

BAB II

KAJIAN PUSTAKA

2.1 Teori Dasar

2.1.1 Mikrokontroler Arduino

Mikrokontroler (pengendali mikro) pada suatu rangkaian elektronik berfungsi sebagai pengendali yang mengatur jalannya proses kerja dari rangkaian elektronik. Di dalam sebuah IC mikrokontroler terdapat CPU, memori, *timer*, saluran komunikasi serial dan paralel, *port input/output*, ADC, dll (Andrianto & Darmawan, 2016).

Pemrograman mikrokontroler merupakan dasar dari prinsip pengontrolan kerja robot. Orientasi penerapan mikrokontroler adalah mengendalikan suatu sistem berdasarkan informasi input yang diterima, lalu diproses oleh mikrokontroler, dan dilakukan aksi pada bagian output sesuai program yang telah ditentukan sebelumnya (Budiharto, 2013).

Mikrokontroler merupakan pengontrol utama perangkat elektronika saat ini termasuk robot tertentu. Mikrokontroler yang terkenal dan mudah didapatkan di Indonesia adalah 89S51, AVR, ATmega 8535, ATmega16, ATmega328 dan ATmega128. Dengan kata lain, mikrokontroler adalah suatu alat elektronika digital yang mempunyai masukan dan keluaran serta kendali dengan program yang bisa ditulis dan dihapus dengan cara khusus.

Arduino dikatakan sebagai sebuah *platform* dari *physical computing* yang bersifat *open source*. Pertama-tama perlu dipahami bahwa kata “*platform*” di sini adalah sebuah pilihan kata yang tepat. Arduino tidak hanya sekedar sebuah alat pengembangan, tetapi ia adalah kombinasi dari *hardware*, bahasa pemrograman dan *intergrated development environment* (IDE) (Sanjaya, 2014).

Arduino adalah suatu perangkat *prototype* elektronik berbasis mikrokontroler yang fleksibel dan *open-source*, perangkat keras dan perangkat lunaknya mudah digunakan. Perangkat ini ditujukan bagi siapapun yang tertarik/memanfaatkan mikrokontroler secara praktis dan mudah (Andrianto & Darmawan, 2016).

Arduino UNO adalah salah satu produk berlabel Arduino yang sebenarnya adalah suatu papan elektronik yang mengandung mikrokontroler ATmega328 (sebuah keping yang secara fungsional bertindak seperti sebuah komputer) (Kadir, 2012).

Arduino sangat cepat berkembangnya, hal ini dapat terjadi karena modul Arduino bersifat *open source* dan *open hardware*, sehingga siapapun dapat berkontribusi untuk menggunakan dan mengembangkannya. Bahkan tim Arduino sendiri telah membebaskan bagi siapapun untuk menduplikasi menjual atau memodifikasi *board* Arduino tanpa harus izin dari pihak atau tim resmi Arduino (Rangkuti, 2016).

Arduino dikembangkan dari thesis Hernando Barragan pada tahun 2004, seorang mahasiswa asal Kolombia. Judul thesisnya yaitu “Arduino-Revolusi Open Hardware”. Arduino diawali di ruang kelas *Interactive Design Institute* di Ivrea

(IDII), pada tahun 2005 di Ivrea, Italia. Arduino ditemukan oleh Massimo Banzi dan David Cuartielles dengan tujuan awal yaitu untuk membantu para siswa membuat perangkat desain dan interaksi dengan harga yang murah dibandingkan dengan perangkat lain yang tersedia saat itu, seperti BASIC Stamp yang harganya cukup mahal bagi pelajar pada saat itu. Arduino berasal dari bahasa Italia yang berarti teman yang berani. Pada bulan Mei 2011, Arduino sudah terjual lebih dari 300.000 unit. Arduino saat ini sudah menjadi *platform* OSHW (*Open Source Hardware*) (Andrianto & Darmawan, 2016).

Di pasaran banyak terdapat model board Arduino, karena bersifat *open source*, maka banyak vendor yang membuat dan menjual variannya baik yang *official* maupun UNO*official*. Contoh board Arduino yang *official* adalah: Arduino UNO, Duemilanove, Leonardo, Nano, Mega 2560/Mega ADK, Mega (ATMega 1280), Esplora, Micro, Mini, NG/older, dll.



Gambar 2. 1 Arduino UNO Board ATmega 328

Pada penelitian ini Peneliti menggunakan board Arduino UNO ATmega328 karena harganya yang murah serta memiliki spesifikasi yang memadai untuk digunakan pada robot *solving line maze*.

2.1.2 Motor DC

Motor DC adalah motor yang bergerak berputar 360 derajat biasanya disebut dinamo dan biasanya di gunakan sebagai penggerak roda. Apabila kutub positif dan negatif sumber yang dipasang ditukar maka motor DC akan berputar berlawanan arah dari arah putar sebelumnya (Andrianto & Darmawan, 2016).



Gambar 2. 2 Motor DC

Motor DC adalah komponen yang dapat mengubah energi listrik menjadi energi gerak. Motor DC pertama kali dibuat oleh ilmuwan fiisika bernama M. Faraday. Di dalam motor DC terdapat kumparan kawat dan magnet. Kebalikan dari motor DC adalah generator listrik yang mampu mengubah gerak menjadi listrik, yang juga ditemukan oleh M. Faraday. Menurut Faraday, ketika pada kumparan yang dialiri arus listrik dari baterai didekatkan medan magnet, maka pada kumparan akan muncul gaya dorong yang akan membuat kumpuran berputar. (Sanjaya, 2014)

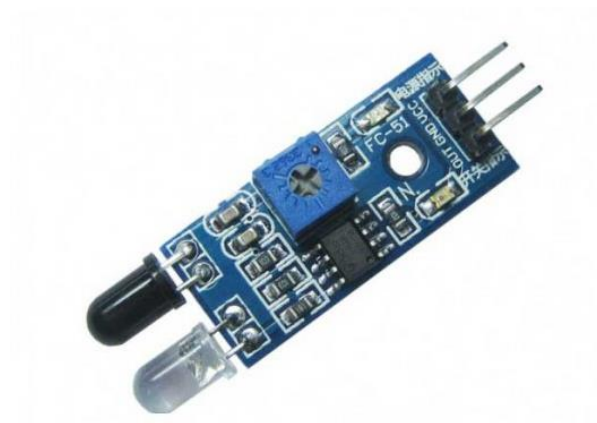
Kecepatan putaran motor DC ditentukan oleh besar tegangan. Semakin tinggi tegangannya, semakin cepat putarannya. Namun, tentu saja tegangan yang dapat di berikan ke motor DC ada batasannya. Tegangan yang terlampau tinggi, yang melampaui batas maksimumnya, dapat membuat motor rusak/terbakar. Sebaliknya

tegangan yang rendah, yang kurang dari batas minimumnya, maka motor tersebut tidak dapat digerakkan.

Sumber tegangan yang digunakan untuk motor DC perlu di ambil dari sumber eksternal, misalnya berupa baterai atau tegulator, bukan dari Arduino UNO. Hal ini disebabkan Arduino UNO tidak dirancang untuk memasok tegangan ke motor DC. Pin-pin Arduino UNO hanya bisa memberikan arus sebesar 60mA, sedangkan kebutuhan motor sekitar 500mA untuk membuat putaran maksimal.

2.1.3 Sensor Garis

Sensor garis adalah suatu perangkat/alat yang digunakan untuk mendeteksi adanya sebuah garis atau tidak (Sanjaya, 2014). Garis yang dimaksud adalah garis berwarna hitam di atas permukaan berwarna putih atau kebalikaannya. Alat ini menggunakan Teknik pantulan cahaya yang ditangkap oleh sensor cahaya dari sebuah LED. Sensor cahaya yang biasanya di gunakan adalah *infrared*, LDR (*Light Dependent Resistance*), *photodiode*, atau *phototransistor*. Sensor cahaya digunakan sebagai pembagi tegangan yang nilainya dapat berubah-ubah. Saat tidak ada cahaya pantulan, dalam hal ini karena diserap oleh permukaan gelap, maka resistansi sensor akan sangat besar. Sebaliknya, saat cahaya pantulan dari permukaan putih, maka resistensi sensor berkurang secara drastic.



Gambar 2. 3 Sensor Garis IR

2.2 Teori Khusus

2.2.1 Robot

Robot adalah seperangkat alat mekanis yang dapat melakukan tugas fisik, menggunakan pengawasan dan kontrol manusia, atau bekerja secara otomatis menggunakan program yang telah didefinisikan terlebih dahulu, atau menggunakan kecerdasan buatan. Robot biasanya digunakan untuk tugas yang berat, berbahaya, pekerjaan yang berulang dan kotor. Biasanya kebanyakan robot industri digunakan dalam bidang produksi. Penggunaan robot lainnya termasuk untuk pembersihan limbah beracun, penjelajahan bawah air dan luar angkasa, pertambangan, pekerjaan "cari dan tolong" (*search and rescue*), dan untuk pencarian tambang. Belakangan ini robot mulai memasuki pasaran konsumen di bidang hiburan, dan alat pembantu rumah tangga (Tjindrawan, 2015).

Menurut (Budiharto, 2013) pada umumnya, sebuah robot memiliki karakteristik:

- a. **Sensing**, robot harus dapat mendeteksi lingkungan di sekitarnya (halangan, panas, suara, dan *images*).
- b. **Mampu Bergerak**, robot umumnya bergerak menggunakan kaki atau roda. Pada beberapa kasus, robot diharapkan dapat terbang atau berenang.
- c. **Cerdas**, robot memiliki kecerdasan buatan supaya dapat memutuskan aksi yang tepat dan akurat.
- d. **Membutuhkan energi yang memadai**, robot membutuhkan satu daya yang memadai supaya unit pengontrol dan actuator dapat menjalankan fungsinya dengan baik.

Robot secara umum terbagi menjadi dua kriteria, yaitu robot *autonomous* dan robot *control*. Robot *autonomous* adalah robot yang dapat bekerja tanpa harus dikontrol secara langsung oleh manusia. Contoh robot *autonomous* adalah robot *line tracer*, robot *light follower*, robot *obstacle avoider*, lampu lalu lintas, pintu otomatis, dan lainnya. Prinsip kerja robot *autonomous* secara sederhana adalah kemampuannya dalam memberikan respons terhadap perubahan kondisi lingkungan dengan berbagai sensor yang dimilikinya. Sedangkan robot *control* adalah robot yang harus dikendalikan secara langsung oleh manusia. Contoh robot jenis ini adalah komputer PC/Laptop, televisi, handphone, robot *remote control*, dan lainnya (Sanjaya, 2014).

Berdasarkan (Tjindrawan, 2015), Sejarah konsep dan penciptan robot dapat diuraikan sebagai berikut.

1. Pada tahun 350 SM, seorang filsuf bernama Homeros menulis dalam karya eposnya berjudul Iliad, sesuatu yang berkerja seperti robot. Dan pada tahun

yang sama juga Aristoteles membayangkan alat mekanis yang bekerja dengan menaati kehendak orang lain, bahkan sebelum orang menyatakan kehendaknya.

2. Pada tahun 60M, Hero of Alexandria menciptakan berbagai macam mesin. Salah satunya bisa disebut sebagai robot dimasa itu, yaitu sebuah alat yang secara otomatis bisa diprogram dalam menuangkan minuman, khususnya anggur.
3. Robot dalam industri diawali dengan kartu berlubang yang bisa secara otomatis mengatur pola kain pada mesin tenun.
4. Robot yang dikendalikan jarak jauh dibuat oleh Nikola Tesla pada tahun 1890. Pengendalian dilakukan menggunakan sinyal radio.
5. Pada tahun 1892, Sewary Babbit membuat suatu alat yang dapat menggantikan tenaga manusia pada industri pada tempat yang berbahaya, yaitu alat pembersih ingot baja pada tungku yang panas.
6. Pada tahun 1920-an sampai tahun 1930-an, Robot sering dijadikan sebagai bahan dalam pembuatan film *science fiction* bahkan hingga diproduksi oleh Hollywood.
7. Pada tahun 1933, dalam pameran *International World Fair* di Chicago, Robot mulai menjadi pusat perhatian pengunjung. Pada tahun-tahun berikutnya robot dalam pameran semakin populer.
8. Pada tahun 1938, diciptakan lengan mekanis yang bisa menyembrotkan cat lukisan secara otomatis.

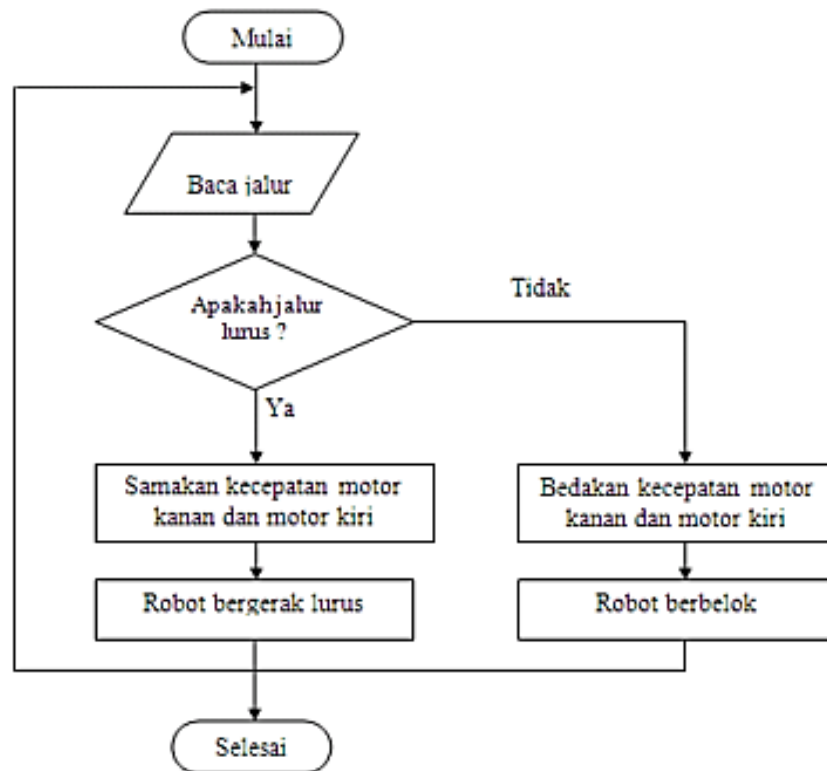
9. Pada tahun 1942, Issac Asimov, penulis kisah-kisah fiksi tentang robot, menetapkan tiga hukum robot yang harus dipatuhi dalam penciptaan robot. Tulisan ini diterbitkan pada tahun 1950. Bunyi ketiga hukum tersebut sebagai berikut:
 - a) Robot tidak boleh melukai manusia, atau dengan berdiri diam, membiarkan manusia menjadi celaka.
 - b) Robot harus mematuhi perintah yang diberikan oleh manusia, kecuali bila perintah tersebut bertentangan dengan hukum pertama
 - c) Robot harus melindungi keberdaan dirinya sendiri selama perlindungan tersebut tidak bertentangan dengan hukum pertama atau hukum kedua.
10. Zaman Robot bisa berpikir dimulai dengan diciptakannya komputer elektrik pertama di dunia, yaitu komputer Eniac di USA pada tahun 1946.
11. Sehubungan dengan anggapan bahwa komputer bisa berpikir, pada tahun 1947 Alan M. Turing menulis artikel tentang kecerdasan buatan alias *artificial intelligence*.
12. Pada tahun 1951 dibuatlah lengan otomatis di tempat yang lebih berbahaya lagi, yaitu ditempat bahan radioaktif.
13. Pada tahun 1976, NASA mengoperasikan lengan robot untuk melakukan berbagai macam penelitian pada pesawat pendaratan pertama di planet Mars.
14. Pada tahun 1997 diterjunkan robot *pathfinder* kemudian *sourjurner* untuk meneliti permukaan Mars.

15. Pada tahun 1980 robot yang diciptakan Marc Raibert di MIT bukan hanya bisa berjalan dengan dua kaki, melainkan juga melompat. Selain itu robot juga semakin pandai membaca, bukan hanya bergerak.
16. Pada tahun 1984 juga, robot bisa melakukan pelayanan dirumah sakit. Dimasa selanjutnya robot bisa melakukan operasi dengan kecermatan melebihi manusia. Pada tahun 1990 robot sudah bisa melakukan operasi pinggul pada anjing.
17. Pada tahun 1997 dimulai sepakbola antarrobot pertama di jepang, namanya *RoboCup*. Selanjutnya sepakbola robot diadakan di kota-kota lain diseluruh dunia sebagaimana *World Cup*. Kompetisi terbaru pada tahun 2014 diadakan di Brazil dengan peserta 45 negara.
18. Pendekatan terbaru dalam robotika adalah kerja sama sejumlah robot kecil secara wireless. Pada tahun 2014, Harvard *University* menguji kerja sama sebanyak 1024 buah robot yang bisa kompak membentuk berbagai macam konfigurasi tanpa dihubungkan kabel.

2.2.2 Robot *Line Follower*

Robot *line follower* merupakan robot yang bisa bergerak mengikuti garis panduan yaitu garis putih yang diletakkan di atas permukaan berwarna hitam, atau sebaliknya garis hitam yang diletakkan di atas permukaan berwarna putih (Yanto & Welly, 2015). Cara kerja robot *line follower* yaitu dengan menggunakan sensor garis seperti *infrared* yang mendeteksi adanya garis pada permukaan lintasan robot, dan informasi yang telah diterima oleh sensor garis akan diteruskan ke penggerak atau

motor tersebut agar motor dapat menyesuaikan dengan gerak tubuh robot sesuai dengan garis yang dideteksi.



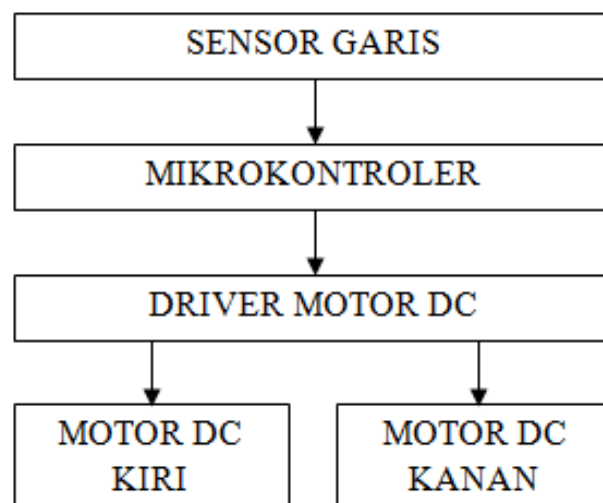
Gambar 2. 4 Flowchart Logika Line Follower

Line follower robot (robot pengikut garis) adalah robot yang dapat berjalan mengikuti sebuah lintasan, ada yang menyebutnya dengan *line tracker*, *line tracer robot*, dan sebagainya. Garis yang dimaksud adalah garis berwarna hitam di atas permukaan putih atau sebaliknya. Ada juga lintasan dengan warna lain dengan permukaan yang kontras dengan warna garisnya. Ada juga garis yang tak terlihat yang digunakan sebagai lintasan robot, misalnya medan magnet (Sanjaya, 2014).

Robot *line follower* adalah sebuah robot sederhana yang mampu bergerak mengikuti lintasan berupa garis. Untuk melihat keberadaan garis, pada robot dipasang beberapa sensor garis, yang terdiri dari pasangan LED dan sensor cahaya

(*photodiode*, *phototransistor*, atau LDR). Pantulan cahaya yang dipancarkan oleh LED akan diterima oleh sensor cahaya, yang kemudian di-konversikan menjadi besaran listrik. Output sensor yang sudah berupa besaran listrik (tegangan) dibaca oleh mikrokontroler, yang kemudian dijadikan rujukan untuk pengambilan keputusan berupa langkah untuk mengatur pergerakan dua motor DC, baik arah maupun kecepatan (Tjindrawan, 2015).

Berikut adalah blok diagram aplikasi robot *line follower* menggunakan mikrokontroler.

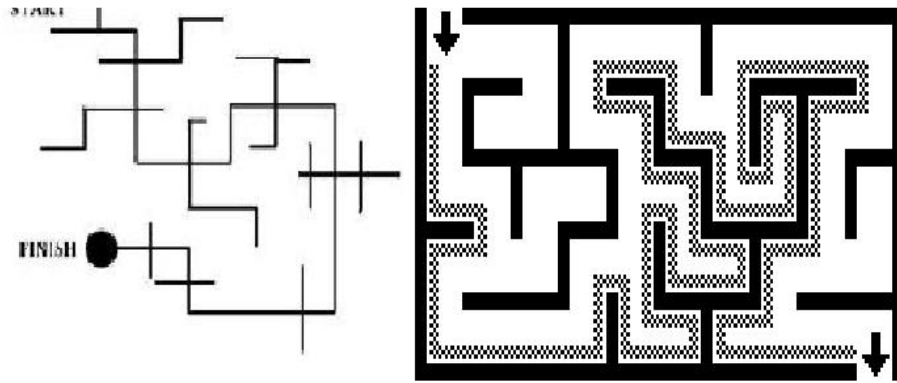


Gambar 2. 5 Blok Diagram Aplikasi Robot *Line Follower*

2.2.3 Robot *Solving Maze*

Robot *solving maze* memiliki kemampuan bergerak menelusuri garis (*line following*) dan harus mampu menyelesaikan kombinasi garis berupa *maze*, yang bentuknya beragam (Yultrisna & Syofian, 2013). Mulai dari persimpangan dengan sudut bervariasi, tikungan, kurva, dan yang lainnya. Pada bidang robotika ada dua jenis *maze* yang umum digunakan, yaitu *wall maze* dan *line maze*. *line maze* bentuk

jalur yang dibuat berupa garis, sedangkan *wall maze* merupakan jaringan jalan yang terbentuk atas lorong-lorong dengan dinding tanpa atap.



Gambar 2. 6 Rancangan Jalur *Line Maze* dan *Wall Maze*

Robot *maze solving* adalah *mobile robot* yang dapat menelusuri jalur yang rumit, Jalur yang dilewati dapat berupa garis (*line*) atau dinding (*wall*) (Franky Candra Saputro, 2015). Robot *maze solving* yang pertama dibuat oleh seorang yang bernama Claude Elwood Shannon pada tahun 1950. Pada saat itu robot *maze solving* dikendalikan oleh suatu rangkaian *relay*, robot tersebut menyerupai tikus yang berukuran sama dengan tikus yang sesungguhnya. Robot tersebut mengembara pada lorong (6 dinding) rumit dengan 25 penyiku. Lorong tersebut bisa di ubah sesuka hati dan robot tersebut kemudian memeriksa secara menyeluruh jalan yang harus ditempuh untuk menemukan titik tujuan. Setelah melalui lorong tersebut, robot bisa ditempatkan di manapun dan akan bergerak secara langsung sampai tujuan. Hal ini nampak sebagai alat yang bisa belajar untuk pertama kalinya.

Permasalahan yang timbul pada *maze* adalah cara untuk menemukan titik target, sehingga dibutuhkan metode untuk menyelesaikannya. Terdapat 2 teknik pencarian dan pelacakan yang digunakan yaitu pencarian buta (*blind search*) dan pencarian terbimbing (*heuristic search*). Pada pencarian buta (*blind search*) terbagi

atas 2 teknik pencarian yaitu pencarian melebar pertama (*breadth-first search*) dan pencarian mendalam pertama (*depth-first search*). Dari beberapa teknik pencarian, teknik pencarian mendalam pertama (*depth-first search*) yang paling sering diterapkan untuk penyelesaian *maze* baik *line maze* maupun *wall maze* (Sutojo, 2011).

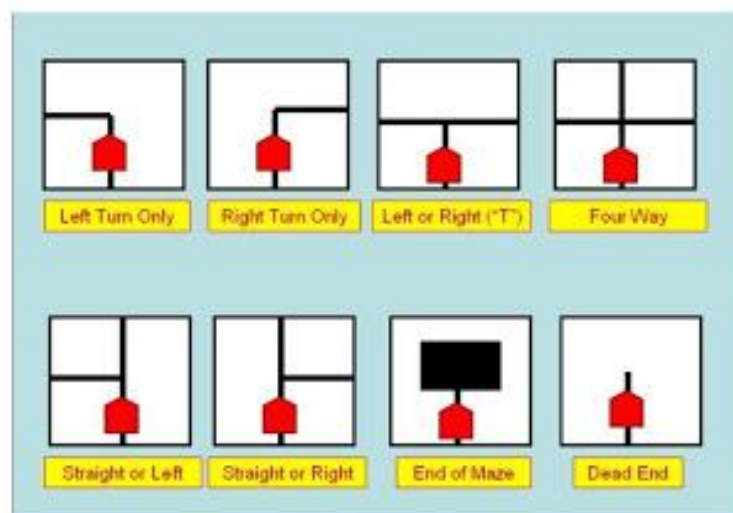
Pada *depth-first search*, proses pencarian akan dilakukan pada semua anaknya sebelum dilakukan pencarian ke node-node yang selevel. Pencarian dimulai dari node akar ke level yang lebih tinggi. Keuntungan *depth-first search* yaitu secara kebetulan, teknik pencarian tersebut dapat menemukan solusi tanpa harus menguji lebih banyak lagi dalam semua ruang keadaan. Sedangkan kelemahannya yaitu memungkinkan tidak ditemukannya tujuan yang diharapkan juga hanya akan mendapatkan satu solusi pada setiap pencarian (Sutojo, 2011). Teknik pencarian *depth-first search* juga lebih kenal dengan istilah algoritma *maze mapping* dalam dunia robotika.

Algoritma *maze mapping* adalah salah satu algoritma yang digunakan pada robot untuk mencari jalur terpendek dari sebuah *line maze*. Terdapat 2 aturan dalam algoritma ini, yaitu *left hand rule* dan *right hand rule*. Dalam *left hand rule*, robot akan lebih memilih untuk belok kiri dari pada lurus atau belok kanan dan jika tidak ada belokan ke kiri akan lebih memilih lurus dari pada belok kanan. Sebaliknya dalam *right hand rule*, robot akan lebih memilih belok kanan dari pada lurus dan lebih memilih lurus dari pada belok kiri. Robot yang telah diprogram dengan algoritma ini diletakkan di titik start sebuah *maze*, kemudian saat menjumpai persimpangan, robot akan memilih untuk berbelok sesuai dengan prioritas dari *rule*

yang digunakan hingga robot menemui akhir *maze* (*end of maze*) (Yanto & Welly, 2015).

Terdapat 8 kemungkinan dalam *maze* yang disebut dengan "*the eight maze possibilities*". Kedelapan kemungkinan tersebut adalah sebagai berikut:

1. tidak ada pilihan lain selain belok kiri (*left turn only*),
2. tidak ada pilihan lain selain belok kanan (*right turn only*),
3. kiri atau kanan (*left or right*),
4. kiri, lurus, atau kanan (*four way*),
5. lurus atau kiri (*straight or Left*),
6. lurus atau kanan (*straight or Right*),
7. jalan buntu (*dead end*),
8. *end of maze*.



Gambar 2. 7 *The Eight Maze Possibilities*

left turn only dan *right turn only* bukanlah merupakan sebuah persimpangan, sebab hanya ada satu belokan. Sehingga robot tidak menyimpan apapun ke dalam memori. *Dead end* juga bukan sebuah persimpangan, namun karena *dead end*

menandakan bahwa jalur yang dipilih adalah salah maka robot melakukan *U-turn* atau putar balik dan menyimpannya ke dalam memori.

2.3 *Tools/Software/Aplikasi/System*

2.3.1 IDE Arduino

IDE itu merupakan kependekan dari *integrated development enviroenment*, atau secara bahasa mudahnya merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Disebut sebagai lingkungan karena melalui *software* inilah Arduino dilakukan pemrograman untuk melakukan fungsi-fungsi yang dibenamkan melalui sintaks pemrograman. Arduino menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Bahasa pemrograman Arduino (*Sketch*) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Sebelum dijual ke pasaran, IC mikrokontroler Arduino telah ditanamkan suatu program bernama *bootloader* yang berfungsi sebagai penengah antara *compiler* Arduino dengan mikrokontroler.

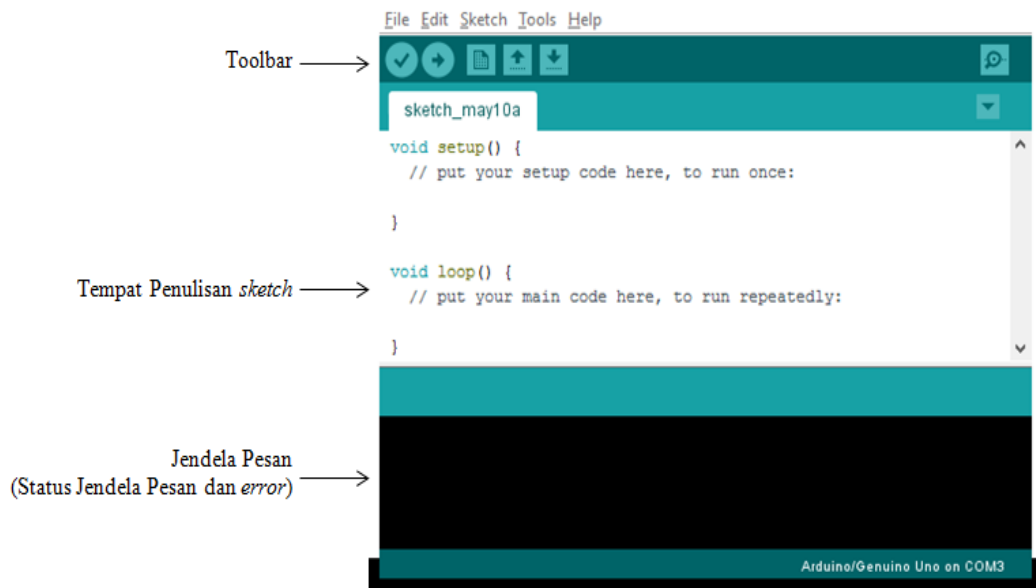
Arduino IDE dibuat dari bahasa pemrograman JAVA. Arduino IDE juga dilengkapi dengan *library* C/C++ yang biasa disebut *wiring* yang membuat operasi *input* dan *output* menjadi lebih mudah. Arduino IDE ini dikembangkan dari *software processing* yang dirombak menjadi Arduino IDE khusus untuk pemrograman dengan Arduino.

Program yang ditulis dengan menggunakan Arduino *software* (IDE) disebut sebagai *sketch*. *Sketch* ditulis dalam suatu *editor teks* dan disimpan dalam *file*

dengan ekstensi .ino. *Teks editor* pada *Arduino software* memiliki fitur-fitur seperti *cutting/paste* dan *searching/replacing* sehingga memudahkan kamu dalam menulis kode program.

Pada *software* *Arduino IDE*, terdapat semacam *message box* berwarna hitam yang berfungsi menampilkan status, seperti pesan *error*, *compile*, dan *upload* program. Di bagian bawah paling kanan *software* *Arduino IDE*, menunjukan *board* yang terkonfigurasi beserta *COM ports* yang digunakan.

Struktur dasar dalam pemrograman *Arduino* cukup sederhana dan terdiri dari dua bagian fungsi, yaitu fungsi persiapan (*setup*) dan fungsi utama (*loop*). *Setup()* adalah persiapan sebelum eksekusi program. *Loop()* adalah tempat menulis program utama yang akan dieksekusi. Fungsi *setup()* digunakan untuk mendefinisikan variabel-variabel yang digunakan dalam program. Fungsi ini berjalan pertama kali ketika program dijalankan, selanjutnya terdapat *loop()* yang merupakan program inti/utama dari *Arduino* yang dijalankan secara terus menerus baik pembacaan *input* maupun pengaktifan *output*. Program ini adalah inti dari semua program *Arduino*.



Gambar 2. 8 Tampilan Program Arduino 1.8.5

Software IDE Arduino adalah *software* yang ditulis dengan menggunakan java. *Software* ini dapat di *download* secara gratis. Jendela utama IDE Arduino terdiri dari tiga bagian utama, yaitu:

1. Bagian atas, yakni *toolbar*, pada bagian atas juga terdapat menu *file*, *edit*, *sketch*, *tools*, dan *help*.
2. Bagian tengah, yaitu tempat penulisan *coding* atau *sketch*. Sebuah kode program Arduino umumnya disebut dengan istilah *sketch* yang memiliki arti yang sama dengan *coding*.
3. Bagian bawah berupa jendela pesan (*message window*) yang berisi status dan pesan *error*.

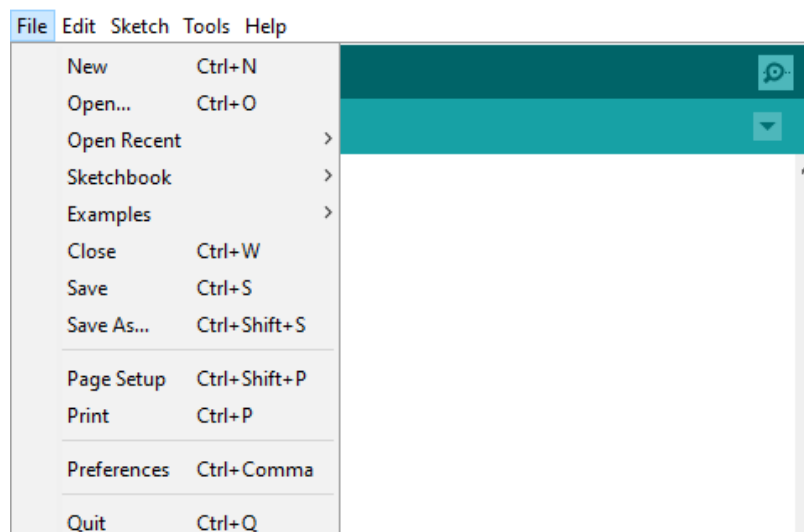


Gambar 2. 9 Tampilan *Toolbar* Arduino 1.8.5

Penjelasan bagian *toolbar*:

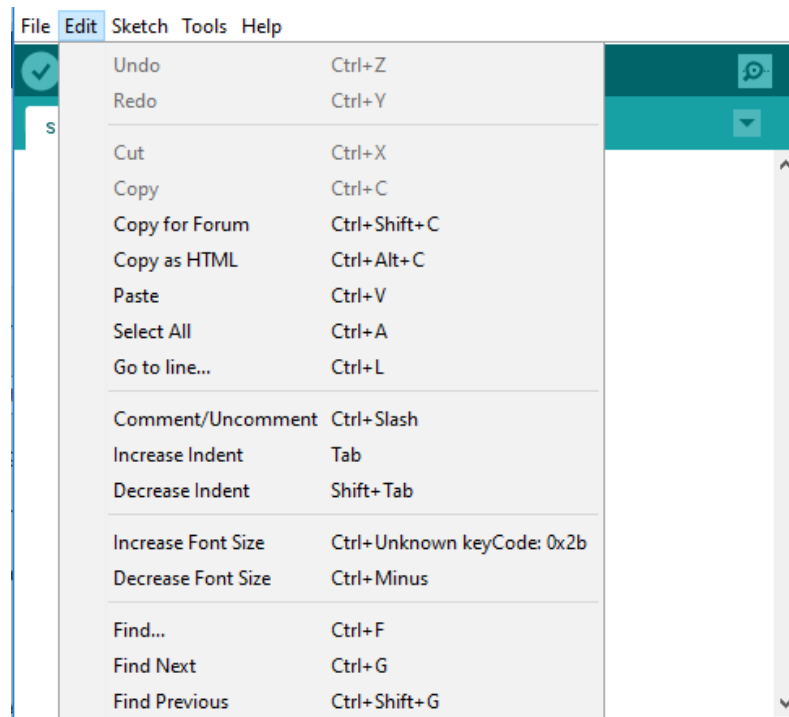
1. *Verify*, mengecek kode *sketch* yang *error* sebelum meng-*upload* ke board Arduino.
2. *Upload*, meng-*upload* *sketch* pada board Arduino.
3. *New*, membuat sebuah *sketch* baru.
4. *Open*, membuka daftar *sketch* pada *sketchbook*.
5. *Save*, menyimpan kode atau *sketch* pada *sketchbook*.
6. *Serial Monitor*, menampilkan data *serial* yang dikirimkan dari board Arduino.

Sedangkan bagian dari IDE menu diperlihatkan seperti berikut:



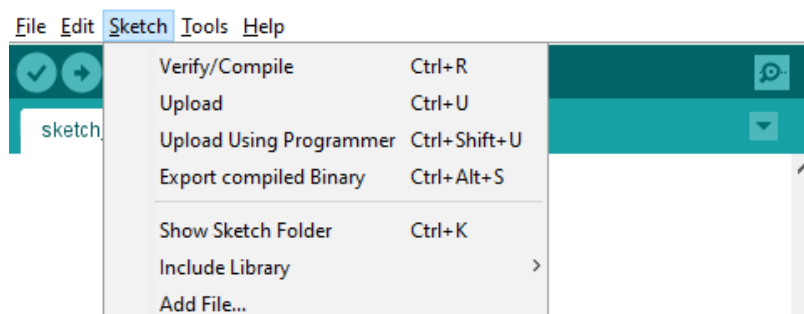
Gambar 2. 10 Tampilan Menu *File* Arduino 1.8.5

Bagian menu file terdiri dari *new*, *open*, *sketchbook*, *example*, *save*, *save as*, dan seterusnya.



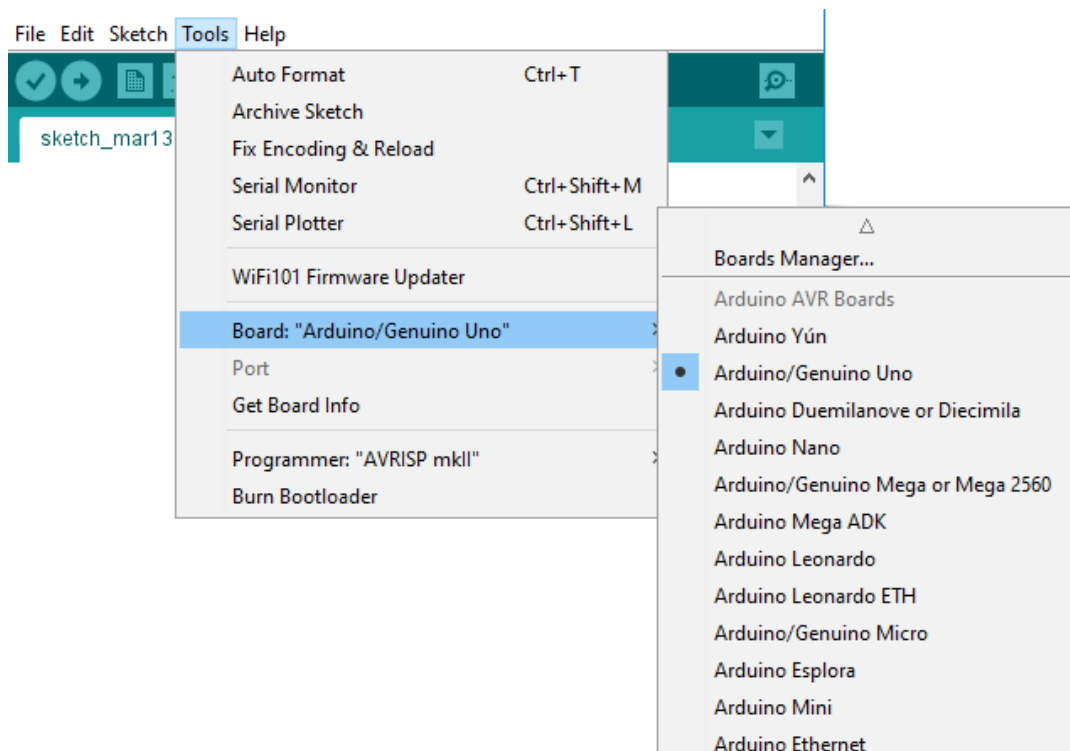
Gambar 2. 11 Tampilan Menu *Edit* Arduino 1.8.5

Bagian menu *edit* terdiri dari *cut*, *copy*, *copy for forum*, *copy as HTML*, *paste*, *select all*, dan seterusnya.



Gambar 2. 12 Tampilan Menu *Sketch* Arduino 1.8.5

Bagian menu *sketch* terdiri dari *verify/compile*, *upload*, *upload using programmer*, *show sketch file*, *add file*, dan seterusnya.

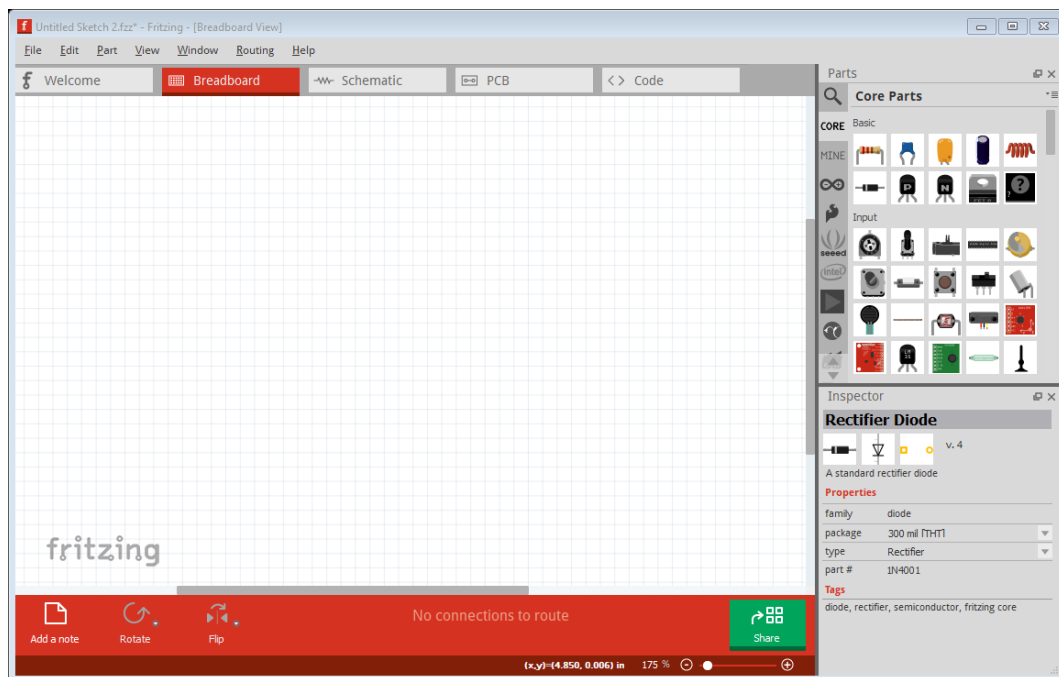


Gambar 2. 13 Tampilan Menu *Tools* Arduino 1.8.5

Pada bagian *tools* terdapat tipe board yang kita gunakan untuk meng-*upload* program, seperti board Arduino UNO, Arduino Nano, Arduino Mega, dan seterusnya.

2.3.2 Fritzing

Fritzing adalah salah satu dari perangkat lunak gratis yang dapat dipergunakan dengan baik untuk belajar elektronika. Perangkat lunak ini bisa bekerja baik di lingkungan sistem operasi GNU/Linux maupun Microsoft Windows. Masing-masing *software* memiliki keunggulannya masing-masing bagi setiap tipe pengguna dan keperluan.



Gambar 2. 14 Tampilan Awal Program *Fritzing*

Cara pemakaiannya pun sangat mudah, karena *software* ini menggunakan konsep *drag and drop*. Anda tinggal memilih komponen yang Anda inginkan pada bagian *parts*, *drag* komponen tersebut pada *main windows* lalu *drop*. Secara otomatis, Fritzing akan menggenerasikan 3 buah *layout*, yaitu gambar *breadboard*, Skematik, serta PCB. *Breadboard* merupakan (gambar) yang akan menampilkan gambar komponen asli (fisik). Skematik merupakan layout yang akan menampilkan gambar berupa rancangan skematik dari rangkaian yang Anda buat. Sedangkan PCB merupakan layout yang akan menampilkan gambar berupa rancangan pada PCB.

2.4 Penelitian Terdahulu

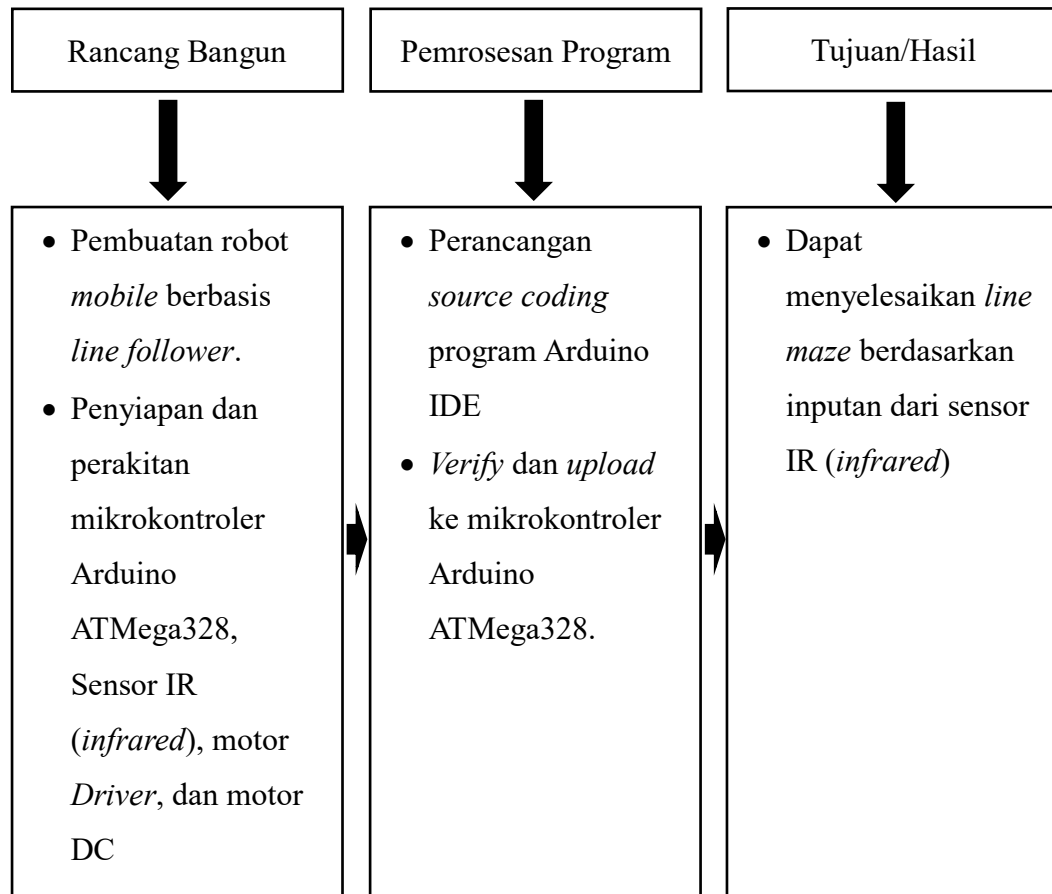
Pada sub-bab ini akan dijelaskan penelitian-penelitian terdahulu yang dilakukan oleh penelitian sebelumnya yang menjadi dasar acuan untuk melakukan penelitian ini.

1. Penelitian Yultrisna ST.,MT, Andi Syofian ST.,MT (2013) dengan Judul “Rancang Bangun Robot *Solving Maze* dengan Algoritma *Depth First Search*”. Penelitian ini menggunakan algoritma *depth first search* sebagai metode penyelesaian jalur *line maze* pada robot *line follower*.
2. Penelitian Achmad Zakki Falani, Setyawan Budi (2015) dengan judul “Robot *Line Follower* Berbasis Mikrokontroler ATmega 16 dengan menampilkan Status Gerak pada LCD”. Penelitian ini menggunakan LCD untuk menampilkan status gerak pada robot *line follower* juga menggunakan ATmega 16 sebagai mikrokontroler pengendalian robot.
3. Penelitian Febi Yanto, Irma Welly (2015) dengan judul “Analisa dan Perbaikan Algoritma *Line Maze Solving* untuk Jalur *Loop*, Lancip, dan Lengkung pada Robot *Line Follower (LFR)*”. Penelitian ini melakukan perbaikan dan tambahan dari program yang baru agar robot *line follower* dapat menyelesaikan *line maze* yang memiliki jalur *loop*, lancip, dan lengkung.
4. Penelitian Franky Candra Saputro, Haryanto (2015) dengan judul “Robot *Maze Solving* untuk Menyelesaikan Tiga Persimpangan”. Hasil penelitian akan menunjukkan bahwa robot *line follower* dapat menyelesaikan jalur dengan tiga persimpangan

5. Penelitian Pandian, n.d., R. Karthick, B. Karthikeyan (2015) dengan judul “*Maze Solving Robot Using Freeduino and LSRB Algorithm*”. Penelitian ini menggunakan algoritma LSRB (*left, straight, right, back/turning around*) sebagai penyelesaian jalur *line maze* pada robot *line follower*.
6. Penelitian Akib Islam, Farogh Ahmad, P.Sathya (2016) dengan judul “*Shortest Distance Maze Solving Robot*”. Penelitian ini menggunakan algoritma LSRB (*left straight right back*) dan algoritma RSLB (*right straight left back*) untuk menyelesaikan *line maze* pada robot *line follower*.
7. Penelitian David (2016) dengan judul “*Kendali Logika Fuzzy Pada Robot Line Follower*”. Penelitian tersebut menerapkan logika *fuzzy* dalam pengendalian robot *line follower*.
8. Penelitian Malkit Singh, Rajnish Kumar, Vaibhav Giradkar, Pallavi Bhole, Minu Kumari (2016), dengan judul “*Artificially Intelligent Maze Solver Robot*”. Penelitian ini menggunakan metode *right hand rule* yang merupakan salah satu penerapan kecerdasan buatan (*artificially intelligent*) untuk menyelesaikan *line maze* pada robot *line follower*.

2.5 Kerangka Pikir

Adapun kerangka pemikiran penelitian sebagai berikut:



Gambar 2. 15 Kerangka Pikir

Pada pembuatan robot *solving line maze* ini ada melalui beberapa Langkah. Langkah pertama adalah melakukan rancang bangun dan perakitan robot *mobile* berbasis *line follower* dengan mikrokontroler Arduino, sensor IR, motor *driver* dan motor DC. Langkah selanjutnya melakukan perancangan *source coding* dengan menggunakan program Arduino IDE dan lakukan *Verify* dan *upload source coding* ke mikrokontroler Arduino ATmega328. Langkah terakhir yaitu robot *line follower* dapat menyelesaikan *line maze* berdasarkan inputan dari sensor IR (*infrared*).