

LAPORAN PRAKTIKUM STRUKTUR DATA
PROGRAM GAME RPG SEDERHANA BAHASA C



Frisca Olivia Gorianto (1608561022)

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

No	Tipe Data	Ukuran	Range	Format	Keterangan
1	char	1 byte	-128 s/d 127	%c	Karakter/string
2	int	2 byte	- 32768 s/d 32767	%i , %d	Integer/bilangan bulat
3	float	4 byte	- 3.4E-38 s/d 3.4E+38	%f	Float/bilangan pecahan
4	double	8 byte	- 1.7E-308 s/d 1.7+308	%lf	Pecahan presisi ganda
5	void	0 byte	-	-	Tidak bertipe
6	String	-	-	%s	String

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 dalam 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 kondisi, program dapat menentukan proses apa yang harus dilakukan selanjutnya berdasarkan keadaan 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.

```
if (kondisi)
{
    /* aksi */;
}
```

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 struktur 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 array berdimensi satu di deklarasikan dalam bentuk umum berupa : Tipe_data nama_var[ukuran];

Tipe_data untuk menyatakan tipe dari elemen array, misalnya int, char, float. Nama_var untuk nama variabel array. Ukuran untuk menyatakan jumlah maksimal elemen array. Contoh pendeklarasian array : Float nilai [5]; Menyatakan bahwa variabel nilai bertipe array of float dan memiliki 5 elemen bertipe float.

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.

Variabel-variabel yang menjadi anggota struct disebut dengan elemen struct. Berikut adalah penulisan struct.

```
struct  NILAI  //struct nama_struct
{
    char    nama[100];
    float   math;
}

int main()
{
    struct NILAI p[10];
    //proses
}
```

Bab II

Permasalahan

Membuat sebuah game RPG 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>

struct data

{
    int pow;
    int att;
    int def;
};

int main()
{
    int saya, lawan;
    struct data player[100];
    player[0].pow=100;player[0].att=50;player[0].def=28;
    player[1].pow=100;player[1].att=40;player[1].def=45;
    player[2].pow=100;player[2].att=60;player[2].def=48;
    player[3].pow=100;player[3].att=50;player[3].def=29;
    player[4].pow=100;player[4].att=30;player[4].def=56;

    printf("\t|Player\t\t\t1\t2\t3\t4\t5\t\t|\n");
    printf("\t|Power\t\t\t%d\t%d\t%d\t%d\t%d\t\t|\n", player[0].pow,
player[1].pow, player[2].pow, player[3].pow, player[4].pow);
    printf("\t|Attack\t\t\t%d\t%d\t%d\t%d\t%d\t\t|\n",player[0].att,
player[1].att,player[2].att, player[3].att, player[4].att);
    printf("\t|Defense\t\t\t%d\t%d\t%d\t%d\t%d\t\t|\n",player[0].def,
player[1].def,player[2].def, player[3].def, player[4].def);

    printf("\n\nPilih Pemain (1/2/3/4/5)\n");
    printf("Saya: "); scanf("%d",&saya);
    printf("Lawan: "); scanf("%d",&lawan);

    saya=saya-1;
    lawan=lawan-1;

    while(player[saya].pow > 0 && player[lawan].pow > 0)
    {
        system("cls");
        int act1, acts, mins;

        printf("\t|Pemain\t\t\tSaya\tLawan\t|\n");
        printf("\t|Power\t\t\t%d\t%d\t|\n", player[saya].pow,
player[lawan].pow);
        printf("\t|Attack\t\t\t%d\t%d\t|\n",player[saya].att,
player[lawan].att);
```

```

        printf("\t|Defense\t%d\t%d\t|\n",player[saya].def,
player[lawan].def);

        printf("\nChoose your action: 1)Attack  2)Defend\n");
        printf("Saya: "); scanf("%d",&acts);
player[saya].pow=player[saya].pow - 10;
        printf("Lawan: ");scanf("%d", &actl);
player[lawan].pow=player[lawan].pow - 10;

        if(acts==1 && actl==1)
        {
            if(player[saya].att > player[lawan].def)
            {
                mins= player[saya].att - player[lawan].def;
                player[lawan].pow = player[lawan].pow - mins;
            }

            if(player[saya].def < player[lawan].att)
            {
                mins= player[lawan].att - player[saya].def;
                player[saya].pow = player[saya].pow - mins;
            }
        }

        if(acts==1 && actl==2)
        {
            if(player[saya].att > player[lawan].def)
            {
                mins= player[saya].att - player[lawan].def;
                player[lawan].pow = player[lawan].pow - mins;
            }
        }

        if(acts==2 && actl ==1)
        {
            if(player[saya].def < player[lawan].att)
            {
                mins= player[lawan].att - player[saya].def;
                player[saya].pow = player[saya].pow - mins;
            }
        }

    }

    system("cls");
    printf("\t|Pemain\t\tSaya\tLawan\t|\n");
    printf("\t|Power\t\t%d\t%d\t|\n", player[saya].pow,
player[lawan].pow);
    printf("\t|Attack\t\t%d\t%d\t|\n",player[saya].att,
player[lawan].att);
    printf("\t|Defense\t%d\t%d\t|\n",player[saya].def,
player[lawan].def);
    if(player[saya].pow <= 0)
        printf("Pemain Saya Kalah!!!\n");
    else

```



```

        printf("Pemain Lawan Kalah!!!\n");
        printf("Permainan Selesai\n\n");

        return 0;
}

```

Berikut adalah hasil run program

The screenshot shows a terminal window with the following content:

Player	1	2	3	4	5
Power	100	100	100	100	100
Attack	50	40	60	50	30
Defense	28	45	48	29	56

Pilih Pemain (1/2/3/4/5)
 Saya: 2
 Lawan: 4_

Gambar 3.1. Tampilan Awal dan pemilihan pemain

The screenshot shows a terminal window with the following content:

Pemain	Saya	Lawan
Power	100	100
Attack	40	50
Defense	45	29

Choose your action: 1)Attack 2)Defend
 Saya: 2
 Lawan: 1

Gambar 3.2. Tampilan memilih action attack atau defend beserta tabel nilai status pemain

The screenshot shows a terminal window with the following content:

Pemain	Saya	Lawan
Power	85	90
Attack	40	50
Defense	45	29

Choose your action: 1)Attack 2)Defend
 Saya: 1
 Lawan: 1_

Gambar 3.3. Tampilan memilih action lagi dan pada tabel status nilai power pemain sudah berkurang

```
"C:\Users\Frisca\Documents\Kuliah\Praktikum Struktur Data\game_hero_Frisca Olivia Goriantc

| Pemain      Saya  Lawan |
| Power       10    -14    |
| Attack      40    50     |
| Defense     45    29     |
Pemain Lawan Kalah!!!
Permainan Selesai

Process returned 0 (0x0)   execution time : 16.550 s
Press any key to continue.
```

Gambar 3.4. Tampilan Akhir dengan pernyataan pemain mana yang kalah beserta tabel status pemain