

Nama : Hanif Muflih Hidayat

NIM : 16523074

Fakultas : STEI-R

```
1  #include <math.h>
2  #include <iostream>
3  #include <fstream>
4  using namespace std;
5
6  // Global variabel
7  int gerak_ke = 0;
8  int mtr = 1;
9  float posisi[1][2] = {{0,0}};
```

Inisialisasi variable global

```
// print lokasi saat ini
void lokasi(){
    float x = (floor(posisi[gerak_ke][0]*100 + 0.5))/100;
    float y = (floor(posisi[gerak_ke][1]*100 + 0.5))/100;
    posisi[gerak_ke][0] = x;
    posisi[gerak_ke][1] = y;
    cout << "lokasi => (";
    printf("%.2f", x) ;
    cout << " , " ;
    printf("%.2f", y);
    cout << ")" << endl;
}
```

```
x: 5.56677
y: 5.67654
lokasi => (5.57 , 5.68)
```

Untuk membulatkan posisi 2 angka dibelakang koma sekaligus mengoutput posisi sekarang

```

20 // bergerak sesuai horizontal dan vertikal
21 void gerak(float x,float y){
22     if (gerak_ke < mtr-1){
23         mtr = gerak_ke + 1;
24     }
25     gerak_ke += 1;
26     mtr += 1;
27     posisi[gerak_ke][0] = posisi[gerak_ke-1][0] + x;
28     posisi[gerak_ke][1] = posisi[gerak_ke-1][1] + y;
29     lokasi();
30 }

```

```

lokasi => (0.00 , 0.00)
Masukkan input kode:
1
gerak 1
x: 4
y: 5
lokasi => (4.00 , 5.00)
Masukkan input kode:
1
gerak 1
x: 5
y: 6
lokasi => (9.00 , 11.00)

```

Fitur 1: gerak

Untuk menggerakkan drone secara horizontal dan vertikal sesuai input pengguna dan mengoutput posisi setelah melakukan gerak

```

32 // bergerak sesuai dengan kecepatan, waktu, dan sudut(derajat)
33 void gerak_2(float v, float t, float theta){
34     if (gerak_ke < mtr-1){
35         mtr = gerak_ke + 1;
36     }
37     gerak_ke += 1;
38     mtr += 1;
39     posisi[gerak_ke][0] = posisi[gerak_ke-1][0] + v*t*cos(theta*M_PI/180);
40     posisi[gerak_ke][1] = posisi[gerak_ke-1][1] + v*t*sin(theta*M_PI/180);
41     lokasi();
42 }
43

```

```

lokasi => (0.00 , 0.00)
Masukkan input kode:
2
gerak 2
v: 20
t: 5
theta: 180
lokasi => (-100.00 , 0.00)
Masukkan input kode:
2
gerak 2
v: 10
t: 10
theta: 30
lokasi => (-13.40 , 50.00)

```

Fitur 2: gerak 2

Untuk menggerakkan drone dengan kecepatan waktu dan sudut dalam derajat kemudian mengoutput posisi setelah drone bergerak

```

44 // undo satu perintah kebelakang
45 void undo(){
46     if (gerak_ke > 0){
47         gerak_ke -= 1;
48         cout << "Undo berhasil" << endl;
49         lokasi();
50     } else {
51         cout << "Tidak dapat undo" << endl;
52     }
53 }

```

```

lokasi => (0.00 , 0.00)
Masukkan input kode:
1
gerak 1
x: 4
y: 5
lokasi => (4.00 , 5.00)
Masukkan input kode:
2
gerak 2
v: 3
t: 4
theta: 7
lokasi => (15.91 , 6.46)
Masukkan input kode:
3
Undo berhasil
lokasi => (4.00 , 5.00)
Masukkan input kode:
3
Undo berhasil
lokasi => (0.00 , 0.00)

```

Fitur 3: Undo

Untuk memberi perintah pada drone untuk kembali ke posisi satu perintah sebelumnya dan mengoutput posisi setelah undo

```

55 //redo satu perintah kedepan
56 void redo(){
57     if (gerak_ke < mtr-1){
58         gerak_ke += 1;
59         cout << "Redo berhasil" << endl;
60         lokasi();
61     } else {
62         cout << "Tidak dapat redo" << endl;
63     }
64 }

```

```

lokasi => (15.91 , 6.46)
Masukkan input kode:
3
Undo berhasil
lokasi => (4.00 , 5.00)
Masukkan input kode:
3
Undo berhasil
lokasi => (0.00 , 0.00)
Masukkan input kode:
4
Redo berhasil
lokasi => (4.00 , 5.00)
Masukkan input kode:
4
Redo berhasil
lokasi => (15.91 , 6.46)

```

Fitur 4: Redo

Untuk mengulang perintah yang telah di-undo dan mengoutput lokasi setelah redo

```

66 //save posisi sekarang pada txt
67 void save(){
68     ofstream myfile("save.txt");
69     myfile << posisi[gerak_ke][0] << " :Komponen-X" <<endl;
70     myfile << posisi[gerak_ke][1] << " :Komponen-Y" <<endl;
71     cout << "Save berhasil"<< endl;
72     myfile.close();
73 }
74

```

```

lokasi => (15.91 , 6.46)
Masukkan input kode:
5
Save berhasil

```

```

15.91 :Komponen-X
6.46 :Komponen-Y

```

Fitur 5: Save

Menyimpan posisi sekarang dalam file save.txt

```

75 //load posisi menjadi posisi awal dengan txt
76 void load(){
77     string teks;
78     int ind = 0;
79     gerak_ke = 0;
80     mtr = 1;
81     ifstream myfile("save.txt");
82     while (getline (myfile, teks)){
83         posisi[gerak_ke][ind] = stof(teks);
84         ind += 1;
85     }
86     myfile.close();
87     cout << "Load berhasil" << endl;
88     lokasi();
89 }

```

```

lokasi => (0.00 , 0.00)
Masukkan input kode:
6
Load berhasil
lokasi => (15.91 , 6.46)

```

Fitur 6: Load

Membuat drone berada pada koordinat yang di-save

```
91 //print kode perintah yang disediakan
92 void bantuan(){
93     cout << "ketik -0- untuk menghentikan program"<< endl;
94     cout << "ketik -1- untuk bergerak dengan jarak"<< endl;
95     cout << "ketik -2- untuk bergerak dengan kecepatan"<< endl;
96     cout << "ketik -3- untuk undo"<< endl;
97     cout << "ketik -4- untuk redo"<< endl;
98     cout << "ketik -5- untuk save"<< endl;
99     cout << "ketik -6- untuk load"<< endl;
100    cout << "ketik -7- untuk bantuan"<< endl;
101 }
```

```
ketik -0- untuk menghentikan program
ketik -1- untuk bergerak dengan jarak
ketik -2- untuk bergerak dengan kecepatan
ketik -3- untuk undo
ketik -4- untuk redo
ketik -5- untuk save
ketik -6- untuk load
ketik -7- untuk bantuan
```

Fitur 7: Bantuan

Mengoutput input yang diterima oleh program kepada pengguna

```

103 //kode utama
104 int main(){
105     bool jalan = true;
106     int input_user;
107     float x; float y; float z;
108
109     cout << "Selamat datang pada program simulasi drone" << endl;
110     bantuan();
111     lokasi();
112
113     while (jalan){
114         cout << "Masukkan input kode:" << endl;
115         cin >> input_user;
116         switch (input_user){
117             case 0:
118                 jalan = false;
119                 cout << "Program diberhentikan";
120                 exit(0);
121                 break;
122             case 1:
123                 cout << "gerak 1" << endl;
124                 cout << "x: ";
125                 cin >> x;
126                 cout << "y: ";
127                 cin >> y;
128                 gerak(x,y);
129                 break;
130             case 2:
131                 cout << "gerak 2" << endl;
132                 cout << "v: ";
133                 cin >> x;
134                 cout << "t: ";
135                 cin >> y;
136                 cout << "theta: ";
137                 cin >> z;
138                 gerak_2(x, y, z);
139                 break;

```



```
139         break;
140     case 3:
141         undo();
142         break;
143     case 4:
144         redo();
145         break;
146     case 5:
147         save();
148         break;
149     case 6:
150         load();
151         break;
152     case 7:
153         bantuan();
154         break;
155     default:
156         cout << "Input anda salah!, silakan coba lagi" << endl;
157         cout << "ketik -7- untuk bantuan"<< endl;
158         break;
159     }
160 }
161 }
162
```

```
lokasi => (0.00 , 0.00)
Masukkan input kode:
1
gerak 1
x: 4
y: 5
lokasi => (4.00 , 5.00)
Masukkan input kode:
2
gerak 2
v: 10
t: 10
theta: 30
lokasi => (90.60 , 55.00)
Masukkan input kode:
3
Undo berhasil
lokasi => (4.00 , 5.00)
Masukkan input kode:
4
Redo berhasil
lokasi => (90.60 , 55.00)
Masukkan input kode:
5
Save berhasil
Masukkan input kode:
1
gerak 1
x: 0
y: 0
lokasi => (90.60 , 55.00)
Masukkan input kode:
0
Program diberhentikan
```

```
lokasi => (0.00 , 0.00)
Masukkan input kode:
6
Load berhasil
lokasi => (90.60 , 55.00)
Masukkan input kode:
7
ketik -0- untuk menghentikan program
ketik -1- untuk bergerak dengan jarak
ketik -2- untuk bergerak dengan kecepatan
ketik -3- untuk undo
ketik -4- untuk redo
ketik -5- untuk save
ketik -6- untuk load
ketik -7- untuk bantuan
Masukkan input kode:
0
Program diberhentikan
```

Kode utama program

Berfungsi untuk menerima input user dan menjalankan perintah yang sesuai