

BAB II

KECERDASAN-BUATAN ROBOT PENCARI JALUR

2.1 KECERDASAN-BUATAN ROBOT

Kecerdasan-buatan (*Artificial Intelligence* atau AI) didefinisikan sebagai kecerdasan yang ditunjukkan oleh suatu entitas buatan. Sistem seperti ini umumnya dianggap komputer. Kecerdasan diciptakan dan dimasukkan ke dalam suatu mesin (komputer) agar dapat melakukan pekerjaan seperti yang dapat dilakukan manusia. Beberapa macam bidang yang menggunakan kecerdasan-buatan antara lain sistem pakar, permainan komputer (*games*), logika *fuzzy*, jaringan syaraf tiruan dan robotika [6].

Aplikasi AI dalam kontrol robotik diilustrasikan oleh Gambar 2.1.



Gambar 2.1. Kontrol robot loop tertutup berbasis AI [7]

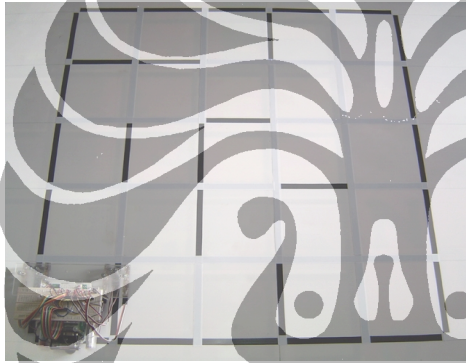
Penggunaan AI dalam kontroler dilakukan untuk mendapatkan sifat dinamik kontroler “secara cerdas”. Secara klasik, kontrol P, I, D atau kombinasi, tidak dapat melakukan adaptasi terhadap perubahan dinamik sistem selama operasi karena parameter P, I dan D itu secara teoritis hanya mampu memberikan efek kontrol terbaik pada kondisi sistem yang sama ketika parameter tersebut di-*tune*. Di sinilah kemudian dikatakan bahwa kontrol klasik ini “belum cerdas” karena belum mampu mengakomodasi sifat-sifat nonlinieritas atau perubahan-perubahan dinamik, baik pada sistem robot itu sendiri maupun terhadap perubahan beban atau gangguan lingkungan [7].

Gambar 2.1 mengilustrasikan tentang skema AI yang digunakan secara langsung sebagai kontroler sistem robot. Dalam aplikasi lain, AI juga dapat digunakan untuk membantu proses identifikasi model dari sistem robot, model lingkungan atau gangguan, model dari tugas robot (task) seperti membuat rencana

trajektori, dan sebagainya. Dalam hal ini konsep AI tidak digunakan secara langsung (*direct*) ke dalam kontroler, namun lebih bersifat tak langsung (*indirect*).

2.2 PERKEMBANGAN ROBOT

Istilah "robot" muncul pertama kali pada Czechoslovakian *satirical play*, *Rossum's Universal Robots*, oleh Karel Capek pada tahun 1920. Robot pada pementasan ini cenderung berperilaku seperti manusia (*human-like*). Berangkat dari hal tersebut, terlihat beberapa cerita fiksi ilmiah melibatkan robot dengan emosi manusia dalam masyarakat. Hal tersebut berubah ketika General Motors memasang robot pertamanya di pabrik manufakturnya pada tahun 1961. Mesin-mesin otomatis ini merepresentasi image yang seluruhnya berbeda dari robot berbentuk manusia (*human form*) dari cerita fiksi ilmiah [8].



Gambar 2.2a. Robot Pencari Jalur



Gambar 2.2b. Robot Bedah [1]

Ketika para pencipta robot pertama kali mencoba meniru manusia dan hewan, mereka menemukan bahwa hal tersebut sangatlah sulit; membutuhkan tenaga penghitungan yang jauh lebih banyak dari yang tersedia pada masa itu. Jadi, penekanan perkembangan diubah ke bidang riset lainnya. Robot sederhana beroda digunakan untuk melakukan eksperimen dalam tingkah laku, navigasi, dan perencanaan jalur. Teknik navigasi tersebut telah berkembang menjadi sistem kontrol robot otonom yang tersedia secara komersial; contoh paling mutakhir dari sistem kontrol navigasi otonom yang tersedia sekarang ini termasuk sistem navigasi berdasarkan-laser dan VSLAM (*Visual Simultaneous Localization and Mapping*) dari ActivMedia Robotics dan Evolution Robotics [1].

Ketika para teknisi siap untuk mencoba robot berjalan kembali, mereka mulai dengan heksapoda dan platform berkaki banyak lainnya. Robot-robot tersebut

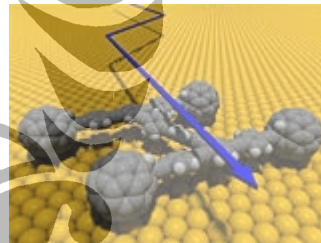
meniru serangga dan arthropoda dalam bentuk dan fungsi. Tren menuju jenis badan tersebut menawarkan fleksibilitas yang besar dan terbukti dapat beradaptasi dengan berbagai macam lingkungan, tetapi biaya dari penambahan kerumitan mekanikal telah mencegah pengadopsian oleh para konsumen. Dengan lebih dari empat kaki, robot-robot ini stabil secara statis yang membuat mereka bekerja lebih mudah. Tujuan dari riset robot berkaki dua adalah mencapai gerakan berjalan menggunakan gerakan pasif-dinamik yang meniru gerakan manusia.

Perkembangan hebat telah dibuat dalam robot medis, dengan dua perusahaan khusus, Computer Motion dan Intuitive Surgical, yang menerima pengesahan pengaturan di Amerika Utara, Eropa, dan Asia atas robot-robotnya untuk digunakan dalam prosedur pembedahan minimal.

Tempat lain di mana robot disukai untuk menggantikan pekerjaan manusia adalah dalam eksplorasi laut dalam dan eksplorasi antariksa. Untuk tugas-tugas ini, bentuk tubuh artropoda umumnya disukai. Mark W. Tilden dahulunya spesialis Laboratorium Nasional Los Alamos membuat robot murah dengan kaki bengkok tetapi tidak menyambung, sementara orang lain mencoba membuat kaki kepiting yang dapat bergerak dan tersambung penuh.



Gambar 2.2c. Robot *Vacuum Cleaner* [1]



Gambar 2.2d. *Nanocar* dari Molekul Tunggal [1]

Robot bersayap eksperimental dan contoh lain mengeksploitasi biomimikri juga dalam tahap pengembangan dini. Yang disebut "nanomotor" dan "kawat cerdas" diperkirakan dapat menyederhanakan daya gerak secara drastis, sementara stabilisasi dalam penerbangan nampaknya cenderung diperbaiki melalui giroskop yang sangat kecil. Dukungan penting pekerjaan ini adalah untuk riset militer teknologi pemata-mataan.

2.3 MIKROKONTROLER BASIC STAMP

Mikrokontroler adalah sistem mikroprosesor lengkap yang terkandung di dalam sebuah *chip*. Mikrokontroler berbeda dari mikroprosesor serba-guna (*general-purpose microprocessor*) yang digunakan dalam sebuah PC, karena sebuah mikrokontroler umumnya telah berisi komponen pendukung sistem minimal mikroprosesor, yakni memori dan antarmuka I/O [9].

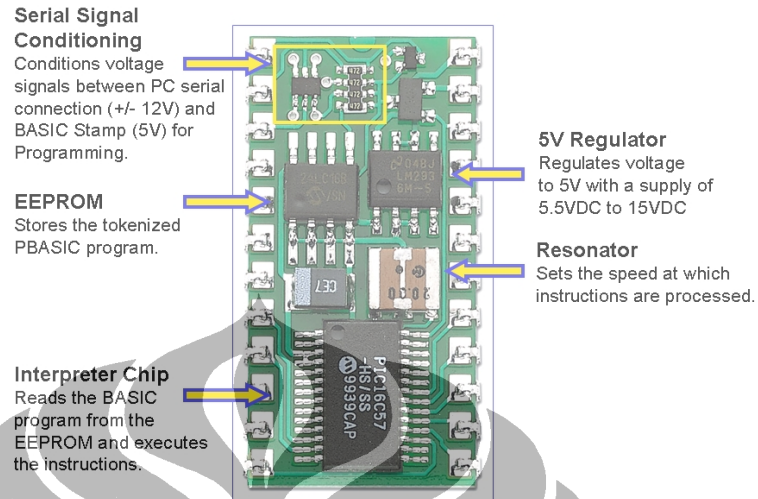
Berbeda dengan mikroprosesor serba-guna, mikrokontroler tidak selalu memerlukan memori eksternal, sehingga mikrokontroler dapat dibuat lebih murah dalam kemasan yang lebih kecil dengan jumlah *pin* yang lebih sedikit.

Sebuah *chip* mikrokontroler umumnya memiliki fitur:

1. *Central Processing Unit* (CPU) - mulai dari prosesor 4-bit yang sederhana hingga prosesor kinerja tinggi 64-bit.
2. Masukan/Keluaran (I/O) antarmuka jaringan seperti port serial (UART)
3. Antarmuka komunikasi serial lain seperti I²C, *Serial Peripheral Interface and Controller Area Network* untuk sambungan sistem
4. *Peripheral* seperti *timer* dan *watchdog*
5. RAM untuk penyimpanan data
6. ROM, EPROM, EEPROM atau memori *flash* untuk menyimpan program komputer
7. Pembangkit *clock* - biasanya berupa resonator rangkaian RC
8. Pengubah analog - ke - digital

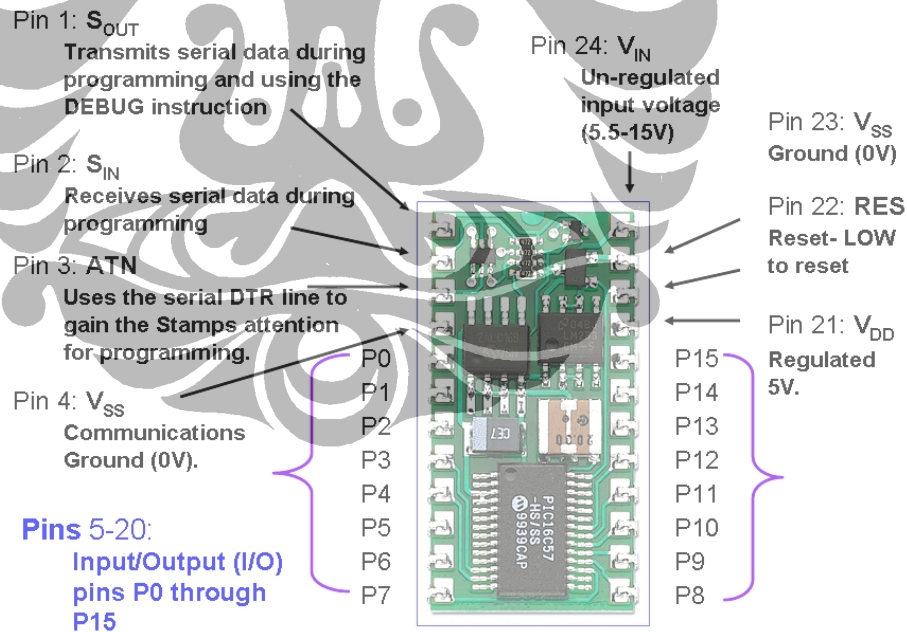
Modul BASIC Stamp dari Parallax merupakan mikrokontroler dengan *chip interpreter* BASIC, memori internal (RAM dan EEPROM), regulator 5 volt, beberapa *pin* I/O multi fungsi (TTL-level, 0-5 volt), dan set instruksi built-in untuk operasi matematika dan *pin* I/O. Modul BASIC Stamp mempunyai kemampuan untuk menjalankan beberapa ribu instruksi per detik dan dapat diprogram dengan sederhana menggunakan PBASIC, yaitu bahasa pemrograman sejenis BASIC yang sudah dimodifikasi untuk mikrokontroler ini [10].

Beberapa modul internal penyusun mikrokontroler BASIC Stamp bisa dilihat pada Gambar 2.3



Gambar 2.3. Modul Internal BASIC Stamp [11]

Gambar 2.4 memperlihatkan 24 I/O pin yang tersedia untuk peripheral.



Gambar 2.4. Pinout BASIC Stamp [11]

Daftar *Pinout* pada Tabel 2.1 memperlihatkan semua *resource I/O* secara detail. Pada kolom sebelah kiri notasi BS2x-24 merupakan semua modul BS2 24-*pin* berdasarkan gambar sebelumnya (“x” mengacu pada nomor model Basic Stamp pada seri BS2), sedangkan kolom kedua hanya untuk BS2p 40-*pin* [12].

Tabel 2.1. *Resource I/O* pada Basic Stamp

<i>Pin</i> BS2x-24 ¹	<i>Pin</i> BS2p-40	<i>Name</i>	<i>Function</i>
1	1	SOUT	Serial output (to RxD of the PC COM Port)
2	2	SIN	Serial input (from TxD of the PC COM Port)
3	3	ATN	Attention (to DTR of the PC COM Port)
4	4	VSS	Ground (to GND of the PC COM Port)
5-20	5-20	P0-P15	Digital I/O pins
	21-36	X0-X15	Digital I/O pins
21	37	VDD	+ 5 V DC I/O (stabilized)
22	38	RES	Reset I/O
23	39	VSS	Ground
24	40	VIN	+ 5,5 - 12 V DC input (not stabilized)

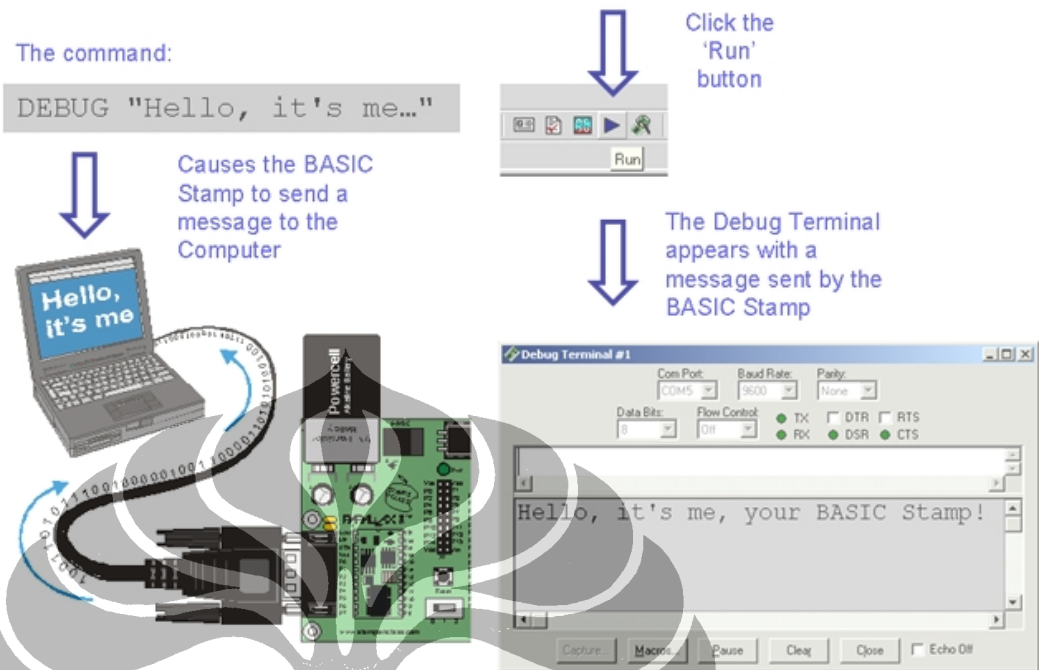
Sumber : Parallax, Inc., USA

Tegangan stabil +5 VDC pada V_{DD} dapat diperoleh dengan menghubungkan 5 sampai +12 VDC ke V_{IN} dan *internal voltage regulator*.

Jika diinginkan untuk memberikan tegangan stabil +5 VDC pada Basic Stamp maka tegangan tersebut dapat dihubungkan secara langsung pada V_{DD} . *Pin* V_{IN} dapat dibuat tidak terhubung dalam hal ini. Modul BASIC Stamp harus disuplai dengan hanya satu tegangan (kemungkinan baterai) dan akan menjalankan program segera setelah di-*download*.

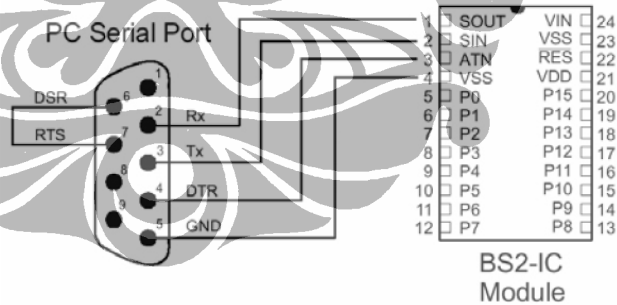
Reset internal (power-down reset) mengakibatkan *pin* RES bernilai *low* selama fasa *reset*.

Gambar 2.5 memperlihatkan infrastruktur pemrograman secara lengkap yang dibutuhkan.



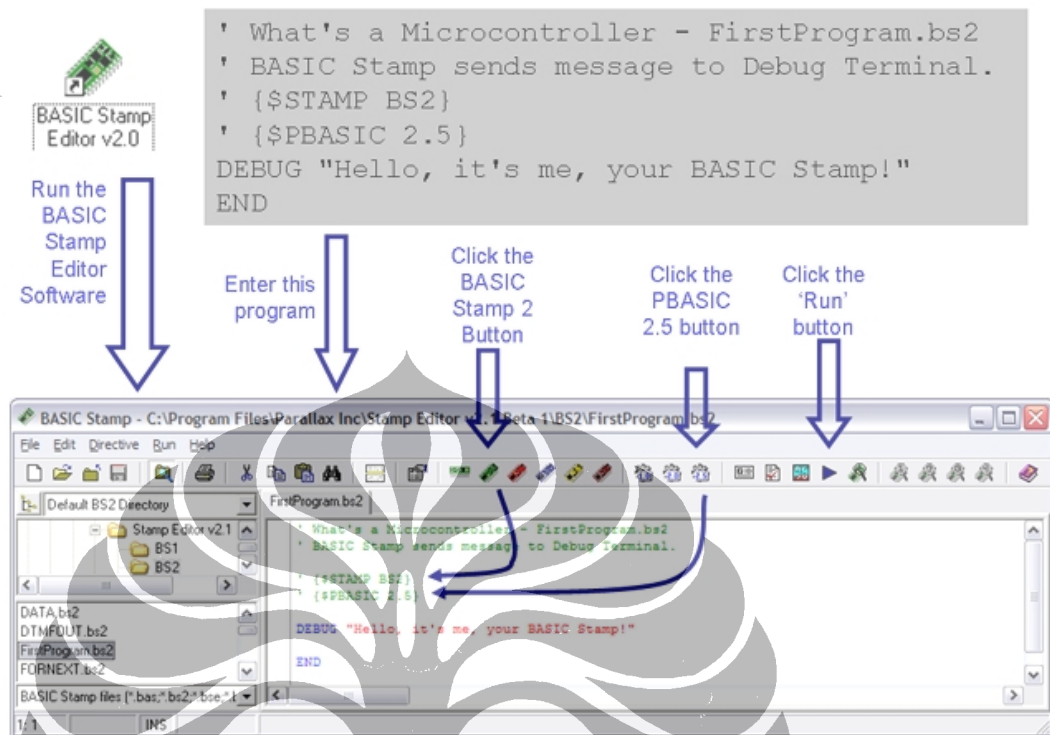
Gambar 2.5. Lingkungan Pengembangan BASIC Stamp [11]

Koneksi kabel *download* dari standard serial cable dengan modul BASIC Stamp bisa dilihat pada Gambar 2.6.



Gambar 2.6. Koneksi *Download* Program dari PC ke BASIC Stamp [12]

Pemrograman mikrokontroler BASIC Stamp dilakukan menggunakan *Editor* BASIC Stamp yang di-*instal* di PC.



Gambar 2.7. Editor Program BASIC Stamp [11]

Jika *source code* PBASIC bebas dari kesalahan, maka *source code* tersebut bisa di-*compile* dan di-*download* ke BASIC Stamp.

Token yang di-*download* akan disimpan di EEPROM eksternal. Mikrokontroler BASIC Stamp mempunyai *firmware* PBASIC yang disebut "*token interpreter*". *Token interpreter* ini bertanggung jawab untuk menjalankan *token* yang di-*download* dan merepresentasikan *core intellectual property* (IP) dari Parallax. Prosedur *download* sama untuk semua jenis BS2.

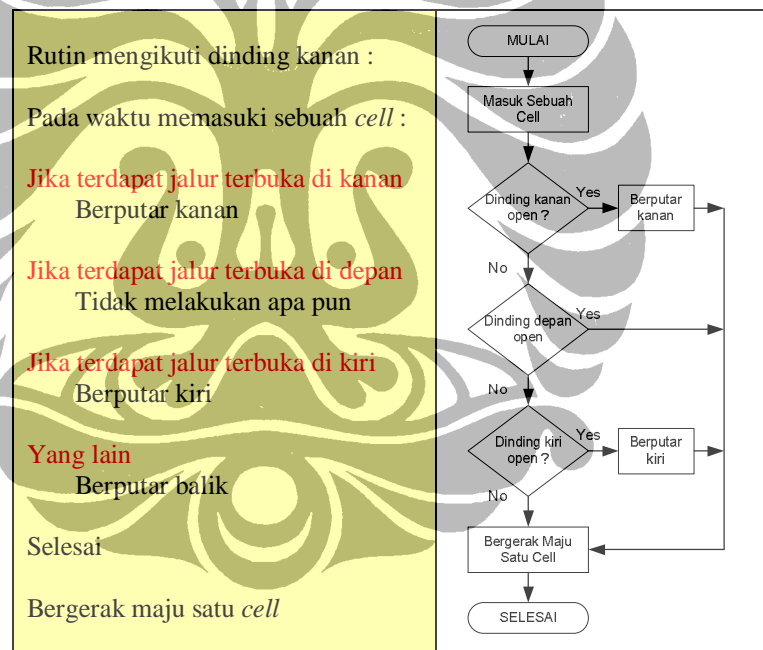
2.4 ALGORITMA PENCARI JALUR

Beberapa algoritma yang digunakan dalam pencarian jalur di lingkungan labirin, yaitu: *Wall Follower*, *Depth First Search*, *Flood-Fill*, dan *modified Flood-Fill*.

Penelitian ini mencoba mengimplementasi secara spesifik dua algoritma terakhir dalam kecerdasan-buatan robot pencari jalur. Uraian lebih detail mengenai kedua algoritma tersebut akan disampaikan di Bab 3, sedangkan untuk algoritma yang lain akan diuraikan secara singkat

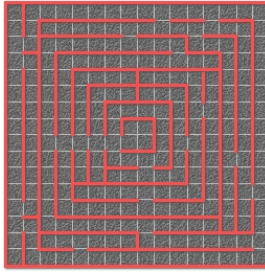
2.4.1 Algoritma *Wall Follower*

Algoritma ini merupakan teknik yang paling sederhana dalam pencarian jalur di lingkungan labirin. Pada dasarnya, robot dengan algoritma ini mengikuti dinding kiri atau kanan sebagai petunjuk di sekeliling labirin.



Gambar 2.8. Algoritma dan Diagram Alir *Wall Follower*

Pada kasus-kasus tertentu, algoritma ini tidak akan bekerja karena robot akan berputar-putar terus di pinggiran labirin (tidak akan bisa mencari jalur menuju ke tengah) seperti ditunjukkan Gambar 2.9.



Gambar 2.9. Lingkungan Labirin di mana Algoritma *Wall Following* Tidak Bekerja

2.4.2 Algoritma *Depth First Search*

Merupakan metode yang intuitif dalam pencarian jalur di lingkungan labirin. Pada dasarnya robot dengan algoritma ini bergerak dan ketika menemukan percabangan, secara acak memilih salah satu jalurnya. Jika jalur ini pada akhirnya buntu, robot ini kembali ke percabangan tadi dan memilih jalur yang lain. Algoritma ini mengakibatkan robot untuk mengeksplorasi setiap kemungkinan jalur di dalam labirin. Dengan mengeksplorasi setiap *cell* di dalam labirin, robot pada akhirnya sampai pada *cell* tujuan di tengah.

Jelas, mengeksplorasi keseluruhan labirin bukan merupakan cara yang efektif dan juga, walaupun menemukan jalur, itu bukan jalur tercepat atau terpendek menuju ke tengah.

2.4.3 Algoritma *Flood-Fill*

Algoritma ini melibatkan pemberian nilai pada masing-masing *cell* penyusun labirin di mana nilai ini merepresentasikan jarak dari sebarang *cell* ke *cell* tujuan. Selanjutnya *cell* tujuan diberikan nilai 0. Jika robot pencari jalur berada di sebuah *cell* dengan nilai 1, robot tersebut 1 *cell* jauhnya dari tujuan. Jika robot berada pada *cell* dengan nilai 3, robot tersebut 3 *cell* jauhnya dari tujuan.

2.4.4 Algoritma *Modified Flood-Fill*

Algoritma ini sama dengan algoritma *Flood-Fill* regular di mana kecerdasan buatan menggunakan nilai jarak untuk bergerak di dalam labirin. Nilai jarak, yang merepresentasi seberapa jauh robot dari *cell* tujuan, diikuti secara menurun (*descending order*) sampai robot mencapai tujuannya.