

## **BAB II TINJAUAN PUSTAKA**

### **2.1 Penelitian Terdahulu**

Penelitian terdahulu adalah referensi-referensi yang berkaitan dengan informasi penelitian berupa hasil penelitian-penelitian terdahulu yang dapat dijadikan sebagai bahan acuan dan bahan perbandingan terhadap penelitian yang akan dilakukan. Penulis memilih beberapa hasil penelitian yang berkaitan dengan topik mengenai robot *line follower*, antara lain :

#### **1. Kiki Prawiroredjo dan Ivan Joseph Nurdin (2009) yang berjudul “Robot Pengikut Garis Berbasis Mikrokontroler ATMEGA 8535”**

Dalam penelitian ini, robot dirakit dengan menggunakan mikrokontroler ATMEGA 8535 sebagai pengontrol dari rangkaian elektronik robot dan 2 modul sensor yang terdiri dari modul *infrared sensor* dan modul *photoreflector sensor*. Modul *infrared sensor* terdiri dari 3 *infrared diode* dan 3 *photodiode*, sebuah modul *infrared sensor* dipasang dibagian depan robot untuk mendeteksi halangan dan dua yang lain dipasang di sisi kiri dan kanan robot untuk mendeteksi lintasan alternatif apabila didepan robot terdapat halangan. *Photoreflector sensor* digunakan sebagai navigasi utama robot untuk mengikuti lintasan. *Infrared sensor* pendeteksi halangan dapat mendeteksi objek halangan sampai sejauh 3,5 cm sedangkan jarak maksimum *infrared sensor* pendeteksi jalur lain sebesar 1,5 cm agar dapat mendeteksi jalur lain. *Photoreflector sensor* mampu mendeteksi garis hitam lintasan dengan jarak minimum 0,5 cm dan jarak maksimumnya sebesar 0,8 cm.

Berdasarkan penjelasan tersebut, dapat disimpulkan bahwa:

1. Robot pengikut garis berbasis mikrokontroler ATMEGA 8535 ini dirancang untuk berjalan mengikuti suatu lintasan, tetapi apabila terdapat halangan didepan robot maka robot akan mencari jalur alternatif.
2. Robot dapat mendeteksi objek halangan sejauh 3,5 cm dan jarak pembacaan dari garis hitam lintasan sebesar 0,5 cm sampai 0,8 cm untuk pendeteksi jalur lintasan utama sedangkan untuk pendeteksi jalur alternatif sebesar 1,5 cm.

**2. Ahmad Sadid Jauhari (2017) yang berjudul “Prototipe Robot Pengangkut Barang Berbasis *Line Follower* Menggunakan Arduino UNO Dan *Load Cell*”**

Robot yang dibuat digunakan untuk mengangkut barang dan masih dalam bentuk *prototype* sehingga berat barang dibatasi. Komponen-komponen yang digunakan dalam pembuatan robot ini diantaranya menggunakan Arduino UNO sebagai pengontrol robot, sensor *photodiode* dan LED yang digunakan dalam pembacaan pola garis serta sensor berat yaitu *load cell* yang digunakan untuk menentukan berat benda yang akan diangkut. Beban yang digunakan diantaranya 250 gram, 500 gram dan 750 gram. Robot akan mengangkut barang ke titik A jika diberi beban 250 gram, sedangkan ketika diberi beban 500 gram robot akan mengangkut barang ke titik B, dan jika diberikan beban 750 gram robot akan mengangkut beban ke titik C. Waktu tempuh *finish* ke *start* lebih besar dibandingkan waktu *start* ke *finish*. Hal ini dikarenakan pada proses pengambilan data dari *finish* ke *start* terdapat tiga aksi yang dilakukan robot diantaranya putar balik robot dari titik *finish*, perjalanan robot menuju titik *start* dan putar balik robot di titik *start*. Sedangkan proses pengambilan data dari *start* ke *finish* hanya terdapat satu kali aksi yakni saat robot mulai berjalan setelah diberi beban sampai robot berhenti di titik *finish*.

Berdasarkan penjelasan tersebut, dapat disimpulkan bahwa:

1. Robot ini dirancang untuk mengangkut barang dengan beban tertentu dimana berat beban akan mempengaruhi ke arah mana robot akan mengangkut beban tersebut.
2. Waktu tempuh dari *finish* ke *start* lebih besar dari waktu tempuh dari *start* ke *finish*.

**3. Dian Agus Prayitno (2015) yang berjudul “Rancang Bangun Robot Pengantar Makanan *Line Tracer* Berbasis Mikrokontroler ATMEGA 16”**

Robot pengantar makanan *line tracer* berbasis mikrokontroler ATMEGA 16 terdiri dari mikrokontroler ATMEGA 16 untuk mengontrol robot, sensor *photodiode* dan LED untuk sebagai sistem navigasi robot, *push button* sebagai input untuk menentukan ke meja mana makanan akan diantarkan. Serta motor servo yang berfungsi sebagai lengan robot untuk membawa makanan. *Push button* ditekan untuk menentukan nomor meja, jika meja a yang dipilih maka robot akan mengantar makanan ke meja a, sedangkan jika meja b yang dipilih maka robot akan mengantar makanan ke meja b, apabila meja c yang dipilih maka robot akan mengantar makanan ke meja c dan jika meja d yang dipilih maka robot akan mengantar makanan ke meja d.

Berdasarkan penjelasan tersebut, dapat disimpulkan bahwa:

1. Robot pengantar makanan ini dapat mengantarkan makanan ke meja sesuai dengan input yang dimasukkan dengan menggunakan metode *line tracer* atau *line follower*.
2. Rangkaian yang dibuat menggunakan mikrokontroler ATMEGA 16, *photodiode*, LED, *push button* dan motor servo. Bahasa pemrograman yang digunakan untuk memprogram robot pengantar makanan ini adalah bahasa C.

**Tabel 2.1** Persamaan dan perbedaan dengan penelitian terdahulu

No	Penelitian	Persamaan	Perbedaan
1	Kiki Prawiroredjo dan Ivan Joseph Nurdin. 2009. <i>Robot Pengikut Garis Berbasis Mikrokontroler ATMEGA 8535</i>	-	1. Menggunakan ATMEGA 8535 sebagai mikrokontroler 2. Menggunakan <i>infrared sensor</i> dan <i>photorelector sensor</i> sebagai pendeteksi lintasan

2	Ahmad Sadid Jauhari. 2017. <i>Prototipe Robot Pengangkut Barang Berbasis Line Follower Menggunakan Arduino UNO Dan Load Cell</i>	-	<ol style="list-style-type: none"> <li>1. Menggunakan Arduino UNO sebagai mikrokontroler</li> <li>2. Menggunakan <i>load cell</i> sebagai sensor berat</li> <li>3. Menggunakan <i>photodiode</i> dan LED pendeteksi lintasan</li> </ol>
3	Dian Agus Prayitno. 2015. <i>Rancang Bangun Robot Pengantar Makanan Line Tracer Berbasis Mikrokontroler ATMEGA 16</i>	1. Sama-sama robot pengantar makanan	<ol style="list-style-type: none"> <li>1. Menggunakan ATMEGA 16 sebagai mikrokontroler</li> <li>2. Menggunakan motor servo sebagai lengan robot</li> <li>3. Menggunakan <i>photodiode</i> dan LED pendeteksi lintasan</li> </ol>

## 2.2 Pengertian, Struktur dan Jenis Robot

### 2.2.1 Pengertian Robot

Robot adalah mesin hasil rakitan manusia yang bisa bekerja tanpa mengenal lelah. Awalnya robot diciptakan untuk menggantikan tenaga manusia (Suyadhi, 2008). Hal ini dapat dilihat dari berbagai inovasi yang melibatkan teknologi robot yang perlahan mulai menggantikan pekerjaan yang dulunya biasanya dilakukan oleh manusia contohnya robot industri, robot resepsionis, hingga robot koki.

Robot merupakan suatu hasil dari kemajuan teknologi berupa mesin yang bentuknya bermacam-macam misalnya robot berbentuk kendaraan, hewan, bahkan berbentuk manusia dan dapat melakukan pekerjaan-pekerjaan manusia. Tujuan pembuatan robot diantaranya untuk mempermudah pekerjaan manusia dikarenakan sifatnya yang dapat diprogram untuk melakukan pekerjaan berat, berulang-ulang

bahkan pekerjaan yang beresiko bagi manusia tanpa mengenal lelah dan tanpa melakukan kesalahan.



**Gambar 2.1** Robot

### 2.2.2 Struktur Robot

Menurut Mega Puspita (2014), pada umumnya sebuah robot memiliki struktur sebagai berikut :

1. *Manipulator*

*Manipulator* memiliki dua bagian dasar, yaitu bagian dasar dan bagian tambahan. Bagian dasar merupakan bentuk dasar robot atau badan robot sedangkan bagian tambahan merupakan bagian mekanik yang menunjukkan pergerakan robot atau lengan robot, contohnya badan (*body*), pergelangan (*wrist*), lengan (*arm*).

2. *End Effector*

*End Effector* merupakan alat yang berfungsi untuk menghubungkan antara bagian *manipulator* dengan benda yang akan dijadikan objek kerja oleh robot, contohnya penjepit (*gripper*), bor, penyemprot cat dll.

3. Penggerak (*Driver*)

Penggerak (*driver*) biasanya digunakan oleh robot untuk berpindah posisi dari satu tempat ke tempat lainnya atau mengangkat beban pada bagian *end effector*, contohnya motor, penggerak hidrolik, penggerak pneumatik.

#### 4. Sensor

Sensor digunakan robot untuk mengetahui informasi tentang berbagai keadaan disekitar robot, contohnya sensor jarak, sensor panas, sensor kecepatan, sensor posisi, dll.

#### 5. Pengendali (*Controller*)

Pengendali berfungsi sebagai pengatur komponen-komponen yang terdapat pada suatu sistem untuk menjalankan tugas-tugas tertentu, biasanya berupa rangkaian elektronik berbasis mikroprosesor, contohnya PC/komputer, mikrokontroler, mikroprosesor, dll.

### 2.2.3 Jenis-Jenis Robot

Menurut Ahmad Muslim (2013), robot diklasifikasikan kedalam 4 kelompok, yaitu:

#### 1. *Mobile Robot*

*Mobile Robot* merupakan jenis robot dengan ciri khas memiliki aktuator berupa roda atau sejenisnya yang memungkinkan robot untuk dapat bergerak atau berpindah posisi dari satu tempat ke tempat lainnya, contohnya robot *line follower*.

#### 2. *Non-Mobile Robot*

*Non-Mobile Robot* adalah robot yang hanya berdiri pada satu titik tempat dan tidak dapat berpindah posisi dari satu tempat ke tempat lainnya. Untuk menjalankan fungsinya, robot jenis ini dapat menggerakkan beberapa bagian dari tubuhnya sesuai dengan arsitektur robot dan program yang diberikan, contohnya robot *manipulator* berlengan.

#### 3. Gabungan *Mobile Robot* dan *Non-Mobile Robot*

Robot jenis ini merupakan gabungan dari fungsi-fungsi *Mobile Robot* dan *Non-Mobile Robot* sehingga saling melengkapi kekurangan masing-masing, contohnya robot *forklift*.

#### 4. Robot *Humanoid*

Robot *Humanoid* adalah robot yang bentuk fisiknya menyerupai manusia, lengkap dengan anggota tubuh seperti kepala, lengan, kaki, mata, dan pergerakan sendi-sendinya. Bagian-bagian tubuh robot tersebut dapat dapat digerakkan dan memiliki fungsi yang sama dengan aslinya, contohnya robot ASIMO.

#### **2.2.4 Line Follower Robot**

*Line follower robot* adalah robot yang didesain untuk dapat berjalan mengikuti garis yang membentuk sebuah alur *mapping* tertentu, biasanya garis yang dipakai berwarna hitam atau putih (Widodo, 2011). Alur *mapping* atau jalur lintasan juga dapat dibuat dengan menggunakan warna-warna lain, hanya saja hasil pembacaan garisnya tidak sebaik ketika menggunakan garis berwarna hitam berlatar putih maupun sebaliknya sehingga membuat *line follower robot* yang dibuat tidak dapat mengikuti garis lintasan secara stabil.

Prinsip kerja pendeteksi garis pandu dari robot tersebut adalah bahwa tiap-tiap warna permukaan memiliki kemampuan memantulkan cahaya yang berbeda-beda. Warna putih memiliki kemampuan memantulkan cahaya lebih banyak, sebaliknya, warna-warna gelap memiliki lebih sedikit kemampuan memantulkan cahaya. Dengan menentukan seberapa besar intensitas cahaya yang akan dijadikan patokan, maka hasil pembacaan dari sensor dapat digunakan untuk membuat robot berjalan mengikuti garis pandu.

### **2.3 Lego Mindstorms EV3**

LEGO Mindstorms EV3 adalah generasi ketiga dari LEGO. Ini merupakan penerus dari LEGO Mindstorms NXT seri 2.0 generasi kedua. Kata "EV3" menunjukkan bahwa EV3 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 LabVIEW untuk LEGO MINDSTORMS. Berbagai bahasa resmi ada, seperti NXC, NBC leJOS NXJ, dan Robot C (Nugraha, 2017).

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 2.2 dapat dirakit misalnya dengan sensor yang mengontrol motor dan bereaksi terhadap cahaya, sentuhan, suara, dan lain-lain.



**Gambar 2.2** Robot LEGO Mindstroms EV3

## **2.4 Komponen LEGO Mindstorms EV3**

### **2.4.1 EV3 Brick**

*Brick* adalah komponen penting dari robot EV3, karena berfungsi sebagai pengendali (otak dan sumber tenaga robot EV3). Untuk membuat *Brick* dapat mengendalikan robot diperlukan suatu program yang terlebih dahulu dibuat menggunakan *software* pemrograman untuk Lego Mindstorms. Program yang sudah dibuat dapat di-*upload* ke EV3 *Brick* untuk di *compile* dan kemudian digunakan pada robot yang telah dirakit sebelumnya (Ika Sunarsih, 2017).

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* (PWM) 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 dilihat pada Gambar 2.3



**Gambar 2.3** EV3 *Brick*

Status cahaya pada *brick* yang mengelilingi *Buttons Brick* berfungsi sebagai indikator status keadaan EV3 *Brick* saat ini. Cahaya pada *Buttons Brick* seperti pada Gambar 2.4 dapat menjadi hijau, oranye, atau merah. Kode status cahaya *Buttons Brick* adalah sebagai berikut:

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



**Gambar 2.4** Status Cahaya EV3 *Brick*

Untuk bagian-bagian yang terdapat pada sisi EV3 *Brick* dapat dilihat pada gambar-gambar dibawah ini :



**Gambar 2.5** 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 2.5.



**Gambar 2.6** 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 2.6.



**Gambar 2.7** EV3 Brick Bagian Kanan

Di sisi 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 2.7.



**Gambar 2.8** EV3 Brick Bagian Kiri

*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 2.8.

*Brick* dapat diibaratkan seperti CPU pada komputer, yang berfungsi untuk mengolah data. *Brick* berfungsi untuk mengendalikan jalannya robot sesuai dengan program yang telah dibuat. EV3 dapat diprogram melalui 2 cara, yaitu :

1. Membuat program secara langsung pada EV3 *Brick*.
2. Membuat program melalui komputer, lalu di-*upload* ke EV3 *Brick*.

Untuk program-program yang sederhana dapat dibuat secara langsung pada EV3 *Brick*, sedangkan untuk program-program yang kompleks dan rumit program dapat dibuat menggunakan komputer terlebih dahulu.

#### **2.4.2 Motor**

Menurut Nugraha (2017), 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.

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 2.9 dan Gambar 2.10 menunjukkan Motor *Large* dan Motor *Medium* LEGO Mindstroms EV3.



**Gambar 2.9** Motor *Large*



**Gambar 2.10** Motor *Medium*

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.

Motor *medium* juga termasuk *built-in* rotasi sensor dengan resolusi satu derajat, tetapi lebih kecil dan lebih ringan. Hal itu berarti motor *medium* 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 didapat suatu perbandingan 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).

#### **2.4.3 Sensor Warna (*Colour Sensor*)**

Salah satu fungsi sensor warna adalah memberikan robot penglihatan (layaknya sensor ultrasonik). Sensor warna adalah sensor digital yang dapat mendeteksi warna atau intensitas cahaya (David dan Ivan, 2012). Selain itu, sensor warna dapat digunakan dalam tiga mode yang berbeda, yaitu :

1. *Colour Mode*, artinya sensor warna digunakan untuk mendeteksi dan membedakan warna. Warna yang dapat dideteksi sebanyak tujuh warna yaitu hitam, biru, hijau, kuning, merah, putih, coklat dan tidak berwarna. Kemampuan untuk membedakan antara warna berarti robot dapat diprogram untuk mengurutkan benda berwarna atau blok, berbicara nama-nama warna seperti yang terdeteksi, atau menghentikan tindakan ketika melihat warna tertentu. Gambar 2.11 menunjukkan sensor warna dalam *colour mode*.



**Gambar 2.11** Sensor Warna dalam *Colour Mode*

2. *Reflected Light Intensity Mode*, sensor warna akan memancarkan cahaya merah lalu pantulan cahaya yang diterima oleh sensor yang kemudian digunakan untuk mengukur intensitas cahaya. Sensor menggunakan skala 0 (sangat gelap) hingga 100 (sangat ringan). Ini berarti robot dapat diprogram untuk bergerak pada permukaan putih sampai garis hitam terdeteksi, atau untuk menafsirkan kartu identitas kode warna. Gambar 2.12 menunjukkan sensor warna dalam *reflected light intensity mode*.



**Gambar 2.12** Sensor Warna dalam *Reflected Light Intensity Mode*

3. *Ambient Light Intensity Mode*, artinya sensor warna digunakan untuk mengukur kekuatan cahaya yang masuk dari jendela lingkungannya, seperti sinar matahari atau sinar senter. Sensor menggunakan skala 0 (sangat gelap) hingga 100 (sangat ringan). Ini berarti robot dapat diprogram untuk menonaktifkan alarm ketika matahari terbit di pagi hari, atau menghentikan tindakan jika lampu mati. Gambar 2.13 menunjukkan sensor warna dalam *ambient light intensity mode*.



**Gambar 2.13** Sensor Warna dalam *Ambient Light Intensity Mode*

Tingkat sampel sensor warna adalah 1 kHz/detik, untuk akurasi terbaik, ketika menggunakan sensor warna dalam *colour mode* atau *reflected light intensity mode*, sensor harus diletakkan pada sudut yang dekat, tetapi tidak menyentuh permukaan.

#### **2.4.4 Sensor Sentuh (*Touch Sensor*)**

Menurut Wisnu Jatmiko (2010), sensor sentuh adalah sensor yang mampu mendeteksi adanya sentuhan. Kemampuan robot untuk mengetahui bahwa robot mengenai benda atau bertabrakan dengan robot lain didapatkan dengan bantuan sensor sentuh. Sensor sentuh memungkinkan tidak hanya untuk mendeteksi ada atau tidaknya sentuhan, tetapi juga apakah sentuhan sudah dilepaskan atau belum. Berikut adalah gambar sensor sentuh seperti pada Gambar 2.14.



**Gambar 2.14** Sensor Sentuh (*Touch Sensor*)

#### 2.4.5 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, 2017).



**Gambar 2.15** Konektor

**Tabel 2.2** EV3 Sensor *Interface pin-out*

<i>Pin</i>	<i>Name</i>	<i>Function</i>	<i>Color</i>
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

#### 2.4.6 Komponen Tambahan

Selain komponen utama, ada juga komponen tambahan untuk membuat robot LEGO Mindstorms EV3. Fungsi komponen tambahan ini adalah untuk merakit *body* robot dan menyatukan berbagai sensor dengan *brick* sehingga membentuk suatu robot yang utuh dan siap untuk diprogram (Ika Sunarsih, 2017). Komponen-komponen yang terdapat di robot LEGO Mindstorms EV3 secara lengkap, dapat dilihat pada Gambar 2.16.



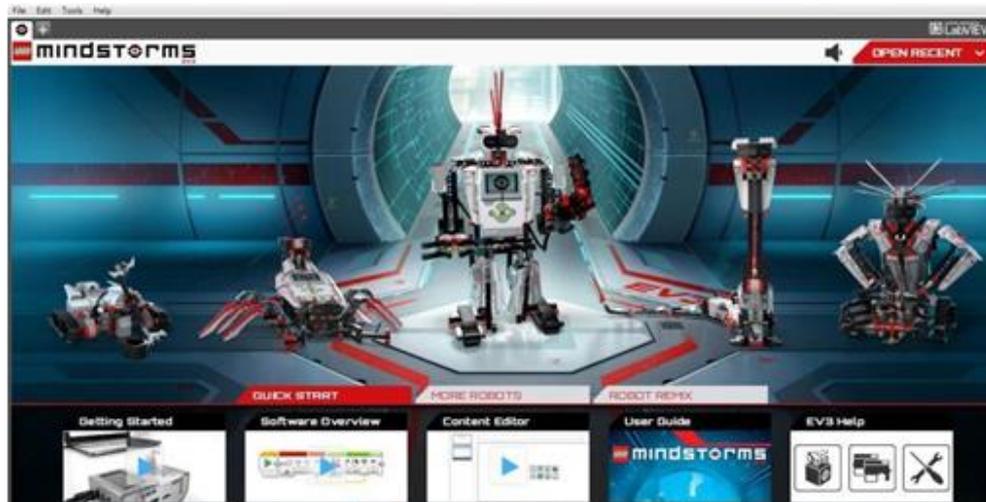
**Gambar 2.16** Komponen Tambahan EV3

#### 2.5 Program LEGO Mindstorms EV3

Menurut Ika Sunarsih (2017) Untuk menjalankan robot EV3, robot harus diprogram dengan algoritma yang diinginkan terlebih dahulu. Ada banyak bahasa pemrograman yang dapat digunakan untuk memprogram EV3, salah satunya adalah LEGO Mindstorms EV3 *Home Edition*. Dengan menggunakan LEGO Mindstorms EV3 *Home Edition*, proses pembuatan program untuk robot menjadi lebih cepat.

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, untuk memprogram robot LEGO Mindstorms EV3 juga dapat dilakukan melalui ponsel / tablet. Dalam program LEGO Mindstorms EV3 *Home Edition*, layar ditampilkan di waktu *startup* disebut *Lobby* seperti pada Gambar 2.17. Isi *Lobby* adalah menu untuk mengakses setiap fungsi dari program LEGO Mindstorms EV3 *Home Edition*.



**Gambar 2.17** LEGO Mindstorms EV3 *Home Edition* Lobby

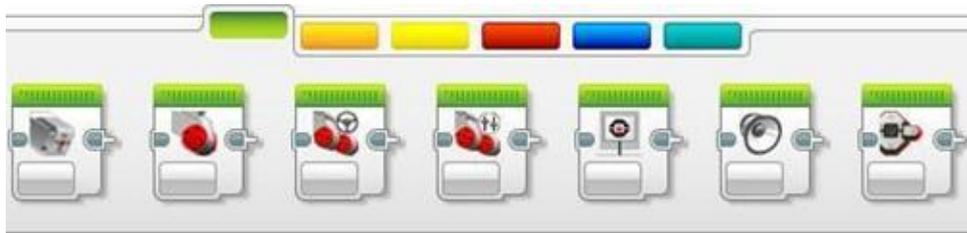
Lembar *Project* adalah halaman yang digunakan untuk membuat program dengan menggunakan blok pemrograman, seperti pada Gambar 2.18.



**Gambar 2.18** Lembar *Project*

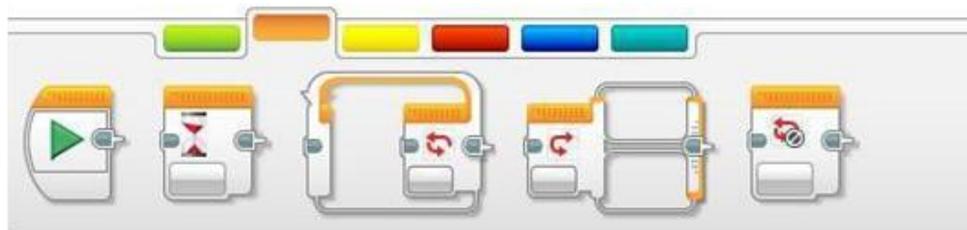
### 2.5.1 Programming Blocks and Palettes

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 (Nugraha, 2017). Di dalam bagian *Programming Palettes* terdapat blok program sebagai berikut:



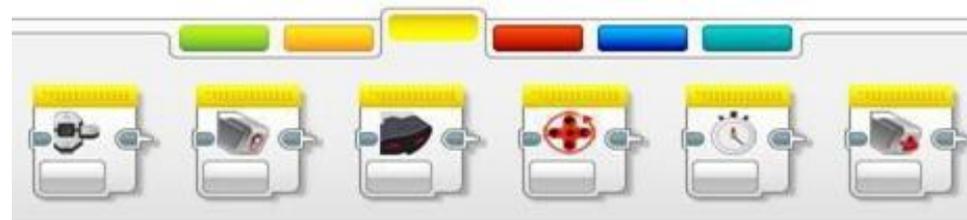
**Gambar 2.19** Action Blocks

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



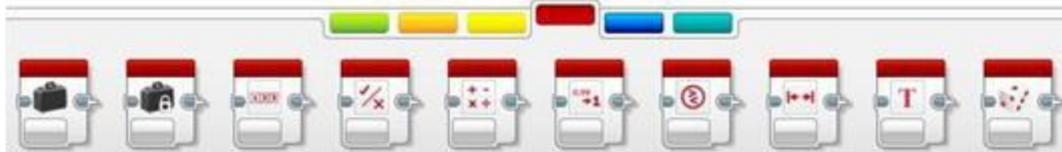
**Gambar 2.20** Flow Control

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



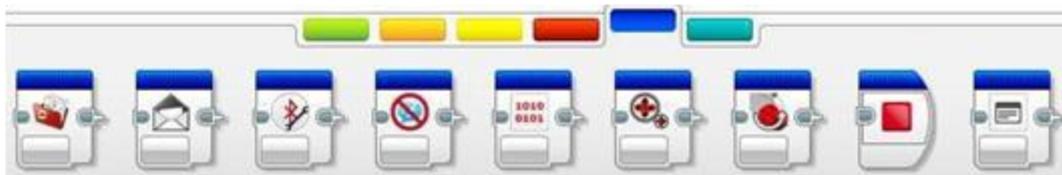
**Gambar 2.21** Block Sensor

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*.



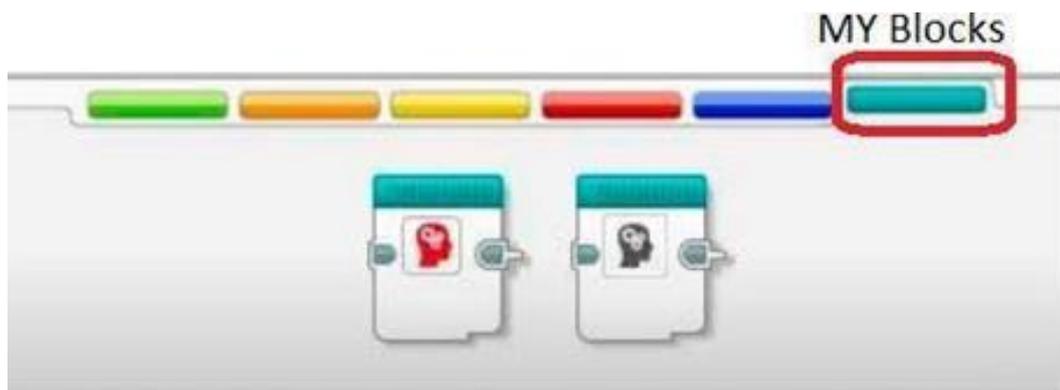
**Gambar 2.22** *Data Operations*

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



**Gambar 2.23** *Advance*

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



**Gambar 2.24** *My Blocks*

*My Blocks* berfungsi ketika bagian yang sama dari sebuah program digunakan secara berulang kali di banyak program, itu merupakan saat yang tepat untuk membuat *My Blocks*. Hal ini akan memudahkan dalam pembuatan program karena tidak perlu membuat program yang sama secara berulang dan menjadikan program yang dibuat lebih sederhana dan mudah dibaca.

### 2.5.2 Data Logging

Menurut Nugraha (2017) EV3-G tidak hanya merekam data, tetapi dapat membantu dalam mengatur dan menganalisisnya. Fungsi-fungsi tersebut dapat mempermudah proses pembuatan program khususnya yang berhubungan dengan pembacaan data dari sensor. *Data Logging* terdiri dari bidang utama sebagai berikut:

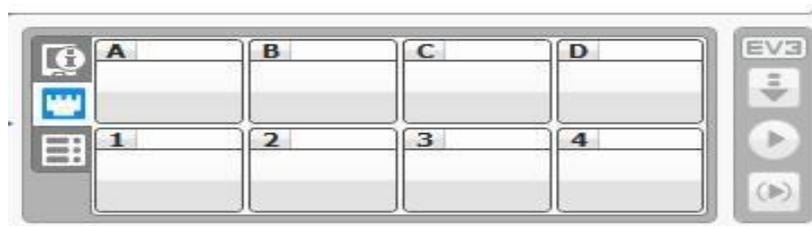
1. *Graph Area* — Untuk melihat dan menganalisa plot data.
2. *Configuration Panel* — Tempat untuk menyiapkan eksperimen, mengelola *dataset*, dan mendapatkan akses ke *Dataset* Perhitungan dan *Grafik Programming*, fungsi *data logging* khusus.
3. *Hardware Page* — Untuk membangun dan mengelola komunikasi 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* – Berfungsi untuk 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 2.25(a)** *Graph Area*



**Gambar 2.25(b)** *Configuration Panel*



**Gambar 2.25(c)** *Hardware Page*

Pada Gambar 2.25(a) digunakan untuk mempersiapkan sebuah eksperimen dengan memilih durasi sampel, menilai dan memilih sensor apa yang digunakan untuk mengumpulkan data. Untuk setiap sensor, harus dipilih mode sensornya, karena sebagian besar sensor dapat memberikan berbagai jenis data.

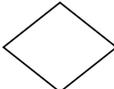
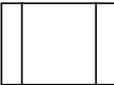
Pada Gambar 2.25(b) dapat memanipulasi *dataset* dan grafik dengan melakukan berbagai perhitungan dengan menggunakan angka, fungsi, dan *dataset* lainnya. Hasil *output* akan ditampilkan berupa sebuah grafik dan nilai-nilai *dataset*.

Pada Gambar 2.25(c) membagi bidang grafik dalam zona yang berbeda, *input* sensor dapat diaktifkan untuk memicu atau mengaktifkan *output*: misalnya, motor atau suara dari EV3 *Brick* ketika nilai-nilai sensor mencapai tingkat tertentu.

## 2.6 *Flowchart*

*Flowchart* adalah bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah (Ladjamudin, 2013). Langkah-langkah tersebut dilambangkan dengan simbol-simbol tertentu yang bertujuan untuk membuat algoritma program menjadi lebih sederhana, mudah dibaca dan jelas tahapan-tahapan penyelesaian masalahnya. Simbol-simbol yang dimaksud dapat dilihat pada tabel 2.3.

Tabel 2.3 Simbol *Flowchart*

No	Simbol/Gambar	Keterangan
1		Simbol <i>Terminator</i> , untuk permulaan atau akhir sebuah <i>flowchart</i>
2		Simbol <i>Flow Direction</i> , untuk menghubungkan antara simbol yang satu dengan yang lain
3		Simbol <i>Processing</i> , untuk menunjukkan proses yang dilakukan
4		Simbol <i>Input - Output</i> untuk menyatakan <i>input</i> dan <i>output</i>
5		Simbol <i>Decision</i> pemilihan proses berdasarkan kondisi yang ada
6		Simbol <i>Offline Connector</i> untuk menyatakan sambungan dari proses ke proses lainnya pada halaman yang berbeda
7		Simbol <i>Connector</i> untuk menyatakan sambungan dari proses ke proses lainnya pada halaman yang sama
8		Simbol <i>Predefine Process</i> untuk melaksanakan bagian (sub-program) / prosedur.
9		Simbol untuk proses inialisasi / pemberian nilai awal