

**LAPORAN UJIAN AKHIR SEMESTER KONSEP PEMROGRAMAN  
MATH GAMES ADVENTURES**



**ANGGOTA KELOMPOK :**

1. Farrell Luthfi Adriansyah (L0125098)
2. Muhammad Haidar Amru (L0125025)
3. Satrio Unggul Prayogo (L0125114)

**PROGRAM STUDI INFORMATIKA  
FAKULTAS TEKNOLOGI DAN SAINS DATA  
UNIVERSITAS SEBELAS MARET**

**2025**

# DAFTAR ISI

<b>DAFTAR ISI.....</b>	<b>1</b>
<b>BAB I.....</b>	<b>2</b>
<b>PENDAHULUAN.....</b>	<b>2</b>
1.1. Penjelasan Umum.....	2
1.2. Library yang Digunakan.....	2
1.3. Alur Kerja.....	3
<b>BAB II.....</b>	<b>4</b>
<b>SOURCE CODE.....</b>	<b>4</b>
2.1. Penjelasan Source Code.....	4
<b>BAB III.....</b>	<b>26</b>
<b>ALUR PROGRAM.....</b>	<b>26</b>
3.1. Cara Menjalankan Program.....	26
<b>BAB IV.....</b>	<b>34</b>
<b>PEMBAGIAN TUGAS.....</b>	<b>34</b>
4.1. Pembagian Tugas dalam Kelompok.....	34
<b>BAB V.....</b>	<b>36</b>
<b>KESIMPULAN.....</b>	<b>36</b>
5.1. Kesimpulan.....	36
5.2. Saran.....	37
5.3. Refleksi.....	37

# BAB I

## PENDAHULUAN

### 1.1. Penjelasan Umum

Proyek ini bertujuan untuk merancang permainan edukasi interaktif bertema petualangan matematika dan logika untuk meningkatkan kemampuan berhitung, logika, dan pemecahan masalah melalui pendekatan permainan. Permainan ini menggabungkan elemen pendidikan dengan hiburan untuk meningkatkan minat dari pengguna dalam belajar karena suasana belajar menjadi lebih menarik, menyenangkan, dan tidak membosankan

Permainan ini bertemakan petualangan di gua misterius. Pengguna diceritakan masuk ke dalam gua misterius dan bertemu dengan jin. Jin memberikan pilihan apabila ingin keluar dari gua misterius, maka harus menyelesaikan tantangan matematika yang diberikan oleh jin. Pemain harus menjawab seluruh soal matematika yang diberikan oleh jin dalam 3 babak. Pemain diberikan 3 nyawa dan akan berkurang apabila pemain salah menjawab soal.

Seperti yang telah dijelaskan sebelumnya, terdapat 3 babak dalam permainan kali ini. Terdapat perbedaan jenis dan jumlah soal di setiap babak. Pada babak pertama, terdapat 10 soal matematika dasar aritmatika. Pada babak kedua, terdapat 7 soal matematika tingkat sedang pada materi persamaan linear satu variabel dan perhitungan luas dan volume bangun. Pada babak ketiga atau terakhir, terdapat 5 soal matematika tingkat sedang dengan format soal cerita, meliputi penerapan materi jarak, debit, diskon, dan rata-rata.

### 1.2. Library yang Digunakan

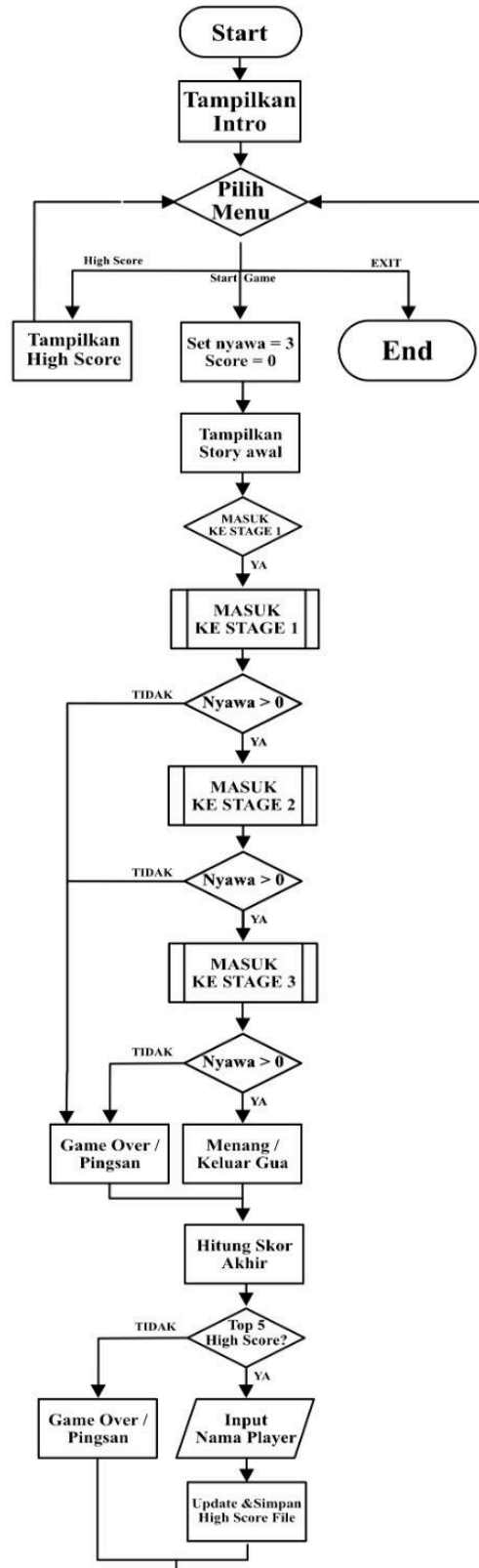
```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <string.h>
#include <windows.h>
```

- #include <stdio.h>

Sebagai standar input output, wajib ada agar fungsi seperti printf, scanf, fopen, dll. bisa digunakan.

- `#include <string.h>`  
Sebagai string handling, digunakan untuk manipulasi teks seperti strcpy.
- `#include <stdlib.h>`  
Standard library, digunakan untuk manajemen memori (malloc) dan kontrol sistem.
- `#include <time.h>`  
Library untuk memanipulasi tanggal dan waktu sistem, akan digunakan untuk srand(time(NULL))
- `#include <windows.h>`  
Windows API, library khusus untuk OS Windows. Digunakan di program ini untuk mengakses fungsi Sleep().

### 1.3. Alur Kerja



## BAB II

### SOURCE CODE

#### 2.1. Penjelasan Source Code

##### 2.1.1. Header, Konstanta, dan Variabel Global

###### A. Konstanta (#define)

```
#define FILE_HIGHSCORE "highscore.txt"
#define MAX_SOAL_2 8
#define MAX_SOAL_3 5
#define PANJANG_SOAL 512
#define LEBAR_KOTAK 70
```

- #define FILE\_HIGHSCORE  
Mendefinisikan nama file untuk menyimpan data skor pemain sebagai sebuah konstanta. Digunakan jika suatu saat kita mengganti nama file, cukup mengganti di baris define saja.
- #define MAX\_SOAL\_2  
Mendefinisikan maksimal soal untuk soal pada stage 2 yang dimana stage 2 berisi sebanyak 8 soal berupa soal aljabar dan geometri
- #define MAX\_SOAL 3  
Mendefinisikan maksimal soal untuk soal pada stage 3 yang dimana stage 3 berisi sebanyak 5 soal berupa soal cerita.
- #define PANJANG\_SOAL 512  
Mendefinisikan maksimal panjang soal dalam karakter.
- #define LEBAR\_KOTAK 70  
Mendefinisikan lebar space yang digunakan untuk mencetak soal yaitu 70 karakter

## B. Variabel Global

```
// Variabel global untuk skor dan nyawa
int lives = 3;
int score = 0;
int death = 0;
```

- Integer `lives` yang di awal dideklarasikan bernilai 3. Nilai ini akan berkurang setiap user menjawab pertanyaan dengan salah. Nilai ini juga akan direset menjadi 3 lagi ketika pemain memulai permainan baru.
- Integer `score` yang di awal dideklarasikan bernilai 0. Nilai ini dapat bertambah ataupun berkurang tergantung bagaimana user menjawab pertanyaan. Nilai akan direset menjadi 0 lagi ketika pemain memulai permainan baru.
- Integer `death` yang di awal akan dideklarasikan bernilai 0. Nilai ini adalah *flag* (penanda) yang menunjukkan apakah pemain sudah pernah kalah sebelumnya.

## C. Struct

```
struct highScore {
    char username[50];
    int score;
}; //Struct untuk highscore

struct highScore topLeaderboard[5];
```

- Mendefinisikan format untuk satu entri data High Score, di mana setiap entri akan berisi username (maksimal 50 karakter) dan score dari pemain
- Bagian `struct highScore topLeaderboard[5]` adalah array yang menampung 5 entri data (top 5 skor) dengan format `struct highScore`. Array ini nantinya akan diisi oleh `loadHighScore` dan file akan diurutkan oleh `updateHighScore`.

### 2.1.2. Utility & Interface

#### A. Manajemen Waktu

```
//Fungsi delay
void delay(int milliseconds) {
    Sleep(milliseconds);
}
```

Menyimpan fungsi sleep() bawaan dari Windows. Fungsi ini menjeda program selama beberapa milidetik, memberikan efek transisi yang halus.

#### B. Animasi Teks

```
// Fungsi untuk mencetak teks per karakter (animasi)
void ketikPelan(char *teks, int kecepatan) {
    // Loop sampai akhir string (karakter '\0')
    while (*teks != '\0') {
        printf("%c", *teks);
        fflush(stdout);
        delay(kecepatan);
        teks++;
    }
}
```

Fungsi untuk memberikan efek animasi mengetik huruf per huruf.

Menggunakan pointer \*teks untuk melihat string dari karakter demi karakter.

Fflush(stdout) digunakan agar fungsi memaksa huruf segera muncul di layar tanpa menunggu baris baru.

#### C. Pembersihan Layar

```
//Fungsi clear screen
void clearScreen() {
    system("cls");
}
```

Menyimpan perintah system("cls") yang berfungsi agar membersihkan layar terminal agar tampilan menu selalu rapi.

#### D. Pembersihan Buffer Input

```
25 //Fungsi clear buffer
26 void clearInputBuffer() {
27     int c;
28     while ((c = getchar()) != '\n' && c != EOF);
29 }
```



- Menutupi masalah yang sering terjadi pada scanf, yakni meninggalkan karakter newline di dalam buffer input setelah membaca angka, jika tidak dibersihkan maka fungsi input berikutnya akan membaca sisa enter tersebut.
- Fungsi ini melakukan perulangan sederhana, ia akan terus menerus mengambil getchar dan membuang karakter yang tersisa di memori sementara sampai ditemukan karakter baris baru atau akhir file.

### 2.1.3. Sistem High Score & Manajemen File

#### A. Fungsi Save High Score

```
//Fungsi menyimpan skor
void saveHighScore(){
    FILE *fp = fopen(FILE_HIGHSORE, "w");
    if (fp == NULL) {
        printf("Error opening file for writing high scores.\n");
        return;
    }

    for (int i = 0; i < 5; i++) {
        if (strlen(topLeaderboard[i].username) == 0) {
            fprintf(fp, "EMPTY 0\n");
        }
        else {
            fprintf(fp, "%s %d\n", topLeaderboard[i].username, topLeaderboard[i].score);
        }
    }

    fclose(fp);
}
```

Fungsi ini akan menyimpan data skor yang ada di memori program (array topLeaderboard) ke dalam file highscore.txt di komputer.

- Pertama-tama file akan dibuka dan ditaruh di pointer fp dengan mode w atau write. Lalu akan dicek apakah fp berisi NULL, jika iya maka berarti pembukaan file gagal (misal disk penuh atau akses ditolak).
- Program akan melakukan perulangan sebanyak 5 kali sesuai kapasitas high score.
- Jika panjang string username adalah 0 (berarti slot kosong) maka program akan mencetak string "EMPTY 0" ke dalam file. Ini dilakukan agar format file selalu terdiri dari 5 baris data.

- Jika file sudah ada isinya, maka program baru akan menulis Nama dan Skor asli menggunakan format %s %d diikuti baris baru. Terakhir file akan ditutup dengan fclose.

## B. Fungsi Load High Score

```
void loadHighScore(){
    FILE *fp = fopen(FILE_HIGHSORE, "r");

    // Jika file tidak ada, isi dengan data kosong (EMPTY)
    if (fp == NULL) {
        for(int i=0; i<5; i++){
            strcpy(topLeaderboard[i].username, "EMPTY");
            topLeaderboard[i].score = 0;
        }
        return;
    }

    for (int i = 0; i < 5; i++) {
        fscanf(fp, "%s %d", topLeaderboard[i].username, &topLeaderboard[i].score);
    }
    fclose(fp);
}
```

Fungsi ini akan memuat data skor dari file highscore.txt dan memindahkan ke dalam memori program.

- Pertama-tama file akan dibuka dan ditaruh di pointer fp dengan mode r atau read. Lalu akan dicek apakah fp berisi NULL, jika iya maka berarti program akan mengisi array topLeaderbord dengan data EMPTY dan skor 0. Hal tersebut dilakukan agar program tidak menampilkan data sampah atau crash.
- Jika file berisi, maka akan dilakukan looping sebanyak 5 kali. Fscanf akan membaca isi file berfasarkan format “%s %d”, berarti program akan membaca satu kata string diikuti satu angka integer. Data yang telah dibaca akan dimasukkan ke struct topLeaderboard pada indeks ke-i. Terakhir file akan ditutup dengan fclose.

### C. Fungsi Update High Score

```
//Fungsi update high score
void updateHighScore(int skorBaru) {
    loadHighScore();

    if (skorBaru > topLeaderboard[4].score || strcmp(topLeaderboard[4].username, "EMPTY") == 0) {
        printf("Selamat! Anda mendapatkan High Score baru!\n");

        char namaBaru[50];
        printf("Masukkan username Anda (satu kata): ");
        scanf("%s", namaBaru);

        strcpy(topLeaderboard[4].username, namaBaru);
        topLeaderboard[4].score = skorBaru;

        struct highScore temp;
        for (int i = 0; i < 5 - 1; i++) {
            for (int j = 0; j < 5 - i - 1; j++) {
                if (topLeaderboard[j].score < topLeaderboard[j + 1].score) {
                    temp = topLeaderboard[j];
                    topLeaderboard[j] = topLeaderboard[j + 1];
                    topLeaderboard[j + 1] = temp;
                }
            }
        }
    }
}
```

Fungsi ini bertugas menentukan apakah skor terbaru pemain layak masuk ke dalam daftar 5 besar. Jika layak maka fungsi akan memasukkan data baru, mengurutkan, dan menyimpannya ke file.

- Pertama-tama fungsi akan menyinkronkan data dengan memanggil fungsi loadHighScore. Kemudian program akan memeriksa apakah skor baru berhak masuk leaderboard menggunakan logika if(skorbaru > ...). Logika tersebut akan membandingkan skorBaru dengan juara 5 di array ke 4 (pemegang skor terendah).
- Jika skor baru lebih besar dari juara 5 maka skor baru berhak masuk. Jika juara 5 masih kosong maka skor baru juga diperbolehkan masuk.
- Jika pemain lolos kualifikasi, maka program akan meminta input username dengan scanf dan program akan menimpa data pada indeks ke-4.
- Setelah data baru masuk di indeks ke-4, urutan leaderboard masih berantakan. Program akan mengurutkannya dengan metode bubble sort tipe descending (lihat bagian for bertingkat). Pertama-tama program akan membandingkan dua elemen yang bersebelahan (indeks j dan j+1), jika elemen atas (j) lebih kecil daripada elemen di bawahnya (j+1) maka posisi

mereka ditukar. Proses ini diulang berkali-kali hingga skor terbesar naik ke indeks 0 dan skor terkecil turun ke indeks 4.

```
        saveHighScore();
        printf("High Score berhasil diperbarui!\n");
    } else {
        printf("Maaf, Anda tidak masuk dalam Top 5 High Score.\n");
    }
}
```

Setelah array terurut rapi, maka fungsi akan memanggil `saveHighScore`. Dan kondisi `else` akan mencetak pesan bahwa user tidak layak masuk ke top 5 highscore.

#### D. Fungsi Show Leaderboard

```
//Fungsi menampilkan Leaderbo
void showLeaderboard() {
    loadHighScore();

    printf("==== TOP 5 HIGH SCORE =====\n");
    printf("Rank  %-15s Score\n", "Username");
    printf("-----");

    for (int i = 0; i < 5; i++)
    {
        printf("\n%d.    %-15s %d\n", i + 1, topLeaderboard[i].username, topLeaderboard[i].score);
    }
    printf("-----");
}
```

Fungsi ini bertugas untuk memvisualisasikan data array `topLeaderboard` ke layar terminal dalam bentuk tabel.

- Di awal program akan memanggil fungsi `loadHighScore`, kemudian header tabel akan dicetak. Format bagian ranking akan diberi padding 15 karakter supaya kolom username tidak bertabrakan dengan kolom skor
- Looping sebanyak 5 kali akan dilakukan untuk mencetak nomor urut ( $i+1$ ), lalu username, dan skornya.

## 2.1.4. Logika Level 1

### A. Fungsi Soal Penjumlahan

```
//Fungsi soal penjumlahan
void soalPenjumlahan() {
    if (lives <= 0) {
        return;
    }
    int x, y, jawaban, kunciJawaban;
    x = (rand()% 201 )- 100;
    y = (rand()% 201 )- 100;

    printf ("Hasil dari %d + %d adalah : ",x, y);
    scanf("%d", &jawaban);

    clearInputBuffer();

    kunciJawaban = x + y;
    if (jawaban == kunciJawaban) {
        printf("Selamat jawaban Anda benar!\n");
        score += 1;
    }
    else {
        printf("Maaf jawaban Anda salah. Jawaban yang benar adalah %d\n", kunciJawaban);
        score -= 1;
        lives --;
    }
}
```

- Pertama-tama akan dicek apakah nyawa kurang dari sama dengan 0, apabila ia program akan direturn.
- Jika tidak, maka akan disiapkan variabel x dan y sebagai dua angka yang dijadikan soal, jawaban untuk input jawaban dari user, serta kunciJawaban untuk jawaban yang benar dari soal.
- X dan y akan dibuat menggunakan random generator yang memanfaatkan rand(). Rand akan menghasilkan sebuah angka acak yang sangat besar, dari 0 hingga RAND\_MAX. Bilangan random tersebut akan dicari sisa baginya dengan % 201, artinya angka acak besar tersebut jika dibagi 201 maka sisanya pasti antara 0 sampai 200. Kemudian supaya bisa menampung angka negatif (-100 sampai 100) maka kita kurangkan dengan 100 sisa bagi angka random tadi. Misal hasil modulo adalah 200, berarti angka yang tercetak adalah 100.
- Variabel kunciJawaban akan berisi penjumlahan dari x dan y. Lalu dicek apakah jawaban user dan kunci jawaban cocok dengan logika if. Jika

cocok maka akan dicetak pesan jika jawaban benar dan skor bertambah 1.  
Jika salah maka skor berkurang 1 dan nyawa berkurang 1.

## B. Fungsi Soal Pengurangan

```
//Fungsi soal pengurangan
void soalPengurangan() {
    if (lives <= 0) {
        return;
    }
    int x, y, jawaban, kunciJawaban;
    x = (rand()% 201 )- 100;
    y = (rand()% 201 )- 100;

    printf ("Hasil dari %d - %d adalah : ",x, y);
    scanf("%d", &jawaban);

    clearInputBuffer();

    kunciJawaban = x - y;
    if (jawaban == kunciJawaban) {
        printf("Selamat jawaban Anda benar!\n");
        score += 1;
    }
    else {
        printf("Maaf jawaban Anda salah. Jawaban yang benar adalah %d\n", kunciJawaban);
        score -= 1;
        lives --;
    }
}
```

- Konsep dasarnya sama dengan soal penjumlahan, hanya saja bedanya di sini adalah variabel kunciJawaban berisi pengurangan dari x dan y.

## C. Fungsi Soal Perkalian

```
//Fungsi soal Perkalian
void soalPerkalian() {
    if (lives <= 0) {
        return;
    }
    int x, y, jawaban, kunciJawaban;
    x = (rand()% 21 )- 10;
    y = (rand()% 21 )- 10;

    printf ("Hasil dari %d x %d adalah : ",x, y);
    scanf("%d", &jawaban);

    clearInputBuffer();

    kunciJawaban = x * y;
    if (jawaban == kunciJawaban) {
        printf("Selamat jawaban Anda benar!\n");
        score += 2;
    }
    else {
        printf("Maaf jawaban Anda salah. Jawaban yang benar adalah %d\n", kunciJawaban);
        score -= 1;
        lives --;
    }
}
```

- Konsep dasar juga masih mirip dengan 2 fungsi sebelumnya. Hanya saja x dan y yang dirandom dimod 21 dan dikurangi 10, sehingga angkanya hanya berkisar dari -10 sampai 10.
- Variabel kunci jawaban akan berisi perkalian dari x dan y. Untuk soal perkalian skor akan bertambah 2 jika jawaban benar.

#### D. Fungsi Soal Pembagian.

```
//Fungsi soal pembagian
void soalPembagian() {
    if (lives <= 0) {
        return;
    }
    int x, y, jawaban, kunciJawaban;
    do{
        y = (rand()% 21 )- 10;
    } while (y == 0);

    kunciJawaban = (rand()% 21 )- 10;
    x = y * kunciJawaban;

    printf ("Hasil dari %d : %d adalah : ",x, y);
    scanf("%d", &jawaban);

    clearInputBuffer();
}
```

- Pada dasarnya konsep masih sama, namun di sini agar soal tidak menghasilkan pembagian dengan 0 program akan menggunakan looping dan logika terbalik. Di mana, selama y == 0 maka nilai y akan dirandom terus menerus. Untuk kunci jawaban juga akan dirandomisasi sehingga nanti, nilai x adalah y dikali kunci jawaban. Hal ini akan membuat kunci jawaban bernilai x bagi y.

```
if (jawaban == kunciJawaban) {
    printf("Selamat jawaban Anda benar!\n");
    score += 2;
}
else {
    printf("Maaf jawaban Anda salah. Jawaban yang benar adalah %d\n", kunciJawaban);
    score -= 1;
    lives --;
}
}
```

- Kemudian seperti soal perkalian, akan dicek apakah jawaban cocok dengan kunci jawaban, jika iya maka skor bertambah 2.

### 2.1.5. Logika Level 2

#### A. Deklarasi fungsi stage 2

```
int stage2() {  
    // Array untuk menyimpan soal yang dibaca dari file  
    char questions[MAX_SOAL_2][PANJANG_SOAL];
```

Pada awal deklarasi stage 2, dideklarasikan array bernama question bertipe char dengan banyak soal sesuai dengan definisi MAX\_SOAL\_2 dan panjang maksimal karakter soal sesuai dengan definisi PANJANG\_SOAL.

#### B. Deklarasi array answer

```
// Array untuk menyimpan jawaban yang benar  
char *answers[MAX_SOAL_2] = {  
    "4",      // Jawaban Soal 1  
    "-13",   // Jawaban Soal 2  
    "12",    // Jawaban Soal 3  
    "66",    // Jawaban Soal 4  
    "550",   // Jawaban Soal 5  
    "342",   // Jawaban Soal 6  
    "60",    // Jawaban Soal 7  
    "25",    // Jawaban Soal 8  
};
```

Kemudian, dilakukan inisialisasi array bernama answer dengan tipe pointer char. Array ini berisi seluruh kunci jawaban dari soal pada stage 2.

#### C. Deklarasi array jawabanUser

```
char jawabanUser[50];  
int i;
```



Selanjutnya, dilakukan deklarasi array jawabanUser dengan tipe char untuk menyimpan jawaban yang dimasukkan oleh pemain. Jawaban ini akan dibandingkan dengan kunci jawaban dari array answer.

#### D. Logika membaca soal dari file

```
// --- BACA SOAL DARI FILE ---
FILE *file = fopen("STAGE 2.txt", "r");

if (file == NULL) {
    printf("Error: Tidak bisa membuka file soal.txt.\n");
    printf("Pastikan file soal.txt ada di folder yang sama dengan program ini.\n");
    return 1;
}

for (i = 0; i < MAX_SOAL_2; i++) {
    if (fgets(questions[i], PANJANG_SOAL, file) == NULL) {
        break;
    }
}
fclose(file);
// --- SELESAI BACA FILE ---
```

Program ini dibuat untuk membuka soal dari file txt. Soal yang akan dibuka dan dibaca akan dicetak untuk ditampilkan saat pemain memainkan permainan.

#### E. Logika menjawab soal

```
for (i = 0; i < MAX_SOAL_2; i++) {
    if (lives <= 0) return 0;

    printf("Soal no. %d\n", i + 1);

    // Cetak soal ke dalam interface kotak
    printf("%s", questions[i]);

    printf("Jawaban Anda: ");
    scanf("%s", jawabanUser);
    clearInputBuffer();
    printf("\n");
}
```

Pada awalnya, dengan file yang telah dibaca oleh program, program akan menyalin isi file ke dalam program satu persatu dan diberikan kesempatan pengguna untuk melakukan input jawaban. Setelah pengguna melakukan input jawaban, maka jawaban akan dimasukkan dalam variabel jawabanUser. Setelah itu, dilakukan pembersihan buffer input agar siap dimasukkan input berikutnya.

#### F. Logika pencocokan jawaban

```
if (strcmp(jawabanUser, answers[i]) == 0) {  
    printf("Selamat jawaban Anda benar!\n\n");  
    score += 3;  
} else {  
    printf("Maaf jawaban Anda salah. Jawaban yang benar adalah %s\n\n", answers[i]);  
    score -= 1;  
    lives--;  
}
```

Setelah input jawaban User telah berhasil disimpan, maka akan dilakukan pencocokan jawaban user dengan kunci jawaban. Menggunakan fungsi strcmp, maka input dari jawaban user akan dibandingkan dengan array answer. Apabila sama, maka score akan bertambah 3, sedangkan apabila tidak sama dengan array answer, maka score dan lives akan berkurang 1.

### 2.1.6. Logika Level 3

#### A. Deklarasi Fungsi stage 3

```
int stage3() {  
    // Array untuk menyimpan soal yang dibaca dari file  
    char questions[MAX_SOAL_3][PANJANG_SOAL];  
}
```

Pada awal deklarasi stage 3, dideklarasikan array bernama question bertipe char dengan banyak soal sesuai dengan definisi MAX\_SOAL\_3 dan panjang maksimal karakter soal sesuai dengan definisi PANJANG\_SOAL.

#### B. Deklarasi array answer stage 3

```
char *answers[MAX_SOAL_3] = {  
    "3",           // Jawaban Soal 1  
    "105000",      // Jawaban Soal 2  
    "175",         // Jawaban Soal 3  
    "90",          // Jawaban Soal 4  
    "1400"         // Jawaban Soal 5  
};
```

lalu, dilakukan inisialisasi array bernama answer dengan tipe pointer char. Array ini berisi seluruh kunci jawaban dari soal pada stage 3.

### C. Deklarasi array jawabanUser

```
char jawabanUser[50];  
int i;
```

Berikutnya, dilakukan deklarasi array jawabanUser dengan tipe char untuk menyimpan jawaban yang dimasukkan oleh pemain. Jawaban ini akan dibandingkan dengan kunci jawaban dari array answer.

### D. Logika membaca soal dari file

```
// --- BACA SOAL DARI FILE ---  
FILE *file = fopen("STAGE 3.txt", "r");  
  
if (file == NULL) {  
    printf("Error: Tidak bisa membuka file soal.txt.\n");  
    printf("Pastikan file soal.txt ada di folder yang sama dengan program ini.\n");  
    return 1;  
}  
  
for (i = 0; i < MAX_SOAL_3; i++) {  
    if (fgets(questions[i], PANJANG_SOAL, file) == NULL) {  
        break;  
    }  
}  
fclose(file);  
// --- SELESAI BACA FILE ---
```

Program ini dibuat untuk membuka soal dari file txt. Soal yang akan dibuka dan dibaca akan dicetak untuk ditampilkan saat pemain memainkan permainan.

### E. Logika Permainan

```
for (i = 0; i < MAX_SOAL_3; i++) {  
    // Cek nyawa  
    if (lives <= 0) return 0;  
  
    printf("Soal no. %d\n", i + 1);  
  
    //untuk merapikan di terminal  
    for (int k = 0; questions[i][k] != '\0'; k++) {  
        if (questions[i][k] == '|') {  
            printf("\n");  
        } else {  
            printf("%c", questions[i][k]);  
        }  
    }  
    printf("\n");  
  
    printf("Jawaban Anda: ");  
    scanf("%s", jawabanUser);  
    clearInputBuffer();  
    printf("\n");  
}
```

Pada awalnya, dengan file yang telah dibaca oleh program, program akan menyalin isi file ke dalam program satu persatu dan diberikan kesempatan pengguna untuk melakukan input jawaban. Setelah pengguna melakukan input jawaban, maka jawaban akan dimasukkan dalam variabel jawabanUser. Setelah itu, dilakukan pembersihan buffer input agar siap dimasukkan input berikutnya.

#### F. Logika Pencocokan Jawaban dan Penampil Skor Akhir

```
if (strcmp(jawabanUser, answers[i]) == 0) {
    printf("Selamat jawaban Anda benar!\n\n");
    score += 5;
} else {
    printf("Maaf jawaban Anda salah. Jawaban yang benar adalah %s\n\n", answers[i]);
    score -= 1;
    lives--;
}

delay(1000);
}

if (lives > 0) {
    printf("\nSelamat! Kamu berhasil menyelesaikan semua stage dengan skor: %d\n", score);
}

return 0;
}
```

Setelah input jawaban User telah berhasil disimpan, maka akan dilakukan pencocokan jawaban user dengan kunci jawaban. Menggunakan fungsi strcmp, maka input dari jawaban user akan dibandingkan dengan array answer. Apabila sama, maka score akan bertambah 5, sedangkan apabila tidak sama dengan array answer, maka score dan lives akan berkurang 1. Lalu program akan menampilkan output berisi ucapan selamat serta skor akhir player ketika nyawa player masih tersisa.

## 2.1.7. Main Program

### A. Loading Screen

```
int main() {  
  
    ketikPelan("                \n", 1);  
    ketikPelan(" | \\\\/ | / \\\\/ | _||| | | / _|| / \\\\/ | \\\\/ | | _|| / _|| \n", 1);  
    ketikPelan(" | \\\\/ | / _ \\\\/ | | | | | | | / _ \\\\/ | \\\\/ | | _|| \\\\/ \\\\/ \n", 1);  
    ketikPelan(" | | | / _ \\\\/ | | | | | | | / _ \\\\/ | | | | | _|| \n", 1);  
    ketikPelan(" | | | / / \\\\/ \\\\/ | | | | | \\\\/ / / \\\\/ \\\\/ | | | | | _|| / \n", 1);  
    ketikPelan("                \n", 5);  
    ketikPelan(" / \\\\/ | _ \\\\/ \\\\/ \\\\/ / / | _|| \\\\/ | | | | | _ \\\\/ | _|| \n", 5);  
    ketikPelan(" / _ \\\\/ | | | | | \\\\/ \\\\/ / / | _|| \\\\/ | | | | | | | | | _|| \n", 5);  
    ketikPelan(" / _ \\\\/ | | | | | \\\\/ \\\\/ / | _|| \\\\/ | | | | | | | | | _ <| _|| \n", 5);  
    ketikPelan("/ / \\\\/ \\\\/ _|| \\\\/ | _|| \\\\/ | | | | | \\\\/ / | | | | | \n", 5);  
    printf("\n"); // Baris kosong biasa (tidak perlu animasi)  
  
    ketikPelan("                BY Kelompok 8                \n", 20);  
    ketikPelan("                Version 1.0.0 - December 2025                \n", 20);  
    printf("\n");  
  
    ketikPelan("Press Enter to continue...", 50);  
    getchar();  
    clearScreen();  
  
    int pilihan;  
    char pilUlang;  
    srand(time(NULL));  
}
```

- Program akan menampilkan tulisan ASCII dari “MATH GAMES ADVENTURE” yang diberi animasi `ketikPelan(teks, kecepatan)`. Kemudian akan dicetak juga tulisan kelompok 8 dan versi dari program dengan `ketikPelan`.
- Lalu user perlu mengetik enter untuk menuju main utama. Program akan mendeteksi enter lalu `clearScreen` dijalankan untuk membersihkan layar
- Kemudian akan dideklarasikan variabel `pilihan` untuk pilihan menu, dan `pilUlang` untuk kembali ke menu atau tidak. `Srand(time(NULL))` juga digunakan untuk menginisialisasi seed pengacak angka agar angka yang dihasilkan selalu beda setiap kali program berjalan.

## B. Main Menu

```
do
{
    ketikPelan("=====\\n", 10);
    ketikPelan("          GAME STARTED          \\n", 10);
    ketikPelan("=====\\n", 10);

    printf("Silakan pilih menu dibawah ini:\\n");
    printf("1. Mulai Game\\n");
    printf("2. High Score\\n");
    printf("3. Exit\\n");

    printf("\\nPilihan Anda: ");
    scanf("%d", &pilihan);

    clearScreen();
}
```

- Menu akan ditampilkan dengan headernya diberi animasi ketikpelan lalu ditampilkan pilihan 1-3.
- User akan diminta untuk menginputkan pilihannya yang akan disimpan dalam variabel pilihan. Terakhir, fungsi clearScreen dijalankan.
- Ada looping di menu ini, di mana selama pilihan tidak sama dengan 3 maka menu akan terus dicetak sampai user memilih menu 3 (exit).

## C. Alur Game

```
switch (pilihan)
{
    case 1:
        // Memulai game
        lives = 3;
        score = 0;

        ketikPelan("Suatu hari anda terbangun di Gua misterius...\\n", 10);
        delay(1000);
        ketikPelan("Tiba-tiba seorang Jin mendatangi anda.\\n", 10);
        ketikPelan("\\nPress Enter to continue...", 50);
        while(getchar() != '\\n');
        getchar();
        clearScreen();
}
```

- Ketika user memilih 1, maka game akan dimulai. Game dimulai dengan nyawa user diatur ke 3 dan skor 0. Lalu program akan menampilkan teks cerita sebelum masuk ke level 1 (ada delay(1000) untuk memberi jeda 1 detik). Setelah user menekan enter maka clearScreen akan dipanggil.

```
//New game
if(death == 0) {

    ketikPelan("Jin : Selamat datang di Gua Misterius!\n", 10);
    ketikPelan("Untuk keluar dari sini, kamu harus menyelesaikan tantangan matematika yang aku berikan.\n", 10);
    ketikPelan("Setiap jawaban yang benar akan memberimu poin, \n", 10);
    ketikPelan("tapi hati-hati, jawaban yang salah akan mengurangi nyawamu! Kamu punya 3 nyawa.\n", 10);

    printf("\nJawabanmu? (y/n): ");
    scanf(" %c", &pilUlang);
    clearScreen();
    if(pilUlang == 'n' || pilUlang == 'N') {
        ketikPelan("Jin: Baiklah, sampai jumpa lain waktu!\n", 10);
        delay(1000);
        clearScreen();
        break;
    }
}
```

- Kemudian akan dicek apabila user pernah kalah sebelumnya, jika tidak (death == 0) maka cerita awal akan dimulai. Jika user menjawab N ketika ditanya sang Jin, maka program akan menampilkan pesan perpisahan dan berhenti. Jika user memilih Y maka program akan lanjut ke level 1

```
//Apabila sudah pernah mati
else {
    ketikPelan("Jin: Kita bertemu lagi nak, apakah kamu siap untuk mengulang tantanganku?\n", 10);
    printf("Jawabanmu? (y/n): ");
    scanf(" %c", &pilUlang);
    clearScreen();
    if(pilUlang == 'n' || pilUlang == 'N') {
        ketikPelan("Jin: 'Baiklah, sampai jumpa lain waktu!'\n", 10);
        delay(1000);
        clearScreen();
        break;
    }
}
```

- Jika user pernah kalah sebelumnya, maka akan ada pesan unik di mana kita bertemu lagi dengan Jin. Untuk pertanyaan Jin, konsepnya sama seperti sebelumnya.

```

//level 1 start
ketikPelan("Game dimulai! Semoga beruntung!\n", 10);
printf("Anda memiliki %d nyawa dan skor awal %d.\n\n", lives, score);
delay(1000);
clearScreen();

printf("Soal No. 1 :\n");
soalPenjumlahan();
delay(1000);

if(lives <= 0) {
    goto cekGameOver;
    delay(1000);
}

```

- Ketika game dimulai, user akan diberitahu dulu berapa nyawa dan skor awla mereka. Kemudian soal no.1 akan dicetak dengan memanggil fungsi soalPenjumlahan. Setiap soal akan dicek, apabila nyawanya kurang dari sama dengan 0 (kalah) maka akan menjalankan goto cekGameOver (akan dibahas nanti).

<pre> printf("\nSoal No. 2 :\n"); soalPenjumlahan(); delay(1000);  if(lives &lt;= 0) {     goto cekGameOver;     delay(1000); }  printf("\nSoal No. 3 :\n"); soalPengurangan(); delay(1000);  if(lives &lt;= 0) {     goto cekGameOver;     delay(1000); }  printf("\nSoal No. 4 :\n"); soalPengurangan(); delay(1000);  if(lives &lt;= 0) {     goto cekGameOver;     delay(1000); } </pre>	<pre> printf("\nSoal No. 5 :\n"); soalPerkalian(); delay(1000);  if(lives &lt;= 0) {     goto cekGameOver;     delay(1000); }  printf("\nSoal No. 6 :\n"); soalPerkalian(); delay(1000);  if(lives &lt;= 0) {     goto cekGameOver;     delay(1000); }  printf("\nSoal No. 7 :\n"); soalPerkalian(); delay(1000);  if(lives &lt;= 0) {     goto cekGameOver;     delay(1000); } </pre>	<pre> printf("\nSoal No. 8 :\n"); soalPembagian(); delay(1000);  if(lives &lt;= 0) {     goto cekGameOver;     delay(1000); }  printf("\nSoal No. 9 :\n"); soalPembagian(); delay(1000);  if(lives &lt;= 0) {     goto cekGameOver;     delay(1000); }  printf("\nSoal No. 10 :\n"); soalPembagian(); delay(1000);  if(lives &lt;= 0) {     goto cekGameOver;     delay(1000); } </pre>
--	--	---

- User harus menjawab soal nomor 1-10 dengan nomor 1 dan 2 berupa penjumlahan, nomor 3 dan 4 pengurangan, nomor 5 sampai 7 perkalian, dan 8 sampai 10 pembagian. Setiap soal dijawab akan dicek terus apabila nyawa user telah habis.



```

clearScreen();
printf("\nSelamat! Level 1 Selesai.\n");
printf("\nPress enter to continue...");
getchar();
clearScreen();

ketikPelan("Jin : Selamat sudah menyelesaikan level pertama!\n", 10);
ketikPelan("Namun perjalananmu belum selesai, masih ada tantangan yang lebih sulit\n", 10);
ketikPelan("Apakah kamu siap untuk melanjutkan?", 10);
printf("\nPress enter to continue...");
getchar();
clearScreen();

//Level 2 start
ketikPelan("Level 2 dimulai!\n", 10);
printf("Anda memiliki %d nyawa dan skor %d.\n", lives, score);
delay(2000);
clearScreen();

```

- Jika user menyelesaikan level 1 tanpa kehabisan nyawa, maka cerita akan berlanjut dan user harus menghadapi level 2. Sebelum dimulai, akan ditampilkan lagi nyawa dan skor user saat ini.

```

stage2();
delay(2000);

if(lives <= 0) {
    goto cekGameOver;
    delay(1000);
}

clearScreen();
printf("\nSelamat! Level 2 Selesai.\n");
printf("\nPress enter to continue...");
getchar();
clearScreen();

ketikPelan("Jin : Hebat, kamu sudah menyelesaikan level kedua!\n", 10);
ketikPelan("Sampailah kamu di level terakhir, semangat!\n", 10);
printf("\nPress enter to continue...");
getchar();
clearScreen();

```

- Fungsi stage2 akan dipanggil, dan jika user akan dicek jika nyawanya habis atau belum di soal ke-berapapun itu.
- Setelah menyelesaikan soal level 2 tanpa kehabisan nyawa, maka story akan berlanjut ke babak final, level 3.

```
//level 3 start
ketikPelan("Level 3 dimulai!\n", 10);
printf("Anda memiliki %d nyawa dan skor %d.\n", lives, score);
delay(2000);
clearScreen();

stage3();
delay(2000);
clearScreen();
```

- Level 3 dimulai dengan menampilkan skor dan nyawa user. Kemudian fungsi stage3 akan dipanggil.

```
cekGameOver:
delay(1000);
clearScreen();

if(lives > 0) {
    ketikPelan("Jin : Nampaknya kamu telah menyelesaikan seluruh tantanganku.\n", 5);
    ketikPelan("Selamat! Pintu keluar Gua ada di belakangku.", 5);
    delay(2000);
    clearScreen();

    printf("=====\n");
    printf(" ANDA KELUAR DARI GUA DENGAN SELAMAT \n");
    printf("=====\n");
} else {
    printf("Kamu pingsan dan terbangun kembali di awal gua.\n");
    death = 1;
}

printf("Skor Akhir Anda: %d\n", score);
updateHighScore(score);

printf("\nPress enter to return to menu...");
getchar();
clearScreen();
break;
```

- Setelah menyelesaikan level 3, maka akan memasuki cekGameOver. Jika user sempat kalah di level 1, 2, ataupun 3, maka program akan melompat ke bagian ini.
- Dijalankan logika sederhana, di mana jika nyawa user masih lebih dari 0, maka pesan selamat karena telah menyelesaikan game ditampilkan. Jika nyawa user kurang dari 0, maka akan muncul pesan kalah dan variabel death menjadi 1.

- Kemudian akan dicetak skor akhir kita, lalu dipanggil fungsi updateHighScore untuk mengecek apakah user layak masuk ke top 5. Game berhenti dan user bisa menekan enter untuk kembali ke menu.

#### D. Cek High Score dan Exit Program

```
case 2:
    clearScreen();
    showLeaderboard();
    printf("\nPress enter to return to menu...");
    while(getchar() != '\n');
    getchar();
    clearScreen();

    break;
```

- Ketika user memilih menu 2 maka akan clearScreen akan dipanggil lalu fungsi showLeaderBoard akan dipanggil juga. Jika sudah melihat highscorenya, maka user bisa menekan enter untuk kembali ke menu

```
case 3:
    ketikPelan("Terima kasih telah bermain! Sampai jumpa lagi!\n", 10);
    printf("\nPress enter to continue...");
    while(getchar() != '\n');
    getchar();
    clearScreen();
    exit(0);
    break;

default:
    ketikPelan("Pilihan tidak valid. Silakan coba lagi.\n", 10);
    break;
}

} while (pilihan != 3);
return 0;
}
```

- Ketika user memilih menu 3, akan ditampilkan pesan perpisahan dan program akan selesai ketika user menekan enter.
- Jika user memasukkan angka 1 dan 2 maka do while akan terus berjalan. Jika user memasukkan selain 1, 2 dan 3, maka akan muncul pesan pilihan tidak valid.

## BAB III

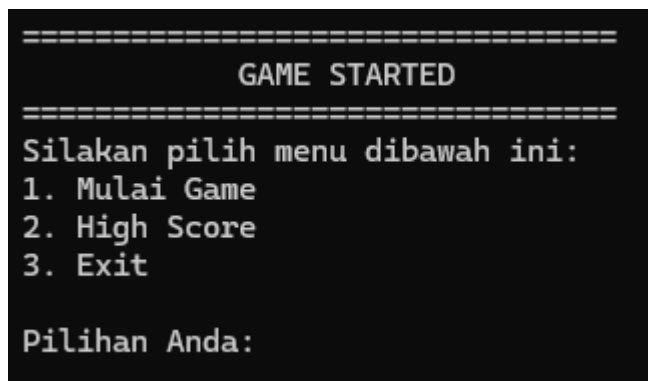
### ALUR PROGRAM

#### 3.1. Cara Menjalankan Program

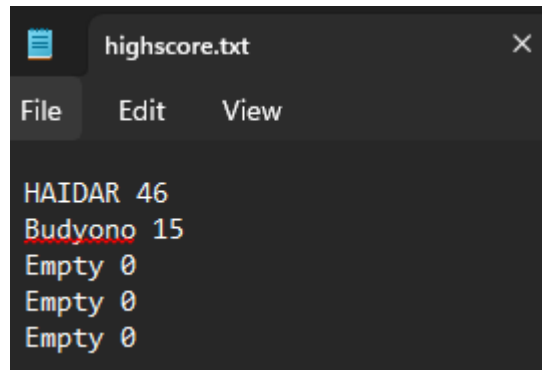
##### 3.1.1. Loading Screen



##### 3.1.2. Menu Awal



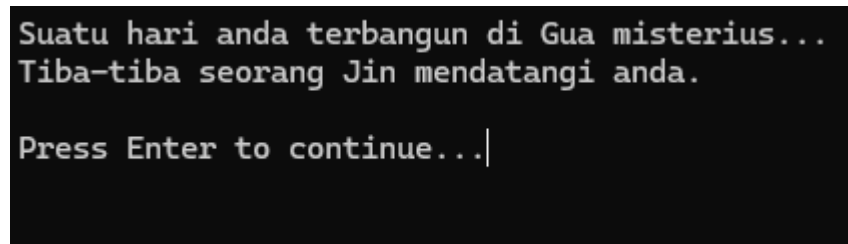
### 3.1.3. Isi High Score Saat Ini



```
highscore.txt
File Edit View
HAIDAR 46
Budyono 15
Empty 0
Empty 0
Empty 0
```

===== TOP 5 HIGH SCORE =====		
Rank	Username	Score
-----		
1.	HAIDAR	46
2.	Budyono	15
3.	Empty	0
4.	Empty	0
5.	Empty	0
-----		
Press enter to return to menu...		

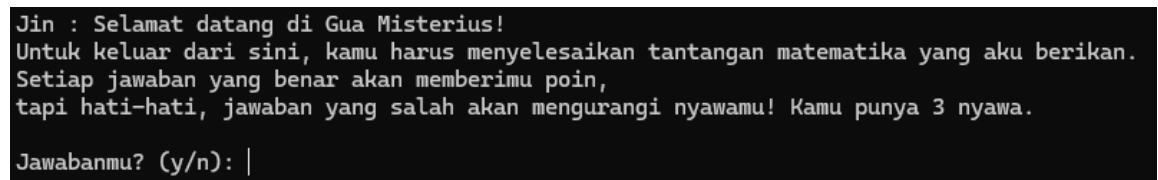
### 3.1.4. Mulai Game (opsi 1)



```
Suatu hari anda terbangun di Gua misterius...
Tiba-tiba seorang Jin mendatangi anda.

Press Enter to continue...|
```

### 3.1.5. Alur Cerita Level 1



```
Jin : Selamat datang di Gua Misterius!
Untuk keluar dari sini, kamu harus menyelesaikan tantangan matematika yang aku berikan.
Setiap jawaban yang benar akan memberimu poin,
tapi hati-hati, jawaban yang salah akan mengurangi nyawamu! Kamu punya 3 nyawa.

Jawabanmu? (y/n): |
```

### 3.1.6. Soal Level 1

```
Game dimulai! Semoga beruntung!  
Anda memiliki 3 nyawa dan skor awal 0.
```

```
Soal No. 1 :  
Hasil dari  $24 + -25$  adalah : -1  
Selamat jawaban Anda benar!  
  
Soal No. 2 :  
Hasil dari  $-38 + -83$  adalah : -121  
Selamat jawaban Anda benar!  
  
Soal No. 3 :  
Hasil dari  $-8 - -45$  adalah : 37  
Selamat jawaban Anda benar!  
  
Soal No. 4 :  
Hasil dari  $-85 - -73$  adalah : -12  
Selamat jawaban Anda benar!  
  
Soal No. 5 :  
Hasil dari  $-7 \times 4$  adalah : -28  
Selamat jawaban Anda benar!
```

```
Soal No. 6 :  
Hasil dari  $-1 \times 3$  adalah : -3  
Selamat jawaban Anda benar!  
  
Soal No. 7 :  
Hasil dari  $-3 \times -9$  adalah : 27  
Selamat jawaban Anda benar!  
  
Soal No. 8 :  
Hasil dari  $8 : 8$  adalah : 1  
Selamat jawaban Anda benar!  
  
Soal No. 9 :  
Hasil dari  $0 : -4$  adalah : 0  
Selamat jawaban Anda benar!  
  
Soal No. 10 :  
Hasil dari  $-6 : 3$  adalah : -2  
Selamat jawaban Anda benar!
```

### 3.1.7. User Menjawab Seluruh Level 1 Dengan Benar

```
Selamat! Level 1 Selesai.  
  
Press enter to continue...|
```

### 3.1.8. Transisi Ke Level 2

```
Jin : Selamat sudah menyelesaikan level pertama!  
Namun perjalananmu belum selesai, masih ada tantangan yang lebih sulit  
Apakah kamu siap untuk melanjutkan?  
Press enter to continue...
```

```
Level 2 dimulai!  
Anda memiliki 3 nyawa dan skor 16.  
|
```

### 3.1.9. Level 2 Dimulai

```
Soal no. 1
tentukan nilai x dari  $2X - 6 = 2$ 
Jawaban Anda: 4

Selamat jawaban Anda benar!

Soal no. 2
tentukan nilai x dari  $7X + 91 = 0$ 
Jawaban Anda: -13

Selamat jawaban Anda benar!

Soal no. 3
tentukan nilai x dari  $5x + 27 = 9X - 21$ 
Jawaban Anda: 12

Selamat jawaban Anda benar!

Soal no. 4
tentukan volume kerucut dengan jari-jari = 3 meter,tinggi = 7 meter
Jawaban Anda: 66

Selamat jawaban Anda benar!

Soal no. 5
tentukan volume tabung dengan tinggi = 7 meter dan diameter = 10 meter
Jawaban Anda: 550

Selamat jawaban Anda benar!
```

```
Soal no. 6
tentukan luas permukaan balok dengan Panjang = 13 meter,tinggi = 7 meter, dan lebar = 4 meter
Jawaban Anda: 342

Selamat jawaban Anda benar!

Soal no. 7
tentukan luas segitiga siku siku dengan alas = 8 meter dan sisi miring = 17 meter
Jawaban Anda: 60

Selamat jawaban Anda benar!

Soal no. 8
tentukan luas persegi dengan panjang sisi = 5 meter
Jawaban Anda: 25

Selamat jawaban Anda benar!
```

### 3.1.10. User Menjawab Seluruh Level 2 Dengan benar

```
Selamat! Level 2 Selesai.

Press enter to continue...|
```

### 3.1.11. Transisi Ke Level 3

```
Jin : Hebat, kamu sudah menyelesaikan level kedua!  
Sampailah kamu di level terakhir, semangat!  
  
Press enter to continue...
```

```
Level 3 dimulai!  
Anda memiliki 3 nyawa dan skor 40.  
|
```

### 3.1.12. Level 3 Dimulai

```
Soal no. 1  
Jarak rumah Candra ke kampus adalah 2 km.  
Setiap akan ke kampus, Candra mengendarai sepeda motor dengan kecepatan rata-rata 40 km/jam.  
Waktu yang dibutuhkan Candra untuk sampai ke kampus adalah ... menit.  
  
Jawaban Anda: 3  
  
Selamat jawaban Anda benar!  
  
Soal no. 2  
Ani pergi ke suatu pusat perbelanjaan untuk mencari baju baru.  
Ia tertarik pada sebuah kemeja dengan harga Rp. 150.000.  
Pada toko tersebut, terdapat diskon untuk pembelian sebuah baju, sebesar 30%.  
Uang yang harus dikeluarkan ani untuk membeli kemeja tersebut adalah... (jawaban tanpa titik)  
  
Jawaban Anda: 105000  
  
Selamat jawaban Anda benar!  
  
Soal no. 3  
Pipa air di suatu tempat mempunyai 354 cabang saluran untuk keperluan sehari-hari.  
Setiap rumah terpasang 1 saluran air.  
Dalam satu hari, pemakaian air mencapai 61.950 liter.  
Rata-rata air yang digunakan di setiap rumah per hari adalah ....  
  
Jawaban Anda: 175  
  
Selamat jawaban Anda benar!
```

```
Soal no. 4  
Jono mendapatkan nilai 78, 87, 82, dan 93 untuk 4 mata kuliah.  
Nilai yang harus diperoleh mata kuliah yang kelima agar nilai rata-rata Jono pada semester tersebut 86 adalah...  
  
Jawaban Anda: 90  
  
Selamat jawaban Anda benar!  
  
Soal no. 5  
Dalam sekali panen, Ahmad mampu mendapatkan 2 ton udang yang baik.  
30% dari hasil tangkapannya ternyata kurang baik untuk dijual.  
Jumlah hasil panen udang Ahmad yang layak dijual adalah .... kg.  
Jawaban Anda: 1400  
  
Selamat jawaban Anda benar!  
  
Selamat! Kamu berhasil menyelesaikan semua stage dengan skor: 65
```

### 3.1.13. Ending Screen dan Pencatatan High Score

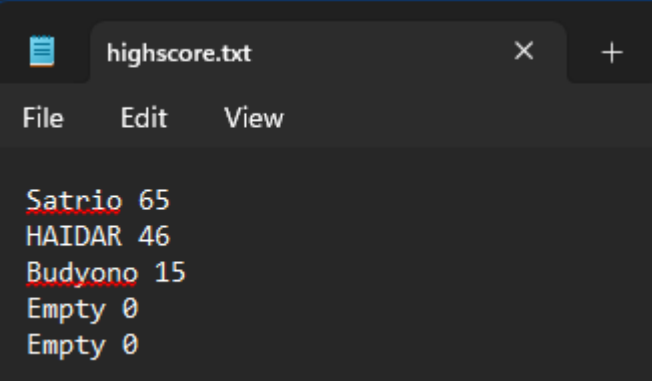
```
Jin : Nampaknya kamu telah menyelesaikan seluruh tantanganku.  
Selamat! Pintu keluar Gua ada di belakangku.|
```



```
=====
AND A KELUAR DARI GUA DENGAN SELAMAT
=====
Skor Akhir Anda: 65
Selamat! Anda mendapatkan High Score baru!
Masukkan username Anda (satu kata): Satrio|
```

### 3.1.14. High Score Baru

```
===== TOP 5 HIGH SCORE =====
Rank Username Score
-----
1. Satrio 65
2. HAIDAR 46
3. Budyono 15
4. Empty 0
5. Empty 0
-----
Press enter to return to menu...|
```



```
highscore.txt
File Edit View
Satrio 65
HAIDAR 46
Budyono 15
Empty 0
Empty 0
```

### 3.1.15. Jika User Kehabisan Nyawa Di Level 1

```
Soal No. 6 :
Hasil dari -4 x -6 adalah : 3
Maaf jawaban Anda salah. Jawaban yang benar adalah 24

Soal No. 7 :
Hasil dari -8 x -8 adalah : 4
Maaf jawaban Anda salah. Jawaban yang benar adalah 64

Soal No. 8 :
Hasil dari -12 : -2 adalah : 6
Selamat jawaban Anda benar!

Soal No. 9 :
Hasil dari 0 : -7 adalah : 1
Maaf jawaban Anda salah. Jawaban yang benar adalah 0
|
```

```
Kamu pingsan dan terbangun kembali di awal gua.
Skor Akhir Anda: 0
Maaf, Anda tidak masuk dalam Top 5 High Score.

Press enter to return to menu...
```

### 3.1.16. User Kehabisan Nyawa Di Level 2

```
Soal no. 2
tentukan nilai x dari  $7X + 91 = 0$ 
Jawaban Anda: 3

Maaf jawaban Anda salah!

Soal no. 3
tentukan nilai x dari  $5x + 27 = 9X - 21$ 
Jawaban Anda: 2

Maaf jawaban Anda salah!

Soal no. 4
tentukan volume kerucut dengan jari-jari = 3 meter,tinggi = 7 meter
Jawaban Anda: 1

Maaf jawaban Anda salah!
```

```
Kamu pingsan dan terbangun kembali di awal gua.
Skor Akhir Anda: 0
Maaf, Anda tidak masuk dalam Top 5 High Score.

Press enter to return to menu...
```

### 3.1.17. User Kehabisan Nyawa Di Level 3

```
Soal no. 1
Jarak rumah Candra ke kampus adalah 2 km.
Setiap akan ke kampus, Candra mengendarai sepeda motor dengan kecepatan rata-rata 40 km/jam.
Waktu yang dibutuhkan Candra untuk sampai ke kampus adalah ... menit.

Jawaban Anda: 23

Maaf jawaban Anda salah. Jawaban yang benar adalah 3

Soal no. 2
Ani pergi ke suatu pusat perbelanjaan untuk mencari baju baru.
Ia tertarik pada sebuah kemeja dengan harga Rp. 150.000.
Pada toko tersebut, terdapat diskon untuk pembelian sebuah baju, sebesar 30%.
Uang yang harus dikeluarkan ani untuk membeli kemeja tersebut adalah... (jawaban tanpa titik)

Jawaban Anda: 22

Maaf jawaban Anda salah. Jawaban yang benar adalah 105000

Soal no. 3
Pipa air di suatu tempat mempunyai 354 cabang saluran untuk keperluan sehari-hari.
Setiap rumah terpasang 1 saluran air.
Dalam satu hari, pemakaian air mencapai 61.950 liter.
Rata-rata air yang digunakan di setiap rumah per hari adalah ....

Jawaban Anda: 66

Maaf jawaban Anda salah. Jawaban yang benar adalah 175
```

```
Kamu pingsan dan terbangun kembali di awal gua.  
Skor Akhir Anda: 21  
Selamat! Anda mendapatkan High Score baru!  
Masukkan username Anda (satu kata):
```

(Kondisi kalah tapi Skor masih bisa masuk ke Top 5)

#### 3.1.18. High Score Akhir

```
===== TOP 5 HIGH SCORE =====  
Rank  Username      Score  
-----  
1.    Satrio         65  
2.    HAIDAR         46  
3.    Heru           21  
4.    Budyono        15  
5.    Empty          0  
-----  
Press enter to return to menu...|
```

#### 3.1.19. Pesan Unik Ketika Mengulang Game Setelah Kalah

```
Jin: 'Kita bertemu lagi nak, apakah kamu siap untuk mengulang tantanganku?'  
Jawabanmu? (y/n):
```

#### 3.1.20. Apabila Menjawab n Ketika ditanya Jin

```
Jin: 'Kita bertemu lagi nak, apakah kamu siap untuk mengulang tantanganku?'  
Jawabanmu? (y/n): n  
Jin: 'Baiklah, sampai jumpa lain waktu!'
```

#### 3.1.21. Exit Program

```
Terima kasih telah bermain! Sampai jumpa lagi!  
Press enter to continue...
```

## **BAB IV**

### **PEMBAGIAN TUGAS**

#### **4.1. Pembagian Tugas dalam Kelompok**

##### **1. Farrell Luthfi Adriansyah**

###### **a. Program**

- Merancang soal bertipe soal cerita pada stage 3
- Merancang Logika Permainan dari Stage 3
- Merancang logika penulisan soal cerita dengan file processing
- Merancang logika file processing

###### **b. Makalah**

- Membuat bagian alur kerja dalam bentuk flowchart pada BAB I
- Menjelaskan bagian library yang digunakan pada program
- Menjelaskan bagian source code stage 3

###### **c. PPT**

- Membuat bagian alur kerja dalam bentuk flowchart pada BAB I
- Menjelaskan bagian library yang digunakan pada program
- Menjelaskan bagian source code stage 3

##### **2. Muhammad Haidar Amru**

###### **a. Program**

- Menyiapkan konstanta untuk menyimpan file highscore dengan `#define FILE_HIGHSCORE`.
- Menyiapkan variabel global nyawa, skor, dan death.
- Menyiapkan Struktur data high score beserta manajemen file untuk simpan, muat, dan update high score.
- Menyiapkan utility function untuk mengatur interface.
- Menyiapkan fungsi soal-soal untuk level 1.
- Menyiapkan main menu beserta storyline dari game.

###### **b. Makalah**

- Menjelaskan bagian source code dari konstanta, variabel global, struktur data, utility function, dan level 1.
- Menjelaskan Alur kerja program di BAB III
- c. PPT
  - Menjelaskan bagian source code dari konstanta, variabel global, struktur data, utility function, dan level 1.
  - Menjelaskan Alur kerja program di BAB III
- 3. Satrio Unggul Prayogo
  - a. Program
    - Merancang soal stage 2
    - Merancang logika permainan stage 2
    - Merancang logika file processing
    - Merancang logika penulisan soal isian dengan file processing
  - b. Makalah
    - Menjelaskan bagian penjelasan umum
    - Menjelaskan bagian source code stage 2
    - Menjelaskan bagian kesimpulan
    - Menjelaskan bagian saran
    - Menjelaskan bagian refleksi
  - c. PPT
    - Menjelaskan bagian penjelasan umum
    - Menjelaskan bagian source code stage 2
    - Menjelaskan bagian kesimpulan
    - Menjelaskan bagian saran
    - Menjelaskan bagian refleksi

## **BAB V**

### **KESIMPULAN**

#### **5.1. Kesimpulan**

Dengan alur cerita dan konsep permainan, pembelajaran matematika dapat tersampaikan dengan sangat menarik, menyenangkan, dan tidak membuat pemain merasa bosan.

Materi yang dijadikan dasar pembuatan soal pada permainan ini mengasah kemampuan dalam menyelesaikan permasalahan yang menggunakan matematika dalam kehidupan sehari-hari. Alur cerita akan membuat pemain merasa berada di dalam cerita dan harus menyelesaikan tantangan berupa soal yang diberikan. Sistem nyawa dan pencatatan skor pada permainan ini akan membuat pemain merasa lebih tertantang untuk menyelesaikan soal yang diberikan. Kesimpulannya, permainan ini merupakan cara yang efektif sekaligus menyenangkan sebagai media pembelajaran matematika dasar dan logika.

## **5.2. Saran**

Berdasarkan program yang telah kami buat, terdapat beberapa hal yang dapat dijadikan saran dalam pengembangan lebih lanjut :

- a. Pengelompokan peserta berdasarkan jenjang pendidikan.
- b. Pengembangan jenis soal yang lebih terstruktur dan sesuai dengan kurikulum sesuai dengan jenjang.
- c. Desain grafis permainan yang dapat dikembangkan menjadi berwarna dengan desain yang menarik sesuai cerita.

## **5.3. Refleksi**

Dalam pengembangan program ini, diperlukan adanya kerja sama dan pemecahan masalah yang dilakukan bersama-sama. Akan tetapi metode sharing project yang kami gunakan masih kurang efektif karena terdapat beberapa kendala seperti koneksi yang terputus dan akses untuk percobaan program yang terbatas. Untuk mengatasi masalah tersebut, dapat menggunakan sarana pendukung dalam manajemen proyek seperti Git dan Github. Selain itu, dengan pengembangan berupa tampilan dan grafis yang lebih menarik, serta sistem soal yang lebih terstruktur, maka program ini dapat menjadi alat bantu pembelajaran yang efektif dan menarik bagi pelajar.