LAPORAN PRAKTIKUM STRUKTUR DATA

ARRAY DAN STRUCT

Oleh:

Ida Bagus Pranawangsa (1608561024)

JURUSAN ILMU KOMPUTER

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS UDAYANA

2017

A. Landasan Teori

1. Array

Array adalah struktur data yang menyimpan sekumpulan elemen yang bertipe sama, setiap elemen diakses langsung melalui indexnya. Index dari array haruslah berupa tipe data yang menyatakan keterurutan, misalnya integer. Contoh dari penamaan array, sebuah array yang bernama A dengan n buah enemen dapat dibayangkan dengan logis seperti sebuah kotak yang dilabeli, dimana label tersebut merupakan index dari array, yang tersusun terurut. Setiap elemen array ditulis sebagai A[0], A[1], ... , A[n].

Setiap elemen array menyimpan sebuah nilai. Karena seluruh elemen array bertipe sama, maka nilai yang disimpan oleh setiap elemen jug harus bertipe sama. Bentuk standar dari pendeklarasian array adalah *tipedata* <namavar> [*arraysize*].

Indeks array dimulai dari nol (0), sedangkan nomor elemen biasanya dimulai dari satu (1). Nomor elemen dapat dibuat sama dengan nomor indeks untuk mempermudah pembuatan program yaitu dengan memberi indeks satu lebih banyak dari jumlah data yang dibutuhkan.

2. Struct

Struct adalah pengelompokan variabel-variabel yang bernaung dalam satu nama yang sama. Berbeda dengan array yang berisi kumpulan variabel-variabel yang bertipe sama dalam satu nama, maka suatu struct dapat terdiri atas variabel-variabel yang berbeda tipenya dalam satu nama struct. Struct biasa dipakai untuk mengelompokkan beberapa informasi yang berkaitan menjadi satu.

Variabel-variabel yang membentuk suatu struktur, selanjutnya disebut sebagai elemen dari struktur atau field. Dengan demikian dimungkinkan suatu struktur dapat berisi elemen-elemen data berbeda tipe seperti char, int, float, double, dan lain-lain.

Contoh pendeklarasiannya :

```
struct date
{
      int  month;
      int  day;
      int  year;
};
```

Elemen dari suatu variabel struct dapat diakses dengan menyebutkan nama variabel struct diikuti dengan operator titik (‘.’) dan nama dari elemen structnya. Bentuk umumnya : variabel_struct.nama_field.

B. Permasalahan

Buatlah suatu permainan sederhana dengan memanfaatkan array dan struct, dimana gameplay dari permainan tersebut berupa player vs player (PvP). Player terakhir yang tetap hidup menjadi pemenangnya.

C. Pembahasan

1. Source Code

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>

#define atk_gen 100
#define def_gen 50
#define pwr_gen 400
#define def_count_max 2

struct attr
{
    int name;
    int role;
    int atk;
    int def;
    int def_count;
    int pwr;
    int stance;
    int cur_target;
    bool stat,atk_reduction;
};
int main()
{
    srand(time(NULL));
    struct attr player[5],comb[2];
    int turn,i,combat,winner,dumm;
    bool finish_flag=false;
    for(i=0;i<5;i++)//defining role
    {
        system("cls");
            printf("Role available : \n1. Fighter (+atk, -def)\n2.
Guard(--atk, +def, +pwr)\n3. Magi(+++atk, -def, --pwr)");
        printf("\nPlayer %d role : ",i+1);
        scanf("%d",&player[i].role);
        player[i].stat=true;
        player[i].atk_reduction=false;
        player[i].def_count=def_count_max;
        player[i].name=i;
        if(player[i].role==1)
        {
```

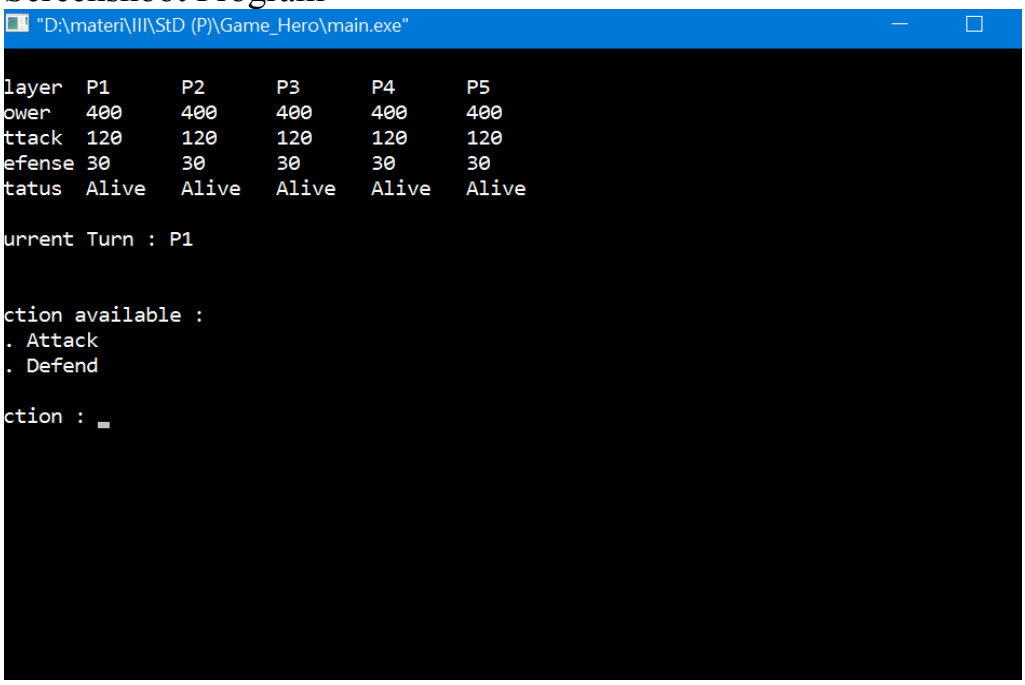
```
            player[i].atk=atk_gen+20;
            player[i].def=def_gen-20;
            player[i].pwr=pwr_gen;
        }
        else if(player[i].role==2)
        {
            player[i].atk=atk_gen-40;
            player[i].def=def_gen+20;
            player[i].pwr=pwr_gen+100;
        }
        else if(player[i].role==3)
        {
            player[i].atk=atk_gen+60;
            player[i].def=def_gen-20;
            player[i].pwr=pwr_gen-150;
        }
    }
    while(finish_flag==false)//Gameplay
    {
        for(turn=0;turn<5;turn++)//Target phase
        {
            system("cls");
            if(player[turn].stat==true)
            {
                player_table(player);
                printf("\n\nCurrent Turn : P%d\n",turn+1);
                     printf("\n\nAction available : \n1. Attack\n2.
Defend");
                printf("\n\nAction : ");
                scanf("%d",&player[turn].stance);
                if(player[turn].stance==1)
                {
                    printf("Select Player Target : ");
                    do
                    {
                     scanf("%d",&player[turn].cur_target);
                              }while(player[player[turn].cur_target-
1].stat==false||
player[turn].cur_target==player[turn].name+1);//prevent self-attack
                }
                                                                    else
if(player[turn].stance==2&&player[turn].def_count!=0)
player[turn].cur_target=0;
                    else//out of defense, stance become attack with
temporal 25% atk reduction, random target
                {
                    player[turn].stance=1;
                    player[turn].atk_reduction=true;
                    do
                    {
                        player[turn].cur_target=rand()%5;
```

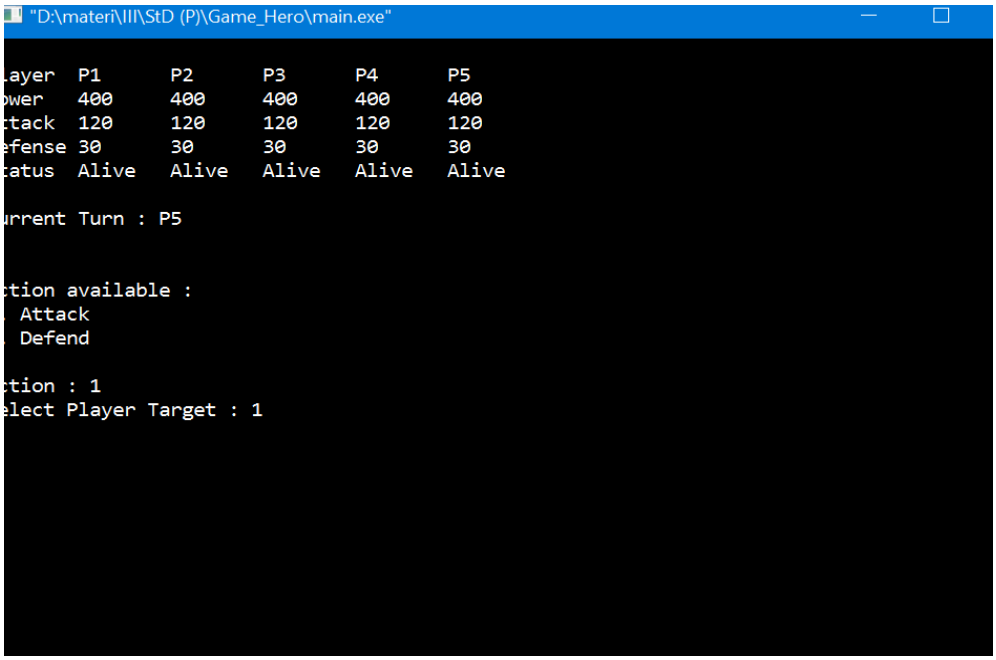
```
                                    }while(player[turn].cur_target==0||
player[player[turn].cur_target-1].stat==false||
player[turn].cur_target==turn||player[turn].cur_target>5);//prevent
self-attack
                }
            }
        }
        for(turn=0;turn<5;turn++)//def reduction phase
        {
            if(player[turn].stance==2) player[turn].def_count-=1;
        }
        for(turn=0;turn<5;turn++)//Combat phase
        {
            //comb[0] = acting player, comb[1] = targeted player
             if(player[turn].stance==1&&player[turn].stat==true)//act
ing player have not died
            {
                comb[0]=player[turn];
                comb[1]=player[player[turn].cur_target-1];
                player[turn].cur_target=0;
                 if(comb[0].stance==1&&comb[1].stance==1&&comb[1].sta
t==true)//atk stance vs atk stance,target have not died
                {
                                        if(comb[0].atk_reduction==true)
comb[0].atk*=0.75;
                                        if(comb[1].atk_reduction==true)
comb[1].atk*=0.75;
                                if(comb[1].role!=3)combat=comb[0].atk-
((comb[1].atk/2)+comb[1].def);//if target is not magi
                               else  combat=comb[0].atk-(comb[1].def);//if
target is magi
                    if(combat<0)//backfire
                    {
                        player[turn].pwr+=combat;
                        if(player[turn].pwr<=0)
                        {
                             player[turn].stat=false;//backfirelse;//
dead
                            player[turn].pwr=0;
                        }
                    }
                    else
                    {
                        player[comb[0].cur_target-1].pwr-=combat;
                        if(player[comb[0].cur_target-1].pwr<=0)
                        {
                                            player[comb[0].cur_target-
1].stat=false;//dead
                            player[comb[0].cur_target-1].pwr=0;
                        }
                    }
```

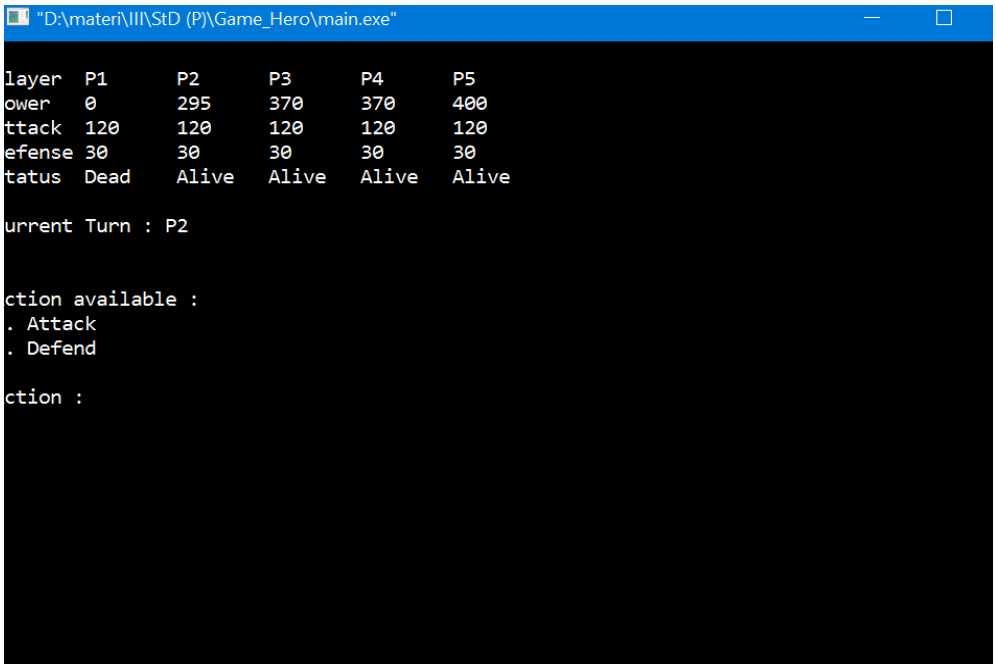
```
                }
                                                                    else
if(comb[0].stance==1&&comb[1].stance==2&&comb[1].stat==true)//atk
stance vs def stance
                {
                                        if(comb[0].atk_reduction==true)
comb[0].atk*=0.75;
                                  if(comb[1].role!=2)combat=comb[0].atk-
(comb[1].def*1.5);//if target is not guard
                           else combat=comb[0].atk-(comb[1].def*2);//if
target is guard
                    if(combat<0)//backfire
                    {
                        player[turn].pwr+=combat;
                        if(player[turn].pwr<=0)
                        {
                            player[turn].stat=false;//dead
                            player[turn].pwr=0;
                        }
                    }
                    else
                    {
                        player[comb[0].cur_target-1].pwr-=combat;
                        if(player[comb[0].cur_target-1].pwr<=0)
                        {
                                            player[comb[0].cur_target-
1].stat=false;//dead
                            player[comb[0].cur_target-1].pwr=0;
                        }
                    }
                }
            }
        }
        for(turn=0;turn<5;turn++)//def count reset
        {
            if(player[turn].atk_reduction==true)
            {
                player[turn].def_count=def_count_max;
                player[turn].atk_reduction=false;
            }
        }
        winner=player_status(player);
        if(winner!=0) finish_flag=true;
    }
    printf("Player %d won the battle",winner);
    return 0;
}

void player_table(struct attr player[])
{
    int j;
    printf("\nPlayer\t");
    for(j=0;j<5;j++) printf("P%d\t",j+1);
    printf("\nPower\t");
```

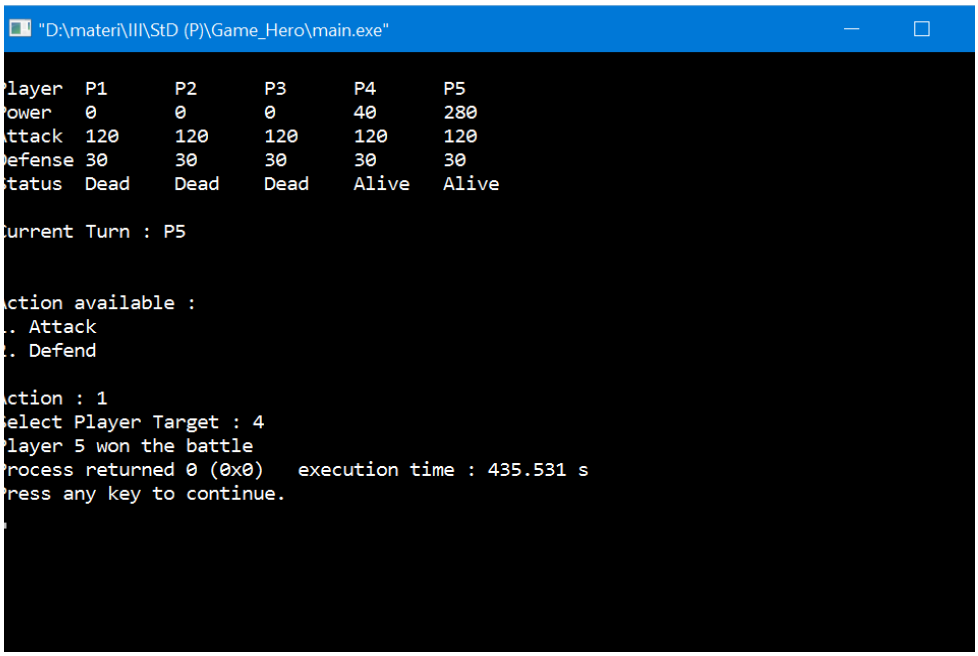
```
    for(j=0;j<5;j++) printf("%d\t",player[j].pwr);
    printf("\nAttack\t");
    for(j=0;j<5;j++) printf("%d\t",player[j].atk);
    printf("\nDefense\t");
    for(j=0;j<5;j++) printf("%d\t",player[j].def);
    printf("\nStatus\t");
    for(j=0;j<5;j++)
    {
        if(player[j].stat==true) printf("Alive\t");
        else printf("Dead\t");
    }
}

int player_status(struct attr player[])
{
    int l,player_count=0,lastplayeralive;
    for(l=0;l<5;l++)
    {
        if(player[l].stat==true)
        {
            player_count++;
            lastplayeralive=l+1;
        }
    }
    if(player_count>1) return 0;
    else return lastplayeralive;
}
```

2. Screenshoot Program




"D:\materi\III\StD (P)\Game_Hero\main.exe"
layer P1 P2 P3 P4 P5
ower 400 400 400 400 400
ttack 120 120 120 120 120
efense 30 30 30 30 30
tatus Alive Alive Alive Alive Alive
urrent Turn : P5
ction available :
. Attack
. Defend
ction : 1
elect Player Target : 1


"D:\materi\III\StD (P)\Game_Hero\main.exe"
layer P1 P2 P3 P4 P5
ower 0 295 370 370 400
ttack 120 120 120 120 120
efense 30 30 30 30 30
tatus Dead Alive Alive Alive Alive
urrent Turn : P2
ction available :
. Attack
. Defend
ction :


"D:\materi\III\StD (P)\Game_Hero\main.exe"
layer P1 P2 P3 P4 P5
ower 0 0 0 40 280
ttack 120 120 120 120 120
efense 30 30 30 30 30
tatus Dead Dead Dead Alive Alive
urrent Turn : P5
ction available :
. Attack
. Defend
ction : 1
elect Player Target : 4
layer 5 won the battle
rocess returned 0 (0x0) execution time : 435.531 s
ress any key to continue.