

## **A. Judul**

Laporan Praktik Robotika tentang Robot Line Follower

## **B. Nama Anggota**

- a) Nurul Arifin                      NIM : 19507334034
- b) Fiosa Putra                      NIM : 19507334021

## **C. Hari/Tanggal Praktikum**

Praktikum dilakukan pada hari Jum'at, 08 Oktober 2021 dan laporan ini ditulis pada hari Senin, 11 Oktober 2021

## **D. Tujuan Praktikum**

Adapun tujuan dari praktikum ini adalah:

- a. Memahami serta mengetahui bagaimana proses pembuatan simulasi robot Line Follower yang ada pada simulator webots sampai berhasil dijalankan dengan baik.
- b. Mempelajari dan mempraktikkan secara simulasi proses algoritma pemrograman untuk menjalankan Robot Line Follower.
- c. Memahami serta memperhatikan mekanisme kerja Robot Line Follower yang berhasil dijalankan.

## **E. Alat dan Bahan Praktikum**

Alat dan Bahan praktikum adalah:

- a. Laptop (OS Windows)
- b. Simulator Webots
- c. Python 3.9

## **F. Prosedur Kerja**

Langkah kerja yang dilakukan adalah sebagai berikut:

- a. Membuka program python compiler untuk proses pemrograman yang diperlukan pada simulasi Robot Line Follower, yang nantinya hasil pemrograman di copy dan paste pada bagian editor yang ada di webots sebelah kanan.

- b. Membuka webots untuk memulai simulasi bisa kita pilih open file lalu pilih webots laboratory dari github yang telah tersedia.
- c. Setelah itu baru akan termuat arena yang telah tersedia tadi dengan robot Line Follower yang telah ada pada arena tersebut.
- d. Pada saat yang bersamaan kita akan mencoba running robot tersebut terlebih dahulu dan akan tampil robot tersebut belum bisa terkendali dengan baik yaitu masih berjalan lurus tanpa mengikuti line pada arena tersebut dengan baik, maka diperlukan tindakan berupa mencari program untuk mengakses sensor – sensor yang ada pada robot tersebut.
- e. Setelahnya mulai untuk merubah dan mencari algoritma yang tepat untuk dapat mengakses sensor pada Robot Line Follower.
- f. Setelah dirasa telah berhasil dijalankan maka diperlukan rekam layar dan screenshot untuk mendukung pembuatan laporan praktikum.

#### **G. Listing Algoritma Pemrograman**

```
from controller import Robot

tembok = 2
garis_kanan = 1
garis_kiri = 1

def run_robot(robot):

    time_step = 16
    max_speed = 8

    def maju():
        rightmotor.setVelocity(6)
        leftmotor.setVelocity(6)
    def belok_kiri():
        rightmotor.setVelocity(4)
```

```
    leftmotor.setVelocity(0)
def belok_kanan():
    rightmotor.setVelocity(0)
    leftmotor.setVelocity(4)
def berhenti():
    rightmotor.setVelocity(0)
    leftmotor.setVelocity(0)

def linefollower():
    if ir_kanan > 1400 and ir_kiri < 1400:
        belok_kiri()
    elif garis < 2900:
        maju()
    elif ir_kanan < 1400 and ir_kiri > 1400:
        belok_kanan()
    elif line_kiri < 500:
        berhenti()
        belok_kiri()
def fix():
    if ir_kanan > 1400 and ir_kiri < 1400:
        belok_kiri()
    elif ir_kanan < 1400 and ir_kiri > 1400:
        belok_kanan()
    # elif line_kanan < 500:
    #     hadap_kanan()
    elif garis > 2900 :
        if line_kanan < 1200 :
            belok_kiri()
        elif (ir_kanan > 1400 and garis_tengah > 400) or (ir_kiri > 1400 and garis_tengah
> 400) :
            maju()
    elif ping_depan > 200:
```

```
maju()
elif ping_depan < 200:
    berhenti()
else :
    maju()
def pas():
    if(garis > 3600 and ping_kanan < 1000) or (garis > 3600 and ping_kiri < 1000):
        maju()
def palang():
    if line_kanan < 500 and line_kiri > 600:
        belok_kanan()
    elif line_kanan < 500 and line_kanan > 350 and garis_tengah > 600:
        maju()
    elif ping_depan < 200:
        berhenti()
    elif line_kanan < 350 and line_kiri < 350:
        berhenti()

robot = Robot()
timestep = int(robot.getBasicTimeStep())

leftmotor = robot.getDevice('motor_1')
rightmotor = robot.getDevice('motor_2')
leftmotor.setPosition(float('inf'))
rightmotor.setPosition(float('inf'))

pingkiri = robot.getDevice('ds_left')
pingkanan = robot.getDevice('ds_right')
pingdepan = robot.getDevice('ds_front')
irl2 = robot.getDevice('IRL2')
irl1 = robot.getDevice('IRL1')
ircl = robot.getDevice('IRCL')
```

```
ircr = robot.getDevice('IRCR')
irr1 = robot.getDevice('IRR1')
irr2 = robot.getDevice('IRR2')

irl2.enable(timestep)
irl1.enable(timestep)
ircl.enable(timestep)
ircr.enable(timestep)
irr1.enable(timestep)
irr2.enable(timestep)
pingkiri.enable(timestep)
pingkanan.enable(timestep)
pingdepan.enable(timestep)

while robot.step(timestep) != -1:
    # rightmotor.setVelocity(10)
    # leftmotor.setVelocity(10)

    irl2_val = irl2.getValue()
    irl1_val = irl1.getValue()
    ircl_val = ircl.getValue()
    ircr_val = ircr.getValue()
    irr1_val = irr1.getValue()
    irr2_val = irr2.getValue()
    ping_kiri = pingkiri.getValue()
    ping_kanan = pingkanan.getValue()
    ping_depan = pingdepan.getValue()

    line_kanan = irr1_val + irr2_val
    line_kiri = irl1_val + irl2_val
    ir_kanan = ircr_val + irr1_val + irr2_val
    ir_kiri = ircl_val + irl1_val + irl2_val
```

```
garis_tengah = ircr_val + ircl_val
garis = ir_kanan + ir_kiri

print('{:.2f} {:.2f} {:.2f} {:.2f} {:.2f} {:.2f} '.format(line_kanan, line_kiri,
ir_kanan, ir_kiri, garis_tengah, garis))

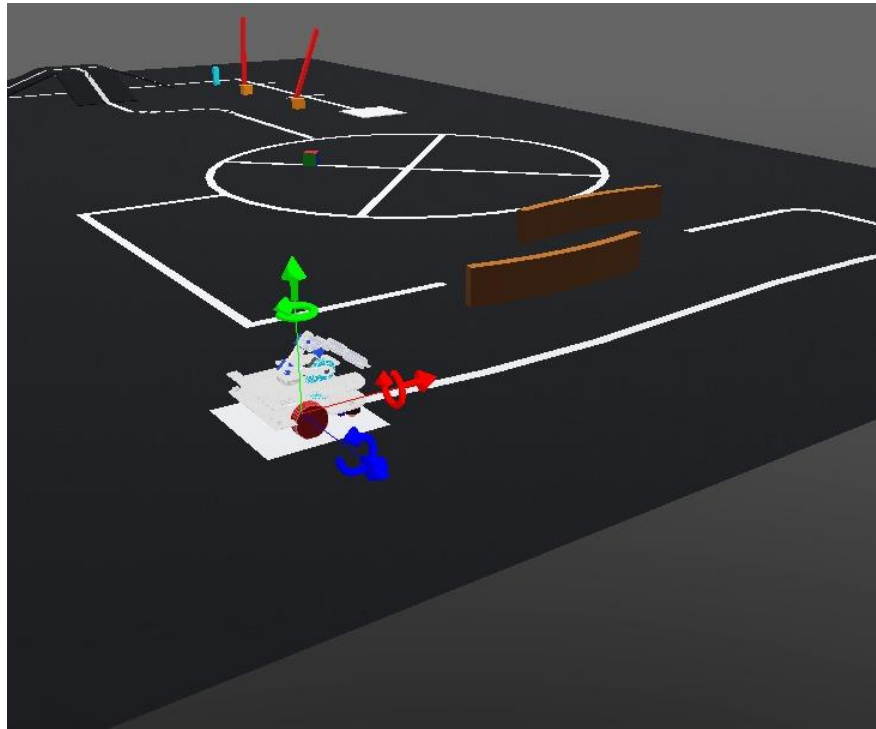
if tembok == 2:
    fix()

# if tembok == 2:
#     misal()
#     pas()
# if garis < 3000 and ping_kiri < 1000:
#     tembok = 3
# elif tembok == 3:
#     fix()
#     if ping_kanan < 900:
#         tembok = 4
# elif tembok == 4:
#     palang()
```

## H. Hasil Pengamatan

Adapun hasil pengamatan pada praktikum ini yaitu percobaan simulasi Robot Line Follower sebagai berikut:

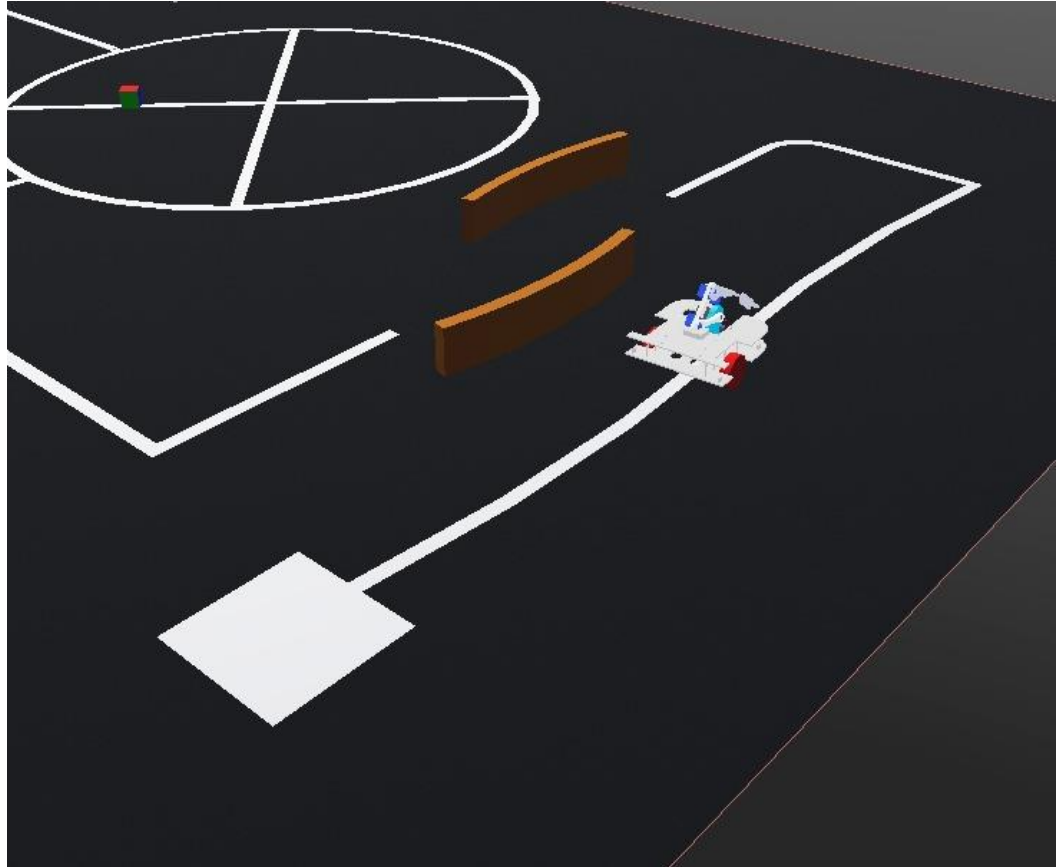
1. Berikut merupakan kondisi awal yang tampak pada arena line follower, yaitu terdapat robot yang telah tersedia dan siap di jalankan.



2. Selanjutnya merupakan kondisi dimana nilai pada sensor berkisar diantara  $\geq 1000$  yang tertampil pada bagian console kiri bawah simulator webots, hal ini terjadi pada saat posisi robot berada pada garis lurus di arena tersebut,.

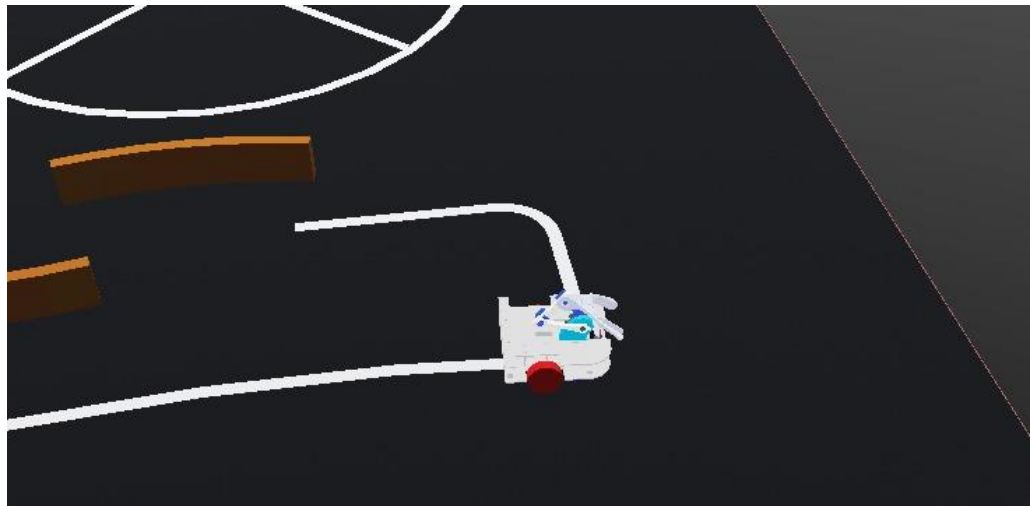
```
1208.55 1205.49 1395.68 1392.43 374.07 2788.11
1211.56 1208.45 1399.16 1395.86 375.00 2795.02
1214.54 1211.50 1402.60 1399.38 375.94 2801.98
1214.39 1211.58 1402.44 1399.46 375.93 2801.90
1214.22 1211.75 1402.27 1399.64 375.93 2801.90
1214.59 1212.31 1402.70 1400.28 376.07 2802.98
1233.85 1231.70 1424.97 1422.69 382.11 2847.66
1229.06 1227.13 1419.45 1417.40 380.65 2836.85
1182.73 1181.18 1365.89 1364.25 366.23 2730.14
1202.45 1201.66 1388.74 1387.89 372.51 2776.62
```

3. Kondisi di bawah ini merupakan keadaan dimana robot line follower sedang berjalan lurus maju kedepan saat melewati arena dengan garis lurus seperti yang tampak pada gambar dibawah ini.

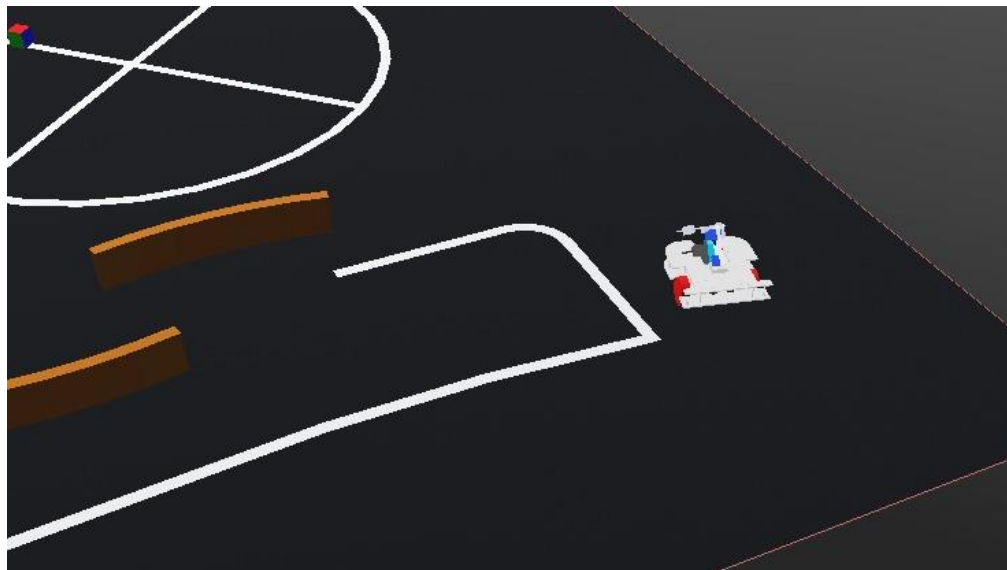


4. Selanjutnya robot berada pada kondisi atau keadaan dimana terdapat tikungan yang pertama, pada kondisi tersebut dapat kita lihat pembacaan sensor terdapat nilai pada sensor kiri yang nilainya berubah – ubah. Akan tetapi kondisi robot tidaklah langsung merespon berbelok ke kiri melainkan robot sedikit mengalami kendal berupa delay, sehingga robot masih mencari dan dalam keadaan berputar hingga robot tersebut mendapat garis atau pada posisi garis lagi.

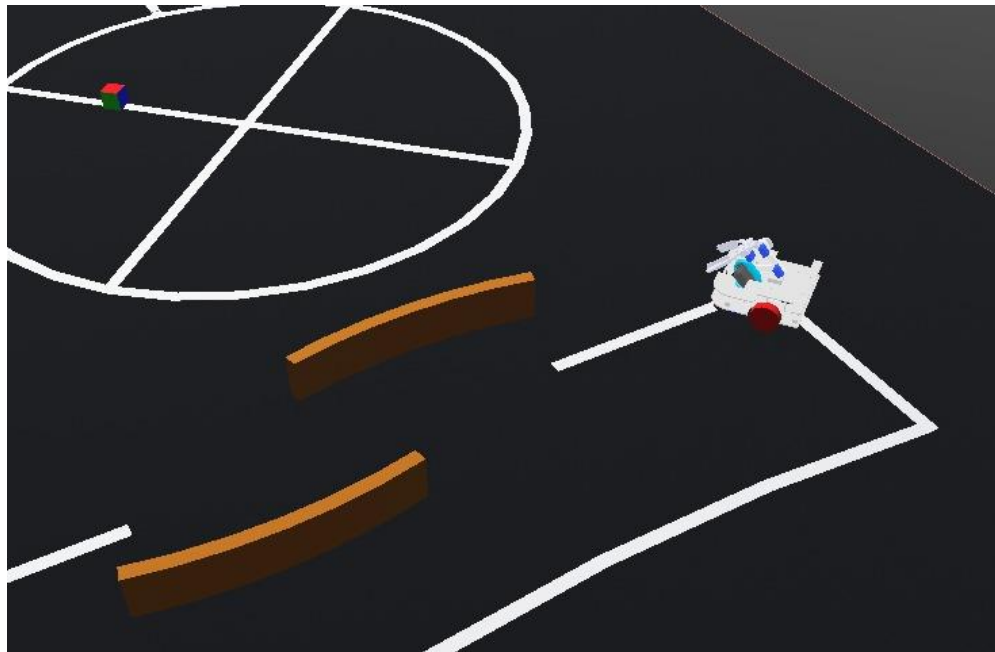




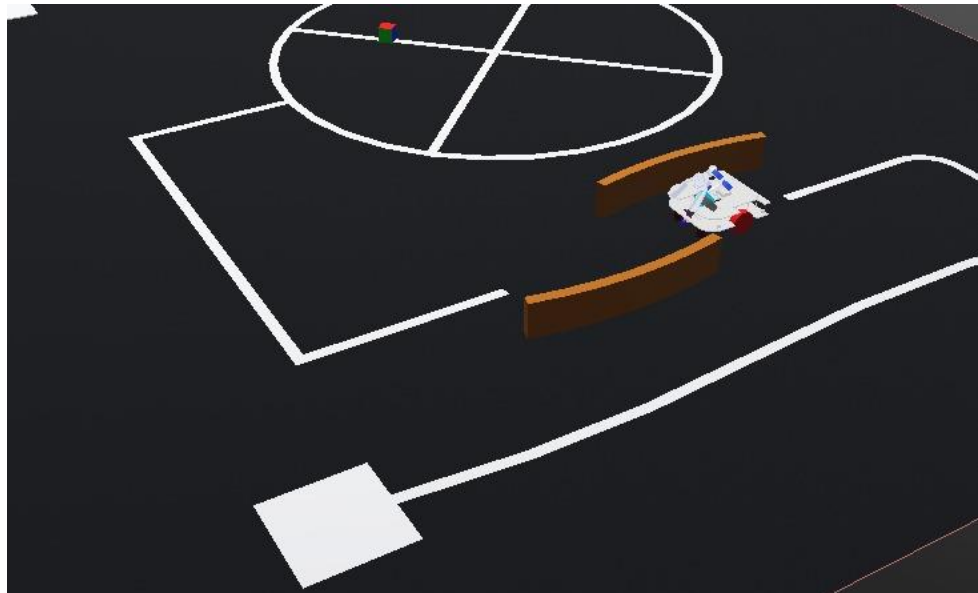
5. Namun terdapat kendala setelah berbelok ke kiri melalui tikungan yang pertama yaitu berupa robot akan keluar dari jalur atau garis yang ada didepannya, robot akan berputar dan keluar dari garis atau arena sampai menemukan garis lagi yang akan dilalui seperti yang tampak pada gambar dibawah ini.



6. Kedaan selanjutnya yaitu robot pada posisi line dengan tikungan kedua yaitu robot akan berbelok ke kiri melalui garis lengkung, pada saat tersebut robot dapat berjalan dengan normal sesuai line yang dilewatinya.

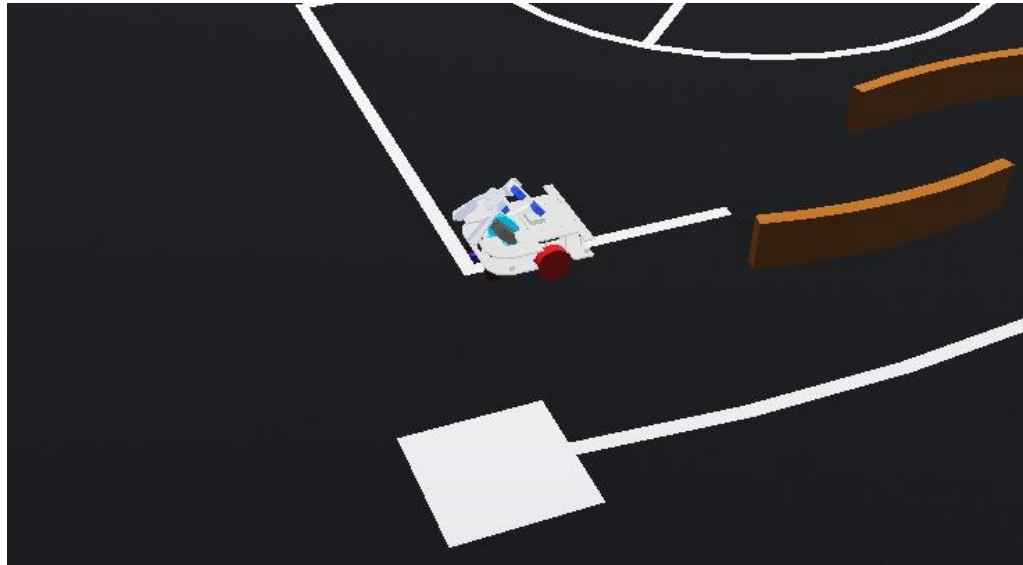


7. Setelah itu robot akan melewati garis putus yang pada samping kanan kirinya terdapat seperti tembok, pada kondisi tersebut dilalui robot dengan lancar yaitu berjalan dengan lurus seperti yang tampak pada gambar dibawah ini.

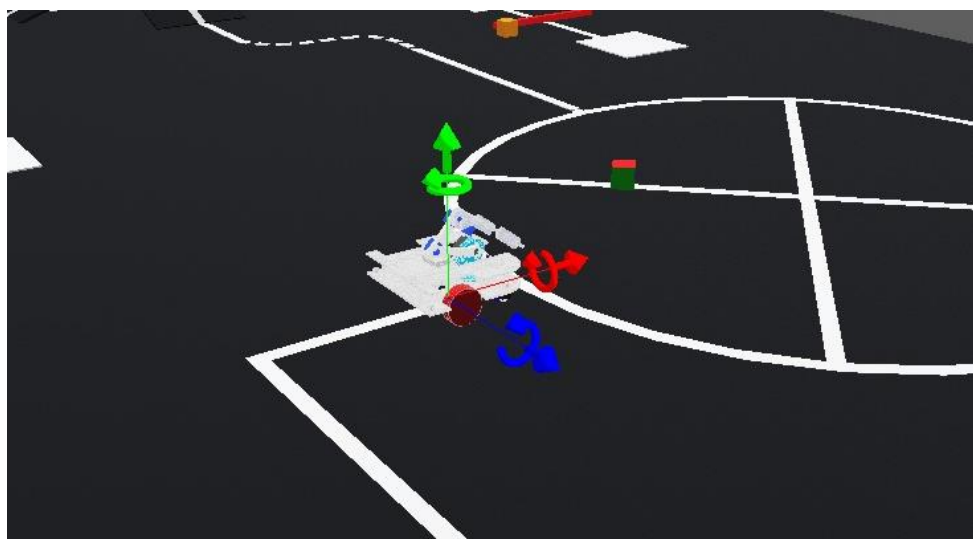


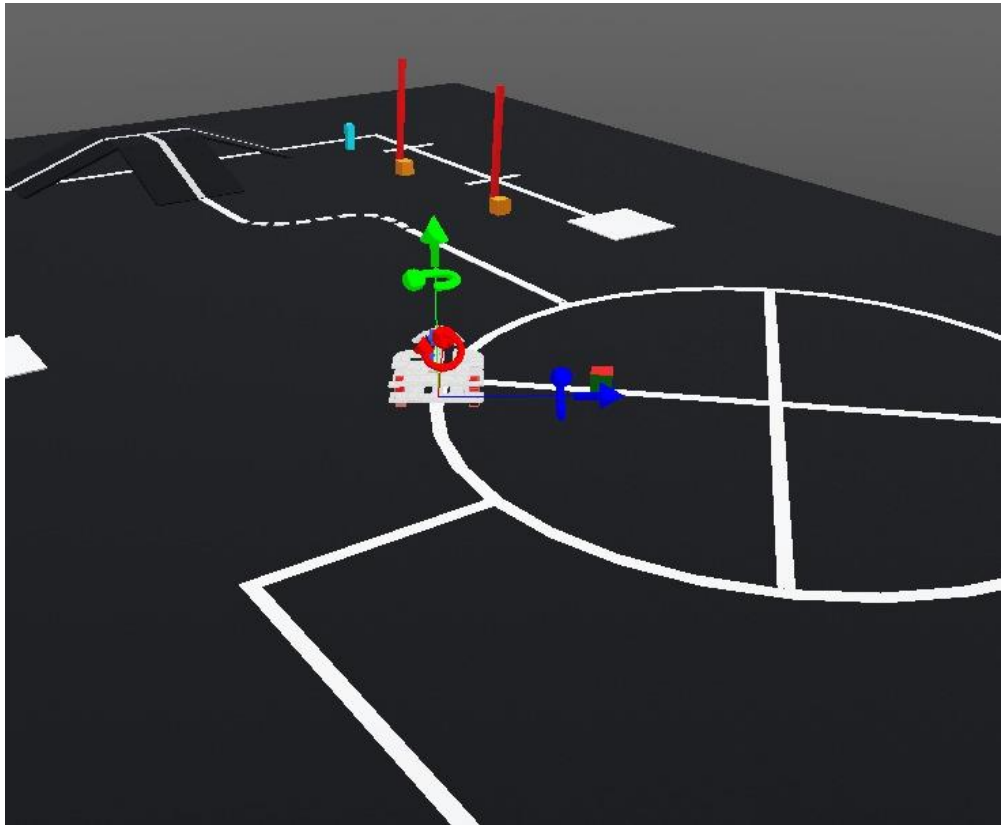
8. Kondisi dibawah ini ketika robot saat mendekati tikungan yang ketiga yang mana pada seharusnya robot akan berbelok ke kanan, namun terdapat kendala

berupa robot belum bisa mengenali medan sehingga robot tidak berbelok ke kanan dengan baik.

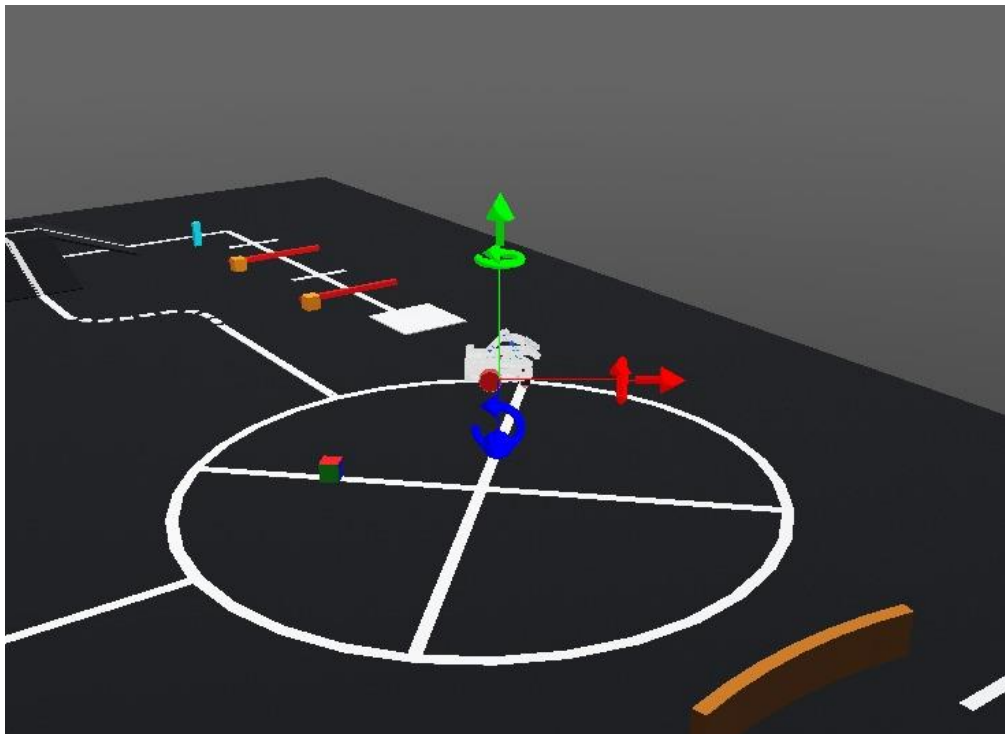


9. Setelah itu pada kondisi berikutnya robot akan memasuki arena atau medan berupa lingkaran, namun pada kondisi tersebut kita harus menggesernya secara manual terlebih dahulu dan seterusnya robot akan berjalan mengikuti line tersebut dengan baik, hal ini dikarenakan robot belum bisa berbelok dan mengikuti arena dengan otomatis maka kita perlu membelokkannya terlebih dahulu dan mengarahkan secara manual pada arena sebelum memasuki lingkaran.



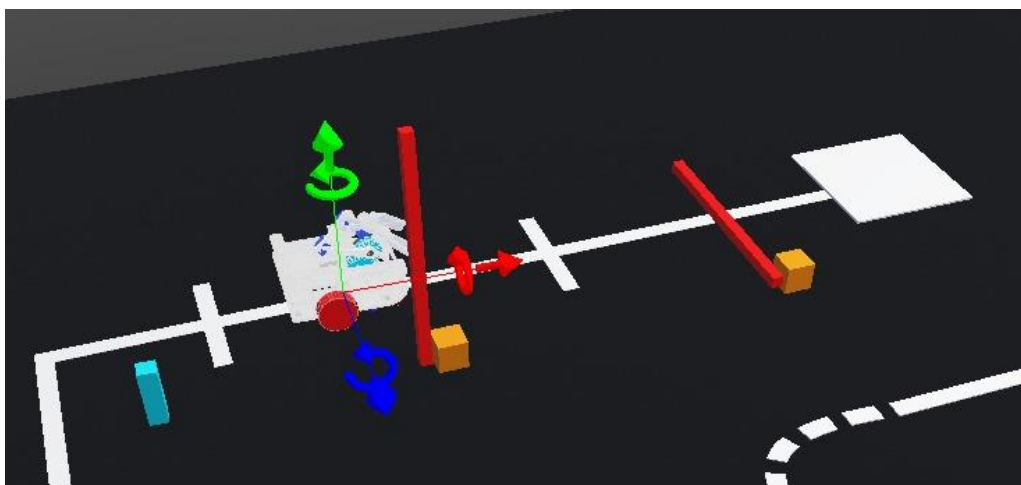
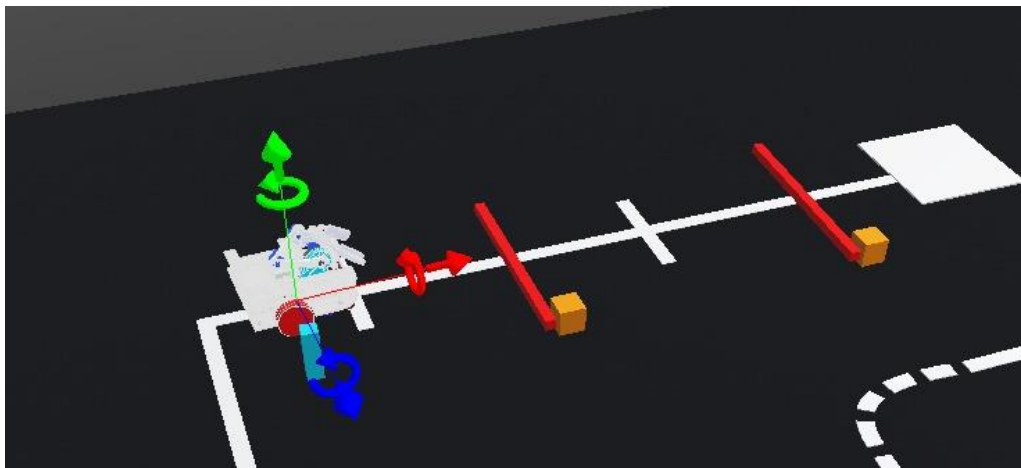
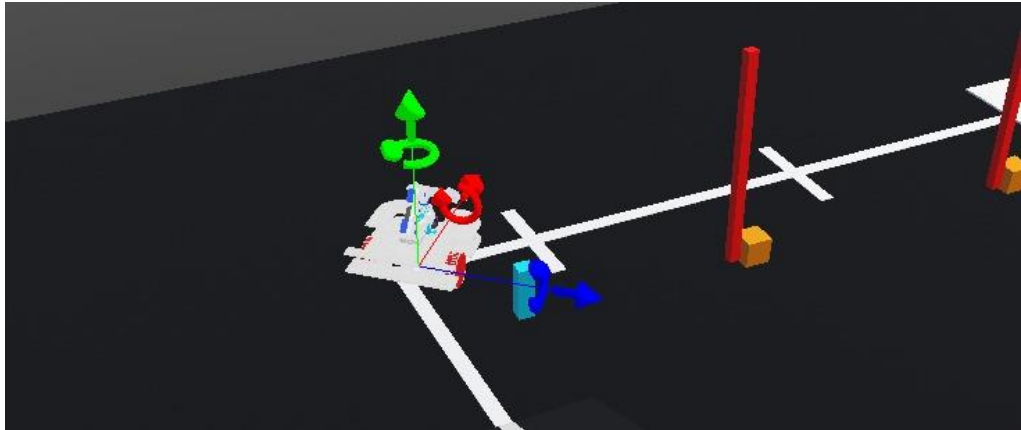


10. Namun terdapat kendala lagi yaitu robot akan terus berjalan berada pada sebuah lingkaran di arena tersebut.

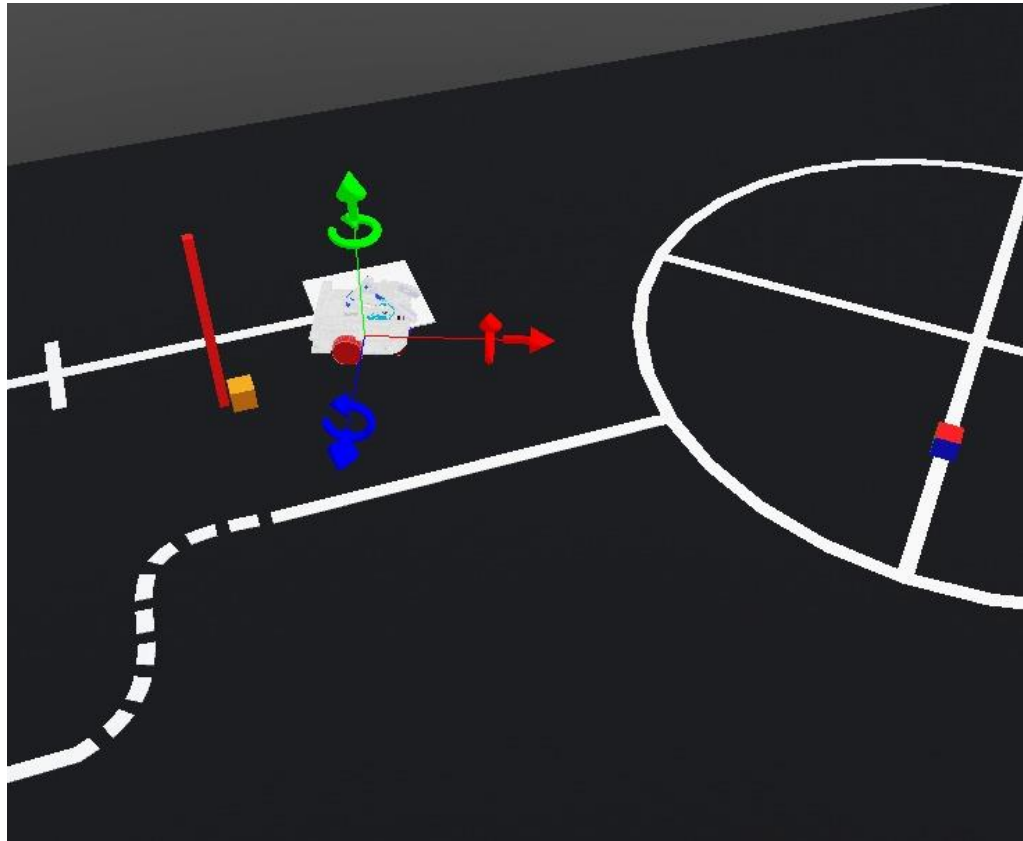


11. Kondisi berikutnya setelah kita arahkan secara manual dari sebelumnya arena lingkaran menuju pada arena berikutnya yaitu pada arena bagian yang terdapat palang seperti gambar dibawah ini, robot akan mengikuti atau berjalan dengan baik yaitu akan berbelok ke kanan.

Dan jika palang tertutup maka robot akan berhenti selanjutnya jika palang dalam kondisi terbuka maka robot akan kembali berjalan lurus ke depan.



12. Selanjutnya robot akan memasuki pada arena garis finish robot terdapat kendala yaitu robot tidak berhenti melainkan akan berputar – putar seperti yang tampak pada gambar simulasi dibawah ini.



#### I. Kesimpulan

Kesimpulan yang dapat ditarik dari hasil pembahasan sebelumnya adalah kita diharapkan dapat melakukan sebuah tantangan obstacle line followers dan sebuah arena yang telah tersedia, pada percobaan yang kami lakukan robot dapat berjalan hingga melewati tikungan pertama ke kiri dan tikungan kedua ke kiri robot dapat berjalan sesuai line, hingga melewati tembok yang bergaris terputus robot masih berjalan dengan baik, namun setelah itu robot tidak dapat berbelok ke kanan hingga kondisi yang dapat dilakukan secara manual, jadi robot yang kami jalankan dengan kelompok kami masih tidak dapat melewati keseluruhan obstaclenya.

**J. Daftar Pustaka**

- a. Buku referensi
- b. Website
- c. Artikel / jurnal
- d. Youtube
- e. Github

**K. Lampiran**

-