**LAPORAN FINAL PROJECT GAME**

**“HUNT”**



Dosen Pengampu

Dr. Eko Mulyanto Yuniarno, S.T., M.T.

Disusun oleh

Indiana Namaul Husnah

NRP 5024201061

Pemograman Lanjut B

**DEPARTEMEN TEKNIK KOMPUTER**

**FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS**

**INSTITUT TEKNOLOGI SEPULUH NOPEMBER**

**2021**

**KATA PENGANTAR**

Puji syukur penulis ucapkan kepada Tuhan Yang Maha Esa berkat taufik dan hidayah-Nya penulis dapat menyelesaikan laporan final project game dengan judul game “HUNT”. Penulis juga mengucapkan terima kasih kepada dosen mata kuliah Pemograman Lanjut B, yaitu Bapak Dr. Eko Mulyanto Yuniarno, S.T., M.T. dan kepada semua pihak yang telah berpartisipasi baik moril maupun materil dalam pembuatan laporan final project game ini.

Laporan final project game ini dibuat untuk memenuhi tugas mata kuliah Pemograman Lanjut, serta sebagai referensi kedepannya terkait final project game ini.

Penulis menyadari dalam penyusunan laporan final project ini masih terdapat kekurangan. Oleh karena itu, penulis mengharapkan masukan dan kritikan dari semua pihak terkait dengan relevansi laporan final project game ini agar bisa menjadi lebih baik di masa yang akan datang.

Tanjungbatu, 7 Juli 2021

Penulis

**DAFTAR ISI**

KATA PENGANTAR ii

DAFTAR ISI iii

**BAB I PENDAHULUAN**

* 1. Latar Belakang 1
  2. Rumusan Masalah 1
  3. Tujuan 1

**BAB II DASAR TEORI**

* 1. Pemograman C/C++ 2
  2. Pemograman C/C++ menggunakan Library Graphics.h 3
  3. Pemograman dengan OOP (Object Oriented Programming) 4

**BAB III PENJELASAN PROGRAM**

* 1. Objek Game 5
  2. Membuat Objek 5
  3. Menggerakkan Objek 8
  4. Score dan Musik 9
  5. User Interface Game 10
  6. Gameplay 13

**BAB IV LISTING PROGRAM**

* 1. Listing Program 21
  2. Screenshot Program 58

**BAB V PENUTUP**

* 1. Kesimpulan 60
  2. Saran 60

**DAFTAR PUSTAKA**

**BAB I**

**PENDAHULUAN**

* 1. **Latar Belakang**

Komputer adalah alat yang digunakan untuk mengolah sesuatu dan tidak dapat dipisahkan dari kehidupan manusia saat ini. Manusia dapat dengan mudah mengolah maupun mengerjakan sesuatu menggunakan berbagai program yang dijalankan oleh komputer. Program yang dijalankan oleh komputer merupakan sesuatu yang bisa dibuat untuk menyelesaikan masalah, mengolah sesuatu, bahkan bisa untuk digunakan dalam hal hiburan.

Pemograman adalah suatu rangkaian instruksi yang akan melakukan suatu fungsi tertentu berdasarkan instruksi yang diberikan. Dalam membuat program, terdapat berbagai bahasa pemograman yang setiap bahasa tersebut tergantung kepada fungsi dan pemakaiannya. Bahasa pemograman yang sering digunakan antara lain, Javascript, Phyton, C, C++, PHP, dan lain-lain. Game adalah salah satu program di komputer yang pada umumnya berfungsi sebagai hiburan dan kesenangan. Dalam membuat suatu program, sering programmer merasa bosan dan tidak tertarik dalam mengerjakan hal tersebut. Oleh karena itu, membuat program game dipilih menjadi tugas ini karena menarik dan tidak membosankan.

* 1. **Rumusan Masalah**

Berdasarkan latar belakang di atas, dapat ditarik rumusan masalah yaitu bagaimana membuat game dengan *library* Graphics.h menggunakan bahasa pemograman C/C++.

* 1. **Tujuan**

Berdasarkan rumusan masalah di atas, tujuan yang diharapkan yaitu membuat game dengan *library* Graphics.h dengan menggunakan bahasa pemograman C/C++.

**BAB II**

**DASAR TEORI**

**2.1 Pemograman C/C++**

C++ adalah suatu bahasa pemograman komputer yang diciptakan oleh Bjarne Stroustrup di AT&T Bell Laboratories pada awal tahun 1980-an. C++ merupakan perkembangan dari bahasa C yang dikembangkan oleh Dennis Ritchie pada tahun 1970-an. Simbol dari ++ menunjukkan bahwa bahasa C++ merupakan versi yang lebih baru dari bahasa C. Komponen dari bahasa C++ antara lain :

**a. Header**

Header adalah suatu file yang berisi fungsi-fungsi yang akan digunakan dalam memprogram bahasa pemograman. File header dalam C++ ini terdapat library tertentu yang dapat digunakan untuk memanggil fungsi tertentu sehingga bahasa bisa dapat digunakan. Contoh dari file adalah iostream, stdio.h, conio.h, math.h, dan lain-lain

**b. Fungsi Utama**

Ketika membuat suatu kode, terdapat suatu tempat utama di mana suatu kode tersebut berjalan. Tempat utama yang dimaksud adalah tempat dimana seluruh fungsi yang telah dibuat akan dipanggil ke dalam suatu tempat utama ini. Fungsi utama ini disebut main(). Jadi pada fungsi inilah semua fungsi yang lain akan dipanggil dan dilakukan proses. Hal ini dikarenakan C/C++ adalah bahasa prosedural yang menerapkan konsep runtutan.

**c. Tipe Data dan Deklarasi Variabel**

Tipe Data adalah suatu nilai yang akan ditampung pada variabel. Tipe data dalam bahasa pemograman C++ memiliki panjang bit dan range tertentu tergantung tipenya. Contoh tipe data yaitu char, unsigned char, int, float, double, bool, dan lain-lain. Variabel adalah suatu tempat untuk menampung suatu nilai tertentu. Dalam deklarasi variabel ini berarti mengenalkan variabel dengan menentukan tipe datanya. Contoh dari mendeklarasikan variabel yaitu :

**int** jumlah; **float** harga; **char** nilai;

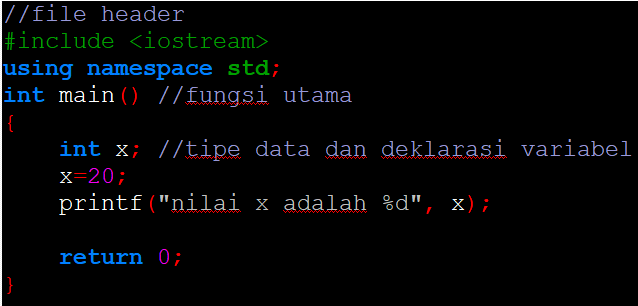
**d. Konstanta**

Konstanta yaitu suatu nilai tetap yang tidak bisa diubah ketika suatu program dijalankan. Dlam pendeklerasianya, penulisan sintaks adalah seperti berikut : **const** mahasiswa = 20;

**e. Komentar**

Komentar adalah komponen untuk memberikan penjelasan terkait sintaks program yang tidak akan dieksekusi oleh program. Cara penulisan komentar dalam bahasa C/C++ yaitu dengan menggunakan garis miring dua kali atau menggunakan garis miring dan bintang.

Berikut adalah contoh program sederhana berdasarkan komponen-komponen bahasa C++ yang telah disebutkan di atas.



**Gambar 2.1 Program Sederhana Bahasa C/C++**

Program di atas adalah program untuk menampilkan suatu nilai yang disimpan dalam suatu variabel. Dari sintaks di atas, terdapat file header yaitu #include <iostream>, include disini bermaksud memasukkan library-library yang terdapat di iostream untuk bisa digunakan dalam program ini. Kemudian di baris selanjutnya terdapat fungsi utama, fungsi utama dideklarasikan terlebih dahulu dengan menambahkan kurung kurawal sebagai pembatas. Kemudian terdapat tipe data yaitu int x dan baris data selanjutnya untuk menampilkan isi dari variabel x.

**2.2 Pemograman C/C++ menggunakan Library Graphics.h**

Graphics.h merupakan sebuah header yang memungkinkan untuk membuat grafis seperti garis, lingkaran, oval, gambar, dan lain-lain. Pemograman C/C++ dengan menggunakan graphics.h dapat dimulai dengan penginputan header di awal listing program dan membuka windows BGI dengan menggunakan fungsi initwindow. Graphics.h memiliki banyak fungsi-fungsi tertentu yang bisa membuat suatu grafik di dalam windows BGI. Fungsi-fungsi yang sering digunakan yaitu seperti :

void [arc](https://home.cs.colorado.edu/~main/cs1300/doc/bgi/arc.html) (int x, int y, int stangle, int endangle, int radius);

void [circle](https://home.cs.colorado.edu/~main/cs1300/doc/bgi/circle.html) (int x, int y, int radius);

void [line](https://home.cs.colorado.edu/~main/cs1300/doc/bgi/line.html) (int x1, int y1, int x2, int y2);

void [setbkcolor](https://home.cs.colorado.edu/~main/cs1300/doc/bgi/setbkcolor.html) (int color);

void [setcolor](https://home.cs.colorado.edu/~main/cs1300/doc/bgi/setcolor.html) (int color);

void [outtext](https://home.cs.colorado.edu/~main/cs1300/doc/bgi/outtext.html) (char \*textstring);

void [outtextxy](https://home.cs.colorado.edu/~main/cs1300/doc/bgi/outtextxy.html) (int x, int y, char \*textstring);

**2.3 Pemograman dengan OOP (Object Oriented Programming)**

Object Oriented Programming adalah suatu metode pemograman dengan berorientasi kepada objek. Metode OOP ini membagikan objek-objek menjadi bagian-bagian kecil agar memudahkan dalam pengembangan program.

**BAB III**

**PENJELASAN PROGRAM**

**3.1 Objek Game**

Dalam program game ini, terdapat 11 fungsi objek yang diantaranya 5 fungsi di dalam class dan sisanya fungsi di luar class. Fungsi-fungsi tersebut yaitu :

Fungsi di dalam class CKapal :

sea()

fish()

fish2()

KapalKN1()

KKE()

Fungsi lainnya :

sun(float xS, float yS, float sc)

moon(float xM, float yM, float sc)

clouds(float xc, float yc, float sc)

gambar(float xp, float yp, float sc

harpoon(float xh, float yh, float sc)

fishe(float xf, float yf, float sc)

**3.2 Membuat Objek**

Untuk membuat objek-objek seperti di 3.1, dapat dilakukan dengan sintaks seperti ini: void line (int x1, int y1, int x2, int y2);

int x1 dan int y1 merupakan deklarasi dengan tipe data integer untuk menunjukkan koordinat atau titik awal untuk dilakukan sebuah garis (line), dan int x2 dan int y2 merupakan koordinat atau titik akhir agar terbentuk sebuah garis, karena garis merupakan kumpulan sebuah titik. Contohnya seperti fungsi objek fish() yaitu sebagai berikut :

float x[25], y[25];

int i; sc=10;

x[0]=-2.42;y[0]=-0.045;

x[1]=-2.1494;y[1]=-0.0244;

x[2]=-2.422;y[2]=0.154;

x[3]=-2.162;y[3]=0.32;

x[4]=-1.724;y[4]=0.52;

x[5]=-1.08;y[5]=0.665;

x[6]=-0.202;y[6]=0.674;

x[7]=0.635;y[7]=0.525;

x[8]=1.28;y[8]=0.24;

x[9]=1.54;y[9]=0.24;

x[10]=2.26;y[10]=0.72;

x[11]=1.925;y[11]=0.035;

x[12]=2.29;y[12]=-0.46;

x[13]=1.532;y[13]=-0.184;

x[14]=1.2933;y[14]=-0.18405;

x[15]=1.085;y[15]=-0.265;

x[16]=0.75;y[16]=-0.4;

x[17]=-0.36;y[17]=-0.555;

x[18]=-1.385;y[18]=-0.515;

x[19]=-1.85;y[19]=-0.345;

x[20]=-2.42;y[20]=-0.045;

setcolor(COLOR(201, 194, 109));

setfillstyle(SOLID\_FILL,COLOR(201, 194, 109));

for(i=0;i<20;i++){

line(x[i]\*sc+xf,-y[i]\*sc+yf,x[i+1]\*sc+xf,-y[i+1]\*sc+yf);

}

floodfill(0.2\*sc+xf,0\*sc+yf,COLOR(201, 194, 109));



**Gambar 3.1.1 Objek fish()**

Dalam membuat objek fish() ini, perlu dilakukan deklarasi titik x dan titik y terlebih dahulu dengan menggunakan array dengan isi array tergantung dari berapa banyak titik yang akan digunakan dalam membuat objek. Setelah dideklarasikan, bisa menggunakan perulangan for untuk memanggil fungsi line(x[i]\*sc+xf,-y[i]\*sc+yf,x[i+1]\*sc+xf,y[i+1]\*sc+yf); sesuai dengan jumlah koordinat yang telah dideklarasikan. Sc dan xf yang ditulis pada sintaks line ini adalah skala dan parameter yang digunakan untuk pemanggilan fungsi di fungsi utama.

Kemudian, terdapat objek-objek lain yang menggunakan fungi line ini seperti objek karakter, kapal, dan lain-lain, yaitu seperti gambar dibawah ini.



**Gambar 3.1.2**

Agar objek-objek tersebut berwarna, maka bisa dipanggil fungsi fungsi sebagai berikut.

setcolor(COLOR);

setfillstyle(SOLID\_FILL,COLOR);

// sintaks gambar bisa berupa line, rectangle, arc, dan lain-lain

floodfill(0.2\*sc+xf,0\*sc+yf,COLOR);

Untuk itu, perlu dilakukan pemanggilan fungsi setcolor untuk mewarnai objek, setfillstyle untuk pengaturan pola pengisian warna ke dalam objek, dan floodfill adalah pengaturan untuk memasukkan warna ke dalam objek dengan koordinat yang tepat.

**3.3 Menggerakkan Objek**

Terdapat beberapa fungsi untuk menggerakkan objek, fungsi tersebut yaitu GerakKanan(), gerakfishA(), gerakfishB(), gerakfishC(), fungsi-fungsi tersebut akan menggerakkan objek, objek tersebut yaitu fish(), fish2(), dan KapalKN1. Kemudian, terdapat fungsi di dalam fungsi gameplay() untuk menggerakkan objek seperti moon(), clouds(), dan sun(). Dalam menggerakkan objek, terdapat key yang harus ditekan terlebih dahulu agar objek bisa berjalan, maka dari itu terdapat sebuah sintaks percabangan dalam fungsi ini. Contohnya seperti sintaks gerakKanan() berikut ini :

void GerakKanan(){

if( GetAsyncKeyState(VK\_RIGHT) & 0x8000){

xh=xh+5; xp=xp+5;}

if( GetAsyncKeyState(VK\_LEFT) & 0x8000){

xh=xh-5; xp=xp-5;}

}

Kemudian contoh lain yaitu sintaks dari gerakfishA(), yaitu :

void gerakfishA(){

xf=xf-fx\*2;

yf=yf-fy\*0.2;

if(xf<-100){

xf=600;}

if(yf<400){

fy=-1;}

if(yf>800){

fy=1;}

}

Untuk menggerakkan objek, diperlukan parameter dari fungsi objek tersebut. Setelah dapat parameter dari fungsi objek tersebut, parameter akan dijadikan koordinat x dan koordinat y untuk menggerakkannya, lalu koordinat x dan koordinat y dikurang atau ditambah dengan angka tertentu, semakin besar angka, maka semakin cepat objek tersebut bergerak. Agar objek tersebut bergerak ke arah tertentu dan memiliki batas tertentu, maka dibuat suatu variabel yang bisa berubah-ubah sesuai dengan kondisi, variabel akan bernilai minus atau plus sesuai kondisi tertentu tergantung objek akan digerakkan ke arah mana.

**3.4 Score dan Musik**

Untuk membuat score di dalam game ini, tahap pertama mendeklarasikan terlebih dahulu score di awal program dengan menggunakan tipe data float dan char. Float digunakan untuk menampilkan score sedangkan char score digunakan untuk menyimpan score di dalam array tersebut.

float scoreK=0; char score[99];

Setelah mendeklarasikan score, di fungsi gameplay(), score dipanggil dengan sintaks seperti ini :

outtextxy(10, 10, score); // ambil score dari array score[99]

sprintf(score,"score = %.f", scoreK); //tampilkan score dari scoreK



**Gambar 3.1.3 Score**

Kemudian untuk memutar musik dalam game, menggunakan sintaks dengan diletakkan pada fungsi dimana akan diputar musik. Contoh memutar musik dengan sintaks seperti ini :

PlaySound(TEXT("water.wav"), NULL, SND\_ASYNC);

Water.wav adalah nama musik yang akan diputar di dalam game dengan berformat file .wav dan file musik tersebut berada di dalam file program game.



**Gambar 3.1.4 File musik .wav**

**3.5 User Interface Game**

Terdapat beberapa fungsi dalam main menu game, yaitu mainmenu(), loadscreen(), credit(), howtoplay(), dan gameplay(). Fungsi mainmenu() berfungsi untuk menampilkan menu dalam game, fungsi loadscreen() menampilkan tampilan loading screen ketika game pertama kali dijalankan, fungsi credit() menampilkan tentang pembuat game, fungsi howtoplay() menampilkan cara bermain game, dan fungsi gameplay() adalah tombol untuk menuju dimana game akan dijalankan atau play. Contoh sintaks dari mainmenu() adalah sebagai berikut :

void mainmenu(){

while(1){

rectangle(0,0,600,700);

setfillstyle(SOLID\_FILL,COLOR(209, 227, 255));

floodfill(100,100,COLOR(209, 227, 255));

int mouse\_x, mouse\_y;

mouse\_x=mousex(); mouse\_y=mousey();

setcolor(BLACK);

setbkcolor(COLOR(209, 227, 255));

settextstyle(3,0,8);

outtextxy(205,70,"HUNT");

settextstyle(3,0,3);

outtextxy(280,185,"play");

outtextxy(255,285,"how to play");

outtextxy(275,385,"credit");

outtextxy(285,485,"exit");

rectangle(180,170,420,230);

rectangle(180,270,420,330);

rectangle(180,370,420,430);

rectangle(180,470,420,530);

if((mouse\_y>170)&&(mouse\_y<230)&&(mouse\_x>180)&&(mouse\_x<420)){

if(ismouseclick(WM\_LBUTTONDOWN)){

cleardevice();

beforeplay();

gameplay();

clearmouseclick(WM\_LBUTTONDOWN);}}

if((mouse\_y>270)&&(mouse\_y<330)&&(mouse\_x>180)&&(mouse\_x<420)){

if(ismouseclick(WM\_LBUTTONDOWN)){

cleardevice();

howtoplay();

clearmouseclick(WM\_LBUTTONDOWN);}}

if((mouse\_y>370)&&(mouse\_y<430)&&(mouse\_x>180)&&(mouse\_x<420)){

if(ismouseclick(WM\_LBUTTONDOWN)){

cleardevice();

credits();

clearmouseclick(WM\_LBUTTONDOWN);}}

if((mouse\_y>470)&&(mouse\_y<530)&&(mouse\_x>180)&&(mouse\_x<420)){

if(ismouseclick(WM\_LBUTTONDOWN)){

closegraph();

exit(0);

clearmouseclick(WM\_LBUTTONDOWN);

break;}}} }



**Gambar 3.1.5 Main Menu**

Dalam pembuatan fungsi mainmenu() ini, menggunakan fungsi rectangle untuk membantuk sebuah kotak sesuai dengan menu pada game, kemudian untuk menuju credits, how to play, dan exit menggunakan percabangan (if-else) dengan memasukkan deteksi mouse click. Kemudian berikut adalah contoh sintaks dari fungsi how to play.

void howtoplay(){

cleardevice();

while(1){

setcolor(BLACK);

CKapal KPHowtoplay;

settextstyle(3,0,8);

outtextxy(138,50,"how to play");

settextstyle(3,0,3);

outtextxy(50,185,"press right arrow key -> to moving your boat");

outtextxy(50,215,"press left arrow key <- to moving your boat");

outtextxy(50,245,"press space bar to start hunting!");

KPHowtoplay.initKapalKN(400,600,20);

KPHowtoplay.KapalKN1();

setcolor(WHITE);

gambar(295, 470, 1); gambar(345, 500, 1);gambar(395, 470, 1);

gambar(445, 500, 1);gambar(495, 470, 1);gambar(545, 500, 1);

if(GetAsyncKeyState(27)<0){

cleardevice();

clearmouseclick(WM\_LBUTTONDOWN);

return mainmenu();

}

}

}



**Gambar 3.1.6 How to play**

**3.6 Gameplay**

Untuk menjalankan gameplay dari game ini, pemanggilan fungsi objek dan fungsi lainnya diketik dalam fungsi gameplay() dengan sintaks sebagai berikut.

void gameplay(){

int page = 0;

setbkcolor(BLACK);

float xp,yp,sc;

float xcl, ycl; xcl = 300; ycl = 220;

float f; f=-1;

float xMo, yMo; xMo=500; yMo=20;

float xMs, yMs; xMs=600; yMs=20;

float xh1, yh1; xh1=160; yh1=300;

float xh, yh; xh=160, yh=300;

float bp=0;

float xff, yff;xff=200;yff=600;

float xff1, yff1;xff1=600;yff1=600;

float flagx; flagx=1;

float flagy; flagy=1;

float flagx1; flagx1=1;

float flagy1; flagy1=1;

CKapal h1, h2, sea1, fs1, fs2, fs3, fs4, KP, KEI;

sea1.initSea(200);

KEI.initKapalKN(160,280,6.5);

KP.initKapalKN(200,330,20);

fs1.initfish(600,400,20);

fs2.initfish(666,450,20);

fs3.initfish(566,500,20);

fs4.initfish(599,550,20);;

while(1){

setactivepage(page);

setvisualpage(1-page);

cleardevice();

sea1.sea();

harpoon(xh, yh, sc); harpoon(xh1, yh1, sc);

KEI.KKE(); KEI.GerakKanan();

KP.KapalKN1(); KP.GerakKanan();

fishe(xff, yff, sc); fishe(xff1, yff1, sc);

fs1.fish(); fs1.gerakfishA();

fs2.fish2(); fs2.gerakfishC();

fs3.fish(); fs3.gerakfishB();

fs4.fish2(); fs4.gerakfishC();

//score

setcolor(WHITE);

settextstyle(3,0,3);

outtextxy(10, 10, score); // ambil score dari array score[99]

sprintf(score,"score = %.f", scoreK); //tampilkan score dari scoreK

setcolor(WHITE);

gambar(100, 70, 1); gambar(150, 80, 1); gambar(200, 90, 1);

gambar(350, 50, 1); gambar(400, 70, 1); gambar(450, 80, 1);

//gerak objek clouds

clouds(xcl, ycl, sc);

xcl=xcl+f\*0.4; //0.4

if(xcl>310){

f=-1;}

if(xcl<290){

f=1;}

moon(xMo, yMo, sc); //gerak objek moon

xMo=xMo-0.4;

if(xMo<-100){

sun(450, yMs, sc); xMs=xMs-1;

setbkcolor(COLOR(38, 74, 255));

}

//shoot

if(GetAsyncKeyState(VK\_SPACE) & 0x8000){

PlaySound(TEXT("water.wav"), NULL, SND\_ASYNC);

xh1=xh;

yh1=yh;

bp=1;}

if(bp==1){

yh1=yh1+6; xh1=xh1-5;

if((xh1>0.42\*20+xff)&&(xh1<6.29\*20+xff)&&(yh1>0.024\*20+yff)&&(yh1<0.665\*20+yff))

{

xff=700; yff=700; scoreK+=1; }

if((xh1>0.42\*20+xff1)&&(xh1<6.29\*20+xff1)&&(yh1>-0.024\*20+yff1)&&(yh1<0.665\*20+yff1))

{ xff1=900; yff1=800; scoreK+=1; }

}

if( GetAsyncKeyState(VK\_RIGHT) & 0x8000) {

xh=xh+5; }

if( GetAsyncKeyState(VK\_LEFT) & 0x8000){

xh=xh-5; }

xff=xff-flagx\*7; //gerak fish

yff=yff-flagy\*0.5;

if(xff<-100){

xff=900;}

if(yff<400){

flagy=-1; }

if(yff>800){

flagy=1; }

xff1=xff1-flagx1\*5;

yff1=yff1+flagy1\*2;

if(xff1<-100){

xff1=800; }

if(yff1<400){

flagy1=1; }

if(yff1>800){

flagy1=-1; }

//gameover

if(xMo<-140){

cleardevice();

setbkcolor(BLACK);

gameover(); }

delay(10);

page = 1-page;}

}

Pada fungsi gameplay() ini, pertama dilakukan pendeklereasian terlebih dahulu untuk fungsi apa yang akan dipanggil. Setelah dideklarasikan, maka fungsi-fungsi dari Class maupun di luar Class bisa dipanggil. Di fungsi gameplay terdapat sintaks untuk menjalankan inti dari game. Inti game yang dimaksud adalah ketika karakter mendapatkan score dari menombak ikan, yaitu dengan sintaks berikut.

if(GetAsyncKeyState(VK\_SPACE) & 0x8000){

PlaySound(TEXT("water.wav"), NULL, SND\_ASYNC);

xh1=xh;

yh1=yh;

bp=1;}

if(bp==1){

yh1=yh1+6; xh1=xh1-5;

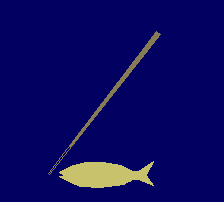
if((xh1>0.42\*20+xff)&&(xh1<6.29\*20+xff)&&(yh1>0.024\*20+yff)&&(yh1<0.665\*20+yff)){

xff=700; yff=700; scoreK+=1; }

if((xh1>0.42\*20+xff1)&&(xh1<6.29\*20+xff1)&&(yh1>-0.024\*20+yff1)&&(yh1<0.665\*20+yff1))

{ xff1=900; yff1=800; scoreK+=1; }

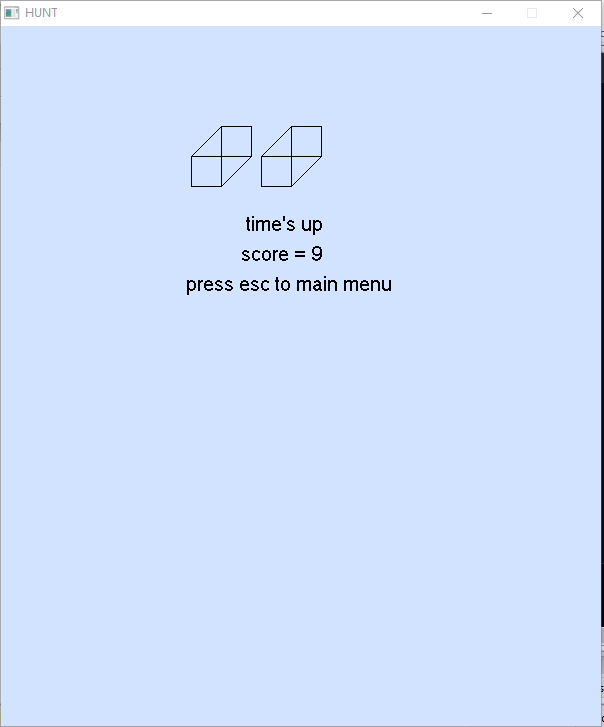
}



**Gambar 3.1.7 Gameplay**

Sintaks diatas adalah inti dari game, ketika user menekan tombol space, maka tombak akan meluncur serong ke bawah, ketika mengenai ikan, maka ikan akan hilang, dan score bertambah, selagi score bertambah, fungsi fish() atau ikan tersebut akan muncul di tempat sesuai dengan koordinat pada sintaks di atas. Fungsi untuk objek tombak mendeteksi ketika mengenai objek ikan adalah dengan menggunakan percabangan if-else. Agar objek tombak mendeteksi mengenai objek, maka dilakukan percabangan(if-else) terhadap koordinat x dan koordinat y tertinggi dan terendah dari objek ikan.

Game berakhir ketika objek dari moon() sudah bergerak hingga koordinat -140. Jadi, sintaks game berakhir menggunakan if-else, ketika objek moon() sudah bergerak ke kiri dengan koordinat sudah kurang dari -140, maka pemanggilan fungsi gameover() akan dilakukan dengan memanggil fungsi cleardevice() terlebih dahulu.



**Gambar 3.1.8 Game over**

Setelah membuat fungsi user interface yang telah disebutkan diatas, seperti main menu, loading screen, gameplay, dan lain-lain, maka fungsi-fungsi tersebut akan dipanggil pada fungsi utama atau int main() dengan sintaks sebagai berikut.

int main(){

initwindow(600,700, "HUNT");

loadscreen();

sndPlaySound("gameplay.wav",SND\_ASYNC|SND\_LOOP);

mainmenu();

gameplay();

}

Fungsi utama terdapat initwindow yang berarti membuka windows BGI dengan ukuran tertentu, kemudian diikuti dengan fungsi loadscreen dan seterusnya sesuai bagaimana game akan dijalankan.

**BAB IV**

**LISTING PROGRAM**

**4.1 Listing Program**

#include <iostream>

#include <windows.h>

#include <graphics.h>

#include <math.h>

#include <stdlib.h>

#include <mmsystem.h>

void gameplay(), howtoplay(), beforeplay(),credits();

float scoreK=0;

char score[99];

using namespace std;

class CKapal

{

public:

float xh,yh,rh;

float xL;

float xp, yp;

float xf, yf, sc;

// int f1=1; //fish

float fx=1; float fy=1;

//float fhy=1;

// float bp=0;

void initSea(float xsea)

{

xL=xsea;

}

void GerakKanan(){

if( GetAsyncKeyState(VK\_RIGHT) & 0x8000)

{

xh=xh+5; xp=xp+5;

}

if( GetAsyncKeyState(VK\_LEFT) & 0x8000)

{

xh=xh-5; xp=xp-5;

}

}

void sea(){

int stangle = 50, endangle = 135;

int stangle2 = 135, endangle2 = 50;

int radius = 20;

xL=200;

setcolor(COLOR(2, 0, 97));

setfillstyle(SOLID\_FILL,COLOR(2, 0, 97));

for (int xL=0; xL<=700; xL=xL+50){

arc(xL, 340, stangle, endangle, radius);

}

for (int xL=25; xL<=700; xL=xL+50){

arc(xL, 310, -stangle2, -endangle2, radius);

}

floodfill(xL,340,COLOR(2, 0, 97));

}

void initfish(float xfish, float yfish, float scfish)

{

xf=xfish;

yf=yfish;

sc=scfish;

}

void fish()

{

float x[25], y[25];

int i;

sc=10;

x[0]=-2.42;y[0]=-0.045;

x[1]=-2.1494;y[1]=-0.0244;

x[2]=-2.422;y[2]=0.154;

x[3]=-2.162;y[3]=0.32;

x[4]=-1.724;y[4]=0.52;

x[5]=-1.08;y[5]=0.665;

x[6]=-0.202;y[6]=0.674;

x[7]=0.635;y[7]=0.525;

x[8]=1.28;y[8]=0.24;

x[9]=1.54;y[9]=0.24;

x[10]=2.26;y[10]=0.72;

x[11]=1.925;y[11]=0.035;

x[12]=2.29;y[12]=-0.46;

x[13]=1.532;y[13]=-0.184;

x[14]=1.2933;y[14]=-0.18405;

x[15]=1.085;y[15]=-0.265;

x[16]=0.75;y[16]=-0.4;

x[17]=-0.36;y[17]=-0.555;

x[18]=-1.385;y[18]=-0.515;

x[19]=-1.85;y[19]=-0.345;

x[20]=-2.42;y[20]=-0.045;

setcolor(COLOR(201, 194, 109));

setfillstyle(SOLID\_FILL,COLOR(201, 194, 109));

for(i=0;i<20;i++){

line(x[i]\*sc+xf,-y[i]\*sc+yf,x[i+1]\*sc+xf,-y[i+1]\*sc+yf);

}

floodfill(0.2\*sc+xf,0\*sc+yf,COLOR(201, 194, 109));

}

void fish2()//fish hadap kiri

{

float x[25], y[25];

int i;

sc=10;

x[0]=2.42;y[0]=0.045;

x[1]=2.1494;y[1]=0.0244;

x[2]=2.422;y[2]=-0.154;

x[3]=2.162;y[3]=-0.32;

x[4]=1.724;y[4]=-0.52;

x[5]=1.08;y[5]=-0.665;

x[6]=0.202;y[6]=-0.674;

x[7]=-0.635;y[7]=-0.525;

x[8]=-1.28;y[8]=-0.24;

x[9]=-1.54;y[9]=-0.24;

x[10]=-2.26;y[10]=-0.72;

x[11]=-1.925;y[11]=-0.035;

x[12]=-2.29;y[12]=0.46;

x[13]=-1.532;y[13]=0.184;

x[14]=-1.2933;y[14]=0.18405;

x[15]=-1.085;y[15]=0.265;

x[16]=-0.75;y[16]=0.4;

x[17]=0.36;y[17]=0.555;

x[18]=1.385;y[18]=0.515;

x[19]=1.85;y[19]=0.345;

x[20]=2.42;y[20]=0.045;

setcolor(COLOR(201, 194, 109));

setfillstyle(SOLID\_FILL,COLOR(201, 194, 109));

for(i=0;i<20;i++){

line(x[i]\*sc+xf,-y[i]\*sc+yf,x[i+1]\*sc+xf,-y[i+1]\*sc+yf);

}

floodfill(0.2\*sc+xf,0\*sc+yf,COLOR(201, 194, 109));

}

void gerakfishA()

{

xf=xf-fx\*2;

yf=yf-fy\*0.2;

if(xf<-100){

xf=600;

}

if(yf<400){

fy=-1;

}

if(yf>800){

fy=1;

}

}

void gerakfishB()

{

xf=xf-fx\*1;

yf=yf-fy\*1;

if(xf<-100){

xf=600;

}

if(yf<400){

fy=-1;

}

if(yf>800){

fy=1;

}

}

void gerakfishC()//ke kanan deh

{

xf=xf+fx\*2;

yf=yf-fy\*0.5;

//kembali ke asal

if(xf>800){

xf=-100;

}

if(yf<400){

fy=-1;

}

if(yf>800){

fy=1;

}

}

void initKapalKN(float xpp, float ypp, float scc){

xp=xpp;

yp=ypp;

sc=scc;

}

void KapalKN1(){

float x0,y0,x1,y1,x2,y2,x3,y3,x4,y4,x5,y5;

float x6,y6,x7,y7,x8,y8,x9,y9,x10,y10,x11,y11,x12,y12,x13,y13;

setcolor(COLOR(179, 90, 39));

//Badan kpl

x0=-4;y0=2;

x1=0;y1=1.79;

x2=4;y2=2;

x3=3;y3=0;

x4=-3;y4=0;

line(x0\*sc+xp,-y0\*sc+yp,x1\*sc+xp,-y1\*sc+yp);

line(x1\*sc+xp,-y1\*sc+yp,x2\*sc+xp,-y2\*sc+yp);

line(x2\*sc+xp,-y2\*sc+yp,x3\*sc+xp,-y3\*sc+yp);

line(x3\*sc+xp,-y3\*sc+yp,x4\*sc+xp,-y4\*sc+yp);

line(x4\*sc+xp,-y4\*sc+yp,x0\*sc+xp,-y0\*sc+yp);

setfillstyle(SOLID\_FILL, COLOR(179, 90, 39));

floodfill(0\*sc+xp,-1\*sc+yp, COLOR(179, 90, 39));

//kiri

setcolor(COLOR(179, 90, 39));

x0=-4;y0=2;

x1=-4.2;y1=2.1;

x2=-4.4;y2=2.2;

x3=-3.2;y3=0;

x4=-3;y4=0;

line(x0\*sc+xp,-y0\*sc+yp,x1\*sc+xp,-y1\*sc+yp);

line(x1\*sc+xp,-y1\*sc+yp,x2\*sc+xp,-y2\*sc+yp);

line(x2\*sc+xp,-y2\*sc+yp,x3\*sc+xp,-y3\*sc+yp);

line(x3\*sc+xp,-y3\*sc+yp,x4\*sc+xp,-y4\*sc+yp);

line(x4\*sc+xp,-y4\*sc+yp,x0\*sc+xp,-y0\*sc+yp);

setfillstyle(SOLID\_FILL, COLOR(179, 90, 39));

floodfill(-3.6\*sc+xp,-1.1\*sc+yp, COLOR(179, 90, 39));

//kanan

setcolor(COLOR(179, 90, 39));

x0=4;y0=2;

x1=4.2;y1=2.1;

x2=4.4;y2=2.2;

x3=3.2;y3=0;

x4=3;y4=0;

line(x0\*sc+xp,-y0\*sc+yp,x1\*sc+xp,-y1\*sc+yp);

line(x1\*sc+xp,-y1\*sc+yp,x2\*sc+xp,-y2\*sc+yp);

line(x2\*sc+xp,-y2\*sc+yp,x3\*sc+xp,-y3\*sc+yp);

line(x3\*sc+xp,-y3\*sc+yp,x4\*sc+xp,-y4\*sc+yp);

line(x4\*sc+xp,-y4\*sc+yp,x0\*sc+xp,-y0\*sc+yp);

setfillstyle(SOLID\_FILL, COLOR(179, 90, 39));

floodfill(3.6\*sc+xp,-1\*sc+yp, COLOR(179, 90, 39));

//tiang

setcolor(LIGHTGRAY);

x0=-0.09;y0=1.79;

x1=-0.09;y1=6;

x2=0.09;y2=6;

x3=0.09;y3=1.79;

line(x0\*sc+xp,-y0\*sc+yp,x1\*sc+xp,-y1\*sc+yp);

line(x1\*sc+xp,-y1\*sc+yp,x2\*sc+xp,-y2\*sc+yp);

line(x2\*sc+xp,-y2\*sc+yp,x3\*sc+xp,-y3\*sc+yp);

line(x3\*sc+xp,-y3\*sc+yp,x0\*sc+xp,-y0\*sc+yp);

setfillstyle(SOLID\_FILL, LIGHTGRAY);

floodfill(0\*sc+xp,-3\*sc+yp, LIGHTGRAY);

//bndera

setcolor(COLOR(255, 239, 212));

x0=0.09;y0=2.2;

x1=0.2;y1=2.2;

x2=4;y2=2.2;

x3=0.2;y3=6;

x4=0.2;y4=6;

x5=0.09;y5=6;

line(x0\*sc+xp,-y0\*sc+yp,x1\*sc+xp,-y1\*sc+yp);

line(x1\*sc+xp,-y1\*sc+yp,x2\*sc+xp,-y2\*sc+yp);

line(x2\*sc+xp,-y2\*sc+yp,x3\*sc+xp,-y3\*sc+yp);

line(x3\*sc+xp,-y3\*sc+yp,x4\*sc+xp,-y4\*sc+yp);

line(x4\*sc+xp,-y4\*sc+yp,x1\*sc+xp,-y1\*sc+yp);

line(x4\*sc+xp,-y4\*sc+yp,x5\*sc+xp,-y5\*sc+yp);

setfillstyle(SOLID\_FILL,COLOR(255, 239, 212));

floodfill(2\*sc+xp,-4\*sc+yp,COLOR(255, 239, 212));

//circle

setcolor(BLACK);

circle(xp-30, yp-24, sc-10);

circle(xp-30, yp-24, sc-13);

circle(xp+30, yp-24, sc-10);

circle(xp+30, yp-24, sc-13);

circle(xp, yp-22, sc-10);

circle(xp, yp-22, sc-13);

}

void KKE(){

float x0,y0,x1,y1,x2,y2,x3,y3,x4,y4,x5,y5;

float x6,y6,x7,y7,x8,y8,x9,y9,x10,y10,x11,y11,x12,y12,x13,y13,x14,y14;

sc=6.5;

//Kke

x0=-1.1;y0=5.7;

x1=-1.9;y1=6.6;

x2=-3.14;y2=7.1;

x3=-3.25;y3=5.65;

x4=-2.9;y4=4.4;

x5=-1.1;y5=5.7;

setcolor(COLOR(255, 208, 133));

//setcolor(WHITE);

line(x0\*sc+xp,-y0\*sc+yp,x1\*sc+xp,-y1\*sc+yp);

line(x1\*sc+xp,-y1\*sc+yp,x2\*sc+xp,-y2\*sc+yp);

line(x2\*sc+xp,-y2\*sc+yp,x3\*sc+xp,-y3\*sc+yp);

line(x3\*sc+xp,-y3\*sc+yp,x4\*sc+xp,-y4\*sc+yp);

line(x4\*sc+xp,-y4\*sc+yp,x5\*sc+xp,-y5\*sc+yp);

setfillstyle(SOLID\_FILL,COLOR(255, 208, 133));

floodfill(-2.2\*sc+xp,-5.14\*sc+yp,COLOR(255, 208, 133));

line(-x0\*sc+xp,-y0\*sc+yp,-x1\*sc+xp,-y1\*sc+yp);

line(-x1\*sc+xp,-y1\*sc+yp,-x2\*sc+xp,-y2\*sc+yp);

line(-x2\*sc+xp,-y2\*sc+yp,-x3\*sc+xp,-y3\*sc+yp);

line(-x3\*sc+xp,-y3\*sc+yp,-x4\*sc+xp,-y4\*sc+yp);

line(-x4\*sc+xp,-y4\*sc+yp,-x5\*sc+xp,-y5\*sc+yp);

setfillstyle(SOLID\_FILL,COLOR(255, 208, 133));

floodfill(2\*sc+xp,-5.14\*sc+yp,COLOR(255, 208, 133));

setcolor(COLOR(255, 208, 133));

x0=0;y0=6;

x1=-1.1;y1=5.7;

x2=-1.95;y2=5.25;

x3=-2.9;y3=4.4;

x4=-3.47;y4=3.51;

x5=-3.675;y5=2.75;

x6=-3.375;y6=1.855;

x7=-2.76;y7=0.985;

x8=-1.878;y8=0.246;

x9=-0.916;y9=0.072;

x10=0;y10=0;

line(x0\*sc+xp,-y0\*sc+yp,x1\*sc+xp,-y1\*sc+yp);

line(x1\*sc+xp,-y1\*sc+yp,x2\*sc+xp,-y2\*sc+yp);

line(x2\*sc+xp,-y2\*sc+yp,x3\*sc+xp,-y3\*sc+yp);

line(x3\*sc+xp,-y3\*sc+yp,x4\*sc+xp,-y4\*sc+yp);

line(x4\*sc+xp,-y4\*sc+yp,x5\*sc+xp,-y5\*sc+yp);

line(x5\*sc+xp,-y5\*sc+yp,x6\*sc+xp,-y6\*sc+yp);

line(x6\*sc+xp,-y6\*sc+yp,x7\*sc+xp,-y7\*sc+yp);

line(x7\*sc+xp,-y7\*sc+yp,x8\*sc+xp,-y8\*sc+yp);

line(x8\*sc+xp,-y8\*sc+yp,x9\*sc+xp,-y9\*sc+yp);

line(x9\*sc+xp,-y9\*sc+yp,x10\*sc+xp,-y10\*sc+yp);

line(-x0\*sc+xp,-y0\*sc+yp,-x1\*sc+xp,-y1\*sc+yp);

line(-x1\*sc+xp,-y1\*sc+yp,-x2\*sc+xp,-y2\*sc+yp);

line(-x2\*sc+xp,-y2\*sc+yp,-x3\*sc+xp,-y3\*sc+yp);

line(-x3\*sc+xp,-y3\*sc+yp,-x4\*sc+xp,-y4\*sc+yp);

line(-x4\*sc+xp,-y4\*sc+yp,-x5\*sc+xp,-y5\*sc+yp);

line(-x5\*sc+xp,-y5\*sc+yp,-x6\*sc+xp,-y6\*sc+yp);

line(-x6\*sc+xp,-y6\*sc+yp,-x7\*sc+xp,-y7\*sc+yp);

line(-x7\*sc+xp,-y7\*sc+yp,-x8\*sc+xp,-y8\*sc+yp);

line(-x8\*sc+xp,-y8\*sc+yp,-x9\*sc+xp,-y9\*sc+yp);

line(-x9\*sc+xp,-y9\*sc+yp,-x10\*sc+xp,-y10\*sc+yp);

setfillstyle(SOLID\_FILL,COLOR(255, 208, 133));

floodfill(-0.3\*sc+xp,-2.05\*sc+yp,COLOR(255, 208, 133));

//bdn

// setcolor(COLOR(255, 208, 133));

x0=-1.878;y0=0.246;

x1=-2.21;y1=-0.54;

x2=-2.48;y2=-1.28;

x3=-2.62;y3=-1.97;

x4=-2.275;y4=-2.575;

x5=-1.76;y5=-2.28;

x6=-1.74;y6=-1.38;

x7=-1.7;y7=-3.64;

x8=-1.46;y8=-4.76;

x9=-1.07;y9=-5.16;

x10=-0.56;y10=-5.16;

x11=-0.3;y11=-4.7;

x12=-0.16;y12=-3.9;

x13=0;y13= -3.9;

x14=1.878;y14=0.246;

line(x0\*sc+xp,-y0\*sc+yp,x14\*sc+xp,-y14\*sc+yp);

line(x0\*sc+xp,-y0\*sc+yp,x1\*sc+xp,-y1\*sc+yp);

line(x1\*sc+xp,-y1\*sc+yp,x2\*sc+xp,-y2\*sc+yp);

line(x2\*sc+xp,-y2\*sc+yp,x3\*sc+xp,-y3\*sc+yp);

line(x3\*sc+xp,-y3\*sc+yp,x4\*sc+xp,-y4\*sc+yp);

line(x4\*sc+xp,-y4\*sc+yp,x5\*sc+xp,-y5\*sc+yp);

line(x5\*sc+xp,-y5\*sc+yp,x6\*sc+xp,-y6\*sc+yp);

line(x5\*sc+xp,-y5\*sc+yp,x7\*sc+xp,-y7\*sc+yp);

line(x7\*sc+xp,-y7\*sc+yp,x8\*sc+xp,-y8\*sc+yp);

line(x8\*sc+xp,-y8\*sc+yp,x9\*sc+xp,-y9\*sc+yp);

line(x9\*sc+xp,-y9\*sc+yp,x10\*sc+xp,-y10\*sc+yp);

line(x10\*sc+xp,-y10\*sc+yp,x11\*sc+xp,-y11\*sc+yp);

line(x11\*sc+xp,-y11\*sc+yp,x12\*sc+xp,-y12\*sc+yp);

line(x12\*sc+xp,-y12\*sc+yp,x13\*sc+xp,-y13\*sc+yp);

line(-x0\*sc+xp,-y0\*sc+yp,-x1\*sc+xp,-y1\*sc+yp);

line(-x1\*sc+xp,-y1\*sc+yp,-x2\*sc+xp,-y2\*sc+yp);

line(-x2\*sc+xp,-y2\*sc+yp,-x3\*sc+xp,-y3\*sc+yp);

line(-x3\*sc+xp,-y3\*sc+yp,-x4\*sc+xp,-y4\*sc+yp);

line(-x4\*sc+xp,-y4\*sc+yp,-x5\*sc+xp,-y5\*sc+yp);

line(-x5\*sc+xp,-y5\*sc+yp,-x6\*sc+xp,-y6\*sc+yp);

line(-x5\*sc+xp,-y5\*sc+yp,-x7\*sc+xp,-y7\*sc+yp);

line(-x7\*sc+xp,-y7\*sc+yp,-x8\*sc+xp,-y8\*sc+yp);

line(-x8\*sc+xp,-y8\*sc+yp,-x9\*sc+xp,-y9\*sc+yp);

line(-x9\*sc+xp,-y9\*sc+yp,-x10\*sc+xp,-y10\*sc+yp);

line(-x10\*sc+xp,-y10\*sc+yp,-x11\*sc+xp,-y11\*sc+yp);

line(-x11\*sc+xp,-y11\*sc+yp,-x12\*sc+xp,-y12\*sc+yp);

line(-x12\*sc+xp,-y12\*sc+yp,-x13\*sc+xp,-y13\*sc+yp);

setfillstyle(SOLID\_FILL,COLOR(255, 208, 133));

floodfill(-0.19290\*sc+xp,0.5524\*sc+yp,COLOR(255, 208, 133));

//mt

x0=-1.74;y0=2.84;

x1=-1.39;y1=2.79;

x2=-0.99;y2=2.68;

x3=-0.816;y3=2.418;

x4=-1.072;y4=2.248;

x5=-1.38;y5=2.335;

x6=-1.47;y6= 2.63;

setcolor(BLACK);

line(x0\*sc+xp,-y0\*sc+yp,x1\*sc+xp,-y1\*sc+yp);

line(x1\*sc+xp,-y1\*sc+yp,x2\*sc+xp,-y2\*sc+yp);

line(x2\*sc+xp,-y2\*sc+yp,x3\*sc+xp,-y3\*sc+yp);

line(x3\*sc+xp,-y3\*sc+yp,x4\*sc+xp,-y4\*sc+yp);

line(x4\*sc+xp,-y4\*sc+yp,x5\*sc+xp,-y5\*sc+yp);

line(x5\*sc+xp,-y5\*sc+yp,x6\*sc+xp,-y6\*sc+yp);

line(x6\*sc+xp,-y6\*sc+yp,x1\*sc+xp,-y1\*sc+yp);

line(-x0\*sc+xp,-y0\*sc+yp,-x1\*sc+xp,-y1\*sc+yp);

line(-x1\*sc+xp,-y1\*sc+yp,-x2\*sc+xp,-y2\*sc+yp);

line(-x2\*sc+xp,-y2\*sc+yp,-x3\*sc+xp,-y3\*sc+yp);

line(-x3\*sc+xp,-y3\*sc+yp,-x4\*sc+xp,-y4\*sc+yp);

line(-x4\*sc+xp,-y4\*sc+yp,-x5\*sc+xp,-y5\*sc+yp);

line(-x5\*sc+xp,-y5\*sc+yp,-x6\*sc+xp,-y6\*sc+yp);

line(-x6\*sc+xp,-y6\*sc+yp,-x1\*sc+xp,-y1\*sc+yp);

//alis

x0=-1.49;y0=3.2928;

x1=-1.4834;y1=3.4294;

x2=-0.93;y2=3.52;

x3=-0.762;y3=3.384;

line(x0\*sc+xp,-y0\*sc+yp,x1\*sc+xp,-y1\*sc+yp);

line(x1\*sc+xp,-y1\*sc+yp,x2\*sc+xp,-y2\*sc+yp);

line(x2\*sc+xp,-y2\*sc+yp,x3\*sc+xp,-y3\*sc+yp);

line(x3\*sc+xp,-y3\*sc+yp,x0\*sc+xp,-y0\*sc+yp);

line(-x0\*sc+xp,-y0\*sc+yp,-x1\*sc+xp,-y1\*sc+yp);

line(-x1\*sc+xp,-y1\*sc+yp,-x2\*sc+xp,-y2\*sc+yp);

line(-x2\*sc+xp,-y2\*sc+yp,-x3\*sc+xp,-y3\*sc+yp);

line(-x3\*sc+xp,-y3\*sc+yp,-x0\*sc+xp,-y0\*sc+yp);

//:)

x0=-0.3;y0=2.05;

x1=-0.1;y1=1.74;

x2=-0.1;y2=1.43;

x3=-0.46;y3=1.5;

x4=-0.49;y4=1.41;

x5=-0.05;y5=1.31;

x6=0.36;y6=1.41;

x7=0.34;y7=1.5;

x8=0;y8=1.43;

x9=0;y9=1.74;

x10=0.2;y10=2.05;

line(x0\*sc+xp,-y0\*sc+yp,x1\*sc+xp,-y1\*sc+yp);

line(x1\*sc+xp,-y1\*sc+yp,x2\*sc+xp,-y2\*sc+yp);

line(x2\*sc+xp,-y2\*sc+yp,x3\*sc+xp,-y3\*sc+yp);

line(x3\*sc+xp,-y3\*sc+yp,x4\*sc+xp,-y4\*sc+yp);

line(x4\*sc+xp,-y4\*sc+yp,x5\*sc+xp,-y5\*sc+yp);

line(x5\*sc+xp,-y5\*sc+yp,x6\*sc+xp,-y6\*sc+yp);

line(x6\*sc+xp,-y6\*sc+yp,x7\*sc+xp,-y7\*sc+yp);

line(x7\*sc+xp,-y7\*sc+yp,x8\*sc+xp,-y8\*sc+yp);

line(x8\*sc+xp,-y8\*sc+yp,x9\*sc+xp,-y9\*sc+yp);

line(x9\*sc+xp,-y9\*sc+yp,x10\*sc+xp,-y10\*sc+yp);

line(x10\*sc+xp,-y10\*sc+yp,x0\*sc+xp,-y0\*sc+yp);

}

};

float sun(float xS, float yS, float sc)

{

setcolor(COLOR(255, 111, 0));

setfillstyle(SOLID\_FILL,COLOR(255, 111, 0));

circle(50+xS,105+yS,30);

floodfill(50+xS,105+yS,COLOR(255, 111, 0));

}

float moon(float xM, float yM, float sc)

{

setcolor(DARKGRAY);

setfillstyle(SOLID\_FILL,DARKGRAY);

circle(50+xM,105+yM,30);

floodfill(50+xM,105+yM,DARKGRAY);

}

float clouds(float xc, float yc, float sc)

{

float x[12], y[12];

int i;

sc=18;

//lini lini

x[0]=-17;y[0]=2;

x[1]=-19;y[1]=3;

x[2]=-18.4;y[2]=5;

x[3]=-17.2;y[3]=6.2;

x[4]=-14.6;y[4]=5.6;

x[5]=-14.4;y[5]=4.6;

x[6]=-13;y[6]=5;

x[7]=-12;y[7]=4;

x[8]=-11;y[8]=4;

x[9]=-10;y[9]=3;

x[10]=-7;y[10]=2;

x[11]=-17;y[11]=2;

setcolor(WHITE);

setfillstyle(SOLID\_FILL,WHITE);

for(i=0;i<11;i++){

line(x[i]\*sc+xc,-y[i]\*sc+yc,x[i+1]\*sc+xc,-y[i+1]\*sc+yc);

}

floodfill(-13\*sc+xc,-3\*sc+yc,WHITE);

setcolor(WHITE);

setfillstyle(SOLID\_FILL,WHITE);

for(i=0; i<11; i++){

line(-x[i]\*sc+xc,-y[i]\*sc+yc,-x[i+1]\*sc+xc,-y[i+1]\*sc+yc);

}

floodfill(13\*sc+xc,-3\*sc+yc,WHITE);

}

float gambar(float xp, float yp, float sc){

float x0,x1,x2,x3,x4, y0,y1,y2,y3,y4;

x0=0;y0=2;

x1=2;y1=2;

x2=2;y2=0;

line(x0\*sc+xp, -y0\*sc+yp, x1\*sc+xp,-y1\*sc+yp);

line(x1\*sc+xp, -y1\*sc+yp, x2\*sc+xp, y2\*sc+yp);

line(x0\*sc+xp, y0\*sc+yp, -x1\*sc+xp, y1\*sc+yp);

line(-x1\*sc+xp, y1\*sc+yp, -x2\*sc+xp, y2\*sc+yp);

x0-2;y0=-2;

x1=2;y1=2;

line(x0\*sc+xp, y1\*sc+yp, x1\*sc+xp, y2\*sc+yp);

line(x0\*sc+xp, -y1\*sc+yp, -x1\*sc+xp, y2\*sc+yp);

//tambah

x0=-2;y0=0;

x1=2;y1=0;

line(x0\*sc+xp, y0\*sc+yp, x1\*sc+xp,y1\*sc+yp);

x0=0;y0=-2;

x1=0;y1=2;

line(x0\*sc+xp, y0\*sc+yp, x1\*sc+xp,y1\*sc+yp);

}

void loadscreen()

{

settextstyle(8,0,1);

outtextxy(180,250, "loading");

circle(180,300,4);

for(int i = 0; i<200; i+=2)

{

setcolor(COLOR(209, 227, 255));

circle(180+i,300,4);

if(i>80)

{

setcolor(WHITE);

outtextxy(180,250,"loading .");

}

if(i>130)

{

setcolor(WHITE);

outtextxy(180,250,"loading . .");

}

if(i>180)

{

setcolor(WHITE);

outtextxy(180,250,"loading . . .");

}

delay(40);

}

cleardevice();

}

void mainmenu(){

while(1){

rectangle(0,0,600,700);

setfillstyle(SOLID\_FILL,COLOR(209, 227, 255));

floodfill(100,100,COLOR(209, 227, 255));

int mouse\_x, mouse\_y;

mouse\_x=mousex();

mouse\_y=mousey();

setcolor(BLACK);

setbkcolor(COLOR(209, 227, 255));

settextstyle(3,0,8);

outtextxy(205,70,"HUNT");

settextstyle(3,0,3);

outtextxy(280,185,"play");

outtextxy(255,285,"how to play");

outtextxy(275,385,"credit");

outtextxy(285,485,"exit");

rectangle(180,170,420,230);

rectangle(180,270,420,330);

rectangle(180,370,420,430);

rectangle(180,470,420,530);

if((mouse\_y>170)&&(mouse\_y<230)&&(mouse\_x>180)&&(mouse\_x<420))

{

if(ismouseclick(WM\_LBUTTONDOWN))

{

cleardevice();

beforeplay();

gameplay();

clearmouseclick(WM\_LBUTTONDOWN);

}

}

if((mouse\_y>270)&&(mouse\_y<330)&&(mouse\_x>180)&&(mouse\_x<420))

{

if(ismouseclick(WM\_LBUTTONDOWN))

{

cleardevice();

howtoplay();

clearmouseclick(WM\_LBUTTONDOWN);

}

}

if((mouse\_y>370)&&(mouse\_y<430)&&(mouse\_x>180)&&(mouse\_x<420))

{

if(ismouseclick(WM\_LBUTTONDOWN))

{

cleardevice();

credits();

clearmouseclick(WM\_LBUTTONDOWN);

}

}

if((mouse\_y>470)&&(mouse\_y<530)&&(mouse\_x>180)&&(mouse\_x<420))

{

if(ismouseclick(WM\_LBUTTONDOWN))

{

closegraph();

exit(0);

clearmouseclick(WM\_LBUTTONDOWN);

break;

}

}

}

}

void howtoplay(){

cleardevice();

while(1){

setcolor(BLACK);

CKapal KPHowtoplay;

//setcolor(WHITE);

settextstyle(3,0,8);

outtextxy(138,50,"how to play");

settextstyle(3,0,3);

outtextxy(50,185,"press right arrow key -> to moving your boat");

outtextxy(50,215,"press left arrow key <- to moving your boat");

outtextxy(50,245,"press space bar to start hunting!");

KPHowtoplay.initKapalKN(400,600,20);

KPHowtoplay.KapalKN1();

setcolor(WHITE);

gambar(295, 470, 1);

gambar(345, 500, 1);

gambar(395, 470, 1);

gambar(445, 500, 1);

gambar(495, 470, 1);

gambar(545, 500, 1);

if(GetAsyncKeyState(27)<0)

{

cleardevice();

clearmouseclick(WM\_LBUTTONDOWN);

return mainmenu();

}

}

}

void credits(){

cleardevice();

sndPlaySound("gameplay.wav",SND\_ASYNC|SND\_LOOP);

readimagefile("logo.jpg",300,400,500,600);

while(1){

setcolor(BLACK);

settextstyle(3,0,8);

outtextxy(185,50,"credits");

settextstyle(3,0,3);

outtextxy(50,185,"Indiana Namaul Husnah");

outtextxy(50,215,"5024201061");

outtextxy(50,245,"Teknik Komputer 2020");

outtextxy(50,275,"Institut Teknologi Sepuluh Nopember");

if(GetAsyncKeyState(27)<0)

{

cleardevice();

clearmouseclick(WM\_LBUTTONDOWN);

return mainmenu();

}

}

}

void beforeplay(){

cleardevice();

while(1){

settextstyle(3,0,3);

outtextxy(130,185,"get as many fish as possible overnight!");

outtextxy(180,215,"press space to start game");

if (GetAsyncKeyState(VK\_SPACE) & 0x8000)

{

cleardevice();

clearmouseclick(WM\_LBUTTONDOWN);

gameplay();

}

if(GetAsyncKeyState(27)<0)

{

cleardevice();

clearmouseclick(WM\_LBUTTONDOWN);

return mainmenu();

}

}

}

void harpoon(float xh, float yh, float sc)

{

float x[5], y[5];

int i;

sc=10;

x[0]=1.4;y[0]= 8.45;

x[1]=1.75;y[1]= 8.15;

x[2]=-9.4;y[2]=-5.85;

x[3]=-9.5;y[3]=-5.75;

x[4]=1.4;y[4]= 8.45;

setcolor(COLOR(135, 123, 85));

setfillstyle(SOLID\_FILL, COLOR(135, 123, 85));

for(i=0;i<4;i++){

line(x[i]\*sc+xh,-y[i]\*sc+yh,x[i+1]\*sc+xh,-y[i+1]\*sc+yh);

}

floodfill(0.15\*sc+xh,-6.5\*sc+yh,COLOR(135, 123, 85));

}

void fishe(float xf, float yf, float sc){

float x[25], y[25];

int i;

sc=20;

x[0]=-2.42;y[0]=-0.045;

x[1]=-2.1494;y[1]=-0.0244;

x[2]=-2.422;y[2]=0.154;

x[3]=-2.162;y[3]=0.32;

x[4]=-1.724;y[4]=0.52;

x[5]=-1.08;y[5]=0.665;

x[6]=-0.202;y[6]=0.674;

x[7]=0.635;y[7]=0.525;

x[8]=1.28;y[8]=0.24;

x[9]=1.54;y[9]=0.24;

x[10]=2.26;y[10]=0.72;

x[11]=1.925;y[11]=0.035;

x[12]=2.29;y[12]=-0.46;

x[13]=1.532;y[13]=-0.184;

x[14]=1.2933;y[14]=-0.18405;

x[15]=1.085;y[15]=-0.265;

x[16]=0.75;y[16]=-0.4;

x[17]=-0.36;y[17]=-0.555;

x[18]=-1.385;y[18]=-0.515;

x[19]=-1.85;y[19]=-0.345;

x[20]=-2.42;y[20]=-0.045;

setcolor(COLOR(201, 194, 109));

setfillstyle(SOLID\_FILL,COLOR(201, 194, 109));

for(i=0;i<20;i++){

line(x[i]\*sc+xf,-y[i]\*sc+yf,x[i+1]\*sc+xf,-y[i+1]\*sc+yf);

}

floodfill(0.2\*sc+xf,0\*sc+yf,COLOR(201, 194, 109));

}

void gameover(){

sndPlaySound("gameova.wav",SND\_ASYNC|SND\_LOOP);

setbkcolor(COLOR(209, 227, 255));

cleardevice();

int page=0;

setactivepage(page);

setvisualpage(page);

while(1){

setcolor(BLACK);

settextstyle(3,0,3);

outtextxy(245,185,"time's up");

settextstyle(3,0,3);

outtextxy(240, 215, score);

settextstyle(3,0,3);

outtextxy(185,245,"press esc to main menu");

setcolor(BLACK);

if(scoreK>=10){

gambar(220, 130, 15);

gambar(290, 130, 15);

gambar(360, 130, 15);

}

if(scoreK>=5){

gambar(220, 130, 15);

gambar(290, 130, 15);

}

if(scoreK>=1){

gambar(220, 130, 15);

}

if(GetAsyncKeyState(27)<0)

{

cleardevice();

return mainmenu();

}

page=1-page;

}

}

void gameplay(){

int page = 0;

setbkcolor(BLACK);

float xp,yp,sc;

float xcl, ycl; xcl = 300; ycl = 220;

float f; f=-1;

float xMo, yMo; xMo=500; yMo=20;

float xMs, yMs; xMs=600; yMs=20;

float xh1, yh1; xh1=160; yh1=300;

float xh, yh; xh=160, yh=300;

float bp=0;

float xff, yff;xff=200;yff=600;

float xff1, yff1;xff1=600;yff1=600;

float flagx; flagx=1;

float flagy; flagy=1;

float flagx1; flagx1=1;

float flagy1; flagy1=1;

CKapal h1, h2, sea1, fs1, fs2, fs3, fs4, KP, KEI;

sea1.initSea(200);

KEI.initKapalKN(160,280,6.5);

KP.initKapalKN(200,330,20);

//init fish

fs1.initfish(600,400,20);

fs2.initfish(666,450,20);

fs3.initfish(566,500,20);

fs4.initfish(599,550,20);;

while(1){

setactivepage(page);

setvisualpage(1-page);

cleardevice();

sea1.sea();

harpoon(xh, yh, sc); harpoon(xh1, yh1, sc);

KEI.KKE(); KEI.GerakKanan();

KP.KapalKN1(); KP.GerakKanan();

fishe(xff, yff, sc); fishe(xff1, yff1, sc);

fs1.fish(); fs1.gerakfishA();

fs2.fish2(); fs2.gerakfishC();

fs3.fish(); fs3.gerakfishB();

fs4.fish2(); fs4.gerakfishC();

//score

setcolor(WHITE);

settextstyle(3,0,3);

outtextxy(10, 10, score); // ambil score dari array score[99]

sprintf(score,"score = %.f", scoreK); //tampilkan score dari scoreK

setcolor(WHITE);

gambar(100, 70, 1); gambar(150, 80, 1); gambar(200, 90, 1);

gambar(350, 50, 1); gambar(400, 70, 1); gambar(450, 80, 1);

clouds(xcl, ycl, sc);

xcl=xcl+f\*0.4; //0.4

if(xcl>310){

f=-1;

}

if(xcl<290){

f=1;

}

moon(xMo, yMo, sc);

xMo=xMo-0.4; //30. 0.4

if(xMo<-100){

sun(450, yMs, sc);

xMs=xMs-1;

setbkcolor(COLOR(38, 74, 255));

}

//shoot

if(GetAsyncKeyState(VK\_SPACE) & 0x8000)

{

PlaySound(TEXT("water.wav"), NULL, SND\_ASYNC);

xh1=xh;

yh1=yh;

bp=1;

}

if(bp==1)

{

yh1=yh1+6;

xh1=xh1-5;

if((xh1>0.42\*20+xff)&&(xh1<6.29\*20+xff)&&(yh1>-0.024\*20+yff)&&(yh1<0.665\*20+yff))

{

xff=700;

yff=700;

scoreK+=1;

}

if((xh1>0.42\*20+xff1)&&(xh1<6.29\*20+xff1)&&(yh1>-0.024\*20+yff1)&&(yh1<0.665\*20+yff1))

{

xff1=900;

yff1=800;

scoreK+=1;

}

}

if( GetAsyncKeyState(VK\_RIGHT) & 0x8000)

{

xh=xh+5;

}

if( GetAsyncKeyState(VK\_LEFT) & 0x8000)

{

xh=xh-5;

}

xff=xff-flagx\*7;

yff=yff-flagy\*0.5;

if(xff<-100){

xff=900;

}

if(yff<400){

flagy=-1;

}

if(yff>800){

flagy=1;

}

xff1=xff1-flagx1\*5;

yff1=yff1+flagy1\*2;

if(xff1<-100){

xff1=800;

}

if(yff1<400){

flagy1=1;

}

if(yff1>800){

flagy1=-1;

}

//gameover

if(xMo<-140){

cleardevice();

setbkcolor(BLACK);

gameover();

if(GetAsyncKeyState(27)<0)

{

cleardevice();

return mainmenu();

}

}

delay(20);

page = 1-page;

}

}

int main(){

initwindow(600,700, "HUNT");

loadscreen();

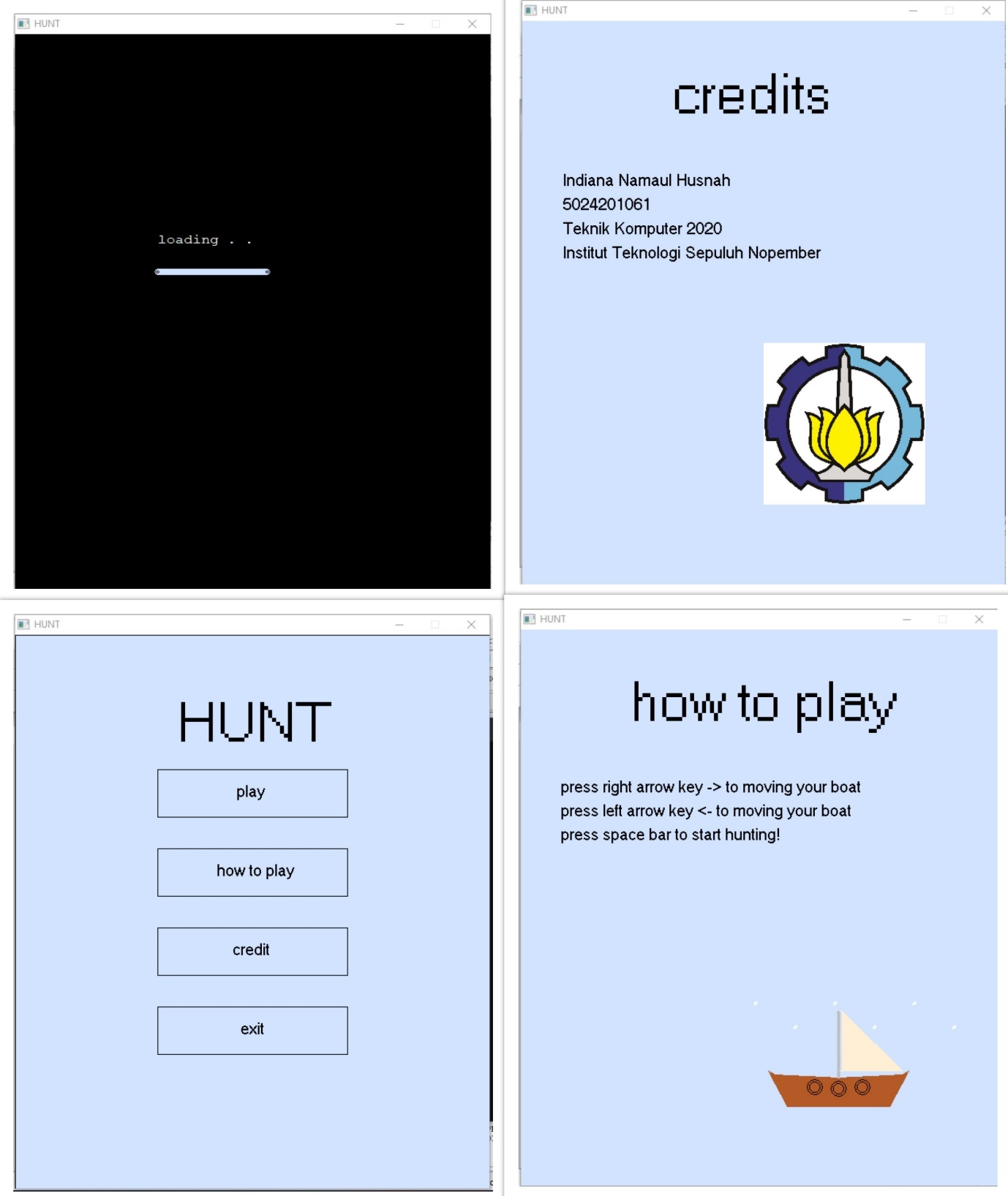
sndPlaySound("gameplay.wav",SND\_ASYNC|SND\_LOOP);

mainmenu();

gameplay();

}

**4.2 Screenshot Program**



**Gambar 4.1.1 Screenshot program**



**Gambar 4.1.1 Screenshot program**

**BAB V**

**PENUTUP**

**5.1 Kesimpulan**

Mengembangkan game dengan bahasa pemograman C/C++ dan dengan menggunakan library graphics.h merupakan suatu kelanjutan dari mempelajari bahasa pemograman. Graphics.h merupakan sebuah awal yang tepat untuk belajar mengembangkan suatu game sebelum menggunakan library yang lebih lanjut. Dengan menggunakan metode pendekatan OOP, pembuatan game dengan bahasa C/C++ menjadi lebih mudah dan tertata. Cara mengembangkan sebuah game dengan bahasa C/C++ dan library graphics.h adalah dengan mempelajari fungsi-fungsi apa saja yang terdapat di graphics.h kemudian mengembangkannya di IDE seperti Codeblocks, Dev-c++, dan lain-lain.

**5.2 Saran**

Pada kenyataannya, pengembangan game ini masih bersifat sangat sederhana. Oleh karena itu, pengembangan game serta laporan ini pun masih memerlukan kritikan dan saran dari semua pihak terkait dengan relevansi laporan final project game ini agar bisa menjadi lebih baik di masa yang akan datang.

**DAFAR PUSTAKA**

Ide, E., Visual, C., C, D. B., & Praktikum, M. (2018). Pengenalan Bahasa C++, Algoritma Pemrograman, Integrated Development Equipment (IDE) Visual C++ dan Dasar Dasar Bahasa C++. *Universitas Negeri Malang*, 1–17.

Suprapto, et al. (2008). Bahasa Pemrograman. *Buku Bahasa Pemrograman*, *1*(1), 1–597. http://staff.uny.ac.id/sites/default/files/penelitian/Drs. Totok Sukardiyono, M.T./Buku Bahasa Pemrograman Lengkap.pdf