

KATA PENGANTAR

Puji syukur kehadiran Tuhan Yang Maha Esa atas segala rahmat-NYA sehingga Laporan ini dapat tersusun hingga selesai . Tidak lupa kami juga mengucapkan banyak terimakasih atas bantuan dari pihak yang telah berkontribusi dengan memberikan sumbangan baik materi maupun pikirannya.

Kami mencoba untuk membuat dan menyelesaikan laporan ini dengan sebaik-baiknya. laporan ini dimaksudkan untuk memberikan wawasan dan pengetahuan tentang Strukturda laporan ini tidak lepas dari kekurangan karena manusia bukanlah makhluk yang sempurna. Tetapi kami berusaha untuk membagi ilmu dan wawasan yang telah kami rangkum dalam makalah ini. semoga makalah ini bisa bermanfaat dan menjadi rujukan dalam memperoleh ilmu pengetahuan.

DAFTAR ISI

KATA PENGANTAR.....	1
BAB I PENDAHULUAN.....	3
1.1. LATAR BELAKANG PEMBUATAN GAME SNAKE.....	3
1.2 . TUJUAN PEMBUATAN GAME.....	3
1.3. KONSEP DASAR GAME SNAKE.....	3
BAB II FLOWCHART.....	4
BAB III PEMBAHASAN.....	6
3.1. RANCANGAN PROGRAM	6
3.1.1. SYNTAX STRUCT.....	6
3.1.2. SYNTAX OPERASI FILE SKOR_READ.....	6
3.1.3. SYNTAX OPERASI FILE SKOR_WRITE.....	7
3.1.4. SYNTAX FUNGSI PUSH.....	7
3.1.5. SYNTAX FUNGSI POP	8
3.1.6. SYNTAX FUNGSI POP_ALL.....	8
3.1.7. SYNTAX FUNGSI GOTOXY.....	8
3.1.8. SYNTAX FUNGSI DISPLAY.....	9
3.1.9. SYNTAX INISIALISASI VARIABEL SET DARI SNAKE.....	9
3.1.10. SYNTAX FUNGSI CEK	10
3.1.11. SYNTAX FUNGSI PLACE_FOOD.....	10
3.1.12. SYNTAX FUNGSI LOAD	10
3.2. TAMPILAN OUTPUT GAME SNAKE	11
3.2.1. Tampilan Saat Program dimulai.....	11
3.2.2. Tampilan Saat permainan dimulai.....	12
BAB IV PENUTUP.....	13
4.1. KESIMPULAN.....	13

BAB I PENDAHULUAN

1.1. LATAR BELAKANG PEMBUATAN GAME SNAKE

Latar belakang dari pembuatan aplikasi game Snake . Zaman teknologi ini para penggemar game mungkin bertanya – tanya bagaimana cara membuat game yang sedang mereka mainkan, walaupun sebagian besar juga mungkin sudah cukup senang dengan hanya memainkan game yang mereka sukai, tanpa perlu berpikir bagaimana game tersebut dibuat. Namun demikian, jika kita mempunyai kemauan untuk ingin lebih memperluas wawasan, maka kita akan berusaha mengetahui cara membuat game tersebut dan apa yang menyebabkan game itu menjadi menarik untuk dimainkan.

Untuk itu Penulis ingin membuat suatu game sederhana berupa game snake yang dapat berjalan dalam sebuah komputer. Permainan ini dapat dimainkan hanya dengan 1 player dengan menggunakan bahasa pemrograman C.

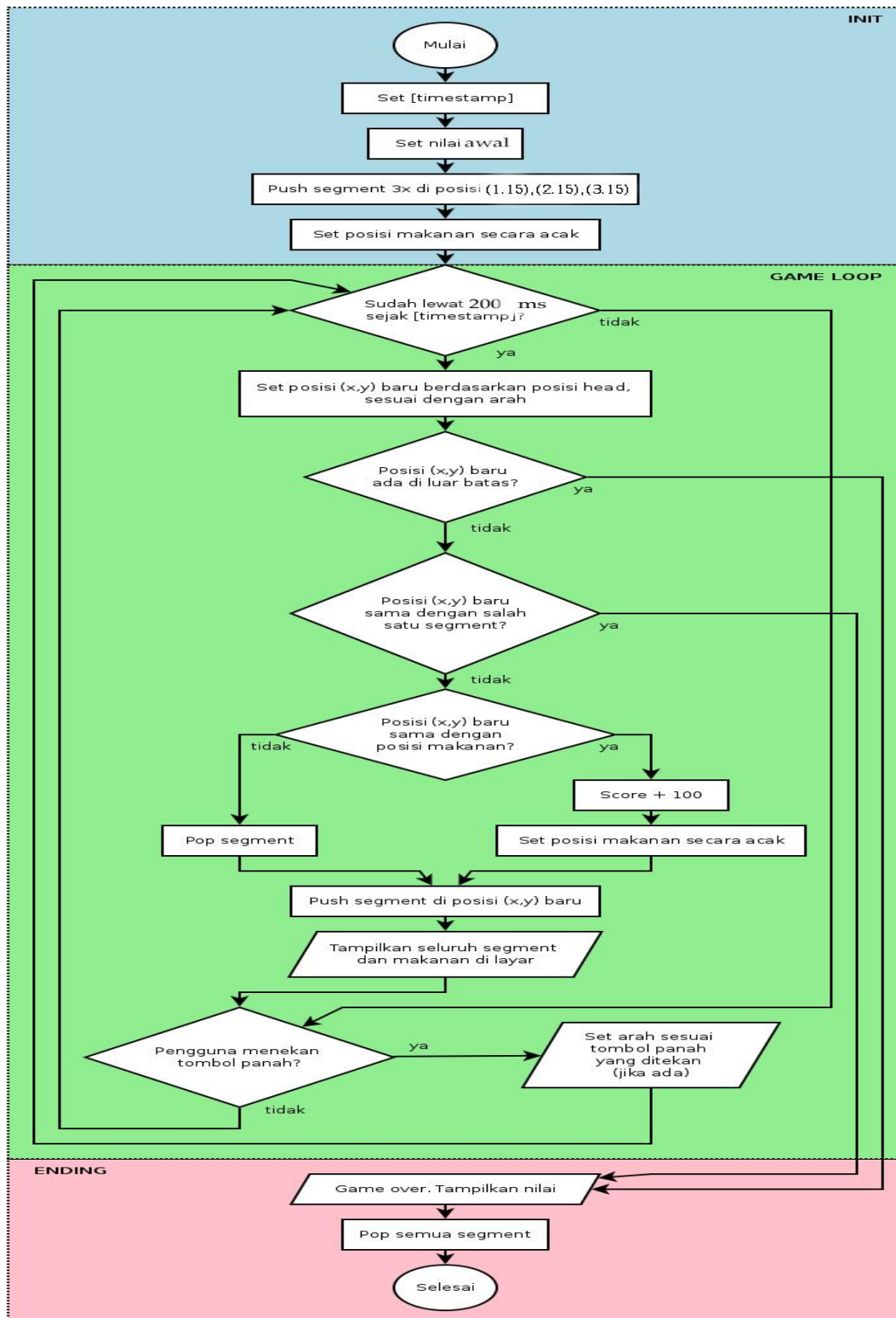
1.2 . TUJUAN PEMBUATAN GAME

Tujuan yang ingin dicapai dari pembuatan game ini adalah untuk membuat sebuah Game Snake yang memiliki semua fungsi dari permainan ular tradisional dan Permainan ini membawa kesenangan dan kesederhanaan permainan ular.

1.3. KONSEP DASAR GAME SNAKE

Game sejenis ular-ularan yang nanti akan memakan sebuah makanan dan akan menjadi bertambah panjang seiring dengan banyaknya makanan yang iya makan. Game akan berakhir jika snake tersebut memakan atau menabrakan dirinya sendiri ke tubuhnya

BAB II FLOWCHART



1. Mulai-> menjalankan program
2. Set[timestamp] -> untuk mengambil nilai tanggal/DATE
3. Set nilai awal-> untuk mengatur nilai awal permainan
4. Push segment 3x di posisi default -> untuk menampilkan posisi awal snake
5. Set posisi makanan acak
6. Sudah lewat 200 ms sejak timestamp-> jika timestamp sudah berjalan selama 200ms maka snake akan bergerak
7. Posisi(x,y) baru berdasarkan posisi head sesuai dengan arah ->untuk mencetak head baru sebagai tumpuan untuk bergerak kearah yang ditentukan
8. Posisi x,y baru ada di luar batas-> jika ya, maka program akan menghapus seluruh segment
9. Posisi x,y baru sama dengan salah satu segment-> jika ya program akan menghapus seluruh segment
10. Posisi x,y sama dengan posisi makanan-> jika ya score ditambah 100 lalu ->set makanan secara acak
11. Jika tidak-> maka pop tail
12. Push segment di posisi baru
13. Tampilan seluruh segment dan makanan di layar(DISPLAY)
14. Pengguna menekan tombol arah, -> untuk menentukan direction kemana snake melaju->jika ya set sesuai dengan tombol panah yang ditekan Jika tidak, tetap bergerak ke arah sebelumnya.

BAB III PEMBAHASAN

3.1. RANCANGAN PROGRAM

Berikut adalah konsep dasar rancangan yang dijabarkan dalam beberapa bentuk fungsi dibawah ini.

3.1.1. SYNTAX STRUCT

```
struct segment{
    int x,y;
    struct segment *next, *prev;
}*head,*tail,*current,*temp;
```

Penggunaan Struct Segment sebagai representasi dan inisialisasi dari variabel x (inisialisasi dari cartesian border melintang/horizontal dari command prompt), variabel y (membujur/vertikal). Kemudian representasikan variabel pointer sebagai alamat dari link list nantinya.

3.1.2. SYNTAX OPERASI FILE SKOR_READ

```
int skor_read()
{
    FILE *rd;
    int high_score;
    rd = fopen("Highscore.txt","r");
    if(!rd){
        return 0;
    }
    else{
        while(!feof(rd)){
            fscanf(rd,"%d",&high_score); fflush(stdin);
        }
        fclose(rd);
    }
    return high_score;
}
```

Representasi sebuah variabel pointer *rd sebagai penunjuk akses dalam membuka file highscore dan membacanya.

3.1.3. SYNTAX OPERASI FILE SKOR_WRITE

```
int skor_write(int score)
{
    FILE *wr;
    int skor;
    skor = score;
    wr = fopen("Highscore.txt","w");
    if(!wr){
        return 0;
    }
    else{
        fprintf(wr,"%d", skor);
    }
    fclose(wr);
}
```

Representasi sebuah variabel pointer *wr sebagai alamat akses dalam mencatat hasil skoring di file highscore.

3.1.4. SYNTAX FUNGSI PUSH

```
void push(int x, int y)
{
    current = (struct segment*)malloc(sizeof(struct segment));
    current->next = current->prev = NULL;
    current->x = x;
    current->y = y;
    if(head == NULL){
        head = tail = current;
    }
    else{
        current->next = head;
        head->prev = current;
        head = current;
    }
}
```

1. Fungsi ini merepresentasikan border - border cartesian dari command prompt, dimana x adalah border sumbu melintang (+) dan (-) begitupula dengan y sebagai border membujur, dimana hal ini adalah titik awal munculnya segmen dari badan ular.

2. Buat list baru sebagai penampil segment, dalam hal ini digunakan konsep double list, segment ini adalah dasar dari proses push dari masing - masing segment badan ular.

3.1.5. SYNTAX FUNGSI POP

```
void pop()
{
    if(head == tail){
        free(head);
        head = tail = NULL;
    }
    else{
        tail = tail->prev;
        free(tail->next);
        tail->next = NULL;
    }
}
```

Sebagai list penghapusan dari badan snake jika maju beberapa langkah.

3.1.6. SYNTAX FUNGSI POP_ALL

```
void pop_all()
{
    while(head!=NULL){
        pop();
    }
}
```

Sebagai list penghapusan keseluruhan badan snake jika mengenai dirinya atau border pembatas.

3.1.7. SYNTAX FUNGSI GOTOXY

```
void gotoxy(int x, int y)
{
    COORD pos;
    pos.X = x;
    pos.Y = y;
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), pos);
}
```


Fungsi ini adalah representasi dari penempatan arah kursor sesuai bidang cartesian command prompt positif X dan Y.

3.1.8. SYNTAX FUNGSI DISPLAY

```
void display()
{
    temp = head;
    while(temp!=NULL){
        gotoxy(temp->x,temp->y);
        printf("o");
        temp = temp->next;
    }
```

Fungsi ini akan menampilkan segmen dari badan ular dengan menyertakan pula fungsi gotoxy sebagai titik tampil awal pada koordinat cartesian command prompt yang telah di set dan ditentukan.

3.1.9. SYNTAX INISIALISASI VARIABEL SET DARI SNAKE

```
int snake_size = 3;
int snake_speed = 200;
int dir = VK_RIGHT;
int game_width = 56;
int game_height = 25;
int food_x, food_y;
```

Variabel - variabel diatas merepresentasikan :

snake_size = mewakili panjang badan segmen snake yang tampil.

snake_speed = mewakili kecepatan snake berjalan tiap detiknya.

dir = merepresentasikan Virtual Key (tombol keyboard) sebagai langkah awal snake berjalan.

game_width = ukuran & batasan panjang melintang dari border map permainan.

game_height = ukuran & batasan lebar vertikal dari border map permainan.

food_x, food_y = titik - titik koordinat tampilnya makanan snake.

3.1.10. SYNTAX FUNGSI CEK

```
int check(int x, int y)
{
    temp = head;
    while(temp!=NULL){
        if(temp->x == x && temp->y == y){
            return 1;
        }
        temp = temp->next;
    }
    return 0;
}
```

Fungsi ini menjabarkan tentang pengecekan segmen badan snake bersentuhan dengan border batas dari map permainan yang penempatannya sesuai koordinat cartesian x dan y. Return 0 artinya tidak bertumpuk, dan 1 artinya bertumpuk.

3.1.11. SYNTAX FUNGSI PLACE_FOOD

```
void place_food()
{
    int x,y;
    do{
        food_x = rand()%(game_width-1) + 1;
        food_y = rand()%(game_height-1) + 1;
    }while (check(food_x, food_y) == 1);
}
```

Fungsi yang merepresentasikan randomize munculnya makanan snake dengan menggunakan mod, -1 menandakan bahwa makanan tidak akan muncul langsung di border batas map dan +1 menandakan makanan tidak akan muncul di badan snake. `while (check (food_x, food_y) == 1)` selalu diulang untuk memastikan munculnya food tidak bersentuhan langsung dengan badan snake.

3.1.12. SYNTAX FUNGSI LOAD

```
void load(){
    int i;
    printf("\n\n\t\tSilahkan tunggu");
    for(i=0;i<5;i++){
        Sleep(600);
        printf(" . ");
    }
    system("cls");
}
```

```

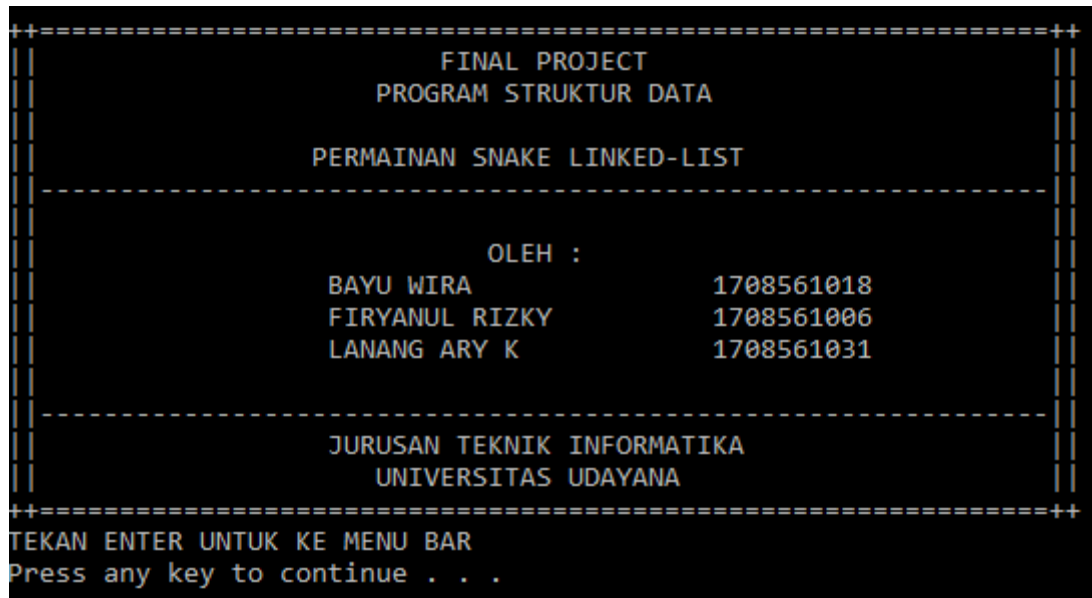
    for(i=0;i<5;i++){
        printf("\n\n\t\t%d",5-i);
        Sleep(1000);
        system("cls");
    }
    printf("\n\n\t\tPERMAINAN DIMULAI");
    Sleep(2000);
    system("cls");
}

```

Fungsi ini ditampilkan pada saat awal permainan sebagai representasi munculnya loading dan angka - angka.

3.2. *TAMPILAN OUTPUT GAME SNAKE*

3.2.1. Tampilan Saat Program dimulai



3.2.2. Tampilan Saat permainan dimulai



3.2.3. Tampilan saat permainan selesai

```
GAME OVER!  
your score      : 0  
HIGH SCORE     : 600  
COBA LAGI !  
Process returned 0 (0x0)   execution time : 14.182 s  
Press any key to continue.
```

BAB IV PENUTUP

4.1. KESIMPULAN

Kesimpulan ini ditujukan untuk mengevaluasi materi yang telah didapatkan, dengan tujuan menambah daya kreatifitas untuk kemudian dapat dikembangkan lebih kompleks lagi di masa mendatang. Program ini dibuat dengan dasar implementasi dari gabungan metode struct, stack, link list, pointer, operasi file yang telah didapat sebelumnya.