

BAB III. METODOLOGI PENELITIAN

3.1. Waktu dan Tempat Penelitian

Penelitian dilakukan di kampus Politeknik Negeri Malang. Penelitian dilaksanakan selama 6 bulan dimulai pada bulan Desember 2020 sampai dengan Mei 2021.

3.2. Metode Pengambilan Data

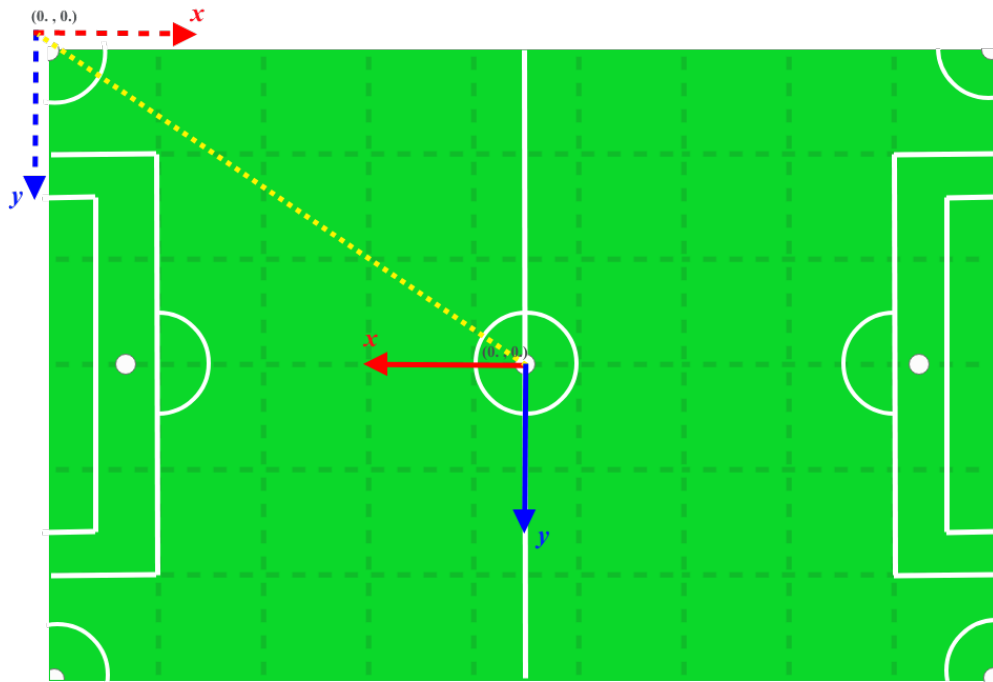
Teknik pengumpulan data pada penelitian ini dilakukan dengan mengambil data posisi koordinat robot asli, dan posisi koordinat robot kawan yang dilakukan secara *real time* dengan menggunakan pemrograman *socket* berbasis *User Datagram Protocol* (UDP). Pengambilan posisi koordinat robot lawan atau *obstacle* dan posisi koordinat akhir atau tujuan dengan menggunakan penentuan posisi koordinat secara manual pada aplikasi. Pengiriman data *socket* UDP memiliki beberapa ketentuan, antara lain sebagai berikut:

1. Format pengiriman data dari robot ke server harus berupa json dengan ketentuan {"pos": (0.,0.,0.), "ball": (0.,0.,0.), "exit": 0, "counter":0} yang memiliki arti:
 - pos: menyatakan data posisi koordinat robot dengan format pengiriman data x, y, z .
 - ball: menyatakan data posisi koordinat tujuan robot yang diinginkan dengan format x, y, z .
 - exit: menyatakan status koneksi *socket* UDP robot, kode 0 menyatakan status aktif, dan kode 1 menyatakan status koneksi terputus.
 - counter: menyatakan posisi path yang dituju oleh robot pada indeks berapa.
2. Format pengiriman data dari server ke robot harus berupa json dengan ketentuan {"cmd":0, "path": [(0.,0.), (2.,3.0)]} yang artinya:
 - cmd: menyatakan kondisi yang harus dijalankan oleh robot, kode 0 menyatakan robot harus berhenti, dan kode 1 menyatakan robot harus bergerak sesuai perencanaan jalur yang dikirim dari server.
 - path: total jalur perencanaan dengan format koordinat x, y .

3.3. Metode Pengolahan Data

Pengolahan data yang dilakukan dengan cara mengelola data seluruh koordinat yang diterima melalui socket *User Datagram Protocol* (UDP) ataupun secara manual dari aplikasi. Kemudian data posisi koordinat keseluruhan dipetakan ke dalam model *grid* pada peta lapangan yang telah ditentukan secara statis. Hasil dari pemetaan tersebut akan dijadikan informasi penting untuk pencarian rute perjalanan menggunakan metode *A-star Search* (A*). Setelah melakukan pencarian rute menggunakan A*, hasil dari rute bersangkutan disederhanakan menggunakan metode *Improved A-star Search* (A*). Proses digambarkan menggunakan *flowchart* yang direpresentasikan pada gambar 3.1. Berdasarkan *flowchart* yang dibuat terdapat langkah – langkah sebagai berikut:

1. Pada saat ingin melakukan perencanaan rute terhadap robot, *user* akan dihadapkan dengan pemilihan model simulasi, yaitu model penerapan rute robot terhadap robot *dummy*, atau terhadap robot asli.
2. Kemudian menentukan posisi koordinat titik akhir tujuan robot secara manual pada aplikasi.
3. Menambahkan posisi koordinat robot lawan atau *obstacle* secara manual pada aplikasi dengan jumlah inputan maksimal 3.
4. Jika pemilihan model penerapan terhadap robot asli, maka *user* diharapkan untuk memulai koneksi *socket* UDP dengan model *broadcast*.
5. Jika koneksi dari *socket* UDP berhasil maka seluruh data koordinat robot yang terkoneksi melalui IP Address *socket* UDP server akan diterima dan dimasukkan ke dalam variabel array untuk menjadikan posisi koordinat robot kawan.
6. Sebelum melakukan penyimpanan koordinat posisi robot yang terkoneksi ke dalam array, maka akan dilakukan transformasi posisi robot terhadap posisi peta lapangan statis yang berukuran 9m x 6m.



Gambar 3.1 Transformasi koordinat robot

Pada gambar 3.1 merupakan representasi dari transformasi koordinat robot. Berikut penjelasan dari transformasi koordinat robot:

- Garis berwarna merah putus – putus menyatakan posisi koordinat x peta lapangan statis.
- Garis berwarna biru putus – putus menyatakan posisi koordinat y peta lapangan statis.
- Garis berwarna kuning putus – putus menyatakan posisi koordinat transformasi robot asli terhadap peta lapangan statis.
- Garis berwarna merah menyatakan posisi koordinat x robot asli.
- Garis berwarna biru menyatakan posisi koordinat y robot asli.

Rumus perhitungan transformasi robot terhadap lapangan direpresentasikan pada persamaan 3.1, dan 3.2.

$$x_{new} = \left(\frac{9}{2}\right) - x \quad (3.1)$$

$$y_{new} = \left(\frac{6}{2}\right) - (y * -1) \quad (3.2)$$

dengan x adalah posisi koordinat x robot, dan y adalah posisi koordinat y robot.

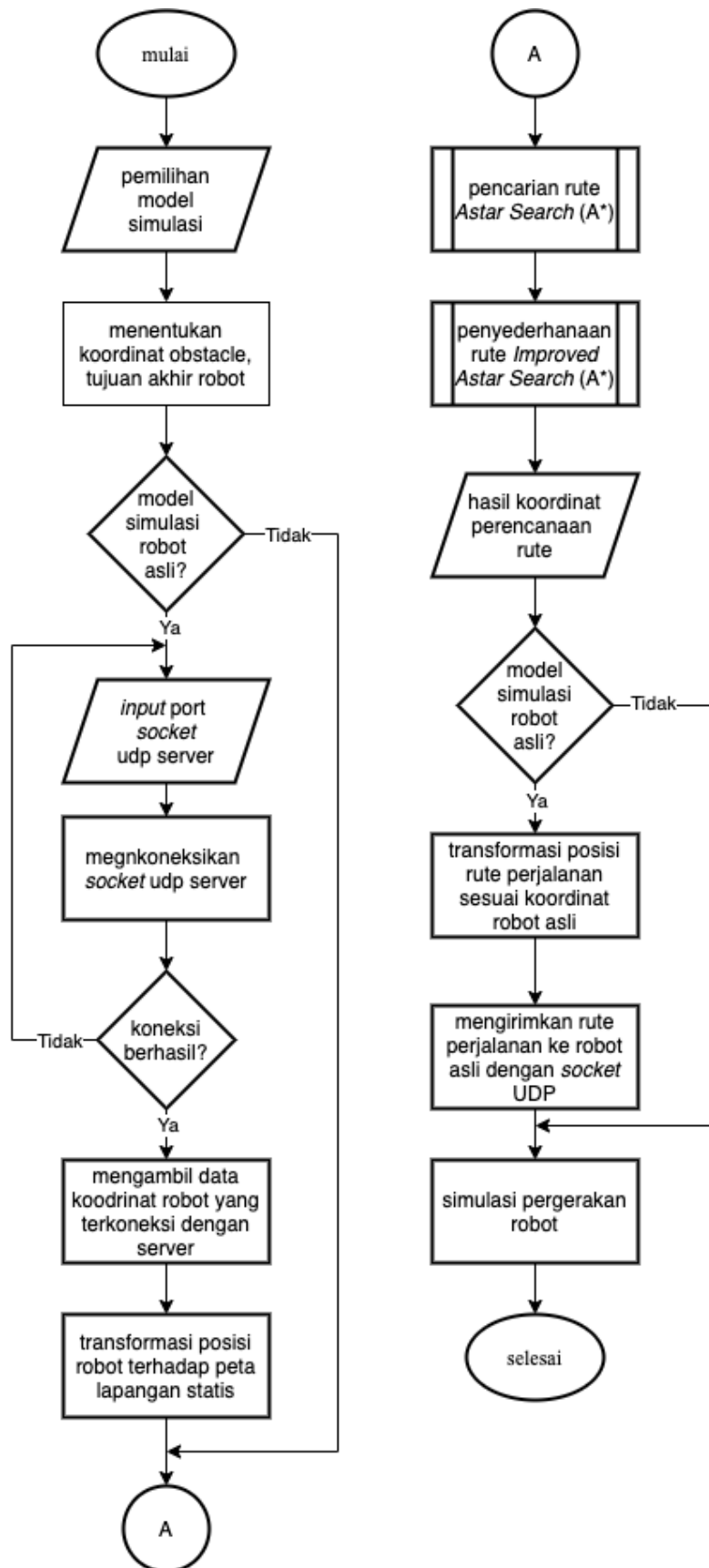
7. Data koordinat yang telah tersimpan dalam variabel akan dijadikan informasi untuk pencarian rute perjalanan dengan menggunakan metode A* yang direpresentasikan dengan persamaan 2.1, 2.2, dan 2.3.
8. Data koordinat posisi yang telah didapatkan dengan metode A* akan disederhanakan kembali dengan metode *Improved A** dengan kombinasi metode *Digital Differential Analyzer (DDA)* yang direpresentasikan dengan persamaan 2.5, 2.6, 2.7, dan 2.8.
9. Jika pemilihan model penerapan terhadap robot asli, maka sebelum melakukan simulasi pergerakan robot, data rute perjalanan yang diterima akan ditransformasikan dari peta lapangan statis terhadap robot asli, atau kebalikan dari transformasi pada langkah ke 7. Rumus perhitungan transformasi lapangan terhadap robot direpresentasikan pada persamaan 3.3, dan 3.4.

$$\sum_{i=0}^n x_i = \left(\frac{9}{2}\right) - xp_i \quad (3.3)$$

$$\sum_{i=0}^n y_i = \left(\left(\frac{6}{2}\right) - yp_i\right) * -1 \quad (3.4)$$

dengan xp_i adalah posisi koordinat x rute perjalanan setiap indeks, dan yp_i adalah posisi koordinat y rute perjalanan setiap indeks.

10. Output dari rute perjalanan yang telah didapatkan akan disimulasikan secara *real time* pada aplikasi.



Gambar 3.2 Flowchart Sistem

3.4. Desain Sistem

3.4.1 Metode Pengembangan Perangkat Lunak

Dalam pengembangan perangkat lunak Penerapan Algoritma *Path Planning* pada Robot Sepak Bola Beroda Politeknik Negeri Malang yaitu menggunakan alur *Software Development Life Cycle (SDLC) Model Prototype*. *SDLC Prototype* memiliki tahapan sebagai berikut:

1. Mengidentifikasi kebutuhan

Tahapan pertama yang dilakukan pada penelitian ini adalah pencarian literatur,serta mengumpulkan data yang dibutuhkan untuk penelitian ini.

2. *Quick Plan* (Perencanaan Cepat)

Tahapan selanjutnya adalah perencanaan cepat dengan memberikan solusi permasalahan yang sudah diidentifikasi pada tahapan mengidentifikasi kebutuhan.

3. *Modeling* (Perencanaan Desain)

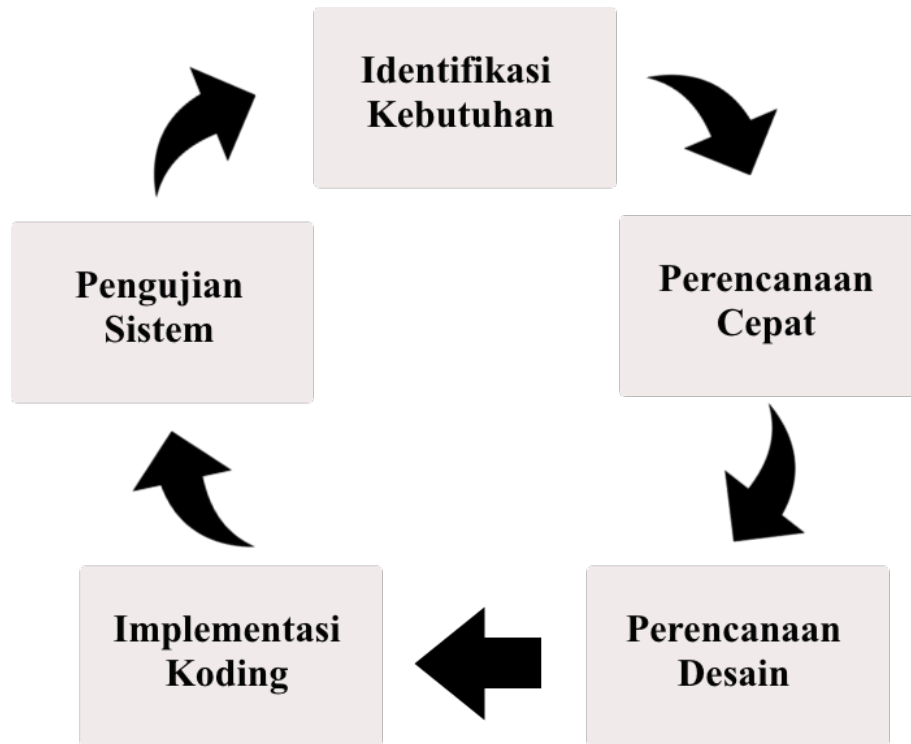
Tahapan selanjutnya adalah perancangan desain sistem atau pemodelan dari sistem yang akan dibangun seperti *flowchart*, use case diagram, mockup, dan melakukan perancangan arsitektur sistem.

4. *Construction* (Implementasi Koding)

Tahapan berikutnya yaitu konstruksi berupa implementasi kedalam bahasa pemrograman yang dapat dimengerti oleh sistem atau pengkodean berdasarkan rancangan-rancangan yang telah dibuat sesuai dengan tahapan modeling.

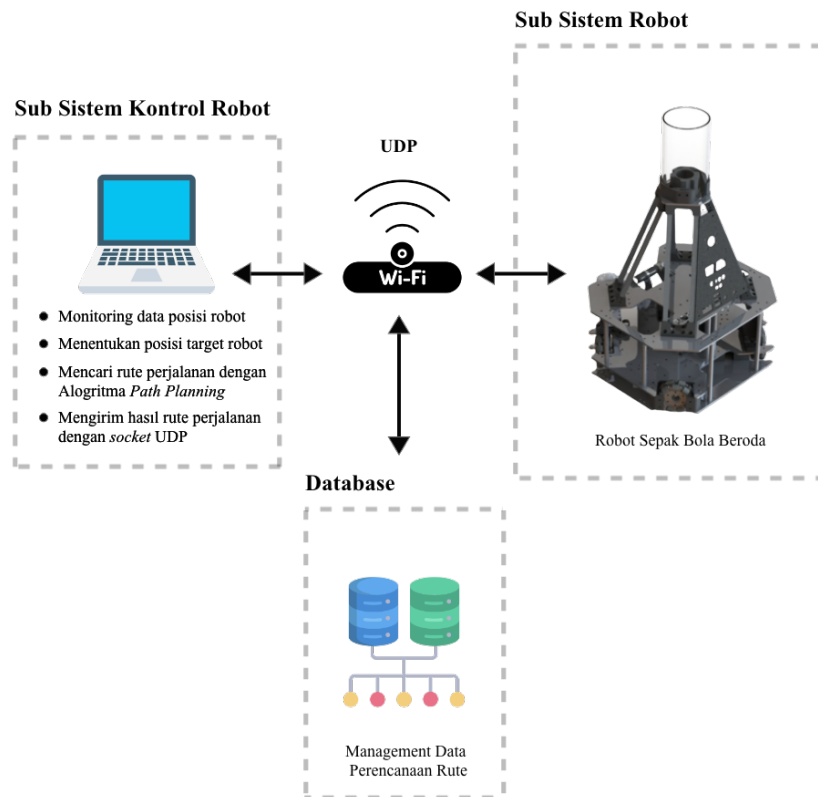
5. Pengujian Sistem

Menguji Sistem Pada tahap pengujian sistem, koding yang telah dibuat sebelumnya akan diuji apakah dapat berjalan dengan baik atau masih ada bagian-bagian yang perlu diperbaiki sehingga sistem yang ingin dibangun sesuai dengan perancangan model yang telah direncanakan sebelumnya.



Gambar 3.3 SDLC *Prototype*

3.4.2 Arsitektur Sistem



Gambar 3.4 Desain Arsitektur Sistem

Arsitektur sistem ditunjukkan dalam Gambar 3.4 terdiri dari beberapa subsistem, yaitu sebagai berikut:

- Sub Sistem Kontrol Robot

Sub sistem kontrol robot menampilkan data posisi koordinat robot saat ini, dan posisi koordinat tujuan atau target. Informasi data yang didapatkan akan diolah oleh komputer dengan menggunakan Algoritma *Improved A-star search* (A*) untuk perencanaan rute perjalanan robot. Hasil data rute perjalanan robot akan dikirimkan melalui *socket* UDP secara “*broadcast*”.

- Sub Sistem Robot

Robot sepak bola beroda meminta data rute perjalanan dari komputer dengan menggunakan *socket* UDP. Data bersangkutan digunakan sebagai informasi untuk pergerakan robot dari titik koordinat robot saat ini menuju titik tujuan selanjutnya. Posisi koordinat robot saat ini akan dikirimkan dengan *socket* UDP untuk dapat diolah oleh komputer untuk penentuan rute perjalanan terbaru.

3.5. Skenario Uji Coba Sistem

Proses pengujian Sistem *path planning* pada penelitian ini dilakukan dengan meliputi beberapa tahapan, yaitu:

1. Pengujian akurasi metode *Improved A-star search* (A*). Pengujian ini bertujuan untuk mengetahui tingkat keakuratan metode *Improved A-star search* (A*) yang sudah diimplementasikan dalam bentuk sebuah program. Tabel 3.1 merepresentasikan data uji coba algoritma *Improved A-star search* (A*) terhadap beberapa sampel yang telah ditentukan.

Tabel 3.1 Persentase Akurasi Improved A*

No.	Jumlah Sampel Percobaan	Tingkat Keberhasilan
1.	3 Sampel	0 – 100 %
2.	5 Sampel	0 – 100 %
3.	7 Sampel	0 – 100 %
4.	10 Sampel	0 – 100 %
5.	13 Sampel	0 – 100 %
6.	15 Sampel	0 – 100 %

Berdasarkan uji coba sampel percobaan terhadap robot sepak bola beroda pada tabel 3.1, Berikut adalah langkah – langkah dari pengujian akurasi *Improved A** dengan menggunakan sampel percobaan:

1. Penentuan jumlah sampel percobaan total.
2. Pengecekan setiap percobaan pada robot hingga percobaan ke n total sampel, apakah mengalami tabrakan terhadap *obstacle*, robot kawan atau tidak, jika mengalami tabrakan maka percobaan dianggap gagal, tetapi jika tidak mengalami tabrakan percobaan dianggap berhasil.
3. Melakukan perhitungan akurasi dengan melihat jumlah keberhasilan dan efektifitas perencanaan rute percobaan robot yang tidak mengalami tabrakan dengan rumus yang direpresentasikan pada persamaan 3.5.

$$\sum_{i=0}^n akurasi = \left(\frac{\sum_{i=0}^n keberhasilan}{\sum total_{sampel}} \right) * 100 \quad (3.5)$$

dengan $\sum_{i=0}^n keberhasilan$ adalah jumlah percobaan keberhasilan setiap indeks ke 0 hingga indeks ke n , dan $\sum total_{sampel}$ menyatakan total sampel percobaan pada setiap indeks.

2. Pengujian anotasi manual dengan hasil pemilihan *node* Algoritma *Improved A**. Pengujian ini bertujuan untuk mengetahui pergerakan robot sepak bola beroda pada setiap *node* yang dilewati apakah sesuai dengan rute perjalanan setiap *node* paling efektif yang seharusnya dipilih oleh anotasi manual berdasarkan persepsi dari manusia. Tabel 3.2 merepresentasikan data uji coba anotasi manual dengan hasil pemilihan *node* dengan Algoritma bersangkutan.

Tabel 3.2 Persentase Akurasi Perencanaan Node

No.	Node Algoritma		
1.	<i>Start: (0., 0.)</i>	<i>Node Persepsi Manusia</i>	Tingkat Keberhasilan
	<i>End: (4., 0.)</i>		
	(0.0, 0.0)	-	0 atau 1
	(1., 0.)	-	0 atau 1
	(2., 0.)	-	0 atau 1
	(3., 0.)	-	0 atau 1
	(4., 0.)	-	0 atau 1
Total Akurasi			0 – 100 %
2.	<i>Start: (4., 0.)</i>	<i>Node Persepsi Manusia</i>	Tingkat Keberhasilan
	<i>End: (0., 0.)</i>		
	(4.0, 0.0)	-	0 atau 1
	(3., 0.)	-	0 atau 1
	(2., 0.)	-	0 atau 1
	(1., 0.)	-	0 atau 1
	(0., 0.)	-	0 atau 1
Total Akurasi			0 – 100 %

Berdasarkan uji coba anotasi manual percobaan terhadap robot sepak bola beroda pada tabel 3.2. Berikut adalah langkah – langkah dari pengujian akurasi *Improved A** berdasarkan perbandingan antara anotasi manual dengan algoritma sebagai berikut:

1. Penentuan *node* paling efektif yang dilewati dari titik awal (*start node*) menuju titik akhir (*end node*) secara anotasi manual berdasarkan persepsi dari manusia.
2. Pengecekan setiap percobaan yang dilewati oleh robot apakah sesuai dengan *node* pemilihan dari anotasi manual, jika *node* yang dilewati sesuai dengan anotasi manual atau bahkan lebih efektif dari anotasi manual maka percobaan dianggap berhasil, jika tidak maka percobaan dianggap gagal.

3. Melakukan perhitungan akurasi dengan melihat jumlah keberhasilan percobaan robot terhadap anotasi manual yang direpresentasikan pada persamaan 3.6.

$$\sum_{i=0}^n akurasi = \left(\frac{\sum_{i=0}^n keberhasilan}{\sum total_{anotasi}} \right) * 100 \quad (3.6)$$

dengan $\sum_{i=0}^n keberhasilan$ adalah jumlah percobaan keberhasilan setiap *node* indeks ke 0 hingga indeks ke n, dan $\sum total_{anotasi}$ menyatakan total anotasi manual terhadap *node* yang seharusnya dilewati oleh robot pada setiap indeks.