

ANALYSIS AND PREDICTION

DATA SCIENCE AND ANALYTICS

- 1 ALASAN MEMILIH TOPIC
- 2 FITUR & MAPPING
- 3 STRUKTUR & TYPE DATA
- 4 CONTOH INPUT & OUTPUT





1. Bersifat edukatif karena tidak hanya menjalankan perhitungan tapi juga melatih kemampuan berhitung pengguna melalui kuis,
2. Selain itu programm ini menggunakan prinsip dasar seperti percabangan, perulangan & fungsi.
3. Pengenalan Dasar Keamanan Data (Enkripsi Sederhana)
jawaban yang dimasukkan oleh pengguna tidak langsung diproses dalam bentuk aslinya, tapi diubah dulu menjadi bentuk lain (disamarkan).

ITERASI/MAPPING

- Data type & Algortihm
→ Penggunaan tipe data int dan char, algoritma soal matematika acak, serta Caesar Cipher (enkripsi & dekripsi jawaban).
- Looping/Perulangan
→ Perulangan for untuk jumlah soal kuis dan do-while untuk pengulangan menu.
- Percabangan (Decision Making)
→ Percabangan if-else untuk menentukan operasi soal (+, -, ×) dan mengecek jawaban benar atau salah.



STRUKTUR DAN TIPE DATA

1. String manipulation

Fungsi di game:

Input nama pemain → disimpan ke file → dienkripsi/dekripsi.

Menggunakan char array, strcpy(), strlen(), dan Caesar Cipher untuk mengubah karakter.

Tujuan:

Latihan mengolah data string.

Memungkinkan kita mengubah karakter satu per satu (contoh: enkripsi nama).

3. if/selection

Menentukan operator soal.

Memilih menu (Main Game / Lihat High Score / Keluar).

Mengecek jawaban pemain → benar atau salah

2. Iteration (loop)

Fungsi di game:

Loop untuk memberi soal kuis (misal 5 soal).

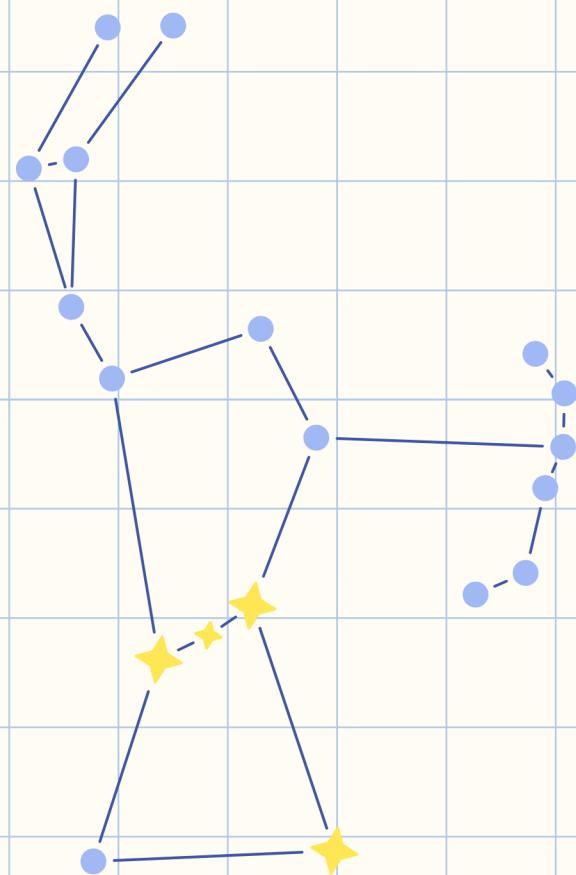
Loop untuk menampilkan leaderboard (mengurutkan skor). Tujuan:

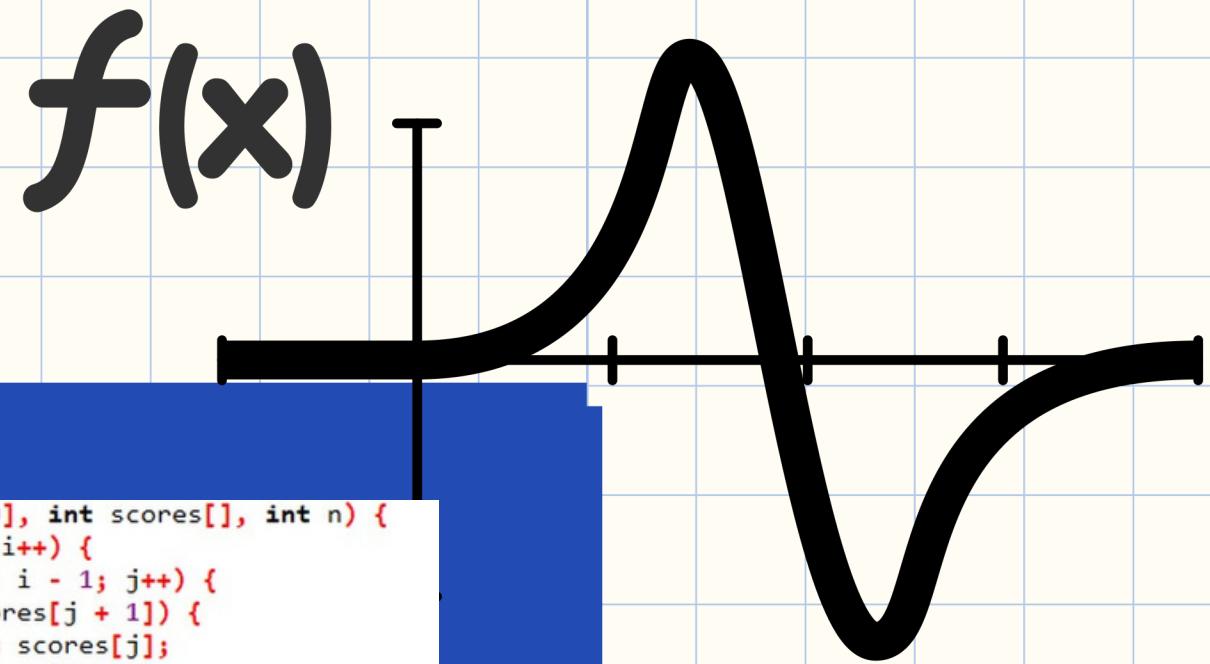
Mengurangi kode duplikat, otomatisasi proses berulang.

Latihan logika perulangan.

4. Data Jawaban Disimpan dalam Bentuk Terenkripsi (caesar cipher)

→ Jawaban pengguna diproses menggunakan Caesar Cipher sebelum didekripsi kembali untuk pengecekan.





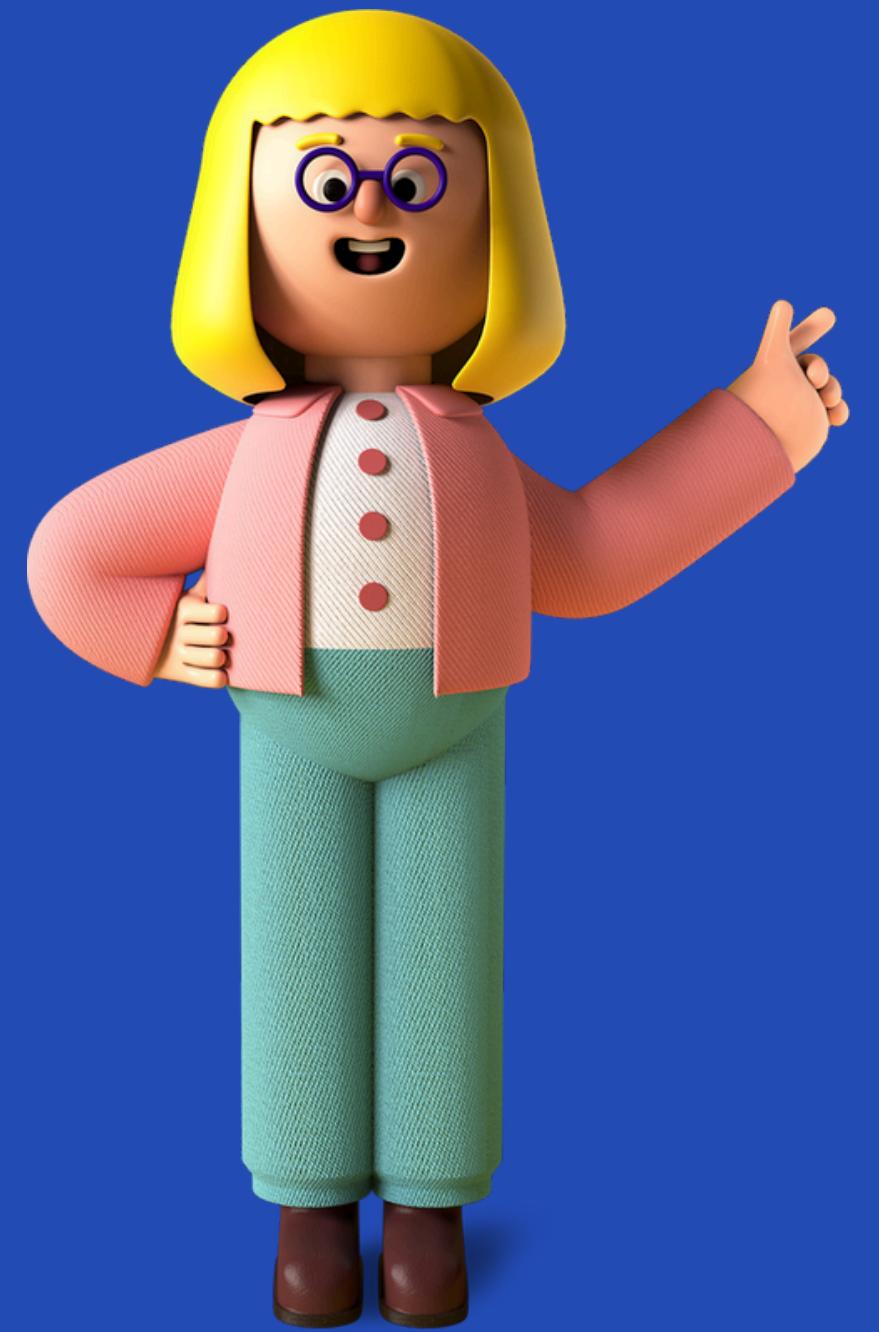
```
void sortScores(char names[][50], int scores[], int n) {
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (scores[j] < scores[j + 1]) {
                int tempScore = scores[j];
                scores[j] = scores[j + 1];
                scores[j + 1] = tempScore;

                char tempName[50];
                strcpy(tempName, names[j]);
                strcpy(names[j], names[j + 1]);
                strcpy(names[j + 1], tempName);
            }
        }
    }
}
```

```
// Fungsi untuk menambah skor ke file
void saveScore(char name[], int score) {
    FILE *fp = fopen("scores.txt", "a");
    fprintf(fp, "%s %d\n", name, score);
    fclose(fp);
}

// Fungsi membaca file dan menyimpan ke array
int readScores(char names[][50], int scores[]) {
    FILE *fp = fopen("scores.txt", "r");
    if (!fp) return 0;

    int count = 0;
    while (fscanf(fp, "%s %d", names[count], &scores[count]) != EOF) {
        count++;
    }
    fclose(fp);
    return count;
}
```



*contoh input
& output*

