

A. Judul

Laporan Praktik Robotika tentang Mobile Robots untuk Pertemuan 5-6

B. Hari/Tanggal Praktikum

Praktikum dilakukan pada hari Jum'at, 8 Oktober 2021 dan laporan ini ditulis pada hari Minggu, 10 Oktober 2021.

C. Tujuan Praktikum

Adapun tujuan dari praktikum ini adalah:

- Melakukan pengamatan data pada sensor
- Menggunakan data yang dihasilkan dari sensor sebagai syarat untuk bergerak robot
- Membuat algoritma robot line follower dengan rintangan yang ada pada arena
- Menyelesaikan rute robot yang telah disediakan dari start hingga finish

D. Alat dan Bahan Praktikum

Alat dan Bahan praktikum adalah:

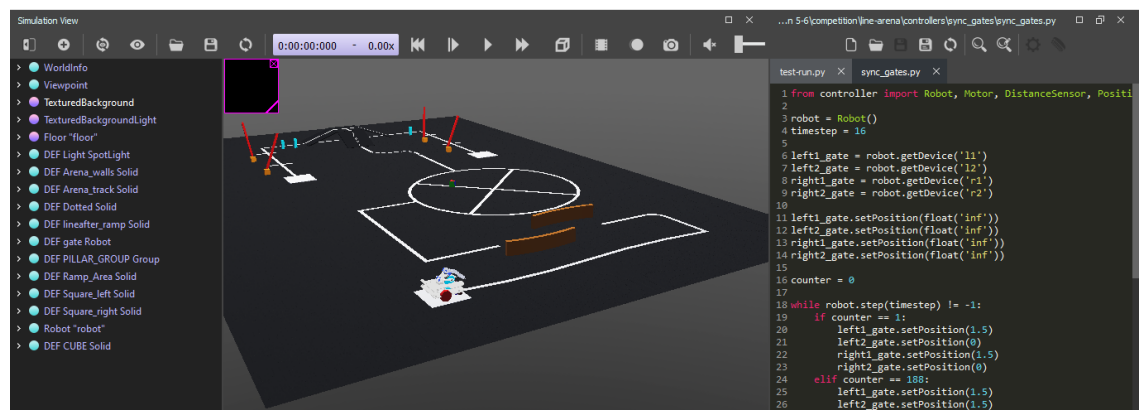
- Laptop (OS Windows)
- Webots
- <https://github.com/2black0/webots-laboratory>

E. Prosedur Kerja

Langkah kerja yang dilakukan adalah sebagai berikut:

- Mendownload / clone file webots-laboratory dari github berikut :
<https://github.com/2black0/webots-laboratory>
- Akses folder:
webots-laboratory\competition\line-arena\worlds\complete_robot.wbt
- Jika ingin mencoba program, ganti path controller robot menjadi test-run.py
- Akses sensor dan sebagainya untuk mendapatkan data yang dibutuhkan
- Buat algoritma agar robot bisa berjalan sesuai arena

F. Hasil Pengamatan



Gambar 1. Tampilan awal webots ketika membuka file complete_robots.wbt

```
1 from controller import Robot
2
3 tembok = 2
4
5 def maju():
6     rightmotor.setVelocity(4.5)
7     leftmotor.setVelocity(4.5)
8 def hari():
9     rightmotor.setVelocity(4)
10    leftmotor.setVelocity(-4)
11 def hanan():
12    rightmotor.setVelocity(-4)
13    leftmotor.setVelocity(4)
14 def diam():
15    rightmotor.setVelocity(0)
16    leftmotor.setVelocity(0)
```

Gambar 2. Deklarasi fungsi pergerakan robot, dan kondisi

```
17 def sebelum_tembok():
18     if garis_kanan > 1400 and garis_kiri < 1400:
19         hari()
20     elif garis < 1200:
21         maju()
22     elif garis_kanan < 1400 and garis_kiri > 1400:
23         hanan()
24     elif garis_kirii < 500:
25         diam()
26         hari()
27 def setelah_tembok():
28     if garis_kanan > 1400 and garis_kiri < 1400:
29         hari()
30     elif garis_kanan < 1400 and garis_kiri > 1400:
31         hanan()
32     elif garis_kanann < 500 :
33         hanan()
34     elif ping_depan > 200:
35         maju()
36     elif ping_depan < 200:
37         diam()
38     else :
39         maju()
40 def ditembok():
41     if (garis > 3600 and ping_kanan < 1000) or (garis > 3600 and ping_kiri < 1000):
42         maju()
```

Gambar 3. Algoritma pergerakan robot dari start-tembok kanan&kiri, dilanjutkan algoritma setelah melewati tembok hingga palang pintu, dan juga algoritma ketika robot diantara tembok

```
43 def palang():
44     if garis_kanann < 500 and garis_kirii > 600:
45         hanan()
46     elif garis_kanann < 500 and garis_kanann > 350 and garis_kirii < 600 and garis_kirii > 350 and ping_depan > 200:
47         maju()
48     elif ping_depan < 200:
49         diam()
50     elif garis_kanann < 350 and garis_kirii < 350:
51         diam()
52
```

Gambar 4. Algoritma pergerakan robot pada kondisi ketika setelah melewati box biru, maka akan masuk algoritma ini hingga finish

```
53 robot = Robot()
54
55 timestep = int(robot.getBasicTimeStep())
56
57 leftmotor = robot.getDevice('motor_1')
58 rightmotor = robot.getDevice('motor_2')
59 leftmotor.setPosition(float('inf'))
60 rightmotor.setPosition(float('inf'))
61
62 pingkiri = robot.getDevice('ds_left')
63 pingkanan = robot.getDevice('ds_right')
64 pingdepan = robot.getDevice('ds_front')
65
66
67 irl2 = robot.getDevice('IRL2')
68 irl1 = robot.getDevice('IRL1')
69 irl1 = robot.getDevice('IRCL')
70 irl2 = robot.getDevice('IRCR')
71 irr1 = robot.getDevice('IRR1')
72 irr2 = robot.getDevice('IRR2')
73
74 irl2.enable(timestep)
75 irl1.enable(timestep)
76 irl1.enable(timestep)
77 irl2.enable(timestep)
78 irr1.enable(timestep)
79 irr2.enable(timestep)
80 pingkiri.enable(timestep)
81 pingkanan.enable(timestep)
82 pingdepan.enable(timestep)
83
```

Gambar 5. Akses sensor sensor yang terdapat pada robot

```
84 while robot.step(timestep) != -1:
85     rightmotor.setVelocity(10)
86     leftmotor.setVelocity(10)
87
88
89     irl2_val = irl2.getValue()
90     irl1_val = irl1.getValue()
91     irl1_val = irl1.getValue()
92     irl2_val = irl2.getValue()
93     irr1_val = irr1.getValue()
94     irr2_val = irr2.getValue()
95     ping_kanan = pingkanan.getValue()
96     ping_kiri = pingkiri.getValue()
97     ping_depan = pingdepan.getValue()
98
99     garis_kanan = irr1_val + irr2_val
100    garis_kiri = irl1_val + irl2_val
101    garis_kanan = irl2_val + irr1_val + irr2_val
102    garis_kiri = irl1_val + irl1_val + irl2_val
103    garis_tengah = irl2_val + irl1_val
104    garis = garis_kanan + garis_kiri
105
106
```

Gambar 6. Program While yang dijalankan di robot

```
107     print('{:.2f} {:.2f} {:.2f} {:.2f} {:.2f} {:.2f}'.format
108
109
110     if tembok == 2:
111         sebelum_tembok()
112         ditembok()
113         if garis < 3000 and ping_kiri < 1000:
114             tembok = 3
115     elif tembok == 3:
116         setelah_tembok()
117         if ping_kanan < 900:
118             tembok = 4
119     elif tembok == 4:
120         palang()
```

Gambar 7. Program utama pada while yang menjalankan robot dari start hingga finish

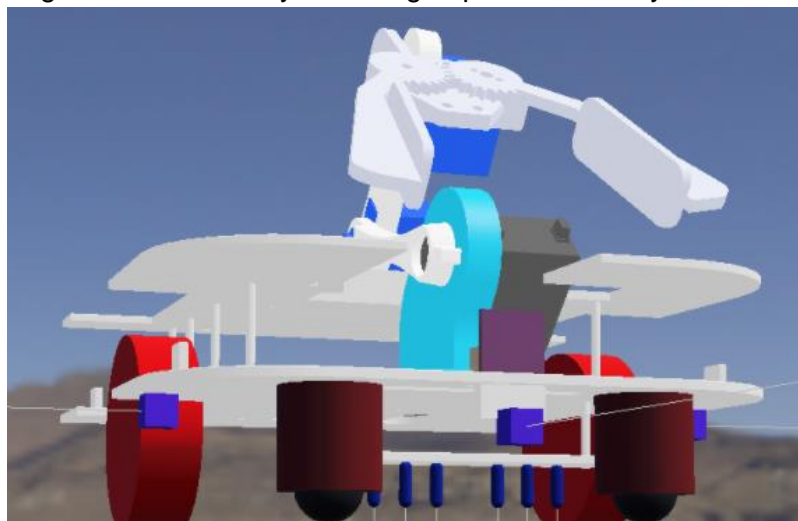
G. Pembahasan / Analisa Data

a. Percobaan file test-run

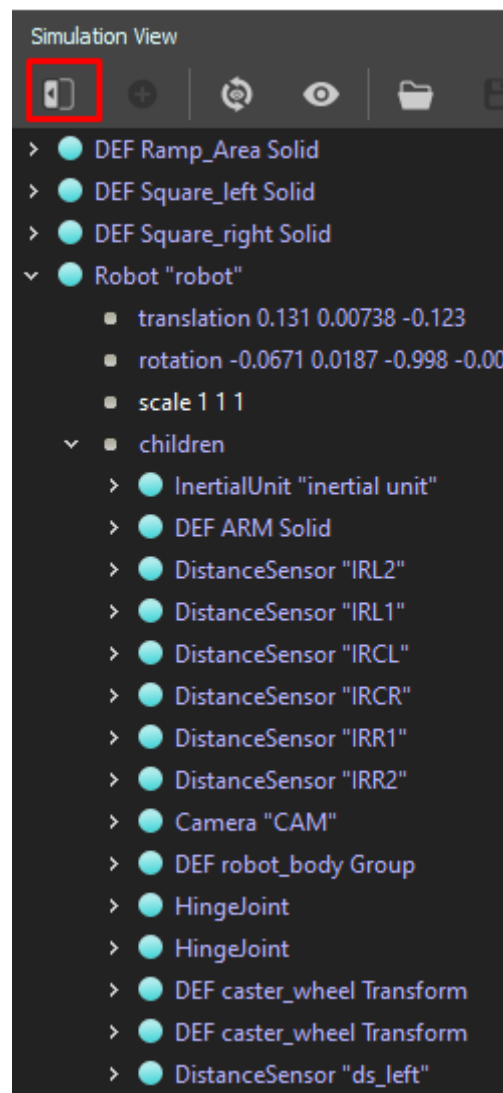
Setelah berhasil membuka file webots tersebut, kemudian cobalah untuk running terlebih dahulu simulasi webotsnya apakah terdapat error atau tidak, jika simulasi berjalan lancar, langkah selanjutnya adalah cek hasil simulasi yang berjalan, apakah sesuai dengan program atau tidak, jika sesuai maka langkah selanjutnya adalah mencoba memahami cara kerja dari robotnya.

b. Memahami robot

Sebelum berlanjut ke pembuatan algoritma, alangkah baiknya kita mengetahui dulu sensor apa yang terdapat pada robot tersebut, pada robot kali ini terdapat 6 IR sensor, 4 Distance Sensor, 2 wheel, Camera, actuator / arm / grip, dan motor. Kemudian kita harus bisa mengetahui posisi dari masing masing sensor tersebut, dengan klik dropdown menu robot pada bagian kiri webots, kemudian nanti disitu sudah ada penjelasan mengenai setiap sensor, klik sensor yang akan dilihat lalu zoom in untuk melihat lebih dekat kepada robotnya agar lebih detil dan yakin dengan posisi sensornya.



Gambar 8. View robot secara utuh



Gambar 9. Menu scene tree sidebar untuk melihat isi / detail dari isi webots yang kita buka, mulai dari world, object, dan bagian bagian robot (tekan tanda merah untuk memunculkan / menyembunyikan menu ini)

c. Pengamatan data sensor

Setelah mengetahui sensor apa saja yang terdapat pada robot, kemudian kita coba akses sensornya (gambar 5) dan coba tampilkan datanya dengan getvalue (gambar 6), agar nantinya bisa digunakan untuk membuat algoritma yang diinginkan. Kemudian jalankan program dan coba amati pergerakan data sensornya.

Console - All			
1184.69	366.68	1182.05	1368.12
1193.48	369.43	1190.89	1378.27
1202.58	372.28	1200.02	1388.80
1210.26	374.68	1207.72	1397.68
1215.05	376.18	1212.55	1403.22
1216.13	376.53	1213.70	1404.47
1213.47	375.72	1211.17	1401.40

Gambar 10. Contoh data sensor yang ditampilkan ketika robot berjalan

Pengamatan yang dasar meliputi kondisi sensor IR ketika mendapatkan garis atau tidak (kondisi diatas warna putih atau warna hitam) nanti dilihat angkanya berapa, kemudian kondisi sensor distance ketika terdapat objek didepan sensornya, dari data yang saya amati, kondisi ketika sensor IR membaca garis yaitu angkanya berkisar antara 186 – 190, dan ketika tidak mendapatkan garis, maka angkanya diatas 600. Kemudian, untuk sensor distance, ketika terdapat objek pada jangkauannya, maka sensor akan bernilai < 1000 , dan ketika loss atau tidak ada objek di jangkauan sensornya, maka nilainya ≥ 1000 .

d. Membuat Algoritma dasar

Setelah mengamati data yang didapatkan sebelumnya, selanjutnya mencoba membuat program lanjutan untuk mengikuti garis terlebih dahulu, disini saya katakan kondisi start hingga sebelum tembok. Program yang digunakan terdapat pada : (gambar 3, line 17 - 26). Algoritma selanjutnya yaitu ketika kondisi robot sudah melewati tembok dan hendak memasuki rintangan lingkaran, saya membuat program pada : (gambar 3, line 27 – 39). Untuk program pada saat diantara 2 tembok (kondisi tanpa garis), saya menggunakan sensor distance sebagai bantuan untuk berjalan diantara sisi tersebut, programnya terdapat pada : (gambar 3, line 40 – 42).

Sebelumnya, disini saya memisahkan beberapa kondisi dengan cara membuat def untuk masing masing kondisi, dikarenakan ketika program tersebut dijalankan pada while semua, maka nanti dikhawatirkan akan terjadi tabrakan atau bertumpuknya data yang mengakibatkan program tidak berjalan dengan optimal, juga disini lain agar mempermudah pembacaan algoritmanya.

e. Membuat algoritma lanjutan

Setelah program sebelumnya berjalan, langkah selanjutnya adalah melewati rintangan putus putus setelah berhasil melewati lingkaran, program yang digunakan terdapat pada gambar 3. Kondisi selanjutnya terjadi ketika $ds_right < 1000$ maka akan memasuki algoritma palang (gambar 4), algoritma tersebut dimulai setelah melewati box biru sebelah kanan hingga finish.

H. Kesimpulan

Praktikum kali ini secara keseluruhan dimulai dari pengaksesan sensor terlebih dahulu, kemudian dilakukan pengamatan dan pengambilan data pada setiap sensor, selanjutnya baru dibuat algoritmanya. Saya menggunakan 3 kondisi, dimana kondisi tembok == 2 adalah kondisi dari start hingga rintangan tembok, kemudian kondisi tembok == 3 adalah kondisi setelah melewati tembok, melewati lingkaran, kemudian tanjakan, hingga turun dan sensor kanan mendapatkan box biru, lalu kondisi tembok == 4 adalah kondisi setelah melewati box biru (belok kanan) hingga melewati palang 2 buah kemudian berakhir pada finish. Kekurangan praktikum kali ini adalah saya belum bisa mengakses arm dan cam pada robot, sehingga objek ditengah lingkaran yang seharusnya dibawa oleh robot, tidak saya bawa dikarenakan tidak bisa akses arm dan cam itu sendiri. Kemudian untuk max speed / velocity yang disarankan maksimal 4.5, saya sudah mencoba diatas itu ternyata robotnya error pada saat

melewati tanjakan, dan juga pembacaan sensor menjadi kurang akurat, alhasil bisa mempengaruhi kinerja robot.

I. Daftar Pustaka

- a. <https://github.com/2black0/webots-laboratory>

J. Lampiran

```
1 from controller import Robot
2
3 robot = Robot()
4 timestep = int(robot.getBasicTimeStep())
5
6 leftmotor = robot.getDevice('motor_1')
7 rightmotor = robot.getDevice('motor_2')
8 leftmotor.setPosition(float('inf'))
9 rightmotor.setPosition(float('inf'))
10
11 ir12 = robot.getDevice('IRL2')
12 ir11 = robot.getDevice('IRL1')
13 ircl = robot.getDevice('IRCL')
14 ircr = robot.getDevice('IRCR')
15 irr1 = robot.getDevice('IRR1')
16 irr2 = robot.getDevice('IRR2')
17
18 ir12.enable(timestep)
19 ir11.enable(timestep)
20 ircl.enable(timestep)
21 ircr.enable(timestep)
22 irr1.enable(timestep)
23 irr2.enable(timestep)
24
25 while robot.step(timestep) != -1:
26     rightmotor.setVelocity(1)
27     leftmotor.setVelocity(1)
28
29     ir12_val = ir12.getValue()
30     ir11_val = ir11.getValue()
31     ircl_val = ircl.getValue()
32     ircr_val = ircr.getValue()
33     irr1_val = irr1.getValue()
34     irr2_val = irr2.getValue()
35
36     print('{:.2f} {:.2f} {:.2f} {:.2f} {:.2f} {:.2f}'.format(ir12_val, ir11_val, ircl_val, ircr_val, irr1_val, irr2_val))
```

File test-run.py original