

**LAPORAN SIMULASI
PICK AND PLACE COMPETITION MENGGUNAKAN WEBOTS**



Oleh:

Irman Prayista/1103210094

**PRODI S1 TEKNIK KOMPUTER
FAKULTAS TEKNIK ELEKTRO
UNIVERSITAS TELKOM
BANDUNG**

2024

DAFTAR ISI

DAFTAR ISI.....	i
DAFTAR GAMBAR	ii
BAB I PENDAHULUAN	3
1.1. Profil Perusahaan	3
BAB II PERSIAPAN	4
2.1. Alat dan Perangkat Lunak yang Diperlukan	4
2.2. Langkah Persiapan	4
2.2.1. Instalasi Webots	4
2.2.2. Clone Repositori Simulasi	4
BAB III LANGKAH IMPLEMENTASI.....	6
3.1. Membuka Simulasi di Webots	6
3.2. Menjalankan Simulasi.....	6
BAB IV HASIL	8
4.1. Source Code dan Output	8
BAB V KESIMPULAN.....	13
5.1. Kesimpulan	13
DAFTAR PUSTAKA	14

DAFTAR GAMBAR

Gambar 1.1.1 Pick and Place Competition.....	3
Gambar 2.2.1.1 Webots.....	4
Gambar 2.2.2.1 Command prompt	4
Gambar 3.1.1 Tampilan Webots.....	6
Gambar 3.2.1 Tampilan file world pick_and_place_competition	6
Gambar 3.2.2 Simulasi robot mengambil objek	7
Gambar 3.2.3 Simulasi robot memindahkan objek	7
Gambar 4.1.1 Output pada Webots	11

BAB I

PENDAHULUAN

1.1. Profil Perusahaan

Simulasi *Pick and Place Competition* merupakan pendekatan yang efektif untuk mempelajari dan mengembangkan teknologi robotika tanpa memerlukan perangkat keras fisik. Simulasi ini berfokus pada kemampuan robot untuk secara otomatis memindahkan objek dari satu lokasi ke lokasi lain. *Webots*, sebagai simulator *open-source*, menawarkan lingkungan yang ideal untuk merancang, menguji, dan menyempurnakan sistem robot.

Tujuan utama simulasi ini adalah menyediakan panduan terstruktur bagi pengguna dari berbagai tingkat keahlian, baik pemula maupun ahli, untuk menjalankan simulasi sekaligus memahami konsep pemrograman robot. Dengan demikian, pengguna diharapkan dapat menyesuaikan logika robot sesuai kebutuhan [1].



Gambar 1.1.1 Pick and Place Competition

BAB II

PERSIAPAN

2.1. Alat dan Perangkat Lunak yang Diperlukan

1. Komputer/Laptop dengan spesifikasi yang mendukung Webots.
2. Aplikasi Webots, simulator yang dapat diunduh dari <https://cyberbotics.com>.
3. Akses Internet untuk mengunduh repositori dari <https://github.com/cyberbotics/pick-and-place-competition>.

2.2. Langkah Persiapan

2.2.1. Instalasi Webots

1. Kunjungi situs [Webots](https://cyberbotics.com).

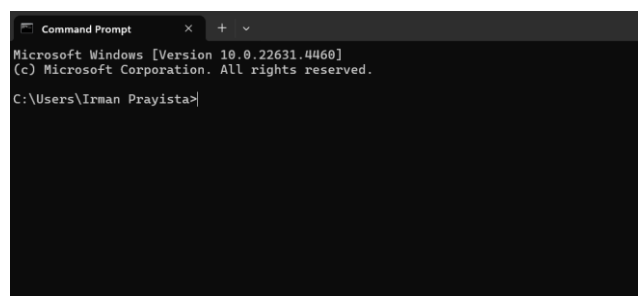


Gambar 2.2.1.1 Webots

2. Unduh aplikasi sesuai sistem operasi (Windows, Linux, atau macOS).
3. Ikuti panduan instalasi hingga selesai.

2.2.2. Clone Repositori Simulasi

1. Buka terminal/command prompt.



Gambar 2.2.2.1 Command prompt

2. Jalankan perintah berikut:

```
git clone https://github.com/cyberbotics/pick-and-place-competition.git
```

3. Pindah ke direktori repositori yang diunduh:

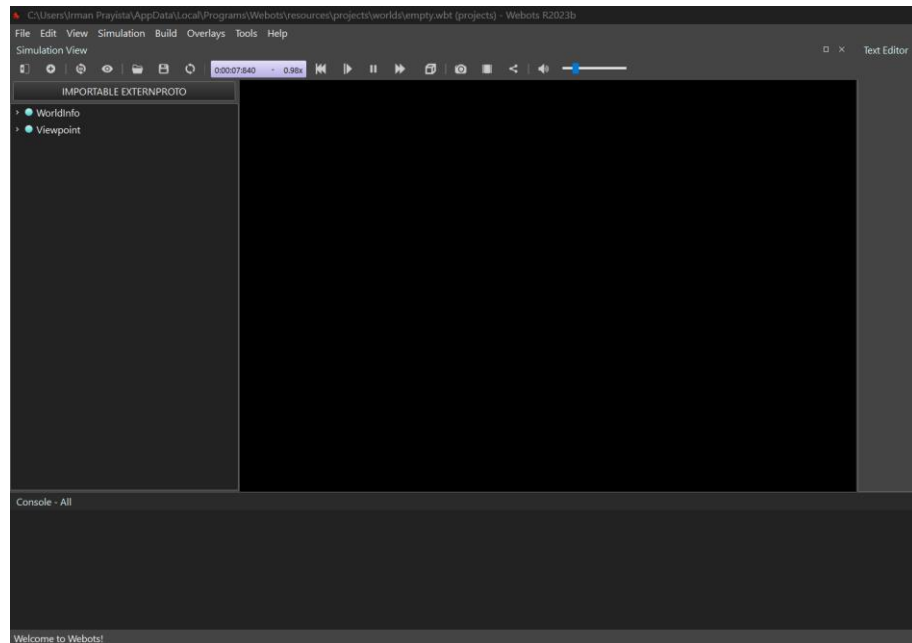
```
cd pick-and-place-competition
```

BAB III

LANGKAH IMPLEMENTASI

3.1. Membuka Simulasi di Webots

1. Buka aplikasi Webots.
2. Klik menu **File**, lalu pilih **Open Project**.

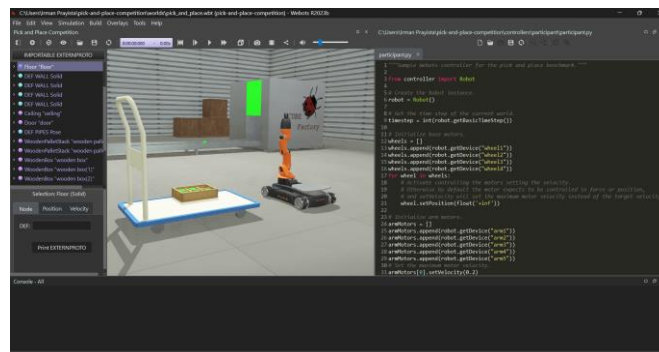


Gambar 3.1.1 Tampilan Webots

3. Arahkan ke folder repositori pada direktori:
pick-and-place-competition/worlds/pick_and_place.wbt
4. Pilih file ***pick_and_place.wbt*** untuk membuka simulasi.

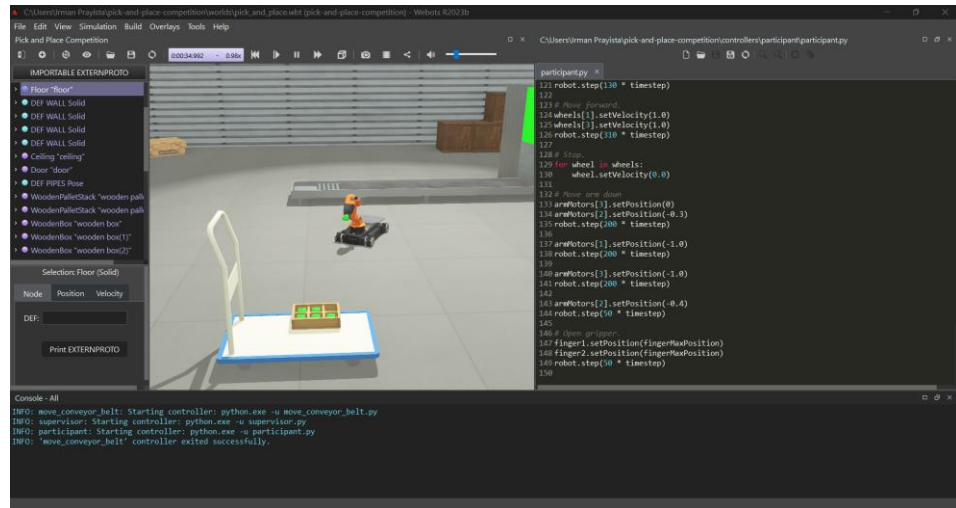
3.2. Menjalankan Simulasi

1. Setelah *Webots* terbuka, pastikan semua elemen, seperti robot dan objek terlihat.

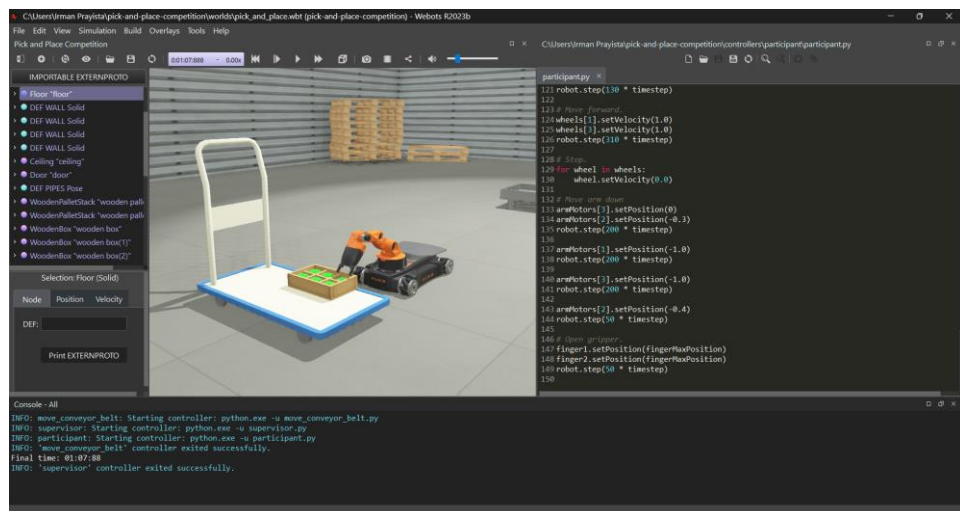


Gambar 3.2.1 Tampilan file world pick_and_place_competition

2. Klik tombol **Play** di bagian atas layar untuk memulai simulasi.
3. Amati jalannya robot saat simulasi.



Gambar 3.2.2 Simulasi robot mengambil objek



Gambar 3.2.3 Simulasi robot memindahkan objek

BAB IV

HASIL

4.1. Source Code dan Output

```
5. from controller import Robot
6.
7. # Create the Robot instance.
8. robot = Robot()
9.
10. # Get the time step of the current world.
11. timestep = int(robot.getBasicTimeStep())
12.
13. # Initialize base motors.
14. wheels = []
15. wheels.append(robot.getDevice("wheel1"))
16. wheels.append(robot.getDevice("wheel2"))
17. wheels.append(robot.getDevice("wheel3"))
18. wheels.append(robot.getDevice("wheel4"))
19. for wheel in wheels:
20.     # Activate controlling the motors setting the
       velocity.
21.     # Otherwise by default the motor expects to be
       controlled in force or position,
22.     # and setVelocity will set the maximum motor velocity
       instead of the target velocity.
23.     wheel.setPosition(float('+inf'))
24.
25. # Initialize arm motors.
26. armMotors = []
27. armMotors.append(robot.getDevice("arm1"))
28. armMotors.append(robot.getDevice("arm2"))
29. armMotors.append(robot.getDevice("arm3"))
30. armMotors.append(robot.getDevice("arm4"))
31. armMotors.append(robot.getDevice("arm5"))
32. # Set the maximum motor velocity.
33. armMotors[0].setVelocity(0.2)
34. armMotors[1].setVelocity(0.5)
35. armMotors[2].setVelocity(0.5)
36. armMotors[3].setVelocity(0.3)
37.
38. # Initialize arm position sensors.
39. # These sensors can be used to get the current joint
       position and monitor the joint movements.
40. armPositionSensors = []
41. armPositionSensors.append(robot.getDevice("arm1sensor"))
```

```

42.armPositionSensors.append(robot.getDevice("arm2sensor"))
43.armPositionSensors.append(robot.getDevice("arm3sensor"))
44.armPositionSensors.append(robot.getDevice("arm4sensor"))
45.armPositionSensors.append(robot.getDevice("arm5sensor"))
46.for sensor in armPositionSensors:
47.    sensor.enable(timestep)
48.
49.# Initialize gripper motors.
50.finger1 = robot.getDevice("finger1")
51.finger2 = robot.getDevice("finger2")
52.# Set the maximum motor velocity.
53.finger1.setVelocity(0.03)
54.finger2.setVelocity(0.03)
55.# Read the minium and maximum position of the gripper
    motors.
56.fingerMinPosition = finger1.getMinPosition()
57.fingerMaxPosition = finger1.getMaxPosition()
58.
59.# Move forward.
60.for wheel in wheels:
61.    wheel.setVelocity(7.0)
62.# Wait until the robot is in front of the box.
63.robot.step(520 * timestep)
64.
65.# Stop moving forward.
66.for wheel in wheels:
67.    wheel.setVelocity(0.0)
68.
69.# Move arm and open gripper.
70.armMotors[1].setPosition(-0.55)
71.armMotors[2].setPosition(-0.9)
72.armMotors[3].setPosition(-1.5)
73.finger1.setPosition(fingerMaxPosition)
74.finger2.setPosition(fingerMaxPosition)
75.
76.# Monitor the arm joint position to detect when the
    motion is completed.
77.while robot.step(timestep) != -1:
78.    if abs(armPositionSensors[3].getValue() - (-1.2)) <
        0.01:
79.        # Motion completed.
80.        break
81.
82.# Close gripper.
83.finger1.setPosition(0.013)
84.finger2.setPosition(0.013)
85.# Wait until the gripper is closed.

```

```
86. robot.step(50 * timestep)
87.
88. # Lift arm.
89. armMotors[1].setPosition(0)
90. # Wait until the arm is lifted.
91. robot.step(200 * timestep)
92.
93. # Rotate the robot.
94. wheels[0].setVelocity(2.5)
95. wheels[1].setVelocity(-2.5)
96. wheels[2].setVelocity(2.5)
97. wheels[3].setVelocity(-2.5)
98. # Wait for a fixed amount to step that the robot rotates.
99. robot.step(690 * timestep)
100.
101.     # Move forward.
102.     wheels[1].setVelocity(2.5)
103.     wheels[3].setVelocity(2.5)
104.     robot.step(900 * timestep)
105.
106.     # Rotate the robot.
107.     wheels[0].setVelocity(1.0)
108.     wheels[1].setVelocity(-1.0)
109.     wheels[2].setVelocity(1.0)
110.     wheels[3].setVelocity(-1.0)
111.     robot.step(200 * timestep)
112.
113.     # Move forward.
114.     wheels[1].setVelocity(1.0)
115.     wheels[3].setVelocity(1.0)
116.     robot.step(300 * timestep)
117.
118.     # Rotate the robot.
119.     wheels[0].setVelocity(1.0)
120.     wheels[1].setVelocity(-1.0)
121.     wheels[2].setVelocity(1.0)
122.     wheels[3].setVelocity(-1.0)
123.     robot.step(130 * timestep)
124.
125.     # Move forward.
126.     wheels[1].setVelocity(1.0)
127.     wheels[3].setVelocity(1.0)
128.     robot.step(310 * timestep)
129.
130.     # Stop.
131.     for wheel in wheels:
132.         wheel.setVelocity(0.0)
```

```

133.
134.     # Move arm down
135.     armMotors[3].setPosition(0)
136.     armMotors[2].setPosition(-0.3)
137.     robot.step(200 * timestep)
138.
139.     armMotors[1].setPosition(-1.0)
140.     robot.step(200 * timestep)
141.
142.     armMotors[3].setPosition(-1.0)
143.     robot.step(200 * timestep)
144.
145.     armMotors[2].setPosition(-0.4)
146.     robot.step(50 * timestep)
147.
148.     # Open gripper.
149.     finger1.setPosition(fingerMaxPosition)
150.     finger2.setPosition(fingerMaxPosition)
151.     robot.step(50 * timestep)

```

Kode ini adalah program untuk mengendalikan robot dalam simulasi *Pick and Place* di *Webots*. Robot diatur untuk bergerak, mengontrol lengan, dan menggunakan *gripper* untuk mengambil serta meletakkan objek. Pertama, roda diatur dalam mode kecepatan, sementara lengan dan *gripper* dikonfigurasi untuk bergerak dengan kecepatan dan posisi tertentu. Sensor pada lengan digunakan untuk memantau pergerakan saat mengambil objek. Robot bergerak mendekati objek, berhenti, lalu menggerakkan lengan dan membuka *gripper* untuk mengambilnya. Setelah objek berhasil diambil, lengan diangkat, dan robot bergerak serta berputar menuju lokasi tujuan. Di lokasi tujuan, robot menurunkan lengan dan membuka *gripper* untuk meletakkan objek. Semua proses ini dijalankan secara terkoordinasi menggunakan fungsi **robot.step()**, sehingga sinkron dengan simulasi.

```

Console - All
INFO: move_conveyor_belt: Starting controller: python.exe -u move_conveyor_belt.py
INFO: supervisor: Starting controller: python.exe -u supervisor.py
INFO: participant: Starting controller: python.exe -u participant.py
INFO: 'move_conveyor_belt' controller exited successfully.
Final time: 01:07:84
INFO: 'supervisor' controller exited successfully.

```

Gambar 4.1.1 Output pada *Webots*

Log ini menunjukkan bahwa tiga program berhasil dijalankan dalam simulasi, yaitu *move_conveyor_belt*, *supervisor*, dan *participant*. Masing-masing

program menggunakan skrip *Python* bernama `move_conveyor_belt.py`, `supervisor.py`, dan `participant.py`. Program *move_conveyor_belt* selesai berjalan tanpa masalah, diikuti oleh supervisor yang juga berhenti dengan sukses. Simulasi selesai dalam waktu 1 menit, 7 detik, dan 84 milidetik. Semua proses berjalan lancar tanpa ada *error*.

BAB V

KESIMPULAN

5.1. Kesimpulan

Simulasi ini dirancang untuk membantu pengguna memahami dasar pemrograman robot dan operasional simulasi *Pick and Place Competition*. Dengan mengikuti langkah-langkah yang telah dijelaskan, manfaat yang pengguna dapatkan, yaitu:

1. Memahami logika sistem robot.
2. Memhamai proses menjalankan simulasi secara efektif.
3. Mengembangkan proyek robot lebih lanjut, baik dalam simulasi maupun aplikasi nyata.

DAFTAR PUSTAKA

- [1] Cyberbotics. (n.d.). *Pick and place competition*. GitHub. Retrieved November 16, 2024, from <https://github.com/cyberbotics/pick-and-place-competition>