

```

1  #include <iostream>
2  #include <windows.h> //menggunakan gotoxy, GetAsyncKeyState,
3  //dan beberapa fungsi lainnya
4  //Awal pendeklarasian beberapa konstanta
5  #define batas_atas 2 //batas_atas game
6  #define batas_bawah 23 //batas_bawah_game
7  #define batas_samping 2 // batas samping kiri game
8  #define batas_samping_k 78 //batas samping kanan game
9  //Akhir pendeklarasian konstanta
10 using namespace std;
11 //pendeklarasian variabel
12 int ix(5), iy(5), x1 (4), y1 (4), xHapus(3), yHapus(3), veloX(1), veloY(0)
13 ,ekorsX[200], ekorsY[200], nEkor = 5, start = 0, makX, makY;
14
15 void gotoxy(int x, int y) //Void untuk gotoxy, memindahkan kursor pada console
16 {
17     COORD pos = {x, y};
18     SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), pos);
19 }
20
21 void naik() //untuk gerak Snake ke atas
22 {
23     veloY = -1; //Kecepatan Y menjadi ke atas
24     veloX = 0; //kecepatan X dinolkan
25 }
26
27 void turun()
28 {
29     veloY = 1; //sda
30     veloX = 0; //sda
31 }
32
33 void samping()
34 {
35     veloX = -1; //sda
36     veloY = 0; //sda
37 }

```

```

38
39 void sampling_k()
40 {
41     veloX = 1; //sda
42     veloY = 0; //sda
43 }
44
45 void hapus()
46 {
47     gotoxy(xHapus, yHapus); //untuk menghapus bagian yang sudah dilewati
48     cout << " ";
49 }
50
51 void ekor()
52 {
53     for(int j = 0; j <= 2; j++)
54     {
55         gotoxy(ekorsX[j], ekorsY[j]);
56         cout << "+";
57     }
58 }
59
60 void tampil()
61 {
62     gotoxy(iX, iY); //iX dan iY adalah posisi gambar kepala
63     cout << char(2); //untuk menggambar gambar kepala
64     //karena ascii dari smiley tsb adalh 2
65     gotoxy(makX, makY);
66     //nilai makX dan makY adalah random
67     //digunakan untuk posisi makanan, jadi posisi makanan random
68     cout << "*"; //gambar makanan
69 }
70

```

```

71 void ganti_posisi() //mengganti posisi dari ekor ekor snake
72 {
73     xHapus = ekorsX[nEkor - 1];
74     yHapus = ekorsY[nEkor - 1];
75     for(int j = nEkor - 1; j >= 1; j--)
76     {
77         ekorsX[j] = ekorsX[j-1];
78         ekorsY[j] = ekorsY[j-1];
79     }
80     ekorsX[0] = iX;
81     ekorsY[0] = iY;
82 }
83
84 void velo() //fungsi untuk melakukan update posisi snake sesuai tombol
85 //yang ditekan
86 {
87     ganti_posisi();
88     hapus();
89     iX += veloX;
90     iY += veloY;
91     tampil();
92     ekor();
93 }
94
95 void trace()
96 {
97     for(int j = 0; j <= nEkor -1; j++)
98     {
99         cout << "x" << j << " y" << j << " = " << ekorsX[j] << "," << ekorsY[j];
100        cout << " // " << iX << "," << iY << endl;
101    }
102    system("pause");
103    system("cls");
104 }
105

```

```

106 bool isDestroy()
107 {
108     //selain untuk mengecek apakah snake sudah kalah
109     //fungsi ini juga menghapus kepala snake yang tersisa jika membentur dinding
110     if(iX == batas_samping_k) { iX = 3; gotoxy(78, iY); cout << " "; }
111     if(iX == batas_samping) { iX = 77; gotoxy(2, iY); cout << " "; }
112     if(iY == batas_atas) { iY = 22; gotoxy(iX, 2); cout << " "; }
113     if(iY == batas_bawah) { iY = 3; gotoxy(iX, 23); cout << " "; }
114     //trace();
115     for(int j = 0; j <= nEkor - 1; j++)
116         if(ekorsX[j] == iX && ekorsY[j] == iY) return true;
117     return false;
118 }
119
120 char getKey()//fungsi untuk mendapatkan tombol yang ditekan
121 {
122     for(int i = 8; i <= 222; i++)
123     {
124         if(GetAsyncKeyState(i) == -32767)
125         {
126             switch(i)
127             {
128                 case 38 : if(veloY != 1)naik();
129                 break; //jika tombol yang ditekan adalah atas
130                 case 40 : if(veloY != -1)turun();
131                 break; //jika tombol yang ditekan adalah bawah
132                 case 37 : if(veloX != 1)samping();
133                 break; //jika tombol yang ditekan adalah kiri
134                 case 39 : if(veloX != -1)samping_k();
135                 break; //jika tombol yang ditekan adalah kanan
136             }
137         }
138     }
139 }

```

```

140
141 void random_makanan()
142 {
143     makX = rand()%(batas_samping_k - 1);
144     if(makX < 4) makX += 3 + (4-makX); //mencegah agar makanan tidak diluar batas
145     makY = rand()%(batas_bawah - 1);
146     if(makY < 4) makY += 3 + (4-makY);
147     gotoxy(makX, makY);
148     cout << "*";
149 }
150
151 bool isEaten()
152 {
153     if(iX == makX && iY == makY) return true; else return false;
154 }
155
156 void cBorder() //Fungsi untuk membuat garis tepi game
157 {
158     for(int i = 1; i <= 78; i++)
159     {
160         for(int j = 1; j <= 24; j += 23)
161         {
162             if(j > 1 || i >= 32)
163             {
164                 gotoxy(i, j);cout << char(219);
165             }
166         }
167     }
168     for(int i = 1; i <= 24; i++)
169     {
170         for(int k = 1; k <= 80; k += 78)
171         {
172             gotoxy(k, i); cout << char(219);
173         }
174     }
175 }

```

```

176 //akhir pembuatan garis tepi
177 //Penulisan Skor
178 void skor()
179 {
180     gotoxy(3,1); cout << "Skor : ";
181     gotoxy(18,1); cout << "Panjang : ";
182 }
183 //Untuk menulis skor terbaru dan panjang dari snake
184 void tulis_skor()
185 {
186     gotoxy(11,1); cout << (nEkor - 5) * 10;
187     gotoxy(28,1); cout << nEkor;
188 }
189 //Fungsi yang menjalankan beberapa fungsi yang berjalan
190 //pada awal program, hanya sekali
191 void inisialisasi()
192 {
193     cBorder(); //buat pinggiran game
194     random_makanan(); //letakkan makanan secara random
195     skor(); //tulis tulisan skor di tepi atas
196     tulis_skor(); //menulis skor
197 }
198
199 int main() //fungsi Utama
200 {
201     system("cls"); //membersihkan layar
202     inisialisasi(); //menjalankan fungsi inisialisasi
203     while(!isDestroy()) //selama snake belum rusak atau kalah
204     {
205         velo(); //merubah posisi snake berdasarkan kecepatan X atau Y
206         getkey(); //untuk mendapatkan tombol apa yang ditekan user dan
207         //menjalankan beberapa fungsi yang diperlukan
208         if(isEaten()) //mencek apakah makanan telah dilalap oleh snake
209         {
210             nEkor += 2; //panjang ekor ditambah 2
211             random_makanan(); //makanan diletakkan lagi
212             tulis_skor(); //skor di update
213         }
214         Sleep(40 - (nEkor / 10)); //delay yang semakin cepat dengan penambahan ekor
215     }
216     system("cls"); //layar dibersihkan
217     gotoxy(32,12); cout << "Skor : " << (nEkor - 5) * 10; //ditampilkan skor
218     gotoxy(25,13);
219     system("pause"); //selesai
220 }

```