Institute of Research and Publication Indonesia
IJEERE: Indonesian Journal of Electrical Engineering and Renewable Energy
Journal Homepage: https://journal.irpi.or.id/index.php/ijeere
Vol. 2 Iss. 1 Juni 2022, pp: 07-18
P-ISSN: 2797-1155 E-ISSN: 2797-0868

Simulation Understanding Line Follower Robot C Program with Webots

Simulasi Pemahaman Robot Pengikut Garis Program C dengan Webots

Aldo Lorenza[1], Bayu Hidayat[2], Lisana Sidka Alia[3], Luqyana Aufa Nurachanta[4], Iskariman Halmahera[5], M.Sonny Irsan Syahputra[6]

[1,2,3,4,5,6]Teknik Elektro, UIN Sultan Syarif Kasim, Indonesia

Corresponden E-Mail : [3]11950525128@students.uin-suska.ac.id

Makalah: Diterima 10 Juli 2021; Diperbaiki 25 Mei 2022; Disetujui 1 Juni 2022
Corresponding Author: 11950525128@students.uin-suska.ac.id

Abstrak

Robot pengikut garis merupakan salah satu jenis robot yang berkembang dalam dunia teknologi dengan sistem robot yang dapat mengikuti garis yang terdiri dari rangkaian komponen elektronika untuk mendapatkan respon dan kecepatan ideal. Robot pengikut garis dilengkapi dengan sensor yang akan berpengaruh terhadap roda yang digerakkan dengan motor dimana pengaturan kecepatan dipengaruhi batas limit kecepatan dan gesek antara roda robot dengan lantai.
Robot pengikut garis terdiri dari sistem software dan hardware. Dalam memahami cara kerja robot pengikut garis dapat digunakan aplikasi simulasi *Webots* yang merupakan aplikasi *open source* yang dapat digunakan oleh para pemula. Simulasi robot pengikut garis dimulai dengan menggunakan file robot *e-puck* yang sudah tertera dalam *Webots* dengan menginput program bahasa yang digunakan yaitu Bahasa C ke mikrokontroller robot. Dari hasil simulasi robot pengikut garis menggunakan *webot*s didapatkan sistem mampu mengidentifikasi jalur lurus dan jalur belok. Sistem mampu memberikan respon dan kecepatan ideal dengan input dari sensor membaca garis yang diproses dengan sistem control Bahasa C dan hasil output robot bergerak maksimal.

Keyword : Pemahaman, Program C, Robot Pengikut Garis, Simulasi, *Webot*s

Abstrack

The line follower robot is one type of robot that develops in the world of technology with a robot system that can follow a line composed of a series of electronic components to get the ideal response and speed. The line follower robot is equipped with sensors that will affect the wheel that is driven by a motor, where the speed is very dependent on the speed limit and friction between the tire robot with the floor.
Line follower robot consists of software and hardware systems. In understanding how the line follower robot works use the Webots simulation application as an open source application that can be used by beginners. The line follower robot simulation begins by using the e-puck robot file that has been listed in Webots by inputting the language program used, namely C language, into the robot microcontroller.From the simulation result of line follower robot using Webots were obtained system is capable of identifying a straight line and turn lanes. The system is able to provide an ideal response and speed with input from the line reading sensor which is processed with the C language control system and outputs the result as the robot move maximum.

Keyword : *Understanding, C Program, Line Follower Robot, Simulation, Webots*

1. Pendahuluan

Teknologi robotik sudah banyak digunakan dalam perkembangan sejarah dunia. Robot dapat dijelaskan sebagai piranti mekanik yang mampu melakukan pekerjaan manusia dengan sistem kecerdasan buatan yang ditanamkan dalam sistem baik secara hardware maupun software. Dalam perkembangan nya robot digunakan dalam teknologi industri yang memiliki nilai efektifitas dan efisiensi yang lebih baik serta lebih cepat. Awal dasar dirancangnya robot oleh manusia untuk membantu manusia dalam melakukan pekerjaan yang memiliki ketelitian tinggi, resiko besar, dan tenaga yang besar.

Salah satu perkembangan robot yang banyak digunakan dalam perkembangan teknologi yaitu robot pengikut garis. Robot pengikut garis merupakan jenis robot yang mendeteksi garis dengan cara mengukur intensitas cahaya yang dipantulkan dari permukaan yang dilaluinnya. Robot pengikut garis (line follower) telah banyak digunakan dalam industry besar yang ada di dunia karena efektifitas dan efisiensi yang dihasilkan.

Secara analog *robot line follower* terdiri dari rangkaian analog untuk melacak garis serta tidak dapat diprogramkan sesuai kebutuhan karena tidak terdapat mikrokontroller didalam nya. Adapun yang menggunakan mikrokontroller tidak dilengkapi perangkat chip program dan bootloader. Dalam mempermudah

pemahaman terkait *robot line follower* bisa dilakukan simulasi dari sistem kerja *robot line follower* dengan simulasi aplikasi robot. Salah satu aplikasi robot yang mudah digunakan dan open source yaitu aplikasi *Webot*s yang sudah memiliki library dari beberapa program robot yang sudah pernah disimulasikan.

Adapun dalam simulasi pemahaman *robot line follower* dengan *Webot*s dapat disimulasikan dengan jenis robot *e-puck* yang merupakan jenis robot teknik edukasi untuk memahami beberapa sistem dasar dalam robotika. Dalam menjalankan sistem simulasi pemahaman robot *e-puck* sebagai sarana pemahaman perlu program bahasa control dalam mikrokontroller sistem dari *e-puck* yaitu bahasa program C sebagai contoh dalam memahami karakteristik *robot line follower*. Dalam penerapan kehidupan nyata *robot line follower* memiliki sistem yang dapat menerima respon dari sensor garis dan memberikan output berupa kecepatan ideal terhadap roda robot serta mampu mengidentifikasi jalur lurus dan jalur belok. Dari pembahasan ini tujuan yang akan dicapai adalah pemahaman terkait rancangan sistem kerja robot pengikut garis dengan program C yang ditanamkan pada robot melalui simulasi robot *Webot*s dengan sensor garis yang memberikan respon terkait input dari sensor garis yang diproses.

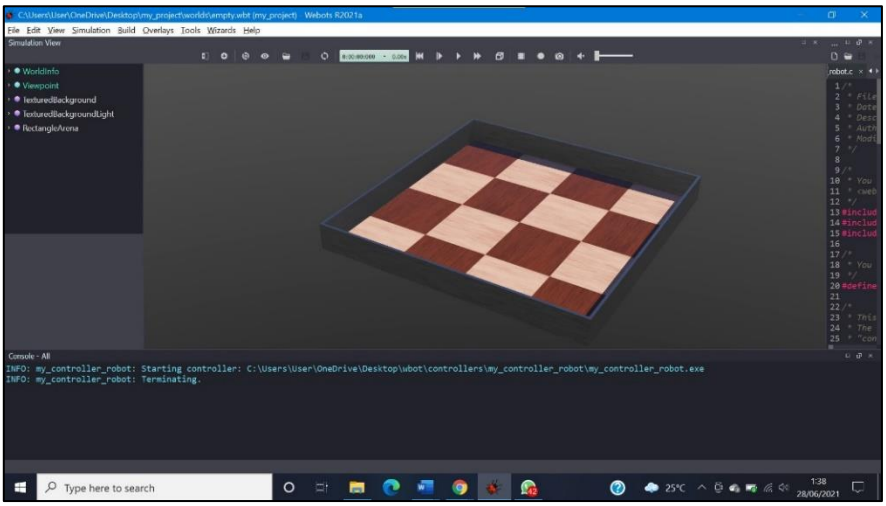
2. Materi dan Metode

Dalam Simulasi Pemahaman *Robot line follower* atau disebut sebagai robot pengikut garis menggunakan aplikasi *Webot* dengan perantara robot *e-puck*, metode yang digunakan dalam penelitian nya adalah metode deskriptif dan pengembangan dimana untuk mendapat nilai output berupa hasil pemahaman akan sistem kerja *robot line follower* dicari jawaban nya dengan gambaran dan cara untuk memperoleh data sehingga dapat dipergunakan untuk menghasilkan dan mengembangkan nilai pemahaman tersebut. Untuk sistem simulasi *webot* penyusunan program dapat disesuaikan dengan library prototipe *e-puck*. Dalam tahap penyusunan nya dapat dilakukan dengan beberapa tahapan pemahaman yang dilakukan :

2.1. *Webot*

*Webot*s adalah sebuah aplikasi atau perangkat lunak yang dikembangkan oleh Swiss Federal Institute of Technology yang berfungsi untuk membuat model, program dan simulasi dari sebuah robot. Pada aplikasi ini kita dapat merancang atau mendesain serta membuat program dari robot yang ingin kita buat terlebih dahulu sehingga nantinya saat pembuatan secara langsung kita dapat mengerjakannya dengan lebih mudah. Aplikasi ini pertama kali dikembangkan pada tahun 1996 oleh Dr. Oliver Michael, lalu pada tahun 1998 dikembangkan lagi oleh Cyberbotics Ltd. sebagai perangkat lunak berlisensi berpemilik. Dan pada tahun 2018 aplikasi ini dirilis sebagai aplikasi yang dapat di download secara gratis dan open source dengan Lisensi Apache. Aplikasi *Webot*s ini dapat kita gunakan pada sistem operasi Windows 10, Linux 64 bit dan Mac Os X 10.14, 10.13. Sedangkan untuk sistem operasi selain itu, sistemnya tidak memadai sehingga kita tidak dapat menggunakan aplikasi ini pada sistem operasi selain yang disebutkan.

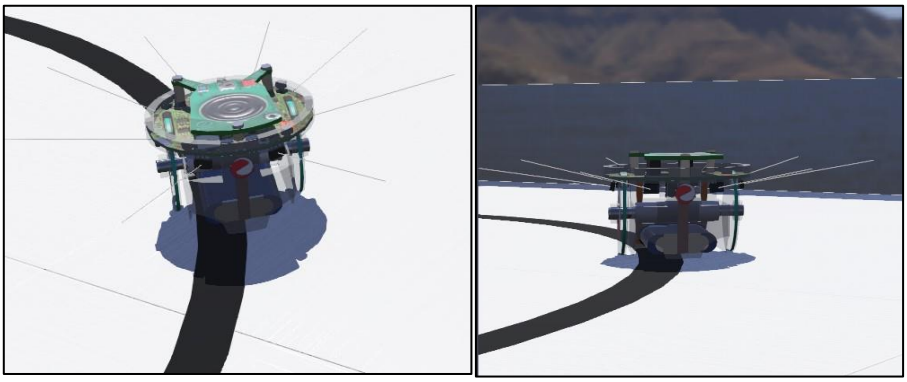
Pada aplikasi ini terdapat banyak model robot, sensor, akuator dan juga objek yang dapat kita modifikasi secara bebas. Selain itu, kita juga dapat membuat model baru yang kita desain sendiri ataupun kita import dari perangkat lunak atau aplikasi 3D CAD. Dimana saat mendesain, kita dapat menentukan properti grafis dan fisik objek untuk model robot yang akan kita buat. Untuk properti grafis terdiri dari bentuk, dimensi, posisi dan orientasi, warna dan tekstur objek. Dan untuk sifat fisik terdiri dari massa, faktor gesekan, pegas dan pembahasan konstansa. Sedangkan untuk pengeontrol robotnya, kita dapat menulisnya diluar *Webot*s dalam C, C++, Python, ROS, Java dan MATLAB dengan menggunakan API sederhana.



Gambar 1. Tampilan Awal Project dari *Webot*

2.2. *E-puck*

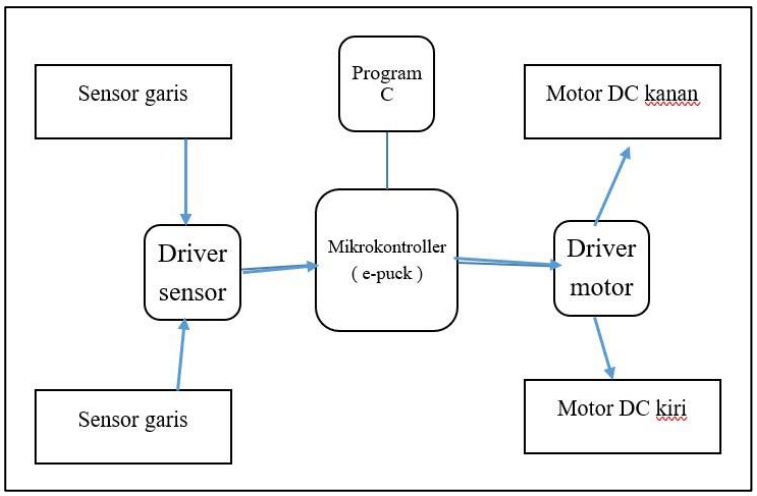
E-puck adalah sebuah robot mobil mini yang awalnya di desain untuk pembelajaran perancangan robot. Perangkat lunak dan perangkat keras dari *e-puck* ini dapat diakses oleh setiap perangkat electronic atau dikatakan open source. *E-puck* ini terdiri dari motor roda diferensial (encoder disimulasikan sebagai sensor posisi), sensor infrared merah yang berfungsi sebagai pengukur jarak dan cahaya, accelerometer, gyro, kamera, 8 LED disekelilingnya, bodi dan Led depan, Bluetooth dan ekstensi sensor ground.



Gambar 2. Tampilan Robot *e-puck*

2.3. Perancangan Sistem

Pada sistem ini kami merancang *robot line follower* atau sebuah robot yang bekerja dengan mengikuti sebuah garis hitam yang telah disediakan. Dan untuk pendeteksinya dengan menambahkan sensor infrared di bawah robot *e-puck*. Untuk pengontrol dari pergerakan robot ini, digunakan mikrokontroller dengan program bahasa C, serta motor driver dan motor DC sebagai penggerak jalannya robot. Pada pengoperasiannya nanti robot diprogram untuk bergerak mengikuti pantulan cahaya, maka dari itu digunakan sensor cahaya infrafred. Atau untuk penggambaran operasi robot nanti nya dapat kita lihat pada diagram blok perancangan sistem berikut :

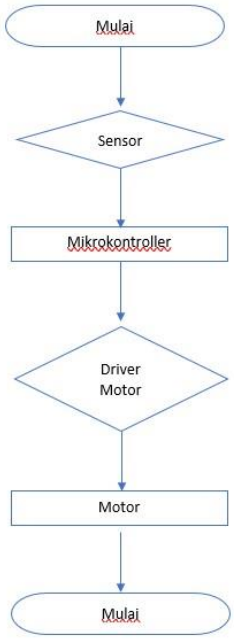


Gambar 3. Diagram Blok Perancangan Robot Simulasi

Keterangan blok :

1. Sensor garis berfungsi sebagai pendeteksi lintasan garis hitam di lantai.
2. Driver sensor berfungsi sebagai penguat sinyal logika high=1 dan low=0, dimana garis hitam dinyatakan 1 dan background lantai dinyatakan 0.
3. Mikrokontroller berfungsi sebagai pengendali robot dengan memasukkan program C yang berisi perintah untuk pergerakan robot pada mikrokontroller.
4. Driver motor berfungsi sebagai pengendali arah gerakan robot ke kana atau ke kiri.
5. Motor DC berfungsi sebagai penggerak mekanik agar robot bisa maju dan mundur dan berhenti.

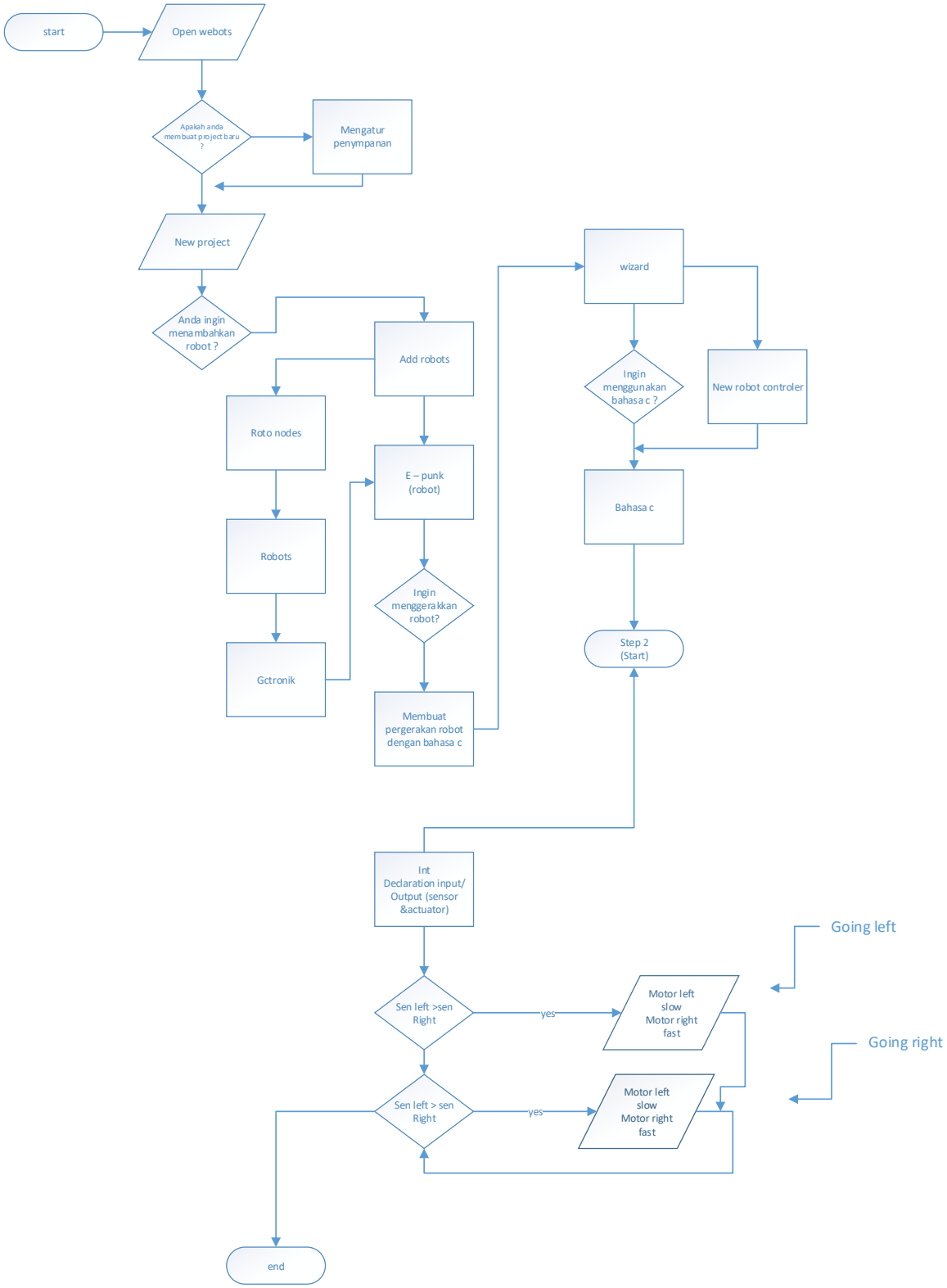
Robot bisa bergerak di lintasan garis hitam karena sudah di program dan dengan adanya sensor yang menangkap pantulan cahaya, sensor yang digunakan adalah cahaya infrared agar robot bisa bergerak mengikuti garis lintasan.



Gambar 4. Flowchart Pembacaan Sensor

2.4. Perancangan Simulasi Sistem Robot

Simulasi merupakan tahap pengaplikasian gambaran dalam perancangan sistem yang telah disusun. Dimana simulasi akan memberikan pertanyaan dan jawaban dalam memahaminya. Karena dengan simulasi ini diharapkan mendapatkan hasil yang baik dan maksimal. Simulasi dilaksanakan dengan menggunakan aplikasi *webot* dan robot perantara yaitu *e-puck*. Simulasi dilakukan dengan Rectangle Arena sesuai kebutuhan *robot line follower*. Dalam perancangan simulasi *robot line follower* ini penulis akan membuat pemahaman akan sistem kerja dari *robot line follower* yang dimulai dengan robot dapat merotasi kan dan memposisi kan kedudukan nya untuk menyesuaikan pergerakan dengan lintasan yang telah disiapkan dengan menanamkan bahasa C ke mikrokontroller *e-puck*. Maka dibuat rancangan program simulasi dalam bentuk diagram alir.



Gambar 5. Flowchart Sistem Simulasi Pemahaman Robot

Gambar diagram di atas merupakan proses yang akan dilakukan dalam menjalankan simulasi *robot line follower*. *Webot* menjadi perantara dalam memahami sistem kerja dengan skema 3D melalu robot *e-puck*. Simulasi dimulai dengan menginput new project sebagai penyiapan world dan arena simulasi. Robot diimport ke dalam arena dan disesuaikan pengaturan mikrokontroller robot menggunakan bahasa C yang telah terdapat perintah dari sistem kerja yang diharapkan yaitu robot dapat membaca lintasan berdasarkan tangkapan input dari sensor infrared atau sensor ir dan bergerak dengan kecepatan yang telah diatur dalam bahasa C mikrokontroller.

2.5. Program Bahasa C

Diketahui dalam simulasi pemahaman *robot line follower* dengan perantara robot *e-puck* melalui aplikasi *webot*s diterapkan dengan bahasa C yang merupakan salah satu bahasa high level dalam pemprograman. Penyusunan program dalam mikrokontroller berkaitan dengan hasil tangkapan sensor sebagai nilai input dalam mikrokontroller yang akan diproses menjadi output hasil ke motor dari robot *e-puck* kanan dan kiri yang disesuaikan kecepatan dan rotasi nya berdasarkan pengaturan yang diatur dalam bahasa programan. Berikut dilampirkan isi dari program bahasa C yang diinput dalam mikrokontroller robot *e-puck*:

E-puck.proto × my_controller_robot.c ×

```c
/*
 * File:          my_controller_robot.c
 * Date:
 * Description:
 * Author:
 * Modifications:
 */

/*
 * You may need to add include files like <webots/distance_sensor.h> or
 * <webots/motor.h>, etc.
 */
#include <webots/robot.h>
#include <webots/distance_sensor.h>
#include <stdio.h>

/*
 * You may want to add macros here.
 */
#define TIME_STEP 64

/*
 * This is the main program.
 * The arguments of the main function can be specified by the
 * "controllerArgs" field of the Robot node
 */
int main(int argc, char **argv) {
  /* necessary to initialize webots stuff */
  wb_robot_init();

  /*
   * You should declare here WbDeviceTag variables for storing
   * robot devices like this:
   *  WbDeviceTag my_sensor = wb_robot_get_device("my_sensor");
   *  WbDeviceTag my_actuator = wb_robot_get_device("my_actuator");
   */
```

Gambar 6. Coding C Library, Time Step, dan Fungsi Main

E-puck.proto × my_controller_robot.c ×

```
37
38    /*motor*/
39    WbDeviceTag mo_kanan = wb_robot_get_device("right wheel motor");
40    WbDeviceTag mo_kiri = wb_robot_get_device("left wheel motor");
41    wb_motor_set_position(mo_kanan, INFINITY);
42    wb_motor_set_position(mo_kiri, INFINITY);
43
44
45    /*sensor*/
46    WbDeviceTag ir_ka = wb_robot_get_device("sen_ka");
47    WbDeviceTag ir_ki = wb_robot_get_device("sen_ki");
48    wb_distance_sensor_enable(ir_ka, TIME_STEP);
49    wb_distance_sensor_enable(ir_ki, TIME_STEP);
50
51
52
53  /* main loop
54    * Perform simulation steps of TIME_STEP milliseconds
55    * and leave the loop when the simulation is over
56    */
57
58  while (wb_robot_step(TIME_STEP) != -1) {
59      /*
60       * Read the sensors :
61       * Enter here functions to read sensor data, like:
62       *  double val = wb_distance_sensor_get_value(my_sensor);
63       */
64      WbDeviceTag mo_kanan = wb_robot_get_device("right wheel motor");
65      WbDeviceTag mo_kiri = wb_robot_get_device("left wheel motor");
66
67      double data_ka = wb_distance_sensor_get_value(ir_ka);
68      double data_ki = wb_distance_sensor_get_value(ir_ki);
69      printf("%f", data_ka);
70      printf("%f\n", data_ka);
```

Gambar 7. Coding C Motor, Sensor, Loop, dan Pelampiran Nilai Data

```
71
72    /* Process sensor data here */
73    if (data_ki>data_ka){
74      printf("go left");
75      wb_motor_set_velocity(mo_kanan, 3.0);
76      wb_motor_set_velocity(mo_kiri, 0.1);
77    }
78
79    if (data_ki<data_ka){
80      printf("go right");
81      wb_motor_set_velocity(mo_kanan, 0.1);
82      wb_motor_set_velocity(mo_kiri, 3.0);
83    }
84
85    /*
86     * Enter here functions to send actuator commands, like:
87     * wb_motor_set_position(my_actuator, 10.0);
88     */
89
90  };
91
92  /* Enter your cleanup code here */
93
94  /* This is necessary to cleanup webots resources */
95  wb_robot_cleanup();
96
97  return 0;
98}
```

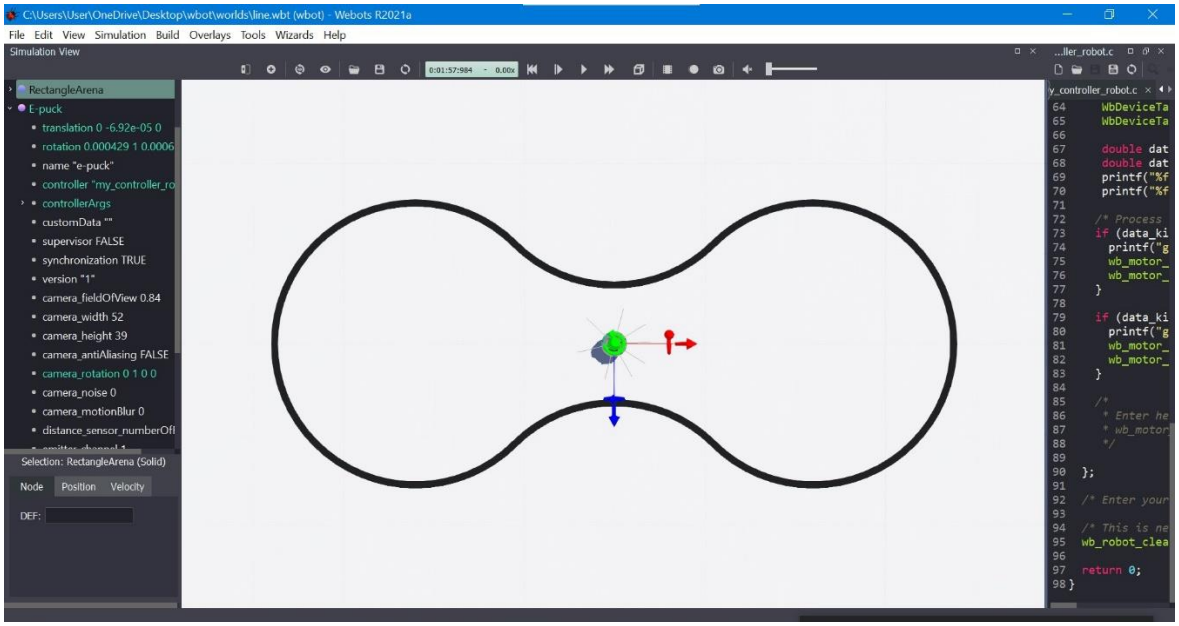
Gambar 8. Coding C Proses Sensor Data dan Akhir

3. Analisa dan Pembahasan

Dalam mengambil nilai output pemahaman terkait *robot line follower* maka dilakukan analisa dan pembahasan bagian – bagian yang akan menjadi indeks pemahaman terkait sistem kerja *robot line follower* dengan mengambil data gambaran dan pengembangan akan program simulasi yang dijalankan. Beberapa hal yang menjadi hasil dan analisa pembahasan yaitu terkait sistem tampilan *robot line follower*, program C yang dijalan kan dan beberapa pengujian terkait posisi dan keefektifitas dari robot.

3.1. Tampilan Simulasi *Webot*

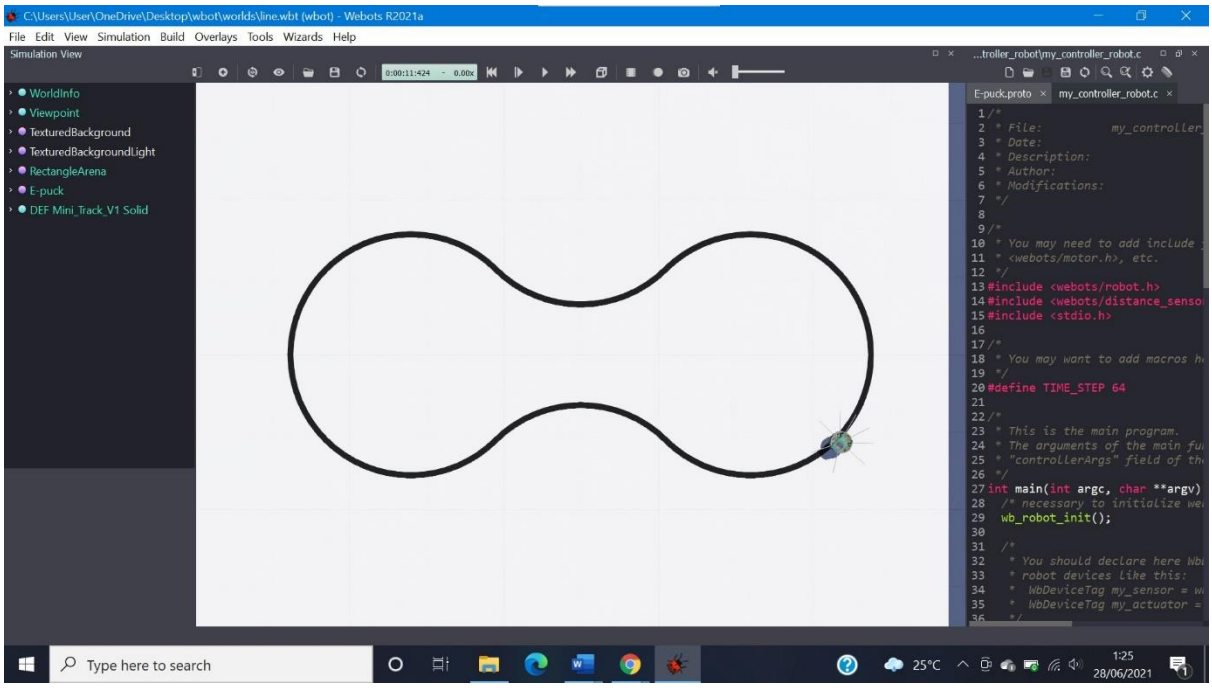
Perancangan sistem dan simulasi *robot line follower* ditampilkan dalam aplikasi *webot*s yang dapat menampilkan hasil simulasi dengan skema 3D. Dalam Bab ini akan mejelaskan bagaimana hasil dari pemahaman simulasi *robot line follower* dengan aplikasi *Webot*s. Pengujian dan percobaan dilakukan untuk memunculkan pertanyaan dan jawaban dalam pemahaman sistem kerja robot tersebut. Sehingga diharapkan dengan adanya simulasi akan mendapatkan hasil dari pemahaman yang baik dan bermanfaat. *Robot line follower* merupakan salah satu jenis robot yang bergerak mengikuti garis. *Robot line follower* dapat mengikuti jalur yang berupa garis hitam pada papan berwarna putih maupun garis putih pada papan berwarna hitam. Pada dasarnya cara kerja *robot line follower* adalah dengan menangkap bias cahaya yang dipantulkan dari papan lintasan menggunakan sensor. Pada awalnya posisi robot ini di atur pada kordinat (x=0, y=0.6, dan z=0). Untuk lebih jelasnya perhatikanlah gambar berikut :



Gambar 9. Tampilan Posisi Awal Simulasi *Robot line follower*

Adapun sensor yang digunakan pada robot ini adalah dua unit Sensor Photodiode (Sensor Infrared) di bagian bawah. Photodiode merupakan salah satu sensor yang resistansinya dapat berubah saat terkena sinar / cahaya yang didapat dari pancaran LED sebagai transmitter. Nilai resistansi pada Photodiode akan semakin besar saat Photodiode tidak terkena cahaya, sebaliknya nilai resistansi pada Photodiode akan semakin kecil jika Photodiode terkena cahaya yang semakin terang.

Setelah beda bias cahaya didapat dari sensor, selanjutnya hasil sensor dikirim pada mikrokontroler yang di pasang pada robot. Data sensor yang sudah diproses dalam mikrokontroler akan menghasilkan data keluaran untuk mengendalikan motor melalui driver motor. Sehingga nantinya robot dapat beroperasi sesuai dengan yang di inginkan, seperti terlihat pada gambar dibawah ini :



Gambar 10. Tampilan Eksekusi Simulasi *Robot line follower*

Selain itu, pada robot ini juga dilengkapi dengan 8 sensor Infrared yang berada di bagian samping berfungsi untuk mendeteksi dinding atau benda yang ada di hadapannya, sehingga robot akan dapat bermanufer untuk menghindari penghalang. Dan kondisi ini dibutuhkan saat robot masih dalam kondisi belum menemukan lintasan.

3.2. Pembahasan Program Bahasa C

Dalam perancangan *Robot line follower*s ini, penulis menggunakan bahasa C sebagai bahasa pemrograman untuk controllersnya. Penyusunan program bahasa C untuk *robot line follower* dengan perantara *e-puck* robot sudah tertera fungsi atau program bawaan dari *e-puck* dimana *webot* telah menyediakan library dan metode program yang berkaitan dengan sistem kerja *e-puck*. Yang perlu dilakukan dalam penyusunan program bahasa C untuk simulasi pemahaman *robot line follower* ini adalah mengatur dan memodifikasi susunan program dalam proto *e-puck* menjadi *robot line follower*.

```
13 #include <webots/robot.h>
14 #include <webots/distance_sensor.h>
15 #include <stdio.h>
16
17 /*
18  * You may want to add macros here.
19  */
20 #define TIME_STEP 64
```

Gambar 11. Coding C Library Program dan Time Step

Pada source code di atas dapat kita lihat, untuk di barisan awal kita terlebih dahulu memasukkan library robot.h, distance_sensor.h dan stdio.h yang berfungsi untuk menjalankan metode yang akan kita jalankan di dalam fungsi main-nya. Setelah itu, kita defenisikan TIME_STEP 64.

```
38   /*motor*/
39   WbDeviceTag mo_kanan = wb_robot_get_device("right wheel motor");
40   WbDeviceTag mo_kiri = wb_robot_get_device("left wheel motor");
41   wb_motor_set_position(mo_kanan, INFINITY);
42   wb_motor_set_position(mo_kiri, INFINITY);
43
44
45   /*sensor*/
46   WbDeviceTag ir_ka = wb_robot_get_device("sen_ka");
47   WbDeviceTag ir_ki = wb_robot_get_device("sen_ki");
48   wb_distance_sensor_enable(ir_ka, TIME_STEP);
49   wb_distance_sensor_enable(ir_ki, TIME_STEP);
```

Gambar 12. Coding C Variabel Fungsi Main

Dan untuk di fungsi mainnya, kita definisikan variabel yang menampung data untuk motor kanan (right wheel motor) dan motor kiri (left wheel motor) pada baris ke-39 hingga baris ke-42. Sedangkan untuk data sensornya kita letakkan pada variabel ir_ka dan ir_ki pada baris ke-46 hingga baris ke-49.

```
58 while (wb_robot_step(TIME_STEP) != -1) {
59   /*
60    * Read the sensors :
61    * Enter here functions to read sensor data, like:
62    *  double val = wb_distance_sensor_get_value(my_sensor);
63    */
64    WbDeviceTag mo_kanan = wb_robot_get_device("right wheel motor");
65    WbDeviceTag mo_kiri = wb_robot_get_device("left wheel motor");
66
67    double data_ka = wb_distance_sensor_get_value(ir_ka);
68    double data_ki = wb_distance_sensor_get_value(ir_ki);
69    printf("%f", data_ka);
70    printf("%f\n", data_ka);
```

Gambar 13. Coding C Looping dan Penampilan Nilai Data

Setelah selesai mendefinisikan beberapa variabel yang menampung data-data sensor dan motornya, setelah itu kita tuliskan sorce code untuk menjalankan robot sesuai kondisi yang telah kita rancang. Dimana pada awalnya kita mengambil data dari kedua sensor yang telah dipasang pada robot, pada awalnya sensor akan membaca posisi robot, apabila robot tidak berada di atas line(garis) yang telah disediakan maka secara

otomatis robot akan mencari garis tersebut. Setelah robot menemukan garis, dia akan terus berjalan di atas garis tersebut, dimana untuk source codenya bisa dilihat pada baris ke-67 hingga 70.

```
71
72    /* Process sensor data here */
73    if (data_ki>data_ka){
74      printf("go left");
75      wb_motor_set_velocity(mo_kanan, 3.0);
76      wb_motor_set_velocity(mo_kiri, 0.1);
77    }
78
79    if (data_ki<data_ka){
80      printf("go right");
81      wb_motor_set_velocity(mo_kanan, 0.1);
82      wb_motor_set_velocity(mo_kiri, 3.0);
83    }
```

Gambar 14. Coding C Prosessan Data Sensor

Setelah robot berada pada garis, maka selanjutnya baris kode ke-73 hingga baris ke-83 akan di eksekusi. Pada tahap ini ada dua kondisi yang akan di jalankan. Pertama, apabila data_ki lebih besar dari pada data_ka maka robot akan bergeser ke kiri dengan cara mengatur motor kanan pada kecepatan 0,1 dan motor kiri pada kecepatan 3,0 sehingga mengakibatkan robot bergeser ke kiri. Kedua, apabila data_ki lebih kecil dari data_ka maka robot akan bergeser ke kanan dengan cara mengatur motor kanan pada kecepatan 3,0 dan motor kiri pada kecepatan 0,1 sehingga mengakibatkan robot bergeser ke kanan. Pada simulasi ini robot akan terus mengikuti garis yang telah di tentukan sebelumnya hingga keadaan robotnya off.

3.3. Pembahasan Sistem Lanjutan

Pada simulasi ini kami melakukan beberapa percobaan dengan beberapa titik koordinat yang dapat diliat pada table dibawah ini:

Tabel 1.1. Hasil percobaan pada simulasi

Posisi	x(m)	y(m)	z(m)	Orientasi(rad)
ps0	-0.001008	0.999999	0.0006434	1.17
ps1	-0.001054	0.999999	0.0006454	1.02
ps2	-0.001185	0.999999	0.0006367	1.14
ps3	0.001273	0.999999	0.0006452	1.03
ps4	0.003467	0.999999	-0.000876	1.09

Pada tabel diatas dapat kita lihat hasil dari 5 percobaan yang dilakukan pada simulasi dengan 5 titik koordinat yang berbeda dan nilai positif negative pada x dan z yang bervariasi. Berdasarkan tabel dapat kita lihat bahwa nilai orientasi yang ada semuanya berkisar pada angka satu dengan angka dibelakang koma yang memiliki jarak tidak terlalu jauh satu sama lainnya, padahal untuk nilai x dan z nya sendiri sudah diubah dengan bermacam angka dan nilai. Maka dari itu, dapat kita simpulkan bahwa mau bagaimana pun kita atur titik koordinatnya, maka *robot line follower* ini akan kembali pada jalurnya dan bergerak mengikuti garis hitam seperti yang sudah kita atur pada pemrogramannya.

4. Kesimpulan

Robot Pengikut Garis adalah salah satu jenis robot sederhana dengan banyak manfaat serta pengembangan. Maka dari itu, robot ini sering digunakan untuk pembelajaran perancangan dan dasar robot. Dan *Webot*s adalah perangkat lunak yang biasa digunakan untuk mendesain serta membuat simulasi robot yang kami gunakan untuk membuat simulasi dari robot pengikut garis tersebut.

Berdasarkan simulasi yang sudah kami buat dan jalankan serta percobaan yang sudah kami lakukan pada simulasi tersebut, dapat kami simpulkan bahwa jika kita meletakkan koordinatnya tidak tepat pada garis hitam, maka program kontrolnya akan mengarahkan *e-puck* kembali ke garis hitam dan berjalan sebagaimana mestinya.

References

[1] Fu, King Sun, Ralph Gonzalez, and CS George Lee. *Robotics: Control Sensing*. Vis. Tata McGraw-Hill Education, 1987.

[2] Haryadi Poedji Sudrajat. " Simulator Robot Soccer Menggunakan *Webot*s". *Jurnal Ilmiah Mahasiswa Universitas Surabaya Vol.1 No.1* (2012)

[3] Mukti, Anggoro, Oky Dwi Nurhayati, and Eko Didik Widianto. "Rancang Bangun Sistem Kontrol *Robot line follower* Menggunakan Logika Fuzzy." *Jurnal Teknologi dan Sistem Komputer* 3, no. 4 (2015): 536-543.

[4] Sijabat, Salomo. "SIMULASI PERGERAKAN DAN PERANCANGAN ROBOT PENGIKUT JEJAK BERBASIS MICROCONTROLLER DENGAN METODE FUZZY LOGIC (PART2)." *Jurnal Mantik Penusa* 17, no. 1 (2015).

[5] Tarquino P. Nickels K. " Programing an *E-puck* robot to create maps of virtual and physical environment. In: Robot intelligence technology and applications 2. Springer, pp 13-28 (2014).

[6] Mondada, Francesco, Michael Bonani, Xavier Raemy, James Pugh, Christopher Cianci, Adam Klaptocz, Stephane Magnenat, Jean-Christophe Zufferey, Dario Floreano, and Alcherio Martinoli. "The e-puck, a robot designed for education in engineering." In Proceedings of the 9th conference on autonomous robot systems and competitions, vol. 1, no. CONF, pp. 59-65. IPCB: Instituto Politécnico de Castelo Branco, 2009.

[7] Guyot, Luc, Nicolas Heiniger, Olivier Michel, and Fabien Rohrer. "Teaching robotics with an open curriculum based on the e-puck robot, simulations and competitions." In Proceedings of the 2nd International Conference on Robotics in Education. Vienna, Austria. 2011.

[8] Pakdaman, Mehran, and M. Mehdi Sanaatiyan. "Design and implementation of line follower robot." In 2009 second international conference on computer and electrical engineering, vol. 2, pp. 585-590. IEEE, 2009.

[9] Hasan, Kazi Mahmud, and Abdullah Al Mamun. "Implementation of autonomous line follower robot." In 2012 International Conference on Informatics, Electronics & Vision (ICIEV), pp. 865-869. IEEE, 2012.

[10] Engin, Mustafa, and Dilşad Engin. "Path planning of line follower robot." In 2012 5th European DSP Education and Research Conference (EDERC), pp. 1-5. IEEE, 2012.