan
6. Menu ke V menutup program dan program selesai.

3.2 Source Code

```
package Kelompok;

import java.util.LinkedList;
import java.util.Queue;
import java.util.Scanner;
import java.util.Stack;

public class apotek {
    Scanner in = new Scanner(System.in);

    public static void main(String[] args) {

        String pilihanUser;
        int total;
        int angka;
        int jumlah, cash, kembalian;
        int nama_obat;
        int harga_obat = 0;
        String a;
        String Obatnya;
        apotek tr = new apotek();
        node root = new node();
        int r = 1;
        int c;
```



```

boolean bool = true;

int[] harga = { 5000, 6000, 3000, 4000, 15000 };
String[] obatt = { "Paracetamol", " Amoxicillin", " Asam mefenamat ", " Bodrex ", "
Masker " };
int[] stock = { 100, 300, 75, 90, 60 };

Queue<String> namapelanggan = new LinkedList<>();
Stack<String> masker = new Stack<String>();
Stack<String> masker2 = new Stack<String>();

System.out.println(" =====");
System.out.println("          SELAMAT DATANG          ");
System.out.println(" ===== ");
System.out.println("-----APOTEK BERKAH-----");
System.out.println(" =====");

System.out.println("\n=====");
System.out.println("          OBAT dan ITEM          ");
System.out.println("=====");
System.out.println(" 1. Paracetamol    || Rp5.000 ");
System.out.println(" 2. Amoxicillin    || Rp6.000");
System.out.println(" 3. Asam mefenamat || Rp3.000");
System.out.println(" 4. Bodrex        || Rp4.000");
System.out.println(" 5. Masker        || Rp15.000");
System.out.println("\n=====");
System.out.println();

Scanner pilih = new Scanner(System.in);
Scanner Pilih = new Scanner(System.in);
do {
    System.out.println("\n=====");
    System.out.println("          MENU          ");
    System.out.println("=====");
    System.out.println(" 1. Cek Stock obat/item");
    System.out.println(" 2. Tambah stock obat/item");
    System.out.println(" 3. Pembelian Obat/item");
    System.out.println(" 4. Cek Antrian Pembelian");
    System.out.println(" 5. Tutup ");

    System.out.print("MASUKAN PILIHAN [1-5]: ");
    pilihanUser = pilih.next();

    switch (pilihanUser) {
        case "1":

            for (int i = 0; i < stock.length; i++) {
                System.out.println("Nama obat/item : " + obatt[i] + " \t stock : " + stock[i]);
                System.out.println();
            }

```

```

    }
    break;

case "2":
    System.out.print("\n > Masukkan jenis Obat/Item : ");
    nama_obat = pilih.nextInt();
    pilih.nextLine();

    if (nama_obat == 1) {
        Obatnya = obatt[0];
        System.out.println("Nama obat : " + Obatnya);
        System.out.print("Masukkan stock yang di tambahkan : ");
        angka = pilih.nextInt();
        stock[0] = stock[0] + angka;
        System.out.println("stock obat yang tersedia : " + stock[0]);
        System.out.println();
    } else if (nama_obat == 2) {
        Obatnya = obatt[1];
        System.out.println("nama obat : " + Obatnya);
        System.out.print(" masukkan stock yang di tambahkan : ");
        angka = pilih.nextInt();
        stock[1] = stock[1] + angka;
        System.out.println("stock obat yang tersedia : " + stock[1]);
        System.out.println();
    } else if (nama_obat == 3) {
        Obatnya = obatt[2];
        System.out.println("nama obat : " + Obatnya);
        System.out.print(" masukkan stock yang di tambahkan : ");
        angka = pilih.nextInt();
        stock[2] = stock[2] + angka;
        System.out.println("stock obat yang tersedia : " + stock[2]);
        System.out.println();
    } else if (nama_obat == 4) {
        Obatnya = obatt[3];
        System.out.println("nama obat : " + Obatnya);
        System.out.print(" masukkan stock yang di tambahkan : ");
        angka = pilih.nextInt();
        stock[3] = stock[3] + angka;
        System.out.println("stock obat yang tersedia : " + stock[3]);
        System.out.println();
    } else if (nama_obat == 5) {
        Obatnya = obatt[4];
        System.out.println("Nama item : " + Obatnya);
        System.out.print("Masukkan stock yang di tambahkan : ");
        angka = pilih.nextInt();
        stock[4] = stock[4] + angka;

        for (int b = 1; b <= angka; b++)// perulangan for
        {

```

```

        masker.push("Kotak Masker " + b);// proses memasukkan data pada stack
    }
    System.out.println(
        "\n== Kondisi Kotak Masker di Kardus ==\nTumpukan Kotak Masker
di Kardus :");

    for (int b = 1; b <= angka; b++)// perulangan for
    {
        System.out.println(masker.peek());
        masker2.push(masker.pop());// proses pengeluaran data pada stack
    }
    System.out.println("\n<< DILAKUKAN PEMINDAHAKN >>\n");
    System.out.println("== Kondisi Kotak Masker di Rak ==\nTumpukan Kotak
Masker di rak :");
    for (int b = angka - 1; b >= 0; b--) {
        System.out.println(masker2.elementAt(b));
    }
    System.out.println("stock obat yang tersedia : " + stock[4]);
    System.out.println();
} else {
    System.out.println("Nomor yang anda masukkan salah");
    System.out.print("\n > Masukkan jenis obat : ");
    nama_obat = pilih.nextInt();
    pilih.nextLine();
}

break;

case "3":
    int n = 0;
    n++;
    for (int j = 0; j < n; j++) {
        System.out.print("Masukkan Nama pelanggan : ");
        a = pilih.next();
        namapelanggan.add(a);
    }

    System.out.print("Masukkan No.Antrian : ");
    c = tr.in.nextInt();
    if (r == 1) {
        root.input(c);
        r--;
    } else
        tr.insert(root, c);
    tr.view(root);
    System.out.println(" ");

    System.out.print("\n > Masukkan jenis Obat/Item : ");

```

```

nama_obat = pilih.nextInt();
pilih.nextLine();
System.out.print(" Masukkan jumlah : ");
jumlah = Pilih.nextInt();

if (nama_obat == 1) {
    harga_obat = harga[0];
    stock[0] = stock[0] - jumlah;
} else if (nama_obat == 2) {
    harga_obat = harga[1];
    stock[1] = stock[1] - jumlah;
} else if (nama_obat == 3) {
    harga_obat = harga[2];
    stock[2] = stock[2] - jumlah;
} else if (nama_obat == 4) {
    harga_obat = harga[3];
    stock[3] = stock[3] - jumlah;
} else if (nama_obat == 5) {
    harga_obat = harga[4];
    stock[4] = stock[4] - jumlah;
} else {

    System.out.println("Nomor yang anda masukkan salah");
    System.out.print("\n > Masukkan jenis Obat/item : ");
    nama_obat = pilih.nextInt();
    pilih.nextLine();
}

total = harga_obat * jumlah;
System.out.print(" Total harga : Rp. " + total);
System.out.println("");
System.out.print(" Cash : Rp. ");
cash = Pilih.nextInt();

if (total > cash) {
    System.out.println("Uang anda kurang");
    System.out.print("\n > Masukkan uang anda : ");
    cash = pilih.nextInt();
    pilih.nextLine();
    System.out.println("");
    kembalian = cash - total;
    System.out.println("Kembalian : Rp. " + kembalian);
    System.out.println("");
} else {
    kembalian = cash - total;
    System.out.println("Kembalian : Rp. " + kembalian);
    System.out.println("");
}

break;

```

```

        case "4":
            System.out.print(" Cek antrian : " + namapelanggan.isEmpty());
            System.out.println(" antrian : " + namapelanggan);
            System.out.println();
            break;
        case "5":
            System.exit(0);
            break;
        default:
            System.err.println("\nInput anda tidak ditemukan\nInputkan [1-5]");
    }
} while (bool);
}
public void insert(node c, int b) {
    if (b < c.value) {
        if (c.left != null)
            insert(c.left, b);
        else {
            c.left = new node();
            c.left.input(b);
            System.out.println("\nPosisi duduk Antrian " + b + " di kiri Antrian " + c.value);
        }
    } else if (b > c.value) {
        if (c.right != null)
            insert(c.right, b);
        else {
            c.right = new node();
            c.right.input(b);
            System.out.println("\nPosisi duduk Antrian " + b + " di kanan Antrian " +
c.value);
        }
    }
}

public void view(node c) {
    System.out.print("Tampilan duduk menurut Pre Order : ");
    preOrder(c);
    System.out.println(" ");
}

public void preOrder(node c) {
    if (c != null) {
        System.out.print(c.value + " ");
        preOrder(c.left);
        preOrder(c.right);
    }
}
}

```

```

class node {
    node left, right;
    int value;

    public void input(int c) {
        value = c;
    }
}

```

3.3 Penjelasan Program

3.3.1 Syntax Program

Import Pada Java Merupakan Suatu Perintah Untuk Memasukan suatu Method atau perintah dalam Bahasa Pemrograman Java sehingga perintah tersebut dapat Aktif dan digunakan atau berfungsi.

```

1  package Kelompok;
2
3  import java.util.LinkedList;
4  import java.util.Queue;
5  import java.util.Scanner;
6  import java.util.Stack;
7
8  public class apotek {
9      Scanner in = new Scanner(System.in);
10

```

Import di atas digunakan untuk berbagai macam perintah, mulai dari Linkedlist untuk menyimpan data liner, Queue untuk menyimpan antrian pelanggan dari perintah dan inputan yang dilakukan user, Scanner berfungsi sebagai kelas untuk melakukan masukan dari keyboard berbasis DOS, dan stack untuk memanggil dan menyimpan tumpukan nama pelanggan dan nama obat pada program.

Public : adalah kalimat pertama pada java yang berfungsi sebagai memberikan akses publik kepada class dari sebuah program yang dijalankan oleh java runtime (main method).Jadi ketika program dijalankan ,maka java runtime akan mengakses sebuah kelas yang berisi string / sintaks tertentu pada java.


```

76         case "2":
77             System.out.print("\n\t\t\t\t > Masukkan jenis Obat/Item : ");
78             nama_obat = pilih.nextInt();
79             pilih.nextLine();
80
81             if (nama_obat == 1) {
82                 Obatnya = obatt[0];
83                 System.out.println("\t\t\t\tNama obat : " + Obatnya);
84                 System.out.print("\t\t\t\t > Masukkan stock yang di tambahkan : ");
85                 angka = pilih.nextInt();
86                 stock[0] = stock[0] + angka;
87                 System.out.println("\t\t\t\tstock obat yang tersedia : " + stock[0]);
88                 System.out.println();
89             } else if (nama_obat == 2) {
90                 Obatnya = obatt[1];
91                 System.out.println("\t\t\t\tnama obat : " + Obatnya);
92                 System.out.print("\t\t\t\t > masukkan stock yang di tambahkan : ");
93                 angka = pilih.nextInt();
94                 stock[1] = stock[1] + angka;
95                 System.out.println("\t\t\t\tstock obat yang tersedia : " + stock[1]);
96                 System.out.println();
97             } else if (nama_obat == 3) {
98                 Obatnya = obatt[2];
99                 System.out.println("\t\t\t\tnama obat : " + Obatnya);
100                System.out.print("\t\t\t\t > masukkan stock yang di tambahkan : ");
101                angka = pilih.nextInt();
102                stock[2] = stock[2] + angka;
103                System.out.println("\t\t\t\tstock obat yang tersedia : " + stock[2]);
104                System.out.println();
105
106            } else if (nama_obat == 4) {
107                Obatnya = obatt[3];
108                System.out.println("\t\t\t\tnama obat : " + Obatnya);
109                System.out.print("\t\t\t\t > masukkan stock yang di tambahkan : ");
110                angka = pilih.nextInt();
111                stock[3] = stock[3] + angka;
112                System.out.println("\t\t\t\tstock obat yang tersedia : " + stock[3]);
113                System.out.println();
114            } else if (nama_obat == 5) {
115                Obatnya = obatt[4];
116                System.out.println("\t\t\t\tNama item : " + Obatnya);
117                System.out.print("\t\t\t\t > Masukkan stock yang di tambahkan : ");
118                angka = pilih.nextInt();
119                stock[4] = stock[4] + angka;
120                System.out.println("\t\t\t\tstock item yang tersedia : " + stock[4]);
121                System.out.println();
122            }
123        }
124    }
125 }

```

Pada case 2 diatas terdapat beberapa perintah yang nantinya dapat menginputkan nama obat yang diinginkan pelanggan dan akan menampilkan keluaran.

Diatas terdapat 3 bentuk percabangan dengan menggunakan kata kunci if, else, switch dan case.

- Pada percabangan IF hanya memiliki satu pilihan. Artinya pilihan di dalam IF hanya akan dikerjakan jika kondisi benar, sebaliknya jika salah tidak akan melakukan perintah apa-apa dan lanjut ke perintah berikutnya. Diatas terdapat sysout nama obat dan obat yang akan ditambahkan. Dimana user dapat menginputkan angka 1-5 dengan indeks 0-4 yang nantinya akan menampilkan output jika stok sudah ditambahkan maka akan muncul jumlah stok akhir.
- Else If dimana selama indeks yang diinputkan user bernilai benar, maka program akan memproses, namun sebaliknya jika salah maka program akan memproses yang

lain atau setelahnya lanjut.

```
119
120         for (int b = 1; b <= angka; b++)// perulangan for
121         {
122             masker.push("\t\t\t\tKotak Masker " + b);// proses memasukkan data pada stack
123         }
124         System.out.println(
125             "\n\t\t\t\t== Kondisi Kotak Masker di Kardus ==\n\t\t\t\tTumpukan Kotak Masker di Kardus :");
126
127         for (int b = 1; b <= angka; b++)// perulangan for
128         {
129             System.out.println(masker.peek());
130             masker2.push(masker.pop());// proses pengeluaran data pada stack
131         }
132         System.out.println("\n\t\t\t\t<< DILAKUKAN PEMINDAHAN >>\n");
133         System.out.println("\t\t\t\t== Kondisi Kotak Masker di Rak ==\n\t\t\t\tTumpukan Kotak Masker di rak :");
134         for (int b = angka - 1; b >= 0; b--) {
135             System.out.println(masker2.elementAt(b));
136         }
137         System.out.println("\t\t\t\tstock obat yang tersedia : " + stock[4]);
138         System.out.println();
139     } else {
140         System.out.println("\t\t\t\tNomor yang anda masukkan salah");
141         System.out.print("\n\t\t\t\t> Masukkan jenis obat : ");
142         nama_obat = pilih.nextInt();
143         pilih.nextLine();
144     }
145
146     break;
```

Di atas terdapat perulangan for dengan integer b. Dimana pada program digunakan perintah push dan pop yang nantinya digunakan untuk memasukkan data dan menghapus data. Pada program diatas digunakan stack, yang nantinya akan dilakukan pemindahan untuk tumpukan masker yang ada di rak. Setiap perintah for diakhiri dengan break.

Case 3

```
148         case "3":
149             int n = 0;
150             n++;
151             for (int j = 0; j < n; j++) {
152                 System.out.print("\t\t\t\tMasukkan Nama pelanggan : ");
153                 a = pilih.next();
154                 namapelanggan.add(a);
155             }
156
157             System.out.print("\t\t\t\tMasukkan No.Antrian : ");
158             c = tr.in.nextInt();
159             if (r == 1) {
160                 root.input(c);
161                 r--;
162             } else
163                 tr.insert(root, c);
164             tr.view(root);
165             System.out.println(" ");
166
167             System.out.print("\n\t\t\t\t> Masukkan jenis Obat/Item : ");
168             nama_obat = pilih.nextInt();
169             pilih.nextLine();
170             System.out.print("\t\t\t\tMasukkan jumlah : ");
171             jumlah = Pilih.nextInt();
172
173             if (nama_obat == 1) {
174                 harga_obat = harga[0];
175                 stock[0] = stock[0] - jumlah;
176             } else if (nama_obat == 2) {
177                 harga_obat = harga[1];
```

```

174         harga_obat = harga[0];
175         stock[0] = stock[0] - jumlah;
176     } else if (nama_obat == 2) {
177         harga_obat = harga[1];
178         stock[1] = stock[1] - jumlah;
179     } else if (nama_obat == 3) {
180         harga_obat = harga[2];
181         stock[2] = stock[2] - jumlah;
182     } else if (nama_obat == 4) {
183         harga_obat = harga[3];
184         stock[3] = stock[3] - jumlah;
185     } else if (nama_obat == 5) {
186         harga_obat = harga[4];
187         stock[4] = stock[4] - jumlah;
188     } else {
189         System.out.println("\t\t\t\t\tNomor yang anda masukkan salah");
190         System.out.print("\n\t\t\t\t\t > Masukkan jenis Obat/item : ");
191         nama_obat = pilih.nextInt();
192         pilih.nextLine();
193     }
194
195     total = harga_obat * jumlah;
196     System.out.print("\t\t\t\t\t Total harga : Rp. " + total);
197     System.out.println("");
198     System.out.print("\t\t\t\t\t Cash : Rp. ");
199     cash = Pilih.nextInt();
200
201     if (total > cash) {
202         System.out.println("\t\t\t\t\t Uang anda kurang");
203         System.out.print("\n\t\t\t\t\t > Masukkan uang anda : ");
204
205         cash = pilih.nextInt();
206         pilih.nextLine();
207         System.out.println("");
208         kembalian = cash - total;
209         System.out.println("\t\t\t\t\tKembalian : Rp. " + kembalian);
210         System.out.println("");
211     } else {
212         kembalian = cash - total;
213         System.out.println("\t\t\t\t\tKembalian : Rp. " + kembalian);
214         System.out.println("");
215     }
216
217     break;

```

Pada Case 3 terdapat perulangan for dengan tipe data integer n=0. Dimana program meminta pengguna untuk menginput nama pelanggan dan nomor antrian. Selanjutnya program akan memasukkan node ke dalam tree. Pengguna diminta untuk menginput jenis obat dan jumlah obat yang ingin dibeli. Disini program menggunakan fungsi percabangan if else if untuk menampilkan data yang telah disimpan pada array sesuai apa yang diminta pelanggan. Setelah itu program akan memproses harga obat yang mesti dibayar oleh pelanggan dengan menggunakan percabangan if else.

Case 4

```

219         case "4":
220             System.out.print("\t\t\t\t\t Cek antrian : " + namapelanggan.isEmpty());
221             System.out.println("\t\t\t\t\t antrian : " + namapelanggan);
222             System.out.println();
223             break;

```

Pada Case 4 terdapat sysout Cek antrian dimana jika terdapat antrian dan antrian bukan kosong atau isEmpty maka akan muncul output false, jika antrian bernilai benar atau antrian memang kosong, maka akan muncul output true.

Case 5

```

225         case "5":
226             System.exit(0);
227             break;
228
229         default:
230             System.err.println("\n\t\t\t\tInput anda tidak ditemukan\nInputkan [1-5]");
231     }
232 } while (bool);
233

```

pada case 5 terdapat system,exit yang nantinya akan memberhentikan program. Dan program selesai.

Jika user menginputkan bukan angka 1-5, maka akan muncul output seperti diatas. Dimana terdapat while dengan kondisi bool yang mana maksudnya akan berhenti sampai kondisi bernilai false.

```

235 public void insert(node c, int b) {
236     if (b < c.value) {
237         if (c.left != null)
238             insert(c.left, b);
239         else {
240             c.left = new node();
241             c.left.input(b);
242             System.out.println("\n\t\t\t\tPosisi duduk Antrian " + b + " di kiri Antrian " + c.value);
243         }
244     } else if (b > c.value) {
245         if (c.right != null)
246             insert(c.right, b);
247         else {
248             c.right = new node();
249             c.right.input(b);
250             System.out.println("\n\t\t\t\tPosisi duduk Antrian " + b + " di kanan Antrian " + c.value);
251         }
252     }
253 }

```

Pada potongan program diatas terdapat syntax IF ELSE IF yang mana terdapat binary tree untuk mengetahui posisi duduk antrian apakah berada disebelah kiri atau kanan.

```

254 public void view(node c) {
255     System.out.print("\t\t\t\tTampilan duduk menurut Pre Order : ");
256     preOrder(c);
257     System.out.println(" ");
258 }
259
260 public void preOrder(node c) {
261     if (c != null) {
262         System.out.print(c.value + " ");
263         preOrder(c.left);
264         preOrder(c.right);
265     }
266 }
267
268 }

```

Pada program di atas terdapat tampilan duduk menurut PreOrder dengan metode ini akan menampilkan angka atau nomor tempat duduk.

```

270 class node {
271     node left, right;
272     int value;
273
274     public void input(int c) {
275         value = c;
276     }
277 }
278

```

3.3.2 Penjelasan Output Program

```
=====
                        SELAMAT DATANG
=====
-----APOTEK BERKAH-----
=====

=====
                        OBAT dan ITEM
=====

1. Paracetamol      || Rp5.000
2. Amoxicillin      || Rp6.000
3. Asam mefenamat   || Rp3.000
4. Bodrex           || Rp4.000
5. Masker           || Rp15.000

=====

=====
                        MENU
=====

1. Cek Stock obat/item
2. Tambah stock obat/item
3. Pembelian Obat/item
4. Cek Antrian Pembelian
5. Tutup
MASUKAN PILIHAN [1-5]:
```

Di atas adalah output pada toko apotek berkah. Dimana terdapat ucapan selamat datang, pilihan obat dan item serta menu yang akan menampilkan output sesuai inputan yang dilakukan user.

Di atas terdapat 4 jenis obat dan 1 item, yaitunya

Paracetamol seharga Rp. 5.000/itemnya

Amoxcillin seharga Rp. 6.000/itemnya

Asam Mefenamat seharga Rp. 3.000/itemnya

Boderex seharga Rp. 4.000/itemnya

Masker seharga Rp. 15.000/itemnya

Dengan menu yang tersedia, yaitu :

1. Cek strok obat/item
2. Tambah stok obat/item
3. Pembelian obat/item

4. Cek antrian pembelian
5. Tutup.

Serta user dapat menginputkan pilihan pada Masukkan pilihan 1-5

- **Menu 1**

```
=====
                        MENU
=====
1. Cek Stock obat/item
2. Tambah stock obat/item
3. Pembelian Obat/item
4. Cek Antrian Pembelian
5. Tutup
MASUKAN PILIHAN [1-5]: 1
Nama obat/item : Paracetamol    stock :100

Nama obat/item : Amoxicillin    stock :300

Nama obat/item : Asam mefenamat    stock :75

Nama obat/item : Bodrex          stock :90

Nama obat/item : Masker          stock :60
```

Pada menu 1 ini terdapat Cek Stok Obat/Item, karena cek stok obat terdapat pada menu 1 dan merupakan indeks 0 dengan urutan pertama, maka jika user menginputkan angka 1 program akan berjalan dan menampilkan stok yang tersedia pada semua jenis obat/item. Terlihat bahwa,

Paracetamol memiliki stok 100

Amoxicillin memiliki stok 300

Asam mefenamat memiliki stok 75

Bodrex memiliki stok 90

Masker memiliki stok 60

- **Menu 2**

```
=====
                        OBAT dan ITEM
=====

1. Paracetamol      || Rp5.000
2. Amoxicillin      || Rp6.000
3. Asam mefenamat   || Rp3.000
4. Bodrex           || Rp4.000
5. Masker           || Rp15.000
=====

=====
                        MENU
=====

1. Cek Stock obat/item
2. Tambah stock obat/item
3. Pembelian Obat/item
4. Cek Antrian Pembelian
5. Tutup
MASUKAN PILIHAN [1-5]: 2

> Masukkan jenis Obat/Item : 1
Nama obat : Paracetamol
Masukkan stock yang di tambahkan : 120
stock obat yang tersedia : 220
```

Selanjutnya, jika user memilih menu 2 dan ingin menambahkan stok suatu obat/item, maka user dapat menginputkan angka 2 karena Tambah Stok berada pada indeks 1 dan no 2.

- Setelah user menginputkan angka 2 maka user akan di minta untuk memasukkan jenis obat/item yang ingin ditambahkan.
- Misal, pada percobaan diatas user menambahkan obat dengan angka 1 dan indeks 0 dimana itu adalah paracetamol.
- Lalu, user diminta untuk menginputkan jumlah stok obat yang ingin ditambahkan, misal pada contoh di atas user menginputkan jumlah 120,
- karena tadinya stok paracetamol adalah 100, maka otomatis program akan menambahkan jumlah stok obat yang tersedia dengan menggunakan matriks. Jadi jumlah stok obat yang tersedia adalah 220.

- **Menu 3**

```
=====
                        MENU
=====
1. Cek Stock obat/item
2. Tambah stock obat/item
3. Pembelian Obat/item
4. Cek Antrian Pembelian
5. Tutup
MASUKAN PILIHAN [1-5]: 3
Masukkan Nama pelanggan : james
Masukkan No.Antrian : 1
Tampilan duduk menurut Pre Order : 1

> Masukkan jenis Obat/Item : 4
Masukkan jumlah : 2
Total harga : Rp. 8000
Cash : Rp. 10000
Kembalian : Rp. 2000
```

Selanjutnya menu 3, pada menu 3 ini terdapat Pembelian Obat/item yang berada pada indeks ke 2 dengan angka 3. Pada menu ini user dapat menginputkan angka 3 untuk menjalankan program, pada menu ini berguna untuk user menginputkan nama pelanggan yang datang.

- Misal pada contoh run diatas, user menginputkan nama pelanggan yang datang yaitunya james, dimana james berada pada no. antrian 1 sesuai inputan user.
- Maka program akan menampilkan duduk menurut Pre Order dimana terdapat penggunaan Binary Tree dan james berada pada tampilan duduk 1.
- Lalu silahkan user inputkan jenis obat/item yang ingin dibeli pelanggan, contoh pelanggan ingin membeli obat dengan angka 4 berarti pada indeks ke 3 yaitunya, Bodrex.
- Silahkan inputkan jumlah oabat/item yang ingin dibeli oleh pelanggan, misal 2 item.
- Maka akan muncul output total harga yang harus dibayarkan pelanggan yanitunya Rp. 8000 dari haril perkalian matriks pada program.
- Karena pelanggan membayar case senilai Rp. 1000.
- Maka akan muncul output kembalian senilai Rp.2000.

- **Menu 4**


```
=====
                        MENU
=====

1. Cek Stock obat/item
2. Tambah stock obat/item
3. Pembelian Obat/item
4. Cek Antrian Pembelian
5. Tutup
MASUKAN PILIHAN [1-5]: 4
Cek antrian : false antrian : [james]
```

Pada menu 4, terdapat Cek antrian pembelian, dimana menu 4 berada pada indeks ke 3 dengan no 4. Karena terdapat antrian dan antrian bukan kosong atau IsEmpty. Maka akan muncul output false antrian yaitun, james.

- **Menu 5**

```
=====
                        MENU
=====

1. Cek Stock obat/item
2. Tambah stock obat/item
3. Pembelian Obat/item
4. Cek Antrian Pembelian
5. Tutup
MASUKAN PILIHAN [1-5]: 5
```

Pada menu 5 merupakan menu yang berada pada indeks ke 4 dengan no 5 dimana terdapat menu tutup yang jika dijadikan pilihan Ketika tidak ada lagi pelanggan atau pengecekan suatu barang. Dan program selesai.

BAB IV

PENUTUP

4.1 Kesimpulan

Dengan membuat program sistem apotek terkomputerisasi dan terintegrasi maka seluruh aktifitas dapat dilakukan dengan optimal. Menggunakan konsep array, LinkedList, Stack, Queue, dan Binary Tree dengan menggunakan aplikasi Visual Studio Code. Dengan adanya penerapan sistem informasi penjualan dapat membantu dan membuat kinerja menjadi lebih maksimal dalam pengelolaan data obat dan penjualan sehingga lebih efektif dan efisien.

4.2 Saran

Setelah perancangan dan pembuatan program sistem apotek penulis menyadari bahwa dalam tugas besar ini masih jauh dari sempurna. Oleh sebab itu, penulis memberikan saran tersebut adalah sebagai berikut

1. Menambah fitur-fitur yang diperlukan
2. Program saat ini masih menggunakan versi lama dan kedepannya diharapkan dapat dilakukan upgrade guna memperbaiki kelemahan-kelemahan.