

***CODE CONVENTION* Pengerjaan Aplikasi  
CINEMATIQUE Tugas Besar  
Mata Kuliah Struktur Data dan Algoritma**



**Disusun oleh:**

241524012 Ivan Setiawan Ihsan

241524017 Muhammad Arrafi

241524018 Muhammad Nabil Syauqi Rasyiq

**Jurusan Teknik Komputer dan Informatika  
Program Studi Sarjana Terapan Teknik Informatika  
Politeknik Negeri Bandung  
2025**

## DAFTAR ISI

DAFTAR ISI.....	2
1. TUJUAN.....	3
2. GAYA PENULISAN UMUM.....	3
3. ARSITEKTUR CODE.....	3
4. PENAMAAN.....	4
5. FORMAT KODE.....	4
6. KOMENTAR.....	5
7. CONTOH IMPLEMENTASI KODE .....	5
8. PENUTUP.....	6

## 1. TUJUAN

Dokumen ini berisi pedoman standar dalam penulisan kode program Bahasa C, pada proyek bioskop tugas besar mata kuliah Struktur Data dan Algoritma (SDA). Dengan adanya dokumen ini diharapkan developer dapat menciptakan kode yang sesuai dengan butir-butir berikut.

1. **Konsisten:** Kode yang ditulis oleh setiap developer memiliki gaya yang seragam, mengurangi kebingungan dan mempermudah pemahaman.
2. **Mudah Dibaca:** Struktur dan gaya penulisan kode yang jelas akan meningkatkan keterbacaan, baik bagi penulis maupun developer lain.
3. **Mudah Dipelihara dan Dikembangkan:** Kode yang bersih dan terstruktur akan lebih mudah diperbaiki, diubah, atau ditambahkan fitur baru oleh banyak orang secara kolaboratif.

## 2. GAYA PENULISAN UMUM

Gaya penulisan kode untuk program bioskop ini akan diatur sedemikian rupa sehingga didapatkan hasil code yang rapih dan jelas. Berikut butir-butir aturan penulisan kode:

1. Indentasi menggunakan 1 tab
2. Kurung kurawal dibuka di baris baru (Allman Style), contohnya pada pembuatan sebuah baris kode logic if else, kurung kurawal tidak disimpan disebelah blok kondisional, akan tetapi, dibaris berikutnya.
3. Panjang fungsi maksimal 30 baris, pengecualian bagi pembuatan tampilan program.
4. Menghindari penggunaan nested selection ataupun repition lebih dari 4 tingkat.
5. Penulisan kode disarankan menggunakan Bahasa Indonesia.
6. Berikan jarak 1 baris antar fungsi
7. Berikan komentar deskripsi untuk setiap fungsi dibagian atas nama fungsi.
8. Penulisan memperhatikan abjad, pastikan dalam satu bagian yang sama kode dapat disusun secara ascending.

## 3. ARSITEKTUR CODE

Proyek bioskop ini dibangun menggunakan arsitektur kode MVC (model, view, controller) agar kode yang dihasilkan mudah dipahami, dikembangkan, dan dipelihara. Berikut uraian terkait tanggung jawab masing-masing MVC.

1. Model → bertanggung jawab untuk menangani data dan logika program.
2. View → bertanggung jawab untuk menampilkan data yang diberikan oleh Model ke pengguna dalam bentuk antarmuka.

3. Controller → Bertindak sebagai penghubung antara model dan view, menangani permintaan pengguna, memproses data, dan memenuhi kebutuhan *user stories*.

## 4. PENAMAAN

Konsistensi penulisan dalam program ini juga diatur, sehingga memudahkan untuk kolaborasi antar developer. Hal-hal yang diatur antara lain adalah gaya penulisan variabel, fungsi dan lain sebagainya. Lihat tabel x.x untuk aturan penamaannya.

NO	JENIS	GAYA	CONTOH
1	Variabel	snake_case	nama_pengguna
2	Fungsi	snake_case	get_nama_pengguna()
3	Konstanta	UPPER_CASE	TYPE_PROVINSI
4	Struct	PascalCase	Negara, EventBioskop
5	File	Lowercase + _	tree_data.c, linked.c

## 5. FORMAT KODE

Butir-butir penting dalam format penulisan kode.

1. Menggunakan Allman Style untuk kurung kurawal

```
if (kondisi)
{
    // aksi
}
else
{
    // aksi alternatif
}
```

2. Gunakan default pada setiap penggunaan switch-case

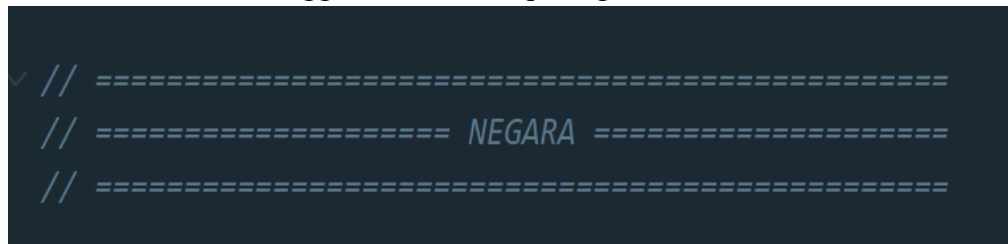
```
switch (menu)
{
    case 1:
        tampilkan_film();
        break;
    case 2:
        pesan_tiket();
        break;
    default:
        printf("Pilihan tidak valid.\n");
        break;
}
```

3. Hindari penulisan kata atau kalimat yang sama, bila kalimat akan digunakan berulang kali, ubah menjadi fungsi tersendiri, contohnya kalimat “Alokasi memori gagal” akan digunakan pada setiap line of code yang mencoba untuk mengalokasikan memori, ubah menjadi sebuah fungsi baru bernama “**display\_gagal\_alokasi()**”.
4. Berikan jarak 1 spasi antara blok logic dengan if, switc, for, dan lain-lain.

## 6. KOMENTAR

Berikut butir-butir yang perlu diperhatikan dalam pembuatan kode.

1. Penulisan komentar dilakukan diatas kode atau bagian yang dijelaskan
2. Penulisan komentar untuk deskripsi fungsi dilakukan diluar kurung kurawal tepat diatas nama function.
3. Penulisan komentar pendek menggunakan ‘//’ dan untuk komentar Panjang yang lebih dari 1 line menggunakan ‘/\* ... \*/’.
4. Penamaan section menggunakan format pada gambar berikut:



```
// =====  
// ===== NEGARA =====  
// =====
```

5. Setiap function wajib diberikan komentar dibagain definisinya menggunakan format inisial state, final state, dan deskripsi.

## 7. CONTOH IMPLEMENTASI KODE

```
void linked_jadwal_to_film(address node, void* context)
{
    if (get_tipe_node(node) != TYPE_STUDIO) return;

    List* list_film = (List*)context;

    List* jadwal_list = get_jadwal_studio(info_studio(node));
    pnode current_jadwal;
    for ( current_jadwal = First(*jadwal_list); current_jadwal != NULL;
current_jadwal = Next(current_jadwal)) {
        Jadwal* jadwal = info_jadwal(current_jadwal);
        if (!jadwal || is_jadwal_linked(jadwal)) continue;

        pnode current_film;
        for ( current_film = First(*list_film); current_film != NULL;
current_film = Next(current_film))
        {
            Film* film = info_film(current_film);
            const char* judul_film = get_judul_film(film);
```

```
    const char* nama_jadwal = get_film_name(jadwal);
    if (!judul_film || !nama_jadwal) continue;

    if (strcmp(judul_film, nama_jadwal) == 0)
    {
        set_film_object(jadwal, film);
        break;
    }
}
}
```

## 8. PENUTUP

Dokumen ini merupakan panduan awal yang dapat terus dikembangkan dan diperbarui seiring perkembangan proyek. Harapannya seluruh anggota tim dapat mengikuti panduan ini demi kelancaran kolaborasi, kemudahan pemeliharaan, serta keberhasilan proyek secara keseluruhan.