



Sistem Reservasi Klinik

30.10.2023

Kelompok 6

Kelas 1B

Sistem Informasi Bisnis

Politeknik negeri Malang

Anggota Kelompok

- Fransiska Widya Krisanti (08)
- Muhammad Reishi Fauzi A (19)
- Revani Nanda Putri (24)

1. BAB I - Pendahuluan

Di era modern, dengan kemajuan teknologi informasi, terdapat peluang besar untuk meningkatkan efisiensi dan kualitas layanan kesehatan melalui pengembangan sistem pemesanan yang canggih. Sistem booking klinik merupakan solusi yang dapat membantu mengatasi permasalahan antrian panjang, memprioritaskan pelayanan berdasarkan pentingnya kasus dan mengoptimalkan penggunaan sumber daya yang terbatas. Pada saat yang sama, penting untuk memperhatikan aspek keamanan data pasien dan integrasinya dengan sistem pengelolaan rekam medis yang ada sehingga informasi medis pasien dapat diakses dengan aman dan efektif. Selain itu, evaluasi kepuasan pasien dan efektivitas sistem merupakan bagian penting untuk memastikan bahwa solusi tersebut benar-benar memberikan manfaat yang diharapkan bagi pasien dan staf klinik.

Dengan latar belakang tersebut, penelitian ini bertujuan untuk mengembangkan dan menerapkan sistem pemesanan klinik yang inovatif, yang diharapkan dapat meningkatkan efisiensi, kualitas layanan dan kepuasan pasien. Penelitian ini juga akan mengidentifikasi dan mengatasi berbagai tantangan teknis, keamanan dan manajemen yang terlibat dalam pengembangan sistem tersebut.

Melalui pemahaman yang lebih baik terhadap permasalahan tersebut dan upaya untuk mengembangkan solusi yang tepat, diharapkan klinik dapat memberikan pelayanan medis yang lebih baik kepada masyarakat, mengurangi waktu tunggu pasien, meningkatkan aksesibilitas dan optimalisasi penggunaan sumber daya yang ada. Oleh karena itu, penelitian ini berpotensi memberikan dampak positif yang signifikan terhadap sistem kesehatan masyarakat dan kualitas hidup masyarakat di wilayah tersebut.

1.1. Latar Belakang

Pelayanan kesehatan merupakan aspek penting di dalam masyarakat modern. Klinik merupakan salah satu institusi kesehatan pemerintah yang

memberikan pelayanan kesehatan kepada masyarakat. Oleh karena itu, kami melakukan pengembangan sistem reservasi dalam klinik yang efisien dan mudah digunakan. Sistem pemesanan klinik yang kami dibuat menggunakan command line di Java merupakan reservasi jasa dokter yang dioperasikan oleh admin dengan tujuan dapat menjadi solusi yang membantu mengatasi beberapa tantangan terkait manajemen pasien, dokter, dan rekam histori di klinik. Penggunaan Java dalam implementasi sistem pemesanan ini memungkinkan pengembang untuk menciptakan solusi yang dapat mempermudah pendataan transaksi yang terjadi di klinik. Selain itu, penggunaan command line memungkinkan admin mendapatkan keuntungan berupa efisiensi dalam hal penggunaan sumber daya sistem, memiliki kecepatan yang lebih dibandingkan dengan GUI (*Graphical User Interface*), dan dapat diakses jarak jauh melalui SSH (*Secure Shell*) atau protokol serupa yang dapat berguna dalam pengelolaan server rumah sakit atau perangkat jarak jauh.

1.2. Rumusan Masalah

1. Bagaimana cara mengembangkan sistem reservasi klinik yang efisien dan mudah digunakan?
2. Bagaimana cara memastikan keamanan data pasien dalam sistem reservasi klinik?
3. Bagaimana sistem reservasi klinik dapat memprioritaskan pelayanan berdasarkan tingkat kepentingan kasus?
4. Bagaimana meningkatkan akurasi estimasi waktu tunggu untuk pasien yang melakukan reservasi?
5. Bagaimana mengintegrasikan sistem reservasi klinik dengan sistem manajemen rekam medis yang sudah ada?
6. Bagaimana mengevaluasi kepuasan pasien dan efektivitas sistem reservasi klinik setelah implementasi?
7. Bagaimana pihak klinik dapat mendata pasien yang berobat dengan lebih tertata?
8. Bagaimana membuat sistem yang mudah dipahami oleh calon user?

1.3. Manfaat

1. Peningkatan Efisiensi Administrasi: Sistem informasi dapat membantu dalam pengelolaan administrasi sehari-hari, seperti pencatatan pasien, manajemen stok obat, dan pengaturan jadwal dokter. Hal ini

dapat mengurangi waktu dan tenaga yang diperlukan untuk tugas-tugas administratif, memungkinkan staf untuk lebih fokus pada pelayanan pasien.

2. Akurasi Data Pasien: Sistem informasi medis dapat menyimpan catatan pasien elektronik yang akurat dan terstruktur. Ini membantu dalam pengelolaan rekam medis pasien dengan lebih baik, sehingga meminimalkan risiko kesalahan atau kehilangan data.
3. Peningkatan Pelayanan Pasien: Dengan akses cepat ke informasi pasien, klinik dapat memberikan layanan yang lebih cepat dan tepat sasaran. Informasi medis yang mudah diakses juga memungkinkan staf medis untuk membuat diagnosis yang lebih akurat.

2. BAB II - Identifikasi kebutuhan Sistem

2.1. Deskripsi Sistem

Sistem Reservasi klinik diciptakan untuk memenuhi kebutuhan pasien atau pengguna (staf administrasi) yang memiliki keterbatasan dalam melakukan pengumpulan, pengelolaan, aksesibilitas atau transaksi data banyak pengguna. Sistem ini dirancang dengan fitur-fitur untuk memberikan kemudahan dan kenyamanan bagi pengguna dalam mengelola sistem klinik. Sistem pemesanan klinik memadukan fungsi sistem klinik untuk menyediakan layanan kesehatan yang lebih efisien, terorganisir, dan terjangkau bagi pasien, sehingga memungkinkan staf administrasi mengelola data pasien secara efisien dan memantau aspek terkait biaya di tingkat klinik.

2.2. Kebutuhan Fungsional

2.2.1. Kebutuhan Perangkat Lunak

Software yang dibutuhkan untuk menjalankan program yang kami buat adalah sebagai berikut:

- Visual Studio Code / terminal lain seperti CMD
- Ekstensi Java di Visual Studio Code (jika menggunakan vscode)
- JRE 8

2.2.2. Kebutuhan Perangkat Keras

Perangkat keras yang diperlukan adalah laptop atau komputer dengan rekomendasi spesifikasi sebagai berikut.

- Prosesor : Intel core i3 atau lebih baik
- Sistem Operasi : Windows, Mac OS, atau Linux
- RAM : 4GB atau lebih
- Ruang Penyimpanan : 20GB atau lebih

2.3. Hasil Observasi

Hasil observasi dari kelompok terhadap klinik yang dituju adalah klinik telah menerapkan alur pelayanan kepada pasien, mulai dari datangnya pasien, pengelompokan pasien berdasar screening, pendaftaran, pemeriksaan, dan pelayanan yang diberikan terhadap pasien. Adapun di klinik tersedia bagan interaksi antar proses manajemen mutu klinik dengan pasien atau masyarakat, sehingga pasien bisa melihat sendiri alur dalam proses realisasi pelayanannya.

2.4. Hasil Wawancara



Pertanyaan

1. Bagaimana alur / proses pasien yang ingin berobat dari awal hingga akhir?
2. Bagaimana proses pemesanan kamar? Apakah pasien bisa melakukan reservasi kamar melalui admin?
3. Apa sebutan bagi petugas yang mendata data pasien Ketika ingin berobat di klinik?
4. Bagaimana proses pembayaran yang ada di klinik?
5. Data apa saja yang dibutuhkan petugas untuk mendata pasien yang baru pertama kali berobat ke klinik?
6. Data apa saja yang dibutuhkan petugas untuk mendata pasien yang sebelumnya pernah berobat ke klinik?
7. Apakah di klinik bisa menggunakan BPJS?
8. Bagaimana proses dalam sistem rujuk dalam klinik?

Jawaban

1. Berikut tahapannya:
 - a. jika pasien sudah pernah berobat ke klinik tersebut, pasien cukup menyerahkan kartu berobat ke petugas pendaftaran.
 - b. jika pasien baru pertama kali berobat ke klinik tersebut, petugas meminta kartu identitas pasien, seperti ktp,sim/kk.
 - c. kemudian petugas mendaftarkan pasien.
 - d. jika pasien lama petugas RM mencarikan berkas RM, jika pasien baru dibuatkan RM baru.
 - e. petugas RM mendistribusikan RM ke unit yang dituju.
 - f. pasien diperiksa dan ditangani oleh petugas yang jaga.
 - g. pasien menunggu antrian obat.
 - h. petugas apotek menyiapkan obat sesuai resep dari dokter.
 - i. proses pembayaran pada kasir.
 - j. petugas apotek memberikan KIE cara penggunaan obat kepada pasien sesuai anjuran dari dokter.
2. Berikut tahapannya:
 - a. keluarga pasien mendaftarkan ke bagian pendaftaran
 - b. petugas pendaftaran meminta kartu bpjs jika ada, jika tidak ada pengecekan melalui nik di kartu ktp pasien.
 - c. pada klinik kami bisa melakukan reservasi jika kamar masih tersedia.
3. TPP (Tempat Pendaftaran Pasien) / PAT (Pelayanan Administrasi Terpadu)
4. Proses pembayaran pada klinik melalui 2 cara yakni bisa membayar dengan cash atau dengan asuransi kesehatan (KIS/KJN)

5. Data yang dibutuhkan petugas untuk mendata pasien sesuai dengan kartu identitas, meliputi: Nama lengkap, TTL, Alamat, NIK.
6. Kartu berobat dan kartu bpjs jika pasien menggunakan Bpjs.
7. Berikut tahapannya:
 - a. bisa jika untuk pasien rawat jalan harus faskes dari klinik yang bersangkutan.
 - b. jika rawat inap bisa menggunakan bpjs jika faskes bukan dari klinik yang bersangkutan tetapi hanya dalam 1 kali.
8. Berikut tahapannya:
 - a. petugas perawat mengkonsulkan kondisi pasien.
 - b. petugas mencari rujukan ke rumah sakit yang bisa menerima pasien.
 - c. jika di acc pasien diantarkan ke rumah sakit yang menerima pasien tersebut.

2.5. Identifikasi Sistem Yang Sudah Ada

Sistem Reservasi yang sudah kami buat selama ini mencakup:

1. Login multi pengguna / user
User dapat menggunakan akun yang berbeda untuk login ke dalam sistem.
2. Memilih beberapa menu yang ingin digunakan user
User dapat memilih menu yang tersedia dalam sistem, sesuai dengan kebutuhan yang dibutuhkan.
3. Membuat master data
User dapat membuat master data berupa mendaftarkan pasien yang belum pernah berobat dengan mendaftarkan pasien di form daftar kartu berobat.
4. Melakukan transaksi pemesanan dokter yang ada di klinik
User dapat memproses transaksi reservasi dokter yang dibutuhkan oleh pasien.
5. Melakukan pembayaran
Pasien dapat menentukan metode pembayaran yang ingin dilakukan.
6. Membuat cetak Struk Transaksi

3. BAB III

3.1. Sistem Versi Alfa

1. Input pemesanan yang di booking
2. Input data master
3. Pembayaran
4. Laporan harian dan bulanan
5. Multi pengguna
6. Cetak Struk booking
7. cetak stok barang / jasa yang diboeking

3.2. Sistem Versi Beta

Fitur sistem komputer untuk staf administrasi klinik adalah kunci untuk mengelola data dan operasi sehari-hari klinik dengan efisien. Berikut adalah beberapa fitur yang penting dalam komputer staf administrasi klinik:

3.2.1. Manajemen Data Pasien:


- A. Pendaftaran Pasien: Memungkinkan staf administrasi untuk mendaftarkan pasien baru dengan mengumpulkan informasi pribadi dan medis dasar.
- B. Riwayat Pasien: Menyimpan dan mengakses riwayat lengkap pasien, termasuk catatan kunjungan sebelumnya, dan transaksi yang telah dilakukan dengan keluhan yang dirasakan.
- C. Pembayaran Pasien: Mengelola pembayaran pasien, termasuk penagihan, penyelesaian klaim asuransi BPJS, dan perhitungan biaya yang terkait dengan layanan medis.

3.2.2. Pemantauan dan Manajemen Biaya:

- A. Pembayaran Pasien: Mengelola pembayaran pasien, termasuk penagihan, penyelesaian klaim asuransi BPJS, dan perhitungan biaya yang terkait dengan layanan medis.
- B. Manajemen Keuangan: Memungkinkan staf untuk melacak pendapatan dan pengeluaran rumah sakit, membuat laporan keuangan, dan mengelola anggaran.

3.2.3. Manajemen Antrian

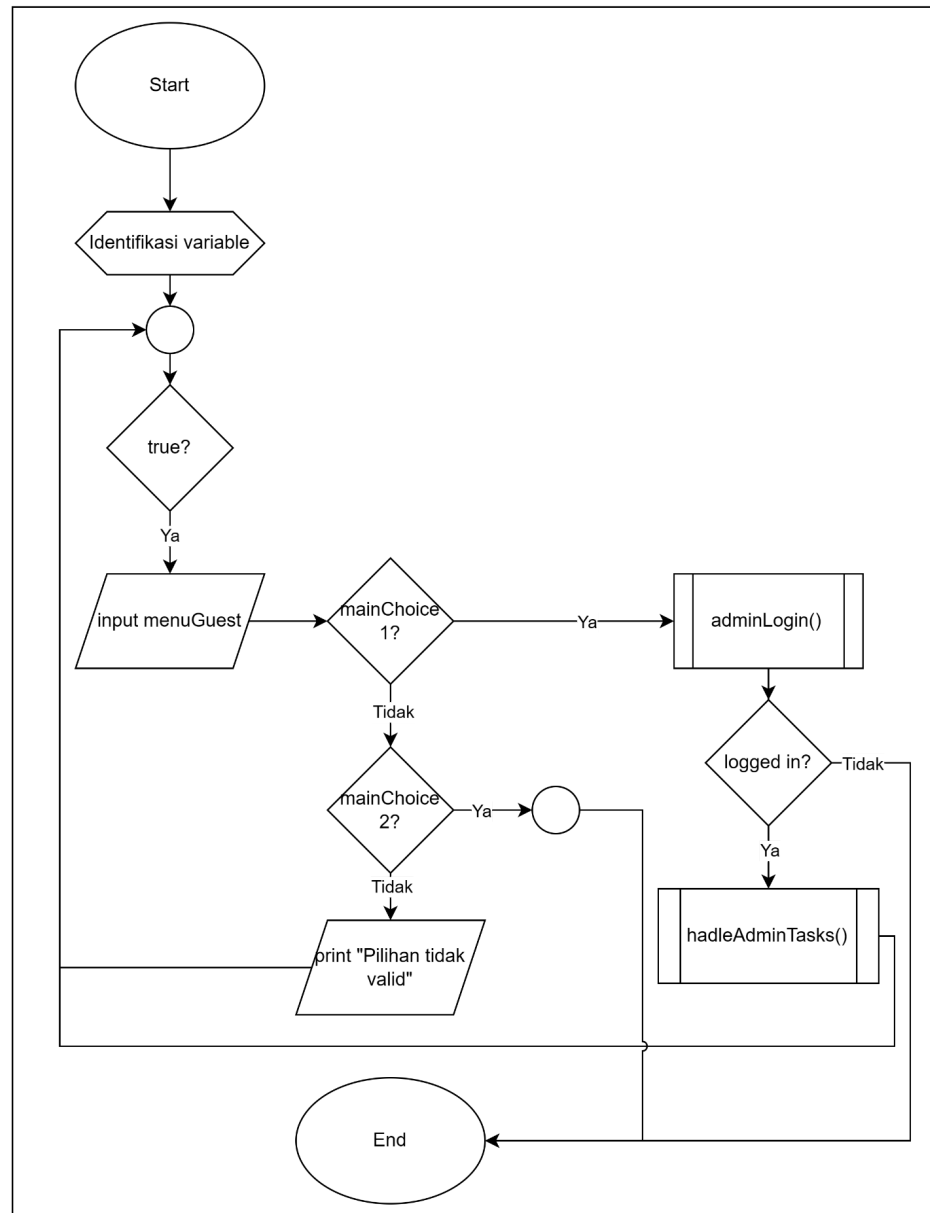
- A. Membantu dalam mengelola antrian dan memberikan informasi kepada pasien atau keluarga pasien nomor antrian yang sedang berlangsung.

- 
- 3.2.4. Aksesibilitas Multi-Level
 - A. Memungkinkan beberapa staff untuk mengakses sistem secara bersamaan dengan tingkat akses yang sesuai dengan peran dan tanggung jawab mereka
 - 3.2.5. Management Laporan
 - A. Laporan berupa struk yang dapat dicetak dan sebagai bukti transaksi
 - B. Laporan Bulanan berupa rekap data transaksi yang terjadi dalam periode hitungan bulan
 - C. Laporan Harian berupa rekap data transaksi yang terjadi dalam satu hari
 - 3.2.6. Management stok
 - A. Laporan berupa stok / ketersediaan jasa dalam sistem klinik
 - 3.2.7. Fitur Pencarian
 - A. Fitur pencarian yang dapat digunakan untuk mencari data yang dibutuhkan dalam sistem

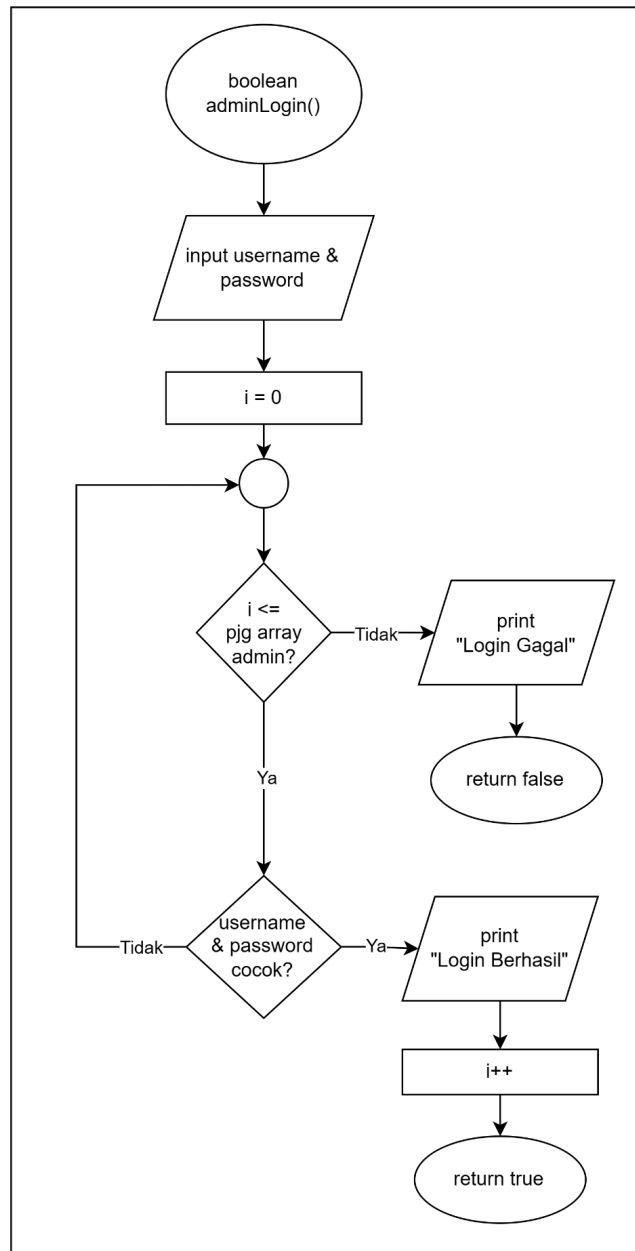
4. BAB IV

4.1. Flowchart Sistem

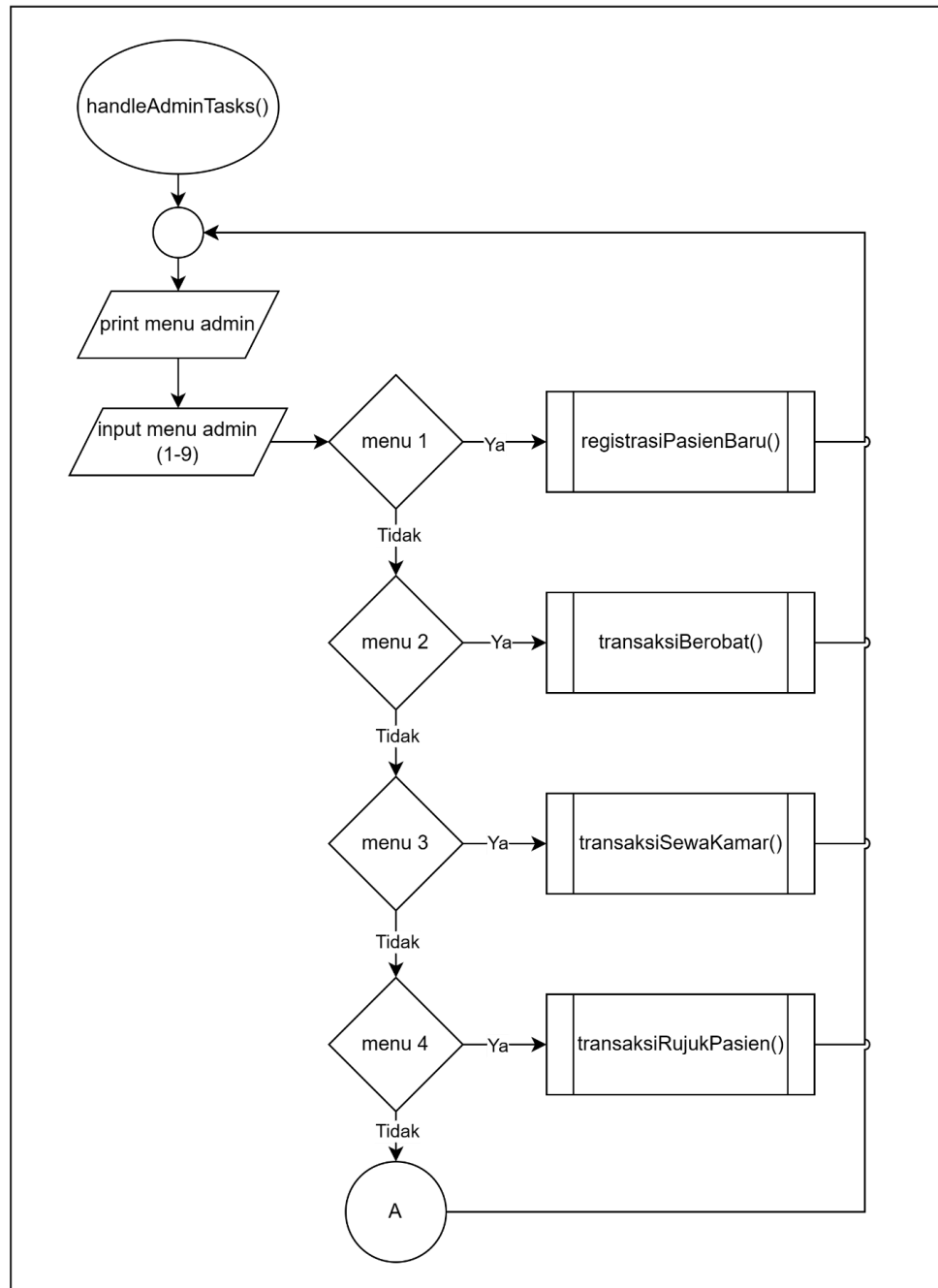
Function Main()

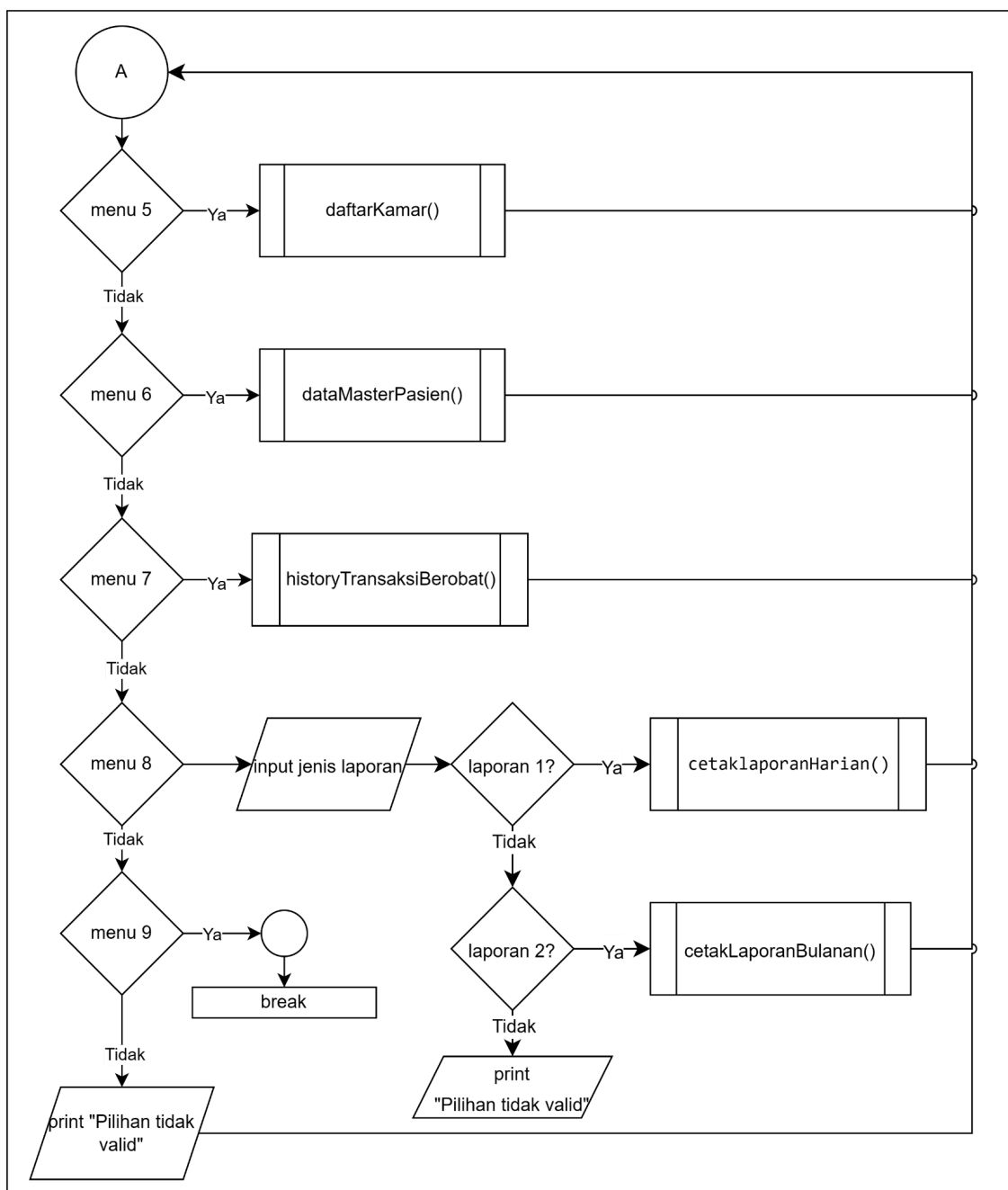


Function boolean adminLogin()

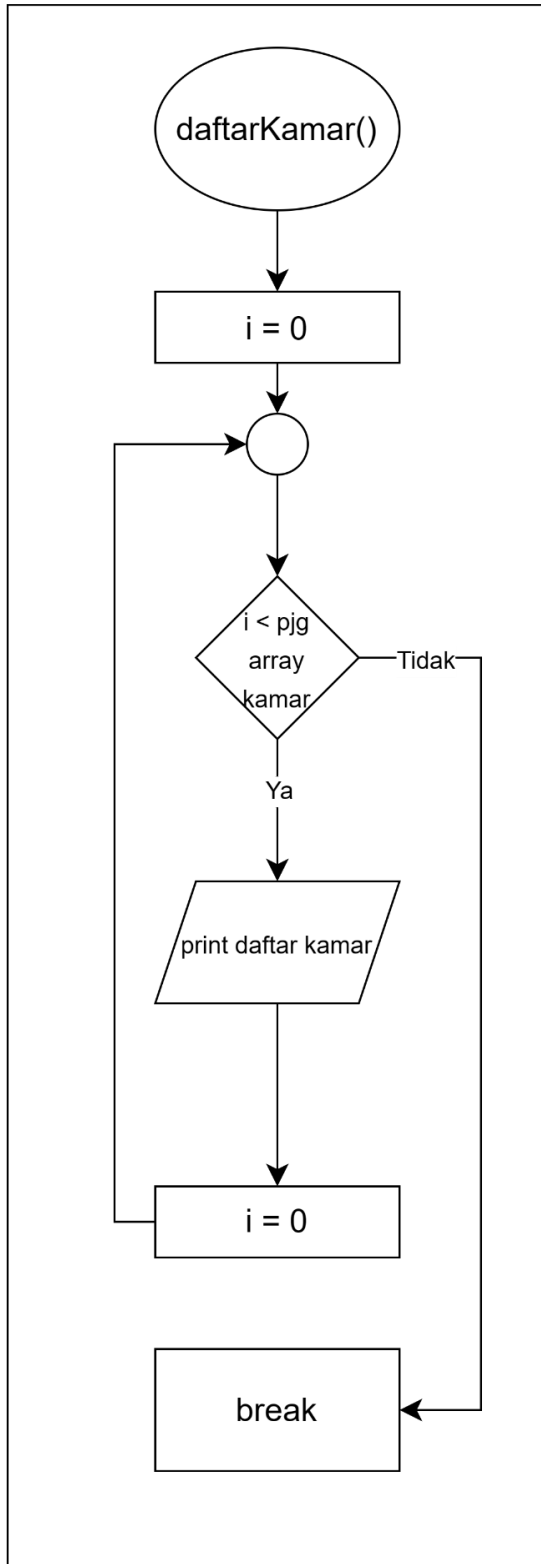


Function handleAdminTasks()

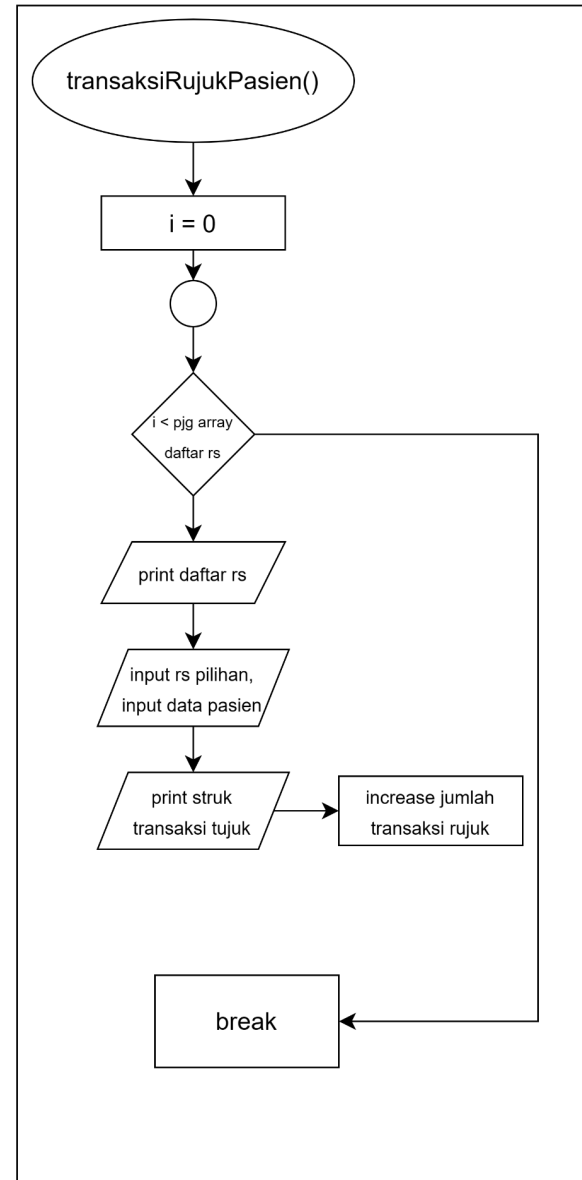




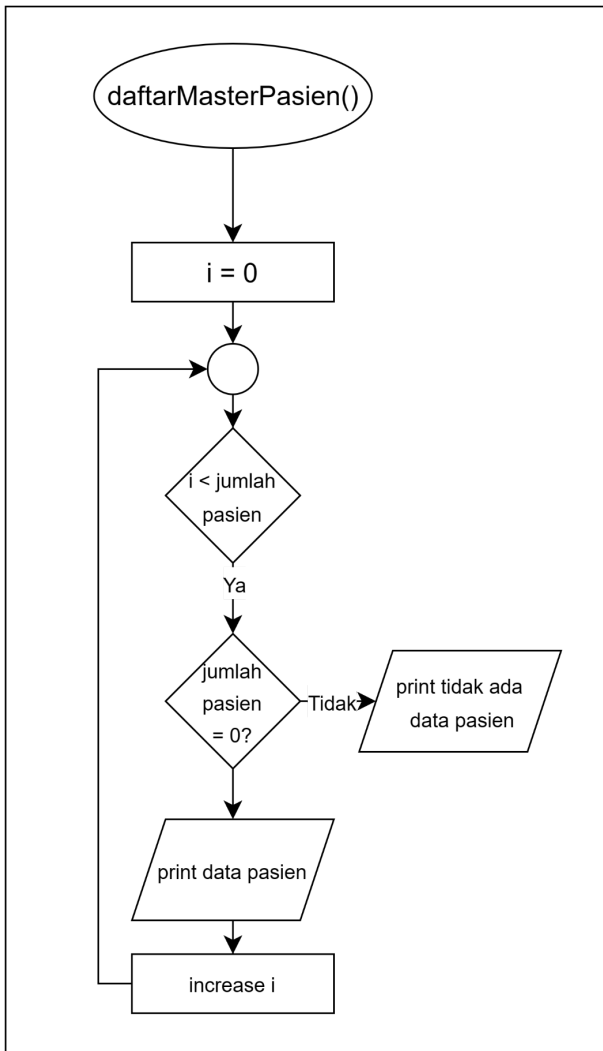
Function daftarKamar()



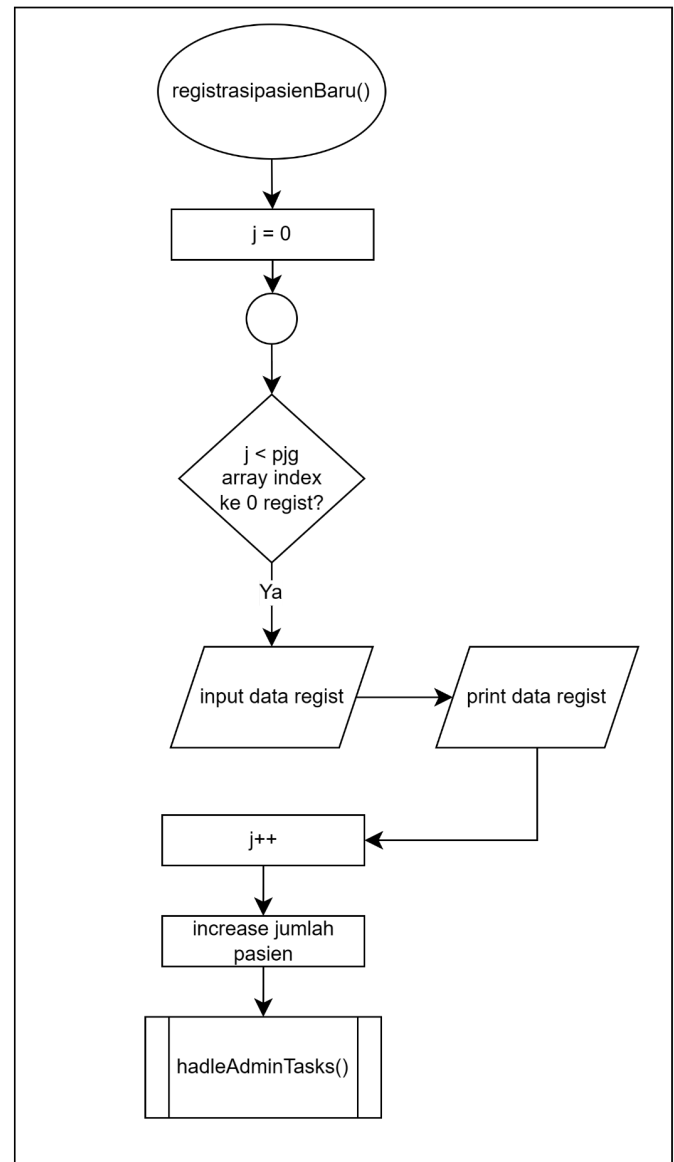
Function transaksiRujukPasien()



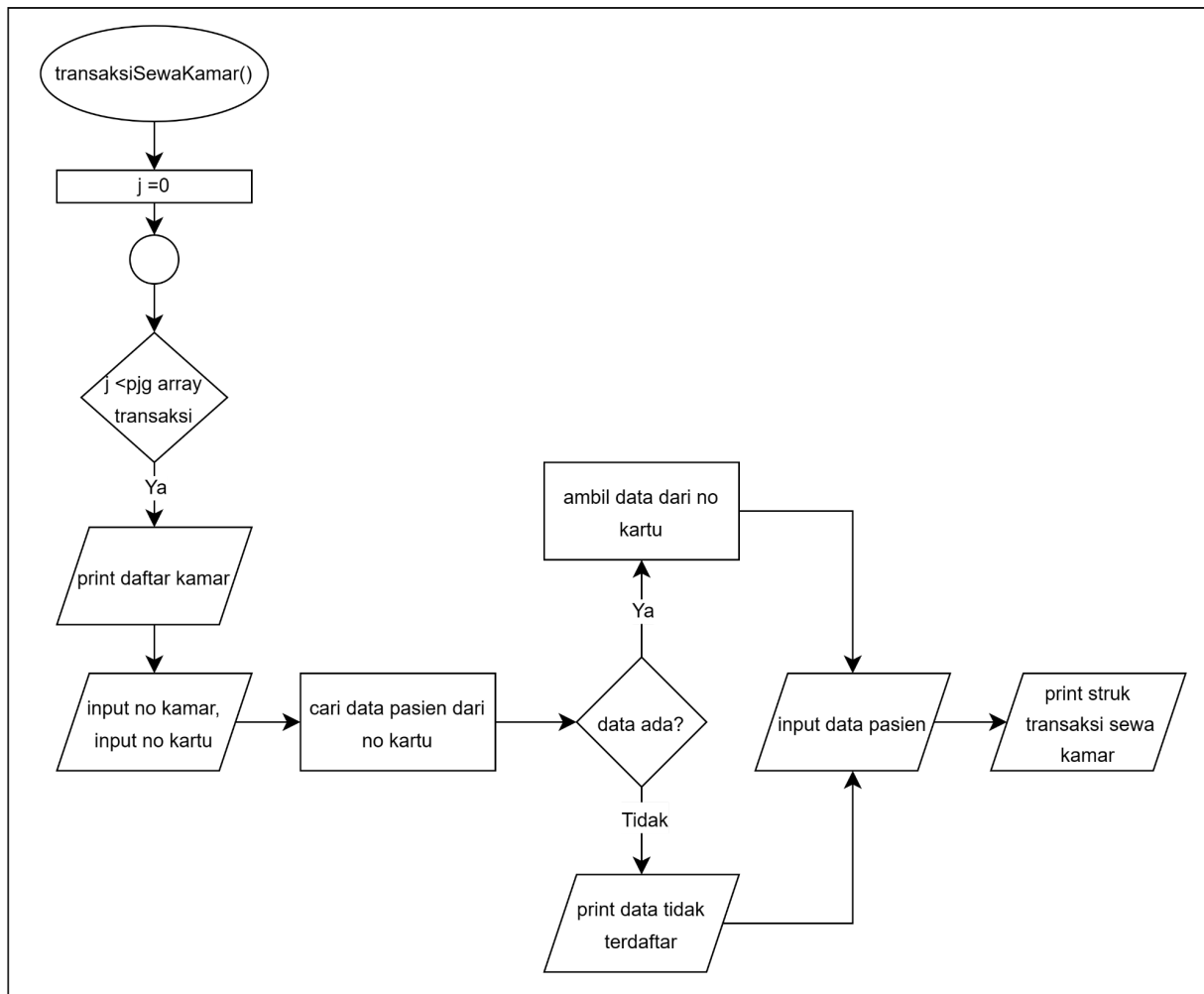
Function daftarMasterPasien()



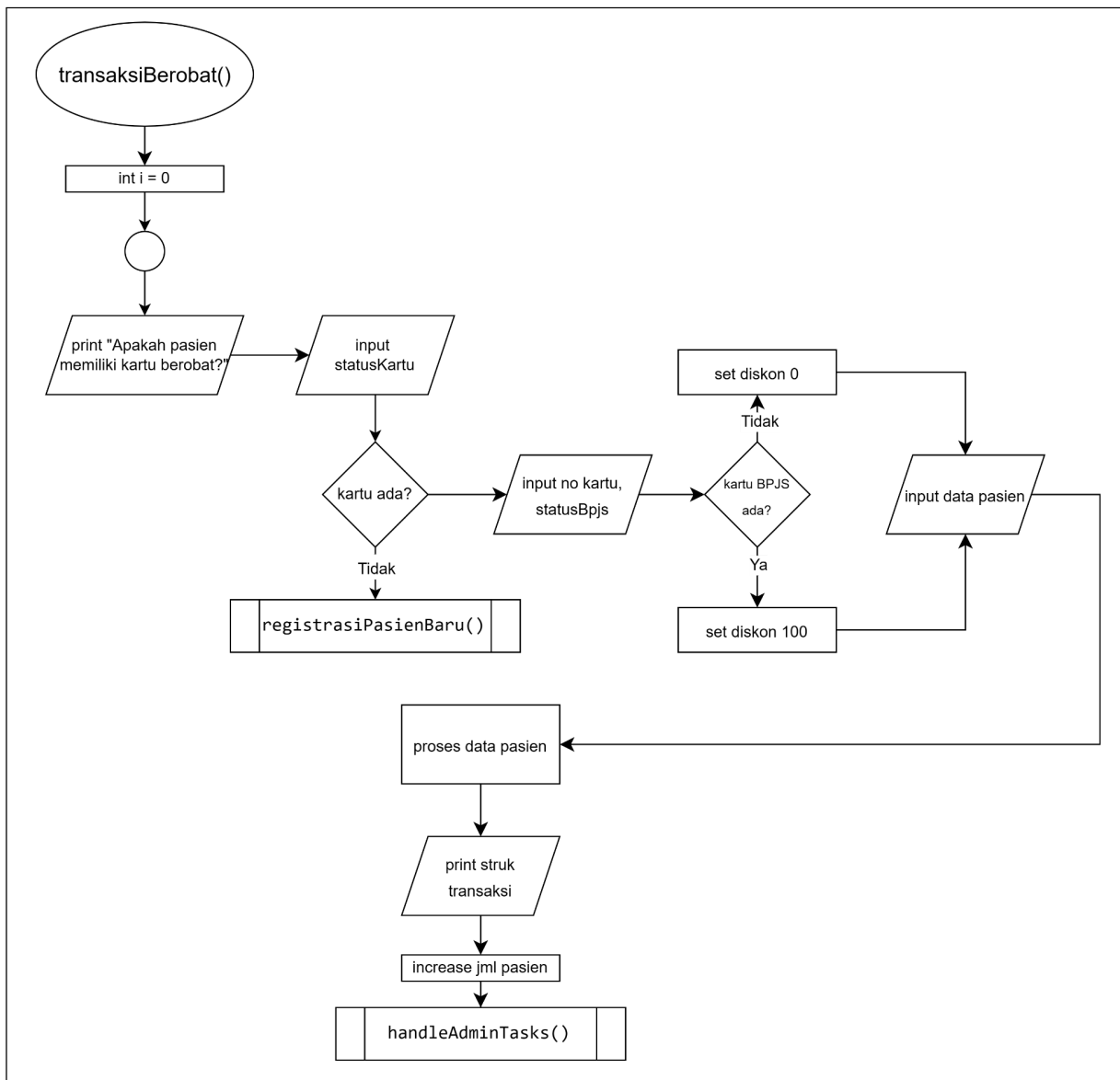
Function registrasiPasienBaru()



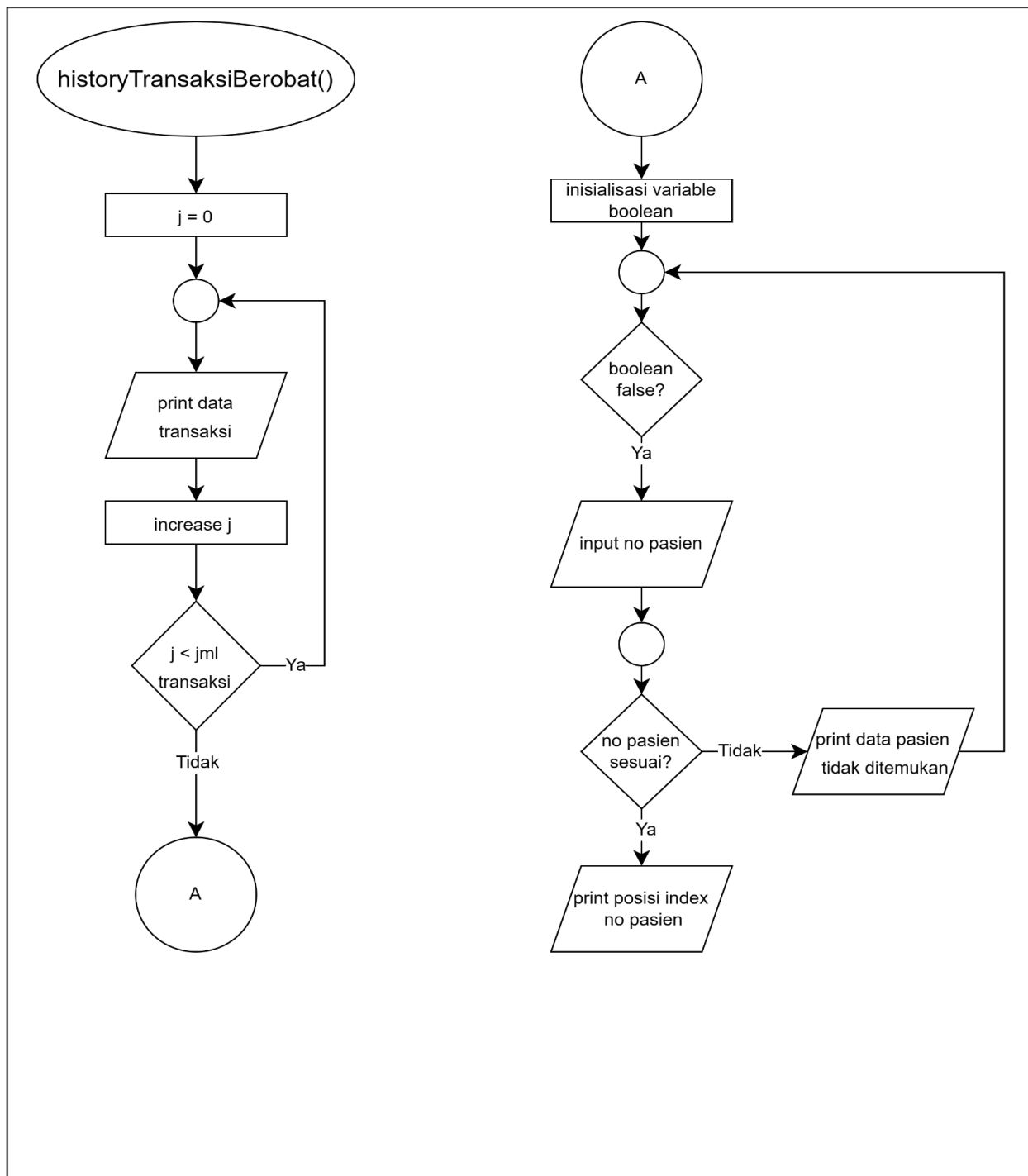
Function transaksiSewaKamar()



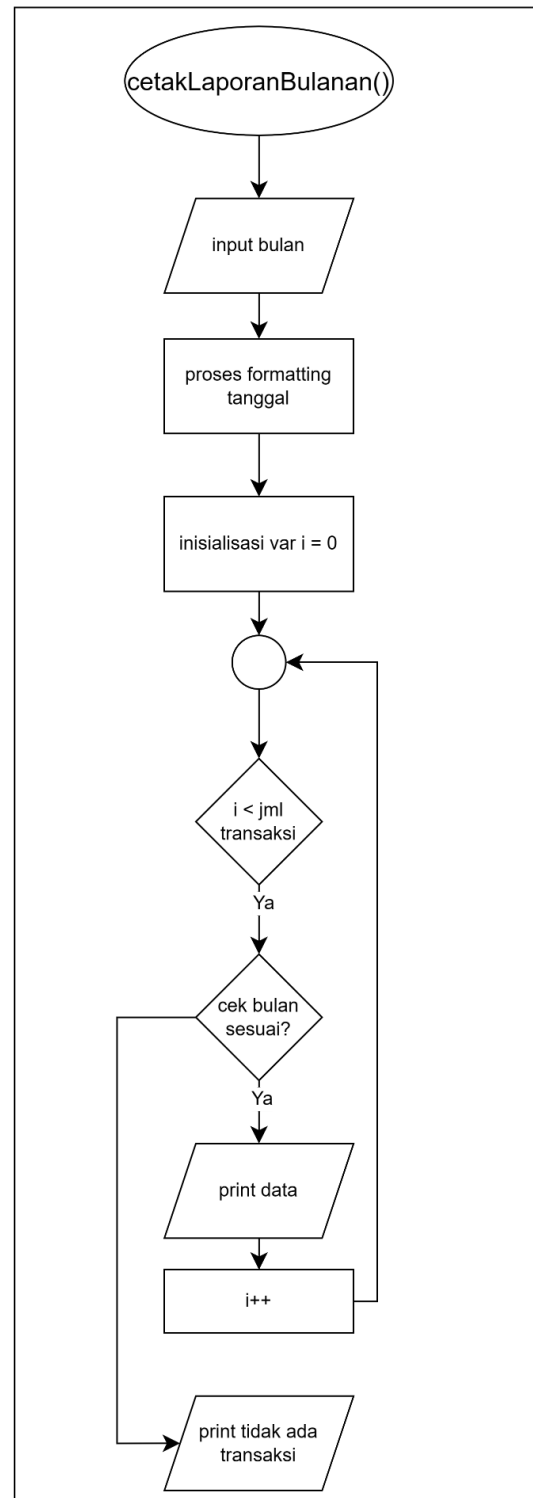
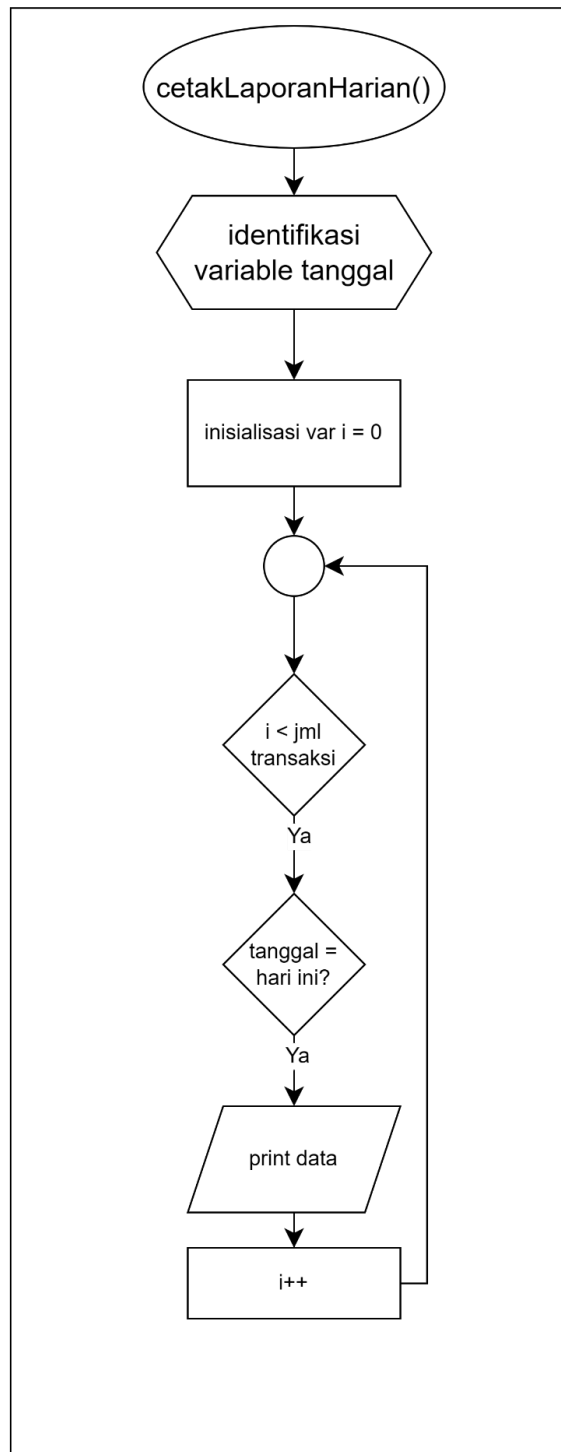
Function transaksiBerobat()



Function historyTransaksiBerobat()



Function cetakLaporanHarian() Function cetakLaporanBulanan()



A. Input booking kamar, transaksi berobat, transaksi rujuk

- B. Input data master pasien
- C. Pembayaran
- D. Cetak laporan harian dan bulanan
- E. Multi pengguna
- F. Cetak struk transaksi
- G. Cek ketersediaan kamar
- H. Fitur pencarian
- I. Melacak histori transaksi
- J. Cek ketersediaan kamar yang akan di booking
- K. Fitur kode otomatis

A. Multi-level

- B. Fitur pembayaran pihak ke 3
C. Fitur diskon
D. Bilingual

```
public static void main(String[] args) {
    while (true) {
        System.out.println(x:"=====");
        System.out.println(x:"| Selamat datang di Sistem Reservasi Klinik |");
        System.out.println(x:"=====");
        System.out.println(x:"| 1. Admin |");
        System.out.println(x:"| 2. Keluar |");
        System.out.println(x:"=====");
        System.out.print(s:"Pilih tipe pengguna (1-2): ");
        int mainChoice = input.nextInt();
        input.nextLine();
    }
}
```

```
=====
| Selamat datang di Sistem Reservasi Klinik
=====
| 1. Admin
| 2. Keluar
=====
Pilih tipe pengguna (1-2): 1
```

4.3.2. Kemudian user memasukkan username dan password

```
static boolean adminLogin() {
    System.out.println(x:"=====");
    System.out.print(s:"Masukkan username admin: ");
    inputUsername = input.nextLine();
    System.out.print(s:"Masukkan password admin: ");
    String inputPassword = input.nextLine();
    System.out.println(x:"=====");

    for (int i = 0; i < usernameAdmin.length; i++) {
        if (inputUsername.equals(usernameAdmin[i]) && inputPassword.equals(passwordAdmin[i])) {
            System.out.println(x:"LOGIN BERHASIL!");
            return true;
        }
    }
    System.out.println(x:"Login gagal. Username atau password salah.");
    return false;
}
```

```
Pilih tipe pengguna (1-2): 1
=====
Masukkan username admin: admin1
Masukkan password admin: adminsatu
=====
LOGIN BERHASIL!
```

4.3.3. Setelah login berhasil, user akan masuk ke menu admin dan dapat memilih menu yang dibutuhkan

```
static void handleAdminTasks() {
    while (true) {
        System.out.println(x:"=====");
        System.out.println(x:"| Selamat datang di menu admin |");
        System.out.println(x:"=====");
        System.out.println(x:"| 1. Registrasi Pasien Baru |");
        System.out.println(x:"| 2. Transaksi Berobat |");
        System.out.println(x:"| 3. Transaksi Sewa Kamar |");
        System.out.println(x:"| 4. Transaksi Rujuk Pasien |");
        System.out.println(x:"| 5. Daftar Kamar |");
        System.out.println(x:"| 6. Daftar Pasien |");
        System.out.println(x:"| 7. History Transaksi Berobat |");
        System.out.println(x:"| 8. Cetak Laporan Transaksi Berobat |");
        System.out.println(x:"| 9. Keluar |");
        System.out.println(x:"=====");
        System.out.print(s:"Pilih menu (1-9): ");
        int choice = input.nextInt();
        input.nextLine();

        switch (choice) {
            case 1:
                registrasiPasienBaru();
                break;
            case 2:
                transaksiBerobat();
                break;
            case 3:
                transaksiSewaKamar();
                break;
            case 4:
                transaksiRujukPasien();
                break;
        }
    }
}
```

```
case 5:
    daftarKamar();
    break;
case 6:
    dataMasterPasien();
    break;
case 7:
    historyTransaksiBerobat();
    break;
case 8:
    System.out.println(x:"Pilih Jenis Laporan: ");
    System.out.println(x:"1. Laporan Harian");
    System.out.println(x:"2. Laporan Bulanan");
    System.out.print(s:"Masukkan Pilihan (1/2): ");
    int pilihLaporan = input.nextInt();
    if (pilihLaporan == 1) {
        cetakLaporanHarian();
    } else if (pilihLaporan == 2) {
        cetakLaporanBulanan();
    } else {
        System.out.println(x:"Inputan Tidak Valid");
    }
    break;
case 9:
    break;
default:
    System.out.println(x:"Pilihan tidak valid.");
    break;
}
```

```
=====
| Selamat datang di menu admin |
=====
| 1. Registrasi Pasien Baru |
| 2. Transaksi Berobat |
| 3. Transaksi Sewa Kamar |
| 4. Transaksi Rujuk Pasien |
| 5. Daftar Kamar |
| 6. Daftar Pasien |
| 7. History Transaksi Berobat |
| 8. Cetak Laporan Transaksi Berobat |
| 9. Keluar |
=====
Pilih menu (1-9): █
```

4.3.4. User dapat melakukan registrasi pasien

```
static void registrasiPasienBaru() {
    System.out.println(x:"");
    System.out.println(x:"");
    System.out.println(x:"KLINIK X - REGISTRASI PASIEN");
    if (jumlahPasien >= registrasiPasien[0].length) {
        return;
    }
    String noKartu = "P" + String.format(format:"%04d", idOtomatis);
    System.out.println(x:"=====");
    System.out.println("No Kartu : " + noKartu);
    registrasiPasien[0][jumlahPasien] = noKartu;
    System.out.print(s:"Nama : ");
    registrasiPasien[1][jumlahPasien] = sc1.nextLine();
    System.out.print(s:"No KTP : ");
    registrasiPasien[2][jumlahPasien] = sc1.nextLine();
    System.out.print(s:"Alamat : ");
    registrasiPasien[3][jumlahPasien] = sc1.nextLine();
    System.out.print(s:"No. Telp : ");
    registrasiPasien[4][jumlahPasien] = sc1.nextLine();
    System.out.println(x:"=====");
    System.out.println(x:"DATA BERHASIL DISIMPAN!");
    System.out.println("No kartu : " + registrasiPasien[0][jumlahPasien]);
    System.out.println("Nama : " + registrasiPasien[1][jumlahPasien]);
    System.out.println("No KTP : " + registrasiPasien[2][jumlahPasien]);
    System.out.println("Alamat : " + registrasiPasien[3][jumlahPasien]);
    System.out.println("No. Telp : " + registrasiPasien[4][jumlahPasien]);
    System.out.println(
        x:"=====");
    jumlahPasien++;
}
```

KLINIK X - REGISTRASI PASIEN

```
=====
No Kartu : P0011
Nama : reva
No KTP : 237823284
Alamat : malang
No. Telp : 0812378
```

DATA BERHASIL DISIMPAN!

```
-----
No kartu : P0011
Nama : reva
No KTP : 237823284
Alamat : malang
No. Telp : 0812378
=====
```

4.3.5. User dapat melakukan transaksi berobat

```

static void transaksiBerobat() {
    for (int i = 0; i < transaksiPasien.length; i++) {
        LocalDate currentDate = LocalDate.now();
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern(pattern: "dd-MM-yyyy");
        String formattedDate = currentDate.format(formatter);
        transaksiPasien[i][jumlahTransaksi] = currentDate;
        System.out.println(s: "");
        System.out.println(s: "");
        System.out.println(s: "KLINIK X - TRANSAKSI BEROBAT");
        System.out.println(s: "-----");
        System.out.println("ADMIN : " + inputUsername);
        System.out.println("tgl : " + formattedDate);
        System.out.println(s: "");
        System.out.println(s: "Apakah pasien memiliki kartu berobat? (0 = tidak, 1 = ya)");
        status = input.next().charAt(index: 0);

        if (status == '0') {
            System.out.println(s: "Silahkan lakukan registrasi pasien baru");
            registrasiPasienBaru();
        } else {
            System.out.println(s: "Masukkan No. Kartu pasien");
            System.out.print(s: "No Kartu : ");
            transaksiPasien[0][jumlahTransaksi] = sc.nextLine();
            // lanjut ke form transaksi

            // penambahan fitur pasien BPJS
            System.out.println(s: "Apakah pasien memiliki kartu BPJS (0 = tidak, 1 = ya)? ");
            statusbpjs = input.next().charAt(index: 0);

            if (statusbpjs == '0') {
                transaksiPasien[3][jumlahTransaksi] = 0;
                transaksiPasien[4][jumlahTransaksi] = "-";
            } else {
                System.out.print(s: "Masukkan No. BPJS pasien : ");
                transaksiPasien[1][jumlahTransaksi] = sc.nextLine();
                transaksiPasien[3][jumlahTransaksi] = 100;
            }
            System.out.println("Nama : " + registrasiPasien[1][1]);
            transaksiPasien[2][jumlahTransaksi] = registrasiPasien[1][1];

            // usia menggunakan ifelse menentukan biaya konsultasi atau berobat
            System.out.print(s: "Usia : ");
            transaksiPasien[0][jumlahTransaksi] = input.nextInt();
            System.out.print(s: "Keluhan : ");
            transaksiPasien[3][jumlahTransaksi] = sc.nextLine();
        }
    }

    do {
        System.out.println(s: "-----");
        System.out.println(s: "No. | Daftar Dokter |");
        System.out.println(s: "-----");
        for (int j = 0; j < daftarDokter.length; j++) {
            System.out.printf(format: "%-3d | %-65s |%n", (j + 1), daftarDokter[j]);
        }
        System.out.println(s: "-----");
        System.out.println(s: "Pilih nomor dokter yang dibutuhkan pasien (1 - 6): ");
        int pemilihan = sc.nextInt();

        if (pemilihan >= 1 && pemilihan <= daftarDokter.length) {
            String dokterPilihan = daftarDokter[pemilihan - 1];
            System.out.println("Anda telah memilih dokter : " + dokterPilihan);
            transaksiPasien[0][jumlahTransaksi] = dokterPilihan;
            System.out.println("Dengan Keluhan : " + transaksiPasien[3][jumlahTransaksi]);
            isDokter = true;
            break;
        } else {
            // User input is not within the valid range
            System.out.println(s: "Nomor dokter tidak valid. Silakan coba lagi.");
            continue;
        }
    } while (!isDokter);

    // potongan BPJS pada klinik 100%
    System.out.println("Potongan BPJS(X) : " + transaksiPasien[3][jumlahTransaksi]);
    // harga di tentukan sesuai kategori usia pasien
    System.out.print(s: "Harga : Rp. ");
    transaksiPasien[2][jumlahTransaksi] = sc.nextInt();

    // memproses perhitungan pembayaran
    transaksiPasien[2][jumlahTransaksi] = transaksiPasien[1][jumlahTransaksi]
        - (transaksiPasien[1][jumlahTransaksi] * transaksiPasien[3][jumlahTransaksi] / 100);
    System.out.println(
        s: "-----");
    System.out.println("Harga Akhir : Rp. " + transaksiPasien[2][jumlahTransaksi]);
    if (transaksiPasien[2][jumlahTransaksi] == 0) {
        System.out.println(s: "Pembayaran Gratis");
        transaksiPasien[5][jumlahTransaksi] = "BPJS";
        System.out.println(
            s: "-----");
        // jumlahTransaksi++;
    } else {
        // pembayaran debit atau tunai by fauzi
        System.out.print(s: "Metode pembayaran menggunakan(1 = debit, 2 = tunai)? ");
        pembayaran = sc.next().charAt(index: 0);
        if (pembayaran == '1') {
            System.out.print(s: "Masukan No kartu debit:");
            transaksiPasien[4][jumlahTransaksi] = input.nextLine();
            transaksiPasien[6][jumlahTransaksi] = "Debit";
        }
    }
}

```



```

System.out.println("Harga Akhir          : Rp. " + transaksiPasien2[2][jumlahTransaksi]);
if (transaksiPasien2[2][jumlahTransaksi] == 0) {
    System.out.println(x:"Pembayaran Gratis");
    transaksiPasien[5][jumlahTransaksi] = "BPJS";
    System.out.println(
        x:"=====");
    // jumlahTransaksi++;
} else {
    // pembayaran debit atau tunai by fauzi
    System.out.print(s:"Metode pembayaran menggunakan(1 = debit, 2 = tunai)? ");
    pembayaran = sc.next().charAt(index:0);
    if (pembayaran == '1') {
        System.out.print(s:"Masukan No kartu debit:");
        transaksiPasien[4][jumlahTransaksi] = input.nextLine();
        transaksiPasien[5][jumlahTransaksi] = "Debit";
        System.out.println(x:"Pembayaran menggunakan Debit - Lunas");
    } else {
        transaksiPasien[5][jumlahTransaksi] = "Tunai";
        transaksiPasien[4][jumlahTransaksi] = "-";
        System.out.println(x:"Pembayaran menggunakan uang tunai - Lunas");
    }
}
System.out.println(x:"=====");
System.out.println(x:"|                               Struk Transaksi Berobat ke Klinik X                               |");
System.out.println(x:"=====");
System.out.printf(format:"| %-20s: %-48s |%n", ...args:"Tanggal", formattedDate);
System.out.printf(format:"| %-20s: %-48s |%n", ...args:"Metode Pembayaran", transaksiPasien[5][jumlahTransaksi]);
System.out.printf(format:"| %-20s: %-48s |%n", ...args:"Konsultasi", transaksiPasien[6][jumlahTransaksi]);
System.out.printf(format:"| %-20s: %-48s |%n", ...args:"No Kartu Berobat", transaksiPasien[0][jumlahTransaksi]);
System.out.printf(format:"| %-20s: %-48s |%n", ...args:"No BPJS", transaksiPasien[1][jumlahTransaksi]);
System.out.printf(format:"| %-20s: %-48s |%n", ...args:"Nama Pasien", transaksiPasien[2][jumlahTransaksi]);
System.out.printf(format:"| %-20s: %-48s |%n", ...args:"Usia Pasien", transaksiPasien2[0][jumlahTransaksi]);
System.out.printf(format:"| %-20s: %-48s |%n", ...args:"Keluhan Pasien", transaksiPasien[3][jumlahTransaksi]);
System.out.println(x:"-----");
System.out.printf(format:"| %-20s: %-48s |%n", ...args:"Potongan BPJS", transaksiPasien2[3][jumlahTransaksi]);
System.out.printf(format:"| %-20s: %-48s |%n", ...args:"Harga", transaksiPasien2[1][jumlahTransaksi]);
System.out.println(x:"-----");
System.out.printf(format:"| %-20s: %-48s |%n", ...args:"TOTAL", transaksiPasien2[2][jumlahTransaksi]);
System.out.println(x:"=====");
System.out.println(x:"");
jumlahTransaksi++;
handleAdminTasks();

```

KLINIK X - TRANSAKSI BEROBAT

```

=====
ADMIN    : admin1
Tgl      : 12-12-2023

Apakah pasien memiliki kartu berobat? (0 = tidak, 1 = ya)
1
Masukkan No. Kartu pasien
No Kartu          : P0011
Apakah pasien memiliki kartu BPJS (0 = tidak, 1 = ya)?
0
Nama              : reva
Usia              : 19
Keluhan           : flu
=====
| No. |  Daftar Dokter  |
=====
| 1 |  Dokter Umum   |
| 2 |  Dokter Gigi   |
| 3 |  Bidan         |
=====
Pilih nomor dokter yang dibutuhkan pasien (1 - 6):
1
Anda telah memilih dokter    : Dokter Umum
Dengan Keluhan              : flu
Potongan BPJS(%)           :      0
Harga                      : Rp. 12000
=====
Harga Akhir                : Rp. 12000
Metode pembayaran menggunakan(1 = debit, 2 = tunai)? 2
Pembayaran menggunakan uang tunai - Lunas

```

Struk Transaksi Berobat ke Klinik X	
Tanggal	: 12-12-2023
Metode Pembayaran	: Tunai
Konsultasi	: Dokter Umum
No Kartu Berobat	: P0011
No BPJS	: -
Nama Pasien	: reva
Usia Pasien	: 19
Keluhan Pasien	: flu
Potongan BPJS	: 0
Harga	: 12000
TOTAL	: 12000

4.3.6. User dapat melakukan transaksi sewa kamar

```
static void transaksiSewaKamar() {
    for (int j = 0; j < transaksiSewaKamar.length; j++) {
        System.out.println(x:"");
        System.out.println(x:"");
        System.out.println(x:"KLINIK X - TRANSAKSI SEWA KAMAR");
        System.out.println(x:"=====");
        System.out.printf(format:"| %-10s | %-20s |%n", ...args:"No Kamar", "Status");
        System.out.println(x:"=====");
        for (int i = 0; i < statusKamar.length; i++) {
            System.out.printf(format:"| %-10d | %-20s |%n", (i + 1), (statusKamar[i] ? "Tersedia" : "Tidak Tersedia"));
        }
        System.out.println(x:"=====");

        boolean isAvailable = false;
        // Looping untuk validasi ketersediaan kamar
        while (isAvailable != true) {
            System.out.print(s:"Masukkan nomor kamar yang ingin Anda sewa: ");
            int nomorKamar = sc.nextInt();
            if (nomorKamar > 0 && nomorKamar <= statusKamar.length) {
                if (statusKamar[nomorKamar - 1]) {
                    statusKamar[nomorKamar - 1] = false;
                    System.out.println("Anda memilih Kamar no " + nomorKamar);
                    transaksiSewaKamar2[0][jumlahTransaksiSewa] = nomorKamar;
                    isAvailable = true;
                    break;
                }
            } else {
                System.out.println(x:"Inputan tidak valid");
                isAvailable = false;
            }
        }
        boolean dataDitemukan = false;

        // Looping untuk validasi data pasien ditemukan
        while (!dataDitemukan) {
            System.out.print(s:"No Kartu Pasien: ");
            String cariIndex = sc.next();

            // Looping untuk mencari index pasien
            for (int i = 0; i < transaksiPasien[0].length; i++) {
                if (cariIndex.equalsIgnoreCase(transaksiPasien[0][i])) {
                    hasil = i;

                    System.out.printf(format:"No Kartu Pasien : %s%n", transaksiPasien[0][hasil]);
                    transaksiSewaKamar[0][jumlahTransaksiSewa] = transaksiPasien[0][hasil];
                    System.out.printf(format:"Nama Pasien : %s%n", transaksiPasien[2][hasil]);
                    transaksiSewaKamar[1][jumlahTransaksiSewa] = transaksiPasien[2][hasil];
                    System.out.printf(format:"Usia Pasien : %s%n", transaksiPasien[2][hasil]);
                }
            }
        }
    }
}
```

```

    }

    if (!dataDitemukan) {
        System.out.println(x:"Data Pasien tidak ditemukan. Silakan lakukan transaksi berobat terlebih dahulu");
        continue;
    }

    System.out.println(x:"Apakah pasien memiliki kartu BPJS (0 = tidak, 1 = ya)? ");
    statusbpjs = input.next().charAt(index:0);

    if (statusbpjs == '0') {
        transaksiSewaKamar[3][jumlahTransaksiSewa] = "-";
        transaksiSewaKamar2[5][jumlahTransaksiSewa] = 0;
    } else {
        System.out.print(s:"Masukkan No.BPJS pasien      : ");
        transaksiPasien[1][jumlahTransaksiSewa] = sc1.nextLine();
        transaksiSewaKamar2[5][jumlahTransaksiSewa] = 100;
    }

    System.out.print(s:"Masukkan lama menginap (hari) : ");
    transaksiSewaKamar2[2][jumlahTransaksiSewa] = sc.nextInt();
    System.out.println("Harga kamar/malam      : " + hargaKamar);
    transaksiSewaKamar2[3][jumlahTransaksiSewa] = hargaKamar;
    transaksiSewaKamar2[4][jumlahTransaksiSewa] = (hargaKamar
        * transaksiSewaKamar2[2][jumlahTransaksiSewa]
        - ((hargaKamar
            * transaksiSewaKamar2[2][jumlahTransaksiSewa]
            * transaksiSewaKamar2[5][jumlahTransaksiSewa] / 100));
    System.out.println("Total Sewa Kamar      : "
        + transaksiSewaKamar2[4][jumlahTransaksiSewa]);

    System.out.println(x:"=====");
    System.out.println(x:"                        Struk Transaksi - Sewa Kamar                        ");
    System.out.println(x:"=====");
    System.out.printf(format:"No Kartu Berobat      : %s\n", transaksiSewaKamar[0][jumlahTransaksiSewa]);
    System.out.printf(format:"Nama Pasien      : %s\n", transaksiSewaKamar[1][jumlahTransaksiSewa]);
    System.out.printf(format:"Usia Pasien      : %s\n", transaksiSewaKamar2[1][jumlahTransaksiSewa]);
    System.out.printf(format:"Keluhan Pasien   : %s\n", transaksiSewaKamar2[2][jumlahTransaksiSewa]);
    System.out.printf(format:"No Kamar yang disewa: %s\n", transaksiSewaKamar2[0][jumlahTransaksiSewa]);
    System.out.printf(format:"Lama sewa kamar   : %s\n", transaksiSewaKamar2[2][jumlahTransaksiSewa]);
    System.out.printf(format:"Harga sewa/malam   : Rp. %s\n", transaksiSewaKamar2[3][jumlahTransaksiSewa]);
    System.out.printf(format:"Total Harga Sewa   : Rp. %s\n", transaksiSewaKamar2[4][jumlahTransaksiSewa]);
    System.out.println(x:"=====");
    jumlahTransaksiSewa++;
}

```

KLINIK X - TRANSAKSI SEWA KAMAR

```

=====
| No Kamar | Status |
=====
| 1         | Tersedia |
| 2         | Tersedia |
| 3         | Tersedia |
| 4         | Tersedia |
| 5         | Tersedia |
=====
Masukkan nomor kamar yang ingin Anda sewa: 1
Anda memilih Kamar no 1
No Kartu Pasien: P0011
No Kartu Pasien : P0011
Nama Pasien      : reva
Usia Pasien      : 19
Keluhan Pasien   : flu
Apakah pasien memiliki kartu BPJS (0 = tidak, 1 = ya)?
0
Masukkan lama menginap (hari) : 2
Harga kamar/malam      : 100000
Total Sewa Kamar      : 200000
=====
                        Struk Transaksi - Sewa Kamar
=====
No Kartu Berobat      : P0011
Nama Pasien           : reva
Usia Pasien           : 19
Keluhan Pasien        : flu
No Kamar yang disewa: 1
Lama sewa kamar       : 2
Harga sewa/malam      : Rp. 100000
Total Harga Sewa      : Rp. 200000
=====

```

KLINIK X - TRANSAKSI SEWA KAMAR		
No Kamar	Status	
1	Tidak Tersedia	
2	Tersedia	
3	Tersedia	
4	Tersedia	
5	Tersedia	

4.3.7. User dapat melakukan transaksi rujuk pasien

```
static void transaksiRujukPasien() {
    boolean dataDitemukan = false;
    System.out.println(x: "");
    System.out.println(x: "");
    System.out.println(x: "KLINIK X - TRANSAKSI RUJUK PASIEN");
    System.out.println(x: "=====");

    for (int i = 0; i < daftarRs.length; i++) {
        System.out.println((i + 1) + ". " + daftarRs[i]);
    }

    boolean isValid = false;
    // Looping untuk validasi inputan RS Rujukan
    int selectedRsIndex = 0;
    while (isValid != true) {
        System.out.println(x: "=====");
        System.out.print("Masukkan nomor (1 - " + daftarRs.length + ") (0 for exit): ");
        selectedRsIndex = sc.nextInt();
        System.out.println(x: "");
        if (selectedRsIndex == 0) {
            dataDitemukan = true;
            break;
        }
        if (selectedRsIndex >= 1 && selectedRsIndex <= daftarRs.length) {
            statusRs = selectedRsIndex - 1; // Mengonversi nomor RS ke indeks dalam array
            System.out.println("Anda telah memilih " + daftarRs[statusRs]);
            transaksiRujukPasien[0][jumlahTransaksiRujuk] = daftarRs[statusRs];
            isValid = true;
            break;
        }
        else {
            System.out.println(x: "data tidak valid");
            isValid = false;
            continue;
        }
    }

    while (!dataDitemukan) {
        System.out.print(s: "No Kartu Pasien (0 for exit): ");
        String cariIndex = sc.next();
        System.out.println(x: "=====");
        int a = 0;
        // looping untuk validasi data pasien yang ingin dirujuk ditemukan
        for (int i = 0; i < transaksiRujukPasien.length; i++) {
            if (cariIndex.equalsIgnoreCase(transaksiPasien[0][i])) {
                hasil = a;
                System.out.println(
                    "No Kartu Pasien          : " + transaksiPasien[0][hasil];
                    transaksiRujukPasien[1][jumlahTransaksiSewa] = transaksiPasien[0][hasil];
            }
        }
    }
}
```

```

        transaksiRujukPasien[2][jumlahTransaksiSewa] = transaksiPasien[2][hasil];
        System.out
            .println("Usia Pasien          : "
                + transaksiPasien2[0][hasil]);
        usiaPasienRujuk[jumlahTransaksiSewa] = transaksiPasien2[0][hasil];
        dataDitemukan = true;
        a++;
        break;
    } else if (cariIndex.equalsIgnoreCase(anotherString:"0")) {
        dataDitemukan = true;
        break;
    } else {
        dataDitemukan = false;
    }
}

if (!dataDitemukan) {
    System.out.println(x:"Data Pasien tidak ditemukan. Silakan coba lagi.");
    dataDitemukan = false;
    continue;
} else if (cariIndex.equalsIgnoreCase(anotherString:"0")) {
    break;
}

System.out.println(x:"Apakah pasien memiliki kartu BPJS (0 = tidak, 1 = ya)? ");
statusbpjs = input.next().charAt(index:0);

if (statusbpjs == '0') {
    transaksiRujukPasien[3][jumlahTransaksi] = "-";
} else {
    System.out.print(s:"Masukkan No.BPJS pasien      : ");
    transaksiRujukPasien[3][jumlahTransaksi] = sc1.nextLine();
}

System.out.print(s:"Alasan Rujuk          :");
transaksiRujukPasien[4][jumlahTransaksiRujuk] = sc1.nextLine();
System.out.println(x:"=====");
System.out.println(x:"Struk Transaksi - Rujuk Pasien          "
    + "=====");
System.out.printf(format:"RS Rujukan          : %s\n", transaksiRujukPasien[0][jumlahTransaksiRuj
System.out.printf(format:"No Kartu Berobat      : %s\n", transaksiRujukPasien[1][jumlahTransaksiRuj
System.out.printf(format:"No BPJS              : %s\n", transaksiRujukPasien[3][jumlahTransaksiRuj
System.out.printf(format:"Nama Pasien          : %s\n", transaksiRujukPasien[2][jumlahTransaksiRuj
System.out.printf(format:"Usia Pasien          : %s\n", usiaPasienRujuk[jumlahTransaksiRujuk]);
System.out.printf(format:"Alasan Rujuk          : %s\n", transaksiRujukPasien[4][jumlahTransaksiRuj
System.out.println(x:"=====");
System.out.println(x:"Transaksi Selesai - Status Pending");
System.out.println(x:"");

    jumlahTransaksiRujuk++;
}
}

```

```

KLINIK X - TRANSAKSI RUJUK PASIEN
=====
1. RS-Panti Nirmala
2. RS-Islam Aisyiyah
3. RS-UD Saiful Anwar
=====
Masukkan nomor (1 - 3) (0 for exit): 1

Anda telah memilih RS-Panti Nirmala
No Kartu Pasien (0 for exit): P0011
-----
No Kartu Pasien      : P0011
Nama Pasien         : reva
Usia Pasien         : 19
Apakah pasien memiliki kartu BPJS (0 = tidak, 1 = ya)?
0
Alasan Rujuk        : sakit
=====
                        Struk Transaksi - Rujuk Pasien
=====
RS Rujukan          : RS-Panti Nirmala
No Kartu Berobat    : P0011
No BPJS            : null
Nama Pasien        : reva
Usia Pasien        : 19
Alasan Rujuk       : sakit
=====

```

4.3.8. User dapat melihat daftar kamar

```

v static void daftarKamar() {
    System.out.println(x:"");
    System.out.println(x:"");
    System.out.println(x:"KLINIK X - DATA DAFTAR KAMAR");
    System.out.println(x:"=====");
    // looping untuk menampilkan daftar kamar
v   for (int i = 0; i < statusKamar.length; i++) {
        System.out.println("Kamar no " + (i + 1) + ": "
            + (statusKamar[i] ? "Tersedia" : "Tidak Tersedia"));
    }
    System.out.println(x:"=====");
}

```

```

KLINIK X - DATA DAFTAR KAMAR
=====
Kamar no 1: Tersedia
Kamar no 2: Tersedia
Kamar no 3: Tersedia
Kamar no 4: Tersedia
Kamar no 5: Tersedia
=====

```

4.3.9. User dapat melihat daftar master pasien

```

static void dataMasterPasien() {
    System.out.println(x:"");
    System.out.println(x:"");
    System.out.println(x:"KLINIK X - DATA MASTER PASIEN");
    System.out.println(
        x:"=====");
    System.out.println(
        x:"| No Kartu          | Nama          | No KTP          | Alamat          | No. Telp        |");
    System.out.println(
        x:"=====");

    if (jumlahPasien == 0) {
        System.out.println(x:"Belum ada data master pasien");
    } else {
        for (int i = 0; i < jumlahPasien; i++) {
            System.out.printf(format:"| %-25s| %-25s| %-25s| %-25s|\n",
                registrasiPasien[0][i],
                registrasiPasien[1][i],
                registrasiPasien[2][i],
                registrasiPasien[3][i],
                registrasiPasien[4][i]);
        }
    }
    System.out.println(
        x:"=====");
    System.out.println(x:"");
    System.out.println(x:"");
}

```

KLINIK X - DATA MASTER PASIEN				
No Kartu	Nama	No KTP	Alamat	No. Telp
P0011	reva	83192319	malang	089

4.3.10. User dapat melihat history transaksi berobat

```

static void historyTransaksiBerobat() {
    System.out.println(x:"");
    System.out.println(x:"");
    System.out.println(x:"KLINIK X - DATA MASTER PASIEN");
    System.out.println(x:"=====");
    int j = 0;
    // Looping do-while untuk mencetak history transaksi
    do {
        System.out.println("TRANSAKSI KE-" + (j + 1));
        System.out.println("- Metode Pembayaran : " + transaksiPasien[5][j]);
        System.out.println("- Konsultasi : " + transaksiPasien[6][j]);
        System.out.println("- No Kartu Berobat : " + transaksiPasien[0][j]);
        System.out.println("- No Kartu BPJS : " + transaksiPasien[1][j]);
        System.out.println("- Nama Pasien : " + transaksiPasien[2][j]);
        System.out.println("- Usia Pasien : " + transaksiPasien[2][0][j]);
        System.out.println("- Keluhan Pasien : " + transaksiPasien[3][j]);
        System.out.println("- Total Biaya : Rp. " + transaksiPasien[2][2][j]);
        System.out.println(
            x:"=====");
        j++;
    } while (j < jumlahTransaksi);

    boolean isValid = false;
    // looping untuk validasi fitur search data pasien berdasarkan noKartu yang
    // dimasukkan
    while (!isValid) {
        System.out.print(s:"Masukkan No Kartu Pasien ('-' for exit): ");
        key = sc1.nextLine();
        if (key.equalsIgnoreCase(anotherString:"-")) {
            break;
        }

        // Looping untuk mencari data sesuai key
        for (int i = 0; i < transaksiPasien.length; i++) {
            if (key.equalsIgnoreCase(transaksiPasien[0][i])) {
                hasil = i;
                System.out.println(
                    "No Kartu Pasien " + key + " terdapat di index ke-" + hasil);
                isValid = true;
                continue;
            }
        }
        if (!isValid) {
            System.out.println(x:"Data Pasien tidak ditemukan. Silakan coba lagi.");
        }
    }
}

```

```

KLINIK X - DATA MASTER PASIEN
=====
TRANSAKSI KE-1
- Metode Pembayaran : Tunai
- Konsultasi       : Dokter Umum
- No Kartu Berobat  : P0011
- No Kartu BPJS     : -
- Nama Pasien      : reva
- Usia Pasien       : 19
- Keluhan Pasien    : flu
- Total Biaya       : Rp. 10000
=====

Masukkan No Kartu Pasien ('-' for exit): P0011
No Kartu Pasien P0011 terdapat di index ke-0

```

4.3.11. User dapat melakukan cetak laporan harian atau bulanan

```

static void cetakLaporanHarian() {
    System.out.println(x:"");
    System.out.println(x:"");
    System.out.println(x:"KLINIK X - LAPORAN TRANSAKSI HARIAN");

    LocalDate today = LocalDate.now();
    DateTimeFormatter formatter = DateTimeFormatter.ofPattern(pattern:"dd-MM-yyyy");

    // Print table header
    System.out.printf(format:"| %-20s | %-20s | %-20s | %-20s | %-20s | %-20s | %-20s | %-20s |\n",
        ...args:"Tanggal Transaksi", "No Kartu Berobat", "No BPJS", "Nama Pasien", "Usia Pasien", "Keluhan Pasien",
        "Metode Pembayaran");
    System.out.println(
        x:"-----");

    for (int i = 0; i < jumlahTransaksi; i++) {
        if (transactionDates[i].equals(today)) {
            // Print table rows
            System.out.printf(format:"| %-20s | %-20s | %-20s | %-20s | %-20s | %-20s | %-20s |\n",
                transactionDates[i].format(formatter),
                transaksiPasien[0][i], transaksiPasien[1][i], transaksiPasien[2][i],
                transaksiPasien[3][i], transaksiPasien[4][i], transaksiPasien[5][i]);
        }
    }

    // Print table footer
    System.out.println(
        x:"-----");
}

```

```

Pilih menu (1-9): 8
Pilih Jenis Laporan:
1. Laporan Harian
2. Laporan Bulanan
Masukkan Pilihan (1/2): 1

KLINIK X - LAPORAN TRANSAKSI HARIAN
| Tanggal Transaksi | No Kartu Berobat | No BPJS | Nama Pasien | Usia Pasien | Keluhan Pasien | Metode Pembayaran |
|-----|-----|-----|-----|-----|-----|-----|
| 12-12-2023 | P0011 | - | reva | 19 | flu | Tunai |
|-----|-----|-----|-----|-----|-----|-----|

```



```

static void cetakLaporanBulanan() {
    System.out.println(x:"");
    System.out.println(x:"");
    System.out.println(x:"KLINIK X - LAPORAN TRANSAKSI BULANAN");

    System.out.print(s:"Masukkan bulan (1-12): ");
    int selectedMonth = input.nextInt();
    System.out.println(
        x:"=====

    LocalDate thisMonth = LocalDate.now().withMonth(selectedMonth).withDayOfMonth(dayOfMonth:1);
    DateTimeFormatter formatter = DateTimeFormatter.ofPattern(pattern:"dd-MM-yyyy");

    System.out.printf(format:"| %-20s | %-20s | %-20s | %-20s | %-20s | %-20s | %-20s | \n",
        ...angs:"Tanggal Transaksi", "No Kartu Berobat", "No BPJS", "Nama Pasien", "Usia Pasien", "Keluhan Pasien",
        "Metode Pembayaran");
    System.out.println(
        x:"=====

    for (int i = 0; i < jumlahTransaksi; i++) {
        if (transactionDates[i].getMonth().equals(thisMonth.getMonth())) {
            // Print table rows
            System.out.printf(format:"| %-20s | %-20s | %-20s | %-20s | %-20s | %-20s | %-20s | \n",
                transactionDates[i].format(formatter),
                transaksiPasien[0][i], t String[][] transaksiPasien csPasien[2][i],
                transaksiPasien2[0][i], transaksiPasien[3][i], transaksiPasien[5][i]);
        } else {
            System.out.println(x:"Tidak ada transaksi");
        }
    }
    // Print table footer
    System.out.println(
        x:"=====
}

```

```

Pilih menu (1-9): 8
Pilih Jenis Laporan:
1. Laporan Harian
2. Laporan Bulanan
Masukkan Pilihan (1/2): 2

KLINIK X - LAPORAN TRANSAKSI BULANAN
Masukkan bulan (1-12): 12

| Tanggal Transaksi | No Kartu Berobat | No BPJS | Nama Pasien | Usia Pasien | Keluhan Pasien | Metode Pembayaran |
| 12-12-2023 | P0011 | - | reva | 19 | flu | Tunai |

```

4.4. Kendala yang dihadapi


- 4.4.1. Bug dan error
- 4.4.2. komunikasi dengan team yang kurang
- 4.4.3. beberapa fitur belum sempurna

4.5. Langkah Penyelesaian Kendala

- 4.5.1. Menyesuaikan bug dan error secara bertahap
- 4.5.2. meningkatkan komunikasi dengan team
- 4.5.3. menyempurnakan fitur-fitur yang yang belum complete

4.6. Kesimpulan

Sistem pemesanan di klinik adalah solusi komprehensif yang secara signifikan meningkatkan efisiensi operasional dan pengalaman pasien.



Dengan mengaktifkan pemesanan online, klinik dapat mengoptimalkan pemanfaatan sumber daya, meminimalkan waktu tunggu, dan memberikan layanan lebih cepat.

Teknologi ini tidak hanya menyediakan penjadwalan janji temu yang lebih nyaman bagi pasien, namun juga meningkatkan manajemen informasi pasien, memudahkan pekerjaan tim layanan kesehatan, dan menciptakan alur kerja yang terintegrasi.

Keamanan data pasien dan kepatuhan layanan kesehatan adalah prioritas, dan seluruh sistem dapat memberdayakan pasien dengan memberi mereka kendali atas layanan kesehatan pribadi mereka.

Oleh karena itu, sistem pemesanan klinik tidak hanya merupakan alat manajemen, namun juga merupakan elemen kunci dalam menyediakan layanan kesehatan berkualitas dengan cara yang efisien dan terjangkau.