**LAPORAN PRAKTIKUM STRUKTUR DATA**

**PROGRAM GAME RPG SEDERHANA BAHASA C**



I Made Tangkas Wahyu Kencana Yuda (1608561031)

**PROGRAM STUDI TEKNIK INFORMATIKA FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM UNIVERSITAS UDAYANA**

**2017**

BAB I

LANDASAN TEORI

1.1.Struktur Dasar Bahasa C

Bahasa C adalah salah satu bahasa pemrograman yang terstruktur. Setiap program C mempunyai satu fungsi dengan nama “main” (program utama). Program akan dieksekusi dimulai dari statement pertama pada fungsi “main” tersebut. Huruf besar dengan huruf kecil diartikan berbeda (case sensitive). Setiap statement diakhiri dengan semi-colon (titik koma (;)). Contoh penulisan bahasa C:

#include <stdio.h> Int main()

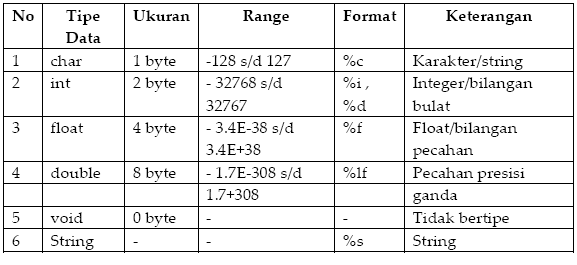
{

Printf(“Hello World!”);

}

1.2.Tipe Data

Dalam bahasa C terdapat lima macam tipe data dasar, yaitu



1.3.Operator

Terdapat tiga macam operator yang biasa digunakan dalam pemrograman, yaitu :

 Operator Aritmatika

Ada lima operator aritmatika yang bisa digunakan di bahasa C:

\* : untuk perkalian

/ : untuk pembagian

% : untuk sisa pembagian (modulus)

+ : untuk pertambahan

- : untuk pengurangan

 Operator Relasional(Perbandingan)

Operator ini membandingkan dua operan dan dapat dilakukan pada dua nilai dengan tipe data yang sama.

< Kurang dari

<= Kurang dari sama dengan

> Lebih dari

>= Lebih dari sama dengan

== Sama dengan

!= Tidak sama dengan

 Operator Logika

Operator logika digunakan untuk membandingkan dua kondisi logika. Operator logika ada tiga macam:

&& : Logika AND (dan)

|| : Logika OR (atau)

! : Logika NOT (ingkaran)

1.4. Memasukkan Data (Input Data)

Fungsi pustaka yang digunakan untuk memasukkan data adalah scanf(). Beberapa hal yang perlu diperhatikan dalan menggunakan scanf():

 Fungsi scanf() memakai penentu format

 Fungsi scanf() memberi pergantian baris secara otomatis

 Operator alamat & ditulis didepan variabel

Kode penentu format:

 %c : Membaca sebuah karakter

 %s : Membaca sebuah string

 %i, %d : Membaca sebuah bilangan bulat (integer)

 %f, %e : Membaca sebuah bilangan pecahan (real)

 %o : Membaca sebuah bilangan octal

 %x : Membaca sebuah bilangan heksadesimal

 %u : Membaca sebuah bilangan tak bertanda.

1.5. Menampilkan Data (Output Data)

Untuk menampilkan dapat menggunakan fungsi printf(), puts(), dan putchar(). Fungsi printf() digunakan untuk menampilkan semua jenis data (numeric dan karakter). Fungsi puts() digunakan untuk menampilkan data string dan secara otomatis akan diakhiri dengan perpindahan baris. Fungsi putchar() digunakan untuk menampilkan sebuah karakter.

1.6.Pemilihan

Penyeleksian kondisi digunakan untuk mengarahkan perjalanan suatu proses. Kekurangan komputer adalah tidak mampu berpikir sendiri sehingga semua hal yang dilakukan berdasarkan perintah. Dengan adanya seleksi konsdisi, program dapat menentukan proses apa yang harus dilakukan selanjutnya berdasarkan keadaaan sebelumnya.

1.7. Operator Relasi

Operator ini membandingkan dua operan dan dapat dilakukan pada dua nilai dengan tipe data yang sama:

< Kurang dari

<= Kurang dari sama dengan

> Lebih dari

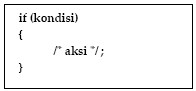
>= Lebih dari sama dengan

== Sama dengan

!= Tidak sama dengan

1.8. Struktur Dasar Pemilihan If

Struktur pemilihan ini digunakan jika kondisi hanya ada satu.

Cara kerjanya sederhana, jika ekspresi di dalam tanda kurung yang benar, maka pernyataan/aksi di dalam kurung kurawal yang akan dijalankan, jika tidak, maka mereka akan diabaikan. Selain penggunaan if dengan satu kondisi, terdapat juga penggunaan if – else dan if – else if – else.

1.9. Perulangan

Perulangan atau iterasi atau yang biasa disebut dengan “looping” adalah proses melakukan tindakan yang sama secara berulang-ulang atau berkali-kali sampai batas yang telah ditentukan. Perulangan digunakan untuk menjalankan satu atau beberapa pernyataan sebanyak beberapa kali.

1.10. Struktur Perulangan WHILE

Fungsi dari perulangan while adalah untuk melakukan tugas berulang selama pernyataan kondisional tertentu adalah bernilai benar. Berikut adalah cara penulisan truktur while.

while (kondisi statement nilai benar){

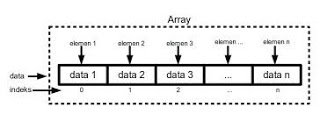
/\*aksi\*/;

}

1.11. Array

Array adalah variabel yang mampu menyimpan sejumlah nilai yang bertipe sama. Untuk mendeklarasikan sebuah array, harus disebutkan tipe dari array yang dibuat misalnya int, float atau double dan juga ukuran array. Untuk menentukan ukuran array,perlu ditempatkan jumlah nilai yang dapat disimpan array dalam sebuah tanda kurung kurawal siku yang terletak sesudah nama array. Suatu aray berdimensi satu di deklarasikan dalam bentuk umum berupa : Tipe\_data nama\_var[ukuran];

Tipe\_data untuk menyatakan tipe dari elemen arrat, misalnya int, char, float. Nama\_var untuk nama variable array. Ukuran untuk menyatakan jumlah maksimal elemen array. Contoh pendeklarasian array : Float nilai [5]; Menyatakan bahwa variable nilai bertipe array of float dan memiliki 5 elemen bertipe float.

Berikut ini gambar yang dapat mengilustrasikan sebuah array :

1.12. Struct

Struct adalah tipe data bentukan yang berisi kumpulan variabel-variabel yang berada dalam satu nama yang sama dan memiliki kaitan satu sama lain. Berbeda dengan array hanya berupa kumpulan variabel yang bertipe data sama, struct bisa memiliki variabel-variabel yang bertipe data sama atau berbeda, bahkan bisa menyimpan variabel yang bertipe data array atau struct itu sendiri. Cara penulisan struct sebagai berikut :

Menggunakan typedef

typedef struct nama\_struct  
{

tipe\_data <nama\_var>;

tipe\_data <nama\_var>;  
};   
contoh :  
typedef struct Mahasiswa  
{

char NIM[8];

char nama[50];

float ipk;

};

Tanpa Menggunakan typedef

struct nama\_struct  
{

tipe\_data field1; Atau

tipe\_data field2;

tipe\_data fieldn;

};

contoh :

struct mahasiswa  
{

char nim[11];

char nama[30]; Atau

char alamat[50];

float ipk;

}

1.13 Linked List

Linked list adalah sekumpulan elemen bertipe sama, yang mempunyai keterurutan tertentu, yang setiap elemennya terdiri dari dua bagian Linked list juga merupakan suatu cara untuk menyimpan data dengan struktur sehingga dapat secara otomatis menciptakan suatu tempat baru untuk menyimpan data yangdiperlukan. Struktur ini lebih dinamis karena banyaknya elemen dengan mudah ditambah atau dikurangi, berbeda dengan array yang ukurannya tetap. berikut gambaran kecil mengenai linked list. Didalam Linked List terdapat beberapa bagian lagi

BAB II

PERMASALAHAN

Membuat sebuah game sederhana berbasis teks dengan ketentuan setiap player mempunyai nilai *power, defense,* dan *attack*

BAB III

PEMBAHASAN

Berikut adalah source code program

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

typedef struct player

{

int power,attack,deff,id,stts;

char namep[10];

struct player \*head,\*next,\*tail,\*temp,\*baru;

}player;

typedef struct room

{

int power,attack,deff,stts,id;

char namep[10];

struct room \*headp,\*nextp,\*prev,\*tempp,\*ruang,\*tempa;

}room;

player \*next,\*head,\*tail,\*temp,\*baru;

room \*headp = NULL,\*nextp,\*prev,\*tempp,\*ruang,\*tempa;

void view ()

{

temp = head;

printf("No Nama\tPower\tAttack\tDeff\tStatus\n\n");

while (temp != NULL)

{

printf("%d %s\t%d\t%d\t%d\t%d\n",temp->id,temp->namep,temp->power,temp->attack,temp->deff,temp->stts);

temp=temp->next;

}

}

int main()

{

//Character

int play=0,pil,x,no=1;

head= (player\*) malloc(sizeof(player));

head->power = 200;head->attack = 47;head->deff = 50;

strcpy(head->namep, "Matumbaman");

head->id = 1;

head->stts = 0;

baru= (player\*) malloc(sizeof(player));

head->next = baru;

baru->id = 2;

baru->stts = 0;

baru->power = 200; baru->attack = 52; baru->deff = 45;

strcpy(baru->namep, "Arteezy");

temp=baru;

baru= (player\*) malloc(sizeof(player));

strcpy(baru->namep, "Sumail");

baru->id = 3;

baru->stts = 0;

temp->next = baru;

baru->power = 200; baru->attack = 49; baru->deff = 44;

temp=baru;

baru= (player\*) malloc(sizeof(player));

temp->next = baru;

baru->id = 4;

baru->stts = 0;

strcpy(baru->namep, "Iceiceice");

baru->power = 200; baru->attack = 57; baru->deff = 35;

temp=baru;

baru= (player\*) malloc(sizeof(player));

temp->next = baru;

baru->id = 5;

baru->stts = 0;

strcpy(baru->namep, "Miracle-");

baru->power = 200; baru->attack = 52; baru->deff = 42;

temp=baru;

baru= (player\*) malloc(sizeof(player));

temp->next = baru;

baru->id = 6;

baru->stts = 0;

strcpy(baru->namep, "MidOne");

baru->power = 200; baru->attack = 54; baru->deff = 49;

baru->next = NULL;

system("cls");

view();

//Pemilihan Character

while (play != 2)

{

do

{

printf("\n\tPlayer %d : ", play+1);

scanf("%d", &pil);

//Jika memilih character yang sama dan pilihan yang dipilih tidak ada

if(pil > 0 || pil < 6 || pil!= x)

{

system("cls");

view();

tempp=headp;

if(play = 1)

{

printf("\n\tPlayer %d :\n",play);

printf("\t%s\n\tPower : %d\n\tAttack : %d\n\tDeffense : %d\n\n",tempp->namep,tempp->power, tempp->attack, tempp->deff);

}

}

}while(pil <= 0 || pil > 6 || pil == x);

x=pil;

play++;

pilihan(pil);

system("cls");

view();

tempp=headp;

do

{

printf("\n\tPlayer %d :\n",no);

printf("\t%s\n\tPower : %d\n\tAttack : %d\n\tDeffense : %d\n\n",tempp->namep,tempp->power, tempp->attack, tempp->deff);

tempp=tempp->nextp;

if(temp==NULL)

break;

no++;

}while(tempp != headp);

}

duel();

return 0;

}

void pilihan (int pil)

{

temp=head;

int i;

while (temp != NULL)

{

if(temp->id == pil)

{

temp->stts = 1;

ruang = (room\*) malloc(sizeof(room));

ruang->attack = temp->attack;

ruang->deff = temp->deff;

ruang->power = temp->power;

ruang->id=i;

strcpy(ruang->namep,temp->namep);

i++;

break;

}

temp=temp->next;

}

if(headp == NULL)

{

headp = ruang;

ruang->stts = 0;

ruang->nextp=NULL;

}

else

{

headp->nextp = ruang;

ruang->stts=1;

ruang->nextp=headp;

}

}

void duel ()

{

char pil;

int no,temp[2];

int serang[2],bertahan[2],x,stts=0;

char back;

tempp=headp;

ruang=tempp->nextp;

//proses bertarung

while(tempp->power > 1)

{

def:

playero();

//giliran setiap pemain

printf("Player %d Turn\n", tempp->stts+1);

//pilih serang atau bertahan

printf("\n1. Serang\n2. Bertahan");

printf("\nPilihan : ");

pil=getch();

x=tempp->stts;

switch(pil)

{

case '1': serang[x]=tempp->attack;

temp[x]=1;

break;

case '2': bertahan[x]=tempp->deff;

temp[x]=2;

break;

default : goto def;

}

tempp->power = tempp->power - 5;

//proses pengurangan power

if(stts==1)

{

stts=0;

if(temp[0] == 1 && temp[1] ==1)

{

ruang->power=ruang->power - serang[0];

headp->power=headp->power - serang[1];

}

else if(temp[0] == 1 && temp[1] ==2)

{

serang[0]=serang[0] - ruang->deff;

if(serang[0] > 0)

ruang->power=ruang->power - serang[0];

}

else if(temp[0] == 2 && temp[1] ==1)

{

serang[1]=serang[1] - headp->deff;

if(serang[1] > 0)

headp->power=headp->power - serang[1];

}

if(ruang->power < 1)

{

ruang->power=0;

break;

}

if(headp->power < 1)

{

headp->power = 0;

break;

}

}

else

{

stts++;

}

tempp=tempp->nextp;

}

playero();

//announcement

if (headp->power == 0 && ruang->power == 0)

printf("Power is Equal, the final Result was Draw\n\n");

else if(ruang->power == 0)

printf("Congratulation The Winner is Player 1\n\n");

else if (headp->power==0)

printf("Congratulation The Winner is Player 2\n\n");

tempa->power = 200;

printf("Play Again ? [Y/N] : "); fflush(stdin);

scanf("%c", &back);

if(back == 'Y' || back=='y')

main();

else

return 0;

}

void playero ()

{

int no;

no=1;

system("cls");

tempa=headp;

do

{

printf("\n\tPlayer %d :\n",no);

printf("\t%s\n\tPower : %d\tAttack : %d\tDeffense : %d\n\n",tempa->namep,tempa->power, tempa->attack, tempa->deff);

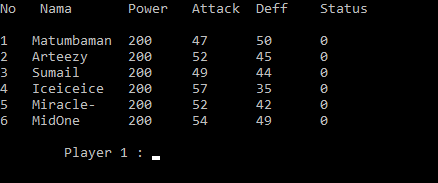
tempa=tempa->nextp;

no++;

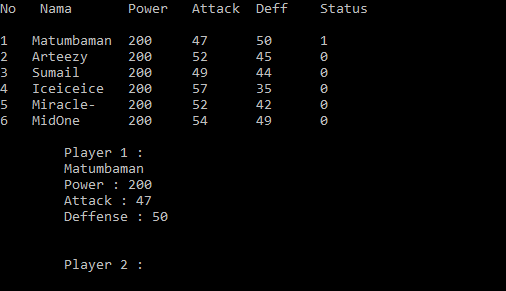
}while(tempa != headp );

}

Berikut tampilan program saat di run

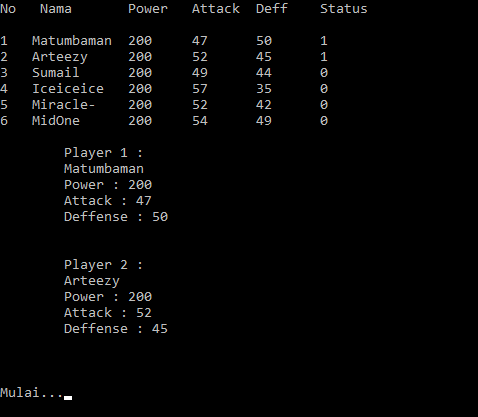


***Gambar 3.1. Tampilan Awal dan pemilihan pemain***

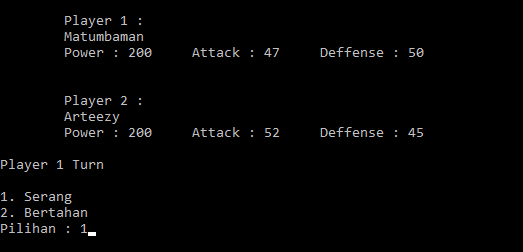


***Gambar 3.2. Tampilan Pemain yang dipilih ( Player 1 )***

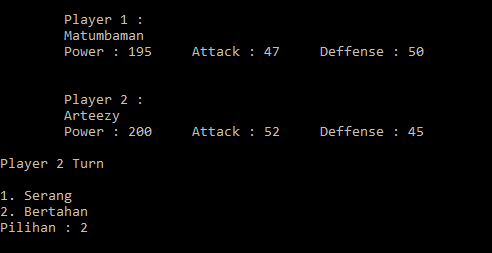
Saat pemain dipilih status character menjadi 1 (Terpilih) sehingga pemain lain tidak dapat memilih character yang sama.



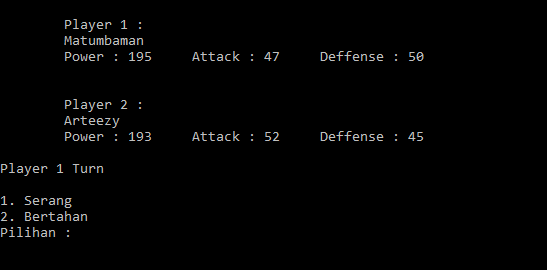
***Gambar 3.3. Tampilan Character yang dipilih ( Player 1 dan Player 2 )***



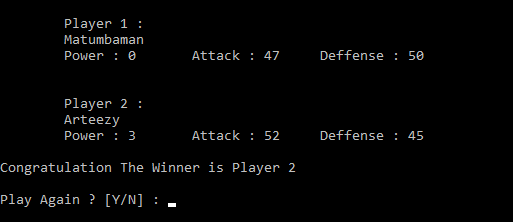
***Gambar 3.4. Tampilan memilih action untuk player 1***



***Gambar 3.5. Tampilan memilih action untuk player 2***



***Gambar 3.6. Tampilan setelah kedua pemain memilih action dan hasil pengurangan Power setiap pemain***



***Gambar 3.6. Tampilan akhir saat salah satu pemain menang***