

HeReVa-HPV: Health Record Patient for Vaccine



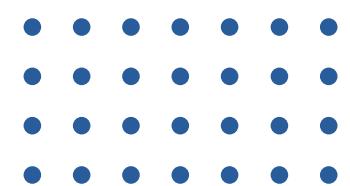
Nadira Najmi R.
2306230243



Shifa Shakila S.
2306215053



Annisa Rafiyanti
2306208640



KELOMPOK 5



Latar Belakang



Pencatatan **data kesehatan pasien secara lengkap dan terstruktur** sangat penting dalam dunia medis, terutama untuk mendukung tindakan **preventif** seperti **vaksinasi**. Namun, pencatatan sering kali masih dilakukan secara **manual dan tidak terorganisir**, sehingga informasi penting seperti **riwayat alergi, status vaksinasi, atau riwayat penyakit** bisa terlewat.

Untuk mengatasi hal ini, dibuatlah **HeReVa-HPV : Health Record Patient for Vaccine** untuk mencatat data kesehatan pasien, termasuk mencatat riwayat penyakit dan alergi, serta mengecek kelayakan vaksin berdasarkan data yang diinput.

Tujuan

- Mencatat data kesehatan pasien seperti riwayat penyakit, alergi, status vaksinasi dan lain-lain secara digital.
- Mengecek kelayakan vaksinasi pasien berdasarkan umur dan riwayat medis.
- Menyimpan data lebih dari satu pasien secara terstruktur
- Mengaplikasikan konsep dasar pemrograman dalam bentuk program yang interaktif dan mudah digunakan banyak khalayak.





Limitasi



- Input tidak aman yaitu adanya risiko buffer overflow (nama/riwayat penyakit)
- Manajemen memori kurang sempurna, memory leak bisa terjadi saat reset data
- Untuk save data harus keluar dari program dahulu
- Bentuk dari tabel masih kurang rapih penempatannya



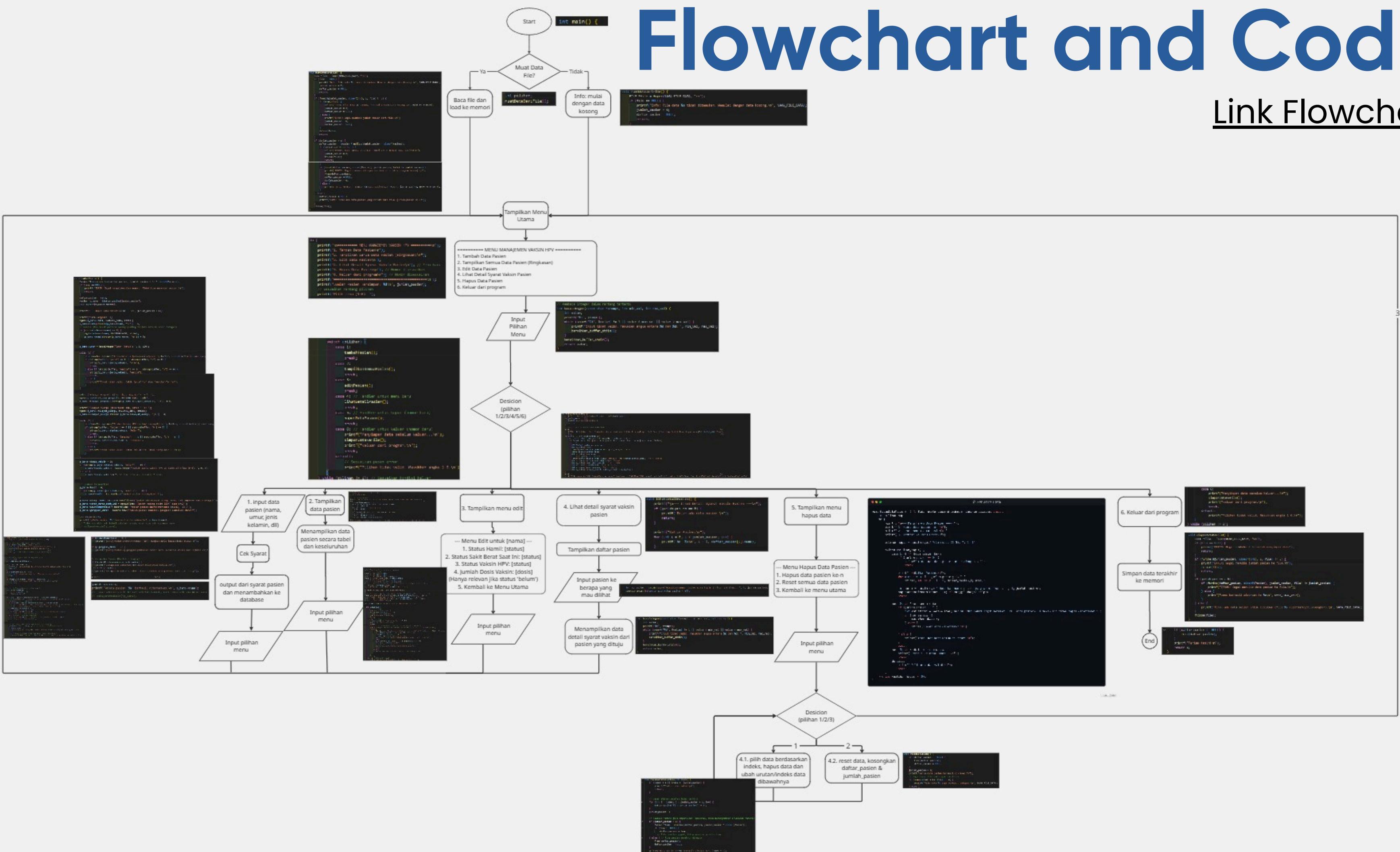
Pengguna atau User

Pengguna dari program ini adalah **tenaga administrasi atau petugas medis** di fasilitas **pelayanan kesehatan** yang bertugas mencatat dan memeriksa data kesehatan pasien terkait vaksinasi HPV. Program ini juga bisa digunakan oleh petugas vaksinasi untuk membantu memastikan kelayakan penerimaan vaksin berdasarkan riwayat kesehatan pasien.

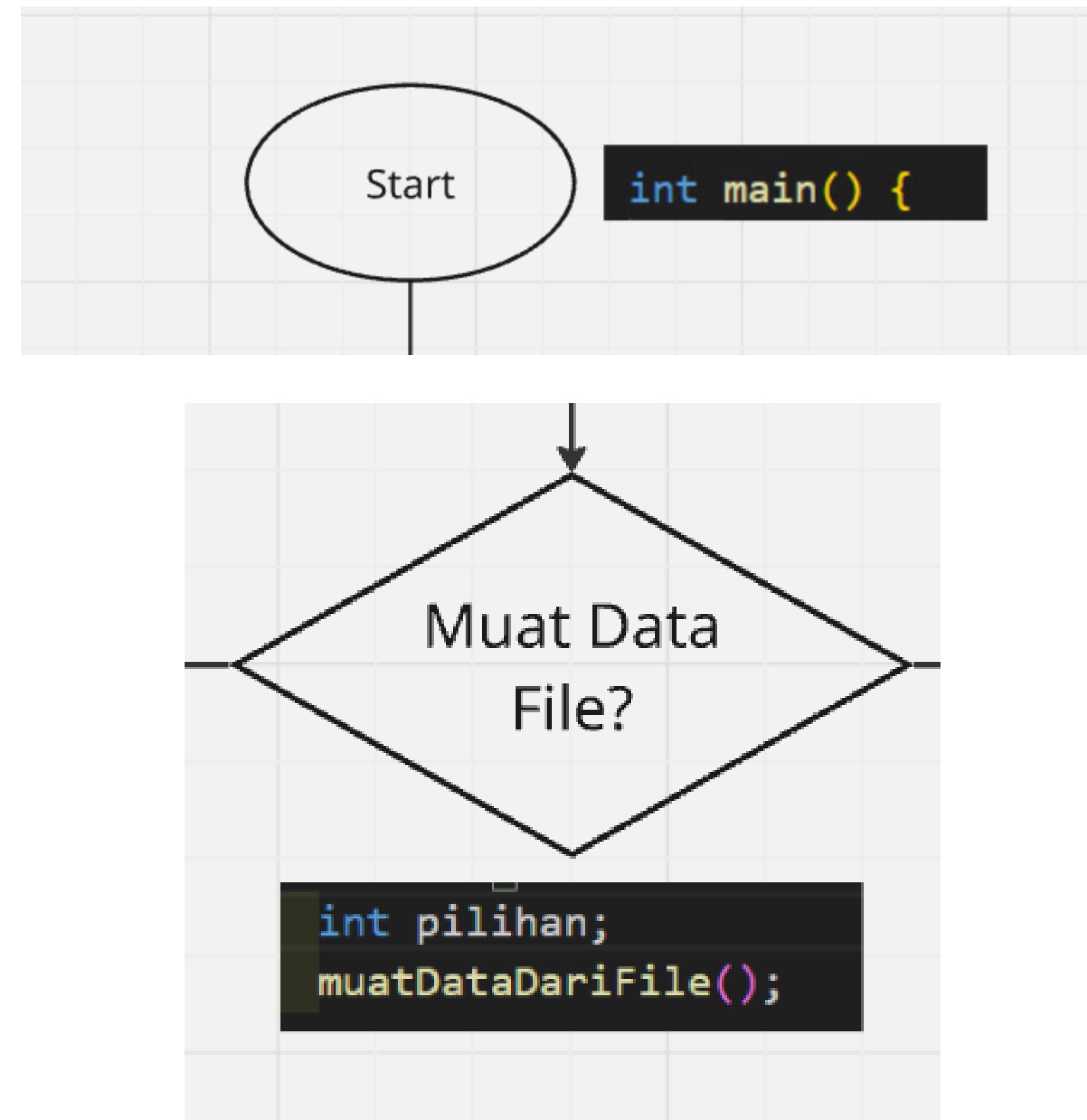


Flowchart and Code

Link Flowchart



Flowchart and Code



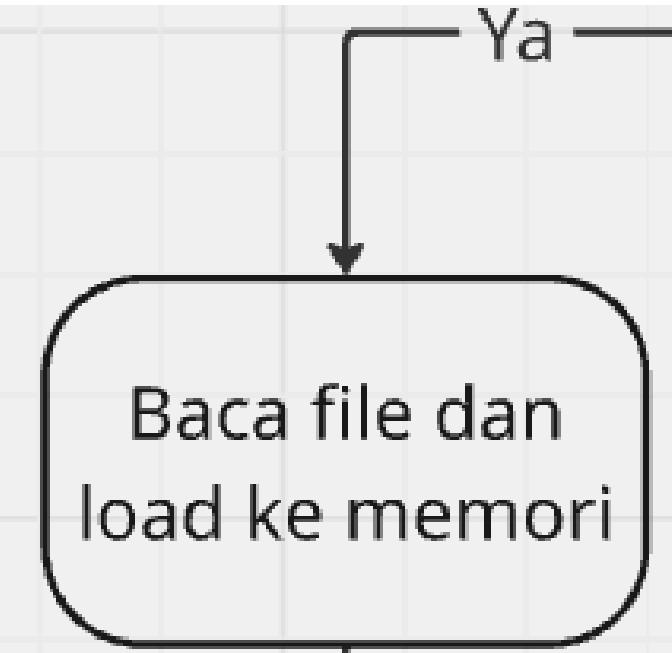
Flowchart and Code



```
void muatDataDariFile() {
    FILE *file = fopen(NAMA_FILE_DATA, "rb");
    if (file == NULL) {
        printf("Info: File data %s tidak ditemukan. Memulai dengan data kosong.\n", NAMA_FILE_DATA);
        jumlah_pasien = 0;
        daftar_pasien = NULL;
        return;
    }
}
```

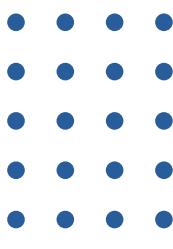
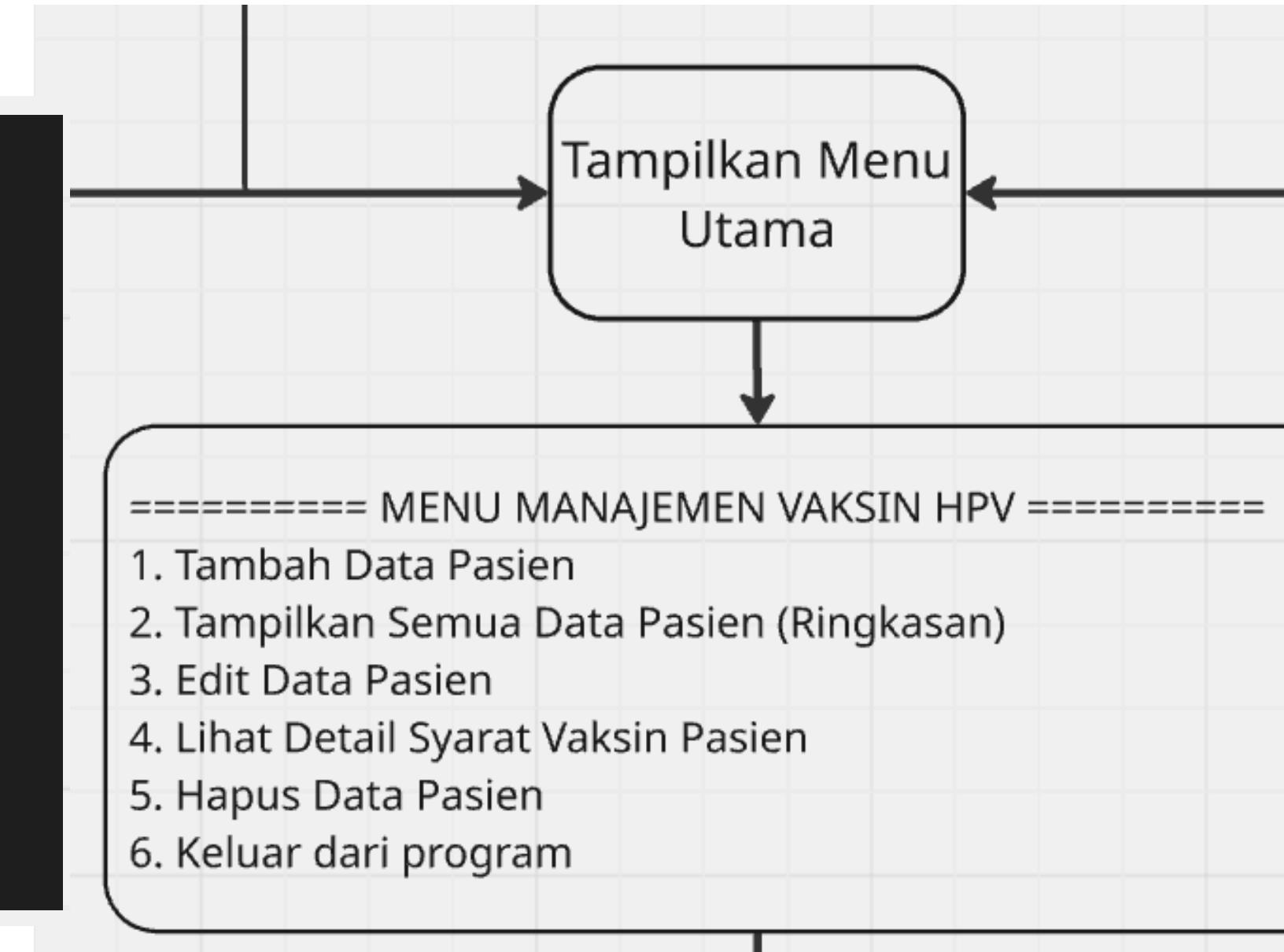
Flowchart and Code

```
void muatDataDariFile() {
    FILE *file = fopen(NAMA_FILE_DATA, "rb");
    if (file == NULL) {
        printf("Info: File data %s tidak ditemukan. Memulai dengan data kosong.\n", NAMA_FILE_DATA);
        jumlah_pasien = 0;
        daftar_pasien = NULL;
        return;
    }
    if (fread(&jumlah_pasien, sizeof(int), 1, file) != 1) {
        if (feof(file)) {
            printf("Info: File data %s kosong. Memulai dengan data kosong.\n", NAMA_FILE_DATA);
            jumlah_pasien = 0;
            daftar_pasien = NULL;
        } else {
            printf("ERROR: Gagal membaca jumlah pasien dari file.\n");
            jumlah_pasien = 0;
            daftar_pasien = NULL;
        }
        fclose(file);
        return;
    }
    if (jumlah_pasien > 0) {
        daftar_pasien = (Pasien *)malloc(jumlah_pasien * sizeof(Pasien));
        if (daftar_pasien == NULL) {
            printf("ERROR: Gagal mengalokasikan memori untuk memuat data pasien!\n");
            jumlah_pasien = 0;
            fclose(file);
            return;
        }
        if (fread(daftar_pasien, sizeof(Pasien), jumlah_pasien, file) != jumlah_pasien) {
            printf("ERROR: Gagal membaca data pasien dari file (data mungkin korup).\n");
            free(daftar_pasien);
            daftar_pasien = NULL;
            jumlah_pasien = 0;
        } else {
            printf("Info: Berhasil memuat %d data pasien dari %s\n", jumlah_pasien, NAMA_FILE_DATA);
        }
    } else {
        daftar_pasien = NULL;
        printf("Info: Tidak ada data pasien yang dimuat dari file (jumlah pasien 0).\n");
    }
    fclose(file);
}
```

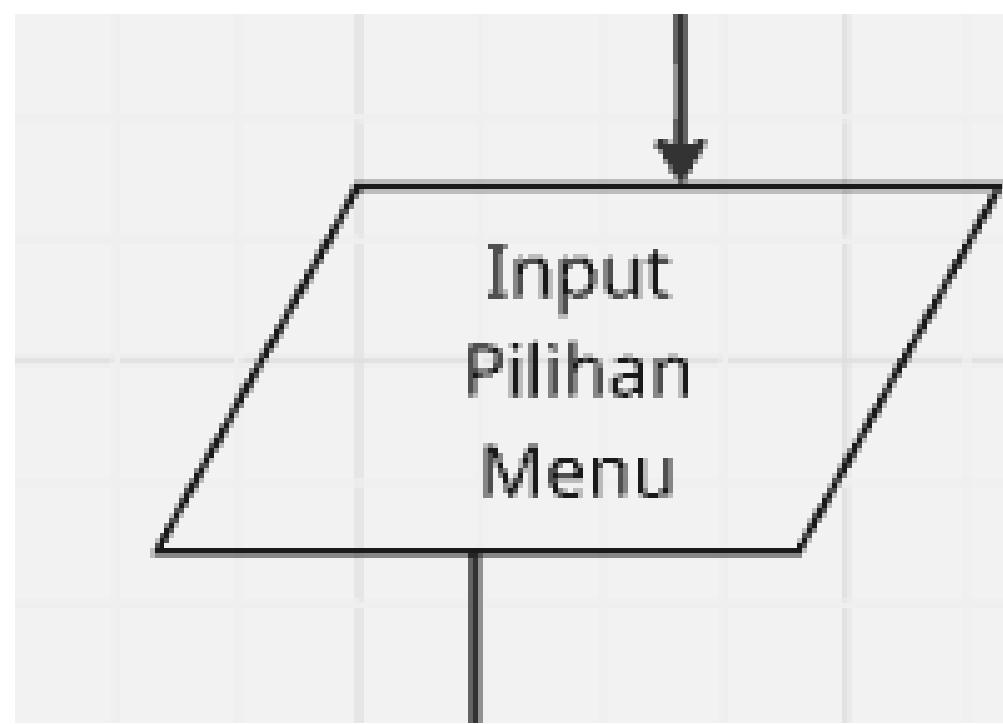


Flowchart and Code

```
do {  
    printf("\n===== MENU MANAJEMEN VAKSIN HPV =====\n");  
    printf("1. Tambah Data Pasien\n");  
    printf("2. Tampilkan Semua Data Pasien (Ringkasan)\n");  
    printf("3. Edit Data Pasien\n");  
    printf("4. Lihat Detail Syarat Vaksin Pasien\n"); // Menu baru  
    printf("5. Hapus Data Pasien\n"); // Nomor disesuaikan  
    printf("6. Keluar dari program\n"); // Nomor disesuaikan  
    printf("=====\\n");  
    printf("Jumlah Pasien Tersimpan: %d\\n", jumlah_pasien);  
    // Sesuaikan rentang pilihan  
    printf("Pilih menu (1-6): ");
```



Flowchart and Code



```
// Membaca integer dalam rentang tertentu
int bacaInteger(const char *prompt, int min_val, int max_val) {
    int value;
    printf("%s", prompt);
    while (scanf("%d", &value) != 1 || value < min_val || value > max_val) {
        printf("Input tidak valid. Masukkan angka antara %d dan %d: ", min_val, max_val);
        bersihkan_buffer_stdin();
    }
    bersihkan_buffer_stdin();
    return value;
}
```

```
switch (pilihan) {
    case 1:
        tambahPasien();
        break;
    case 2:
        tampilkanSemuaPasien();
        break;
    case 3:
        editPasien();
        break;
    case 4: // Handler untuk menu baru
        lihatDetailPasien();
        break;
    case 5: // Handler untuk hapus (nomor baru)
        hapusDataPasien();
        break;
    case 6: // Handler untuk keluar (nomor baru)
        printf("Menyimpan data sebelum keluar...\n");
        simpanDataKeFile();
        printf("Keluar dari program.\n");
        break;
    default:
        // Sesuaikan pesan error
        printf("Pilihan tidak valid. Masukkan angka 1-6.\n")
}
} while (pilihan != 6); // Sesuaikan kondisi keluar
```



Flowchart and Code

```
void tambahPasien() {
    Pasien *temp = realloc(daftar_pasien, (jumlah_pasien + 1) * sizeof(Pasien));
    if (temp == NULL) {
        printf("ERROR: Gagal mengalokasikan memori! Tidak bisa menambah pasien.\n");
        return;
    }
    daftar_pasien = temp;
    Pasien *p_baru = &daftar_pasien[jumlah_pasien];
    char buffer[PANJANG_BUFFER];

    printf("--- Input Data Pasien ke-%d ---\n", jumlah_pasien + 1);

    printf("Nama Lengkap: ");
    fgets(p_baru->nama, PANJANG_NAMA, stdin);
    p_baru->nama[strcspn(p_baru->nama, "\n")] = 0;
    // Handle jika input pertama kosong (kadang terjadi setelah scanf integer)
    if (strlen(p_baru->nama) == 0) {
        fgets(p_baru->nama, PANJANG_NAMA, stdin);
        p_baru->nama[strcspn(p_baru->nama, "\n")] = 0;
    }

    p_baru->umur = bacaInteger("Umur (tahun): ", 0, 150);

    while (1) {
        if (!bacaStringLower("Jenis Kelamin (pria/wanita/p/w): ", buffer, sizeof(buffer))) continue;
        if (strcmp(buffer, "pria") == 0 || strcmp(buffer, "p") == 0) {
            strcpy(p_baru->jenis_kelamin, "pria");
            break;
        } else if (strcmp(buffer, "wanita") == 0 || strcmp(buffer, "w") == 0) {
            strcpy(p_baru->jenis_kelamin, "wanita");
            break;
        } else {
            printf("Input tidak valid. Ketik 'pria'/'p' atau 'wanita'/'w'.\n");
        }
    }

    printf("Riwayat Penyakit (jika tidak ada, ketik '-'): ");
    fgets(p_baru->riwayat_penyakit, PANJANG_INFO, stdin);
    p_baru->riwayat_penyakit[strcspn(p_baru->riwayat_penyakit, "\n")] = 0;

    printf("Riwayat Allergi (jika tidak ada, ketik '-'): ");
    fgets(p_baru->riwayat_allergi, PANJANG_INFO, stdin);
    p_baru->riwayat_allergi[strcspn(p_baru->riwayat_allergi, "\n")] = 0;

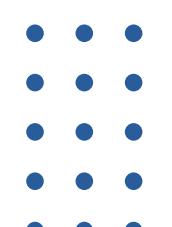
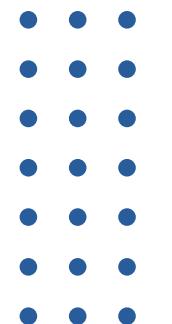
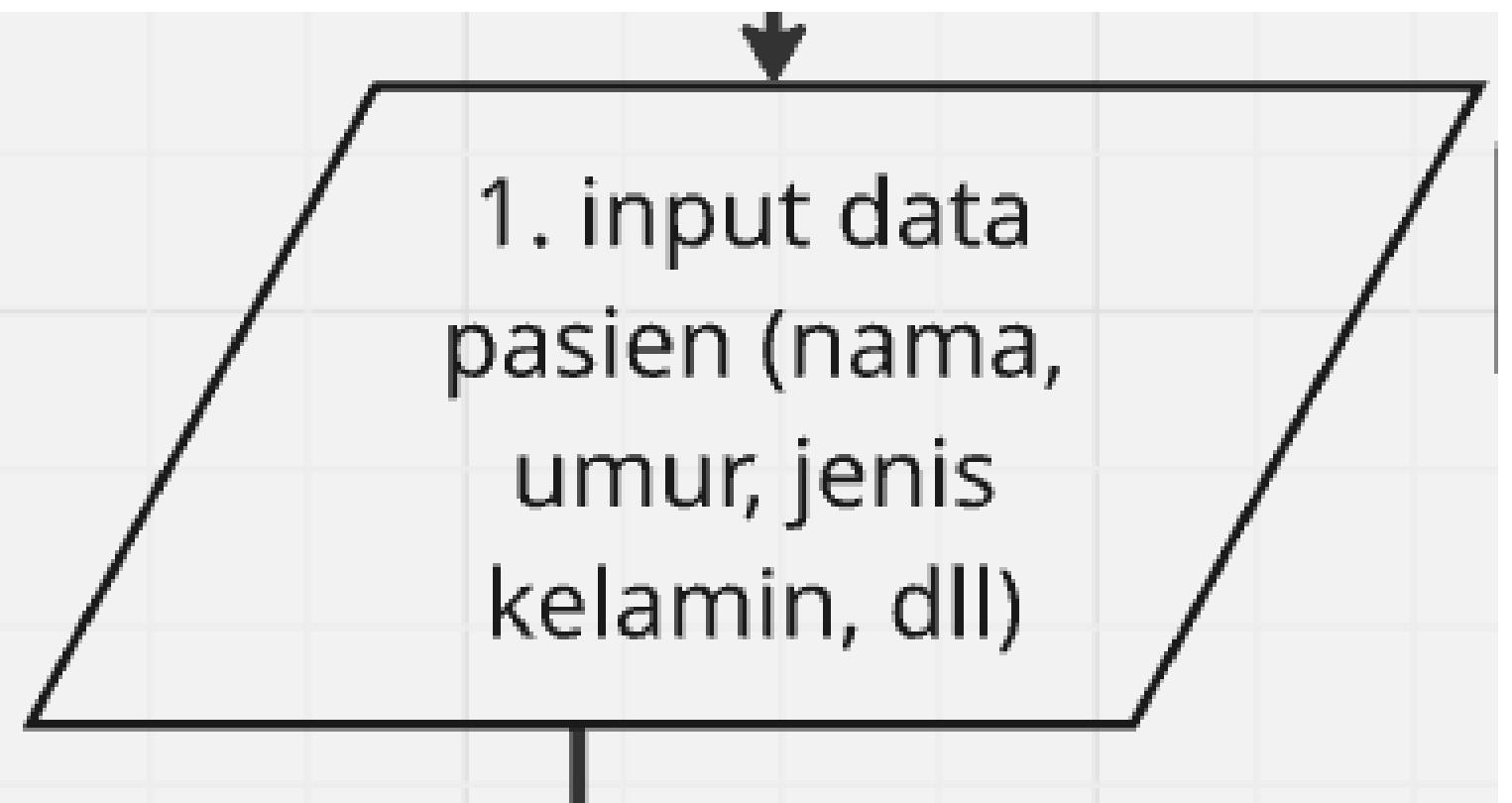
    while (1) {
        if (!bacaStringLower("Status Vaksin HPV (belum/lengkap/b/l): ", buffer, sizeof(buffer))) continue;
        if (strcmp(buffer, "belum") == 0 || strcmp(buffer, "b") == 0) {
            strcpy(p_baru->status_vaksin, "belum");
            break;
        } else if (strcmp(buffer, "lengkap") == 0 || strcmp(buffer, "l") == 0) {
            strcpy(p_baru->status_vaksin, "lengkap");
            break;
        } else {
            printf("Input tidak valid. Ketik 'belum'/'b' atau 'lengkap'/'l'.\n");
        }
    }

    p_baru->dosis_vaksin = 0;
    if (strcmp(p_baru->status_vaksin, "belum") == 0) {
        p_baru->dosis_vaksin = bacaInteger("Jumlah dosis vaksin HPV yg sudah diterima (0-3): ", 0, 3);
    } else {
        p_baru->dosis_vaksin = 3; // jika lengkap, otomatis 3 dosis
    }

    // Gunakan bacaYaTidak
    p_baru->hamil = 0;
    if (strcmp(p_baru->jenis_kelamin, "wanita") == 0) {
        p_baru->hamil = bacaYaTidak("Apakah pasien sedang hamil?");
    }

    p_baru->allergi_berat_vaksin = bacaYaTidak("Apakah ada riwayat alergi berat thd komponen vaksin (ragi)?");
    p_baru->nsakit_berat_saat_ini = bacaYaTidak("Apakah sedang sakit NRAT saat ini?");
    p_baru->isunokompromais = bacaYaTidak("Apakah pasien isunokompromais (misal, HIV)?");
    p_baru->gangguan_darah = bacaYaTidak("Apakah pasien memiliki gangguan pembekuan darah?");

    jumlah_pasien++;
    printf("\nData pasien %s berhasil ditambahkan.\n", p_baru->nama);
    // Tidak otomatis cek detail setelah tambah, user bisa cek via menu baru
    // cekSyaratVaksin("p_baru");
}
```



Flowchart and Code

Cek Syarat



```
void cekSyaratVaksin(Pasien p) {
    printf("--- Cek Syarat Vaksin HPV untuk %s ---\n", p.nama);

    // Cek paling awal: Jika sudah lengkap, selesai.
    if (strcmp(p.status_vaksin, "lengkap") == 0) {
        printf("[OK] Vaksin HPV sudah lengkap (berdasarkan data).\n");
        printf("Kesimpulan: Vaksinasi HPV sudah lengkap.\n");
        printf("-----\n");
        return; // Tidak perlu cek syarat lain jika sudah lengkap
    }

    // Jika belum lengkap, lanjutkan pengecekan syarat
    int layak = 1;
    int rekomendasi_usia_ditemukan = 0;

    if (p.alergi_berat_vaksin == 1) {
        printf("[X] Tidak disarankan: Riwayat alergi berat thd komponen vaksin (ragi).\n");
        layak = 0;
    }
    if (p.sakit_berat_saat_ini == 1) {
        printf("[X] Tunda: Sedang sakit berat. Vaksinasi setelah sembuh.\n");
        layak = 0;
    }
    if (strcmp(p.jenis_kelamin, "wanita") == 0 && p.hamil == 1) {
        printf("[X] Tunda: Sedang hamil. Vaksinasi setelah melahirkan.\n");
        layak = 0;
    }

    // Cek Usia (hanya relevan jika belum lengkap)
    int usia_min_rekomendasi = 9;
    int usia_max_rekomendasi = 26;
    int usia_max_catchup = 45;

    if (p.umur >= usia_min_rekomendasi && p.umur <= usia_max_rekomendasi) {
        printf("[OK] Usia (%d thn) masuk rentang rekomendasi utama (9-26 thn).\n", p.umur);
        rekomendasi_usia_ditemukan = 1;
    } else if (p.umur > usia_max_rekomendasi && p.umur <= usia_max_catchup) {
        printf("[!] Usia (%d thn) di luar rekomendasi utama, masuk rentang catch-up (27-45 thn).\n", p.umur);
        printf(" -> Pertimbangkan vaksinasi setelah diskusi dg dokter.\n");
        rekomendasi_usia_ditemukan = 1;
    } else if (p.umur > usia_max_catchup) {
        printf("[X] Usia (%d thn) melebihi batas usia catch-up (45 thn).\n", p.umur);
        layak = 0; // Usia tidak memenuhi syarat
    } else { // p.umur < usia_min_rekomendasi
        printf("[X] Usia (%d thn) di bawah batas usia minimal (9 thn).\n", p.umur);
        layak = 0; // Usia tidak memenuhi syarat
    }

    // Cek Dosis (hanya jika usia memungkinkan dan belum lengkap)
    if (rekomendasi_usia_ditemukan) { // Status 'belum' sudah pasti dari cek awal
        if (p.dosis_vaksin < 3) {
            printf("[!] Perlu melanjutkan/memulai vaksinasi HPV (Dosis ke-%d).\n", p.dosis_vaksin + 1);
        } else {
            // Seharusnya tidak terjadi jika status 'belum' tapi dosis 3
            printf("[INFO] Telah menerima 3 dosis, status seharusnya 'lengkap'. Harap perbarui data.\n");
        }
    }
}
```

```
if (p.imunokompromais == 1) {
    printf("[INFO] Pasien imunokompromais (HIV): mungkin perlu jadwal/dosis khusus.\n");
}
if (p.gangguan_darah == 1) {
    printf("[INFO] Pasien dg gangguan pembekuan darah: perlu perhatian khusus saat injeksi.\n");
}

// Kesimpulan (hanya jika belum lengkap)
if (layak && rekomendasi_usia_ditemukan) {
    printf("Kesimpulan: Vaksinasi HPV dapat dilanjutkan/dimulai.\n");
} else if (!layak) {
    printf("Kesimpulan: Vaksinasi HPV tidak disarankan atau perlu ditunda saat ini.\n");
}
printf("-----\n");
```

output dari syarat pasien
dan menambahkan ke
database

```
jumlah_pasien++;
printf("\nData pasien '%s' berhasil ditambahkan.\n", p_baru->nama);
// Tidak otomatis cek detail setelah tambah, user bisa cek via menu baru
// cekSyaratVaksin(*p_baru);
```

Flowchart and Code

2. Tampilkan
data pasien

Menampilkan data
pasien secara tabel
dan keseluruhan

```
void tampilanSemuaPasien() {
    printf("\n--- Daftar Semua Pasien (%d orang) ---\n", jumlah_pasien);
    if (jumlah_pasien == 0) {
        printf("Belum ada data pasien.\n");
        return;
    }
    // Menyesuaikan lebar tabel untuk kolom baru
    printf("-----\n");
    printf("No | %-*s | Umur | Sex | Status HPV | Dosis | Layak Vaksin? | Hamil | Alergi Ragi | Sakit Berat | Imuno | Ggn Darah | Riwayat Penyakit/Alergi\n", PANJANG_NAMA, "Nama");
    printf("-----\n");
    for (int i = 0; i < jumlah_pasien; i++) {
        const char* status_kelayakan = getKelayakanStatus(daftar_pasien[i]);
        printf("%2d | %-*s | %3d | %-3s | %-10s | %-5d | %-13s | %-5s | %-11s | %-5s | %-9s | %s/%s\n",
            i + 1,
            PANJANG_NAMA, daftar_pasien[i].nama,
            daftar_pasien[i].umur,
            (strcmp(daftar_pasien[i].jenis_kelamin, "pria") == 0) ? "P" : "W",
            daftar_pasien[i].status_vaksin,
            daftar_pasien[i].dosis_vaksin,
            status_kelayakan, // Kolom baru
            (strcmp(daftar_pasien[i].jenis_kelamin, "wanita") == 0) ? (daftar_pasien[i].hamil ? "Ya" : "Tidak") : "-",
            daftar_pasien[i].alergi_berat_vaksin ? "Ya" : "Tidak",
            daftar_pasien[i].sakit_berat_saat_ini ? "Ya" : "Tidak",
            daftar_pasien[i].imunokompromais ? "Ya" : "Tidak",
            daftar_pasien[i].gangguan_darah ? "Ya" : "Tidak",
            daftar_pasien[i].riwayat_penyakit, daftar_pasien[i].riwayat_alergi);
    }
    printf("-----\n");
    printf("Keterangan: Sex(P/W), Status HPV(belum/lengkap), Layak Vaksin?(BISA/TIDAK/TUNDA/Lengkap), Hamil(Ya/Tidak/-), Alergi Ragi(Ya/Tidak), Sakit Berat(Ya/Tidak), Imuno(Ya/Tidak), Ggn Darah(Ya/Tidak)\n");
}
```

Flowchart and Code

3. Tampilkan menu edit

```
lo {
    printf("\n--- Menu Edit untuk %s ---\n", p->nama);
    // Tampilkan data yang bisa diedit
    printf("1. Nama: %s\n", p->nama);
    printf("2. Umur: %d\n", p->umur);
    printf("3. Jenis Kelamin: %s\n", p->jenis_kelamin);
    printf("4. Riwayat Penyakit: %s\n", p->riwayat_penyakit);
    printf("5. Riwayat Alergi: %s\n", p->riwayat_alergi);
    printf("6. Status Vaksin HPV: %s\n", p->status_vaksin);
    printf("7. Jumlah Dosis Vaksin: %d (Hanya relevan jika status 'belum')\n", p->dosis_vaksin);
    if (strcmp(p->jenis_kelamin, "wanita") == 0) {
        printf("8. Status Hamil: %s\n", p->hamil ? "Ya" : "Tidak");
    } else {
        printf("8. Status Hamil: (Tidak Berlaku untuk Pria)\n");
    }
    printf("9. Riwayat Alergi Berat Vaksin (Ragi): %s\n", p->alergi_berat_vaksin ? "Ya" : "Tidak");
    printf("10. Status Sakit Berat Saat Ini: %s\n", p->sakit_berat_saat_ini ? "Ya" : "Tidak");
    printf("11. Status Imunokompromais (HTV): %s\n", p->imunokompromais ? "Ya" : "Tidak");
    printf("12. Status Gangguan Pembekuan Darah: %s\n", p->gangguan_darah ? "Ya" : "Tidak");
    printf("13. Kembali ke Menu Utama\n");

    pilihan_edit = bacaInteger("Pilih data yang ingin diubah (1-13): ", 1, 13);

    switch (pilihan_edit) {
        case 1: // Edit Nama
            printf("Masukkan Nama Baru: ");
            fgets(p->nama, PANJANG_NAMA, stdin);
            p->nama[strcspn(p->nama, "\n")] = 0;
            printf("Nama berhasil diubah.\n");
            break;
        case 2: // Edit Umur
            p->umur = bacaInteger("Masukkan Umur Baru: ", 0, 150);
            printf("Umur berhasil diubah.\n");
            break;
        case 3: // Edit Jenis Kelamin (Perlu hati-hati jika mempengaruhi status hamil)
            printf("Peringatan: Mengubah jenis kelamin dapat mereset status hamil jika diubah ke pria.\n");
            while (1) {
                if (!bacaStringLower("Masukkan Jenis Kelamin Baru (pria/wanita/p/w): ", buffer, sizeof(buffer))) continue;
                if (strcmp(buffer, "pria") == 0 || strcmp(buffer, "p") == 0) {
                    strcpy(p->jenis_kelamin, "pria");
                    p->hamil = 0; // Reset hamil jika jadi pria
                    printf("Jenis kelamin berhasil diubah menjadi pria (status hamil direset).\n");
                    break;
                } else if (strcmp(buffer, "wanita") == 0 || strcmp(buffer, "w") == 0) {
                    strcpy(p->jenis_kelamin, "wanita");
                    // Status hamil tidak diubah, user harus edit manual jika perlu
                    printf("Jenis kelamin berhasil diubah menjadi wanita.\n");
                    break;
                } else {
                    printf("Input tidak valid. Ketik 'pria'/'p' atau 'wanita'/'w'.\n");
                }
            }
            break;
    }
}
```



```
case 5: // Edit Riwayat Alergi
    printf("Masukkan Riwayat Alergi Baru (jika tidak ada, ketik '-'): ");
    fgets(p->riwayat_alergi, PANJANG_INFO, stdin);
    p->riwayat_alergi[strcspn(p->riwayat_alergi, "\n")] = 0;
    printf("Riwayat alergi berhasil diubah.\n");
    break;

case 6: // Edit Status Vaksin HPV
    while (1) {
        if (!bacaStringLower("Masukkan status vaksin HPV baru (belum/lengkap/b/l): ", buffer, sizeof(buffer))) continue;
        if (strcmp(buffer, "belum") == 0 || strcmp(buffer, "b") == 0) {
            strcpy(p->status_vaksin, "belum");
            // Tanya dosis jika status 'belum'
            p->dosis_vaksin = bacaInteger("Masukkan jumlah dosis yang sudah diterima (0-3): ", 0, 3);
            printf("Status vaksin dan dosis berhasil diubah.\n");
            break;
        } else if (strcmp(buffer, "lengkap") == 0 || strcmp(buffer, "l") == 0) {
            strcpy(p->status_vaksin, "lengkap");
            p->dosis_vaksin = 3; // Otomatis 3 jika lengkap
            printf("Status vaksin berhasil diubah menjadi 'lengkap' (dosis otomatis 3).\n");
            break;
        } else {
            printf("Input tidak valid. Ketik 'belum'/'b' atau 'lengkap'/'l'.\n");
        }
    }
    break;

case 7: // Edit Jumlah Dosis Vaksin
    if (strcmp(p->status_vaksin, "belum") == 0) {
        p->dosis_vaksin = bacaInteger("Masukkan jumlah dosis baru (0-3): ", 0, 3);
        printf("Jumlah dosis berhasil diubah.\n");
    } else {
        printf("Tidak dapat mengubah dosis jika status vaksin adalah 'lengkap'. Ubah status ke 'belum' terlebih dahulu jika perlu.\n");
    }
    break;

case 8: // Edit Status Hamil
    if (strcmp(p->jenis_kelamin, "wanita") == 0) {
        p->hamil = bacaYaTidak("Masukkan status hamil baru");
        printf("Status hamil berhasil diubah.\n");
    } else {
        printf("Tidak dapat mengubah status hamil untuk pasien pria.\n");
    }
    break;

case 9: // Edit Alergi Berat Vaksin
    p->alergi_berat_vaksin = bacaYaTidak("Masukkan status riwayat alergi berat vaksin (ragi) baru");
    printf("Status alergi berat vaksin berhasil diubah.\n");
    break;

case 10: // Edit Status Sakit Berat
    p->sakit_berat_saat_ini = bacaYaTidak("Masukkan status sakit berat baru");
    printf("Status sakit berat berhasil diubah.\n");
    break;

case 11: // Edit Status Imunokompromais
    p->imunokompromais = bacaYaTidak("Masukkan status imunokompromais baru");
    printf("Status imunokompromais berhasil diubah.\n");
    break;

case 12: // Edit Status Gangguan Darah
```

```
case 12: // Edit Status Gangguan Darah
    p->gangguan_darah = bacaYaTidak("Masukkan status gangguan pembekuan darah baru");
    printf("Status gangguan darah berhasil diubah.\n");
    break;

case 13: // Kembali
    printf("Selesai mengedit data %s. Kembali ke menu utama.\n", p->nama);
    break;

default:
    printf("Pilihan edit tidak valid.\n");
    break;
}
```

Flowchart and Code



• • •

• • •

• • •

• • •

• • •

• • •

• • •

• • •



• • •

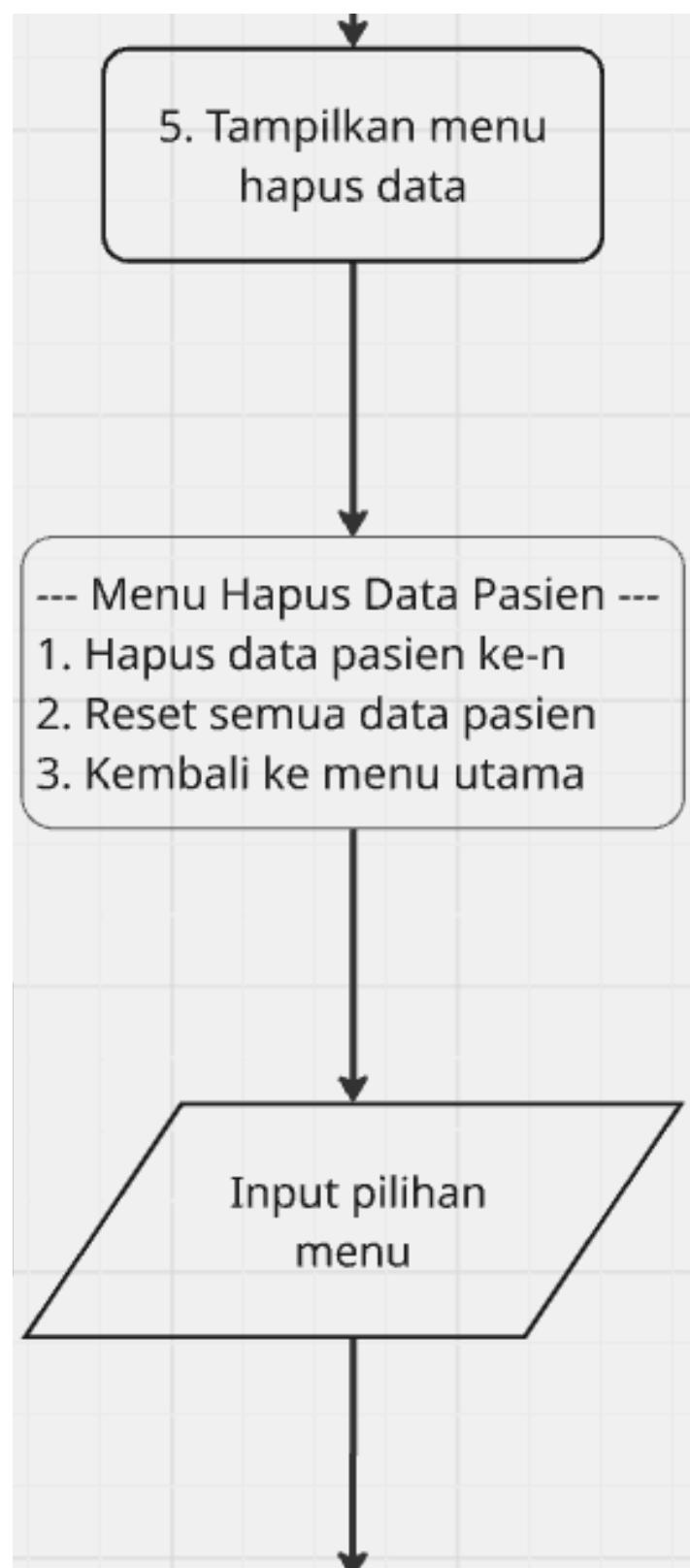
• • •

• • •

• • •



Flowchart and Code



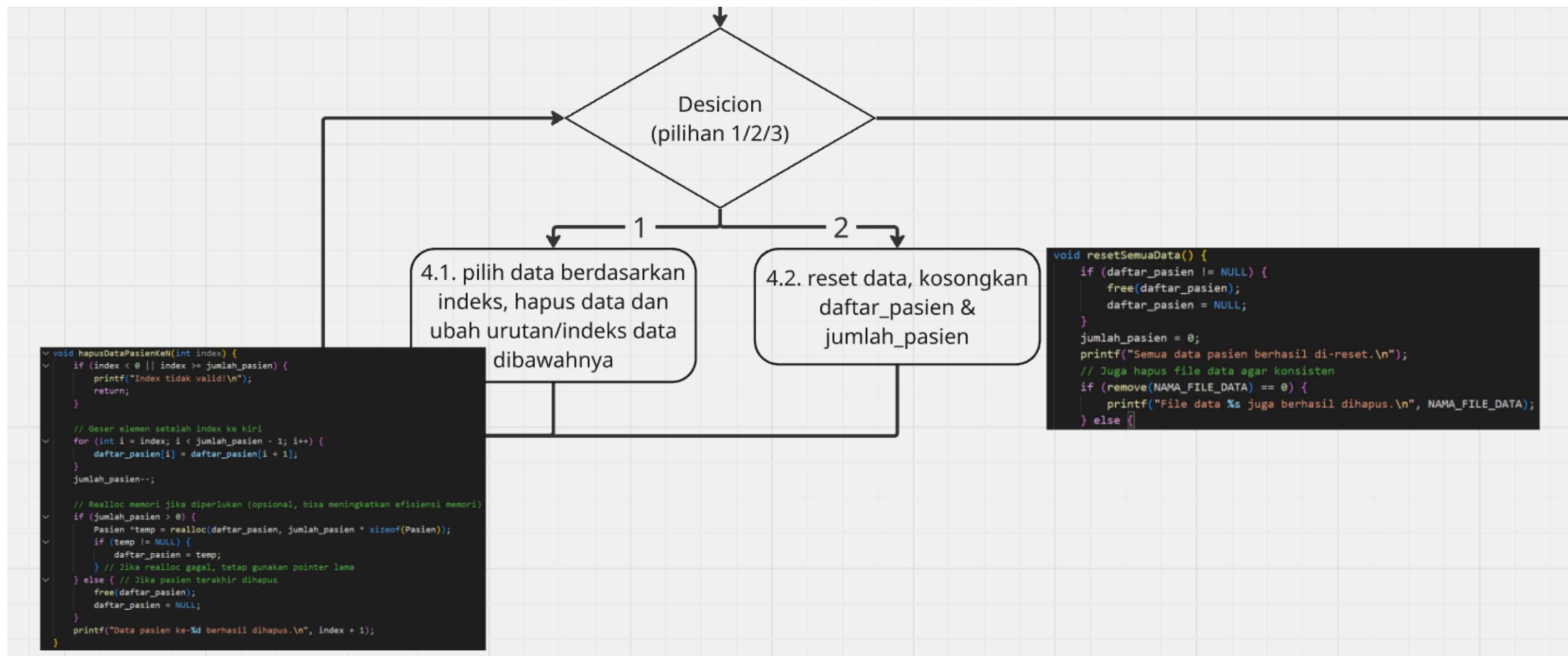
```
void hapusDataPasien() { // Tidak perlu passing argumen, gunakan variabel global
    int pilihan_hapus;
    do {
        printf("\n--- Menu Hapus Data Pasien ---\n");
        printf("1. Hapus data pasien ke-n\n");
        printf("2. Reset semua data pasien\n");
        printf("3. Kembali ke menu utama\n");

        pilihan_hapus = bacaInteger("Pilih opsi (1-3): ", 1, 3);

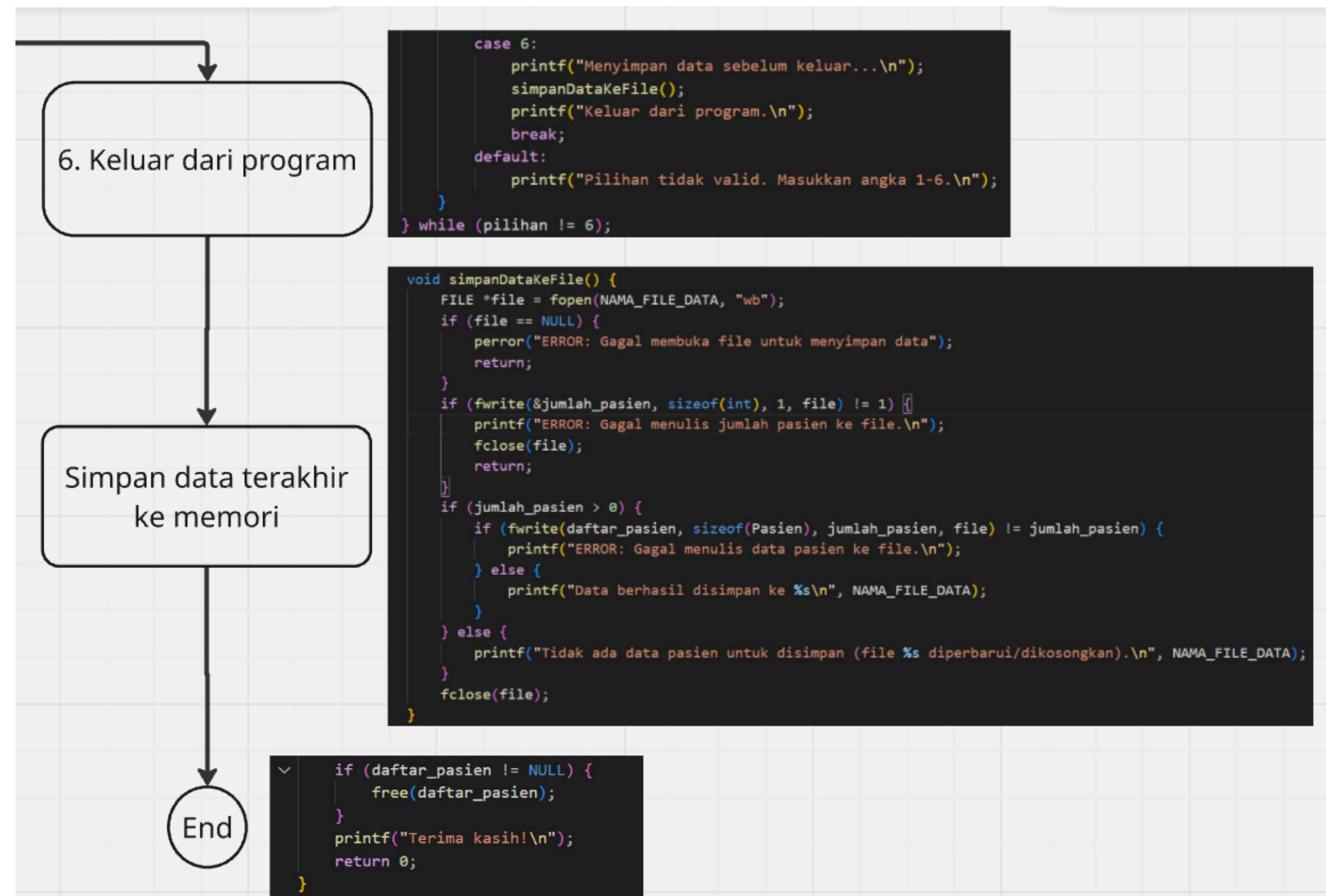
        switch (pilihan_hapus) {
            case 1: { // Hapus pasien ke-n
                if (jumlah_pasien == 0) {
                    printf("Tidak ada data pasien untuk dihapus.\n");
                    break;
                }
                printf("\nDaftar Pasien:\n");
                for (int i = 0; i < jumlah_pasien; i++) {
                    printf("%d. %s\n", i + 1, daftar_pasien[i].nama);
                }
                int nomor = bacaInteger("Masukkan nomor pasien yang akan dihapus: ", 1, jumlah_pasien);
                hapusDataPasienKeN(nomor - 1); // Panggil fungsi helper
                break;
            }
            case 2: // Reset semua data
                if (jumlah_pasien > 0) {
                    int konfirmasi = bacaYaTidak("APAKAH ANDA YAKIN ingin mereset SEMUA data pasien? Tindakan ini tidak dapat dibatalkan.");
                    if (konfirmasi) {
                        resetSemuaData();
                    } else {
                        printf("Reset data dibatalkan.\n");
                    }
                } else {
                    printf("Tidak ada data untuk di-reset.\n");
                }
                break;
            case 3: // Kembali ke menu utama
                printf("Kembali ke menu utama...\n");
                break;
            default:
                printf("Pilihan tidak valid.\n");
                break;
        }
    } while (pilihan_hapus != 3);
}
```

The screenshot shows a terminal window titled "Menu hapus data" displaying the Java code for managing patient data deletion. The code defines a method `hapusDataPasien()` that displays a menu with three options: 1. Delete patient data, 2. Reset all patient data, or 3. Return to the main menu. The code handles each option by printing a list of patients, prompting for a selection, and then calling helper methods like `hapusDataPasienKeN()` or `resetSemuaData()`. The code uses a `do-while` loop to keep the menu active until the user selects option 3.

Flowchart and Code



Flowchart and Code





Code

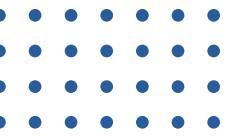
The screenshot shows a terminal window with a dark background and light-colored text. At the top left are three colored dots (red, yellow, green). At the top right is the text "Finpro". The code itself is a C program. It includes standard library headers and defines constants for string lengths and buffer sizes. It then defines a struct for patient data, including fields for name, age, gender, medical history, allergies, vaccination status, and various health indicators. Finally, it initializes pointers to dynamic memory and sets an initial count to zero.

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h> // Untuk tolower()

// Struktur Data Pasien
#define PANJANG_NAMA 30
#define PANJANG_INFO 50
#define PANJANG_BUFFER 100 // Buffer untuk input string
#define NAMA_FILE_DATA "pasien_data.bin"

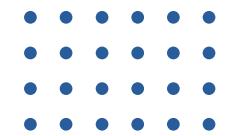
typedef struct {
    char nama[PANJANG_NAMA];
    int umur;
    char jenis_kelamin[10]; // pria/wanita
    char riwayat_penyakit[PANJANG_INFO];
    char riwayat_alergi[PANJANG_INFO];
    char status_vaksin[10]; // belum/lengkap
    int dosis_vaksin;
    int hamil; // 0 = Tidak, 1 = Ya
    int alergi_berat_vaksin; // 0 = Tidak, 1 = Ya (alergi ragi)
    int sakit_berat_saat_ini; // 0 = Tidak, 1 = Ya
    int imunokompromais; // 0 = Tidak, 1 = Ya (HIV)
    int gangguan_darah; // 0 = Tidak, 1 = Ya
} Pasien;

// Penyimpanan Data Dinamis
Pasien *daftar_pasien = NULL;
int jumlah_pasien = 0;
```





Referensi



As a library, the National Library of Medicine (NLM) provides access to scientific literature.

Inclusion in an NLM database does not imply endorsement of, or agreement with, the contents by NLM or the National Institutes of Health.



Updated clinical guideline for human papillomavirus vaccine: the Korean Society of Gynecologic Oncology...

Since the human papillomavirus (HPV) vaccine guidelines were developed by the Korean Society of Gynecologic Oncology (KSGO) in 2011, 2016, and 2019, several recent studies on the efficacy and safety of HPV...

 [PubMed Central \(PMC\)](#)

- • • •



THANK YOU

