

**LAPORAN PRATIKUM
TUGAS ALPRO MINGGU KE-10**



Disusun Oleh :

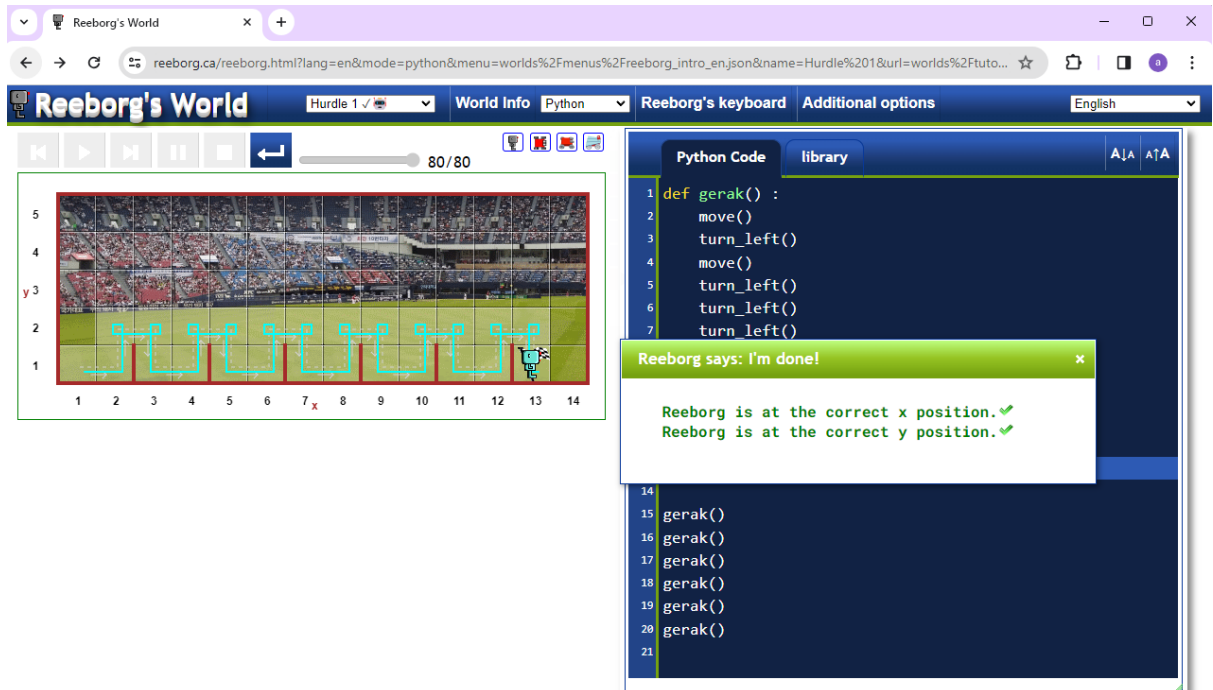
1. Achmad Diky Setiawan (23091397178)
2. Tiara Dwi Aninda Putri (23091397197)
3. Achmad Harris Abdi (23091397204)

**Program Studi D4 Manajemen Informatika
Program Vokasi
Universitas Negeri Surabaya
2023**

Tugas Minggu ke 10

- Program Hurdle 1

Source Code :

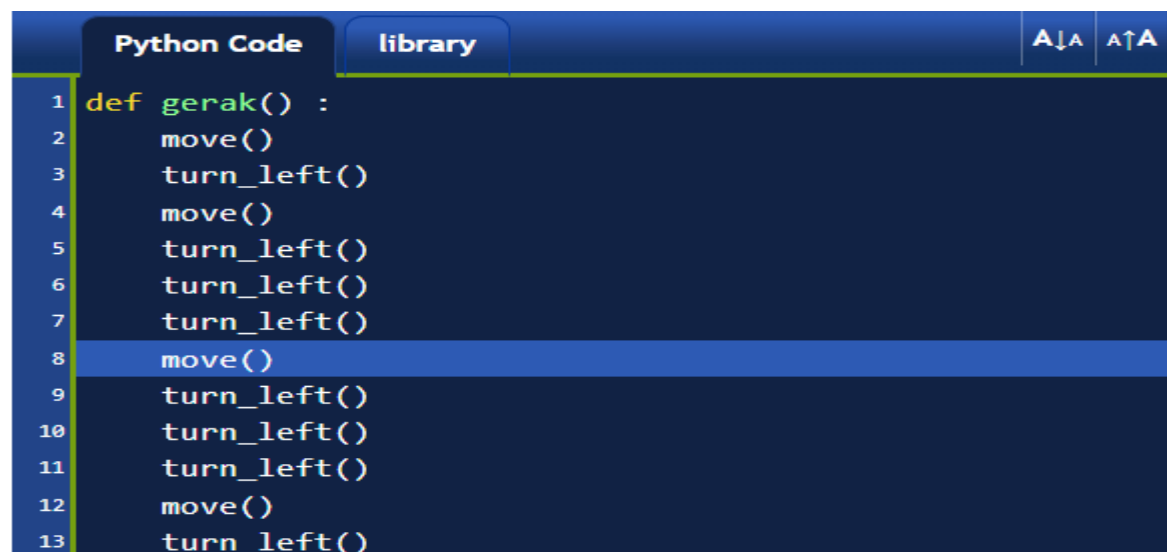


Penjelasan Step By Step :

1. “Def” Digunakan untuk menyimpan suatu proses agar bisa di gunakan terus-menerus,



2. Berikan suatu perintah didalam “def” agar bisa gunakan secara terus menerus, “move()” digunakan untuk maju kedepan, sedangkan turn_left() digunakan agar robot menoleh kekiri



3. lalu panggil “def” agar bisa digunakan secara terus menerus hingga mencapai finish

```
14
15 gerak()
16 gerak()
17 gerak()
18 gerak()
19 gerak()
20 gerak()
```

Output :

The screenshot displays the Reeborg's World Python IDE interface. At the top, the title bar reads "Reeborg's World" with tabs for "Hurdle 1", "World Info", "Python", "Reeborg's keyboard", and "Additional options". Below the title bar is a toolbar with navigation buttons and a progress indicator showing "80/80". The main window is divided into two panes. The left pane shows a 14x5 grid representing the arena, with a blue robot at the bottom right (x=13, y=1) and a series of blue hurdles arranged in a path. The right pane is titled "Python Code" and contains the following code:

```
1 def gerak() :
2     move()
3     turn_left()
4     move()
5     turn_left()
6     turn_left()
7     turn_left()
```

Below the code editor, a green notification box displays the message "Reeborg says: I'm done!". Below this, a white box shows two green checkmarks with the text "Reeborg is at the correct x position." and "Reeborg is at the correct y position.". At the bottom of the right pane, the executed code is shown:

```
14
15 gerak()
16 gerak()
17 gerak()
18 gerak()
19 gerak()
20 gerak()
```

- **Program Hurdle 2**
Source Code :

```
Python Code library A⬇A A⬆A
1 def belok_kanan():
2     turn_left()
3     turn_left()
4     turn_left()
5
6 def lompat():
7     move()
8     turn_left()
9     move()
10    belok_kanan()
11    move()
12    belok_kanan()
13    move()
14    turn_left()
15
16 while not at_goal():
17     lompat()
```

Penjelasan Step By Step :

1. “**def belok_kanan():**” digunakan untuk menyimpan suatu perintah yang berada didalamnya, dan “**turn_left()**” digunakan untuk memerintah robot karel hadap ke kiri

```
Python Code library A⬇A A⬆A
1 def belok_kanan():
2     turn_left()
3     turn_left()
4     turn_left()
```

2. “**def lompat():**” digunakan untuk menyimpan suatu perintah yang berada didalamnya, “**move()**” digunakan untuk memerintah robot agar berjalan, “**belok_kanan()**” digunakan untuk memanggil function yang sudah didefinisikan sebelumnya

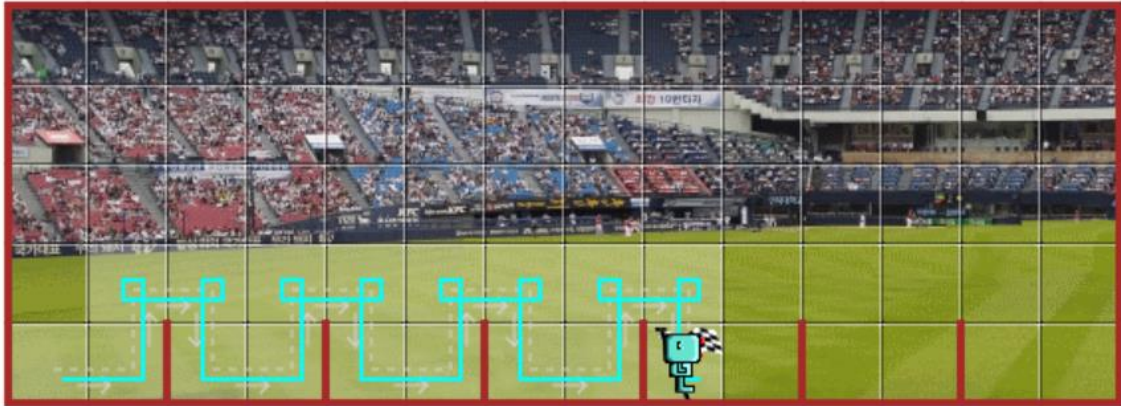
```
6 def lompat():
7     move()
8     turn_left()
9     move()
10    belok_kanan()
11    move()
12    belok_kanan()
13    move()
14    turn_left()
```

3. “**while not at_goal():**” digunakan untuk memulai perulangan jika kondisi fungsinya masih dalam keadaan false yang menandakan bahwa tujuan belum tercapai, dan

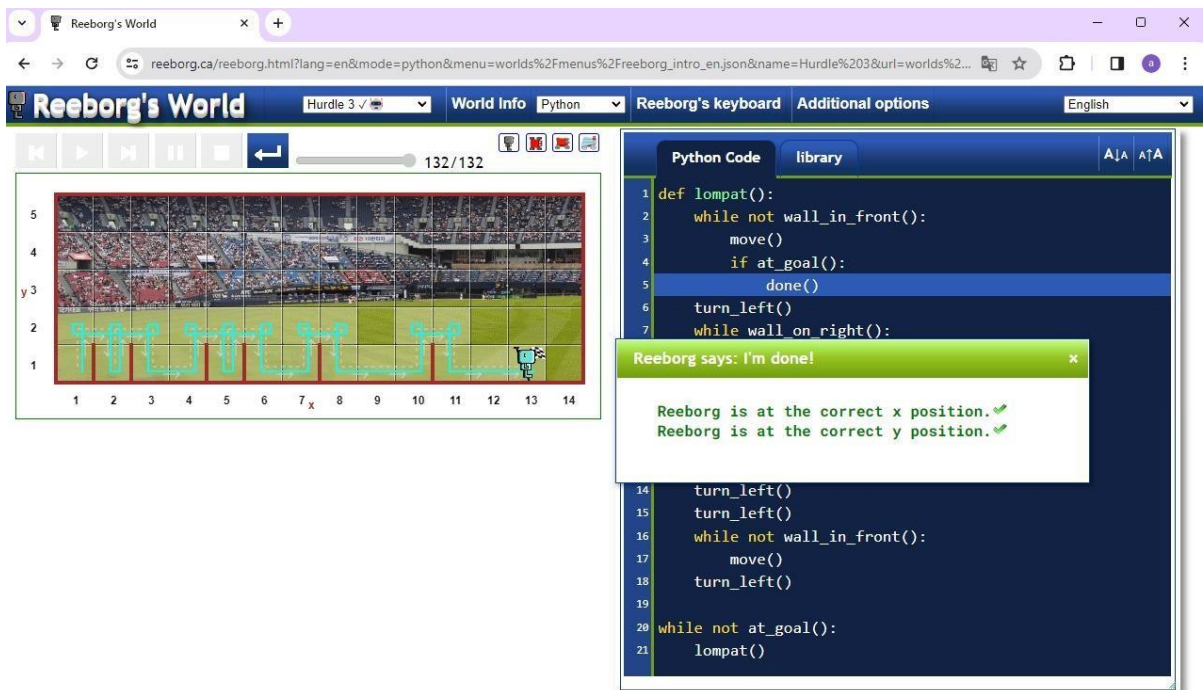
“**lompat()**” digunakan untuk memanggil function yang sebelumnya sudah terdefiniskan

```
16 while not at_goal():  
17     lompat()
```

Output :



- **Program Hurdle 3**
Source Code :

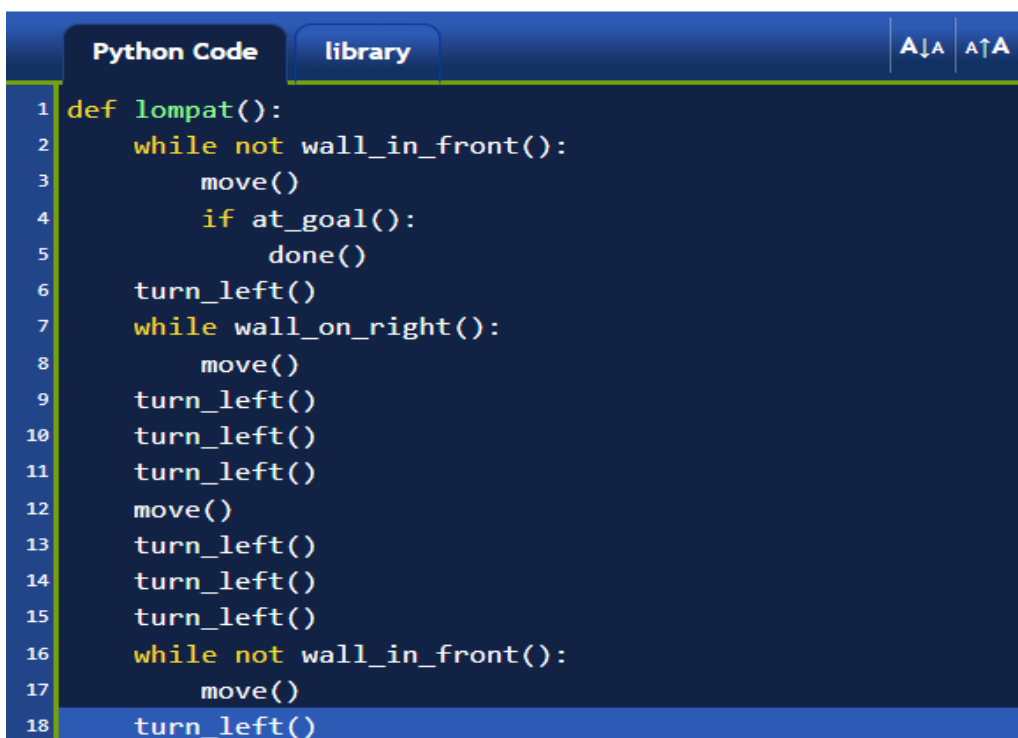


Penjelasan Step By Step :

1. “def” Digunakan untuk menyimpan suatu proses agar bisa digunakan secara terus menerus dan digunakan untuk mengelompokkan suatu perintah,



2. berikan perintah di dalam “def” agar bisa dipanggil secara terus menerus.



3. “While” artinya “Ketika” jadi ketika tidak ada wall didepan robot “while not wall_in_front”, maka robot akan melakukan sesuai perintah yang tertera,

```
2   while not wall_in_front():
3       move()

16  while not wall_in_front():
17      move()
18  turn_left()
```

4. Begitupun sebaliknya, ketika ada wall di samping robot “while wall_on_right”, robot akan melakukan perintah yang tertera juga.

```
7   while wall_on_right():
8       move()
9       turn_left()
10      turn_left()
11      turn_left()
12      move()
13      turn_left()
14      turn_left()
15      turn_left()
```

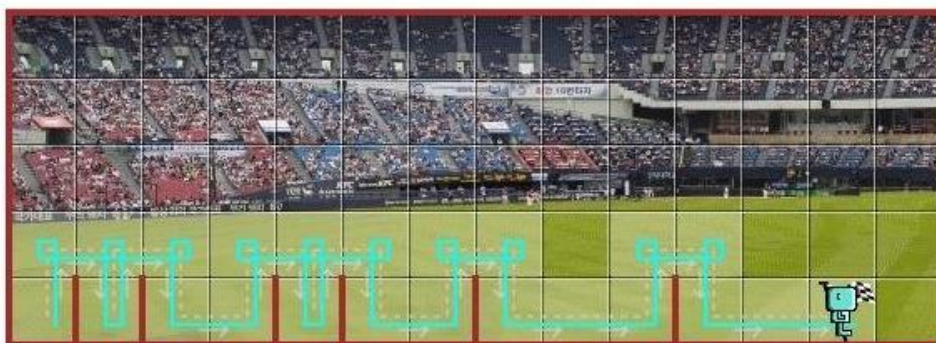
5. “if” artinya “Jika” jadi jika robot telah mencapai finish maka tugas robot selesai “done”

```
4   if at_goal():
5       done()
```

6. Dan sebaliknya jika robot belum mencapai finish robot akan melanjutkan perintah yang sudah tertera.

```
20  while not at_goal():
21      lompat()
```

Output :



- **Program Hurdle 4**
Source Code :

```

Python Code library A↓A A↑A
1 def lompat():
2     while not wall_in_front():
3         move()
4         if at_goal():
5             done()
6     turn_left()
7     while wall_on_right():
8         move()
9     turn_left()
10    turn_left()
11    turn_left()
12    move()
13    turn_left()
14    turn_left()
15    turn_left()
16    while not wall_in_front():
17        move()
18        turn_left()
19
20 while not at_goal():
21     lompat()

```

Penjelasan Step By Step :

1. “def” Digunakan untuk menyimpan suatu proses agar bisa digunakan secara terus menerus dan digunakan untuk mengelompokkan suatu perintah,

```

Python Code library A↓A A↑A
1 def lompat():

```

2. berikan perintah di dalam “def” agar bisa dipanggil secara terus menerus

```

Python Code library A↓A A↑A
1 def lompat():
2     while not wall_in_front():
3         move()
4         if at_goal():
5             done()
6     turn_left()
7     while wall_on_right():
8         move()
9     turn_left()
10    turn_left()
11    turn_left()
12    move()
13    turn_left()
14    turn_left()
15    turn_left()
16    while not wall_in_front():
17        move()
18    turn_left()

```


3. “While” artinya “Ketika” jadi ketika tidak ada wall didepan robot “while not wall_in_front”, maka robot akan melakukan sesuai perintah yang tertera,

```
2   while not wall_in_front():
3       move()

16  while not wall_in_front():
17      move()
18  turn_left()
```

4. Begitupun sebaliknya, ketika ada wall di samping robot “while wall_on_right”, robot akan melakukan perintah yang tertera juga.

```
7   while wall_on_right():
8       move()
9       turn_left()
10      turn_left()
11      turn_left()
12      move()
13      turn_left()
14      turn_left()
15      turn_left()
```

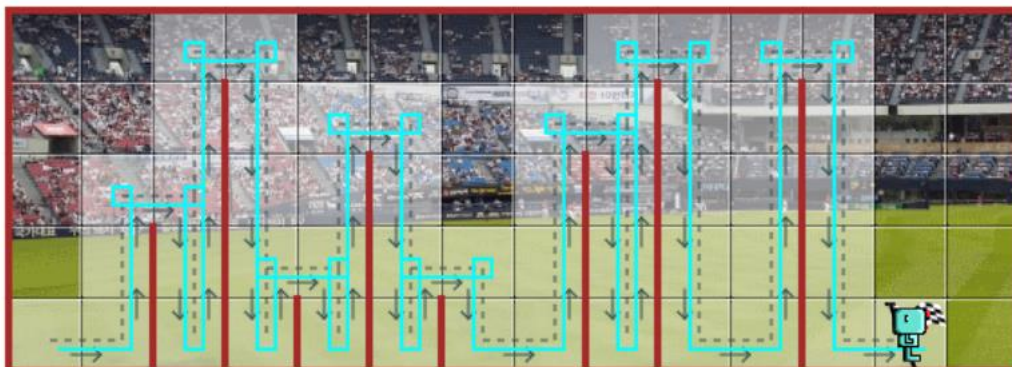
5. “if” artinya “Jika” jadi jika robot telah mencapai finish maka tugas robot selesai “done”

```
4   if at_goal():
5       done()
```

6. Dan sebaliknya jika robot belum mencapai finish robot akan melanjutkan perintah yang sudah tertera.

```
20  while not at_goal():
21      lompat()
```

Output :



- Maze

Source Code :

```

Python Code library A↓A A↑A
1 def turn_right():
2     turn_left()
3     turn_left()
4     turn_left()
5
6 def maze():
7     if not wall_in_front():
8         move()
9         if at_goal():
10            done
11        elif not wall_on_right():
12            turn_right()
13    else:
14        turn_left()
15
16 while not at_goal():
17     maze()

```

Penjelasan Step By Step :

1. “def” Digunakan untuk menyimpan suatu proses agar bisa digunakan secara terus menerus dan digunakan untuk mengelompokkan suatu perintah.

```

Python Code library A↓A A↑A
1 def turn_right():
6 def maze():

```

2. berikan perintah di dalam “def” agar bisa dipanggil secara terus menerus.

```

Python Code library A↓A A↑A
1 def turn_right():
2     turn_left()
3     turn_left()
4     turn_left()

```

```

6 def maze():
7     if not wall_in_front():
8         move()
9         if at_goal():
10            done()
11        elif not wall_on_right():
12            turn_right()
13    else:
14        turn_left()

```

3. “if” artinya “Jika”, “elif not” artinya “Jika Tidak”, dan “else” artinya “Selain itu” jadi jika tidak ada dinding di depan robot “if not wall_in_front()”, maka robot akan melakukan sesuai perintah yang tertera,

```

7     if not wall_in_front():
8         move()

```

4. Lalu, jika tidak ada dinding di samping kanan robot “elif not wall_on_right()”, maka robot akan melakukan sesuai perintahnya

```

11        elif not wall_on_right():
12            turn_right()

```

5. Selain “if” diatas maka robot akan melakukan perintah “turn left()”,

```

13    else:
14        turn_left()

```

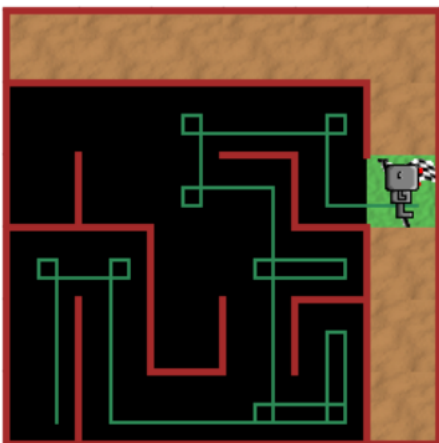
6. Ketika (While) kondisi tersebut tidak sesuai maka mengulang perintah berikut.

```

16 while not at_goal():
17     maze()

```

Output :



Link Github :

https://github.com/dikysetiawan21/2023F_KELOMPOK_7