

# **LAPORAN PROYEK**

## **Pemograman Sistem**

### **TEBAK KATA**

**Disusun untuk meyelesaikan Tugas Proyek Pemograman Sistem**

**Mahasiswa Institut Teknologi DEL**



**OLEH:**

**PAIAN MANALU**

**13322008**

**D3 - Teknologi Komputer**

**INSTITUT TEKNOLOGI DEL**

**FAKULTAS VOKASI**

## **Deskripsi**

Deskripsi Game Tebak Kata:

Game Tebak Kata adalah permainan interaktif yang menguji kecerdasan, kreativitas, dan kemampuan bahasa pemain. Dalam game ini, terdapat dua pemain, di mana satu pemain bertindak sebagai pembuat kata yang membuat kata dan memberikan kisi-kisi, sedangkan pemain lainnya berperan sebagai penebak kata yang mencoba menebak kata yang telah dibuat.

## **Tujuan Game:**

Tujuan dari Game Tebak Kata adalah:

1. Peningkatan Kreativitas: Mendorong pemain untuk menggunakan imajinasi dan kreativitas dalam membuat kata dan kisi-kisi yang menarik.
2. Ketajaman Bahasa: Mengasah kemampuan bahasa dan kosakata pemain, baik sebagai pembuat kata maupun penebak kata.
3. Kerjasama: Membangun kerjasama antara dua pemain dalam sebuah permainan yang mengandalkan komunikasi dan pemahaman.

## **Cara Bermain:**

### **1. Persiapan:**

- Setiap pemain diminta untuk memasukkan namanya sebagai identifikasi dalam permainan.
- Pemain 1 (Pembuat Kata) membuat sebuah kata dan memberikan tiga kisi-kisi terkait kata tersebut.
- Data kata dan kisi-kisi disimpan dalam shared memory dan dikirim ke Pemain 2.

### **2. Ronde Permainan:**

- Pemain 2 (Penebak Kata) menerima kata dan kisi-kisi dari Pemain 1 melalui shared memory.
- Kisi-kisi ditampilkan kepada Pemain 2.
- Pemain 2 memiliki 5 kesempatan untuk menebak kata yang dibuat oleh Pemain 1.
- Skor ronde ditentukan berdasarkan keberhasilan menebak kata.

### **3. Skor dan Kelanjutan:**

- Setelah menebak atau habis kesempatan, skor ronde ditampilkan.
- Setelah 10 ronde, skor total Pemain 2 ditampilkan.
- Pemain 1 dan Pemain 2 dapat memutuskan untuk melanjutkan ke ronde berikutnya atau mengakhiri permainan.

**Catatan kerja:**

- Game ini menggunakan shared memory dan file "pemain2.txt dan pemain1.txt" sebagai media komunikasi antara dua pemain.
- Pemain 1 memasukkan sebuah kata dan tiga kisi-kisi terkait.
- Informasi tersebut disimpan dalam shared memory dan ditulis ke file "pemain2.txt" untuk dikirim ke pemain 2.
- Pemain 2 harus menggunakan informasi kisi-kisi untuk menebak kata yang dibuat oleh Pemain 1.
- Skor diberikan berdasarkan keberhasilan menebak kata.
- Game menekankan kerjasama dan kreativitas antara dua pemain sambil menantang pemain 2 untuk menebak kata yang dibuat oleh pemain 1.

**Manfaat:**

- Meningkatkan keterampilan berbahasa.
- Mendorong kerjasama dan komunikasi antar pemain.
- Merangsang kreativitas dan imajinasi dalam menciptakan kata dan kisi-kisi.

## PLAYER 1

```
paian@centos7-lab:~  
GNU nano 2.3.1 File: pemain1.c  
  
#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
#include <sys/ipc.h>  
#include <sys/shm.h>  
#include <time.h>  
  
#define MAX_NAME_LENGTH 50  
#define MAX_KISI_LENGTH 1000  
  
// Struct untuk data bersama antara pemain 1 dan pemain 2  
struct SharedData {  
    char kata[MAX_NAME_LENGTH];  
    char kisi1[MAX_KISI_LENGTH];  
    char kisi2[MAX_KISI_LENGTH];  
    char kisi3[MAX_KISI_LENGTH];  
};  
  
// ANSI escape codes untuk format teks  
#define ANSI_COLOR_RESET "\033[0m"  
#define ANSI_BOLD "\033[1m"  
#define ANSI_SIZE "\033[8m"  
#define ANSI_COLOR_BLUE "\033[34m"  
#define ANSI_COLOR_YELLOW "\033[33m"  
#define ANSI_COLOR_RED "\033[31m"  
  
int main() {  
    char namaPemain[MAX_NAME_LENGTH];  
  
    printf(ANSI_BOLD ANSI_SIZE ANSI_COLOR_BLUE "Selamat datang di game tebak kata!\n\n");  
    printf("Mari kita mulai.\n" ANSI_COLOR_RESET);  
  
    // Meminta nama pemain 1  
    printf("Masukkan nama Pemain 1: ");  
    scanf("%s", namaPemain);  
  
    // Membuat kunci dari path file menggunakan ftok  
    key_t key = ftok("pemain1.c", 'R');  
    if (key == -1) {  
        perror(ANSI_COLOR_YELLOW "ftok" ANSI_COLOR_RESET);  
        exit(EXIT_FAILURE);  
    }  
  
    // Mendapatkan atau membuat shared memory  
    int shmid = shmget(key, sizeof(struct SharedData), 0666 | IPC_CREAT);  
    if (shmid == -1) {  
  
        // Mendapatkan atau membuat shared memory  
        int shmid = shmget(key, sizeof(struct SharedData), 0666 | IPC_CREAT);  
        if (shmid == -1) {  
            perror(ANSI_COLOR_YELLOW "shmget" ANSI_COLOR_RESET);  
            exit(EXIT_FAILURE);  
        }  
    }  
}
```

```

    }

    // Menyambungkan program ke shared memory
    struct SharedData *sharedData = (struct SharedData *)shmat(shmid, (void *)0, 0);
    if (sharedData == (void *)-1) {
        perror(ANSI_COLOR_YELLOW "shmat" ANSI_COLOR_RESET);
        exit(EXIT_FAILURE);
    }

    int lanjut;
    int totalSkor = 0;

    srand(time(NULL));

    // Loop untuk setiap ronde (maksimum 10 ronde)
    for (int ronde = 1; ronde <= 10; ronde++) {
        printf("\n%sRonde %d - %s:\n", ANSI_COLOR_YELLOW, ronde, namaPemain);

        printf("%s, sekarang giliran Anda membuat kata!\n", namaPemain);
        printf("Masukkan kata: ");
        scanf("%s", sharedData->kata);

        // Memasukkan kisi-kisi atau petunjuk (lebih dari tiga kata)
        printf("Masukkan kisi-kisi atau petunjuk (lebih dari tiga kata):\n");
        printf("Yang ke-1: ");
        scanf(" %[^\\n]s", sharedData->kisi1);
        printf("Yang ke-2: ");
        scanf(" %[^\\n]s", sharedData->kisi2);
        printf("Yang ke-3: ");
        scanf(" %[^\\n]s", sharedData->kisi3);

        // Membuka file "pemain2.txt" untuk menulis data kata dan kisi-kisi
        FILE *file = fopen("pemain2.txt", "w");
        if (file == NULL) {
            perror("Error membuka file");
            return 1;
        }

        // Menulis data ke file
        fprintf(file, "%s\\n", sharedData->kata);
        fprintf(file, "%s\\n", sharedData->kisi1);
        fprintf(file, "%s\\n", sharedData->kisi2);
        fprintf(file, "%s\\n", sharedData->kisi3);

        fclose(file);

        printf("Kata dan kisi-kisi berhasil dikirim ke Pemain 2.\\n");

        if (ronde < 10) {
            // Menanyakan apakah pemain ingin melanjutkan ke ronde berikutnya
            printf("Apakah Anda ingin melanjutkan ke ronde berikutnya? (1: Ya, 0: Tidak): ");
            scanf("%d", &lanjut);
            if (!lanjut) {
                break;
            }
        }
    }

    // Melepaskan shared memory
    shmdt(sharedData);
    shmctl(shmid, IPC_RMID, NULL);

    return 0;
}

```

## PLAYER 2

```
paian@centos7-lab:~  
GNU nano 2.3.1 File: pemain2.c  
#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
#include <sys/ipc.h>  
#include <sys/shm.h>  
#include <time.h>  
  
// Batasan panjang nama dan kisi-kisi  
#define MAX_NAME_LENGTH 50  
#define MAX_KISI_LENGTH 1000  
  
// Struktur SharedData untuk menyimpan kata dan kisi-kisi  
struct SharedData {  
    char kata[MAX_NAME_LENGTH];  
    char kisi1[MAX_KISI_LENGTH];  
    char kisi2[MAX_KISI_LENGTH];  
    char kisi3[MAX_KISI_LENGTH];  
};  
  
// ANSI escape codes untuk format teks  
#define ANSI_COLOR_RESET "\033[0m"  
#define ANSI_BOLD "\033[1m"  
#define ANSI_COLOR_BLUE "\033[34m"  
#define ANSI_COLOR_YELLOW "\033[33m"  
#define ANSI_COLOR_RED "\033[31m"  
  
int main() {  
    char namaPemain[MAX_NAME_LENGTH];  
    int lanjut;  
    int totalSkor = 0;  
  
    // Pesan selamat datang  
    printf(ANSI_BOLD ANSI_COLOR_BLUE "Selamat datang di game tebak kata!\n");  
    printf("Mari kita mulai.\n" ANSI_COLOR_RESET);  
  
    // Meminta nama Pemain 2  
    printf("Masukkan nama Pemain 2: ");  
    scanf("%s", namaPemain);  
  
    // Membuat kunci untuk shared memory  
    key_t key = ftok("pemain1.c", 'R');  
    if (key == -1) {  
        perror(ANSI_COLOR_YELLOW "ftok" ANSI_COLOR_RESET);  
        exit(EXIT_FAILURE);  
    }  
  
    // Mendapatkan atau membuat shared memory  
    int shmid = shmget(key, sizeof(struct SharedData), 0666 | IPC_CREAT);  
    if (shmid == -1) {  
        perror(ANSI_COLOR_YELLOW "shmget" ANSI_COLOR_RESET);  
        exit(EXIT_FAILURE);  
    }  
  
    // Menyambungkan program ke shared memory  
    struct SharedData *sharedData = (struct SharedData *)shmat(shmid, (void *)0, 0);  
    if (sharedData == (void *)-1) {  
        perror(ANSI_COLOR_YELLOW "shmat" ANSI_COLOR_RESET);  
        exit(EXIT_FAILURE);  
    }  
}
```

```

// Inisialisasi seed untuk generator bilangan acak
srand(time(NULL));

// Loop permainan untuk 10 ronde
for (int ronde = 1; ronde <= 10; ronde++) {
    // Menampilkan informasi ronde
    printf("\n%sRonde %d - %s:\n", ANSI_COLOR_YELLOW, ronde, namaPemain);

    // Menampilkan kisi-kisi yang diterima dari Pemain 1
    printf("Kisi 1: %s\n", sharedData->kisi1);
    printf("Kisi 2: %s\n", sharedData->kisi2);
    printf("Kisi 3: %s\n", sharedData->kisi3);

    int chance;
    int skorRonde = 0; // Skor untuk ronde ini

    // Loop untuk menebak kata dengan maksimal 5 kesempatan
    for (chance = 1; chance <= 5; chance++) {
        printf("Tebak kata (Kesempatan ke-%d): ", chance);
        char jawaban[MAX_NAME_LENGTH];
        scanf("%s", jawaban);

        // Memeriksa apakah jawaban benar
        if (strcmp(sharedData->kata, jawaban) == 0) {
            printf(ANSI_COLOR_YELLOW "Benar!\n" ANSI_COLOR_RESET);
            skorRonde += 10; // Poin untuk jawaban benar
            break;
        } else {
            printf(ANSI_COLOR_RED "Salah! Coba lagi.\n" ANSI_COLOR_RESET);
            if (chance == 5) {
                printf("Kesempatan habis. Jawaban yang benar adalah: %s\n", sharedData->kata);
            }
        }
    }
}

```

```

// Menambahkan skor ronde ke total skor
totalSkor += skorRonde;

// Menampilkan skor untuk ronde dan skor total
printf("Skor untuk Ronde %d: %d\n", ronde, skorRonde);
printf("Skor Total: %d\n", totalSkor);

// Menanyakan apakah pemain ingin melanjutkan ke ronde berikutnya
if (ronde < 10) {
    printf("Apakah Anda ingin melanjutkan ke ronde berikutnya? (1: Ya, 0: Tidak): ");
    scanf("%d", &lanjut);
    if (!lanjut) {
        break; // Keluar dari loop jika tidak ingin melanjutkan
    }
}

// Melepaskan program dari shared memory
shmdt(sharedData);

// Mengembalikan nilai 0 sebagai tanda program berakhir dengan sukses
return 0;
}

```

## OUTPUT:

- Terminal 1 : Player 1

### Membuat tebak kata

```
paian@centos7-lab:~  
[paian@centos7-lab ~]$ gcc -std=c99 -D_XOPEN_SOURCE=500 -o pemain1 pemain1.c  
[paian@centos7-lab ~]$ ./pemain1  
Selamat datang di game tebak kata!  
  
Mari kita mulai.  
Masukkan nama Pemain 1: Paian  
  
Ronde 1 - Paian:  
Paian, sekarang giliran Anda membuat kata!  
Masukkan kata: Harimau  
Masukkan kisi-kisi atau petunjuk (lebih dari tiga kata):  
Yang ke-1: Hewan buas  
Yang ke-2: kaki empat  
Yang ke-3: hutan  
Kata dan kisi-kisi berhasil dikirim ke Pemain 2.  
Apakah Anda ingin melanjutkan ke ronde berikutnya? (1: Ya, 0: Tidak): █
```

- Terminal 2 : Player 2

### Menebak kata

```
[paian@centos7-lab ~]$ gcc -std=c99 -D_XOPEN_SOURCE=500 -o pemain2 pemain2.c  
[paian@centos7-lab ~]$ ./pemain2  
Selamat datang di game tebak kata!  
Mari kita mulai.  
Masukkan nama Pemain 2: Daniel  
  
Ronde 1 - Daniel:  
Kisi 1: Hewan buas  
Kisi 2: kaki empat  
Kisi 3: hutan  
Tebak kata (Kesempatan ke-1): Harimau  
Benar!  
Skor untuk Ronde 1: 10  
Skor Total: 10  
Apakah Anda ingin melanjutkan ke ronde berikutnya? (1: Ya, 0: Tidak): █
```

```
paian@centos7-lab:~  
[paian@centos7-lab ~]$ gcc -std=c99 -D_XOPEN_SOURCE=500 -o pemain2 pemain2.c  
[paian@centos7-lab ~]$ ./pemain2  
Selamat datang di game tebak kata!  
Mari kita mulai.  
Masukkan nama Pemain 2: Daniel  
  
Ronde 1 - Daniel:  
Kisi 1: Hewan buas  
Kisi 2: kaki empat  
Kisi 3: hutan  
Tebak kata (Kesempatan ke-1): Singa  
Salah! Coba lagi.  
Tebak kata (Kesempatan ke-2): Buaya  
Salah! Coba lagi.  
Tebak kata (Kesempatan ke-3): Gorilla  
Salah! Coba lagi.  
Tebak kata (Kesempatan ke-4): Hrimau  
Salah! Coba lagi.  
Tebak kata (Kesempatan ke-5): Harimau  
Benar!  
Skor untuk Ronde 1: 10  
Skor Total: 10  
Apakah Anda ingin melanjutkan ke ronde berikutnya? (1: Ya, 0: Tidak): █
```



## Penjelasan Kodingan Pemain1.c

### Bagian 1: Pesan Selamat Datang dan Input Nama Pemain 1

```
char namaPemain[MAX_NAME_LENGTH];

printf(ANSI_BOLD ANSI_SIZE ANSI_COLOR_BLUE "Selamat datang di game tebak kata!\n");

printf("Mari kita mulai.\n" ANSI_COLOR_RESET);

// Request Player 1's name

printf("Masukkan nama Pemain 1: ");

scanf("%s", namaPemain);
```

#### Penjelasan:

Program memulai dengan menyapa pemain dan meminta nama dari Pemain 1.

Escape codes ANSI digunakan untuk memberikan efek teks pada pesan selamat datang.

### Bagian 2: Pembuatan Kunci untuk Shared Memory

```
key_t key = ftok("pemain1.c", 'R');

if (key == -1) {

    perror(ANSI_COLOR_YELLOW "ftok" ANSI_COLOR_RESET);

    exit(EXIT_FAILURE);

}
```

#### Penjelasan:

Fungsi ftok digunakan untuk menghasilkan kunci berdasarkan path file ("pemain1.c") dan karakter 'R'.

Kunci digunakan untuk mengidentifikasi shared memory.

### Bagian 3: Membuat atau Mendapatkan Shared Memory

```
int shmid = shmget(key, sizeof(struct SharedData), 0666 | IPC_CREAT);

if (shmid == -1) {

    perror(ANSI_COLOR_YELLOW "shmget" ANSI_COLOR_RESET);

    exit(EXIT_FAILURE);

}
```

#### Penjelasan:

Fungsi shmget digunakan untuk membuat atau mendapatkan shared memory.

Shared memory akan memiliki ukuran yang sesuai dengan struktur data struct SharedData.

## Bagian 4: Menyambungkan Program ke Shared Memory

```
struct SharedData *sharedData = (struct SharedData *)shmat(shmid, (void *)0, 0);

if (sharedData == (void *)-1) {

    perror(ANSI_COLOR_YELLOW "shmat" ANSI_COLOR_RESET);

    exit(EXIT_FAILURE);

}
```

Penjelasan:

Fungsi shmat digunakan untuk menyambungkan program ke shared memory

## Bagian 5: Inisialisasi Variabel dan Loop Permainan

```
int lanjut;

srand(time(NULL));

for (int ronde = 1; ronde <= 10; ronde++) {
```

Penjelasan:

Inisialisasi variabel lanjut yang akan digunakan untuk menentukan apakah pemain ingin melanjutkan ke ronde berikutnya.

Penggunaan fungsi srand untuk menginisialisasi generator bilangan acak berdasarkan waktu saat ini.

## Bagian 6: Input Kata dan Kisi-kisi dari Pemain 1

```
printf("\n%sRonde %d - %s:\n", ANSI_COLOR_YELLOW, ronde, namaPemain);

printf("%s, sekarang giliran Anda membuat kata!\n", namaPemain);

printf("Masukkan kata: ");

scanf("%s", sharedData->kata);

printf("Masukkan kisi-kisi atau petunjuk (lebih dari tiga kata):\n");

printf("Yang ke-1: ");

scanf("%[^\n]s", sharedData->kisi1);

printf("Yang ke-2: ");

scanf("%[^\n]s", sharedData->kisi2);

printf("Yang ke-3: ");

scanf("%[^\n]s", sharedData->kisi3);
```

Penjelasan:

Pemain 1 diminta untuk memasukkan kata yang akan ditebak dan tiga kisi-kisi atau petunjuk.

scanf digunakan untuk membaca kata dan kisi-kisi dari pemain, termasuk kisi-kisi yang dapat memiliki spasi.

## Bagian 7: Menulis Data ke File "pemain2.txt"

```
FILE *file = fopen("pemain2.txt", "w");

if (file == NULL) {

    perror("Error membuka file");

    return 1;

}

fprintf(file, "%s\n", sharedData->kata);
fprintf(file, "%s\n", sharedData->kisi1);
fprintf(file, "%s\n", sharedData->kisi2);
fprintf(file, "%s\n", sharedData->kisi3);

fclose(file);

printf("Kata dan kisi-kisi berhasil dikirim ke Pemain 2.\n");
```

Penjelasan:

Program membuka file "pemain2.txt" untuk menulis data kata dan kisi-kisi yang dimasukkan oleh Pemain 1.

Menggunakan fprintf untuk menulis data ke file.

File ditutup setelah selesai.

## Bagian 8: Pertanyaan Lanjut atau Berhenti Setelah Setiap Runde

```
if (ronde < 10) {

    printf("Apakah Anda ingin melanjutkan ke ronde berikutnya? (1: Ya, 0: Tidak): ");

    scanf("%d", &lanjut);

    if (!lanjut) {

        break;

    }

}

}
```

Penjelasan:

Setelah setiap ronde, pemain ditanya apakah mereka ingin melanjutkan ke ronde berikutnya atau tidak.

Jika pemain tidak ingin melanjutkan, program keluar dari loop menggunakan break.

## Penjelasan Kodingan Pemain2.c

### Bagian 1: Inisialisasi Variabel dan Pesan Selamat Datang

```
char namaPemain[MAX_NAME_LENGTH];

int lanjut;

int totalSkor = 0;

printf(ANSI_BOLD ANSI_COLOR_BLUE "Selamat datang di game tebak kata!\n");

printf("Mari kita mulai.\n" ANSI_COLOR_RESET);

// Request Player 2's name

printf("Masukkan nama Pemain 2: ");

scanf("%s", namaPemain);
```

Penjelasan:

Variabel namaPemain digunakan untuk menyimpan nama Pemain 2.

Variabel lanjut digunakan untuk menentukan apakah pemain ingin melanjutkan ke ronde berikutnya.

Variabel totalSkor digunakan untuk menghitung skor total pemain.

Pesan selamat datang dan permintaan nama Pemain 2 ditampilkan.

### Bagian 2: Pembuatan Kunci untuk Shared Memory

```
key_t key = ftok("pemain1.c", 'R');

if (key == -1) {

    perror(ANSI_COLOR_YELLOW "ftok" ANSI_COLOR_RESET);

    exit(EXIT_FAILURE);

}

int shmid = shmget(key, sizeof(struct SharedData), 0666 | IPC_CREAT);

if (shmid == -1) {

    perror(ANSI_COLOR_YELLOW "shmget" ANSI_COLOR_RESET);

    exit(EXIT_FAILURE);

}

struct SharedData *sharedData = (struct SharedData *)shmat(shmid, (void *)0, 0);

if (sharedData == (void *)-1) {

    perror(ANSI_COLOR_YELLOW "shmat" ANSI_COLOR_RESET);

    exit(EXIT_FAILURE);

}
```

Penjelasan:

Langkah-langkah yang sama seperti pada program Pemain 1 untuk membuat atau mendapatkan shared memory.

Kunci yang dihasilkan oleh ftok harus sama dengan kunci yang digunakan oleh Pemain 1.

### Bagian 3: Loop Permainan Pemain 2

```
srand(time(NULL));  
  
for (int ronde = 1; ronde <= 10; ronde++) {
```

Penjelasan:

Pemain 2 mulai memasuki loop permainan sebanyak 10 ronde.

Fungsi srand digunakan untuk menginisialisasi generator bilangan acak berdasarkan waktu saat ini.

### Bagian 4: Menampilkan Kisi-Kisi yang Diterima

```
printf("\n%sRonde %d - %s:\n", ANSI_COLOR_YELLOW, ronde, namaPemain);  
  
// Output Kisi-Kisi yang diterima  
  
printf("Kisi 1: %s\n", sharedData->kisi1);  
  
printf("Kisi 2: %s\n", sharedData->kisi2);  
  
printf("Kisi 3: %s\n", sharedData->kisi3);
```

Penjelasan:

Pemain 2 melihat kisi-kisi yang telah diterima dari Pemain 1 untuk ronde saat ini.

### Bagian 5: Loop Tebak Kata Pemain 2

```
int chance;  
  
int skorRonde = 0; // Skor untuk ronde ini  
  
for (chance = 1; chance <= 5; chance++) {
```

Penjelasan:

Loop untuk memberi Pemain 2 lima kesempatan untuk menebak kata yang telah dibuat oleh Pemain 1.

Variabel skorRonde digunakan untuk menghitung skor dari ronde ini.

### Bagian 6: Input Jawaban dan Evaluasi

```
printf("Tebak kata (Kesempatan ke-%d): ", chance);  
  
char jawaban[MAX_NAME_LENGTH];  
  
scanf("%s", jawaban);
```

```

if (strcmp(sharedData->kata, jawaban) == 0) {

    printf(ANSI_COLOR_YELLOW "Benar!\n" ANSI_COLOR_RESET);

    skorRonde += 10; // Poin untuk jawaban benar

    break;

} else {

    printf(ANSI_COLOR_RED "Salah! Coba lagi.\n" ANSI_COLOR_RESET);

    if (chance == 5) {

        printf("Kesempatan habis. Jawaban yang benar adalah: %s\n", sharedData->kata);

    }

}

}

```

Penjelasan:

Pemain 2 diminta untuk menebak kata dengan maksimal lima kesempatan.

Jawaban dibandingkan dengan kata yang telah dibuat oleh Pemain 1.

Jika jawaban benar, skor ronde ditambahkan dan loop dihentikan.

Jika jawaban salah dan kesempatan habis, jawaban yang benar ditampilkan.

## Bagian 7: Menghitung dan Menampilkan Skor

```

totalSkor += skorRonde; // Tambahkan skor ronde ke total skor

printf("Skor untuk Runde %d: %d\n", ronde, skorRonde);

printf("Skor Total: %d\n", totalSkor);

```

Penjelasan:

Skor ronde ditambahkan ke total skor.

Skor untuk ronde dan skor total ditampilkan.

## Bagian 8: Pertanyaan Lanjut atau Berhenti Setelah Setiap Runde

```

if (ronde < 10) {

    printf("Apakah Anda ingin melanjutkan ke ronde berikutnya? (1: Ya, 0: Tidak): ");

    scanf("%d", &lanjut);

    if (!lanjut) {

        break;

    }

}

```

```
}  
}
```

Penjelasan:

Setelah setiap ronde, pemain ditanya apakah mereka ingin melanjutkan ke ronde berikutnya atau tidak.

Jika pemain tidak ingin melanjutkan, program keluar dari loop menggunakan break.