

DOKUMENTASI TEST CASE MODUL 2

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
=====
Lokasi Drone (0.00 , 0.00)
=====
1. Simulasi Gerak Drone dengan Perpindahan
2. Simulasi Gerak Drone dengan Kecepatan dan Waktu
3. Undo
4. Redo
5. exit

Masukkan angka (1-5) : █
```

Pada program ini disediakan beberapa opsi , dengan tampilan **Lokasi drone** pada baris atas.

Di bawah terdapat 5 opsi, yaitu :

1. **Simulasi Gerak Drone dengan Perpindahan {gerak()}**
2. **Simulasi Gerak Drone dengan kecepatan dan waktu {gerak_2()}**
3. **Undo {undo()}**
4. **Redo {redo()}**
5. **Exit (Terminasi program)**

Pada bagian ini disebut **Interface {interface()}** dimana user harus menginput angka dari 1-5 , input lain akan dianggap invalid dan user harus menginput angkanya lagi.

```
PROBLEMS OUTPUT DEBUG CONSOLE
Masukkan angka valid█
```

Berikut merupakan Tampilan **Simulasi Gerak Drone dengan Perpindahan** saat user menekan angka '1' :

gerak()

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Gerak drone pada sumbu-x : 3
Gerak drone pada sumbu-y : 4█
```

Program akan meminta input x dan y pada user dan lokasi drone akan ter-update

```
=====
Lokasi Drone (3.00 , 4.00)
=====
1. Simulasi Gerak Drone dengan Perpindahan
2. Simulasi Gerak Drone dengan Kecepatan dan Waktu
3. Undo
4. Redo
5. exit

Masukkan angka (1-5) : █
```

Selanjutnya merupakan tampilan pada **Simulasi Gerak Drone dengan Kecepatan dan Waktu** saat user menekan angka '2' :

gerak_2()

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Kecepatan drone bergerak : 10
Waktu tempuh drone : 10
Sudut : 30
```

Program akan meminta user untuk menginput '**v**', '**t**', '**tetha**', dan lokasi drone ter-update

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

=====
Lokasi Drone (89.62 , 53.98)
=====
1. Simulasi Gerak Drone dengan Perpindahan
2. Simulasi Gerak Drone dengan Kecepatan dan Waktu
3. Undo
4. Redo
5. exit

Masukkan angka (1-5) :
```

HISTORY

Selanjutnya pada fitur **History** terdapat **undo()** dan **redo()**, pada fungsi ini, program dapat meng-undo posisi drone sampai posisi awal, dan dapat me-redo sampai posisi terakhir drone.

Undo()

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

=====
Lokasi Drone (89.62 , 53.98)
=====
1. Simulasi Gerak Drone dengan Perpindahan
2. Simulasi Gerak Drone dengan Kecepatan dan Waktu
3. Undo
4. Redo
5. exit

Masukkan angka (1-5) : 3
```

```
Undo berhasil !
```

```
=====
Lokasi Drone (3.00 , 4.00)
=====
1. Simulasi Gerak Drone dengan Perpindahan
2. Simulasi Gerak Drone dengan Kecepatan dan Waktu
3. Undo
4. Redo
5. exit

Masukkan angka (1-5) : █
```

Redo()

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
=====
Lokasi Drone (3.00 , 4.00)
=====
1. Simulasi Gerak Drone dengan Perpindahan
2. Simulasi Gerak Drone dengan Kecepatan dan Waktu
3. Undo
4. Redo
5. exit

Masukkan angka (1-5) : 4█
```

Redo berhasil ! █

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
=====
Lokasi Drone (89.62 , 53.98)
=====
1. Simulasi Gerak Drone dengan Perpindahan
2. Simulasi Gerak Drone dengan Kecepatan dan Waktu
3. Undo
4. Redo
5. exit

Masukkan angka (1-5) : █
```

Pada saat, drone Kembali ke posisi paling awal -> (0 , 0) undo tidak dapat dilakukan:
undo()

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

=====
Lokasi Drone (0.00 , 0.00)
=====
1. Simulasi Gerak Drone dengan Perpindahan
2. Simulasi Gerak Drone dengan Kecepatan dan Waktu
3. Undo
4. Redo
5. exit

Masukkan angka (1-5) : 3
```

Tidak dapat Undo!

Hal yang sama juga berlaku pada redo, jika posisi drone sudah menyentuh posisi terakhir, atau setelah melakukan **gerak()** / **gerak_2()** :

redo()

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

=====
Lokasi Drone (89.62 , 53.98)
=====
1. Simulasi Gerak Drone dengan Perpindahan
2. Simulasi Gerak Drone dengan Kecepatan dan Waktu
3. Undo
4. Redo
5. exit

Masukkan angka (1-5) : 4
```

Tidak dapat Redo!