

LAPORAN PRAKTIKUM REEBORG

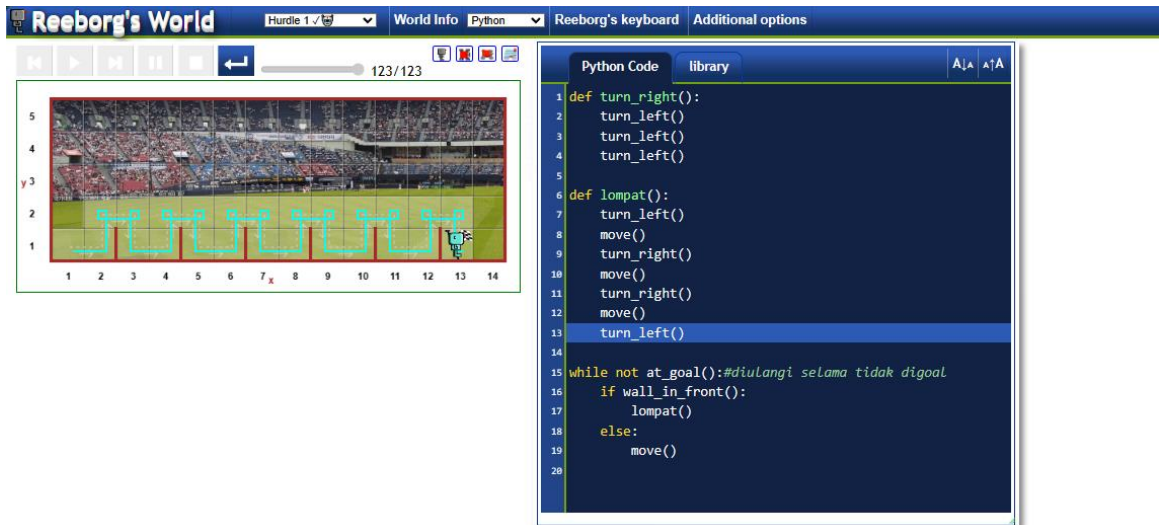


Nama Anggota:

- Baharuddin Ziyat Agustian/23091397094
- Dicky Dippos Sihite/23091397071
- Roberto Christian/23091397101

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

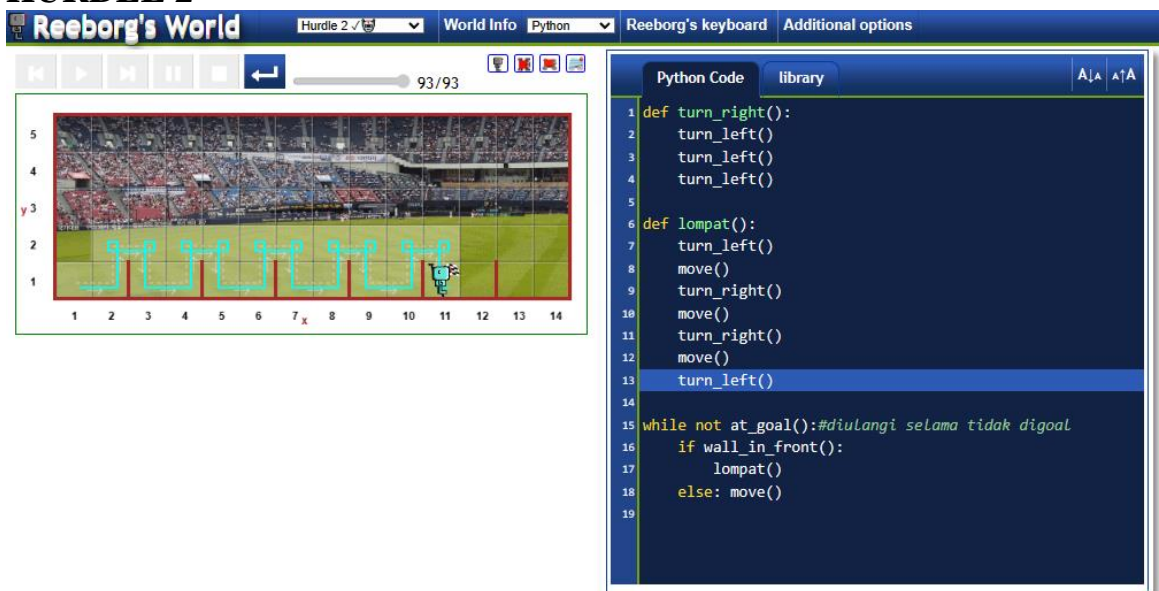
❖ HURDLE 1



• Penjelasan

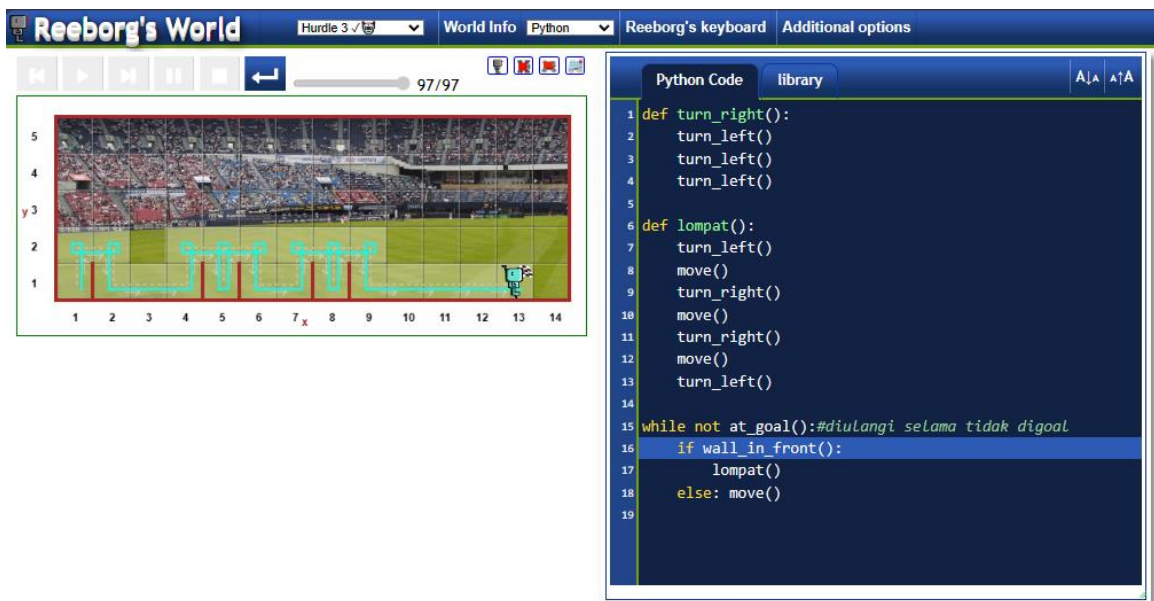
- `turn_right()`: Fungsi ini memanggil fungsi `turn_left()` tiga kali, yang membuat robot melakukan putaran ke kanan.
- `lompat()`: Fungsi ini memiliki urutan gerakan. Pertama, robot akan berputar ke kiri, kemudian bergerak maju, berputar ke kanan tiga kali, dan bergerak maju lagi. Akhirnya, robot akan berputar ke kiri.
- `while not at_goal()`: Perulangan ini akan terus berjalan selama robot belum mencapai tujuan (goal).
- Di dalam perulangan, terdapat kondisi `if wall_in_front()`: Jika terdapat dinding di depan robot, maka fungsi `lompat()` akan dipanggil, sebaliknya, jika tidak ada dinding di depan, robot akan melakukan gerakan maju dengan memanggil fungsi `move()`.

❖ HURDLE 2



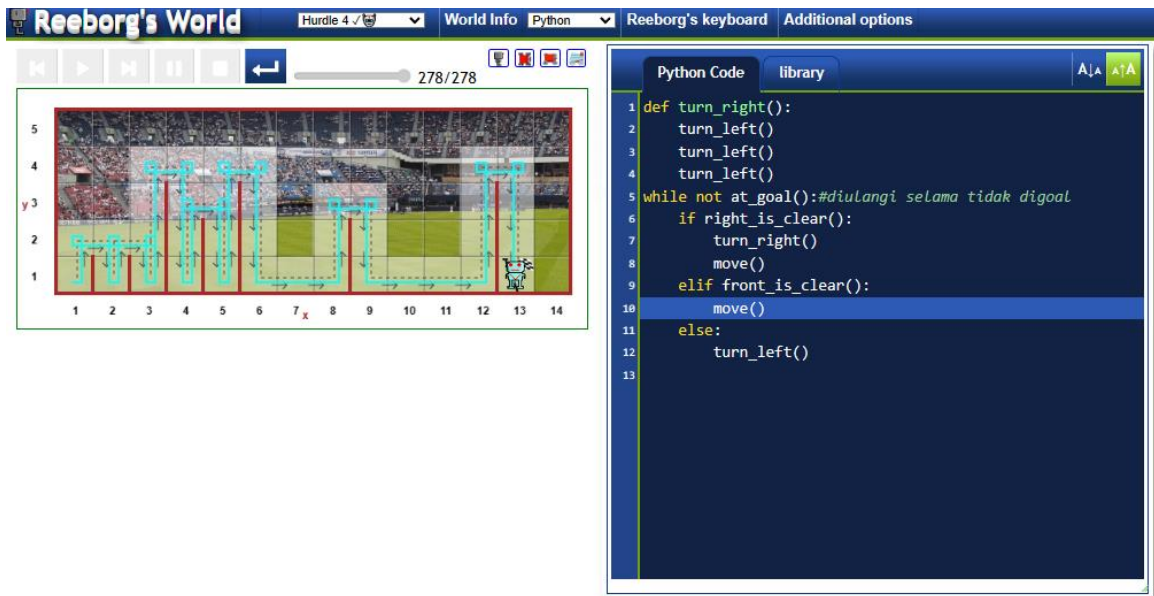
- Penjelasan
 - `turn_right()`: Fungsi ini memanggil fungsi `turn_left()` tiga kali, yang membuat robot melakukan putaran ke kanan.
 - `lompat()`: Fungsi ini memiliki urutan gerakan. Pertama, robot akan berputar ke kiri, kemudian bergerak maju, berputar ke kanan tiga kali, dan bergerak maju lagi. Akhirnya, robot akan berputar ke kiri.
 - `while not at_goal()`: Perulangan ini akan terus berjalan selama robot belum mencapai tujuan (goal).
 - Di dalam perulangan, terdapat kondisi `if wall_in_front()`: Jika terdapat dinding di depan robot, maka fungsi `lompat()` akan dipanggil, sebaliknya, jika tidak ada dinding di depan, robot akan melakukan gerakan maju dengan memanggil fungsi `move()`.

❖ HURDLE 3



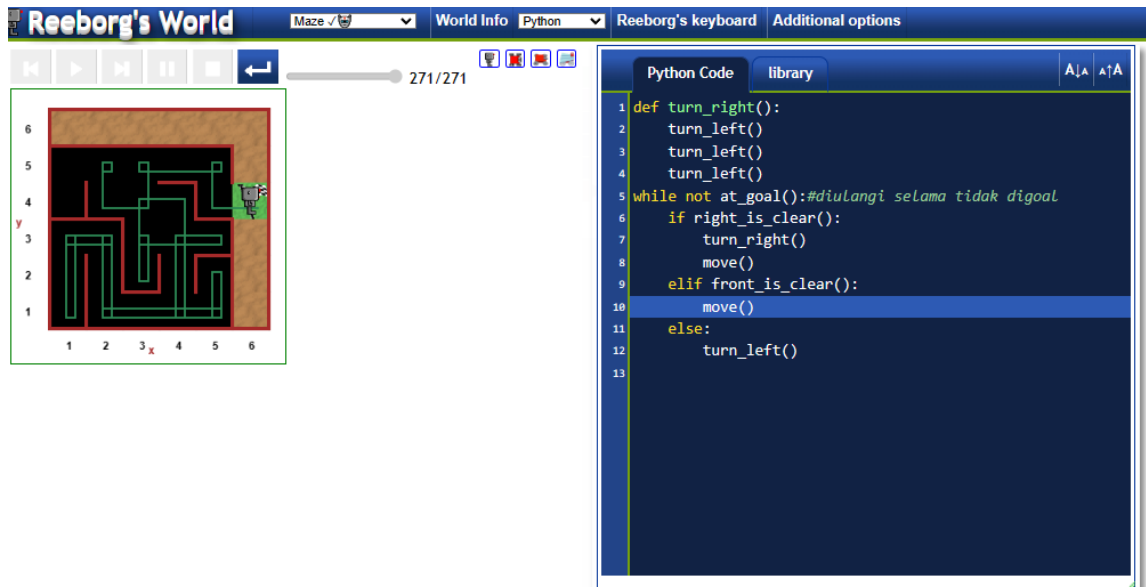
- Penjelasan
 - `turn_right()`: Fungsi ini memanggil fungsi `turn_left()` tiga kali, yang membuat robot melakukan putaran ke kanan.
 - `lompat()`: Fungsi ini memiliki urutan gerakan. Pertama, robot akan berputar ke kiri, kemudian bergerak maju, berputar ke kanan tiga kali, dan bergerak maju lagi. Akhirnya, robot akan berputar ke kiri.
 - `while not at_goal()`: Perulangan ini akan terus berjalan selama robot belum mencapai tujuan (goal).
 - Di dalam perulangan, terdapat kondisi `if wall_in_front()`: Jika terdapat dinding di depan robot, maka fungsi `lompat()` akan dipanggil, sebaliknya, jika tidak ada dinding di depan, robot akan melakukan gerakan maju dengan memanggil fungsi `move()`.

❖ HURDLE 4



- Penjelasan
 - `turn_right()`: Fungsi ini memanggil fungsi `turn_left()` tiga kali, yang membuat robot melakukan putaran ke kanan.
 - `while not at_goal()`: Perulangan ini akan terus berjalan selama robot belum mencapai tujuan (goal).
 - `if right_is_clear()`:
jika jalur kanan bebas tidak ada tembok di kanan, maka code berikut akan dijalankan:
 - `turn_right()`
Jika tidak ada tembok di kanan, maka fungsi `turn_right()` akan dipanggil. Seperti yang telah dijelaskan sebelumnya, ini setara dengan tiga kali pemanggilan fungsi `turn_left()`.
 - `move()`
Setelah memanggil `turn_right()`, fungsi `move()` dipanggil. Ini akan membuat robot bergerak maju satu langkah.
 - `elif front_is_clear()`:
Jika jalur didepan bebas tidak ada tembok didepan. maka code berikut akan dijalankan.
 - `move()`
Jika jalur ke depan bebas, maka fungsi `move()` dipanggil. Ini akan membuat robot bergerak maju satu langkah.
 - `else:`
Jika tidak ada jalur ke kanan yang bebas dan tidak ada jalur ke depan yang bebas, maka code berikut akan dijalankan.:
 - `turn_left()`
Di dalam `else`, fungsi `turn_left()` dipanggil, yang membuat robot berputar ke kiri.

❖ MAZE



- Penjelasan
 - turn_right(): Fungsi ini memanggil fungsi turn_left() tiga kali, yang membuat robot melakukan putaran ke kanan.
 - while not at_goal(): Perulangan ini akan terus berjalan selama robot belum mencapai tujuan (goal).
 - if right_is_clear():
jika jalur kanan bebas tidak ada tembok di kanan, maka code berikut akan dijalankan:
 - turn_right()
Jika tidak ada tembok di kanan, maka fungsi turn_right() akan dipanggil. Seperti yang telah dijelaskan sebelumnya, ini setara dengan tiga kali pemanggilan fungsi turn_left().
 - move()
Setelah memanggil turn_right(), fungsi move() dipanggil. Ini akan membuat robot bergerak maju satu langkah.
 - elif front_is_clear():
Jika jalur di depan bebas tidak ada tembok di depan. maka code berikut akan dijalankan.
 - move()
Jika jalur ke depan bebas, maka fungsi move() dipanggil. Ini akan membuat robot bergerak maju satu langkah.
 - else:
Jika tidak ada jalur ke kanan yang bebas dan tidak ada jalur ke depan yang bebas, maka code berikut akan dijalankan.:
 - turn_left()
Di dalam else, fungsi turn_left() dipanggil, yang membuat robot berputar ke kiri.

Link Github :

<https://github.com/Agustian1408/Tugas-2>