LAPORAN TUGAS ALGORITMA DAN PEMROGRAMAN

HURDLE 1-4



Disusun oleh:

KELOMPOK 8

1. Rafi Azis Rachman (23091397185)

2. Jocellina Keiza Vallentina (23091397186)

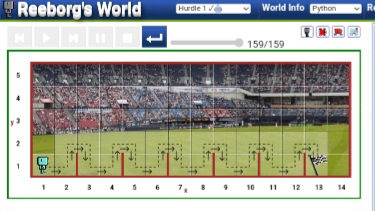
3. Achmad Zulfahmi (23091397198)

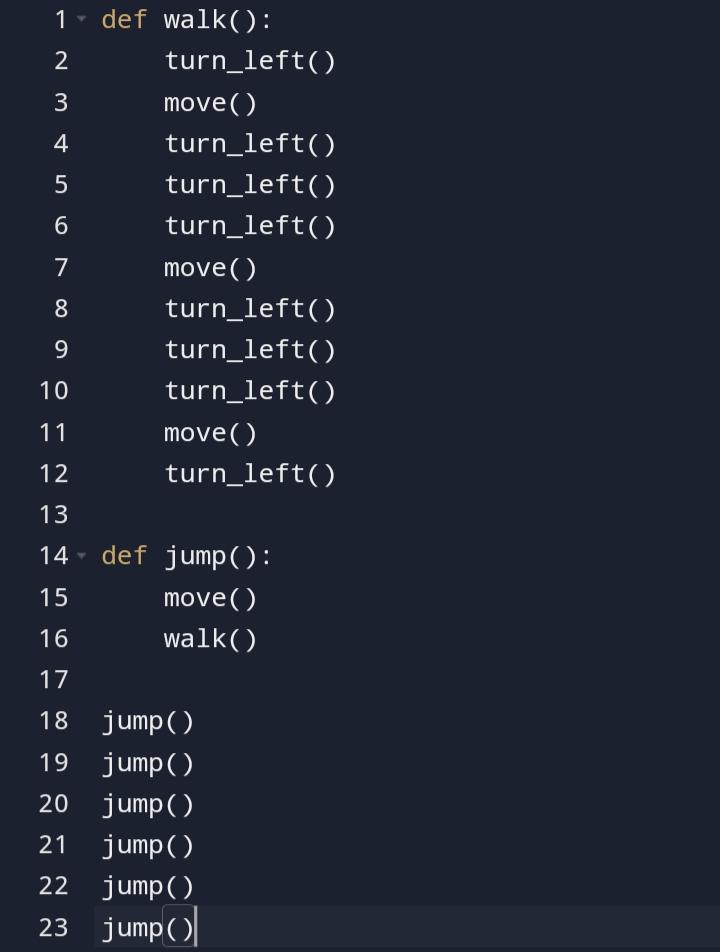
PROGRAM STUDI D4 MANAJEMEN INFORMATIKA

FAKULTAS VOKASI

UNIVERSITAS NEGERI SURABAYA

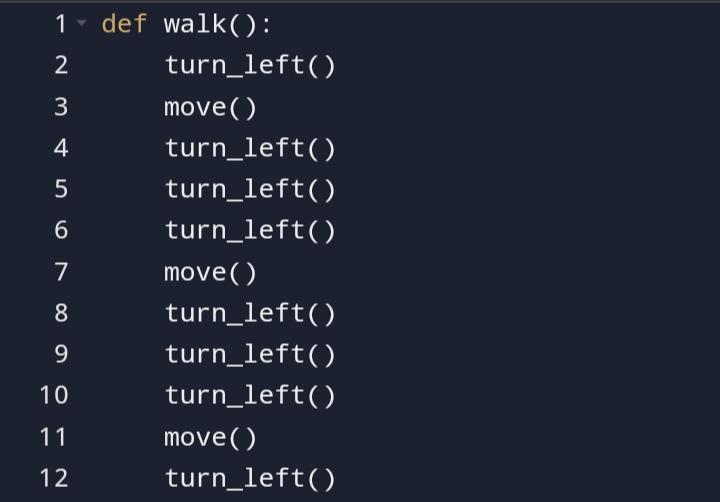
2023/2024

**Hurdle 1**

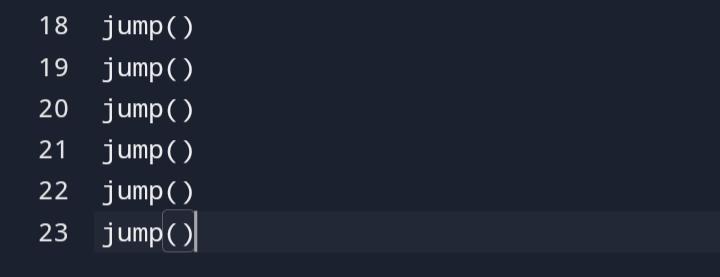
* **Source Code**

Menggunakan function (def )

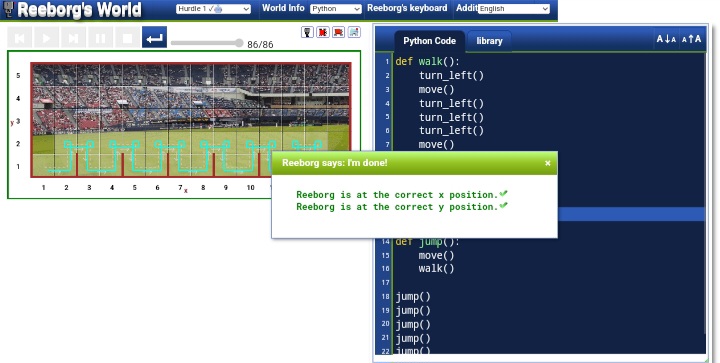
Hurdle 1 menggunakan Def yang berfungsi untuk mendefinisikan sebuah fungsi.

* **Penjelasan step by step**

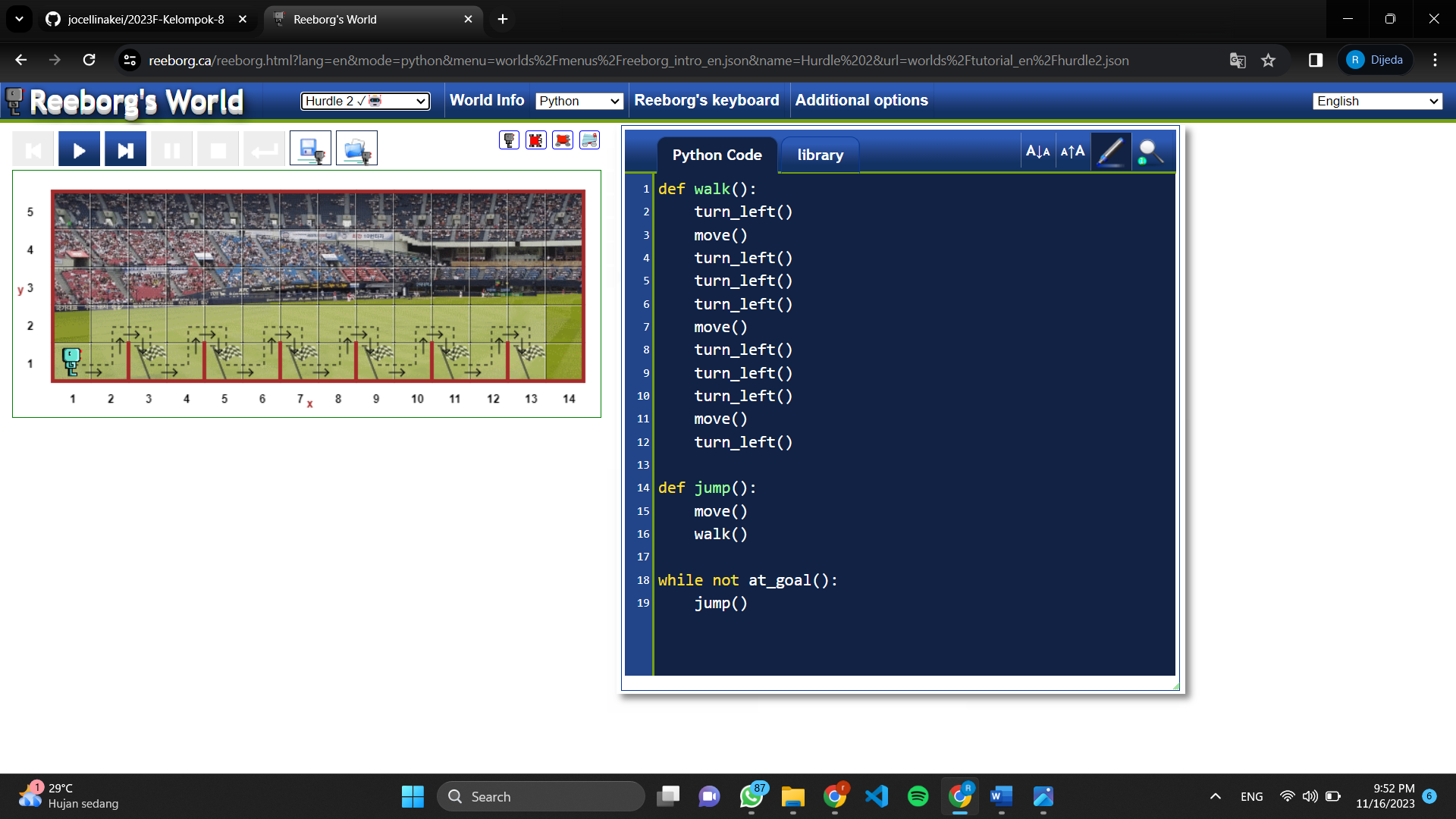
Pertama menggunakan fungsi walk() untuk menjalankan robot cara berjalan. Dengan menggunakan turn\_left() dan move().

Kedua menggunakan fungsi jump() untuk mengetahui cara kerja robot cara melewati rintangan. Dengan menjalankan move() dan walk().

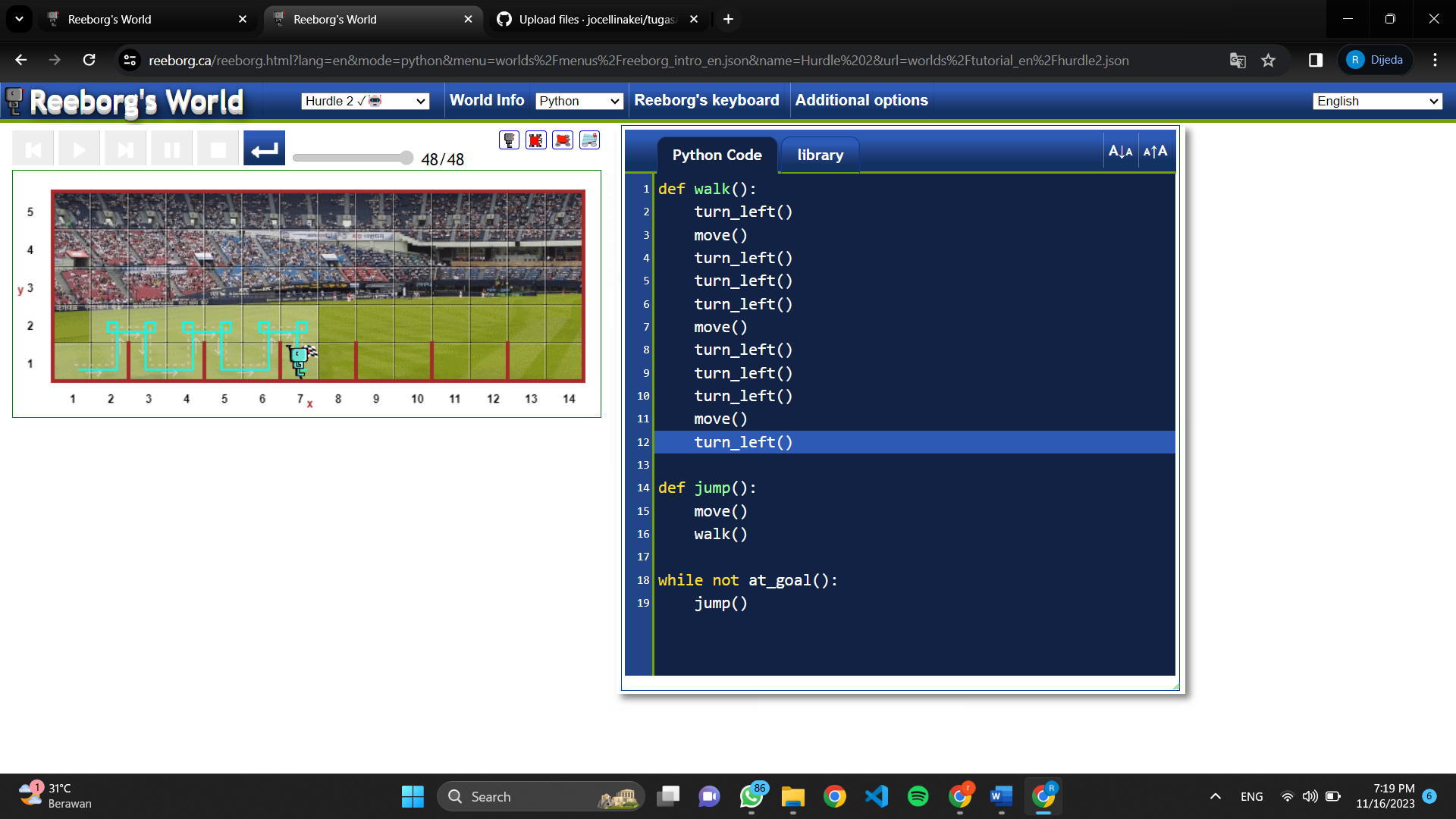
Terakhir menggunakan jump() untuk menjalankan fungsi jump() dengan sebanyak 6 kali karena terdapat rintangan sebanyak 6 dinding.

* **Output**

**Hurdle 2**



* **Source Code**



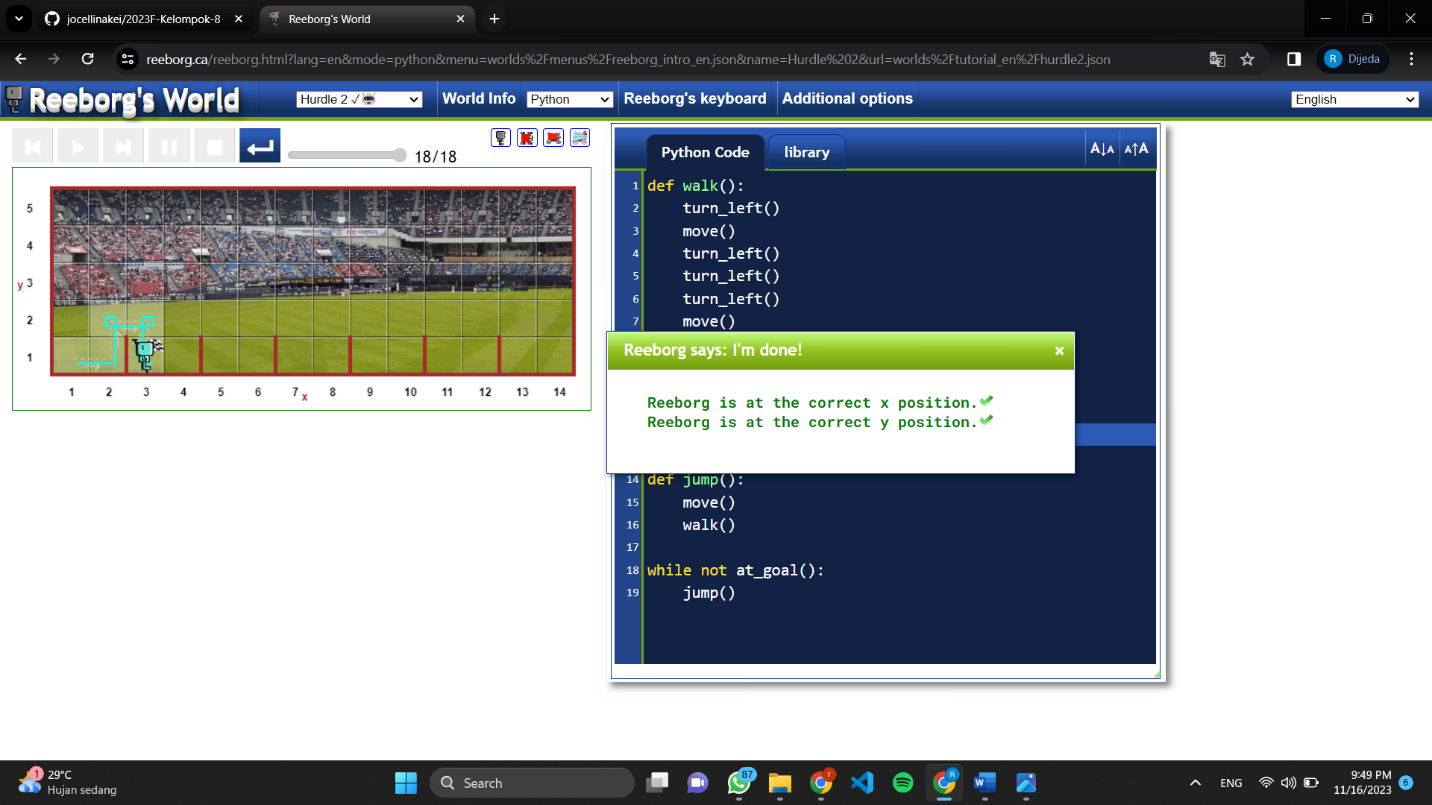
* **Penjelasan Step by step**

Fungsi walk() digunakan untuk menjalankan robot agar dapat berjalan. Kemudian turn\_left() berfungsi untuk memutar robot ke kiri dan move() untuk membuat robot bergerak maju.

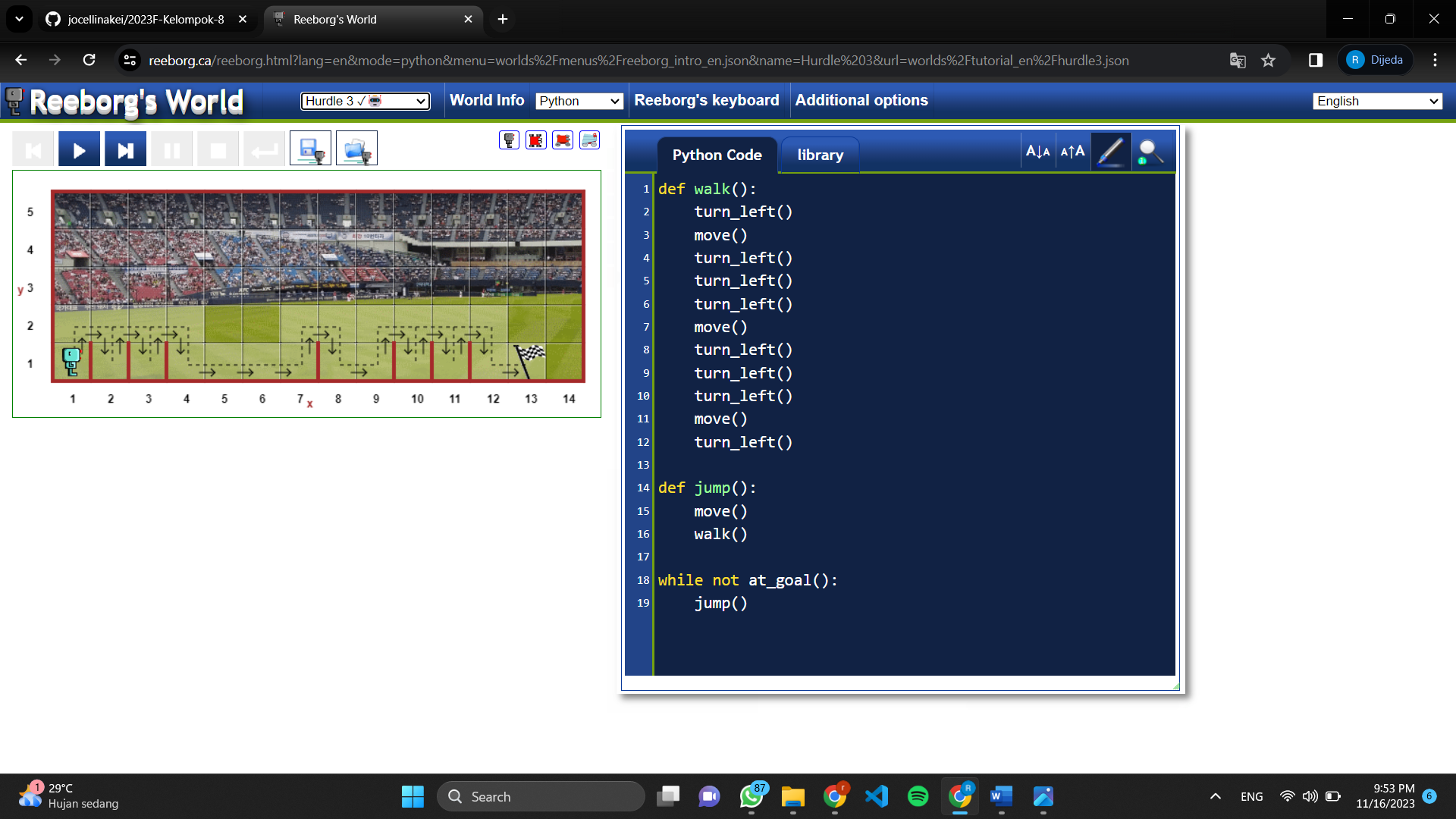
Fungsi jump() digunakan untuk memerintah robot agar dapat melompat. move() agar robot bergerak maju dan walk() agar robot berjalan.

Fungsi while not at\_goal() digunakan ketika robot belum mencapai garis finish, maka akan lanjut melakukan perulangan jump() sampai mencapai garis finish.

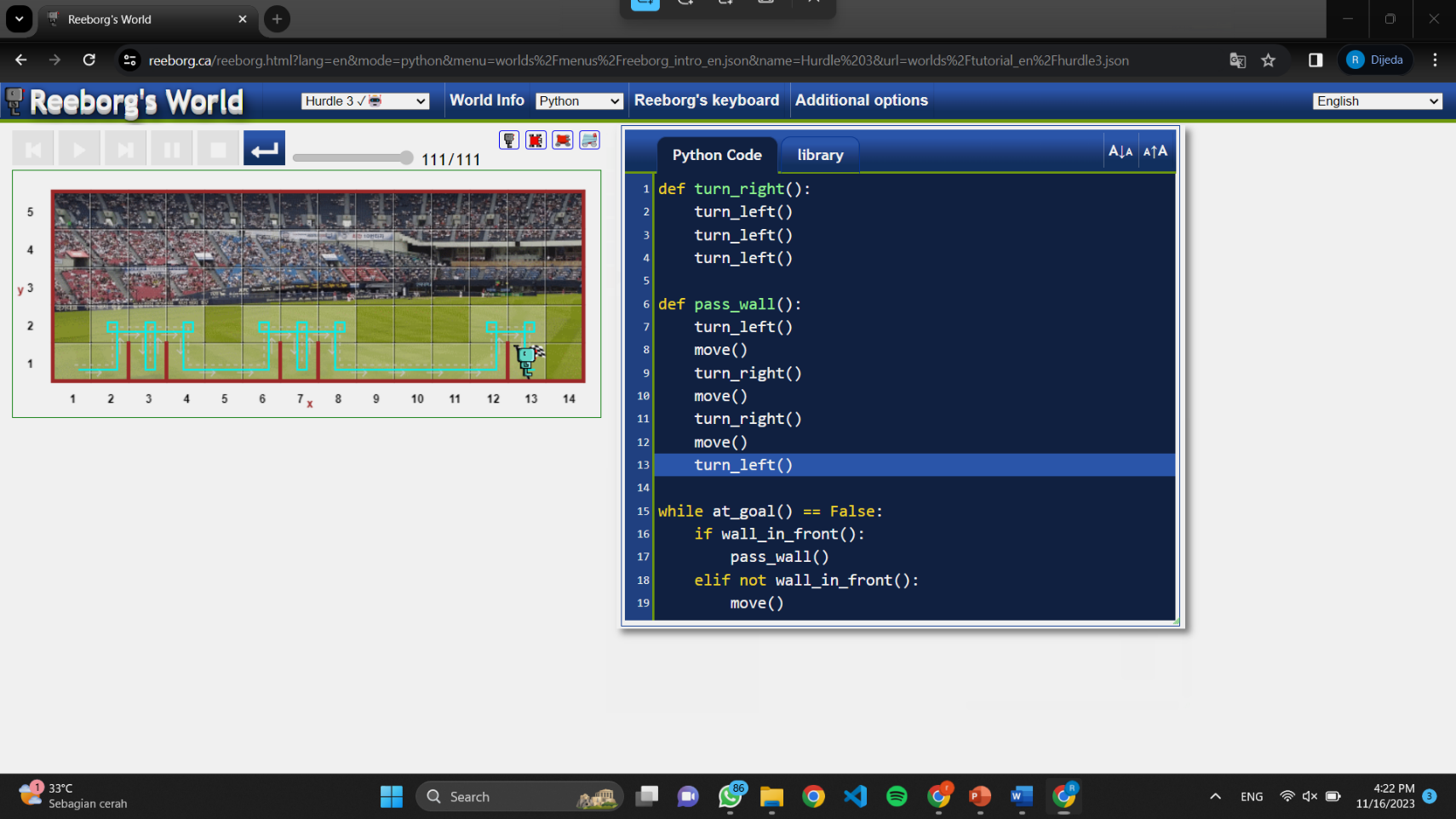
* **Output**



**Hurdle 3**



* **Souce Code**

****

* **Penjelasan Step by Step**

def turn\_right():

    turn\_left()

    turn\_left()

    turn\_left()

pertama untuk menjalankan robot adalah dengan turn\_right dan turn\_left 3 kali

def pass\_wall():

Lalu menggunakan fungsi pass\_wall agar robot melewati sebuah dinding

 turn\_left()

    move()

    turn\_right()

    move()

    turn\_right()

    move()

    turn\_left()

dilanjutkan dengan fungsi turn\_left, turn\_right dan move() agar robot tetap lanjut berjalan

while at\_goal() == False:

fungsi diatas digunakan agar dapat melooping program sebelum robot sampai tujuan terakhir

 if wall\_in\_front():

        pass\_wall()

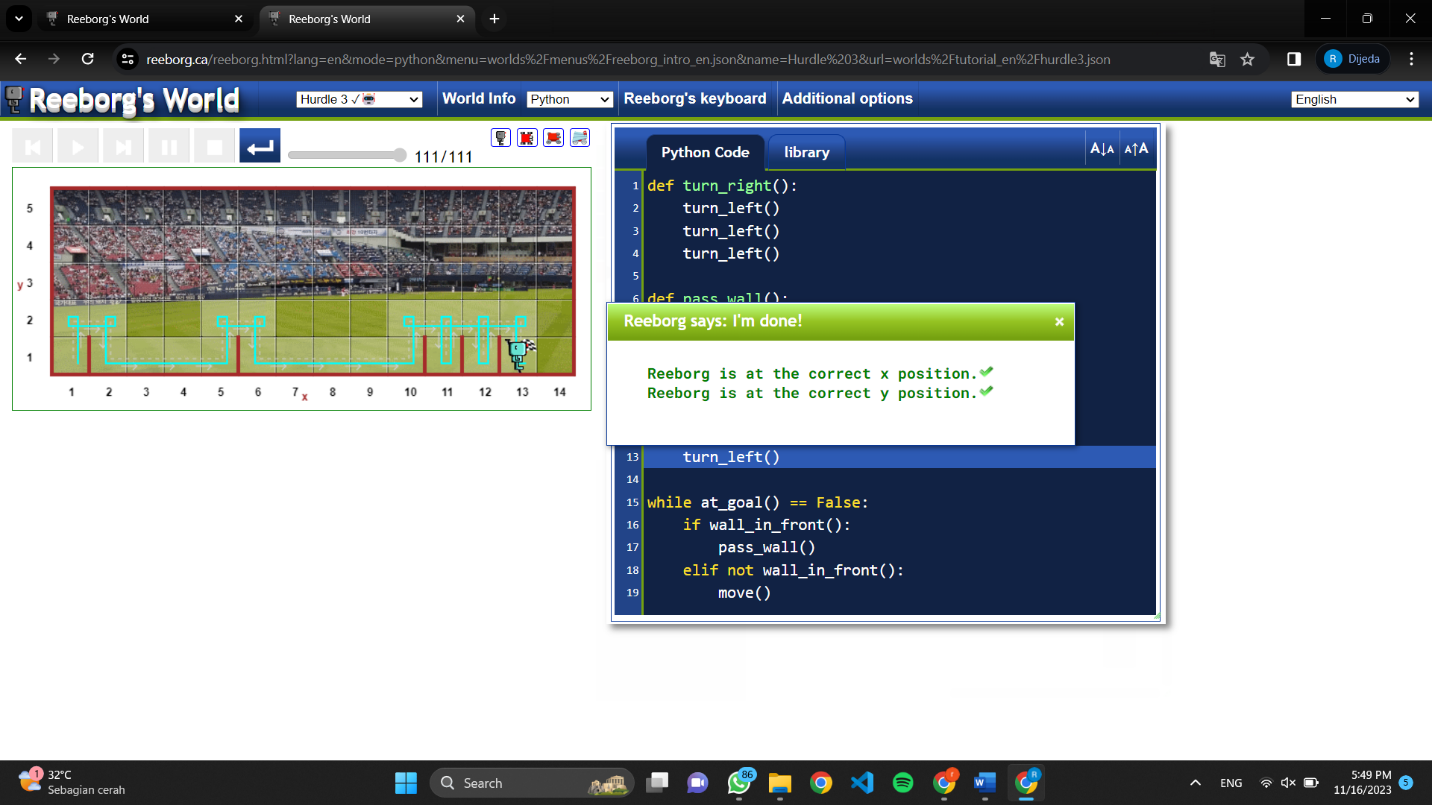
    elif not wall\_in\_front():

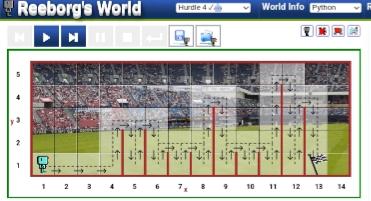
        move()

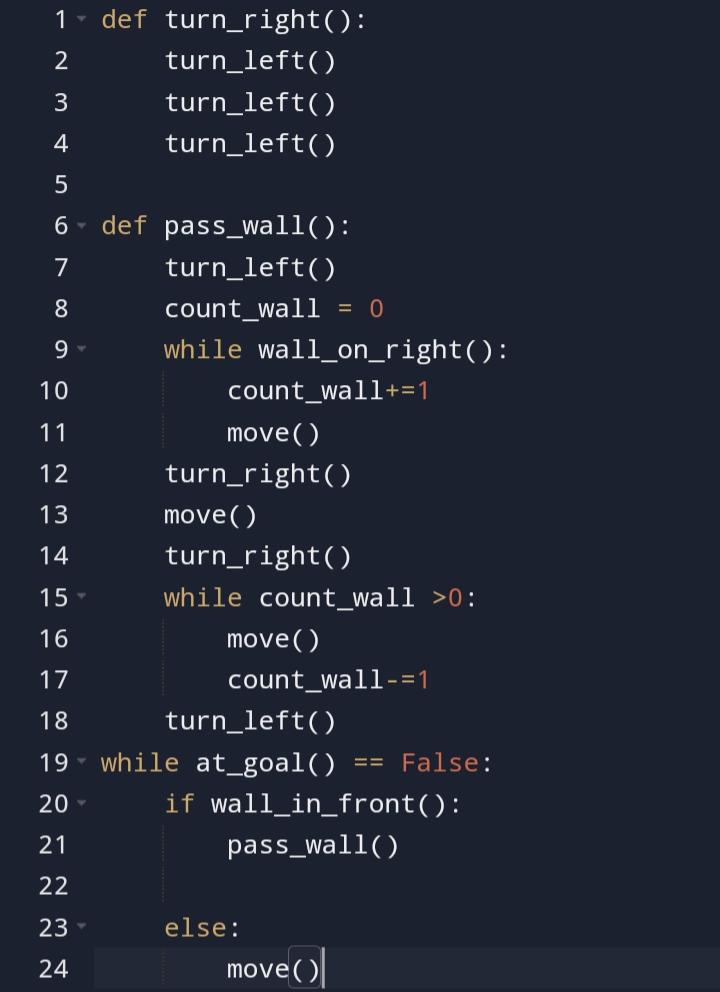
Jika ada dinding di depan wall\_in\_front() bernilai benar, maka program akan menjalankan fungsi pass\_wall() untuk melewati dinding.

Jika tidak ada dinding di depan not wall\_in\_front(), maka robot akan maju satu langkah dengan menggunakan perintah move().

* **Output**



**Hurdle 4**

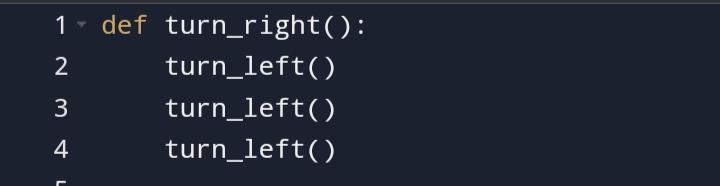
* **Source Code**

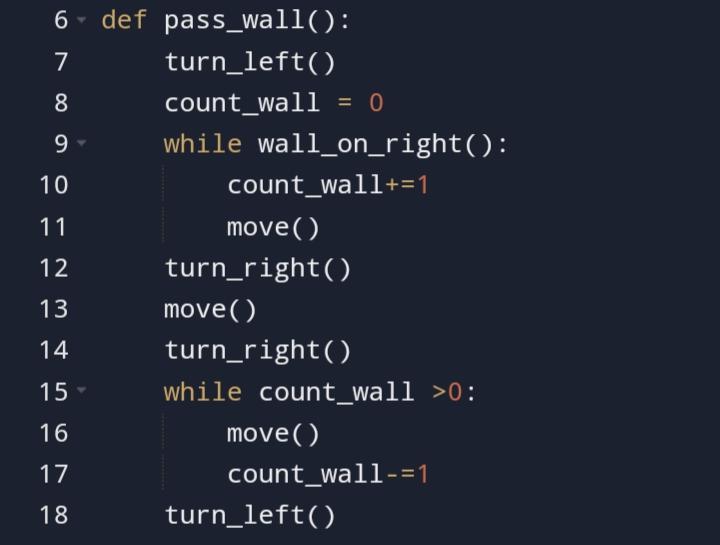
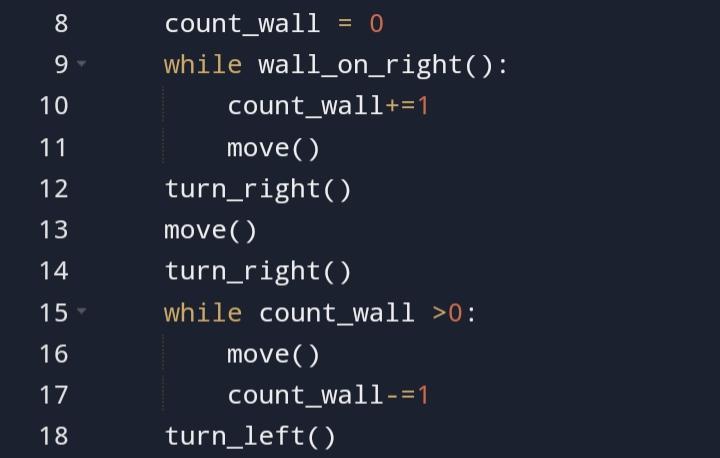
Menggunakan fungsi def, while

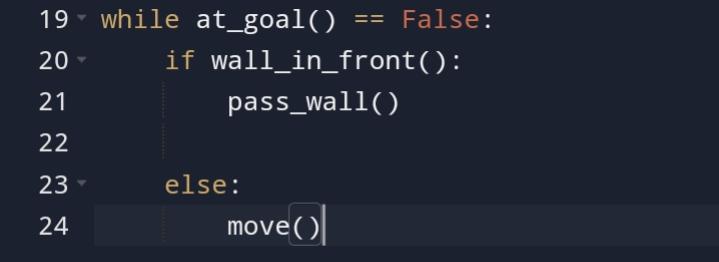
Def sendiri berfungsi untuk mendefinisikan sebuah fungsi.

While berfungsi untuk melooping program.

* **Penjelasan step by step**

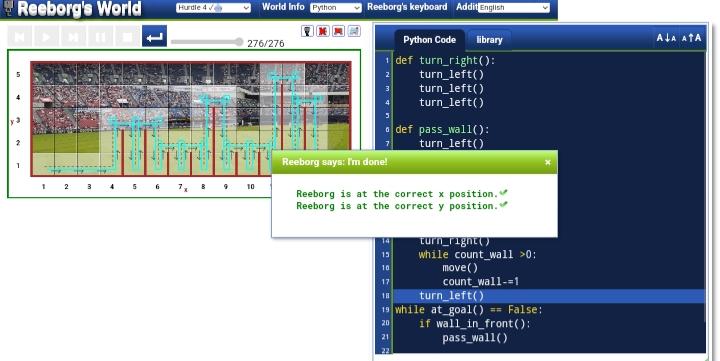
Pertama memasukkan fungsi def turn\_right() agar bisa menjalankan robot dengan belok ke kanan. Dengan cara turn\_left sebanyak 3 kali.

Kedua menggunakan fungsi def pass\_wall() agar robot bisa melewati dinding. Dengan cara turn\_left agar robot bisa berbelok ke kiri. Count\_wall untuk menghitung jumlah rintangan yang akan dilewati.

Ketiga menggunakan while yang berfungsi untuk melooping. Apabila dinding berada di kanan maka rintangan bertambah satu. Lalu menjalankan robot dengan move(), turn\_right(), move(), dan turn\_right(). Apabila count\_wall >0 maka tidak ada rintangan lalu menjalankan fungsi move(). Apabila count\_wall -=1 maka rintangannya tidak berhadapan atau melewati sama sekali lalu menjalankan turn\_left() agar robot bisa belok.

Keempat menggunakan while yang berfungsi melooping program agar sampai tujuan.

Jika terdapat dinding di depan, maka menjalankan fungsi pass\_wall(). Jika tidak terdapat dinding di depan maka menjalankan fungsi move().

****

* **Output**