

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Sebelumnya

Perkembangan robotika di tanah air selama beberapa tahun mengalami peningkatan. Ternyata perkembangan robotika tidak hanya di kalangan mahasiswa, tetapi juga pada kalangan SD, SMP, SMA. Robot yang digunakan adalah robot LEGO yaitu sebuah produk lego yang bisa di rangkai menjadi sebuah robot yang bertujuan menemukan jalur atau jalan keluar pada sebuah labirin.

Berdasarkan penelitian sebelumnya (Anshori, Andrasto 2019) pemecahan labirin yaitu algoritma dijkstra dan metode *maze solver* yang digunakan robot untuk memecahkan lintasan yang dibuat untuk mendapatkan jalur yang tepat.. labirin bisa terdiri dari dinding atau garis, Algoritma ini memberikan opsi untuk berjalan mengikuti dinding kiri atau kanan pada proses penelusuran labirin.

Dalam dunia robotika ada yang dinamakan robot *wall follower*, robot ini juga menggunakan program untuk menyelesaikan tugasnya dalam suatu *maze wall follower*. Pada robot ini program sudah dapat dibuat sedemikian rupa sehingga robot dapat menemukan jalur terpendek. Beberapa metode yang digunakan sebagai program dari robot wall follower di antaranya ada metode simple maze, metode flood fill, dan metode pledge (Shofa, 2015).

2.2 LEGO Mindstorms EV3

LEGO Mindstorms EV3 adalah generasi ketiga dari LEGO. Ini merupakan penerus dari LEGO Mindstorms NXT seri 2.0 generasi kedua. The "EV3" penunjukan berarti bahwa itu ialah evolusi dari seri NXT sebelumnya. Robot LEGO Mindstorms EV3 Secara resmi diumumkan pada tanggal 4 Januari 2013.

Perubahan terbesar dari NXT untuk seri EV3 adalah perbaikan teknologi *brick* yang dapat diprogram. Prosesor utama dari NXT merupakan mikrokontroler ARM7, sedangkan EV3 memiliki prosesor ARM9. EV3 memiliki sebuah konektor USB dan slot Micro SD, serta dilengkapi pemrograman perangkat lunak atau opsional lab VIEW untuk LEGO MINDSTORMS. Berbagai bahasa resmi ada, seperti NXC, NBC leJOS NXJ, dan Robot C (Dewi, 2012).

LEGO Mindstorms EV3 dapat dibangun dan diprogram, robot tersebut bisa melakukan apa yang diinginkan *user*. Satu set perlengkapan pada LEGO EV3 dapat digunakan untuk membangun dan memprogram robot LEGO cerdas, dan membuatnya melakukan banyak operasi yang berbeda. Robot seperti pada Gambar 2.1 dapat dirakit misalnya dengan sensor yang mengontrol motor dan bereaksi terhadap cahaya, sentuhan, suara, dan lain-lain.



Gambar 2.1 Robot LEGO Mindstorms EV3

2.3 Komponen LEGO Mindstorms EV3

2.3.1 EV3 Brick

Brick adalah komponen penting dari robot EV3, karena berfungsi sebagai pengendali (otak dan sumber tenaga robot EV3). Program yang sudah dibuat dapat di-*upload* ke EV3 *Brick* untuk di *compile*.

Spesifikasi teknis dari EV3 *Brick* pada buku pedomannya, yaitu:

1. ARM9 main microprocessor @300MHz
2. LCD *display* 178 x 128 pixel.
3. Bluetooth V2.1
4. Satu port USB 2.0 *interface* memungkinkan untuk konektivitas WiFi.
5. Empat port *input*: port 1, port 2, port 3, dan port 4 yang menghubungkan hingga 4 sensor pada saat yang sama termasuk sensor NXT.
6. Empat port *output*: port A, port B, port C, dan port D yang menghubungkan hingga 4 motor.
7. *Speaker* terintegrasi untuk mengeluarkan output suara.
8. Tiga tombol: kembali, pusat, navigasi(kiri, kanan, atas, bawah).

9. Kompatibel untuk iOS dan Android.

Penggunaan dua prosesor membuat LEGO Mindstroms EV3 dapat menjalankan lebih dari satu *Thread* pada program. Hal ini disebabkan oleh adanya 2 prosesor yang mengerjakan fungsi yang berbeda pada saat bersamaan. *Microcontroller* ARM9 berfungsi sebagai master *controller* yang fungsi utamanya mengatur jalur komunikasi. Fungsi dari *microcontroller* (PMW) untuk mengendalikan empat motor serta *Analog to Digital Converter* (ADC) dari terminal masukan. *Brick* dapat menerima masukan dari empat sensor, dan menjalankan empat motor sekaligus, EV3 *brick* dapat ditunjukkan pada Gambar 2.2(a) dan tampilan layar LCD *brick* dapat ditunjukkan pada Gambar 2.2(b) :



Gambar 2.2(a) EV3 Brick **Gambar 2.2(b)** Tampilan pada Layar LCD Brick

Status cahaya pada *brick* yang mengelilingi *Buttons Brick* memberitahu kita, status keadaan EV3 *Brick* saat ini. Cahaya pada *Buttons Brick* seperti pada Gambar 2.3 dapat menjadi hijau, orange, atau merah. Kode status cahaya *Buttons Brick* adalah sebagai berikut:

1. Merah = *Startup, Updating, Shutdown.*
2. Merah berdenyut = Sibuk.
3. Orange = *Alert, Siap.*
4. Orange berdenyut = *Alert, Menjalankan.*
5. Hijau = *Siap.*
6. Hijau berdenyut = *Menjalankan Program.*



Gambar 2.3 Status Cahaya EV3 Brick

Untuk bagian-bagian yang terdapat pada sisi EV3 brick bisa kita lihat pada gambar-gambar di bawah ini.



Gambar 2.4 EV3 Brick Bagian Atas

Pada bagian *port* PC terdapat mini-USB yang terletak disebelah *port* D, digunakan untuk menghubungkan EV3 Brick ke Komputer. *Port* A, B, C, dan D sebagai *port output* yang digunakan untuk menghubungkan motor ke EV3 Brick. Brick bagian atas dapat dilihat pada Gambar 2.4.



Gambar 2.5 EV3 Brick Bagian Bawah

Untuk bagian bawah EV3 Brick terdapat *port* 1, 2, 3 dan 4 sebagai *port input* yang digunakan untuk menghubungkan sensor dengan EV3 Brick. Tampilan pada sisi Brick bagian bawah dapat dilihat pada Gambar 2.5.



Gambar 2.6 EV3 Brick Bagian Kanan

Pada bagian sebelah kanan EV3 Brick terdapat Speaker yang berfungsi sebagai *output* suara yang digunakan dalam pemrograman robot. Tampilan Brick bagian sebelah kanan dapat dilihat pada Gambar 2.6.



Gambar 2.7 EV3 Brick Bagian Kiri

Pada bagian *port USB Host* dapat digunakan untuk menambahkan *USB Wi-Fi dongle* untuk menghubungkan ke jaringan nirkabel, atau untuk menghubungkan dua hingga empat EV3 Brick secara bersamaan. *Port SD Card* untuk meningkatkan memori yang sudah tersedia pada EV3 Brick dengan *SD Card* (Maksimum 32 GB). Brick bagian kiri dapat dilihat pada Gambar 2.7.

Brick dapat kita ibaratkan seperti CPU pada komputer, yang berfungsi untuk mengolah data. Brick berfungsi untuk mengendalikan jalannya robot sesuai dengan program yang kita buat. Pada pembuatan program dengan EV3 kita dapat melakukan dengan 2 cara :

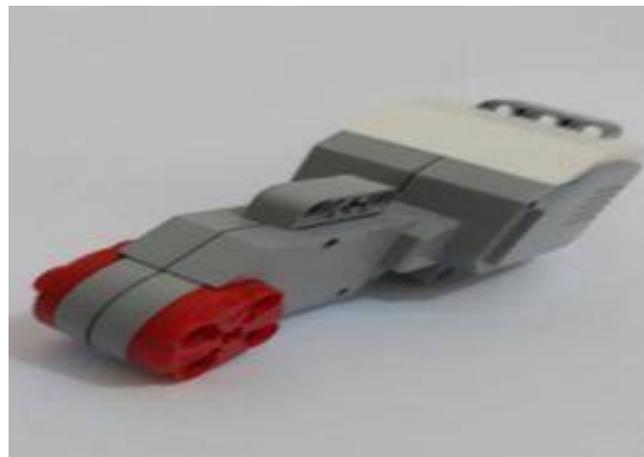
1. Membuat program secara langsung pada EV3 Brick.
2. Membuat program melalui komputer, selanjutnya kita *upload* ke EV3 Brick.

Untuk program-program yang sederhana kita dapat membuatnya secara langsung pada EV3 Brick, sedangkan untuk program-program yang kompleks dan rumit kita dapat membuatnya di komputer terlebih dahulu.

2.3.2 Motor

Motor pada LEGO Mindstroms EV3 mencakup dua jenis motor, Motor *Large* dan Motor *Medium* yang berfungsi untuk menggerakkan bagian robot seperti memutar roda atau menjadi sendi. Satu *brick* bisa dipasang hingga empat buah motor. Motor pada EV3 Mindstroms tidak menggunakan motor DC biasa. Motor DC memiliki keterbatasan dalam hal kemampuan torsi putar, karena motor DC tidak mempunyai umpan balik untuk mengadaptasi beban pada motor DC. (Dewi, 2015).

Pada robot LEGO Motor yang dipakai adalah motor DC servo yang dilengkapi dengan sebuah *encoder* yang berfungsi sebagai umpan balik, sehingga pusat pengendalian dapat memberikan arus yang sesuai dengan beban pada motor. Kecepatan sudut maksimum motor adalah satu putaran per detik. Servo juga dapat digunakan untuk menghitung derajat perputaran atau rotasi. Akurasi dari servo motor mencapai kurang satu derajat. Torsi yang besar yang didapat dalam waktu singkat merupakan kelebihan motor servo. Kekurangan motor servo adalah kurangnya akurasi sehingga diperlukan suatu pengendali yang dapat meningkatkan keakurasian. Gambar 2.8 dan Gambar 2.9 menunjukkan Motor *Large* dan Motor *Medium* LEGO Mindstroms EV3.



Gambar 2.8 Motor *Large*

Motor *Large* merupakan motor kuat dan “cerdas” yang memiliki *built-in* Rotasi Sensor dengan resolusi 1 derajat untuk kontrol yang tepat. Motor *Large* dioptimalkan untuk menjadi basis mengemudi pada robot. Dengan menggunakan

Move Steering atau pindahkan blok pemrograman *Tank* di Software EV3-G, motor besar akan mengkoordinasikan tindakan secara bersamaan.



Gambar 2.9 Motor *Medium*

Motor *medium* juga termasuk *built-in* Rotasi Sensor dengan resolusi satu derajat, tetapi lebih kecil dan lebih ringan. Itu berarti ia mampu merespon lebih cepat. Motor *Medium* dapat diprogram untuk mengaktifkan atau menonaktifkan, mengendalikan tingkat daya, atau untuk menjalankan untuk jumlah waktu tertentu atau rotasi. Dari keterangan dua motor diatas dapat kita bandingkan yakni:

1. Motor *Large* berjalan pada 160-170 rpm, dengan torsi berjalan dari 20 Ncm dan torsi 40 Ncm (lambat, tapi kuat).
2. Motor *Medium* berjalan pada 240-250 rpm, dengan torsi berjalan dari 8 Ncm dan torsi 12 Ncm (lebih cepat, tapi kurang kuat).

2.3.3 Sensor *Infrared* dan *Remote Infrared Beacon*

Sensor *infrared* adalah sensor digital yang dapat mendeteksi cahaya *infrared* yang terpantul dari benda padat. Hal ini juga dapat mendeteksi sinyal cahaya *infrared* yang dikirim dari *Remote Infrared Beacon*. Sensor ini dapat digunakan dalam tiga mode yang berbeda: *mode proximity*, *mode beacon*, dan *mode remote*. Dalam *mode proximity*, sensor *infrared* menggunakan gelombang cahaya yang di pantulkan kembali dari objek untuk memperkirakan jarak antara sensor dan objek dengan laporan jarak menggunakan nilai antara 0 (sangat dekat) dengan 100 (jauh), bukan dengan *centimeter* atau *inci*. Sensor dapat mendeteksi objek sampai 70cm, tergantung pada ukuran dan bentuk objek (Sunarsih, 2017).



Gambar 2.10 Sensor *Infrared*

Pada *mode Beacon*, salah satu dari *Remote Infrared Beacon* empat saluran dari *Channel Selector* merah. Sensor *Infrared* akan mendeteksi sinyal yang cocok dengan *channel* yang di tetapkan dalam program sampai dengan jarak sekitar 200 cm kearah depan. Setelah terdeteksi, sensor bisa memperkirakan arah dan jarak (*proximity*) ke *beacon*. Dengan informasi ini, terdapat nilai antara -25 dan 25, dengan 0 menunjukkan bahwa sinyal tersebut secara langsung di depan sensor *infrared* dengan kedekatan nilai 0 dan 100. Sensor *infrared* dapat dilihat pada Gambar 2.11.



Gambar 2.11 *Remote Infrared Beacon*

Remote Infrared Beacon adalah perangkat terpisah yang dapat di genggam atau dibangun ke dalam model LEGO. Hal ini membutuhkan dua baterai AAA. Untuk mengaktifkan *Remote Infrared Beacon on*, tekan tombol *mode Beacon* di bagian atas perangkat. Indikator LED hijau akan menyala, menunjukkan perangkat aktif dan transmisi terus menerus. Tekan lagi tombol *mode Beacon* akan mematikannya (setelah satu jam tidak aktif, sinyal secara otomatis akan mati). *Remote infrared beacon* dapat dilihat pada Gambar 2.12.

Mode remote, *Remote Infrared Beacon* sebagai *remote control* untuk robot. Ketika di *remote Mode*, sensor *Infrared* dapat mendeteksi tombol (tombol kombinasi) pada sinyal di tekan. Ada sebelas tombol kombinasi pada *Remote Infrared Beacon* :

- 0 = Tidak ada tombol (*Mode off*)
- 1 = Tombol 1
- 2 = Tombol 2
- 3 = Tombol 3
- 4 = Tombol 4
- 5 = Baik Tombol 1 dan Tombol 3
- 6 = Kedua Tombol 1 dan Tombol 4
- 7 = Tombol 2 dan Tombol 3
- 8 = Tombol 2 dan Tombol 4
- 9 = *Beacon Mode on*
- 10 = Kedua Tombol 1 dan Tombol 2
- 11 = Tombol 3 dan Tombol 4

2.3.4 Sensor Touch

Sensor Touch adalah sensor digital yang dapat mendeteksi cahaya *infrared* yang tercermin dari benda padat. Hal ini juga dapat mendeteksi benda yang disentuhnya dan mengatur jarak untuk berhenti, maju, mundur dan berbelok kiri maupun kanan



Gambar 2.12 Sensor Touch

2.3.5 Konektor

Sensor dihubungkan ke EV3 *Brick* menggunakan suatu 6-position modular *connector* yang mengutamakan kedua antarmuka digital dan analog. Antarmuka yang analog adalah *backward-compatible* (dengan menggunakan suatu adapter) dengan *Robotics Invention System* yang lama. Antarmuka yang digital mampu untuk kedua komunikasi I2C dan RS-485 (Nugraha, 9, 2017).



Gambar 2.13 Konektor

Tabel 2.1 EV3 Sensor *Interface pin-out*

Pin	Name	Function	Color
1	ANALOG	Analog interface, +9V Supply	White
2	GND	Ground	Black
3	GND	Ground	Red
4	IPOWERA	+4.3V Supply	Green
5	DIGIAI0	I2C Clock (SCL),RS-485 B	Yellow
6	DIGIAI1	I2C Data (SDA), RS-485 A	Blue

2.3.6 Komponen Tambahan

Selain komponen utama, ada juga komponen tambahan untuk membuat robot LEGO Mindstorms EV3. Komponen-komponen yang terdapat di robot LEGO Mindstorms EV3 secara lengkap, dapat dilihat pada Gambar 2.13.



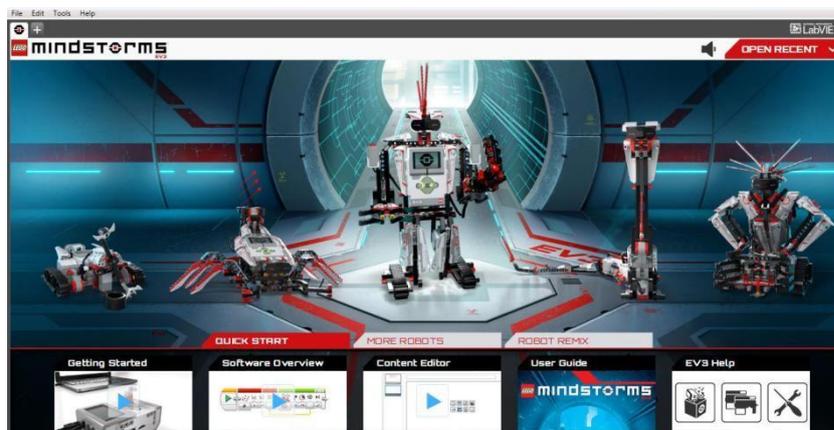
Gambar 2.12 Komponen Tambahan EV3

2.4 Program LEGO Mindstorms EV3

Menurut Sunarsih (2017:15) Untuk menjalankan robot EV3, kita harus memprogram robot tersebut dengan algoritma yang kita inginkan. Ada banyak bahasa pemrograman yang dapat digunakan untuk memprogram EV3, salah satunya adalah LEGO Mindstorms EV3 *Home Edition*.

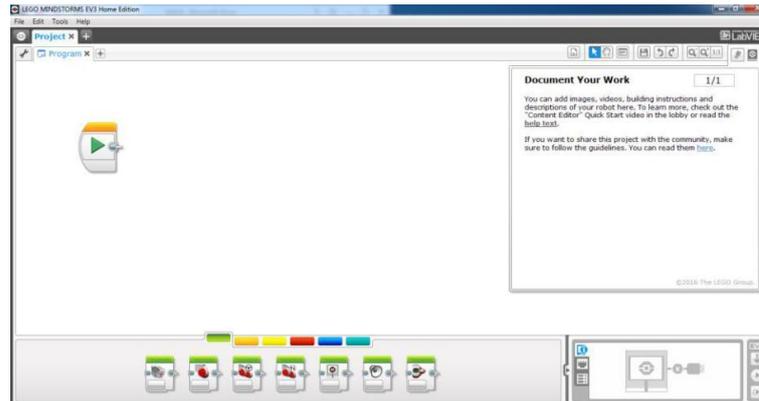
LEGO Mindstorms EV3 *Home Edition* adalah *software* untuk memprogram EV3 *Brick* dari komputer yang dapat dilakukan secara grafikal. *Software* ini menggunakan *Icon-Based* sehingga mempermudah untuk memprogram robot yang dirancang. Selain dapat memprogram melalui PC / Laptop, kita juga bisa memprogram robot LEGO Mindstorms EV3 dari ponsel / tablet.

Dalam program LEGO Mindstorms EV3 *Home Edition*, layar ditampilkan di waktu *startup* disebut *Lobby* seperti pada Gambar 2.14. Isi *Lobby* adalah menu untuk mengakses setiap fungsi dari program LEGO Mindstorms EV3 *Home Edition*.



Gambar 2.13 LEGO Mindstorms EV3 *Home Edition* Lobby

Lembar *Project* adalah halaman yang digunakan untuk membuat program dengan menggunakan blok pemrograman, seperti pada Gambar 2.15.

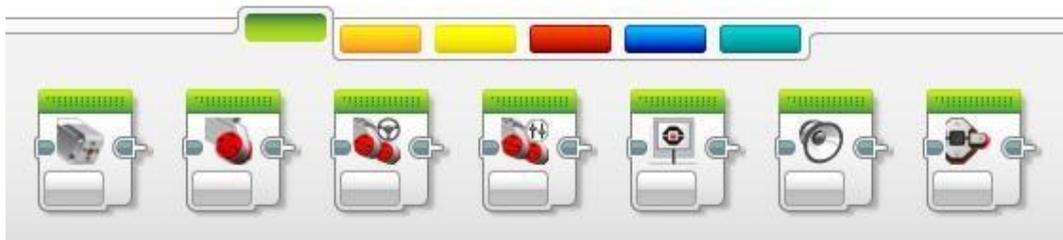


Gambar 2.14 Lembar *Project*

2.4.1 *Programming Blocks and Palettes*

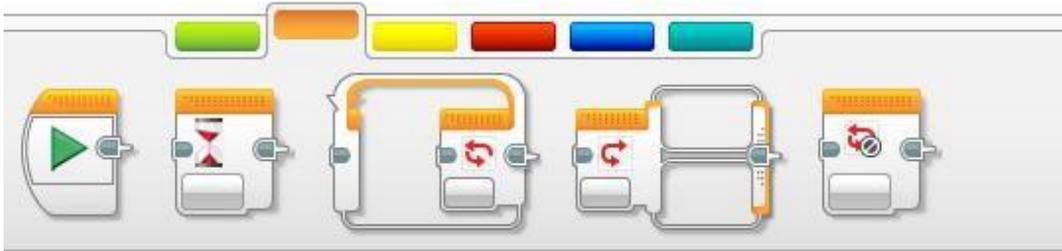
Semua blok pemrograman yang digunakan untuk mengendalikan robot berada di *Programming Palettes* pada bagian bawah *Programming Canvas*. Blok Pemrograman dibagi ke dalam kategori sesuai dengan jenis dan sifat, sehingga mudah untuk menemukan blok yang dibutuhkan.

Untuk sekilas video pemrograman, bisa dilihat di bagian *Quick Start* dari *Lobby* dan juga dapat menemukan informasi lebih lanjut tentang bagaimana program di teks help pada EV3-G. Pada *Programming Palettes* terdapat blok program sebagai berikut:



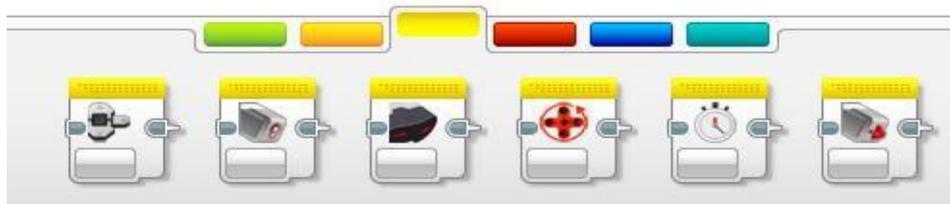
Gambar 2.15 *Action Blocks*

Pada *Action Blocks* terdapat block program untuk Motor *Medium*, Motor *Large*, *Move Steering*, *Move Tank*, *Display*, *Sound*, *Brick Status Light*.



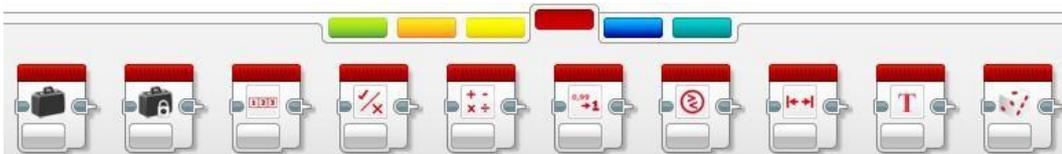
Gambar 2.16 *Flow Control*

Bagian *Flow Control* berisikan *block Start, Wait, Loop, Switch, Loop Interrupt*. *Block* ini biasa digunakan untuk memprogram robot.



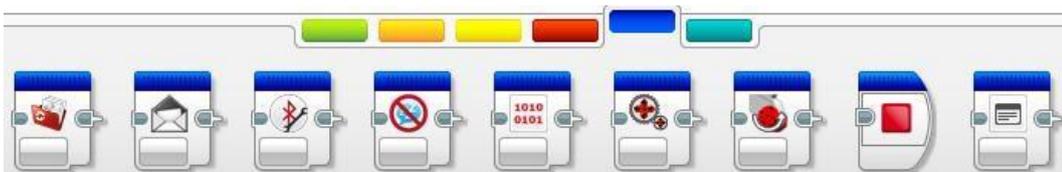
Gambar 2.17 *Block Sensor*

Pada *block Sensor* terdapat *block Brick Buttons, Sensor Colour, Sensor Gyro, Sensor Infrared, Motor Rotation, Sensor Temperature, Timer, Sensor Touch, Sensor Ultrasonic, Energy Meter, Sensor Sound NXT*.



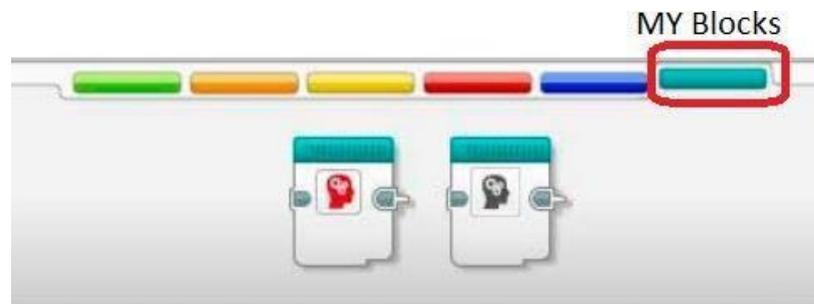
Gambar 2.18 *Data Operations*

Dibagian *Data Operations* berisikan *block Variable, Constant, Array Operations, Logic Operations, Math, Round, Compare, Range, Text, Random*.



Gambar 2.19 *Advance*

Pada *Advanced* terdapat block *File Access*, *Data Logging*, *Messaging*, *Bluetooth Connection*, *Keep Awake*, *Raw Sensor Value*, *Unregulated Motor*, *Invert Motor*, *Stop Program*.



Gambar 2.20 *My Blocks*

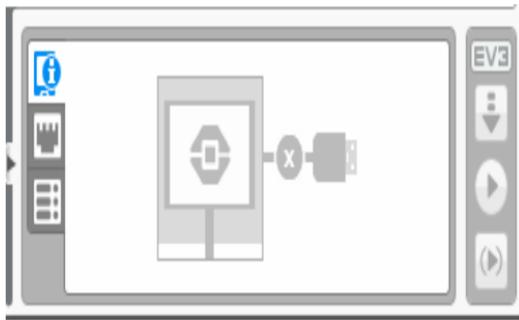
My Blocks berfungsi ketika kita berulang kali menggunakan bagian yang sama dari sebuah program di banyak program, itu merupakan saat yang tepat untuk membuat *My Blocks*.

2.4.2 *Data Logging*

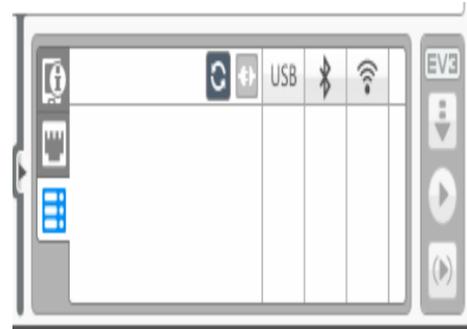
EV3-G tidak hanya merekam data, tetapi dapat membantu kita untuk mengatur dan menganalisanya. *Data Logging* terdiri dari bidang utama sebagai berikut:

1. *Graph Area* — Untuk melihat dan menganalisa plot data.
2. *Configuration Panel* — Disini kita menyiapkan eksperimen, mengelola dataset, dan mendapatkan akses ke Dataset Perhitungan dan Grafik Programming, fungsi data logging khusus.
3. *Hardware Page* — Untuk membangun dan mengelola komunikasi dengan EV3 Brick, melihat posisi motor dan sensor yang terhubung. *Hardware Page* tempat untuk men-download program ke EV3 Brick. Tombol pada pengendali *Hardware Page* memiliki fungsi sebagai berikut:
 - *Download* — Untuk men-download program ke EV3 Brick.
 - *Download and Run* — Men-download program EV3 Brick dan segera menjalankan program.
 - *Download and Run Selected* — Men-download hanya blok yang disorot ke EV3 Brick dan segera menjalankan program.

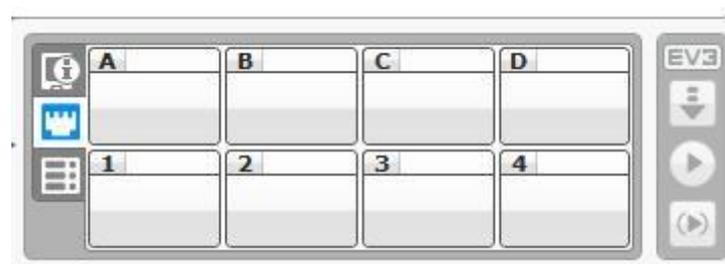
- *Upload* –Meng-*Upload* dataset yang dikumpulkan dari EV3 Brick ke Percobaan.



Gambar 2.21(a) *Graph Area*



Gambar 2.21(b) *Configuration Panel*



Gambar 2.21(c) *Hardware Page*

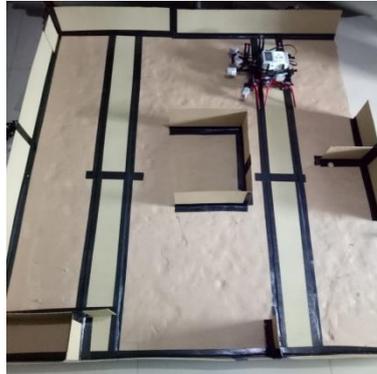
Pada Gambar 2.21(a) di sini kita mempersiapkan sebuah eksperimen dengan memilih durasi sampel, menilai dan memilih sensor apa yang digunakan untuk mengumpulkan data. Untuk setiap sensor, kita juga harus memilih mode sensor, karena sebagian besar sensor dapat memberikan berbagai jenis data.

Pada Gambar 2.21(b) dapat memanipulasi *dataset* dan grafik dengan melakukan berbagai perhitungan dengan menggunakan angka, fungsi, dan dataset lainnya. Hasil *output* akan ditampilkan berupa sebuah grafik dan nilai-nilai dataset.

Pada Gambar 2.21(c) membagi bidang grafik kita dalam zona yang berbeda, kita dapat mengaktifkan *input* sensor untuk memicu atau mengaktifkan *output*: misalnya, motor atau suara dari EV3 Brick ketika nilai-nilai sensor mencapai tingkat atau ambang tertentu.

2.5 *Maze atau Labirin*

Maze atau Labirin merupakan sebuah sistem jalur yang rumit, berliku-liku, serta memiliki banyak jalan buntu. Tujuan permainan ini adalah pemain harus menemukan jalan keluar dari sebuah pintu masuk ke satu atau lebih pintu keluar .



Gambar 2.22 Labirin.

2.6 *Wall Following Robot*

Wall Following Robot adalah robot yang dapat bergerak atau berpindah tempat dengan menggunakan roda atau tiruan bentuk kaki (Suradana dan Sudiarsa,2013:96). Untuk mendeteksi penghalang atau dinding, robot dilengkapi dengan sensor infra merah yang diletakkan diujung depan dan menghadap ke samping dari robot tersebut. Sensor *infrared* terdiri dari led *infrared* sebagai pemancar dan fototransistor sebagai penerima cahaya infra merah, lalu cahaya akan dipantulkan oleh penghalang atau dinding dan kemudian akan dibaca oleh fototransistor sebagai penerima cahaya. Jika fototransistor menerima cahaya, itu berarti sensor cahaya telah berhasil membaca adanya penghalang atau dinding, dan robot akan mulai mengikuti dinding tersebut.

Prinsipnya adalah sifat dari penghalang atau dinding dengan mengatur jarak dinding dengan robot tetap konstan. Bila terjadi perubahan, maka robot akan bergerak untuk kemudian menyesuaikan jarak lagi (Erlina, Tati. Laksono, H.D. Ari, A.S, 2014).

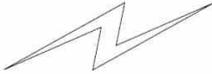
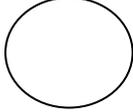
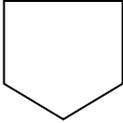
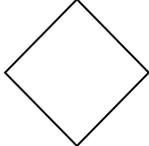
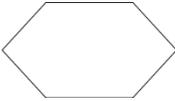
2.7 *Flowchart*

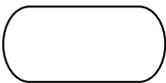
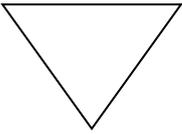
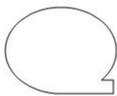
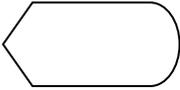
Flowchart adalah representasi secara simbolik dari suatu algoritma atau prosedur untuk menyelesaikan suatu masalah, dengan menggunakan *flowchart* akan memudahkan pengguna melakukan pengecekan bagian-bagian yang

terlupakan dalam analisis masalah, disamping itu *flowchart* juga berguna sebagai fasilitas untuk berkomunikasi antara pemrogram yang bekerja dalam tim suatu proyek (Sunarsih, 2017).

Berikut simbol-simbol yang digunakan dalam pembuatan *flowchart*.

Tabel 2.2 Simbol-simbol *Flowchart*

NO	SIMBOL	KETERANGAN
1		Simbol arus / <i>flow</i> , menyatakan jalannya arus suatu proses
2		Simbol communication link, menyatakan transmisi data dari satu lokasi ke lokasi lain.
3		Simbol <i>connector</i> , menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama
4		Simbol <i>offline connector</i> , menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda
5		Simbol proses, yaitu menyatakan suatu proses yang dilakukan oleh komputer
6		Simbol <i>manual</i> , menyatakan suatu proses yang tidak dilakukan oleh komputer
7		Simbol <i>decision</i> , menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban yaitu ya atau tidak
8		Simbol <i>predefined process</i> , menyatakan persediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal

9		Simbol terminal, menyatakan awal atau akhir suatu program
10		Simbol <i>keying operation</i> , menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai <i>keyboard</i>
11		Simbol <i>offline-storage</i> , menunjukkan bahwa data dalam simbol ini akan disimpan ke dalam suatu media tertentu
12		Simbol <i>manual input</i> , menyatakan data secara manual dengan menggunakan <i>online keyboard</i>
13		Simbol <i>input / output</i> , menyatakan proses <i>input</i> atau <i>output</i> tanpa tergantung jenis peralatannya
14		Simbol <i>punched card</i> , menyatakan <i>input</i> berasal dari kartu atau <i>output</i> ditulis ke kartu
15		Simbol <i>magnetic tape</i> , menyatakan <i>input</i> berasal dari pita magnetis atau <i>output</i> tersimpan ke dalam pita magnetis
16		Simbol <i>disk storage</i> , menyatakan <i>input</i> berasal dari <i>disk</i> atau <i>output</i> tersimpan kedalam <i>disk</i>
17		Simbol <i>document</i> , mencetak keluaran dalam bentuk dokumen (memulai <i>printer</i>)
18		Simbol <i>display</i> , mencetak keluaran dalam layar <i>monitor</i>