

Dokumentasi Fitur Simulasi Drone

1. lokasi()

Fitur lokasi mengecek lokasi drone saat ini.

```
void round2(double num)
{
    double rnum = num * 100;
    if (round(rnum) == floor(rnum))
    {
        cout << fixed << setprecision(2) << trunc(rnum) / 100;
    }
    else if (round(rnum) == ceil(rnum))
    {
        cout << fixed << setprecision(2) << trunc(rnum) / 100 + 0.01;
    }
}

void lokasi()
{
    cout << "(";
    round2(drone_x);      // print koordinat x drone
    cout << ", ";
    round2(drone_y);      // print koordinat y drone
    cout << ")" << endl;
}
```

Parameter fungsi round2 bertipe double untuk menyesuaikan dengan fungsi round.

2. gerak(x, y)

Fitur gerak menggerakkan drone dengan gerak sederhana, yaitu gerak di arah horizontal (x) dan vertikal (y). Pergerakan ditambahkan dengan koordinat awal untuk mendapatkan koordinat baru.

```
void gerak1(double x, double y)
{
    drone_x += x;
    drone_y += y;
}
```

3. gerak_2(v, t, theta)

Fitur gerak_2 menggerakkan drone dengan kecepatan tertentu selama beberapa detik. Drone bergerak dengan sudut tertentu diukur dari garis horizontal berlawanan arah jarum jam.

```

void gerak2(double vel, double time, double angle)
{
    double pi = 3.14159265;
    drone_x += vel * time * cos((angle / 180) * pi);
    drone_y += vel * time * sin((angle / 180) * pi);
}

```

4. undo dan redo

Fitur undo membalikkan koordinat drone ke koordinat sebelumnya. Koordinat ini disimpan dalam vector (satu vector untuk koordinat x, satu untuk y) dan koordinat yang diundo disimpan dalam vector redo.

Fitur redo mengembalikan koordinat drone yang diundo. Koordinat ini diambil dari vector redo. Ketika setelah undo drone digerakkan, isi vector redo dikosongkan sehingga tidak ada lagi koordinat yang bisa diredo.

```

else if (inp == "undo")
{
    // size == n maka indeks terakhir == n - 1
    // push back koordinat terakhir ke redo, lalu pop back
    if (vec_x.size() > 1)
    {
        redo_x.push_back(vec_x[vec_x.size() - 1]);
        redo_y.push_back(vec_y[vec_y.size() - 1]);
        vec_x.pop_back();
        vec_y.pop_back();
        drone_x = vec_x[vec_x.size() - 1];
        drone_y = vec_y[vec_y.size() - 1];
        cout << "Undo berhasil." << endl;
        cout << "Lokasi sekarang: ";
        lokasi();
    }
    else
    {
        cout << "Undo tidak bisa dijalankan." << endl;
    }
}

```

```

else if (inp == "redo")
{
    if (redo_x.size() > 0)
    {
        vec_x.push_back(redo_x[redo_x.size() - 1]);
        vec_y.push_back(redo_y[redo_y.size() - 1]);
        drone_x = redo_x[redo_x.size() - 1];
        drone_y = redo_y[redo_y.size() - 1];
        cout << "Redo berhasil." << endl;
        cout << "Lokasi sekarang: ";
        lokasi();
        redo_x.pop_back();
        redo_y.pop_back();
    }
    else
    {
        cout << "Redo tidak bisa dijalankan." << endl;
    }
}

```

5. exit

Ketika exit dijalankan, program keluar dari loop.

```

while (quit == false)
else if (inp == "exit")
{
    quit = true;
}

```