

**PERANCANGAN ROBOT COLOR
LIGHT FOLLOWER**

TUGAS AKHIR

Diajukan untuk Memenuhi Salah Satu Syarat Kelulusan
Program Pendidikan Sarjana

Oleh:
Bagus Budi Putra
2016130007



JURUSAN INFORMATIKA
SEKOLAH TINGGI MANAJEMEN INFORMATIKA & KOMPUTER LIKMI
BANDUNG
2020

**PERANCANGAN ROBOT COLOR
*LIGHT FOLLOWER***

Oleh:
Bagus Budi Putra
2016130007

Bandung, 25 Agustus 2020
Menyetujui,

Sudimanto, S.T., M.Kom.

Pembimbing

Dhanny Setiawan, S.T., M.T.

Ketua Jurusan

JURUSAN INFORMATIKA
SEKOLAH TINGGI MANAJEMEN INFORMATIKA & KOMPUTER LIKMI
BANDUNG
2020

ABSTRAK

Perkembangan teknologi yang semakin meningkat di dunia ini, khususnya pada bidang industri yang menjadikan semua proses hampir dikerjakan oleh robot. Salah satu proses yang dapat dikerjakan otomatis oleh robot yaitu dalam bidang transportasi dalam industri tersebut. Perancangan robot *color light follower* ini bertujuan agar mempermudah dalam bidang transportasi dalam industri, robot ini akan berjalan otomatis sesuai dengan arahan warna cahaya yang diberikan, dan robot akan terus menerus berjalan jika warna cahaya yang diberikan terdeteksi dan dikenali oleh robot dan akan berhenti jika warna cahaya yang diberikan tidak dikenali robot atau robot tidak mendeteksi adanya warna cahaya.

Pengontrolan robot *color light follower* ini menggunakan Arduino MEGA 2560 sebagai pusat kendali pada robot. Dalam perancangan robot ini penggunaan sensor LDR (*Light Dependent Resistor*) ditujukan untuk melakukan validasi cahaya yang didapat dari senter. Robot ini juga menggunakan sensor warna TCS3200 yang digunakan untuk melakukan pengecekan warna cahaya yang didapat untuk menentukan arah robot berjalan. Motor DC pada robot ini digunakan untuk menggerakkan robot untuk berpindah tempat.

Hasil dari perancangan robot *color light follower* ini dapat bekerja maksimal dalam pencahayaan ruangan yang minim dan warna cahaya pada senter yang memiliki intensitas yang tinggi. Robot ini dapat berjalan sesuai dengan arah warna cahaya yang diberikan jika robot mengenali warna cahaya yang diberikan. Robot akan terus menerus berjalan jika warna cahaya terus disorotkan ke arah robot. Jika intensitas warna cahaya yang diberikan pada robot *color light follower* lebih besar dari intensitas warna cahaya dari senter maka robot tidak akan membaca warna cahaya pada senter dan robot akan mendeteksi warna merah terus menerus selama intensitas cahaya yang ditangkap robot lebih besar dari intensitas cahaya pada senter.

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa, karena berkat serta rahmat-Nya sehingga peneliti akhirnya dapat menempuh serta menyelesaikan Tugas Akhir ini dengan baik. Tugas Akhir ini berjudul “Perancangan Robot *Color Light Follower*” disusun untuk memenuhi persyaratan akademik serta mendapatkan gelar Sarjana Strata 1 (S1), di Sekolah Tinggi Manajemen Informatika dan Komputer LIKMI, Bandung.

Peneliti mengakui bahwa penulisan penelitian ini jauh dari kata sempurna, masih banyak kekurangan dalam diri peneliti untuk menyusun penelitian ini oleh sebab itu, peneliti terbuka dan menerima setiap masukan maupun kritikan dari para pembaca.

Dalam penyusunan penelitian ini, peneliti menyadari bahwa tidak lepas dari bantuan dan dorongan baik secara langsung atau tidak langsung dari berbagai pihak sehingga peneliti dapat menyelesaikan Tugas Akhir ini. Untuk itu perkenankanlah peneliti menyampaikan rasa terima kasih yang sebesar-besarnya dan penghargaan yang tidak terhingga terutama yang terhormat :

1. Bapak Sudimanto, S.Kom., M.Kom. selaku dosen pembimbing yang sudah banyak membantu penulis baik dalam meluangkan waktu, tenaga, dan pikiran untuk memberikan banyak saran dan arahan dalam membimbing peneliti mengerjakan Tugas Akhir ini.
2. Bapak Dhanny Setiawan, S.T., M.T. selaku ketua jurusan program studi S1 Informatika di STMIK LIKMI yang telah memberikan dukungan serta arahan kepada penulis.
3. Kedua orang tua serta keluarga besar yang telah mendukung dan memberi motivasi dan fasilitas selama penulis menyelesaikan tugas akhir ini.
4. STMIK LIKMI yang telah memberikan kesempatan bagi peneliti untuk menempuh pendidikan serta seluruh dosen di STMIK LIKMI atas ilmu dan kesabarannya dalam memberikan pengajaran yang terbaik.

5. Seluruh teman, sahabat bahkan kekasih peneliti yaitu Cindy Christianti yang selalu mendukung, membantu dan memberikan motivasi dalam menyelesaikan tugas akhir ini.
6. Seluruh Staff Elgreat, yaitu Bapak Wiradhi Ridwan dan Bapak Iwan Septiawan yang telah membantu mendukung penulis dan semangat yang diberikan selama peneliti mengerjakan Tugas Akhir ini.
7. Semua pihak yang terlibat secara tidak langsung dalam penyusunan Tugas Akhir ini dimana peneliti tidak dapat disebutkan satu-persatu.

Akhir kata peneliti hanya dapat berterima kasih dan berdoa sekiranya Tuhan memberikan berkat karunia dan rahmat-Nya serta membalas segala kebaikan pada semua pihak yang telah banyak membantu peneliti. Peneliti berharap semoga Tugas Akhir ini dapat bermanfaat bagi para pembaca.

Bandung, 25 Agustus 2020

Penulis

DAFTAR ISI

ABSTRAK.....	ii
KATA PENGANTAR	iii
DAFTAR ISI.....	v
DAFTAR GAMBAR	vii
DAFTAR TABEL	viii
DAFTAR SIMBOL	ix
DAFTAR LAMPIRAN	xi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan Masalah	3
1.5 Kegunaan Hasil.....	3
1.6 Metodologi Penelitian	3
1.7 Sistematika Penulisan.....	4
BAB II LANDASAN TEORI.....	5
2.1 Robot.....	5
2.1.1 Tipe Robot	5
2.1.2 Karakteristik Robot.....	6
2.2 Cahaya	7
2.3 Warna.....	7
2.4 Mikrokontroler.....	8
2.4.1 Mikrokontroler ATmega2560	8
2.4.2 Deskripsi Pin ATmega2560	9
2.5 Arduino	11
2.6 Sensor	12
2.6.1 Sensor Cahaya LDR (<i>Light Dependent Resistor</i>).....	13
2.6.2 Sensor warna TCS3200.....	13
2.7 Motor DC	15
2.8 <i>Driver</i> Motor L298N	16
2.9 Arduino IDE (<i>Integrated Development Environment</i>).....	17
2.10 Bahasa C.....	18
2.11 Rekayasa Perangkat Lunak	19
2.12 Pemrograman Berorientasi Objek (<i>Object Oriented Programming</i>)	20
2.13 UML (<i>Unified Modeling Language</i>)	20
2.13.1 Diagram <i>Use Case</i>	20

2.13.2	Diagram Kelas (<i>Class Diagram</i>)	21
2.13.3	<i>Deployment Diagram</i>	21
2.13.4	<i>State Machine Diagram</i>	22
BAB III ANALISIS DAN PERANCANGAN		23
3.1	Gambaran Umum.....	23
3.2	Spesifikasi Kebutuhan.....	23
3.2.1	Kebutuhan Perangkat Keras.....	23
3.2.2	Kebutuhan Perangkat Lunak	24
3.3	<i>Deployment Diagram</i>	24
3.4	<i>Use case Diagram</i>	26
3.5	<i>Class Diagram</i>	29
3.6	<i>State Machine Diagram</i>	30
3.7	Skematik Robot <i>Color Light Follower</i>	31
BAB IV IMPLEMENTASI DAN PENGUJIAN		32
4.1	Implementasi Perangkat Keras	32
4.2	Pengujian Modul dan Analisa.....	34
4.2.1	Pengujian Motor DC (<i>Direct Current</i>) dan <i>Motor Driver</i> L298N	34
4.2.2	Pengujian Sensor LDR (<i>Light Dependent Resistor</i>) dan Sensor Warna TCS3200 Pada Robot.....	35
4.3	Pengujian Sistem	38
BAB V KESIMPULAN DAN SARAN		42
5.1	Kesimpulan.....	42
5.2	Saran.....	42
DAFTAR PUSTAKA		44
DAFTAR LAMPIRAN		46
<i>Listing Program</i> Mikrokontroler Robot <i>Color Light Follower</i>		46

DAFTAR GAMBAR

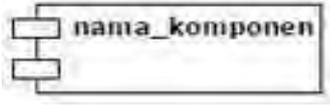
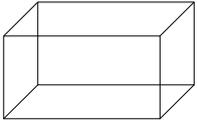
Gambar 2. 1 Konfigurasi Pin ATmega2560	9
Gambar 2. 2 Diagram Mikrokontroler ATmega2560	10
Gambar 2. 3 <i>Board</i> Arduino Mega2560	12
Gambar 2. 4 Sensor Cahaya LDR (<i>Light Dependent Resistor</i>)	13
Gambar 2. 5 Sensor Warna TCS3200	14
Gambar 2. 6 Motor DC <i>GearBox</i>	15
Gambar 2. 7 Motor <i>Driver</i> L298N	16
Gambar 2. 8 Jendela Utama Arduino IDE	17
Gambar 2. 9 Bagian <i>Toolbar</i> Arduino IDE	18
Gambar 3. 1 <i>Deployment Diagram</i> Robot <i>Color Light Follower</i>	24
Gambar 3. 2 <i>Use case diagram</i> Robot <i>Color Light Follower</i>	26
Gambar 3. 3 <i>class diagram</i> Robot <i>Color Light Follower</i>	29
Gambar 3. 4 <i>State Mesin Diagram</i> Robot <i>Color Light Follower</i>	30
Gambar 3. 5 Skematik Robot <i>Color Light Follower</i>	31
Gambar 4. 1 Robot <i>Color Light Follower</i> Tampak Depan	33
Gambar 4. 2 Robot <i>Color Light Follower</i> Tampak Samping	33
Gambar 4. 3 Robot <i>Color Light Follower</i> Tampak Atas	34

DAFTAR TABEL

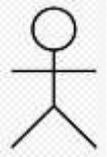
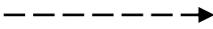
Tabel 2. 1 Deskripsi Papan Arduino Mega 2560	11
Tabel 3. 1 Skenario Normal diagram <i>use case</i> memberi warna cahaya	26
Tabel 3. 2 Skenario Alternatif diagram <i>use case</i> memberi warna cahaya	27
Tabel 3. 3 Skenario Normal diagram <i>use case</i> memproses perintah warna cahaya	27
Tabel 3. 4 Skenario Alternatif diagram <i>use case</i> memproses perintah warna cahaya.....	27
Tabel 3. 5 Skenario Normal diagram <i>use case</i> deteksi warna cahaya.....	28
Tabel 3. 6 Skenario Alternatif diagram <i>use case</i> Deteksi warna cahaya	28
Tabel 4. 1 Tabel hasil pengujian Motor DC.....	35
Tabel 4. 2 Tabel Hasil Pengujian Arah Putaran <i>Driver</i> Motor	35
Tabel 4. 3 Tabel Hasil Pengujian Sensor LDR	36
Tabel 4. 4 Tabel Hasil Pengujian Sensor LDR dan Sensor Warna TCS3200.....	37
Tabel 4. 5 Tabel Hasil Pengujian Sistem Robot	40

DAFTAR SIMBOL

1. Simbol *Deployment Diagram*

Nama Simbol	Simbol	Keterangan
Komponen		Komponen sistem
Node		Bagian – bagian hardware dalam sebuah sistem
Generalisasi		Menunjukkan hubungan antara elemen yang lebih umum ke elemen yang lebih spesifik

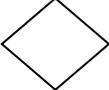
2. Simbol *Use case Diagram*

Nama Simbol	Simbol	Keterangan
Aktor		Orang, proses, atau aplikasi lain yang berinteraksi dengan sistem yang akan dibuat
<i>Use case</i>		Fungsi – fungsi atau proses – proses yang disediakan sistem sebagai unit – unit yang saling berukar pesan antar unit atau aktor
Asosiasi		Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
Sistem		Menspesifikasikan paket yang menampilkan sistem secara terbatas
<i>Include</i>		<i>Use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan

3. Simbol *Class Diagram*

Nama Simbol	Simbol	Keterangan
<i>Class</i>		Kelas pada struktur sistem
Asosiasi		Relasi antar kelas dengan makna kelas satu digunakan oleh kelas lainnya

4. Simbol *State Machine Diagram*

Nama Simbol	Simbol	Keterangan
<i>Start</i>		Digunakan untuk menggambarkan awal dari kejadian dalam diagram <i>state</i> mesin
<i>End</i>		Digunakan untuk menggambarkan titik akhir dari kejadian dalam diagram <i>state</i> mesin
<i>Event</i>		Kegiatan yang menyebabkan berubahnya status mesin
<i>State</i>		Sistem pada waktu tertentu
<i>Decision</i>		<i>Use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan
<i>Composite State</i>		Digunakan untuk sistem yang kompleks

DAFTAR LAMPIRAN

<i>Listing Program Mikrokontroler Robot Color Light Follower</i>	45
--	----

BAB I

PENDAHULUAN

1.1 Latar Belakang

Seiring dengan perkembangan teknologi dan informasi di zaman ini, membuktikan bahwa hampir semua peralatan *modern* di dunia ini dibuat untuk membantu manusia dalam menunjang pekerjaan terlebih dalam hal ketepatan, kecepatan dan dalam keadaan darurat sekalipun. Hal pokok inilah yang harus dirancang agar berjalannya proses terkait tidak terhambat atau mengganggu akibat teknologi yang digunakan.

Perkembangan teknologi meliputi berbagai macam bidang diantaranya di bidang industri, bidang hiburan, bidang pendidikan, kedokteran, bidang komunikasi, dan bidang bisnis. Sekarang ini, bidang industri dan hiburan menjadi bidang yang erat dalam menggunakan sebuah robot. Mulai dari peralatan industri yang mulai banyak digantikan dengan kinerja robot dan proses yang ada di dalamnya pun ikut digantikan dengan kinerja robot. Dalam dunia hiburan pun banyak digunakan untuk seseorang dalam menyalurkan hobi, bahkan digunakan untuk menunjang dalam kegiatan manusia itu sendiri.

Berdasarkan hasil penelitian yang dibuat oleh Roslidar, Alfatirta Mufti, dan Aris Akbarsyah dalam jurnalnya yang berjudul "Perancangan Robot *Light Follower* Untuk Kursi Otomatis dengan Menggunakan Mikrokontroler ATmega 328P" menyatakan bahwa robot akan bergerak ke arah intensitas cahaya yang paling besar. Dan pada hasil penelitian yang dibuat oleh Caroline, Ike Bayusari, dan Hermawati dalam jurnalnya yang berjudul "Pengaruh Cahaya Yang Diterima Sensor LDR (*Light Dependent Resistor*) Pada Robot Pengikut Cahaya" menyatakan bahwa semakin besar intensitas cahaya lingkungan sekitar maka semakin kecil intensitas cahaya senter yang dapat ditangkap oleh sensor LDR. Sehingga dari hasil penelitian yang dilakukan dalam jurnal tersebut, penulis ingin mengembangkan Robot *Light Follower* dengan menambahkan warna tertentu pada cahaya sebuah senter yang disorotkan ke robot tersebut untuk menentukan arah jalannya robot. Robot *Color Light Follower* adalah teknologi dimana sebuah robot dapat mengikuti

sebuah cahaya yang berguna sebagai penanda kemana robot harus berjalan dan menentukan arah. Robot *Color Light Follower* dapat diterapkan atau diaplikasikan pada transportasi di dalam industri pada saat keadaan darurat untuk menggantikan fungsi pada robot *Line Follower*, robot *Line Follower* bekerja dengan cara membaca garis yang dibuat untuk menentukan jalurnya, Robot *Line Follower* akan merasa kesulitan dalam membaca garis yang dibuat jika *Line* yang dibuat bergelombang atau rusak dan *Line* yang dibuat tertutup oleh air atau benda lain seperti tanah. Oleh sebab itu, Robot *Color Light Follower* dirancang untuk melengkapi kelemahan dari sebuah Robot *Line Follower*. Dalam membuat jalur atau rute untuk Robot *Color Light Follower*, penataan lampu menjadi hal yang utama agar robot dapat bergerak secara otomatis mengikuti jalur atau rute sesuai arahan yang telah ditentukan meskipun dalam tingkat pencahayaan yang kurang terang sekalipun.

Berdasarkan dari latar belakang yang disampaikan di atas, penulis akan membuat Tugas Akhir yang berkaitan dengan Robot pengikut cahaya atau Robot *Color Light Follower*. Maka, penulis akan merancang sebuah Tugas Akhir yang berjudul : **“PERANCANGAN ROBOT COLOR LIGHT FOLLOWER”**.

1.2 Rumusan Masalah

Adapun rumusan masalah yang disampaikan dalam pembuatan laporan tugas akhir ini adalah sebagai berikut :

1. Bagaimana Robot *Color Light Follower* dapat mengikuti cahaya yang telah ditentukan untuk menentukan arah?
2. Bagaimana sensor mendeteksi dan mengenali sebuah warna cahaya yang sudah ditentukan untuk robot tersebut berjalan sesuai warna cahaya yang diharapkan?

1.3 Tujuan

Tujuan dalam pembuatan perancangan robot berbasis *Color Light Follower* adalah untuk melengkapi dari kekurangan robot *Line Follower* dalam keadaan darurat pada sebuah industri.

1.4 Batasan Masalah

Adapun ruang lingkup bahasan yang akan dibatasi pada laporan pembuatan tugas akhir ini adalah sebagai berikut :

1. Jumlah warna yang dapat dikenali yaitu tiga warna primer cahaya yaitu warna merah, biru, hijau dan satu warna primer pigmen atau yang dikenal dengan warna primer cat yaitu warna kuning.
2. Untuk mendeteksi adanya sebuah cahaya pada robot, menggunakan sensor LDR (*Light Dependent Resistor*) dan untuk mengenali warna cahaya menggunakan sensor warna TCS3200.

1.5 Kegunaan Hasil

Hasil yang diharapkan dalam pembuatan perancangan robot *color light follower* antara lain adalah robot yang dapat secara otomatis dapat mengikuti arahan cahaya dan warna yang diberikan. Robot *color light follower* juga dapat menggantikan robot *line follower* dalam keadaan darurat di bidang industri.

1.6 Metodologi Penelitian

Metodologi yang digunakan pada penelitian ini antara lain:

1. Studi Literatur, yaitu dengan mengumpulkan kajian-kajian yang berkaitan dengan penelitian ini, berupa buku referensi, jurnal penelitian dan artikel yang terkait dengan penelitian ini, dan berbagai sumber lainnya yang berhubungan dengan penelitian ini.
2. Perancangan Sistem robot, yaitu dengan merancang sistem robot sesuai dengan tujuan penelitian.
3. Pembuatan Robot, yaitu mengimplimentasikan sistem yang telah dirancang ke dalam robot.
4. Pengujian Sistem, yaitu menguji sistem robot yang telah dibuat agar sesuai dengan tujuan penelitian.

5. Menganalisis Hasil, yaitu menganalisis hasil dari pengujian sistem robot yang telah dirancang.
6. Kesimpulan, yaitu memberikan kesimpulan dari hasil analisis yang dilakukan pada sistem robot yang dirancang.

1.7 Sistematika Penulisan

Untuk memudahkan pemahaman dari isi penulisan ini, maka disusun sistematika penulisan sebagai berikut :

BAB I PENDAHULUAN

Pada bab ini berisi mengenai latar belakang yang menjadi alasan penelitian, rumusan masalah, tujuan, batasan masalah, kegunaan hasil, metodologi penelitian dan sistematika penulisan yang digunakan.

BAB II LANDASAN TEORI

Bab ini membahas mengenai sumber-sumber teori dari perangkat keras seperti Arduino Mega 2560 , sensor warna TCS3200, sensor LDR, motor driver L298N, motor DC , perangkat lunak yang dibutuhkan dalam penelitian yaitu Arduino IDE, dan diagram – diagram yang dipakai seperti *deployment* diagram, *use case* diagram, *class* diagram, dan *state* mesin diagram.

BAB III ANALISIS DAN PERANCANGAN

Berisi mengenai gambaran umum dan tahapan-tahapan dari perancangan perangkat lunak dan perangkat keras serta diagram – diagram seperti *use case* diagram, skenario diagram, *class* diagram, *deployment* diagram, *state* mesin diagram dan skematik robot.

BAB IV IMPLEMENTASI DAN PENGUJIAN

Bab ini berisi mengenai implementasi dan hasil-hasil pengujian dari model yang dirancang seperti pengujian pada sensor yang digunakan.

BAB V KESIMPULAN DAN SARAN

Berisi mengenai kesimpulan yang didapat dari hasil-hasil perancangan yang dibuat dan saran untuk pengembangan perancangan lebih lanjut.

BAB II

LANDASAN TEORI

2.1 Robot

Robot berasal dari kata *Robota* (Cekoslovakia) yang bermakna pekerja yang dipaksa. (Kadir,2019:6).

Definisi robot menurut Supriyanto yang ditulis di bukunya yang berjudul "*Robotika*" adalah sebagai berikut :

"Robot adalah sistem atau alat yang dapat berperilaku atau meniru perilaku manusia dengan tujuan untuk menggantikan dan mempermudah kerja/aktifitas manusia."
(Supriyanto,2010:1)

Definisi dari International Standard Organization (ISO 8373) adalah sebagai berikut :

"Robot is an actuated mechanism programmable in two or more axes with a degree of autonomy, moving within its environment, to perform intended task."

Dari definisi tersebut dapat disimpulkan pengertian robot adalah sebuah perangkat yang dapat berjalan secara otomatis dan memiliki berbagai macam fungsi atau tugas yang dapat diprogram ulang dan diperintah oleh manusia.

2.1.1 Tipe Robot

Secara umum tipe robot dibagi menjadi dua kategori yaitu seperti yang tertera di bawah ini:

1. *Robot Manipulator*

Ciri robot ini adalah memiliki lengan(*arm robot*).

2. *Mobilerobot*

Mobilerobot adalah jenis robot yang dapat diartikan robot yang dapat memindahkan dirinya dari suatu tempat ke tempat lainnya. *Mobilerobot* dibagi juga dibagi dalam 3 bagian diantaranya:

a. *Ground Robot*

Ground robot dibagi menjadi 2 bagian diantaranya:

- (1) *Robot* beroda
- (2) *Robot* berkaki

Berkembangnya *ground robot* menjadikan munculnya *robot humanoid* dengan tampilan menyerupai tubuh manusia. *Robot humanoid* ini dapat berhubungan secara sosial.

b. *Robot submarine*

c. *Robot aerial*

(Budiharto, 2014:4-5)

2.1.2 Karakteristik Robot

Karakteristik pada robot dibagi menjadi 4, diantaranya:

1. *Sensing*

Karakteristik ini mengharuskan robot untuk dapat mendeteksi lingkungan di sekitarnya. Contohnya adalah dapat mendeteksi halangan, dapat mendeteksi panas, dapat mendeteksi adanya suara dan *images*.

2. Mampu bergerak

Karakteristik ini mengharuskan robot dapat bergerak menggunakan kaki atau roda. Tetapi dalam beberapa kondisi, tidak selalu robot harus berjalan menggunakan kaki atau roda, contoh kondisi ini adalah robot yang diharapkan dapat terbang atau berenang.

3. Cerdas

Robot diharuskan memiliki kecerdasan yang diprogram oleh manusia untuk memutuskan aksi atau tindakan yang tepat dan akurat.

4. Membutuhkan energi yang memadai

Untuk melakukan aksi atau menjalankan sesuai dengan fungsinya, robot harus diberikan catu daya yang memadai agar unit pengontrol dan aktuator dapat berjalan dengan baik.

(Budiharto,2014:6)

2.2 Cahaya

Definisi cahaya menurut Faridah yang ditulis di dalam buku yang berjudul “Menenal Lebih Dekat dengan Cahaya dan Warna” sebagai berikut:

“Cahaya adalah energi eletromagnetik dengan spektrum frekuensi yang mempunyai panjang gelombang 360 nm – 780 nm.”

(Faridah,2018:4)

Dari definisi – definisi yang dijabarkan tersebut dapat disimpulkan bahwa cahaya adalah sebuah energi yang memiliki panjang gelombang dan dapat dilihat oleh makhluk hidup.

2.3 Warna

Di dalam buku yang berjudul “*Journalism Today*” yang ditulis oleh Fachruddin:

“Warna adalah spektrum tertentu yang terdapat di dalam suatu cahaya sempurna atau berwarna putih”

(Fachruddin,2019:263).

Warna menurut Hendi Hendratman yang ditulis di dalam bukunya yang berjudul “*Computer Graphic Design*” sebagai berikut :

“Warna adalah salah satu komponen desain yang membentuk keindahan sekaligus menimbulkan persepsi psikologis, sugesti, suasana tertentu.”

(Hendratman,2017:105)

Menurut Wucius dalam buku yang ditulis oleh Nugroho yang berjudul “Manajemen Warna dan Desain” terdapat klasifikasi dan nama-nama warna yaitu warna primer, warna sekunder, warna *intermediate*, warna tersier, dan warna kuartar.

Menurut kejadiannya, warna dibagi menjadi dua, yang pertama warna *additive* yaitu warna yang berasal dari cahaya yang disebut dengan spektrum dan yang kedua warna *subtractive* yaitu warna yang berasal dari bahan yang disebut dengan pigmen. (Nugroho,2015:25 dan 33)

Menurut Anditya dalam bukunya yang berjudul “Buku Pintar Desain Arsitektur” pengertian warna adalah sebagai berikut :

“Warna adalah sebuah gejala visual yang terkadang tidak begitu diperhatikan oleh manusia, namun kehadirannya menambah nilai tersendiri dalam kehidupan manusia.”

(Anditya,2016:11)

Dari definisi - definisi diatas maka dapat disimpulkan definisi warna adalah bagian dari suatu cahaya yang memiliki spektrum tertentu dan perbedaan dari suatu warna akibat dari panjang – panjang gelombang yang berbeda.

2.4 Mikrokontroler

Menurut Andrianto, mikrokontroler adalah sebuah komputer kecil (*“special purpose computers”*) di dalam sebuah IC/chip yang didalamnya terdapat CPU, memori, timer, saluran komunikasi serial dan paralel, port masukan atau keluaran, ADC, dll yang digunakan sebagai pengendali yang mengatur semua proses. (Andrianto,2015:1)

Mikrokontroler menurut jurnal yang berjudul “Implementasi Model Robot Edukasi menggunakan Mikrokontroler ATMEGA8 untuk Robot Pemadam Api” adalah suatu alat elektronika digital yang mempunyai masukan dan keluaran serta kendali dengan program yang bisa ditulis dan dihapus dengan cara khusus, cara kerja mikrokontroler sebenarnya membaca dan menulis data.

2.4.1 Mikrokontroler ATmega2560

Mikrokontroler ATmega2560 memiliki beberapa fitur antara lain (*datasheet Atmega2560*):

1. Mikrokontroler ATmega2560 memiliki 32 x 8-bit register serbaguna.

2. Memiliki 135 macam instruksi, yang hampir semua dieksekusi dalam satu siklus *clock*.
3. Kecepatan mikrokontroler ATmega2560 mencapai 16 MIPS dengan *clock* 16 MHz.
4. Memiliki 256 KB *flash* memori dan pada arduino sendiri memiliki *bootloader* yang menggunakan 8 KB dari *flash* memori yang digunakan sebagai *bootloader*.
5. Mikrokontroler ATmega2560 memiliki SRAM (*Static Random Access Memory*) sebesar 8 KB.
6. Memiliki EEPROM sebesar 4 KB.
7. Master / Slave SPI Serial *interface*.
8. Memiliki pin I/O digital sebanyak 54 pin 15 diantaranya PWM *output*.
9. Memiliki pin analog *input* sebanyak 16 pin.

2.4.2 Deskripsi Pin ATmega2560



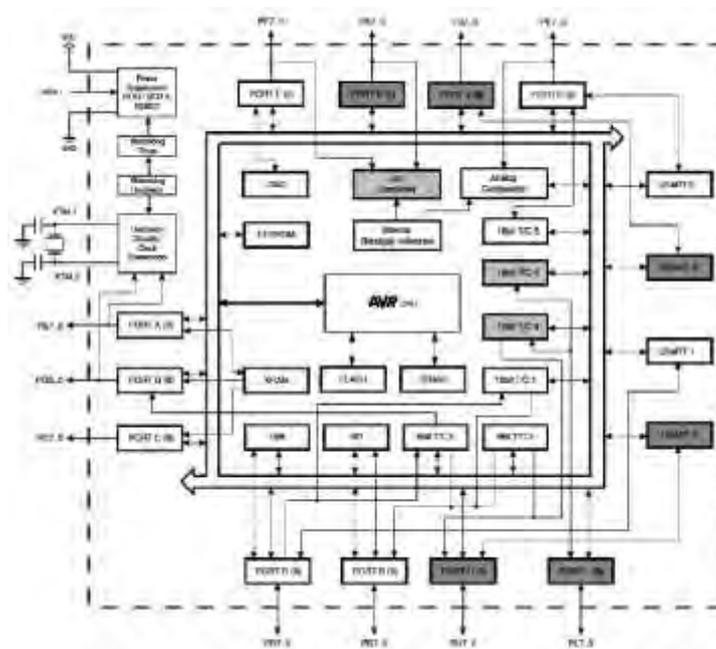
Gambar 2. 1
Konfigurasi Pin ATmega2560

(Sumber : <https://www.arduino.cc/en/Hacking/PinMapping2560>)

Adapun deskripsi dari susunan gambar pin mikrokontroler ATmega2560 diatas adalah sebagai berikut (*datasheet* Atmega2560) :

1. VCC merupakan pin *digital* untuk sumber masukan tegangan atau catu daya.
2. GND merupakan *ground*.
3. Port A (PA0 - PA7), Port B (PB0 – PB7), Port C (PC0 – PC7), Port D (PD0 – PD3, PD7), Port E (PE0, PE1, PE3 – PE5), Port G (PG0-PG2, PG5), Port H (PH0, PH1, PH3-PH6), Port J (PJ0-PJ1), Port L (PL0-PL7) merupakan pin *digital* untuk *input/output* untuk jalur data 8-bit dengan *internal pull-up resistor*.

4. ADC Port (PF0 – PF7, PK0 – PK7) merupakan pin *input* ADC (*Analog to Digital Converter*).
5. Port PE0, PD2, PJ0, PH0 merupakan pin RX untuk menerima data serial TTL.
6. Port PE1, PH1, PD3, PJ1 merupakan pin TX untuk mengirimkan data serial TTL.
7. Port SPI (Pin 50 (MISO), Pin 51 (MOSI), Pin 52 (SCK), Pin 53 (SS), pada pin ini mendukung untuk komunikasi SPI menggunakan *library* SPI.
8. TWI (Pin 20 (SDA) dan Pin 21 (SCL)) adalah pin untuk komunikasi TWI menggunakan *library* *Wire*.
9. Reset merupakan pin untuk menghidupkan ulang mikrokontroler.
10. XTAL1 dan XTAL2 merupakan pin *external clock*.
11. AVCC merupakan pin untuk memberikan *supply* tegangan ADC.
12. AREF merupakan referensi tegangan untuk *input* analog.
13. LED (Pin 13) merupakan pin untuk mengatur LED, ketika pin diset bernilai *HIGH* maka LED akan menyala, dan ketika diset bernilai *LOW* LED akan mati.



Gambar 2. 2
 Diagram Mikrokontroler ATmega2560
 (Sumber : datasheet atmega2560)

2.5 Arduino

Definisi arduino menurut McRoberts adalah sebagai berikut (McRoberts,2010:3):

“Arduino is a tiny computer that you can program to process inputs and outputs between the device and external components you connect to it.”

Sedangkan definisi arduino menurut Mochamad Fajar Wicaksono di bukunya yang berjudul “Mudah belajar Mikrokontroler Arduino yaitu (Wicaksono,2017:1):

“Arduino adalah sebuah board mikrokontroler yang bersifat open source dimana desain skematik dan PCB bersifat open source, sehingga dapat dilakukan modifikasi” (Andrianto,2016:12).

Dapat disimpulkan dari pengertian diatas, arduino adalah suatu perangkat elektronik yang berfungsi untuk menghubungkan perangkat – perangkat lainnya yang mendukung dengan dikendalikan oleh mikrokontroler sebagai otak pengendalian.

Menurut Andrianto dalam bukunya yang berjudul “ Arduino Belajar Cepat dan Pemograman “ Arduino Mega 2560 adalah sebuah *board* arduino yang menggunakan ic mikrokontroler Atmega 2560 (Andrianto,2016:27).

Spesifikasi papan arduino mega 2560 yang di lansir dari *datasheet* mega 2560 sebagai berikut :

Tabel 2. 1
Deskripsi Papan Arduino Mega 2560

Mikrokontroler	Atmega2560
Tegangan operasi	5 V
Tegangan <i>input</i> (rekomendasi)	7 – 12 V
Batas Tegangan <i>input</i>	6 – 20 V
Pin digital I/O	54 pin(6 pin <i>output</i> PWM)
Pin analog <i>input</i>	16 pin
Arus DC per I/O pin	20 mA
Arus DC untuk pin 3.3 V	50 mA
<i>Flash memory</i>	256 KB, 8 KB pada memori ini digunakan oleh bootloader
SRAM	8 KB
EEPROM	4 KB
<i>Clock</i>	16 MHz
Panjang	101,5 mm
Lebar	53,4 mm
Berat	37 g

(Sumber : DatasheetMega2560)



Gambar 2. 3

Board Arduino Mega2560

(sumber <https://www.eduhublk.com/product/arduino-mega-2560/>)

2.6 Sensor

Sensor adalah piranti yang ditunjukkan untuk mendeteksi keberadaan suatu kejadian atau perubahan nilai di sekitar lingkungan piranti tersebut dan memberikan tanggapan berupa suatu keluaran atau *output* (Kadir,2018:2).

Jadi sensor dapat diartikan sebagai alat yang digunakan untuk mendeteksi adanya perubahan lingkungan berupa fisik atau kimia. Pada masa kini,sensor telah dibuat dengan ukuran sekecil mungkin dengan orde nanometer, hal ini dibuat agar dapat memudahkan dalam pemakaian dan untuk menghemat energi. Adapun macam – macam sensor sebagai berikut :

1. Sensor *Proximity*
2. Sensor Magnet
3. Sensor Sinar

Sensor ini dibagi dalam tiga macam yaitu *fotovoltaic*, fotokonduktif dan fotolistrik.

4. Sensor Ultrasonik
5. Sensor Tekanan
6. Sensor Kecepatan atau RPM

Sensor kecepatan bekerja dengan cara mengukur pulsa magnetis atau induksi yang timbul saat medan listrik terjadi akibat dari adanya perputaran roda pada suatu poros.

7. Sensor *Encoder* (penyandi)

Sensor *encoder* biasanya digunakan untuk merubah putaran atau gerakan linear menjadi sinyal digital.

8. Sensor Suhu

Sensor suhu adalah sensor yang dapat mengubah besaran panas atau kalor menjadi sinyal listrik untuk dapat mendeteksi adanya perubahan suhu dari suatu objek. Sensor suhu terdapat 4 jenis yaitu *thermocouple*, *resistance temperature detector* (RTD), *termistor* dan *IC sensor*.

2.6.1 Sensor Cahaya LDR (*Light Dependent Resistor*)

Menurut Andrianto, sensor cahaya LDR didefinisikan sebagai berikut “*Sebuah resistor yang dapat berubah nilainya sesuai dari jumlah cahaya yang menyinari permukaan sensor. Nilai resistansi pada sensor LDR ini akan berubah turun ketika cahaya semakin terang.*” (Andrianto,2016:83).



Gambar 2. 4

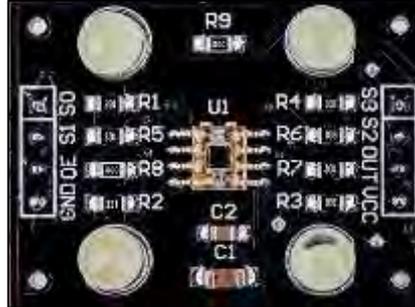
Sensor Cahaya LDR (*Light Dependent Resistor*)

(Sumber : <https://electroino.com/sensor-cahaya-ldr/>)

2.6.2 Sensor warna TCS3200

Sensor warna jenis TCS3200 adalah sensor warna buatan perusahaan *Texas Advanced Optoelectronic Solutions* (TAOS). Sensor ini bekerja dengan cara mengubah cahaya warna menjadi sebuah frekuensi (Kadir,2018:319).

Sensor warna TCS3200 adalah sebuah sensor yang dibangun dengan menggunakan chip sensor TAOS TCS3200 RGB, sensor ini dapat mendeteksi berbagai jenis warna berdasarkan panjang gelombang (Wicaksono,2019:125).



Gambar 2. 5
Sensor Warna TCS3200

(sumber : <http://www.belajarduino.com/2016/12/tutorial-penggunaan-sensor-warna-tcs230.html>)

Modul sensor warna TCS3200 mengandung 8 pin yaitu (Kadir,2018:319):

1. VCC
Pin ini adalah pin yang harus dihubungkan ke tegangan 5V.
2. GND
Pin ini adalah pin *ground*.
3. S0 dan S1
Pin ini adalah pin masukan atau *input* yang menentukan penyekala frekuensi keluaran.
4. S2 dan S3
Pin ini adalah pin masukan atau *input* untuk menentukan tipe fotodiode yang digunakan untuk memfilter warna.
5. *Output Enable* (OE)
Pin ini adalah pin masukan atau *input* untuk menentukan keadaan impedansi tinggi jika diberi nilai *HIGH*. Normalnya OE cukup diisi dengan nilai *LOW* atau tidak dihubungkan kemana – mana.
6. *OUT*
Pin ini berfungsi untuk mengeluarkan frekuensi.

2.7 Motor DC

Motor DC adalah sebuah motor yang dapat bergerak berputar 360 derajat, biasanya disebut dinamo dan biasanya digunakan untuk penggerak roda (Andrianto,2016:131). Sirkuit internal motor DC terdiri dari kumparan atau lilitan konduktor. Setiap arus yang mengalir melalui sebuah konduktor akan menimbulkan medan magnet, karena medan magnet ini yang menyebabkan motor DC akan berputar.



Gambar 2. 6
Motor DC GearBox

(Sumber : https://www.alibaba.com/product-detail/DC-12V-250rpm-Smart-robot-car_60783921683.html)

Dari sebuah artikel yang ditulis oleh Universitas Binus menyebutkan 3 komponen utama pada Motor DC sebagai berikut :

1. Kutub Medan Magnet

Pada Motor DC terjadi 2 interaksi, yaitu Kutub Utara dan Kutub Selatan. Motor DC memiliki kutub medan yang tetap dan kumparan pada motor DC yang bertugas untuk menggerakkan *bearing* pada ruang di antara kutub medan.

2. Dinamo

Arus yang masuk, menuju ke kumparan Motor DC akan menjadi elektromagnet. Kumparan Motor DC yang berbentuk silinder dihubungkan ke as penggerak untuk menggerakkan beban.

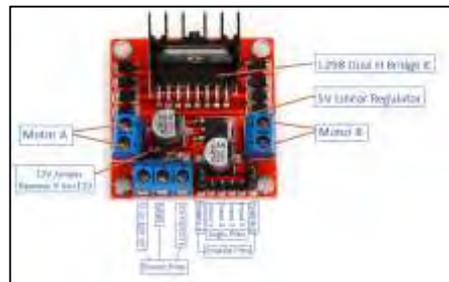
3. *Commutator* Motor *Direct Current* (DC)

Commutator pada Motor DC memiliki kegunaan untuk membalikan arah arus listrik dalam kumparan Motor DC. Komponen ini juga berguna untuk membantu transmisi arus antara kumparan Motor DC dengan catu daya.

2.8 **Driver Motor L298N**

Driver Motor L298N adalah sebuah *H-Bridge Dual Motor Controller 2A* yang memungkinkan kita untuk mengatur arah putaran maupun kecepatan dari satu atau dua motor DC (Wicaksono,2019:394).

Fungsi dari driver motor LN298N adalah untuk membalikan polaritas tegangan yang berada pada motor, sehingga pada motor terbagi menjadi 2 arah yaitu, arah yang pertama adalah arah yang searah dengan jarum jam dan arah yang kedua adalah arah yang berlawanan dengan arah jarum jam. Pada *driver* motor ini, memungkinkan dapat mengatur atau menjalankan 2 Motor DC secara bersamaan atau pun secara bergantian. *Driver* Motor ini juga berfungsi sebagai pengatur tegangan pada Motor DC dengan rentang 0 V sampai dengan tegangan maksimum yang ada.



Gambar 2. 7
Motor *Driver* L298N

(Sumber : <https://electronics hobbyists.com>)

Adapun fitur kegunaan dari *Driver* Motor LN298N yaitu :

1. *Speed Control Mode*

Speed Control Mode ini berisi barisan *jumper* yang menunjukkan mode yang akan digunakan yaitu, *Pulse Width Modulation* (PWM) atau *Phased Locked Loop* (PPL). Pada pin E1 dan E2 digunakan untuk pengaturan PWM, sedangkan pada pin M1 dan M2 digunakan untuk pengaturan PPL.

2. *Power Selection Mode*

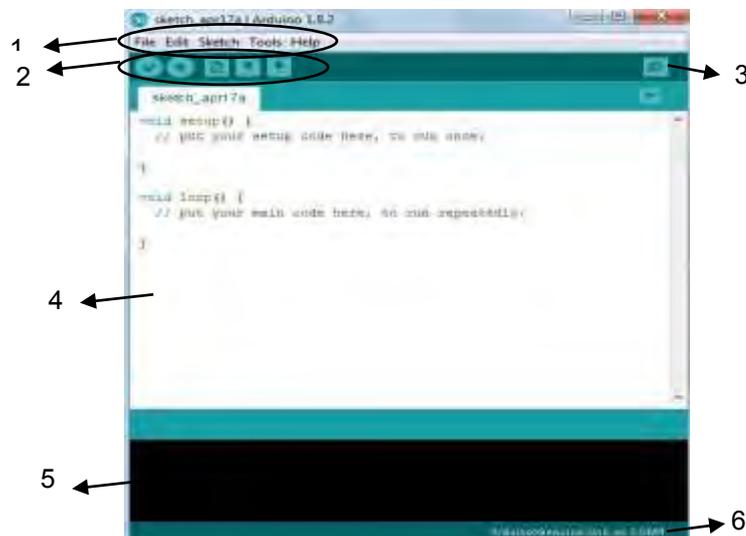
Pada fitur ini berisi pin untuk memilih *power supply* yang digunakan *Motor Driver* yang berasal dari Arduino atau *eksternal power supply* dengan cara menggeser pin VIN (tegangan yang berasal dari Arduino) menjadi PWRIN (tegangan yang berasal dari luar atau *eksternal*).

2.9 Arduino IDE (*Integrated Development Environment*)

Arduino IDE adalah *software* yang digunakan untuk menulis kode program Arduino, mengkompilasi menjadi kode biner dan meng-*upload*-nya pada *board* Arduino (Syahwil, 2013:39).

Arduino IDE adalah perkakas yang memungkinkan untuk menulis sketsa, mengkompilasi, dan mengunggahkannya ke papan Arduino (Kadir,2018:5)

Dari pengertian diatas dapat disimpulkan Arduino IDE adalah sebuah perangkat lunak yang digunakan untuk menulis kode program pada Arduino serta memprogram papan Arduino dengan bahasa java yang dapat berjalan pada sistem operasi Mac, Linux, dan Windows.



Gambar 2. 8
Jendela Utama Arduino IDE

(Sumber : <https://support.content.office.net/en-us/media/e8c360e1-2b32-45db-b9d7-d43abc86af2f.png>)

Jendela utama pada Arduino IDE pada gambar diatas terdiri dari 6 bagian yaitu :

1. Bagian atas (nomor 1) yang terdiri dari menu bar yang berisi *file*, *edit*, *sketch*, *tools* dan *help*.
2. Nomor 2 adalah *toolbar* yang berisi *verify*, *Upload* kode program ke Arduino, *new file*, *open project*, *save*.
3. Nomor 3 serial monitor
4. Nomor 4 yang merupakan tempat untuk melakukan penulisan kode program.
5. Nomor 5 merupakan area untuk menampilkan pesan atau teks *console* yang terdiri dari status dan pesan *error*.
6. Nomor 6 berisi tipe Arduino dan *port* yang dipakai.



Gambar 2. 9
Bagian Toolbar Arduino IDE

(Sumber : <https://support.content.office.net/en-us/media/e8c360e1-2b32-45db-b9d7-d43abc86af2f.png>)

Fungsi pada bagian *toolbar* di atas adalah :

1. *Verify* berfungsi untuk mengecek kode program atau *sketch* yang *error* sebelum meng-*upload* kode program ke papan Arduino.
2. *Upload* berfungsi untuk meng-*compile* dan meng-*upload* ke papan Arduino.
3. *New Sketch* berfungsi untuk membuat *sketch* baru.
4. *Open* berfungsi untuk membuka daftar *sketch* yang pernah dibuat.
5. *Save* berfungsi untuk menyimpan *sketch* yang dibuat pada *sketchbook*.
6. *Serial Monitor* berfungsi untuk menampilkan data serial yang terhubung dengan papan Arduino.

2.10 Bahasa C

Bahasa C adalah bahasa pemrograman yang digunakan dalam Arduino IDE.

Bahasa C adalah bahasa yang paling umum digunakan dalam memrogram sebuah

software dan mikrokontroler karena menghasilkan kode objek yang sangat kecil dan juga dapat dijalankan pada banyak *platform* seperti Windows, Unix dan Linux.

Dalam menyusun *sketch*, Arduino menggunakan bahasa C yang *multiplatform*, oleh karena itu Arduino dapat dijalankan pada sistem operasi yang umum digunakan, seperti Windows, Linux dan Mac. *Sketch* adalah kode program yang ditulis pada Arduino. Di dalam *sketch* sendiri memiliki 2 fungsi yaitu *void setup() { }* dan *void loop() { }* (Syahwill, 2013:80).

Void setup () { } memiliki fungsi yang digunakan untuk memulai sebuah *sketch*. Di dalam fungsi ini berisi inisiasi pada variabel, pin dan *libraries*. Pada fungsi ini hanya dijalankan satu kali ketika *sketch* diunggah atau pada saat papan Arduino di *reset*.

Void loop () { } memiliki fungsi yang akan dieksekusi ketika *void setup () { }* selesai dikerjakan. Fungsi ini akan terus diulang terus menerus untuk mengontrol papan Arduino sampai catu daya (*power*) dilepas.

2.11 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak menurut Pressman dalam bukunya yang berjudul "*Rekayasa Perangkat Lunak*" yang diterjemahkan oleh Adi Nugroho dkk adalah sebagai berikut :

"Rekayasa perangkat lunak mencakup proses, metode, dan alat-alat yang memungkinkan sistem berbasis komputer yang kompleks dibangun secara tepat waktu dan berkualitas" (Pressman,2010:15).

Perancangan perangkat lunak yang baik menurut Rosa (Rosa ,2015:2) yaitu :

1. Perangkat lunak tersebut dibangun dengan rekayasa (*software engineering*) bukan diproduksi secara manufaktur atau pabrikan.
2. Perangkat lunak sendiri tidak pernah usang atau disebut juga *ware out* karena kecacatan dalam perangkat lunak dapat diperbaiki.
3. Barang produksi di dalam dunia pabrikan biasanya komponen barunya akan terus diproduksi, sedangkan pada perangkat lunak biasanya terus diperbaiki seiring bertambahnya kebutuhan.

2.12 Pemrograman Berorientasi Objek (*Object Oriented Programming*)

Definisi pemrograman berorientasi objek menurut Sianipar dalam bukunya yang berjudul "Teori dan Implementasi Pemrograman Berorientasi Objek menggunakan C++" adalah:

"Suatu pendekatan konseptual tertentu untuk mendesain program."

(Sianipar,2012:1)

Definisi pemrograman berorientasi objek menurut Rosa dalam bukunya yang berjudul "Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek" adalah :

"Suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya."

(Rosa,2018:100)

2.13 UML (*Unified Modeling Language*)

Menurut Munawar dalam bukunya yang berjudul " Analisis Perancangan Sistem Berorientasi Objek dengan UML ", pengertian UML seperti di bawah ini:

"UML adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi obyek."

(Munawar,2018:49)

UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung (Rossa,2018:137).

Menurut Nugroho dalam bukunya yang berjudul " Rekayasa Perangkat Lunak Berorientasi Objek dengan Metode USDP" pengertian UML seperti di bawah ini:

"Bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma berorientasi objek."

(Nugroho,2010:6).

2.13.1 Diagram Use Case

Use case adalah salah satu diagram yang ada dalam UML (*Unified Modeling Language*) yang merupakan pemodelan untuk melakukan (*behavior*) aplikasi perangkat lunak yang akan dibuat (Rosa ,2010:215).

Diagram *use case* adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. (Munawar,2018:89).

Diagram *use case* merupakan unit fungsionalitas koheren yang diekspresikan sebagai transaksi-transaksi yang terjadi antara aktor-aktor dan *use case-use case* dan memperlihatkan aktor-aktor mana yang berpartisipasi dalam masing-masing *use case* (Nugroho,2010:34).

2.13.2 Diagram Kelas (*Class Diagram*)

Diagram kelas atau *class diagram* merupakan gambaran struktur aplikasi berorientasi objek dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun aplikasi (Rosa ,2010:218).

Diagram kelas menggambarkan atribut, *operation*, dan juga *constraint* yang terjadi pada sistem.*Class diagram* menunjukkan koleksi *Class*, antarmuka, asosiasi, kolaborasi, dan *constraint* (Munawar,2018:101).

Tujuan dari diagram kelas adalah sebagai berikut (Munawar,2018:102) :

1. Diagram kelas digunakan untuk analisis dan desain pandangan statis aplikasi.
2. Diagram kelas digunakan untuk menjelaskan tanggungjawab suatu sistem.
3. Diagram kelas bertujuan sebagai basis untuk diagram komponen dan penyebaran (*deployment*).
4. Diagram kelas digunakan untuk *forward and reverse engineering*.

2.13.3 Deployment Diagram

Tujuan dari *deployment diagram* adalah sebagai berikut (Munawar,2018:174):

1. *Deployment* diagram digunakan untuk visualisasi topologi perangkat keras suatu sistem.
2. *Deployment* diagram digunakan untuk menjelaskan komponen perangkat keras yang digunakan untuk penyebaran komponen *software*.

3. *Deployment* diagram digunakan untuk menjelaskan pemrosesan *runtime* pada suatu *node*.

2.13.4 **State Machine Diagram**

State mesin diagram adalah diagram yang mendeskripsikan perilaku dinamis pada objek-objek selama berjalannya waktu dengan memodelkan siklus hidup objek-objek yang berasal dari masing-masing kelas.

(Nugroho,2010:47)

State mesin diagram menurut S Rossa dan Shalahuddin dalam bukunya yang berjudul “ Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek edisi Revisi “ merupakan diagram yang digunakan untuk menggambarkan perubahan status atau transisi status dari sebuah mesin atau sistem atau objek.

(S Rossa ,2018:163)

Tujuan dari *state machine diagram* adalah sebagai berikut
(Munawar,2018:110) :

1. *State machine diagram* digunakan untuk memodelkan aspek dinamis dari suatu sistem.
2. *State machine diagram* digunakan untuk memodelkan waktu hidup reaktif suatu sistem.
3. *State machine diagram* bertujuan untuk mendeskripsikan berbagai status objek.

BAB III

ANALISIS DAN PERANCANGAN

3.1 Gambaran Umum

Robot pengikut cahaya atau Robot *Color Light Follower* merupakan robot yang dirancang agar dapat mengikuti sebuah cahaya secara otomatis. Robot ini bertujuan untuk melengkapi fungsi robot *line follower* sebagai penunjuk arahnya.

Robot *Color light follower* ini dibuat dengan menggunakan sensor LDR (*Light Dependent Resistor*) sebanyak 4 buah yang berfungsi untuk mendeteksi adanya sebuah cahaya. Robot ini juga menggunakan bantuan sensor warna (TCS3200) sebanyak 4 buah yang berfungsi untuk memvalidasi warna yang diberikan sebagai arahan arah, serta menggunakan 2 buah Motor DC dan 1 buah Driver Motor L298N dan satu buah roda bebas (*Ball Caster*) yang digunakan agar robot dapat berpindah dari satu tempat ke tempat lainnya, dan juga menggunakan papan Arduino sebagai pusat pengendali robot.

Robot pengikut cahaya ini bekerja dengan cara cahaya yang berwarna akan disorotkan ke sebelah robot, cahaya yang disorotkan akan divalidasi dengan bantuan sensor LDR sebagai validasi adanya sebuah cahaya dan robot tersebut akan memvalidasi kembali warna yang diberikan melalui sensor warna, dan robot akan berjalan sesuai dengan arahan warna cahaya yang diberikan sesuai dengan warna yang sudah ditentukan yaitu warna kuning untuk bergerak maju, warna biru untuk bergerak mundur, warna merah untuk belok kanan dan warna hijau untuk belok ke kiri. Jika warna yang disorotkan ke robot tersebut belum terdaftar, maka robot tidak akan bergerak.

3.2 Spesifikasi Kebutuhan

3.2.1 Kebutuhan Perangkat Keras

Adapun kebutuhan perangkat keras yang digunakan adalah :

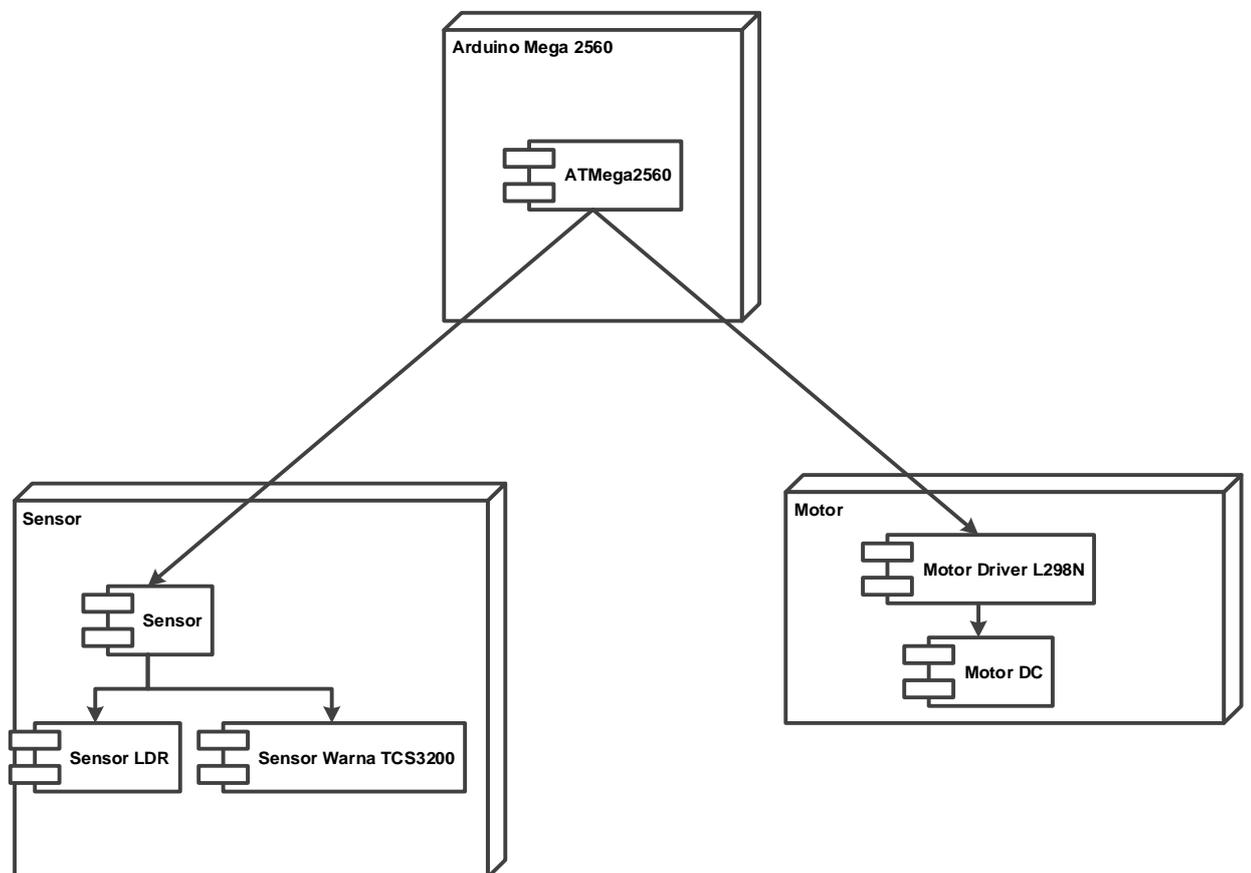
1. Papan Arduino Mega 2560
2. Sensor Cahaya LDR (*Light Dependent Resistor*)
3. Sensor Warna TCS320

4. Motor Driver L298N
5. Motor DC
6. Papan Rangkaian (*Breadboard*)

3.2.2 Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak yang digunakan adalah Arduino IDE.

3.3 Deployment Diagram



Gambar 3. 1
Deployment Diagram Robot Color Light Follower

Pada gambar 3.1 adalah *deployment* diagram Robot *Color Light Follower* :

1. Sensor LDR / sensor cahaya

Sensor LDR atau sensor cahaya yang digunakan berjumlah 4 (empat) buah. Sensor tersebut berada pada posisi depan, kanan, kiri, dan belakang. Sensor LDR ini berfungsi untuk mendeteksi adanya sebuah cahaya. Sensor ini adalah masukan yang akan diproses di dalam papan Arduino Mega 2560, sehingga papan Arduino Mega 2560 dapat mengenali intensitas cahaya yang diberikan.

2. Sensor Warna TCS3200

Sensor TCS3200 yaitu sensor untuk mendeteksi adanya sebuah warna yang akan diubah menjadi gelombang frekuensi. Sensor ini berjumlah 4 (empat) buah pada posisi depan, belakang, kanan dan kiri yang berfungsi untuk mengenali warna yang diberikan untuk menentukan arah berjalannya robot tersebut dan menentukan robot dapat berjalan atau tidak.

3. *Driver* motor L298N

Driver Motor yang digunakan pada robot *Color light follower* berjumlah 1 (satu) buah. Jenis *Driver* Motor yang dipakai berjenis L298N, yang berfungsi sebagai pengendali motor DC. Papan Arduino Mega 2560 akan memberikan masukan kepada *Driver* Motor dengan nilai *HIGH* / *LOW* yang kemudian memberikan keluaran kepada Motor DC yang digunakan sebagai roda pada robot *Color light follower*.

4. Motor DC

Motor DC yang dipakai dalam robot *Color light follower* berjumlah 2 (dua) buah yang dipasangkan pada sisi kanan dan sisi kiri pada robot, yang berfungsi sebagai roda agar robot dapat berjalan. Motor DC yang digunakan pada robot *color light follower* ini dilengkapi sebuah *gearbox*. *Gearbox* ini berfungsi untuk menambah tenaga putaran pada Motor DC.

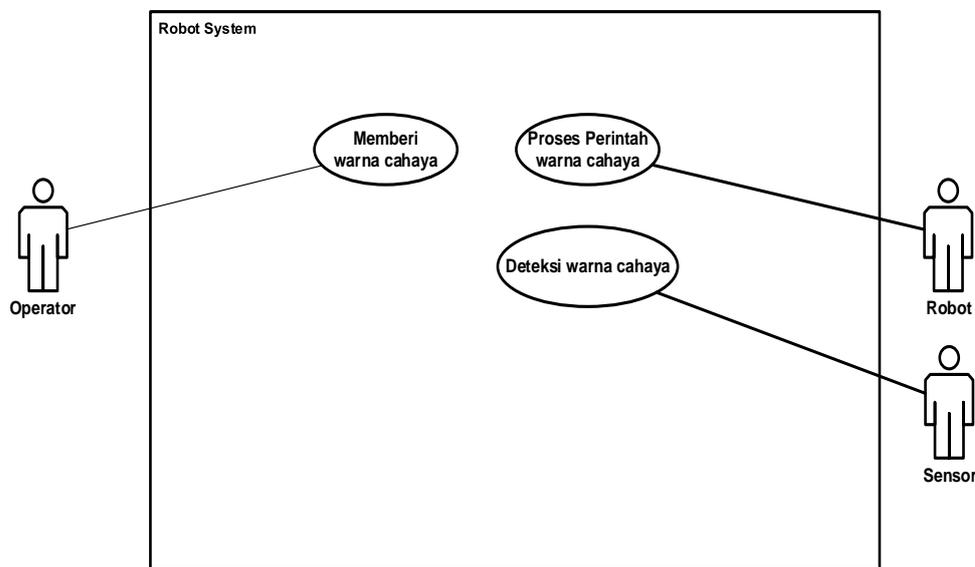
5. Papan Arduino Mega 2560

Papan Arduino Mega 2560 yang dipakai pada robot *Color light follower* ini berfungsi sebagai pusat pengendali dari robot *color light follower*. Papan Arduino Mega 2560 ini menggunakan mikrokontroler Atmega2560. Papan Arduino Mega 2560 ini

berfungsi sebagai pengolah data yang diterima oleh sensor LDR dan sensor warna TCS3200 yang kemudian menjadi sebuah keluaran untuk mengendalikan *Driver* motor L298N dan *Driver* Motor L298N akan mengendalikan motor DC yang digunakan sebagai roda robot.

3.4 Use case Diagram

Adapun *use case* diagram dalam pembuatan sistem robot pengikut cahaya yaitu :



Gambar 3. 2
Use case diagram Robot Color Light Follower

Nama *use case* : Memberi Warna Cahaya

Aktor : Operator

Deskripsi : Operator memberikan warna cahaya ke robot

Pre-condition : Warna cahaya belum disorotkan ke robot

Post-condition : Robot berhasil menangkap warna cahaya

Skenario :

Tabel 3. 1
Skenario Normal diagram *use case* memberi warna cahaya

Aksi Aktor	Aksi Sistem
Skenario Normal	
1. Mengarahkan warna cahaya ke robot	
	2. Menerima warna cahaya yang didapat

Skenario Alternatif

Nama *use case* : Memberi Warna Cahaya

Aktor : Operator

Deskripsi : Warna cahaya yang diberikan operator ke robot kurang terang

Pre-condition : Robot gagal menangkap cahaya dikarenakan warna cahaya kurang terang

Post-condition : Robot berhasil menangkap warna cahaya

Skenario :

Tabel 3. 2

Skenario Alternatif diagram *use case* memberi warna cahaya

Aksi Aktor	Aksi Sistem
Skenario alternatif	
1. Mengarahkan warna cahaya ke robot	
	2. Warna cahaya tidak diterima oleh sistem
3. Mengarahkan warna cahaya yang lebih terang	
	4. Menerima warna cahaya yang didapat.

Nama *use case* : Memproses Perintah Warna Cahaya

Aktor : Robot

Deskripsi : Warna cahaya yang didapat dari operator diproses oleh sistem

Pre-condition : Warna cahaya yang diterima terdaftar pada sistem

Post-condition : warna cahaya berhasil diproses dan robot berjalan sesuai dengan arah yang diberikan

Skenario :

Tabel 3. 3

Skenario Normal diagram *use case* memproses perintah warna cahaya

Aksi Aktor	Aksi Sistem
Skenario Normal	
	1. Menerima data warna cahaya
	2. Memvalidasi data warna cahaya
	3. Memberi perintah arah ke robot
4. Robot berjalan sesuai dengan perintah warna cahaya	

Skenario Alternatif

Nama *use case* : Memproses Perintah Warna Cahaya

Aktor : Robot

Deskripsi : Warna cahaya yang didapat dari operator diproses oleh sistem

Pre-condition : Warna cahaya yang diterima sistem tidak terdaftar

Post-condition : warna cahaya berhasil diproses dan robot akan diam

Skenario :

Tabel 3. 4

Skenario Alternatif diagram *use case* memproses perintah warna cahaya

Aksi Aktor	Aksi Sistem
Skenario Alternatif	
	1. Menerima data warna cahaya
	2. Memvalidasi data warna cahaya
	3. Warna cahaya tidak dikenali
	4. Memberi perintah diam ke robot
5. Robot diam	

Nama *use case* : Deteksi Warna Cahaya

Aktor : Sensor

Deskripsi : Sensor menerima warna cahaya yang disorotkan oleh operator untuk diproses

Pre-condition : Warna cahaya sudah disorotkan operator ke robot

Post-condition : warna cahaya berhasil ditangkap dan divalidasi oleh sensor warna

Skenario :

Tabel 3. 5
Skenario Normal diagram *use case* deteksi warna cahaya

Aksi Aktor	Aksi Sistem
Skenario Normal	
	1.Sistem mengaktifkan sensor LDR
2.Deteksi cahaya	
	3. Memvalidasi cahaya
	4. Mengaktifkan sensor warna
5.Deteksi warna cahaya	
	6. Mervalidasi warna cahaya
	7.Data warna cahaya diterima

Skenario Alternatif

Nama *use case* : Deteksi warna cahaya

Aktor : Sensor

Deskripsi : Sensor tidak menerima warna cahaya yang disorotkan oleh operator

Pre-condition : Warna cahaya sudah disorotkan operator ke robot

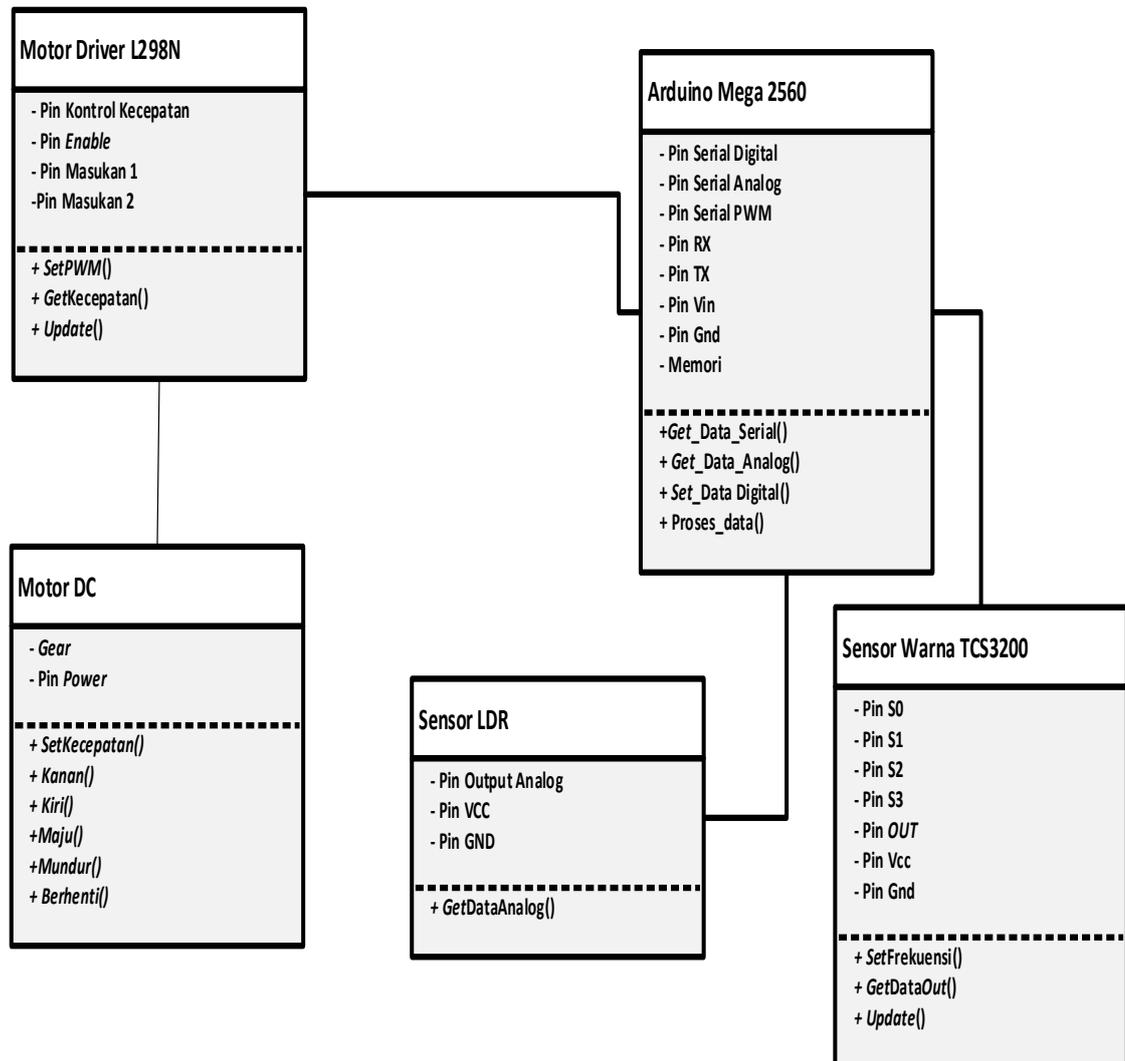
Post-condition : Warna cahaya tidak terdeteksi oleh sensor

Skenario :

Tabel 3. 6
Skenario Alternatif diagram *use case* Deteksi warna cahaya

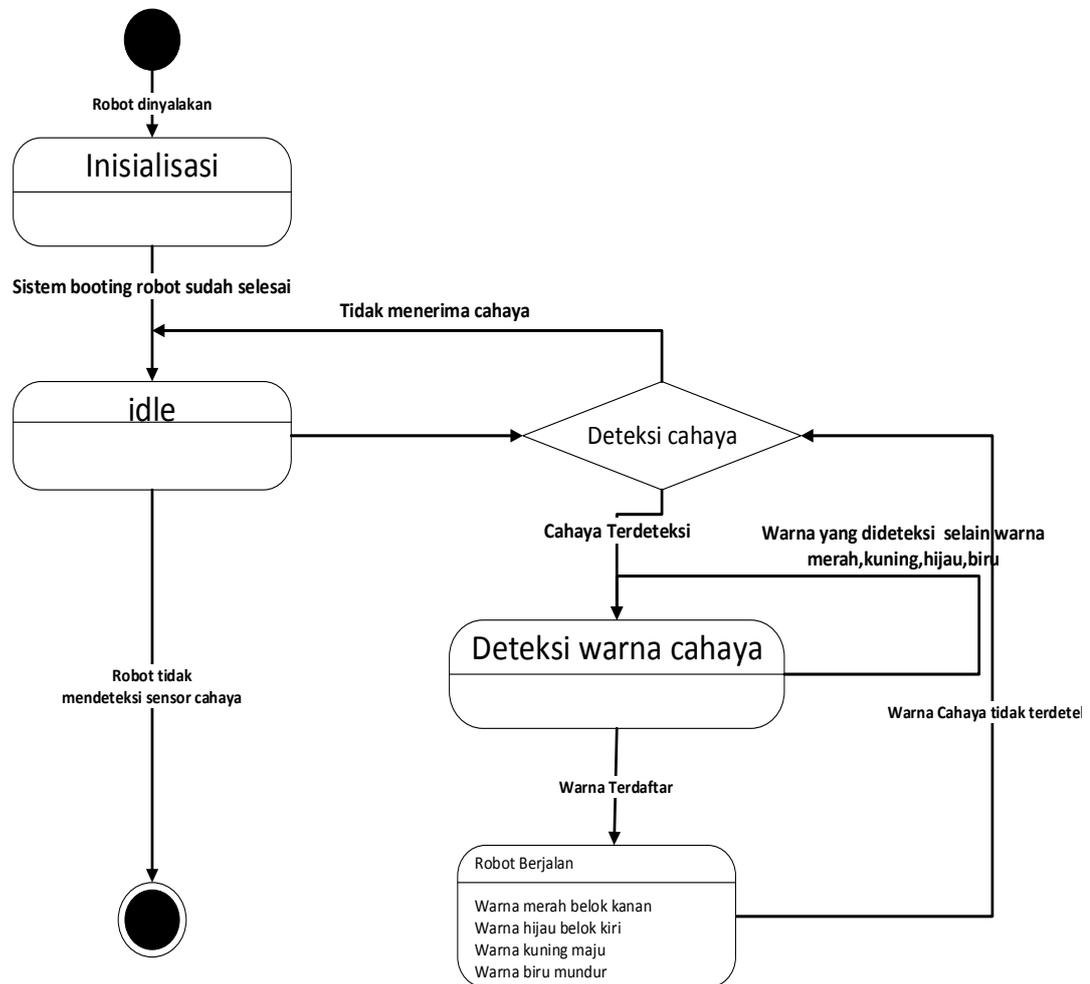
Aksi Aktor	Aksi Sistem
Skenario Alternatif	
	1.Sistem mengaktifkan sensor LDR
2.Deteksi cahaya	
	3.Data cahaya tidak terdeteksi
	4.Data cahaya bernilai 0

3.5 Class Diagram



Gambar 3. 3
class diagram Robot Color Light Follower

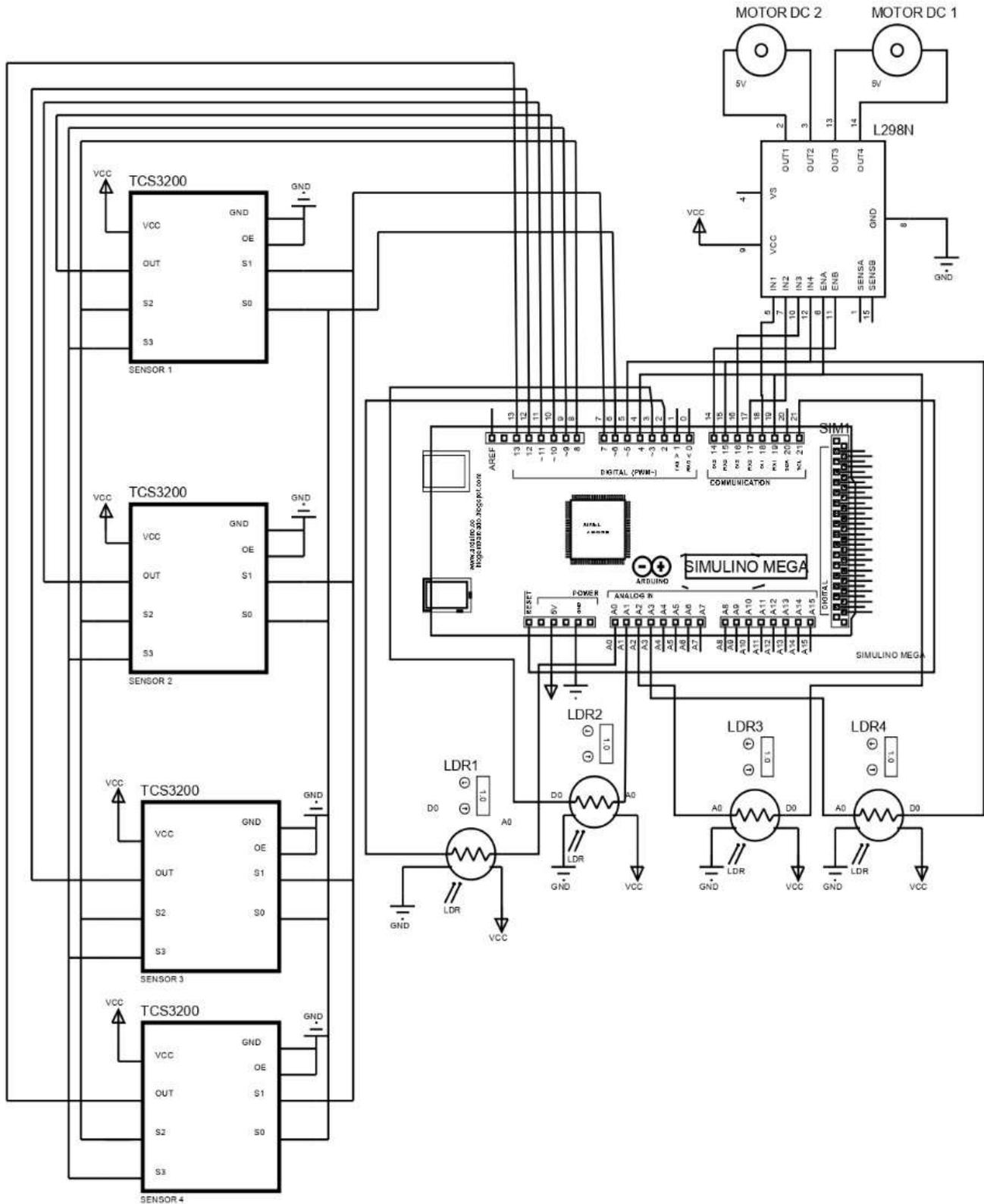
3.6 State Machine Diagram



Gambar 3. 4

State Machine Diagram Robot Color Light Follower

3.7 Skematik Robot Color Light Follower



Gambar 3. 5
Skematik Robot Color Light Follower

BAB IV

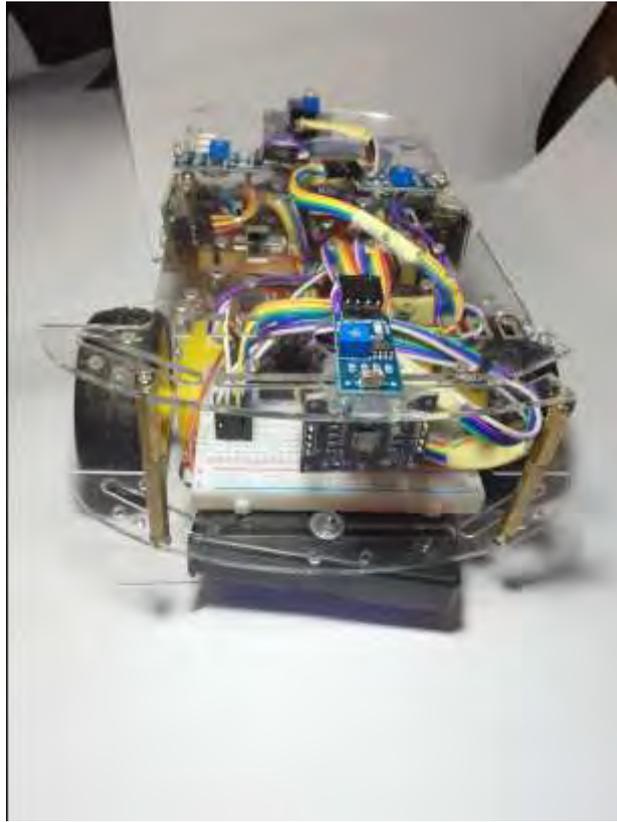
IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi Perangkat Keras

Hasil Akhir dari perncangan robot *color light follower* seperti pada gambar 4.1, gambar 4.2 dan gambar 4.3 berbentuk seperti sebuah mobil dengan menggunakan tiga buah roda yang dimana dua roda di belakang bersifat aktif dan satu roda bersifat pasif atau roda bebas. Pada dua roda belakang yang bersifat aktif memiliki sumber gerak dari motor DC dan gerakan dari motor DC dikendalikan oleh *driver* motor L298N.

Robot *color light follower* akan bekerja jika ada sebuah warna cahaya yang diberikan ke arah robot dan warna cahaya tersebut divalidasi oleh robot untuk menggerakkan robot sesuai dengan perintah warna cahayanya. Warna cahaya yang bisa diberikan hanya empat warna yaitu warna merah, warna kuning, warna hijau, dan warna biru. Masing-masing warna cahaya memiliki arah yang berbeda untuk menggerakkan sebuah robot *color light follower*. Warna cahaya merah menggerakkan robot untuk belok kanan, warna kuning menggerakkan robot maju, warna hijau menggerakkan robot untuk belok ke kiri dan warna biru untuk menggerakkan robot mundur.

Cara robot *color light follower* adalah robot akan mendeteksi sebuah cahaya yang disorotkan ke robot *color light follower* dengan bantuan sensor LDR (*Light Dependent Sensor*), jika cahaya yang didapatkan sesuai dengan batasan yang diberikan maka robot akan kembali memvalidasi warna cahaya yang diberikan menggunakan sensor warna TCS3200, jika warna yang dideteksi dikenali maka robot akan bergerak sesuai dengan perintah warna cahaya yang diberikan dan jika warna cahaya yang diberikan belum terdaftar maka robot tidak akan bergerak atau berhenti.



Gambar 4. 1
Robot *Color Light Follower* Tampak Depan



Gambar 4. 2
Robot *Color Light Follower* Tampak Samping



Gambar 4. 3
Robot *Color Light Follower* Tampak Atas

4.2 Pengujian Modul dan Analisa

Pengujian dan Analisa yang dilakukan pada sub bab ini bertujuan untuk menguji setiap modul yang dipakai untuk merancang robot ini. Pengujian ini juga bertujuan untuk menganalisa kinerja robot, kesalahan, keakuratan, dan keterbatasan setiap modul dan sistem yang dipakai untuk merancang robot *color light follower*. Hasil dari pengujian ini akan dipakai untuk menarik kesimpulan serta menjadi acuan dalam menjawab rumusan masalah yang telah dibuat.

4.2.1 Pengujian Motor DC (*Direct Current*) dan Motor *Driver* L298N

Hasil Pengujian Motor DC yang dikendalikan oleh *Driver* Motor L298N pada Tabel 4.1. Nilai *Enable* (EN) memiliki nilai yakni *LOW* dan *HIGH*. Pin *Enable* (EN) ini berfungsi untuk mengatur kecepatan pada Motor DC, jika pin *Enable* diberikan *LOW* maka Motor DC akan berhenti atau tidak berputar dan jika pin *Enable* diberikan *HIGH* maka Motor DC akan berputar dengan kecepatan PWM maksimal.

Nilai *IN1* (*input 1*) dan *IN2* (*input 2*) adalah pin *input* yang berfungsi untuk menentukan arah perputaran motor DC, nilai *input* ini masing-masing memiliki nilai yang bernilai *HIGH* dan *LOW*. Arah putaran pada Motor DC bergantung pada arus yang diberikan oleh *Driver Motor*, arus yang diberikan dapat mengarah dari positif ke negatif atau dari negatif ke positif. Arah arus yang diberikan akan berpengaruh pada arah putaran pada Motor DC, arah putar Motor DC dapat searah jarum jam atau sebaliknya. Dan jika tegangan yang diberikan pada *IN1* dan *IN2* sama, maka motor DC tidak akan berputar.

Tabel 4. 1
Tabel hasil pengujian Motor DC

<i>Enable (EN)</i>	<i>IN1</i>	<i>IN2</i>	Keadaan Motor	Hasil Pengujian
LOW	0	0	Tidak berputar	Sesuai
	0	1	Tidak berputar	Sesuai
	1	0	Tidak berputar	Sesuai
	1	1	Tidak berputar	Sesuai
HIGH	0	0	Tidak berputar	Sesuai
	0	1	Berputar searah jarum jam	Sesuai
	1	0	Berputar berlawanan arah jarum jam	Sesuai
	1	1	Tidak berputar	Sesuai

Pada Tabel 4.2 bertujuan untuk mengendalikan arah robot berdasarkan putaran roda. Robot dapat melakukan maju, mundur, belok kanan, dan belok kiri dengan membalikkan arah arus pada *driver* motor dan juga dapat memberhentikan salah satu roda pada saat robot berbelok.

Tabel 4. 2
Tabel Hasil Pengujian Arah Putaran *Driver* Motor

EN_A	EN_B	IN1	IN2	IN3	IN4	Keterangan	Hasil Pengujian
1	1	1	0	0	1	Maju	Sesuai
1	1	0	1	1	0	Mundur	Sesuai
1	1	1	0	0	0	Kanan	Sesuai
1	1	0	0	0	1	Kiri	Sesuai
0	0	0	0	0	0	Diam/berhenti	Sesuai
1	1	1	1	1	1	Diam/berhenti	Sesuai

4.2.2 Pengujian Sensor LDR (*Light Dependent Resistor*) dan Sensor Warna TCS3200 Pada Robot

Pengujian pada tabel 4.3, sensor LDR pada robot ini bertujuan untuk mengetahui seberapa akurat sensor LDR mendeteksi intensitas cahaya yang masuk ketika robot mendapatkan sebuah cahaya yang diberikan. Intensitas cahaya yang dihasilkan menggunakan satuan Kohm (*Kilo Ohm*). Pada pengujian ini, Intensitas cahaya yang dihasilkan oleh sensor LDR robot berkisar 51 Kohm sampai 850 KOhm.

Tabel 4. 3
Tabel Hasil Pengujian Sensor LDR

Sumber Cahaya	Luas Ruangan	Keluaran Intensitas Cahaya Sensor LDR (KOhm)
Cahaya matahari	9 m ²	51
Cahaya lampu LED 13 watt	9 m ²	240
Cahaya lampu LED 8 watt	9 m ²	277
Cahaya lampu neon 40 watt	9 m ²	206
Tidak ada sumber cahaya (gelap)	9 m ²	850

Hasil pengujian pada tabel 4.4 , jika sensor LDR mendeteksi adanya sebuah warna cahaya yang diberikan dan jika cahaya yang diberikan terdeteksi kurang dari 350 KOhm maka robot akan memberikan nilai *true* yang kemudian robot akan kembali memvalidasi warna cahaya menggunakan sensor warna TCS3200. Jika cahaya yang dideteksi oleh sensor LDR lebih dari 350 KOhm maka robot tidak akan meneruskan validasi warna cahaya dan robot akan diam.

Pengujian sensor warna TCS3200 pada robot ini bertujuan untuk mengetahui seberapa akurat sensor warna TCS3200 ini dalam mendeteksi adanya sebuah warna cahaya yang didapat. Hasil keluaran pada sensor warna TCS3200 ini adalah nilai frekuensi dengan satuan hertz (Hz).

Dalam pengujian ini, sensor warna TCS3200 akan mendeteksi sebuah warna cahaya dan mengeluarkan hasil berupa frekuensi dengan masing - masing filter sensor, dan frekuensi tersebut yang akan diproses untuk dibandingkan sesuai dengan masing - masing filter sensor untuk mendapatkan kesimpulan warna yang didapat.

Tabel 4. 4
Tabel Hasil Pengujian Sensor LDR dan Sensor Warna TCS3200

Sensor LDR dan Sensor Warna TCS3200	Keluaran Intensitas Cahaya Sensor LDR (KOhm)	Masukan warna cahaya pada sensor warna TCS3200	Hasil Pengujian	Kesimpulan
Sensor LDR1 dan Sensor TCS3200_1	108	Merah	Robot berbelok ke kanan	Sesuai
	137	Kuning	Robot bergerak maju	Sesuai
	111	Hijau	Robot berbelok ke kiri	Sesuai
	126	Biru	Robot bergerak mundur	Sesuai
	130	Cyan	Robot bergerak mundur	Tidak Sesuai
	115	Magenta	Robot berbelok ke kanan	Tidak Sesuai
Sensor LDR2 dan Sensor TCS3200_2	113	Merah	Robot berbelok ke kanan	Sesuai
	138	Kuning	Robot bergerak maju	Sesuai
	122	Hijau	Robot berbelok ke kiri	Sesuai
	130	Biru	Robot bergerak mundur	Sesuai
	134	Cyan	Robot bergerak mundur	Tidak Sesuai
	105	Magenta	Robot berbelok ke kanan	Tidak Sesuai
Sensor LDR3 dan Sensor TCS3200_3	100	Merah	Robot berbelok ke kanan	Sesuai
	136	Kuning	Robot bergerak maju	Sesuai
	134	Hijau	Robot berbelok ke kiri	Sesuai
	131	Biru	Robot bergerak mundur	Sesuai
	125	Cyan	Robot bergerak mundur	Tidak Sesuai
	104	Magenta	Robot berbelok ke kanan	Tidak Sesuai
Sensor LDR4 dan Sensor TCS3200_4	105	Merah	Robot berbelok ke kanan	Sesuai
	138	Kuning	Robot bergerak maju	Sesuai
	109	Hijau	Robot berbelok ke kiri	Sesuai
	140	Biru	Robot bergerak mundur	Sesuai
	127	Cyan	Robot bergerak mundur	Tidak Sesuai
	101	Magenta	Robot berbelok ke kanan	Tidak Sesuai

4.3 Pengujian Sistem

Pengujian sistem ini bertujuan untuk melihat sejauh mana robot dapat mendeteksi adanya sebuah warna cahaya yang diberikan, sehingga robot dapat memutuskan untuk berjalan sesuai dengan arahan cahaya yang diberikan.

Hasil dari pengujian ini dapat dilihat pada tabel 4.5, dalam pengujian ini menjelaskan bahwa terdapat empat pasang sensor yaitu empat sensor LDR dan sensor warna TCS3200 sebanyak empat buah dimana sensor 1, sensor 2, sensor 3, dan sensor 4 mewakili sensor LDR dan sensor warna diletakkan berpasangan saling berdekatan. Pada tabel 4.5, sensor 1 mewakili sensor LDR dan sensor warna bagian depan, sensor 2 mewakili sensor LDR dan sensor warna bagian belakang, sensor 3 mewakili sensor LDR dan sensor warna bagian kanan, dan sensor 4 mewakili sensor LDR dan sensor warna bagian kiri. Sensor LDR dan sensor warna TCS3200 dapat bernilai 0 atau 1. Dalam nilai tersebut menjelaskan bahwa nilai 0 adalah kondisi jika sensor LDR dan sensor warna TCS3200 tidak mendeteksi sebuah warna cahaya, dan nilai 1 jika sensor LDR dan sensor warna TCS3200 mendeteksi adanya sebuah warna cahaya. Robot akan berjalan jika salah satu sensor LDR terdeteksi cahaya dan sensor warna yang dipasangkan oleh sensor LDR tersebut juga mengenali warna yang diberikan. Jika sensor LDR tidak mendeteksi adanya cahaya maka robot tidak akan bergerak atau diam.

Hasil pengujian pada tabel 4.5, menghasilkan presentase keberhasilan sebesar (70.96%), dan mengalami kegagalan sebesar (29.03%). Kegagalan yang terjadi dikarenakan ketika robot bergerak, warna cahaya biru memiliki intensitas cahaya yang lebih kecil dibandingkan warna merah, kuning, dan hijau, dan faktor lain ketika robot menemukan intensitas cahaya yang terlalu tinggi yang menyebabkan robot menjadi kebingungan untuk mendeteksi warna cahaya yang berasal dari senter. Dan kegagalan yang dialami dikarenakan intensitas yang terlalu tinggi menyebabkan robot mendeteksi adanya warna yang mendominasi warna merah yang menjadikan robot tersebut menjadi bergerak belok kanan terus menerus. Kegagalan yang terjadi juga dikarenakan oleh komposisi pada warna cahaya cyan mendekati warna cahaya biru pada senter yang

menyebabkan robot mendeteksi warna biru dan komposisi pada warna cahaya magenta mendekati warna cahaya merah yang menyebabkan robot berbelok ke kanan.

Tabel 4. 5
Tabel Hasil Pengujian Sistem Robot

Percobaan	Status cahaya LDR dan Sensor Warna TCS3200				Warna yang diuji	Hasil Pengujian	Kesimpulan
	Sensor 1	Sensor 2	Sensor 3	Sensor 4			
1	0	0	0	0	Merah	Diam	BERHASIL
2	1	0	0	0		Belok kanan	BERHASIL
3	0	1	0	0		Belok kanan	BERHASIL
4	0	0	1	0		Belok kanan	BERHASIL
5	0	0	0	1		Belok kanan	BERHASIL
6	0	0	0	0	Kuning	Diam	BERHASIL
7	1	0	0	0		Maju	BERHASIL
8	0	1	0	0		Maju	BERHASIL
9	0	0	1	0		Maju	BERHASIL
10	0	0	0	1		Belok kanan	GAGAL
11	0	0	0	1	Maju	BERHASIL	
12	0	0	0	0	Hijau	Diam	BERHASIL
13	1	0	0	0		Belok kiri	BERHASIL
14	0	1	0	0		Belok kiri	BERHASIL
15	0	0	1	0		Belok kiri	BERHASIL
16	0	0	0	1	Belok kiri	BERHASIL	
17	0	0	0	0	Biru	Diam	BERHASIL
18	1	0	0	0		Mundur	BERHASIL
19	0	1	0	0		Mundur	BERHASIL
20	0	0	1	0		Mundur	BERHASIL
21	0	0	0	1	Mundur	BERHASIL	

Percobaan	Status cahaya LDR dan Sensor Warna TCS3200				Warna yang diuji	Hasil Pengujian	Kesimpulan
	Sensor 1	Sensor 2	Sensor 3	Sensor 4			
22	0	0	0	0	Cyan	Diam	BERHASIL
23	1	0	0	0		Mundur	GAGAL
24	0	1	0	0		Mundur	GAGAL
25	0	0	1	0		Mundur	GAGAL
26	0	0	0	1		Mundur	GAGAL
27	0	0	0	0	Magenta	Diam	BERHASIL
28	1	0	0	0		Belok kanan	GAGAL
39	0	1	0	0		Belok kanan	GAGAL
30	0	0	1	0		Belok kanan	GAGAL
31	0	0	0	1		Belok kanan	GAGAL
Presentasi Keberhasilan							70.96%
Presentasi Kegagalan							29.03%

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan perancangan yang telah disusun, serta melakukan beberapa rangkaian pengujian, maka dapat diperoleh beberapa kesimpulan, yaitu :

1. Robot *color light follower* dapat mengikuti cahaya sesuai arah yang ditentukan dengan cara lampu senter yang berwarna merah, kuning, biru atau hijau diarahkan ke arah robot dan robot akan mengikuti sesuai dengan arah yang ditentukan, diantaranya warna merah robot akan berbelok ke kanan, warna hijau berbelok ke kiri, warna kuning bergerak maju, dan warna biru bergerak mundur .
2. Sensor dapat mendeteksi adanya cahaya dengan bantuan sensor LDR dengan mengukur intensitas cahaya yang didapat dan sensor dapat mengenali warna cahaya dengan bantuan sensor warna TCS3200 yang dapat mengenali adanya sebuah warna pada cahaya yang disorotkan ke robot.
3. Robot *color light follower* akan terus berjalan jika warna cahaya masih terdeteksi oleh robot dan robot akan diam jika warna cahaya tidak dikenali.
4. Intensitas cahaya ruangan sekitar mempengaruhi pada pembacaan warna cahaya yang berasal dari cahaya senter.

5.2 Saran

Rancangan Robot *color light follower* ini bukanlah suatu robot sempurna dan masih banyak memiliki kekurangan. Robot yang dirancang butuh untuk lebih disempurnakan dan karena beberapa faktor penulis, sehingga penulis memiliki keterbatasan untuk melakukan penambahan, sehingga beberapa saran berikut mungkin akan membantu dalam penyempurnaan robot *color light follower* ini, dan berikut ini adalah saran yang diberikan penulis :

1. Sensor warna yang digunakan sebaiknya menggunakan sensor warna yang dirancang khusus untuk menangkap warna cahaya, sehingga kepekaan dan keakuratan pembacaan sensor warna cahaya akan lebih baik dan akurat.
2. Sumber warna cahaya dibuat intensitasnya lebih tinggi sehingga sensor dapat menangkap data warna cahaya dengan lebih akurat.
3. Perancangan kerangka robot dapat dibuat lebih rapih lagi, sehingga sensor LDR dan sensor warna dapat diletakkan di tempat yang strategis untuk menangkap warna cahaya, sehingga pembacaan data warna cahaya dapat terbaca dengan akurat dan cepat.

DAFTAR PUSTAKA

- Anditya, 2016, "*Buku Pintar Desain Arsitektur*", ANDI.
- Andrianto, Aan, Heri & Darmawan, 2016, "*ARDUINO Belajar cepat dan pemograman*", INFORMATIKA
- Andrianto, Heri, 2015, "*Pemograman Mikrokontroler AVR Atmega16 menggunakan Bahasa C*", INFORMATIKA.
- Andrianto & Darmawan, 2016,"*Arduino Belajar Cepat dan Pemograman*", INFORMATIKA.
- Barret, Steven F., 2013, "Arduino Microcontroller Processing for Everyone!: Thrid Edition", Morgan & Claypool.
- Budiharto, Widodo, 2014, "*Perancangan dan Pemograman Hasta Karya Robot*", ANDI.
- Budiharto, Widodo, 2014, "*ROBOTIKA MODERN*", ANDI.
- Caroline, dkk., 2014, "*Pengaruh Cahaya Yang Diterima Sensor LDR (Light Dependent Resistor) Pada Robot Pengikut Cahaya*", Universitas Sriwijaya.
- Dharmawan, Hari, 2017, "*MIKROKONTROLER Konsep Dasar dan Praktis*", UB Press.
- Fachruddin, Andi, 2019,"*Journalism Today*", Kencana.
- Faridah, Nur, 2018, "*Mengenal Lebih Dekat dengan Cahaya dan Warna*", LeutikaPrio.
- Hendratman, Hendi, 2017, "*Computer Graphic Design*", INFORMATIKA.
- ISO (8373:2012),2012,"*International Organization for Standardzation*", Robots and Robotic Devices.
- Kadir, Abdul, 2018, "*Arduino dan Sensor*", ANDI.
- Kadir, Abdul, 2019, "*Dasar Pemograman Robot Menggunakan Arduino*", ANDI.
- McRoberts, Michael, 2010, "*Begining Arduino*", Technology In Action.
- Munawar, 2018 "*Analisis Perancangan Sistem Berorientasi Objek dengan UML*", INFORMATIKA.
- Nugroho, Adi, 2010, "*Rekayasa Perangkat Lunak Berorientasi Objek dengan Metode USDP*", ANDI
- Nugroho, Sarwo, 2015, "*Manajemen Warna dan Desain*", ANDI.
- Pressman, Roger, 2010, "*Rekayasa Perangkat Lunak* ", ANDI
- Rosa & Shalahuddin, 2010, "*Modul Pembelajaran Pemograman Berorientasi Objek dengan Bahasa Pemograman C++, PHP, dan Java*", MODULA.
- Rosa & Shalahuddin, 2015, "*Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*", INFORMATIKA.

Rosa & Shalahuddin, 2018, "*Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek Edisi Revisi*", INFORMATIKA.

Roslidar, dkk., 2017, "*Perancangan Robot Light Follower Untuk Kursi Otomatis dengan Menggunakan Mikrokontroler ATmega 328P*", Universitas Syiah Kuala.

Sianipar dkk, 2012, "*Teori dan Implementasi Pemograman Berorientasi Objek menggunakan C++*", ANDI.

Supriyanto Raden, Hustinawati, dkk, 2010, "*Robotika*", PHK-I Universitas Gunadarma.

Syahwill, Muhammad, 2013, "*Panduan Mudah Simulasi dan Praktik Mikrokontroler Arduino*", ANDI.

Wicaksono, Mochamad, 2019, "*APLIKASI ARDUINO dan SENSOR*", INFORMATIKA.

https://www.academia.edu/35106798/IMPLEMENTASI_MODEL_ROBOT_EDUKASI_MENGGUNAKAN_MIKROKONTROLER_ATMEGA8_UNTUK_ROBOT_PEMADAM_API, diakses tanggal 19 september 2019 jam 13.13 WIB

<http://scdc.binus.ac.id/himtekn/2017/05/08/motor-dc-dan-jenis-jenisnya/>, diakses tanggal 17 November 2019 jam 13.03

http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_Summary.pdf, diakses 20 Februari 2020 jam 09.30

DAFTAR LAMPIRAN

Listing Program Mikrokontroler Robot Color Light Follower :

```
#include <tcs3200.h>

int red;

int green;

int blue;

int white;

//deklarasi pin sensor warna

tcs3200 tcs1(6, 7, 8, 9, 10); // (S0, S1, S2, S3, output pin) //
tcs3200 tcs2(6, 7, 8, 9, 11); // (S0, S1, S2, S3, output pin) //
tcs3200 tcs3(6, 7, 8, 9, 12); // (S0, S1, S2, S3, output pin) //
tcs3200 tcs4(6, 7, 8, 9, 13); // (S0, S1, S2, S3, output pin) //

#define MERAH 1
#define BIRU 2
#define HIJAU 3
#define KUNING 4
#define PUTIH 5

#define KUAT_CAHAYA 350 // di sesuaikan dengan cahaya dr led

//deklarasi pinout L298

int L298_EN1 = 19, L298_A1 = 18, L298_A2 = 17;
int L298_EN2 = 14, L298_B1 = 15, L298_B2 = 16;

int LDR1 = 2;
int LDR2 = 3;
int LDR3 = 4;
int LDR4 = 5;

int AN_LDR1 = A0;
int AN_LDR2 = A1;
int AN_LDR3 = A2;
int AN_LDR4 = A3;

int cahaya = 0;

int sensor_ldr_depan = 0
```

```
int sensor_ldr_kiri = 0;
int sensor_ldr_kanan = 0;
int sensor_ldr_belakang = 0;
int res=21;
bool statldr1=false;
bool statldr2=false;
bool statldr3=false;
bool statldr4=false;
void setup() {
    Serial.begin(9600);
    digitalWrite(res,HIGH);
    pinMode(L298_EN1,OUTPUT);
    pinMode(L298_EN2,OUTPUT);
    pinMode(L298_A1,OUTPUT);
    pinMode(L298_A2,OUTPUT);
    pinMode(L298_B1,OUTPUT);
    pinMode(L298_B2,OUTPUT);
    digitalWrite(L298_EN1,HIGH);
    digitalWrite(L298_EN2,HIGH);
    pinMode(LDR1,INPUT_PULLUP);
    pinMode(LDR2,INPUT_PULLUP);
    pinMode(LDR3,INPUT_PULLUP);
    pinMode(LDR4,INPUT_PULLUP);
    digitalWrite(LDR1,HIGH);
    digitalWrite(LDR2,HIGH);
    digitalWrite(LDR3,HIGH);
    digitalWrite(LDR4,HIGH);
    robot_stop();
}
void loop() {
    robot_stop();
    digitalWrite(LDR1,HIGH);
    digitalWrite(LDR2,HIGH);
```

```
digitalWrite(LDR3,HIGH);
digitalWrite(LDR4,HIGH);
statldr1=false;
statldr2=false;
statldr3=false;
statldr4=false;
if(digitalRead(LDR1)==0){
digitalWrite(LDR2,LOW);
digitalWrite(LDR3,LOW);
digitalWrite(LDR4,LOW);
statldr1=true;
statldr2=false;
statldr3=false;
statldr4=false;
while(digitalRead(LDR1)==0 && statldr1==true){
    cahaya = analogRead(AN_LDR1);
    Serial.println(cahaya);
    if(cahaya<=KUAT_CAHAYA){
        int warna = getWarna(1);
        if(warna==MERAH){
            Serial.println("Merah");
            robot_belok_kanan();
            digitalWrite(res,LOW);
        }
        else if(warna==BIRU){
            Serial.println("Biru");
            robot_mundur();
            digitalWrite(res,LOW);
        }
        else if(warna==HIJAU){
            Serial.println("Hijau");
            robot_belok_kiri();
            digitalWrite(res,LOW);
        }
    }
}
```

```
else if(warna==KUNING){
    Serial.println("Kuning");
    robot_maju();
    digitalWrite(res,LOW);
}
else robot_stop();
digitalWrite(res,LOW);

}
}
statldr1=false;
}
else if(digitalRead(LDR2)==0){
    digitalWrite(LDR1,LOW);
    digitalWrite(LDR3,LOW);
    digitalWrite(LDR4,LOW);
    statldr1=false;
    statldr2=true;
    statldr3=false;
    statldr4=false;
    while(digitalRead(LDR2)==0 &&statldr2==true){
        cahaya = analogRead(AN_LDR2);
        Serial.println(cahaya);
        if(cahaya<=KUAT_CAHAYA){

            int warna = getWarna(2);
            if(warna==MERAH){
                Serial.println("Merah");
                robot_belok_kanan();
                digitalWrite(res,LOW);
            }
            else if(warna==BIRU){
                Serial.println("Biru");
                robot_mundur();
```

```
        digitalWrite(res,LOW);
    }
    else if(warna==HIJAU){
        Serial.println("Hijau");
        robot_belok_kiri();
        digitalWrite(res,LOW);
    }
    else if(warna==KUNING){
        Serial.println("Kuning");
        robot_maju();
        digitalWrite(res,LOW);
    }
    else robot_stop();
    digitalWrite(res,LOW);
}
}
statldr2=false;
}
else if(digitalRead(LDR3)==0){
    digitalWrite(LDR1,LOW);
    digitalWrite(LDR2,LOW);

    digitalWrite(LDR4,LOW);
    statldr1=false;
    statldr2=false;
    statldr3=true;
    statldr4=false;
    while(digitalRead(LDR3)==0 && statldr3==true){
        cahaya = analogRead(AN_LDR3);
        Serial.println(cahaya);
        if(cahaya<=KUAT_CAHAYA){
            int warna = getWarna(3);
            if(warna==MERAH){
                Serial.println("Merah");
```

```
        robot_belok_kanan();
        digitalWrite(res,LOW);
    }
    else if(warna==BIRU){
        Serial.println("Biru");
        robot_mundur();
        digitalWrite(res,LOW);
    }
    else if(warna==HIJAU){
        Serial.println("Hijau");
        robot_belok_kiri();
        digitalWrite(res,LOW);
    }
    else if(warna==KUNING){
        Serial.println("Kuning");
        robot_maju();
        digitalWrite(res,LOW);
    }
    else robot_stop();
    digitalWrite(res,LOW);
}
}
statldr3=false;
}
else if(digitalRead(LDR4)==0){
    digitalWrite(LDR1,LOW);
    digitalWrite(LDR2,LOW);
    digitalWrite(LDR3,LOW);

    statldr1=false;
    statldr2=false;
    statldr3=false;
    statldr4=true;
    while(digitalRead(LDR4)==0 && statldr4==true){
```

```
cahaya = analogRead(AN_LDR4);
Serial.println(cahaya);
  if(cahaya<=KUAT_CAHAYA){

    int warna = getWarna(4);
    if(warna==MERAH){
      Serial.println("Merah");
      robot_belok_kanan();
      digitalWrite(res,LOW);
    }
    else if(warna==BIRU){
      Serial.println("Biru");
      robot_mundur();
      digitalWrite(res,LOW);
    }
    else if(warna==HIJAU){
      Serial.println("Hijau");
      robot_belok_kiri();
      digitalWrite(res,LOW);
    }
    else if(warna==KUNING){
      Serial.println("Kuning");
      robot_maju();
      digitalWrite(res,LOW);
    }
    else robot_stop();
      digitalWrite(res,LOW);
  }
}
statldr4=false;
}
else robot_stop();
robot_stop();
```

```
}

void robot_maju(){
    digitalWrite(L298_A2,LOW);
    digitalWrite(L298_B1,LOW);
    digitalWrite(L298_A1,HIGH);
    digitalWrite(L298_B2,HIGH);
    digitalWrite(L298_EN1,HIGH);
    digitalWrite(L298_EN2,HIGH);
}

void robot_mundur(){
    digitalWrite(L298_A1,LOW);
    digitalWrite(L298_B2,LOW);
    digitalWrite(L298_A2,HIGH);
    digitalWrite(L298_B1,HIGH);
    digitalWrite(L298_EN1,HIGH);
    digitalWrite(L298_EN2,HIGH);
}

void robot_belok_kiri(){
    digitalWrite(L298_B2,HIGH);
    digitalWrite(L298_B1,LOW);
    digitalWrite(L298_EN1,LOW);
    digitalWrite(L298_EN2,HIGH);
}

void robot_belok_kanan(){
    digitalWrite(L298_A1,HIGH);
    digitalWrite(L298_A2,LOW);
    digitalWrite(L298_EN1,HIGH);
    digitalWrite(L298_EN2,LOW);
}

void robot_stop(){
```

```

digitalWrite(L298_A2,LOW);
digitalWrite(L298_B2,LOW);
digitalWrite(L298_A1,LOW);
digitalWrite(L298_B1,LOW);
digitalWrite(L298_EN1,LOW);
digitalWrite(L298_EN2,LOW);
}

int Red,Blue,Green,White;

int getWarna(int sensor){
    Red = 0;
    Green = 0;
    Blue = 0;
    White = 0;
    if(sensor==1 &&statldr1==true){
        for(int i = 0; i<10;i++){
            Red += tcs1.colorRead('r'); //reads color value for red
            Green += tcs1.colorRead('g'); //reads color value for green
            Blue += tcs1.colorRead('b'); //reads color value for blue
            White += tcs1.colorRead('c'); //reads color value for white(clear)
        }
    }
    else if(sensor==2 &&statldr2==true){
        for(int i = 0; i<10;i++){
            Red += tcs2.colorRead('r'); //reads color value for red
            Green += tcs2.colorRead('g'); //reads color value for green
            Blue += tcs2.colorRead('b'); //reads color value for blue
            White += tcs2.colorRead('c'); //reads color value for white(clear)
        }
    }
    else if(sensor==3 &&statldr3==true){
        for(int i = 0; i<10;i++){
            Red += tcs3.colorRead('r'); //reads color value for red

```

```

    Green += tcs3.colorRead('g'); //reads color value for green
    Blue  += tcs3.colorRead('b'); //reads color value for blue
    White += tcs3.colorRead('c'); //reads color value for white(clear)
  }
}
else if(sensor==4 &&statldr4==true){
  for(int i = 0; i<10;i++){
    Red  += tcs4.colorRead('r'); //reads color value for red
    Green += tcs4.colorRead('g'); //reads color value for green
    Blue  += tcs4.colorRead('b'); //reads color value for blue
    White += tcs4.colorRead('c'); //reads color value for white(clear)
  }
}
red  = Red/10;
green = Green/10;
blue  = Blue/10;
white = White/10;

if((red>10||blue>10||green>10)){
  if(red>blue && red>green && green<blue&&red>10){
    return MERAH;
  }
  else if(blue>red && blue>green&&blue>10&&red<7){
    return BIRU;
  }
  else if(green>blue && green>red&&green>10){
    return HIJAU;
  }
  else if(red>green && green>blue&&red>10){
    return KUNING;
  }
  else return PUTIH;
}
return 0; }

```