

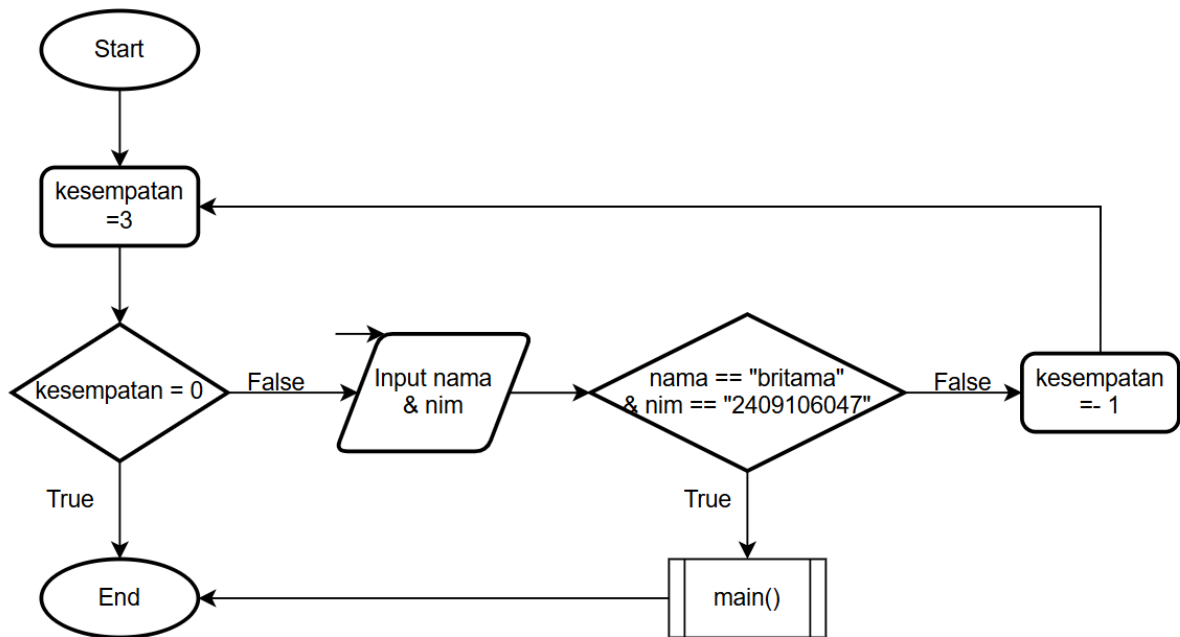
LAPORAN PRAKTIKUM
POSTTEST 6
ALGORITMA PEMROGRAMAN LANJUT



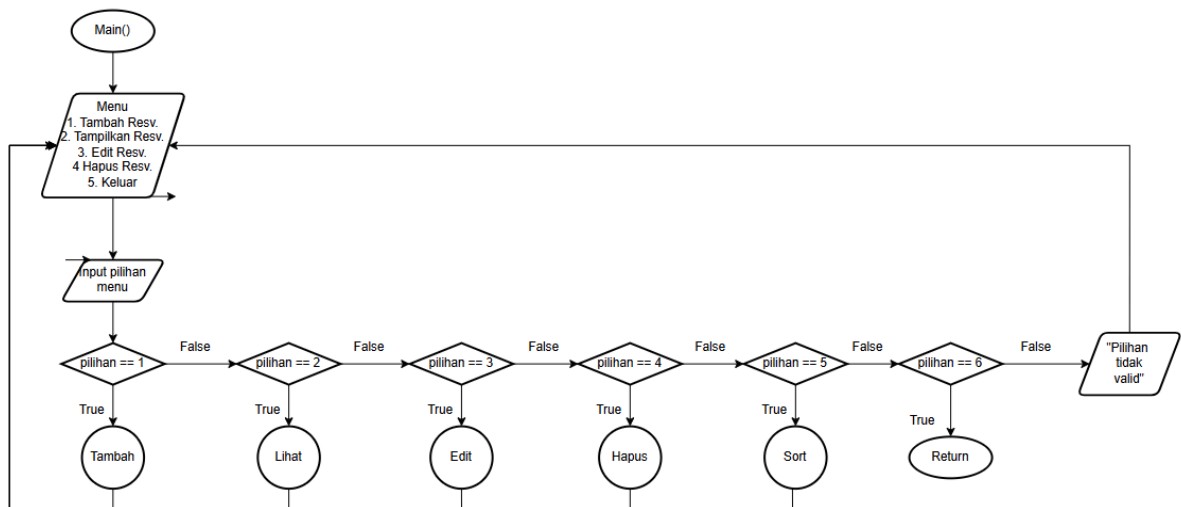
Disusun oleh:
Muhammad Britama Putra Jaya (2409106047)
Kelas (B1 '24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

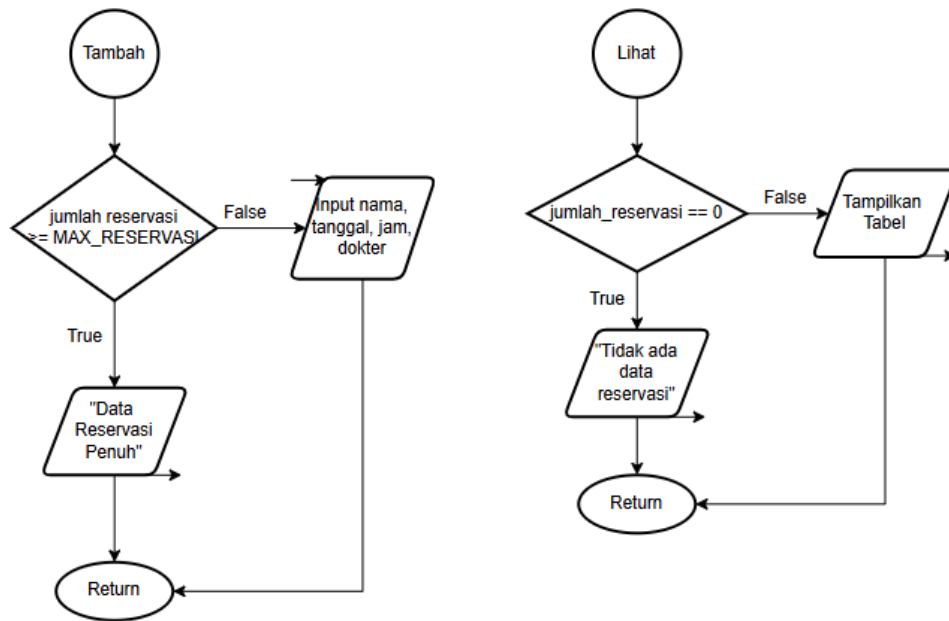
1. Flowchart



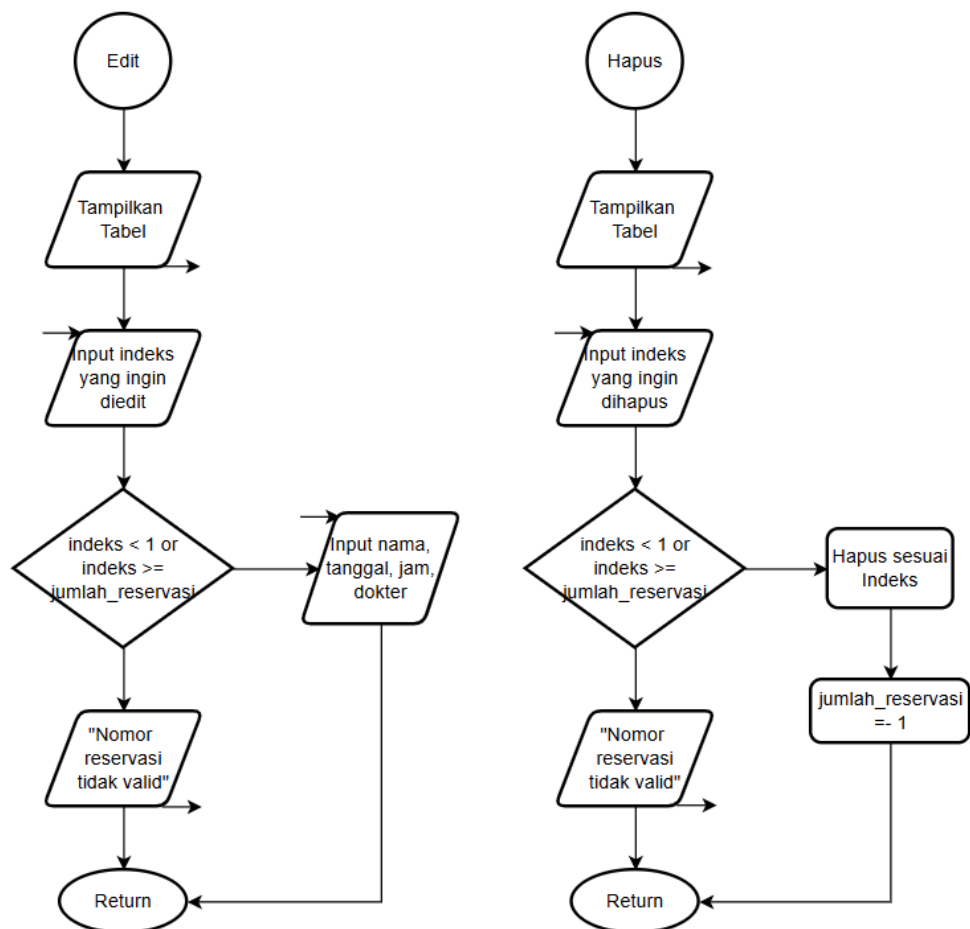
Gambar 1.1 Flowchart Login



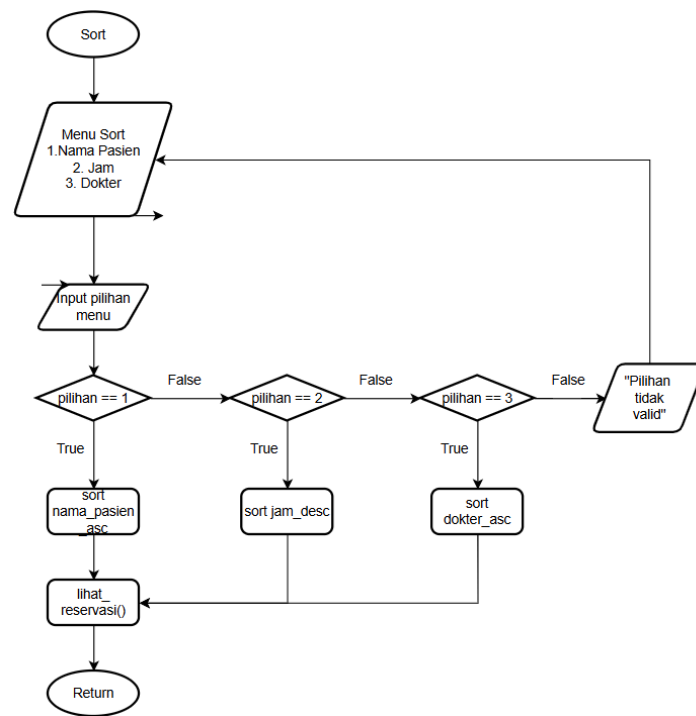
Gambar 1.2 Flowchart Menu



Gambar 1.3 Flowchart Tambah dan Lihat Reservasi



Gambar 1.4 Flowchart Edit dan Hapus Reservasi



Gambar 1.5 Menu Sort

2. Analisis Program

2.1 Deskripsi Singkat Program

Pada posttest 6 kali ini, saya menambahkan menu sort pada program saya. Sesuai dengan ketentuan saya menambahkan 3 metode sort, yaitu Bubble Sort secara ascending pada nama pasien, Selection Sort secara descending pada tanggal dan jam reservasi, dan Insertion Sort secara ascending pada nama dokter.

3. Source Code

A. Struct untuk Data Reservasi (Nested Struct)

Fitur ini digunakan untuk menyimpan data. Dimana “Pasien” menyimpan nama pasien; “Jadwal” menyimpan tanggal dan jam reservasi; “Dokter” menyimpan nama dokter. Untuk Nested Struct nya disini adalah reservasi, dimana menggunakan struct pasien, jadwal, dan dokter.

Source Code:

```
struct Pasien {  
    string nama;  
};  
  
struct Jadwal {  
    string tanggal;  
    string jam;  
};  
  
struct Dokter {  
    string nama;  
};  
  
struct Reservasi {  
    Pasien pasien;  
    Jadwal jadwal;  
    Dokter dokter;  
};
```

B. Tambah Reservasi

Fitur ini digunakan untuk menambah reservasi klinik gigi, dimana program akan mengecek apakah reservasi sudah penuh, meminta user untuk menginput nama pasien, tanggal, jam, dan dokter, serta menambah “jumlah_reservasi” setelah data berhasil disimpan.

Source Code:

```
void tambah_reservasi(Reservasi* data, int* jumlah) {
    if (*jumlah >= MAX_RESERVASI) {
        cout << "Data reservasi penuh!" << endl;
        return;
    }

    cout << "\nMasukkan Nama Pasien : ";
    getline(cin, data[*jumlah].pasien.nama);
    cout << "Masukkan Tanggal (DD-MM-YYYY) : ";
    getline(cin, data[*jumlah].jadwal.tanggal);
    cout << "Masukkan Jam (HH:MM) : ";
    getline(cin, data[*jumlah].jadwal.jam);
    cout << "Masukkan Nama Dokter : ";
    string nama_dokter;
    getline(cin, nama_dokter);
    data[*jumlah].dokter.nama = "Drg. " + nama_dokter;
    (*jumlah)++;
    cout << "Reservasi berhasil ditambahkan!" << endl;
}
```

C. Menampilkan Reservasi

Fitur ini digunakan untuk menampilkan data reservasi.

Source Code:

```
void lihat_reservasi(Reservasi* data, int jumlah) {
    cout <<
    "\n===== " <<
    endl;
    cout << "| No | Nama Pasien | Tanggal | Jam | Dokter"
    |" << endl;
    cout <<
    "===== " << endl;

    for (int i = 0; i < jumlah; i++) {
        cout << "| " << setw(3) << i + 1 << " | "
        << setw(20) << left << (*(data + i)).pasien.nama << " | "
        << setw(10) << (*(data + i)).jadwal.tanggal << " | "
        << setw(5) << (*(data + i)).jadwal.jam << " | "
        << setw(15) << left << (*(data + i)).dokter.nama << " |" << endl;
    }
    cout <<
    "===== " << endl;
    cout << "Total reservasi saat ini: " << hitung_reservasi(jumlah) << endl;
}
```

D. Mengedit Reservasi

Fitur ini digunakan mengedit reservasi yang ada sesuai dengan nomor reservasi yang ingin diedit.

Source Code:

```
void edit_reservasi(Reservasi* data, int jumlah) {
    lihat_reservasi(data, jumlah);

    int indeks;
    cout << "Masukkan nomor reservasi yang ingin diedit : ";
    cin >> indeks;
    cin.ignore();
    if (indeks < 1 || indeks > jumlah) {
        cout << "Nomor reservasi tidak valid!" << endl;
        return;
    }

    Reservasi* p = &data[indeks - 1];
    cout << "\n--- Data Lama ---" << endl;
    cout << "Nama Pasien : " << p->pasien.nama << endl;
    cout << "Tanggal      : " << p->jadwal.tanggal << endl;
    cout << "Jam           : " << p->jadwal.jam << endl;
    cout << "Dokter       : " << p->dokter.nama << endl;

    cout << "\n--- Masukkan Data Baru ---" << endl;
    cout << "Nama Pasien : ";
    getline(cin, p->pasien.nama);
    cout << "Tanggal (DD-MM-YYYY) : ";
    getline(cin, p->jadwal.tanggal);
    cout << "Jam (HH:MM) : ";
    getline(cin, p->jadwal.jam);
    cout << "Nama Dokter : ";
    string nama_dokter;
    getline(cin, nama_dokter);
    p->dokter.nama = "Drg. " + nama_dokter;

    cout << "Reservasi berhasil diperbarui!" << endl;
}
```


E. Menghapus Reservasi

Fitur ini digunakan untuk menghapus reservasi yang ada sesuai dengan nomor reservasi yang ingin dihapus.

Source Code:

```
void hapus_reservasi(Reservasi* data, int* jumlah) {
    lihat_reservasi(data, *jumlah);

    int indeks;
    cout << "Masukkan nomor reservasi yang ingin dihapus : ";
    cin >> indeks;
    cin.ignore();
    if (indeks < 1 || indeks > *jumlah) {
        cout << "Nomor reservasi tidak valid!" << endl;
        return;
    }

    Reservasi* p = &data[indeks - 1];

    cout << "\nAnda akan menghapus reservasi berikut:" << endl;
    cout << "Nama Pasien : " << p->pasien.nama << endl;
    cout << "Tanggal      : " << p->jadwal.tanggal << endl;
    cout << "Jam           : " << p->jadwal.jam << endl;
    cout << "Dokter        : " << p->dokter.nama << endl;

    char konfirmasi;
    cout << "Yakin ingin menghapus? (y/n): ";
    cin >> konfirmasi;
    cin.ignore();

    if (konfirmasi == 'y' || konfirmasi == 'Y') {
        for (int i = indeks - 1; i < *jumlah - 1; i++) {
            data[i] = data[i + 1];
        }
        (*jumlah)--;
        cout << "Reservasi berhasil dihapus!" << endl;
    } else {
        cout << "Penghapusan dibatalkan." << endl;
    }
}
```

F. Menu Sort

Dalam program ini, saya menambahkan fitur sorting untuk mengatur data reservasi klinik supaya lebih rapi dan terstruktur sesuai kebutuhan pengguna.

Saya menerapkan tiga metode sorting berbeda :

- Bubble Sort : untuk mengurutkan nama pasien secara ascending (A-Z).

Karena nama pasien berbentuk huruf, Bubble Sort dipilih karena simpel, cocok untuk data berjumlah kecil hingga menengah.

- Selection Sort : untuk mengurutkan tanggal reservasi secara descending (terbaru ke terlama).

Karena tanggal bersifat angka (dan prioritas waktu), Selection Sort efektif untuk mencari nilai maksimum dan mengatur berdasarkan urutan waktu mundur.

- Insertion Sort : untuk mengurutkan nama dokter secara ascending (A-Z).

Insertion Sort cepat untuk data kecil dan stabil, cocok digunakan karena nama dokter umumnya jumlahnya tidak terlalu banyak.

```
void sort_nama_pasien_asc(Reservasi* data, int jumlah) {
    for (int i = 0; i < jumlah - 1; i++) {
        for (int j = i + 1; j < jumlah; j++) {
            if (data[i].pasien.nama > data[j].pasien.nama) {
                swap(data[i], data[j]);
            }
        }
    }
}

void sort_jam_desc(Reservasi* data, int jumlah) {
    for (int i = 0; i < jumlah - 1; i++) {
        for (int j = i + 1; j < jumlah; j++) {
            if (data[i].jadwal.jam < data[j].jadwal.jam) {
                swap(data[i], data[j]);
            }
        }
    }
}

void sort_dokter_asc(Reservasi* data, int jumlah) {
    for (int i = 0; i < jumlah - 1; i++) {
        for (int j = i + 1; j < jumlah; j++) {
            if (data[i].dokter.nama > data[j].dokter.nama) {
                swap(data[i], data[j]);
            }
        }
    }
}
```

```

}

// Menu sorting
void menu_sorting(Reservasi* data, int jumlah) {
    int pilihan_sort;
    cout << "\n=== MENU SORTING ===\n";
    cout << "1. Sort Nama Pasien (A-Z)\n";
    cout << "2. Sort Jam (Terbaru ke Terlama)\n";
    cout << "3. Sort Nama Dokter (A-Z)\n";
    cout << "Pilih metode sorting: ";
    cin >> pilihan_sort;
    cin.ignore();

    switch (pilihan_sort) {
        case 1:
            sort_nama_pasien_asc(data, jumlah);
            cout << "Data berhasil diurutkan berdasarkan Nama Pasien\n";
            break;
        case 2:
            sort_jam_desc(data, jumlah);
            cout << "Data berhasil diurutkan berdasarkan Jam (Terbaru ke\n";
            Terlama).\n";
            break;
        case 3:
            sort_dokter_asc(data, jumlah);
            cout << "Data berhasil diurutkan berdasarkan Nama Dokter\n";
            (A-Z).\n";
            break;
        default:
            cout << "Pilihan tidak valid!\n";
            break;
    }
    lihat_reservasi(data, jumlah);
}

```

4. Uji Coba dan Hasil Output

4.1 Hasil Output

```
=====
                        LOGIN SISTEM KLINIK
=====
Masukkan Nama : britama
Masukkan NIM : 2409106047

=====
            MENU MANAJEMEN RESERVASI KLINIK
=====
1. Tambah Reservasi
2. Tampilkan Reservasi
3. Edit Reservasi
4. Hapus Reservasi
5. Keluar
=====
Pilih menu : █
```

Gambar 4.1 Login Berhasil dan Menu Utama

```
=====
            MENU MANAJEMEN RESERVASI KLINIK
=====
1. Tambah Reservasi
2. Tampilkan Reservasi
3. Edit Reservasi
4. Hapus Reservasi
5. Keluar
=====
Pilih menu : 1

Masukkan Nama Pasien : Jarwo
Masukkan Tanggal (DD-MM-YYYY) : 14-09-2025
Masukkan Jam (HH:MM) : 14:00
Masukkan Nama Dokter : Harun
Reservasi berhasil ditambahkan!
```

Gambar 4.2 Tambah Reservasi

```
=====
MENU MANAJEMEN RESERVASI KLINIK
=====
1. Tambah Reservasi
2. Tampilkan Reservasi
3. Edit Reservasi
4. Hapus Reservasi
5. Keluar
=====
Pilih menu : 2

=====
| No | Nama Pasien | Tanggal | Jam | Dokter |
=====
| 1 | Harun | 14-09-2025 | 14:00 | Drg. Jarwo |
=====
Total reservasi saat ini: 1
```

Gambar 4.3 Tampilkan Reservasi

```

=====
MENU MANAJEMEN RESERVASI KLINIK
=====
1. Tambah Reservasi
2. Tampilkan Reservasi
3. Edit Reservasi
4. Hapus Reservasi
5. Keluar
=====
Pilih menu : 3

=====
| No | Nama Pasien | Tanggal | Jam | Dokter |
=====
| 1 | Harun | 14-09-2025 | 14:00 | Drg. Jarwo |
=====
Total reservasi saat ini: 1
Masukkan nomor reservasi yang ingin diedit : 1

--- Data Lama ---
Nama Pasien : Harun
Tanggal : 14-09-2025
Jam : 14:00
Dokter : Drg. Jarwo

--- Masukkan Data Baru ---
Nama Pasien : Jawir
Tanggal (DD-MM-YYYY) : 14-09-2025
Jam (HH:MM) : 14:30
Nama Dokter : Harun
Reservasi berhasil diperbarui!

```

Gambar 4.4 Mengedit Reservasi

```
=====
MENU MANAJEMEN RESERVASI KLINIK
=====
1. Tambah Reservasi
2. Tampilkan Reservasi
3. Edit Reservasi
4. Hapus Reservasi
5. Keluar
=====
Pilih menu : 4

=====
| No | Nama Pasien          | Tanggal   | Jam   | Dokter          |
=====
| 1  | Udin                 | 14-09-2025 | 14:30 | Drg. Tarno      |
=====

Masukkan nomor reservasi yang ingin dihapus : 1
Reservasi berhasil dihapus!

=====
MENU MANAJEMEN RESERVASI KLINIK
=====
1. Tambah Reservasi
2. Tampilkan Reservasi
3. Edit Reservasi
4. Hapus Reservasi
5. Keluar
=====
Pilih menu : 2
Tidak ada data reservasi!
```

Gambar 4.5 Menghapus Reservasi

```

=====
                        LOGIN SISTEM KLINIK
=====
Masukkan Nama : abc
Masukkan NIM : abc
Login gagal! Sisa kesempatan : 2

=====
                        LOGIN SISTEM KLINIK
=====
Masukkan Nama : abc
Masukkan NIM : abc
Login gagal! Sisa kesempatan : 1

=====
                        LOGIN SISTEM KLINIK
=====
Masukkan Nama : abc
Masukkan NIM : abc
Login gagal! Sisa kesempatan : 0
Anda diblokir! Silahkan hubungi admin.

```

Gambar 4.6 Gagal Login

```

=====
MENU MANAJEMEN RESERVASI KLINIK
=====
1. Tambah Reservasi
2. Tampilkan Reservasi
3. Edit Reservasi
4. Hapus Reservasi
5. Sorting Reservasi
6. Keluar
=====
Pilih menu : 5

=== MENU SORTING ===
1. Sort Nama Pasien (A-Z)
2. Sort Jam (Terbaru ke Terlama)
3. Sort Nama Dokter (A-Z)
Pilih metode sorting: 1
Data berhasil diurutkan berdasarkan Nama Pasien (A-Z).

=====
| No | Nama Pasien | Tanggal | Jam | Dokter |
=====
| 1 | Haris | 16-09-2025 | 12:30 | Drg. Rudi |
| 2 | Harun | 14-09-2025 | 13:00 | Drg. Gunawan |
| 3 | Jawir | 14-09-2025 | 14:00 | Drg. Gunawan |
=====
Total reservasi saat ini: 3

```

Gambar 4.7 Sort berdasar nama pasien


```

=== MENU SORTING ===
1. Sort Nama Pasien (A-Z)
2. Sort Jam (Terbaru ke Terlama)
3. Sort Nama Dokter (A-Z)
Pilih metode sorting: 2
Data berhasil diurutkan berdasarkan Jam (Terbaru ke Terlama).

=====
| No | Nama Pasien | Tanggal | Jam | Dokter |
=====
| 1 | Jawir | 14-09-2025 | 14:00 | Drg. Gunawan |
| 2 | Harun | 14-09-2025 | 13:00 | Drg. Gunawan |
| 3 | Haris | 16-09-2025 | 12:30 | Drg. Rudi |
=====
Total reservasi saat ini: 3

```

Gambar 4.8 Sort berdasar jam dan tanggal

```

=== MENU SORTING ===
1. Sort Nama Pasien (A-Z)
2. Sort Jam (Terbaru ke Terlama)
3. Sort Nama Dokter (A-Z)
Pilih metode sorting: 3
Data berhasil diurutkan berdasarkan Nama Dokter (A-Z).

=====
| No | Nama Pasien | Tanggal | Jam | Dokter |
=====
| 1 | Jawir | 14-09-2025 | 14:00 | Drg. Gunawan |
| 2 | Harun | 14-09-2025 | 13:00 | Drg. Gunawan |
| 3 | Haris | 16-09-2025 | 12:30 | Drg. Rudi |
=====
Total reservasi saat ini: 3

```

Gambar 4.9 Sort berdasar nama dokter

5. Git

```
PS C:\Users\my asus\OneDrive\Desktop\praktikum-apl> git add .
warning: in the working copy of 'post-test/post-test-2/2409106047_Muhammad Britama Putra Jaya_PT_2.drawio', LF will be replaced by CRLF
the next time Git touches it
warning: in the working copy of 'post-test/post-test-6/.vscode/tasks.json', LF will be replaced by CRLF the next time Git touches it
```

Gambar 5.1 Git Add : Menambah file ke git commit. Dengan menambahkan “.” secara otomatis semua perubahan file yang ada di direktori “praktikum-apl”.

```
PS C:\Users\my asus\OneDrive\Desktop\praktikum-apl> git commit -m "Finish Code Post Test 6"
[main 3e35305] Finish Code Post Test 6
4 files changed, 626 insertions(+), 175 deletions(-)
create mode 100644 post-test/post-test-6/.vscode/tasks.json
create mode 100644 post-test/post-test-6/2409106047_Muhammad Britama Putra Jaya_PT_6.cpp
create mode 100644 post-test/post-test-6/2409106047_Muhammad Britama Putra Jaya_PT_6.exe
```

Gambar 5.2 Git Commit : digunakan untuk menyimpan perubahan dalam repository Git secara permanen.

```
PS C:\Users\my asus\OneDrive\Desktop\praktikum-apl> git push -u origin main
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 4 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (12/12), 984.90 KiB | 737.00 KiB/s, done.
Total 12 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.
To https://github.com/Britamaxx/praktikum-apl.git
 6f09ff5..88bfb1c main -> main
branch 'main' set up to track 'origin/main'.
```

Gambar 5.3 Git Push : digunakan untuk mengirim perubahan (commit) dari repository lokal ke repository jarak jauh, sehingga bisa diakses oleh orang lain atau disinkronkan dengan repository utama.