

HALAMAN PENGESAHAN

Laporan tugas akhir dengan judul “Rancang Bangun Sistem Pendeteksi Objek Berbasis Laser Pada *Mobile Robot*” Oleh Atzary Luqman M 322 19 028 dan Khalil Abdurrazzaq 322 20 013 dinyatakan layak untuk diujikan.

Makassar, 29 September 2023

Pembimbing I



Lidemar Halide, S. T., M. T.

NIP. 19700413 199602 1 001

Pembimbing 2

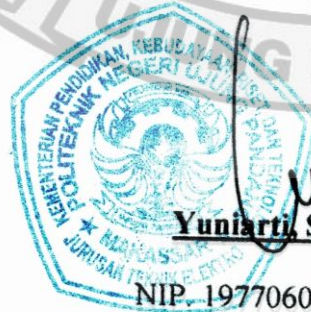


Sahabuddin Abdul Kadir, S. T., M. T.

NIP. 19751130 200604 1 001

Mengetahui

Koordinator Program Studi



Yuniarti, S. ST., M. T.

NIP. 19770603 200212 2 002

HALAMAN PENERIMAAN

Pada hari ini, tanggal 29 September 2023, tim penguji ujian sidang laporan tugas akhir telah menerima hasil ujian sidang laporan tugas akhir oleh mahasiswa Atzary Luqman M 322 19 028 dan Khalil Abdurrazzaq 322 20 013 dengan judul “Rancang Bangun Sistem Pendeteksi Objek Berbasis Laser Pada *Mobile Robot*”.

Makassar, 29 September 2023

Tim Penguji Ujian Sidang Laporan Tugas Akhir :

1. Yedi George Yefri Lely, S. ST., M. T. Ketua 
2. Yuniarti, S. ST., M. T. Sekretaris 
3. Arni Litha, S. T., M. T. Anggota 
4. Mardhiyah Nas, S. T., M. T. Anggota 

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT karena berkat rahmat dan karunia-Nya, penulisan Laporan Tugas Akhir ini yang berjudul “Rancang Bangun Sistem Pendeteksi Objek Berbasis Laser Pada *Mobile Robot*” dapat diselesaikan dengan baik.

Penyusunan Tugas Akhir ini disusun guna memperoleh ijazah diploma III pada Jurusan Teknik Elektro Program Studi Teknik Telekomunikasi Politeknik Negeri Ujung Pandang.

Dalam penyusunan Tugas Akhir ini, tidak sedikit kendala yang penulis hadapi. Walaupun demikian, kendala tersebut dapat penulis hadapi berkat bantuan dari berbagai pihak, baik berupa bantuan moral, material, maupun motivasi dan berbagai pihak sehingga Tugas Akhir ini dapat terselesaikan dengan baik.

Atas segala bantuan tersebut, penulis mengucapkan terima kasih kepada :

1. Kedua orang tua penulis yang senantiasa memberikan semangat dan motivasi yang tiada henti serta membantu dalam dukungan moral dan material kepada penulis sehingga terselesaikannya Laporan Tugas Akhir ini dengan baik.
2. Bapak Ir. Ilyas Mansur, M. T. selaku Direktur Politeknik Negeri Ujung Pandang.
3. Bapak Ahmad Rizal Sultan, S. T., M. T., Ph. D. selaku Ketua Jurusan Teknik Elektro Politeknik Negeri Ujung Pandang.

4. Ibu Yuniarti, S. ST., M. T. selaku Ketua Program Studi Teknik Telekomunikasi Politeknik Negeri Ujung Pandang.
5. Tim pembimbing yakni Bapak Lidemar Halide, S. T., M. T. dan Bapak Sahabuddin Abdul Kadir, S. T., M. T. yang telah mencurahkan waktu dan kesempatannya untuk mengarahkan dan membimbing penulis dalam menyelesaikan Laporan Tugas Akhir.
6. Dosen-dosen yang telah membimbing dan mengajarkan berbagai bidang ilmu selama penulis berada di kampus sejak semester 1 hingga penulis menyelesaikan studi di Politeknik Negeri Ujung Pandang.
7. Seluruh teman-teman mahasiswa(i) yang telah memberikan dukungan terkhusus mahasiswa(i) D-3 Teknik Telekomunikasi angkatan tahun 2019 dan tahun 2020.

Penulis menyadari bahwa laporan tugas akhir ini masih kurang sempurna, sehingga penulis mengharapkan kritik dan saran yang sifatnya membangun untuk perbaikan di masa datang. Semoga tulisan ini bermanfaat bagi para pembaca.

Makassar, 29 September 2023



Penulis

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PENGESAHAN	ii
HALAMAN PENERIMAAN.....	iii
KATA PENGANTAR.....	iv
DAFTAR ISI	vi
DAFTAR TABEL.....	ix
DAFTAR GAMBAR	x
DAFTAR LAMPIRAN	xii
SURAT PERNYATAAN.....	xiii
RINGKASAN.....	xv
BAB I. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Ruang Lingkup Kegiatan	2
1.4 Tujuan Kegiatan	3
1.5 Manfaat Kegiatan	3

BAB II. TINJAUAN PUSTAKA.....	4
2.1 Arduino Uno.....	4
2.2 Laserping.....	10
2.3 <i>Liquid Crystal Display</i> (LCD)	13
2.4 <i>Inter-Integrated Circuit</i> (I2C).....	14
2.5 Motor Servo SG90.....	22
2.6 Motor DC.....	24
2.7 Driver Motor L298N.....	30
BAB III. METODOLOGI PENELITIAN.....	32
3.1 Waktu dan Tempat Penelitian	32
3.2 Alat dan Bahan Penelitian.....	32
3.3 Tahap Penelitian	33
3.4 Pengujian Alat	37
BAB IV. HASIL DAN PEMBAHASAN.....	39
4.1 Hasil Perancangan Alat.....	39
4.2 Hasil dan Analisa Pengujian Alat.....	40
BAB V. PENUTUP.....	43
5.1 Kesimpulan.....	43
5.2 Saran	45

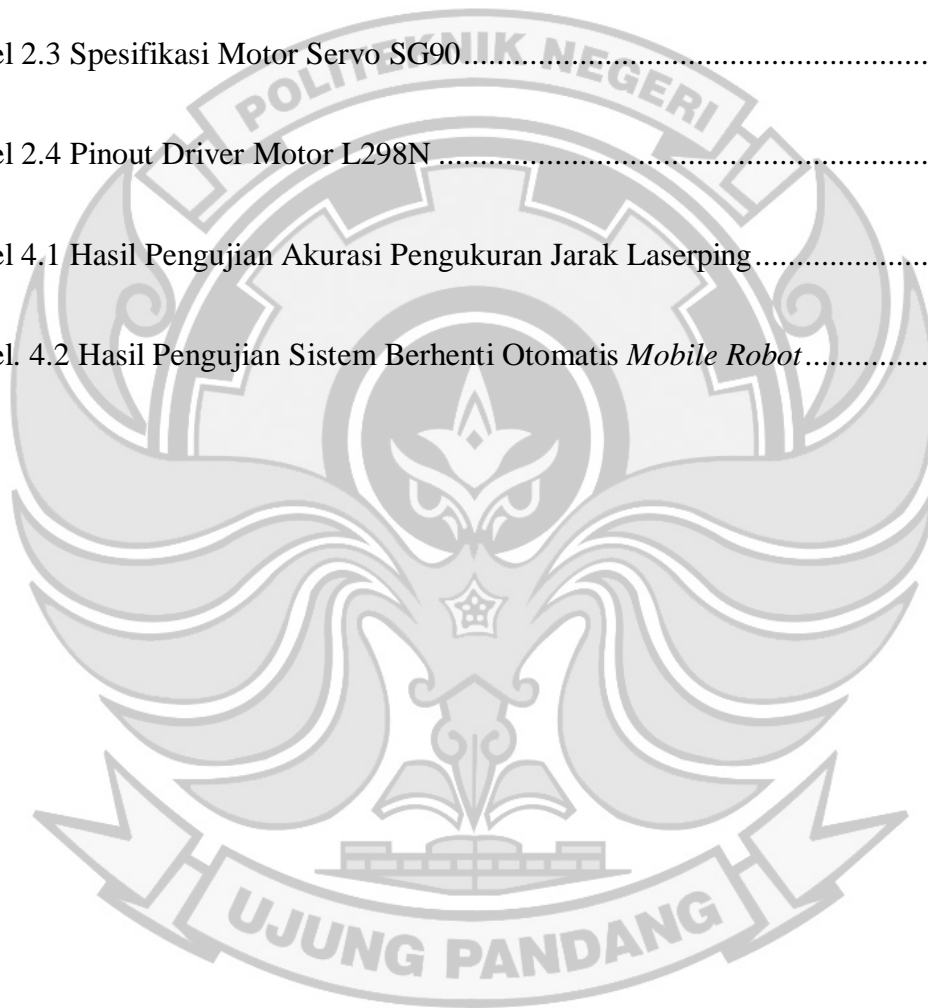
DAFTAR PUSTAKA46

LAMPIRAN47



DAFTAR TABEL

Tabel 2.1 Spesifikasi Laserping.....	12
Tabel 2.2 Pin-Pin LCD.....	13
Tabel 2.3 Spesifikasi Motor Servo SG90.....	23
Tabel 2.4 Pinout Driver Motor L298N.....	31
Tabel 4.1 Hasil Pengujian Akurasi Pengukuran Jarak Laserping.....	40
Tabel. 4.2 Hasil Pengujian Sistem Berhenti Otomatis <i>Mobile Robot</i>	42



DAFTAR GAMBAR

Gambar 2.1 <i>Board</i> Arduino Uno	6
Gambar 2.2 Sensor Laserring.....	12
Gambar 2.3 LCD (<i>Liquid Crystal Display</i>).....	13
Gambar 2.4 I2C (<i>Inter-Integrated Circuit</i>)	15
Gambar 2.5 Pemasangan Resistor <i>Pull-Up</i>	16
Gambar 2.6 Perangkat <i>Master</i> dan <i>Slave</i>	17
Gambar 2.7 <i>Start Sequence</i> dan <i>Stop Sequence</i>	18
Gambar 2.8 Transmisi Data Pada <i>SDA Line</i> dan <i>SCL Line</i>	18
Gambar 2.9 Transmisi Alamat Perangkat <i>Slave</i> Pada <i>Sequence Protocol I2C</i>	19
Gambar 2.10 Motor Servo SG90	22
Gambar 2.11 Motor DC	25
Gambar 2.12 Bagian-Bagian Motor DC.....	25
Gambar 2.13 Prinsip Kerja Motor DC	27
Gambar 2.14 Blok Diagram Kaskade Kontrol Motor DC	28
Gambar 2.15 Pinout Driver Motor L289N.....	31
Gambar 3.1 Diagram Blok Perancangan Sistem	34

Gambar 3.2 Rangkaian Perangkat Keras *Mobile Robot*.....35

Gambar 3.3 Diagram Alir Rancangan Software.....36

Gambar 4.1 *Prototype Mobile Robot*39



DAFTAR LAMPIRAN

Lampiran 1 Program Sistem <i>Mobile Robot</i> menggunakan Arduino IDE	47
Lampiran 2 Pengujian Sistem Berhenti Otomatis Pada <i>Mobile Robot</i>	53
Lampiran 3 Datasheet Arduino Uno	55
Lampiran 4 Datasheet Laserping	57
Lampiran 5 Datasheet <i>Liquid Crystal Display</i> (LCD)	58
Lampiran 6 Datasheet <i>Inter-Intergrated Circuit</i> (I2C)	59
Lampiran 7 Datasheet Motor Servo SG90	60
Lampiran 8 Datasheet Motor DC.....	61
Lampiran 9 Datasheet Driver Motor L298N.....	62

SURAT PERNYATAAN

Saya yang bertandatangan di bawah ini:

Nama : Atzary Luqman M

NIM : 32219028

menyatakan dengan sebenar-benarnya bahwa segala pernyataan dalam laporan tugas akhir ini yang berjudul “Rancang Bangun Sistem Pendeteksi Objek Berbasis Laser Pada *Mobile Robot*” merupakan gagasan hasil karya saya sendiri dengan arahan komisi pembimbing dari belum pernah diajukan dalam bentuk apapun pada perguruan tinggi dan institusi manapun.

Semua data dan informasi yang digunakan telah dinyatakan secara jelas dan dapat diperiksa kebenarannya. Sumber informasi yang berasal atau dikutip dari karya penulis lain telah disebutkan dalam naskah dan dicantumkan dalam laporan tugas akhir ini.

Jika pernyataan saya tersebut di atas tidak benar, saya siap menanggung risiko yang ditetapkan oleh Politeknik Negeri Ujung Pandang.

Makassar, 29 September 2023



Atzary Luqman M

322 19 028

SURAT PERNYATAAN

Saya yang bertandatangan di bawah ini:

Nama : Khalil Abdurrazzaq

NIM : 32220013

menyatakan dengan sebenar-benarnya bahwa segala pernyataan dalam laporan tugas akhir ini yang berjudul “Rancang Bangun Sistem Pendeteksi Objek Berbasis Laser Pada *Mobile Robot*” merupakan gagasan hasil karya saya sendiri dengan arahan komisi pembimbing dari belum pernah diajukan dalam bentuk apapun pada perguruan tinggi dan institusi manapun.

Semua data dan informasi yang digunakan telah dinyatakan secara jelas dan dapat diperiksa kebenarannya. Sumber informasi yang berasal atau dikutip dari karya penulis lain telah disebutkan dalam naskah dan dicantumkan dalam laporan tugas akhir ini.

Jika pernyataan saya tersebut di atas tidak benar, saya siap menanggung risiko yang ditetapkan oleh Politeknik Negeri Ujung Pandang.

Makassar, 29 September 2023



Khalil Abdurrazzaq

322 20 013

RINGKASAN

Teknologi deteksi objek adalah komponen penting dalam pengembangan *mobile robot* untuk meningkatkan navigasi dan interaksi di berbagai lingkungan seperti logistik, perawatan kesehatan, dan eksplorasi. Sensor laserping menawarkan solusi yang lebih baik dibandingkan metode deteksi konvensional seperti kamera visual, sensor ultrasonik, dan radar, yang memiliki keterbatasan terkait kondisi cahaya, jangkauan, dan biaya. Sensor ini mampu mengukur jarak dengan presisi tinggi dalam berbagai kondisi pencahayaan dan memberikan data real-time, menjadikannya ideal untuk respons cepat dan akurat. Penelitian ini berfokus pada pengembangan sistem pendeteksi berbasis laser pada *mobile robot*, dengan mengintegrasikan hardware dan algoritma untuk navigasi dan penghindaran rintangan, diharapkan dapat meningkatkan efisiensi dan akurasi *robot* dalam berbagai industri.

Metodologi penelitian mencakup perancangan dan pengujian sistem pendeteksi objek berbasis sensor laserping pada *mobile robot*. Proses ini melibatkan penggunaan perangkat keras seperti Arduino Uno, sensor laserping, motor DC, servo SG90, driver motor L298N, dan LCD, serta pemrograman menggunakan Arduino IDE. Studi literatur dilakukan untuk memahami spesifikasi alat, diikuti oleh perancangan prototipe dan pengujian kinerja sistem.

Hasil pengujian menunjukkan bahwa sensor laserping akurat hingga 200 cm, namun akurasinya menurun di atas jarak tersebut. Sistem berhenti otomatis juga memerlukan peningkatan karena inersia motor DC menyebabkan *mobile robot* melewati titik berhenti yang diinginkan. Pengujian implementasi menunjukkan bahwa sensor ini bekerja baik pada jarak pendek, namun membutuhkan mekanisme pengereman tambahan untuk meningkatkan presisi berhenti. Meskipun hasilnya positif, perbaikan masih diperlukan untuk menyempurnakan performa keseluruhan *mobile robot*.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Teknologi deteksi objek adalah elemen vital dalam pengembangan *mobile robot*, yang berfungsi untuk meningkatkan interaksi dan navigasi *mobile robot* dalam berbagai lingkungan. Sistem deteksi yang andal memungkinkan *mobile robot* untuk beroperasi dengan efisien dalam aplikasi seperti logistik, perawatan kesehatan, dan eksplorasi. Namun, metode deteksi yang ada saat ini, seperti kamera visual, sensor ultrasonik, dan radar, sering menghadapi kendala seperti ketergantungan pada kondisi cahaya, keterbatasan jangkauan, atau biaya yang tinggi. Oleh karena itu, solusi baru diperlukan untuk mengatasi kelemahan-kelemahan ini dan meningkatkan kemampuan deteksi *mobile robot*.

Sensor laserping, yang memanfaatkan teknologi laser, menawarkan potensi besar dalam mengatasi keterbatasan teknologi deteksi yang ada. Dengan kemampuan mengukur jarak dengan presisi tinggi dan beroperasi dengan baik dalam berbagai kondisi pencahayaan, sensor ini mengatasi beberapa masalah yang sering dihadapi oleh sensor ultrasonik dan kamera visual. Selain itu, laserping memiliki jangkauan deteksi yang lebih luas dan dapat memberikan data secara real-time, yang memungkinkan respons cepat dan akurat dalam situasi dinamis.

Penelitian ini bertujuan untuk merancang dan mengembangkan sistem pendeteksi objek berbasis laser pada *mobile robot* menggunakan sensor laserping. Fokus utama dari penelitian ini mencakup perancangan sistem hardware untuk

integrasi sensor dengan *mobile robot*, serta pengembangan algoritma pemrosesan data untuk memanfaatkan informasi dari sensor dalam navigasi dan penghindaran rintangan. Evaluasi kinerja sistem akan dilakukan dalam berbagai kondisi untuk memastikan efektivitas sistem dalam mendeteksi dan menghindari objek dengan akurat.

Dengan memanfaatkan sensor laserping, penelitian ini diharapkan dapat membawa kontribusi signifikan dalam bidang teknologi robotika, khususnya dalam sistem deteksi objek. Pengembangan ini diharapkan dapat mengatasi beberapa keterbatasan metode deteksi konvensional dan menawarkan solusi yang lebih efisien dan akurat. Implikasi dari hasil penelitian ini dapat memperluas aplikasi *mobile robot* di berbagai industri meningkatkan kemampuannya dalam berinteraksi dengan lingkungan dan menghadapi tantangan yang ada.

1.2 Rumusan Masalah

1. Bagaimana merancang sistem pendeteksi menggunakan sensor laserping?
2. Bagaimana implementasi sistem pendeteksi objek menggunakan laserpping pada *mobile robot*?

1.3 Ruang Lingkup Kegiatan

1. Menggunakan Arduino Uno sebagai prosesor utama.
2. Menggunakan laserping sebagai pendetctksi obejk.
3. Menggunakan LCD sebagai media pembacaan output laserping.

4. Merancang beberapa sistem pendeteksi objek pada *mobile robot* dengan jarak minimal masing-masing 10 cm, 20 cm, 40 cm, dan 120 cm sehingga *mobile robot* berhenti pada jarak tersebut.
5. Menguji akurasi perhitungan jarak dari laserping.
6. Menguji sistem *mobile robot* yang secara otomatis akan berhenti saat mendeteksi objek yang berada pada jarak minimal yang telah ditentukan.

1.4 Tujuan Kegiatan

1. Merancang sistem pendeteksi objek menggunakan sensor laserping.
2. Implementasi sistem pendeteksi objek menggunakan sensor laserping pada *mobile robot*.

1.5 Manfaat Kegiatan

1. Memahami cara merancang *mobile robot* menggunakan komponen-komponen yang digunakan seperti laserping, Arduino Uno, LCD, I2C, motor servo SG90, motor DC, dan driver motor L298N.
2. Memahami prinsip kerja dari komponen-komponen yang digunakan tersebut sehingga dapat merancang *mobile robot* yang berfungsi dengan baik seperti yang diinginkan.

BAB II

TINJAUAN PUSTAKA

2.1 Arduino Uno

2.1.1 Perangkat Keras Arduino Uno

Arduino Uno adalah pengendali mikro single-board yang bersifat sumber terbuka, diturunkan dari *wiring platform*, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. Perangkat kerasnya memiliki prosesor Atmel AVR dan software-nya memiliki bahasa pemrograman sendiri. Arduino merupakan senarai perangkat keras terbuka yang ditujukan kepada siapa saja yang ingin membuat purwarupa peralatan elektronik interaktif berdasarkan *hardware* dan *software* yang fleksibel dan mudah digunakan.

Mikrokontroler diprogram menggunakan bahasa pemrograman arduino yang memiliki kemiripan *syntax* dengan bahasa pemrograman C, karena sifatnya yang terbuka maka siapa saja dapat mengunduh skema *hardware* Arduino dan membangunnya. Arduino menggunakan keluarga mikrokontroler Atmega yang dirilis oleh Atmel sebagai basis, namun ada individu/perusahaan yang membuat *clone* Arduino dengan menggunakan mikrokontroler lain dan tetap kompatibel dengan arduino pada level hardware. Untuk fleksibilitas, program dimasukkan melalui *bootloader* meskipun ada opsi untuk *bypass bootloader* dan menggunakan pengunduh untuk memprogram mikrokontroler secara langsung melalui *port ISP*.

Arduino berawal dari sebuah tesis yang berasal dari Institute Ivrea, perangkat ini diberi nama *Arduin of Ivrea*. Lalu kemudian nama perangkat ini diganti menjadi Arduino yang dalam Bahasa Italia berarti “teman yang berani”. Tujuan awal dibuatnya Arduino adalah untuk membuat perangkat yang mudah dan murah dari perangkat-perangkat yang ada pada saat itu. Hingga saat ini pengembangan Arduino selalu mengupayakan empat hal utama, yaitu:

1. Harga yang terjangkau.
2. Dapat dijalankan di berbagai sistem operasi seperti Windows, Linus, Mac, dan sebagainya.
3. Sederhana, dengan bahasa pemrograman yang mudah dan biasa dipelajari orang awam bukan untuk orang teknik saja.
4. *Open source, hardware* maupun *software*.

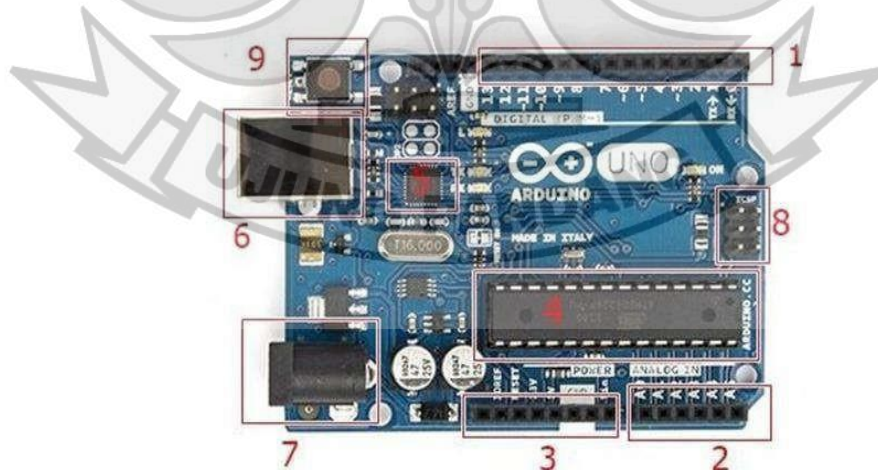
Sifat Arduino yang *open source*, membuat Arduino berkembang sangat cepat. Sehingga banyak lahir perangkat-perangkat sejenis Arduino. Seperti DFRduino atau Freeduino, sedangkan untuk lokal ada Cipduino, lalu ada Murmerduino, dan ada lagi AViShduino. Sampai saat ini pihak resmi, sudah membuat berbagai jenis-jenis Arduino. Mulai dari yang paling banyak digunakan, yaitu Arduino Uno. Hingga Arduino yang sudah menggunakan ARM Cortex, berbentuk Mini PC.

2.1.2 Bagian-Bagian Arduino Uno

Arduino Uno adalah salah satu jenis papan mikrokontroler berdasarkan ATmega328 dan *uno* adalah istilah Italia yang berarti satu. Arduino Uno dinamai untuk menandai rilis papan mikrokontroler yaitu Arduino Uno Board 1.0. Papan ini mencakup pin I/O digital-14, colokan listrik analog i/ps-6, resonator keramik A16MHz, koneksi USB, tombol RST, dan *header* ICSP.

Arduino Uno adalah salah satu produk berlabel Arduino yang sebenarnya adalah suatu papan elektronik yang mengandung mikrokontroler ATmega328 (sebuah *keeping* yang secara fungsional bertindak seperti sebuah komputer). Peranti ini dapat dimanfaatkan untuk mewujudkan rangkaian elektronik dari yang sederhana hingga yang kompleks.

Pada papan Arduino terdapat bagian-bagian antara lain ialah seperti terlihat pada Gambar 2.1.



Gambar 2.1 *Board* Arduino Uno

1. Pin *output/input* digital (diberi label “0 sampai 13”)

Secara umum pin I/O ini adalah pin digital, yakni pin yang bekerja pada level tegangan umum pin I/O adalah pin digital (0V sampai 5V) baik untuk *input* atau *output*. Namun pada beberapa pin *output* analog, yang dapat mengeluarkan tegangan analog 0V sampai 5V, pin tersebut adalah pin 3, 5, 6, 9, 10 dan 11, selain itu untuk pin 0 dan 1 juga memiliki fungsi bertindak khusus sebagai pin komunikasi serial.

2. Pin *input* analog (diberi label “A0 sampai A5”)

Pin tersebut dapat menerima *input* tegangan analog antara 0V sampai 5V, tegangan ini akan direpresentasikan sebagai bilangan 0-1023 dalam program.

3. Pin untuk sumber tegangan

Kelompok ini merupakan kumpulan pin yang berhubungan dengan sumber tenaga, misalnya output 5V, Output 3,3V, GND (2 pin) dan Vref (tegangan referensi untuk pembacaan ADC internal).

4. IC ATmega328

Seperti yang telah dijelaskan IC ini bertindak sebagai pusat kendali pemrosesan data.

5. IC ATmega16U

IC ini diprogram untuk menangani komunikasi data dengan PC melalui *port* USB.

6. Jack USB

Merupakan soket USB tipe B sebagai penghubung data serial dengan PC.

7. Jack power

Merupakan soket untuk catu daya eksternal antara 9V sampai 12V DC.

8. Port ICSP (*In-Circuit Serial Programming*)

Port ini digunakan untuk memprogram Arduino tanpa *bootloader*.

9. Tombol *reset*

Digunakan untuk *me-reset* papan mikrokontroler Arduino untuk memulai program dari awal.

2.1.3 Bahasa Pemrograman Arduino Uno

Arduino board merupakan perangkat yang berbasis mikrokontroler. Perangkat lunak (*software*) merupakan komponen yang membuat sebuah mikrokontroler dapat bekerja. Arduino *board* akan bekerja sesuai dengan perintah yang ada dalam perangkat lunak yang ditanamkan padanya. Bahasa

perograman Arduino adalah bahasa perograman utama yang digunakan untuk membuat program untuk Arduino *board*. Bahasa perograman Arduino menggunakan bahasa pemrograman C sebagai dasarnya. Karena menggunakan bahasa pemrograman C sebagai dasarnya, bahasa perograman Arduino memiliki banyak sekali kemiripan, walaupun beberapa hal telah berubah.

Arduino *Development Environment* terdiri dari editor teks untuk menulis kode, sebuah area pesan, sebuah konsol, sebuah toolbar dengan tombol tombol untuk fungsi yang umum dan beberapa menu. Arduino *Development Environment* terhubung ke Arduino *board* untuk meng-*upload* program dan juga untuk berkomunikasi dengan Arduino *board*.

Perangkat lunak yang ditulis menggunakan Arduino *Development Environment* disebut *sketch*. *Sketch* ditulis pada editor teks. *Sketch* disimpan dengan *file* berekstensi (.ino). Area pesan memberikan informasi dan pesan *error* ketika *sketch* disimpan atau dibuka. Konsol menampilkan pesan *error* ketika *sketch* disimpan atau dibuka. Konsol menampilkan *output* teks dari Arduino *Development Environment* dan juga menampilkan pesan *error* ketika *sketch* di-*compile*. Pada sudut kanan bawah dan *port* serial yang sedang digunakan. Tombol *toolbar* digunakan untuk mengecek dan meng-*upload* *sketch*, membuat, membuka atau menyimpan *sketch*, dan menampilkan serial monitor.

2.2 Laserping

2.2.1 LASER (*Light Amplification by Stimulated Emission of Radiation*)

Laser merupakan mekanisme sebuah alat yang memancarkan radiasi elektromagnetik, biasanya dalam bentuk cahaya yang tidak bisa diamati maupun dilihat dengan mata normal, menempuh anggota pancaran terstimulasi. Pancaran laser biasanya tunggal, memancarkan fotondalam pancaran koheren. Laser juga bisa dibicarakan efek dari mekanika kuantum.

Dalam teknologi laser, cahaya yang koheren menunjukkan sebuah sumber cahaya yang memancarkan panjang gelombang yang diidentifikasi dari frekuensi yang sama, beda fase yang konstan dan polarisasinya. Selanjutnya untuk menghasilkan sebuah cahaya yang koheren dari medium lasing adalah dengan mengontrol kebersihan, ukuran, dan bentuknya. Keluaran yang berkelanjutan dari laser dengan amplitude-konstan (dikenal sebagai CW atau gelombang berkelanjutan) adalah dengan memakai teknik *Q-switching*, *modelocking*, atau *gain-switching*.

Dalam operasi detak, dimana sejumlah daya puncak yang bertambah tinggi bisa dicapai. Sebuah medium laser juga bisa berfungsi sebagai penguat optik ketika di-*seed* dengan cahaya dari sumber lainnya. Sinyal yang diperkuat bisa dijadikan sangat mirip dengan sinyal *input* dalam istilah panjang gelombang, fase, dan polarisasi, ini tentunya penting dalam telekomunikasi serat optik.

Sumber cahaya umum, seperti bola lampu *incandescent*, memancarkan foton hampir ke seluruh arah, biasanya melewati spektrum elektromagnetik dari panjang gelombang yang lapang. Sifat koheren sulit ditemui pada sumber cahaya atau incoherens, dimana terjadi beda fase yang berubah-ubah selang foton yang dipancarkan oleh sumber cahaya. Secara kontras, laser biasanya memancarkan foton dalam cahaya yang sempit, terpolarisasi, sinar koheren mendekati monokromatik, terdiri dari panjang gelombang tunggal atau satu warna.

Beberapa jenis laser, seperti laser *dye* dan laser vibronik benda padat (*vibronic solid-state lasers*) bisa menghasilkan cahaya lewat jangka luas gelombang, properti ini sangat cocok untuk penciptaan detak singkat sangat pendek dari cahaya, dalam jangka femtodetik (10-15 detik). Jumlah teori mekanika kuantum dan termodinamika bisa digunakan untuk gerakan laser, walaupun nyatanya jumlah jenis laser ditemukan dengan cara *trial and error*.

2.2.2 Spesifikasi Laserring

Laserring merupakan mikrokontroler yang menggunakan metode pengukuran jarak yang mudah. Laserring ini menggunakan sensor *time of flight* (ToF) yang sangat ideal untuk melakukan pengukuran antara objek yang bergerak atau tidak bergerak. ToF (*Time of Flight*) adalah sebuah metode yang digunakan untuk mengukur jarak antara sensor dan obyek, perhitungan jarak didapatkan dari perbedaan pengiriman sinyal dan saat sinyal yang dikirimkan diterima kembali oleh sensor. Sinyal yang dikirimkan berupa paket yang

terdiri dari gelombang mikro dengan pola yang unik, sehingga sensor akan dapat mengenali sinyal tersebut.



Gambar 2.2 Sensor Laserping

Laserping dapat digunakan pada hampir semua mikrokontroler, menggunakan mode PWM atau mode serial opsional. Hal ini dirancang agar komp tibel dengan sirkuit dan kode dengan ping. Laserping dapat melakukan pengukuran jarak non-kontak dengan rentang 2-200 cm. Koneksi I/O Laserping beroperasi pada tegangan yang sama dengan yang dipasok ke pin VIN, untuk kompatibilitas dengan mikrokontroler 3.3V dan 5V. Laserping memiliki spesifikasi sebagaimana yang ditunjukkan pada Tabel 2.1.

Tabel 2.1 Spesifikasi Laserping

No.	Spesifikasi	Keterangan
1.	Jarak pengukuran	2 cm sampai 200 cm
2.	Laser	850nm VCSEL (<i>Vertical Cavity Surface Emitting Laser</i>)
3.	Kecepatan <i>refresh</i>	15 Hz (mode PWM), 22 Hz (mode serial)
4.	Suhu operasional	-10 °C sampai 60 °C
5.	Sudut pencahayaan	23°
6.	Sudut pandang	55°
7.	Dimensi PCB	22 mm × 16 mm

2.3 Liquid Crystal Display (LCD)

Liquid Crystal Display (LCD) adalah suatu jenis media tampil yang menggunakan kristal cair sebagai penampil utama. LCD sudah digunakan di berbagai bidang misalnya alat-alat elektronik seperti televisi, kalkulator, ataupun layar komputer. Pada rangkaian peralatan aplikasi LCD yang digunakan ialah LCD dot matrik dengan jumlah karakter 2×16 . Komponen yang menampilkan tulisan yang salah satu jenisnya memiliki dua baris dengan setiap baris terdiri dari 16 karakter. LCD seperti itu bisa disebut LCD 16×2 .



Gambar 2.3 LCD (*Liquid Crystal Display*)

LCD ini memiliki 16 pin dengan fungsi pin masing-masing diperlihatkan pada Tabel 2.2.

Tabel 2.2 Pin-Pin LCD

No. Pin	Nama Pin	I/O	Fungsi
1	VSSS	Power	Catu daya, <i>ground</i> (0V)
2	VDD	Power	Catu daya positif untuk <i>logic</i> (+5V)
3	VEE	Power	Pengatur kontras. Menurut datasheet, pin ini perlu dihubungkan dengan pin VSS melalui resistor 5k Ω . Namun, dalam praktik, resistor yang digunakan sekitar 2,2 k Ω .

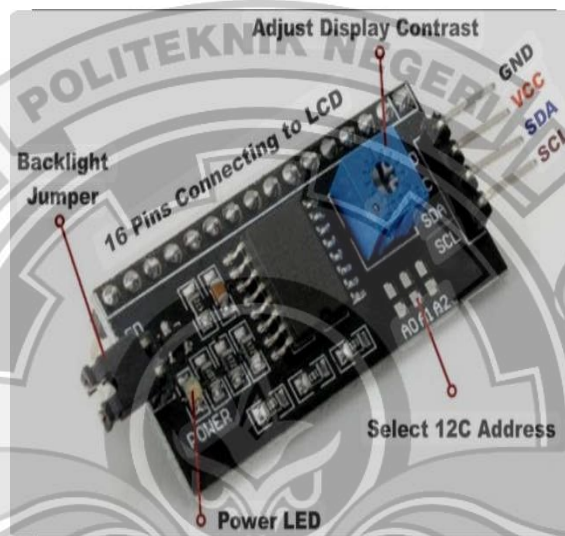
No. Pin	Nama Pin	I/O	Fungsi
4	RS	<i>Input</i>	Register <i>Select</i> <ul style="list-style-type: none"> • RS = <i>HIGH</i>: untuk mengirim data. • RS = <i>LOW</i>: untuk mengirim instruksi.
5	RW	<i>Input</i>	Read/Write control bus <ul style="list-style-type: none"> • R/W = <i>HIGH</i>: mode untuk membaca data di LCD. • R/W = <i>LOW</i>: mode penulisan ke LCD. • Dihubungkan dengan <i>LOW</i> untuk mengirim data ke layar.
6	E	<i>Input</i>	Data enable signal, untuk mengontrol ke LCD. Ketika bernilai <i>LOW</i> , LCD tidak dapat diakses.
7-14	DB0-DB7	<i>I/O</i>	Data bus line.
15	BLA	<i>Power</i>	Catu daya layar, positif (+5).
16	BLK	<i>Power</i>	Catu daya layar, positif (+5).

Pada sistem *mobile robot* ini digunakan digunakan LCD 16×2 yang memiliki 2 baris 16 kolom. LCD ini menggunakan IC I2C sebagai kontroler. Dalam aplikasinya LCD berfungsi sebagai penampil hasil *output* dari sensor yang digunakan. Sinyal yang ditampilkan berupa hasil pengukuran dari laserping.

2.4 Inter-Integrated Circuit (I2C)

I2C merupakan singkatan dari *Inter-Integrated Circuit*, yang disebut dengan *I-squared-C* atau *I-two-C*. I2C merupakan protokol yang digunakan pada *multi-master serial computer bus* yang diciptakan oleh Philips yang digunakan untuk saling berkomunikasi dengan perangkat *low-speed* lainnya yang diaplikasikan pada *motherboard*, *embedded system*, atau *cellphone*. Jalur I2C bus hanya

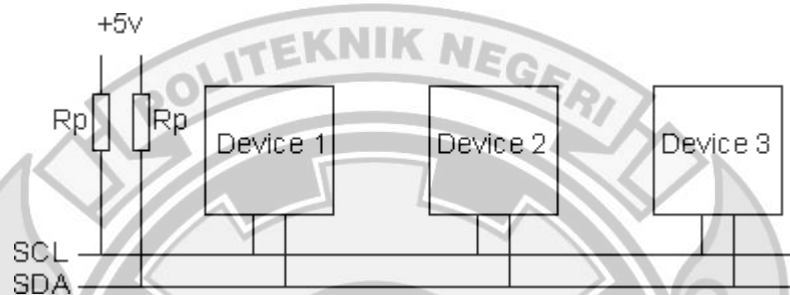
merupakan 2 jalur yang disebut dengan *SDA line* dan *SCL line*, dimana *SCL line* merupakan jalur untuk clock dan *SDA line* merupakan jalur untuk data. Semua peralatan yang akan digunakan dihubungkan seluruhnya pada jalur *SDA line* dan *SCL line* dari I2C bus tersebut.



Gambar 2.4 I2C (*Inter-Integrated Circuit*)

Jenis komunikasi yang dilakukan antar peralatan dengan menggunakan protokol I2C mempunyai sifat *serial synchronous half duplex bidirectional*, dimana yang data ditransmisikan dan diterima hanya melalui satu jalur data *SDA line* (bersifat *serial*), setiap penggunaan jalur data bergantian antar perangkat (bersifat *half duplex*) dan data dapat ditransmisikan dari dan ke sebuah perangkat (bersifat *bidirectional*). Sumber *clock* yang digunakan pada I2C bus hanya berasal dari satu perangkat *master* melalui jalur clock *SCL line* (bersifat *synchronous*). Kedua jalur *SDA* dan *SCL* merupakan *driver* yang bersifat "*open drain*", yang berarti bahwa IC yang digunakan dapat men-*drive output*-nya *low*, tetapi tidak dapat men-*drive* menjadi *high*. Untuk dapat mendapatkan data yang *high* maka

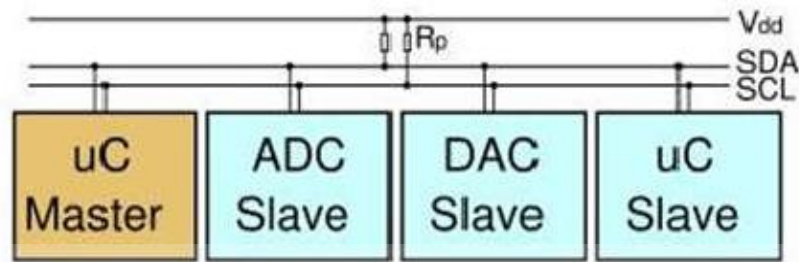
harus disediakan resistor *pull-up* pada tegangan *power supply* sebesar 5 volt terhadap jalur SDA dan SCL tersebut. Hanya membutuhkan satu set *pull-up* resistor untuk semua jalur I2C bus, tidak untuk semua perangkat yang digunakan, pemasangan resistor *pull-up* dapat dilihat seperti pada Gambar 2.5.



Gambar 2.5 Pemasangan Resistor *Pull-up*

Jika resistor-resistor tersebut tidak ada, maka jalur SCL dan SDA akan selalu mendekati *low* -mendekati 0 volt dan jalur-jalur I2C bus tidak dapat bekerja. Nilai resistor yang dapat digunakan berkisar dari 1,8K Ω hingga 4,7K Ω . Biasanya nilai 1,8K, 4,7K dan 10K merupakan nilai-nilai yang umum digunakan tetapi semua nilai yang berada dalam *range* nilai di atas dapat digunakan dan bekerja dengan baik.

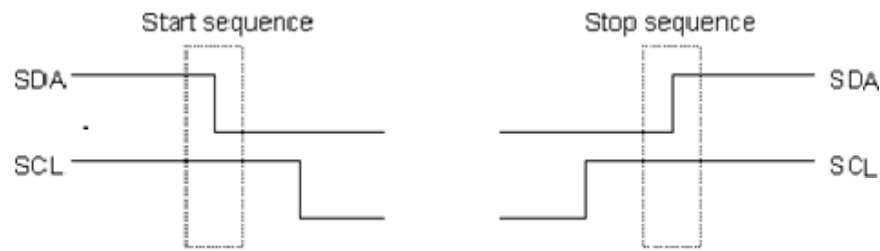
Semua perangkat yang terdapat dalam jalur I2C bus merupakan perangkat *slave* dan *master*, dimana *master* merupakan perangkat yang berfungsi sebagai pengatur (*controller*) dan sumber *clock* bagi perangkat-perangkat *slave* yang terdapat dalam I2C bus.



Gambar 2.6 Perangkat *Master* dan *Slave*

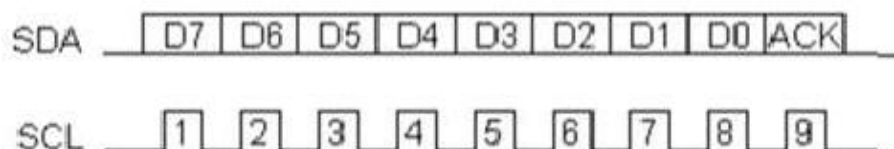
Dalam sebuah jalur I2C bus selain memungkinkan adanya penggunaan banyak perangkat *slave*, juga bisa menggunakan beberapa perangkat *master* dalam sebuah jalur I2C bus yang sama. Jika menggunakan *multiple* perangkat *master* dalam sebuah jalur I2C bus maka penggunaan jalur bergantian antar tiap-tiap perangkat *master*, tetapi umumnya hanya digunakan satu *master device* dengan *multiple* perangkat *slave* pada sebuah jalur I2C bus. Kecepatan transfer data dari protokol I2C ditentukan oleh besar *clock speed* yang digunakan pada jalur SCL *line*. Kecepatan *clock* standar yang diberikan pada jalur SCL *line* pada I2C sebesar 100 KHz. Philips sebagai pencipta protocol I2C ini membuat standar kecepatan I2C lainnya yaitu *Fast Mode* yang mempunyai kecepatan *clock* sebesar 400 KHz dan *High Speed Mode* yang mempunyai kecepatan hingga sebesar 3.4 MHz.

Untuk melakukan transmisi data pada sebuah jalur I2C bus, dimulai dengan mengirimkan sebuah *start sequence* dan diakhiri dengan mengirimkan *stop sequence*. *Start sequence* dan *stop sequence* menandakan awal dan akhir dari proses transmisi data dengan perangkat yang lainnya dalam sebuah jalur I2C bus. Berikut merupakan gambar dari *start sequence* dan *stop sequence*.



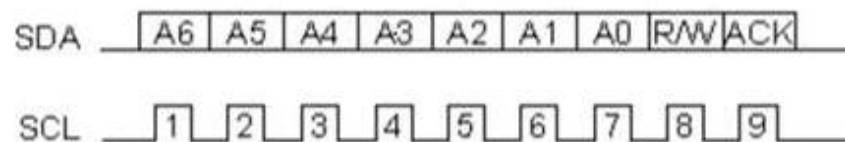
Gambar 2.7 *Start Sequence* dan *Stop Sequence*

Transmisi data antar perangkat terjadi setelah *start sequence* dan sebelum *stop sequence*. Data yang ditransmisikan sejumlah 8 bit dengan MSB (*Most Significant Bit*) yang dikirimkan terlebih dahulu hingga kepada LSB (*Least Significant Bit*) kemudian selalu terdapat tambahan satu bit yang merupakan *ACK Bit (Acknowledgement Bit)*. *ACK bit* digunakan untuk mengetahui kondisi transmisi data, jika *ACK bit* berupa kondisi *low* maka perangkat yang ada sudah menerima data dan siap untuk menerima data yang selanjutnya, sedangkan *ACK bit* berupa kondisi *high* maka perangkat yang ada sudah tidak dapat melakukan transmisi data dan *master* harus mengirimkan *stop sequence* untuk menghentikan komunikasi yang sedang berlangsung. Pada saat berlangsung komunikasi antar perangkat dalam sebuah jalur I2C bus, *bit* data dikirimkan pada saat jalur SCL *line* dalam kondisi *high* dan pergantian *bit* data terjadi pada saat jalur SCL *line* dalam kondisi *low*, yang seperti dapat dilihat pada Gambar 2.8.



Gambar 2.8 Transmisi Data Pada SDA *Line* dan SCL *Line*

Pada sebuah jalur I2C bus ditujukan untuk mengendalikan beberapa perangkat *slave* dengan menggunakan sebuah perangkat *master*. Setiap perangkat *slave* pada jalur I2C bus masing-masing mempunyai alamat I2C yang berbeda-beda. Jumlah pengalamatan yang umumnya digunakan pada sebuah protokol I2C sebesar 7 bit alamat, sehingga pada sebuah jalur I2C bus dapat digunakan perangkat *slave* sebanyak 2^7 perangkat dengan alamat yang berkisar antara 0 sampai dengan 127. Pada saat mengirimkan 7 bit alamat sebuah perangkat *slave* selalu mengirimkan 8 bit data, yaitu 7 bit alamat dan 1 bit R/W. Bit R/W (*Read/Write*) digunakan untuk memberitahukan kepada perangkat *slave* yang dipanggil tindakan yang akan dilakukan perangkat *master* pada perangkat *slave* tersebut, dimana *read* perangkat *master* akan melakukan pembacaan data dari perangkat *slave* tersebut dan *write* perangkat *master* akan melakukan pengiriman data pada perangkat *slave* tersebut. Untuk melakukan *read* maka pada bit R/W diberikan kondisi *logic high*, sedangkan untuk melakukan *write* maka pada bit R/W diberikan kondisi *logic low*. Pengiriman alamat perangkat *slave* pada sebuah *sequence protocol* I2C dapat dilihat pada Gambar 2.9.



Gambar 2.9 Transmisi Alamat Perangkat *Slave* Pada *Sequence Protocol* I2C

Sebuah perangkat *master* dapat melakukan dua tindakan pada perangkat *slave* yang terhubung dalam sebuah jalur I2C bus, yaitu melakukan *read* dari perangkat *slave* atau melakukan *write* ke perangkat *slave*. Penggunaan protokol

untuk melakukan *read* atau *write* sedikit berbeda, namun untuk memulai komunikasi antara *master* dan *slave* sama untuk *write* atau *read*, yaitu dengan mengirimkan *start sequence* kemudian alamat perangkat *slave* yang dituju dan tindakan yang akan dilakukan *write* atau *read*, dan pada akhirnya mengirimkan *stop sequence* untuk menyelesaikan komunikasi yang sedang berlangsung.

Pembahasan protokol yang digunakan oleh perangkat *master* untuk melakukan *read data byte* dan *write data byte* pada sebuah perangkat *slave* dibahas terpisah sebagai berikut:

- Untuk melakukan *write* pada sebuah perangkat *slave* maka langkah-langkah yang harus dilakukan oleh sebuah perangkat *master* sebagai berikut:
 1. Mengirimkan *start sequence*.
 2. Mengirimkan alamat perangkat *slave* dengan *bit R/W low*.
 3. Mengirimkan (*write*) *command register* yang diinginkan.
 4. Mengirimkan (*write*) *data byte* ke perangkat *slave*.
 5. Optional, mengirimkan (*write*) *data bytes* lainnya.
 6. Mengirimkan *stop sequence*.
- Untuk melakukan *read* pada sebuah perangkat *slave*, pertama kali perangkat *master* harus memberitahukan *internal address* perangkat *slave* yang ingin dibaca. Jadi untuk melakukan *read* dari sebuah *slave*, sebenarnya dimulai dengan melakukan *write* pada perangkat *slave* tersebut. Untuk melakukan

read pada sebuah perangkat *slave* maka langkah-langkah yang harus dilakukan oleh sebuah perangkat *master* sebagai berikut:

1. Mengirimkan *start sequence*.
2. Mengirimkan alamat perangkat *slave* dengan *bit R/W low*.
3. Mengirimkan (*write*) *command register* yang diinginkan.
4. Mengirimkan *start sequence* kembali (*repeated start*).
5. Mengirimkan alamat perangkat *slave* dengan *bit R/W high*.
6. Membaca (*read*) *data byte* dari perangkat *slave*.
7. Opsional, membaca (*read*) *data bytes* lainnya.
8. Mengirimkan *stop sequence*.

I2C mempunyai keterbatasan dalam hal pengalamatan perangkat *slave* yang digunakan. Kebanyakan vendor perangkat *slave* I2C tidak membebaskan kepada pengguna untuk merancang sendiri alamat *slave* yang diinginkan, kebanyakan vendor hanya memberikan 3 pin alamat yang bebas dikonfigurasi sendiri sedangkan 4 bit lainnya merupakan *fixed* yang merupakan identitas model dari perangkat tersebut ataupun identitas dari vendor pembuat perangkat tersebut. Hal ini diatasi dengan pengalamatan I2C dengan 10 bit alamat namun penggunaannya belum umum. Selain itu I2C mempunyai kecepatan yang terbatas sehingga implementasi protokol I2C pada kelas *high end* dapat menyebabkan terjadinya kekurangan *bandwidth*, sedangkan untuk mengimplementasikan protokol I2C

pada kelas *low end* membutuhkan *dedicated hardware* untuk mengatasi *master* berada dalam kondisi *busy*. Penggunaan kecepatan transfer *cl ck* sangat kritis karena I2C hanya dapat bekerja dengan kecepatan *clock* standar yang ada, jika tidak protokol I2C tersebut tidak dapat digunakan.

2.5 Motor Servo SG90

Motor Servo SG90 adalah jenis motor servo yang paling sering digunakan untuk proyek-proyek sederhana menggunakan Arduino. Motor servo ini memiliki 3 kaki yang terdiri dari *Ground*, *Vcc* dan *Sg* (PWM), bekerja pada tegangan 4.8 – 6 volt. Motor servo biasanya digunakan dimana sesuatu yang dibutuhkan pindah ke posisi tertentu dan kemudian berhenti dan bertahan pada posisi tersebut. Motor servo dapat diperintahkan untuk memutar dengan sudut tertentu dan selanjutnya kemudi akan tetap bertahan posisi tersebut. Motor servo menggunakan mekanisme umpan balik, sehingga dapat merasakan kesalahan dalam posisinya. Ketika Motor servo berada dan terkunci pada posisi 30 derajat dan kemudian dicoba untuk memutar dengan tangan, maka servo akan berusaha keras sebaik mungkin untuk mengatasi gaya dan menjaga servo terkunci pada sudut yang ditentukan. Berikut tampilan dari Motor servo SG90.



Gambar 2.10 Motor Servo SG90

Input Motor servo ke kontrolnya bisa berupa sinyal analog dan juga sinyal digital. Motor servo digunakan sebagai aktuator yang membutuhkan posisi putaran motor yang presisi, sedangkan sudut dari sumbu Motor servo diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel motor.

Sistem kontrol kalang tertutup pada Motor servo digunakan untuk mengontrol gerakan dan posisi akhir dari poros Motor servo. Posisi poros *output* akan dideteksi untuk mengetahui posisi poros sudah tepat atau belum. Jika posisi poros belum tepat maka kontrol *input* akan mengirim sinyal kendali yang membuat posisi poros tersebut tepat sesuai yang diinginkan. Motor servo yang digunakan dalam penelitian ini bertipe SG90 dengan spesifikasi sebagaimana ditunjukkan pada Tabel 2.3.

Tabel 2.3 Spesifikasi Motor Servo SG90

No.	Spesifikasi	Keterangan
1.	Dimensi	22.2 mm x 11.8 mm x 31 mm
2.	Berat	9 gram
3.	<i>Stall torque</i>	1.8 kg.cm
4.	Kecepatan Operasi	0.1 s/60 derajat
5.	Tegangan Operasi	4.8 V (5 V)
6.	<i>Dead band width</i>	10 μ s
7.	Rentang Suhu	0 $^{\circ}$ C – 55 $^{\circ}$ C

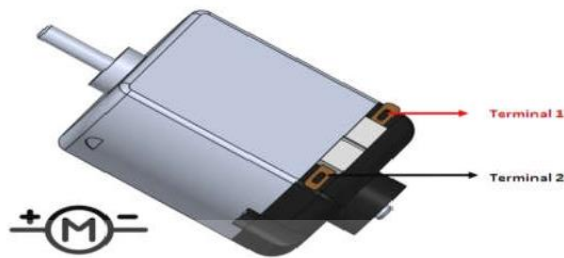
Teknik pengendalian Motor servo menggunakan PWM dengan memanfaatkan lebar pulsa. Untuk menggerakkan Motor servo ke kanan atau kiri tergantung pada *delay* ketika PWM kondisi *high*. Misalkan untuk membuat servo pada posisi center memberikan *delay* 1,5 μ s pada periode PWM 20 μ s. Dengan

cara kerja yang sederhana ini, Motor servo sangat cocok bila digunakan pada suatu alat mekanis yang membutuhkan presisi yang tinggi. Desain mekanik, komponen kontroler, pengendalian yang menggunakan Motor servo menunjukkan kestabilan kerja sehingga layak untuk dioperasikan. Oleh sebab itu pada *mobile robot* ini menggunakan Motor servo sebagai dudukan laserping sehingga laserping dapat berdiri stabil dan mendeteksi objek secara akurat, Motor servo juga dapat memutar laserping ke kanan dan kiri sehingga objek di sebelah *mobile robot* dapat terdeteksi saat *mobile robot* akan berbelok.

2.6 Motor DC

2.6.1 Pengertian Motor DC

Motor DC merupakan sebuah motor listrik mengubah energi listrik menjadi energi mekanik. Kebanyakan motor listrik beroperasi melalui interaksi medan magnet dan konduktor pembawa arus untuk menghasilkan kekuatan, meskipun motor elektrostatik menggunakan gaya elektrostatik. Proses sebaliknya, menghasilkan energi listrik dari energi mekanik, yang dilakukan oleh generator seperti alternator, atau dinamo. Banyak jenis motor listrik dapat dijalankan sebagai generator, dan sebaliknya. Motor listrik dan generator yang sering disebut sebagai mesin-mesin listrik.

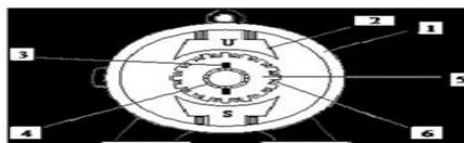


Gambar 2.11 Motor DC

Motor listrik DC (arus searah) merupakan salah satu dari motor DC. Mesin arus searah dapat berupa generator DC atau motor DC. Generator DC alat yang mengubah energi mekanik menjadi energi listrik DC. Motor DC alat yang mengubah energi listrik DC menjadi energi mekanik putaran. Sebuah motor DC dapat difungsikan sebagai generator atau sebaliknya generator DC dapat difungsikan sebagai motor DC.

Pada motor DC kumparan medan disebut stator (bagian yang tidak berputar) dan kumparan jangkar disebut rotor (bagian yang berputar). Jika terjadi putaran pada kumparan jangkar dalam pada medan magnet, maka akan timbul tegangan yang berubah-ubah arah pada setiap setengah putaran, sehingga merupakan tegangan bolak-balik.

2.6.2 Bagian-Bagian Motor DC



Gambar 2.12 Bagian-Bagian Motor DC

1. Badan mesin

Badan mesin ini berfungsi sebagai tempat mengalirnya fluks magnet yang dihasilkan kutub magnet, sehingga harus terbuat dari bahan *ferromagnetic*. Fungsi lainnya adalah untuk meletakkan alat-alat tertentu dan mengelilingi bagian-bagian dari mesin, sehingga harus terbuat dari bahan yang benar-benar kuat, seperti dari besi tuang dan plat campuran baja.

2. Inti kutub magnet dan belitan penguat magnet

Inti kutub magnet dan belitan penguat magnet ini berfungsi untuk mengalirkan arus listrik agar dapat terjadi proses elektromagnetik. Adapun aliran fluks magnet dari kutub utara melalui celah udara yang melewati badan mesin.

3. Sikat-sikat

Sikat-sikat ini berfungsi sebagai jembatan bagi aliran arus jangkar dengan bebas, dan juga memegang peranan penting untuk terjadinya proses komutasi.

4. Komutator

Komutator ini berfungsi sebagai penyearah mekanik yang akan dipakai bersama-sama dengan sikat. Sikat-sikat ditempatkan sedemikian rupa sehingga komutasi terjadi pada saat sisi kumparan berbeda.

5. Jangkar

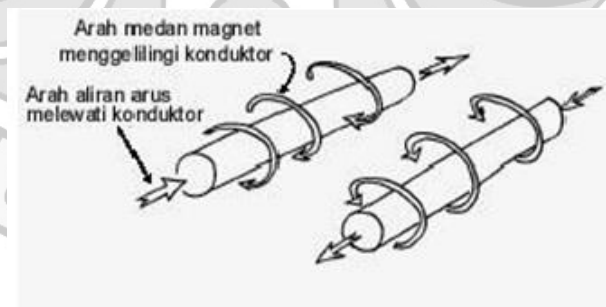
Jangkar dibuat dari bahan *ferromagnetic* dengan maksud agar kumparan jangkar terletak dalam daerah yang induksi magnetiknya besar, agar tegangan induksi yang dihasilkan dapat bertambah besar.

6. Belitan jangkar

Belitan jangkar merupakan bagian yang terpenting pada mesin arus searah, berfungsi untuk tempat timbulnya tenaga putar motor.

2.6.3 Prinsip Kerja Motor DC

Prinsip kerja pada Motor DC adalah jika arus lewat pada suatu konduktor, timbul medan magnet di sekitar konduktor. Medan magnet hanya terjadi di sekitar sebuah konduktor jika ada arus mengalir pada konduktor tersebut. Arah medan magnet ditentukan oleh arah aliran arus pada konduktor dapat dilihat pada Gambar 2.13.



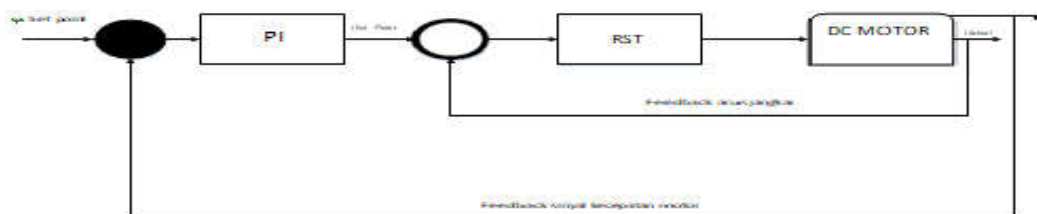
Gambar 2.13 Prinsip Kerja Motor DC

Pada motor DC, daerah kumparan medan yang dialiri arus listrik akan menghasilkan medan magnet yang melingkupi kumparan jangkar dengan arah tertentu. Konversi dari energi listrik menjadi energi mekanik (motor) maupun sebaliknya berlangsung melalui medan magnet, dengan demikian medan

magnet disini selain berfungsi sebagai tempat untuk menyimpan energi, sekaligus sebagai tempat berlangsungnya proses perubahan energi, agar proses perubahan energi mekanik dapat berlangsung secara sempurna, maka tegangan sumber harus lebih besar dari pada tegangan gerak yang disebabkan reaksi lawan. Dengan memberi arus pada kumparan jangkar, karena arus motor berbanding lurus dengan torsi motor kemudian akan masuk ke sistem penggerak maka menimbulkan perputaran pada motor.

2.6.4 Kaskade Kontrol Motor DC

Kontrol kaskade adalah sistem pengendalian yang diperlukan pada suatu loop kontrol yang membutuhkan satu sistem pengontrolan yang bertingkat. Pengendalian kaskade atau bertingkat ini sering juga disebut dengan pengendalian *master* dan *slave* dimana *master* bertindak sebagai pengontrol pertama sedangkan *slave* bertindak sebagai pengendali kedua yang mendapat *signal input remote* dari *master loop*. Berikut ini adalah contoh blok diagram suatu sistem kontrol bertingkat dimana terdapat *primary control* sebagai pengontrol utama dan *secondary control* sebagai pengendali kedua, seperti pada Gambar 2.14.



Gambar 2.14 Blok Diagram Kaskade Kontrol Motor DC

Dari Gambar 2.14 terlihat bahwa ada dua jalur umpan balik pada sistem pengendalian bertingkat (*cascade control*), sehingga terbentuk dua mata rantai pengendalian (kalang). Mata rantai atau kalang bagian luar (*outer loop*) disebut *primary loop* atau *master*, dan mata rantai atau kalang bagian dalam (*inner loop*) disebut *secondary loop* atau *slave*. *Master* atau *primary loop* mengendalikan proses variabel primer (kecepatan motor dc). Sedangkan *slave* atau *secondary loop* mengendalikan proses variabel sekunder (arus jangkar). Alasan penggunaan pengendalian bertingkat (*cascade control*) dalam mengendalikan *plant* adalah sebagai berikut.

1. Respon keluaran dari *single control* tidak sesuai dengan yang diharapkan.
2. Terdapat penambahan variabel sekunder di dalam pengendalian *plant*.
3. Dengan adanya pengendali sekunder yang lebih cepat, dapat mengatasi gangguan pada kalang sekunder.

Alasan tidak digunakannya pengendalian bertingkat (*cascade control*) adalah.

1. Biaya atau rugi-rugi pengukuran variabel sekunder.
2. Keruwetan pada pengendaliannya.

Berikut langkah-langkah penalaan di dalam pengendalian bertingkat (*cascade control*).

1. Meletakkan kedua kalang pada posisi manual.

2. Pengoperasian kalang harus selalu dimulai dari kalang sekunder.
3. Memilih mode manual untuk mencari parameter kontrol pada kalang sekunder, namun tidak sampai mengganggu proses variabel primer.
4. Setelah menyetting di kalang sekunder menghasilkan respon yang cukup mantap, kemudian mempersiapkan metode *tuning* untuk mencari parameter kontrol pada kalang primer.
5. Meletakkan kalang primer pada posisi *tuning* dan melakukan penalaan kalang primer

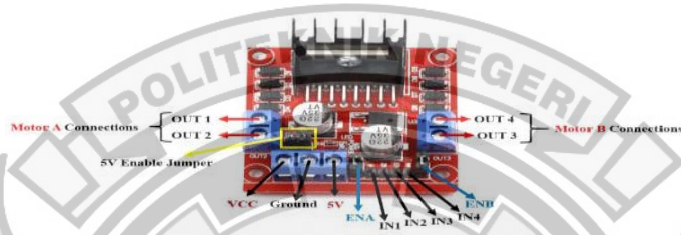
2.7 Driver Motor L298N

Driver motor L298N merupakan module driver motor DC yang paling banyak digunakan atau dipakai di dunia elektronika yang difungsikan untuk mengontrol kecepatan serta arah perputaran motor DC.

IC L298N merupakan sebuah IC tipe *H-bridge* yang mampu mengendalikan beban- beban induktif seperti *relay*, *solenoid*, motor DC dan motor *stepper*. Pada IC L298N terdiri dari transistor-transistor logik (TTL) dengan gerbang NAND yang berfungsi untuk memudahkan dalam menentukan arah putaran suatu motor dc maupun motor *stepper*.

Untuk di pasaran sudah terdapat modul driver motor menggunakan IC L298N ini, sehingga lebih praktis dalam penggunaannya karena pin I/O nya sudah tersusun dengan rapi dan mudah digunakan. Kelebihan akan modul driver motor

L298N ini yaitu dalam hal kepresisian dalam mengontrol motor sehingga motor lebih mudah untuk dikontrol. Adapun gambar dari pinout dari driver motor L298N beserta keterangannya dapat diperhatikan pada Gambar 2.15 dan Tabel 2.4.



Gambar 2.15 Pinout Driver Motor L298N

Tabel 2.4 Pinout Driver Motor L298N

Nama Pin	Keterangan
VCC	Ini adalah pin yang menyalurkan daya ke motor. Pin ini memiliki tegangan +12V pada papan tetapi dapat dialiri daya antara 6-12V.
Ground	Ini adalah pin ground seperti pada umumnya.
5V	Pin ini menyuplai daya (5V) untuk sirkuit internal (IC L298N). Hanya akan digunakan jika jumper pengaktif 5V tidak utuh. Jika jumper utuh, maka ia bertindak sebagai pin output.
ENA	Pin ini mengontrol kecepatan motor A dengan mengaktifkan sinyal PWM.
IN1 & IN2	Ini adalah pin input untuk motor A. Pin ini mengendalikan arah putaran untuk motor tertentu.
IN3 & IN4	Ini adalah pin input untuk motor B. Pin ini mengendalikan arah putaran untuk motor tertentu.
ENB	Pin ini mengontrol kecepatan motor B dengan mengaktifkan sinyal PWM.
OUT1 & OUT2	OUT1: Terminal positif. OUT2: Terminal negatif Ini adalah pin keluaran untuk motor A. Motor A yang memiliki tegangan antara 5-35V, akan dihubungkan melalui kedua terminal ini.
OUT3 & OUT4	OUT3: Terminal positif. OUT4: Terminal negatif

BAB III

METODOLOGI PENELITIAN

3.1 Waktu dan Tempat Penelitian

Perancangan dan pengujian alat dilaksanakan di Jurusan Teknik Elektro, Kampus 1 Politeknik Negeri Ujung Pandang serta di sekitar ruang lingkup kampus PNUP dan tempat tinggal penulis. Penelitian ini dilaksanakan pada bulan April sampai dengan September 2023.

3.2 Alat dan Bahan Penelitian

3.2.1 Perangkat Keras (*Hardware*)

Perangkat keras (*Hardware*) adalah perangkat keras yang akan menjalankan suatu sistem. Adapun perangkat keras yang digunakan pada penelitian ini ialah.

1. Laptop.
2. Arduino Uno.
3. Laserping 2m range finder.
4. LCD 16 × 2.
5. I2C.
6. Motor Servo SG90.
7. Motor DC.
8. Driver Motor L289N.

3.2.2 Perangkat Lunak (*Software*)

Perangkat lunak (*software*) adalah perangkat lunak yang akan menjalankan perangkat keras dalam suatu sistem. Adapun perangkat lunak yang digunakan pada penelitian ini ialah:

1. Arduino IDE.
2. Lybrary AFMotor.h.
3. Lybrary I2C LCD.
4. Lybrary Servo.h.

3.3 Tahap Penelitian

3.3.1 Studi Literatur

Langkah awal dari suatu proses perancangan adalah studi, berfungsi sebagai landasan dari pembuatan proyek akhir baik dalam proses perencanaan, hingga proses penyelesaian proyek akhir yang di buat. Studi literatur berguna dalam pemahaman spesifikasi dan cara kerja alat secara keseluruhan.

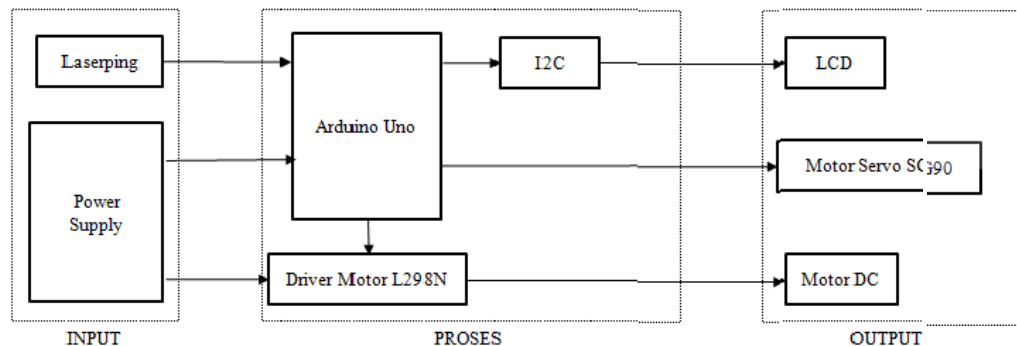
3.3.2 Perancangan *Hardware*

Pada perancangan *hardware* ini laserping dan *power supply* akan berfungsi sebagai *input*. Power supply menjadi sumber tenaga bagi Arduino Uno dan Motor Driver L298N. Laserping berfungsi mangambil data

pengukuran jarak objek dari *mobile robot* yang nantinya akan dikirim ke Arduino Uno.

Arduino Uno, Motor Driver L298N, dan I2C akan memproses data yang diterima dari laserping. Arduino membaca data-data yang disampaikan oleh laserping yang nantinya berdasarkan data tersebut Arduino akan mengirimkan perintah ke perangkat keras yang lainnya. Motor Driver L289N nantinya akan menerima perintah untuk menggerakkan Motor DC berdasarkan data yang diterima oleh Arduino, dan I2C akan diperintahkan untuk menampilkan data laserping di LCD.

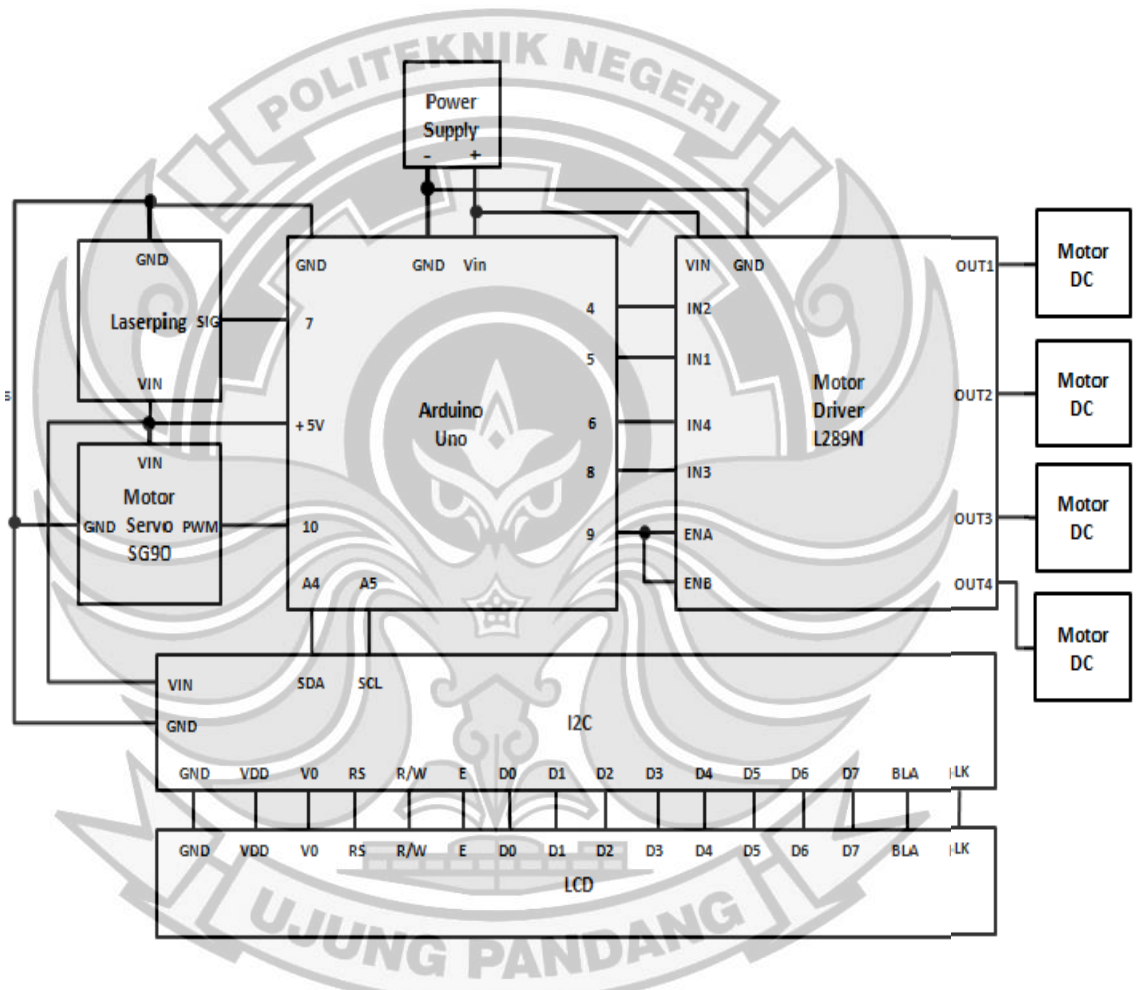
LCD nantinya berfungsi sebagai *output* yang menampilkan data pengukuran dari laserping yang telah diterima dari I2C dan Arduino. Motor DC akan menggerakkan roda *robot* berdasarkan perintah yang diterima dari Driver Motor L298N. Dan Motor Servo akan memutar laserping ke kanan dan kiri saat diperintahkan oleh Arduino Uno. Untuk penggambaran lebih jelas dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Blok Perancangan Sistem *Hardware Mobile Robot*

3.3.3 Gambar Rangkaian

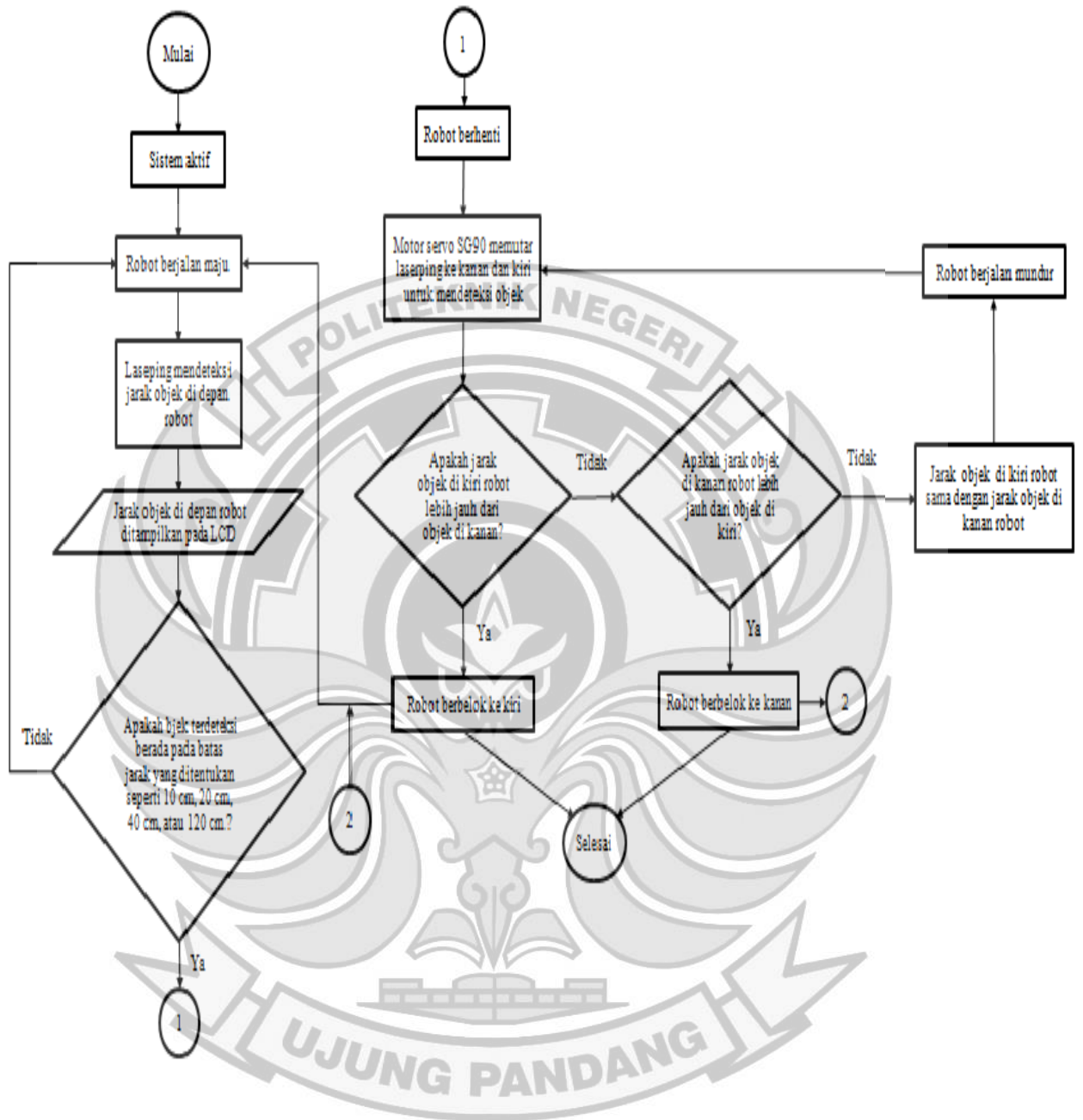
Gambar rangkaian ini dibuat dengan tujuan sebagai acuan dalam perangkaian perangkat keras yang dapat dilihat pada Gambar 3.2. Rangkaian ini dirancang sesuai dari apa yang telah dijelaskan pada sub-subbab 3.3.2.



Gambar 3.2 Rangkaian Perangkat Keras *Mobile Robot*

3.3.4 Perancangan *Software*

Perancangan software pada penelitian ini menggunakan aplikasi Arduino IDE. Gambar 3.3 menunjukkan diagram alir program yang dibuat.



Gambar 3.3 Diagram Alir Program *Mobile Robot*

Program dibuat untuk menjalankan seluruh perangkat keras dalam satu sistem sebagai *mobile robot*. Sistem dimulai dengan *mobile robot* berjalan maju seperti pada umumnya dan laser ping mendeteksi objek di depan *mobile robot*. Apabila laser ping mendeteksi adanya objek yang berada di depan

mobile robot di bawah jarak minimal yang ditentukan oleh program maka *mobile robot* berhenti secara otomatis. Setelah *mobile robot* berhenti laserping akan mendeteksi keberadaan objek di samping *mobile robot*, nantinya *mobile robot* secara otomatis akan berbelok ke arah yang tidak dihalangi oleh objek.

3.4 Pengujian Alat

Setelah proses perancangan dan pembuatan prototype dari *mobile robot* selesai, tahapan selanjutnya adalah pengujian alat tersebut. Tahapan ini dimaksudkan untuk mengetahui kinerja alat dan sistem yang telah dibuat. Tahapan pengujian ini meliputi laserping, sistem berhenti otomatis dari *mobile robot*, dan sistem belok otomatis dari *mobile robot*.

3.4.1 Tujuan Pengujian

Pengujian yang dilakukan bertujuan untuk mengetahui cara kerja dari alat yang telah dibuat. Dari data-data yang diperoleh dari pengujian ini dapat mempermudah dalam membahas cara kerja alat secara keseluruhan. Selain itu, dengan melakukan pengujian juga akan memudahkan untuk menemukan gangguan dan kerusakan pada alat dan sistem yang telah dibuat.

3.4.2 Langkah-Langkah Pengujian

Ada tiga hal yang akan diuji pada pengujian kali ini.

1. Pengujian akurasi pengukuran jarak laserping.

2. Pengujian sistem berhenti otomatis *mobile robot*.

3.4.2.1 Langkah-Langkah Pengujian Akurasi Pengukuran Jarak Laserping

1. Taruh objek di depan laserping dengan jarak yang telah ditentukan seperti 20 cm, 50 cm, 100 cm, 150 cm, 200 cm, 210 cm, dan 220 cm.
2. Catat hasil pengukuran jarak yang dihitung laserping.

3.4.2.2 Langkah-Langkah Pengujian Sistem Berhenti Otomatis *Mobile Robot*.

1. Atur program sistem *mobile robot* dengan batas jarak pengukuran laserping 10 cm, 20 cm, 40 cm, atau 120 cm. Ini akan membuat *mobile robot* akan berhenti secara otomatis saat laserping mendeteksi objek di depan *mobile robot* berada pada jarak tersebut.
2. Jalankan *mobile robot* sampai *mobile robot* berhenti secara otomatis.
3. Catat jarak titik berhenti *mobile robot* dengan objek yang ada di depannya.
4. Ulangi hingga tiga kali percobaan.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Perancangan Alat

4.1.1 Hasil *Prototype* Pendeteksi Objek Berbasis Laser Pada *Mobile Robot*

Prototype yang dibuat memiliki bentuk seperti kendaraan beroda empat yang mana tiap rodanya dipasang di Motor DC. Motor DC terhubung dengan L298N. Laserping sebagai perangkat pendeteksi objek ditaruh di bagian depan dan berada di atas motor servo yang keduanya sendiri akan terhubung dengan Arduino Uno di bagian tengah *mobile robot*.



Gambar 4.1 *Prototype Mobile Robot*

4.1.2 Cara Kerja Alat

1. Nyalakan *mobile robot*.
2. *Mobile robot* akan berjalan maju.
3. Laserping menghitung jarak objek yang berada di depan *mobile robot* dan hasil perhitungan ditampilkan pada LCD.

4. *Mobile robot* akan berhenti saat jarak antara *robot* dengan objek berada di batas jarak minimal.
5. Motor servo akan memutar laserping ke kanan dan kiri agar laserping dapat menghitung jarak objek di samping *mobile robot*.
6. Setelah laserping menghitung jarak objek di samping *mobile robot*, L298N akan mengarahkan *robot* untuk berbelok ke arah yang tidak dihalangi oleh objek.

4.2 Hasil dan Analisa Pengujian Alat

4.2.1 Hasil dan Analisa Pengujian Akurasi Pengukuran Jarak Laserping

Pengujian sensor laserping dilakukan dengan membandingkan hasil pengukuran laserping dengan jarak sebenarnya dari *mobile robot* ke objek untuk mengetahui seberapa akurat perhitungan laserping. Hasil pengujian ditampilkan pada Tabel 4.1.

Tabel 4.1 Hasil Pengujian Akurasi Pengukuran Jarak Laserping

No.	Jarak Sebenarnya (cm)	Hasil Pengukuran Laserping (cm)	Error (%)
1.	20	21	0,5
2.	50	51	2
3.	100	102	2
4.	200	199	0,5
5.	210	199	5,2
6.	220	199	9,54

Laserping merupakan sensor yang digunakan untuk mengukur jarak dengan cara memancarkan sinar laser dan mengukur waktu yang diperlukan sinar laser tersebut kembali setelah memantul dari objek yang mana panjang gelombang sinar laser tersebut adalah 850 nm yang berarti frekuensi gelombang sinar laser tersebut adalah $3,53 \times 10^{14}$ Hz. Ada beberapa kesalahan atau masalah yang dapat terjadi pada sensor laserping, beberapa di antaranya termasuk koneksi kabel, tegangan pasokan, pemrograman, pencahayaan yang buruk dan beberapa faktor lain.

Dari data pada Tabel 4.1 dapat diketahui bahwa laserping memiliki persentase error yang rendah dalam pengukuran jarak di bawah 200 cm namun saat mengukur jarak lebih dari 200 cm laserping memiliki persentase error lebih dari 2%. Ini menunjukkan bahwa laserping memiliki akurasi yang tinggi saat mengukur jarak objek di bawah 200 cm namun tidak mampu mengukur jarak objek di atas 200 cm dengan akurat. Tapi dengan semua itu laserping masih dapat bekerja dengan efisien dan mampu memberikan data yang akurat sehingga *mobile robot* dapat berhenti tepat waktu sebelum menabrak objek di depannya.

4.2.2 Hasil dan Analisa Pengujian Sistem Berhenti Otomatis *Mobile Robot*

Pengujian sistem berhenti otomatis *mobile robot* bertujuan untuk menganalisa apakah *mobile robot* dapat berhenti tepat di titik minimal jarak objek dari *mobile robot* atau *mobile robot* berhenti setelah melewati titik tersebut. Hasil pengujian ditampilkan pada Tabel 4.2.

Tabel. 4.2 Hasil Pengujian Sistem Berhenti Otomatis *Mobile Robot*

No.	Batas jarak <i>mobile robot</i> ke objek (titik henti <i>mobile robot</i> pada program) (cm)	Titik henti <i>mobile robot</i> dari objek saat pengujian (cm)		
		Percobaan 1	Percobaan 2	Percobaan 3
1.	10	0	0	0
2.	20	2	3	4
3.	40	23	24	22
4.	120	103	104	104

Dari data Tabel 4.2 dapat diketahui bahwa *mobile robot* melewati titik henti yang sudah diprogram sebelum *mobile robot* benar-benar berhenti. Setelah dianalisa ini bukan disebabkan oleh kesalahan program melainkan karena Motor DC yang menjalankan roda *mobile robot*. Saat *mobile robot* mencapai titik henti sistem secara otomatis menghentikan kerja dari Motor DC namun komutator Motor DC masih berputar saat itu sehingga *mobile robot* baru benar-benar berhenti setelah melewati titik henti beberapa sentimeter. Hal ini juga disebabkan karena *mobile robot* tidak memiliki perangkat yang dapat mengerem roda saat sistem Motor DC berhenti. Maka berdasarkan percobaan ini batas jarak minimal untuk program *mobile robot* adalah 20 cm.

BAB V

PENUTUP

5.1 Kesimpulan

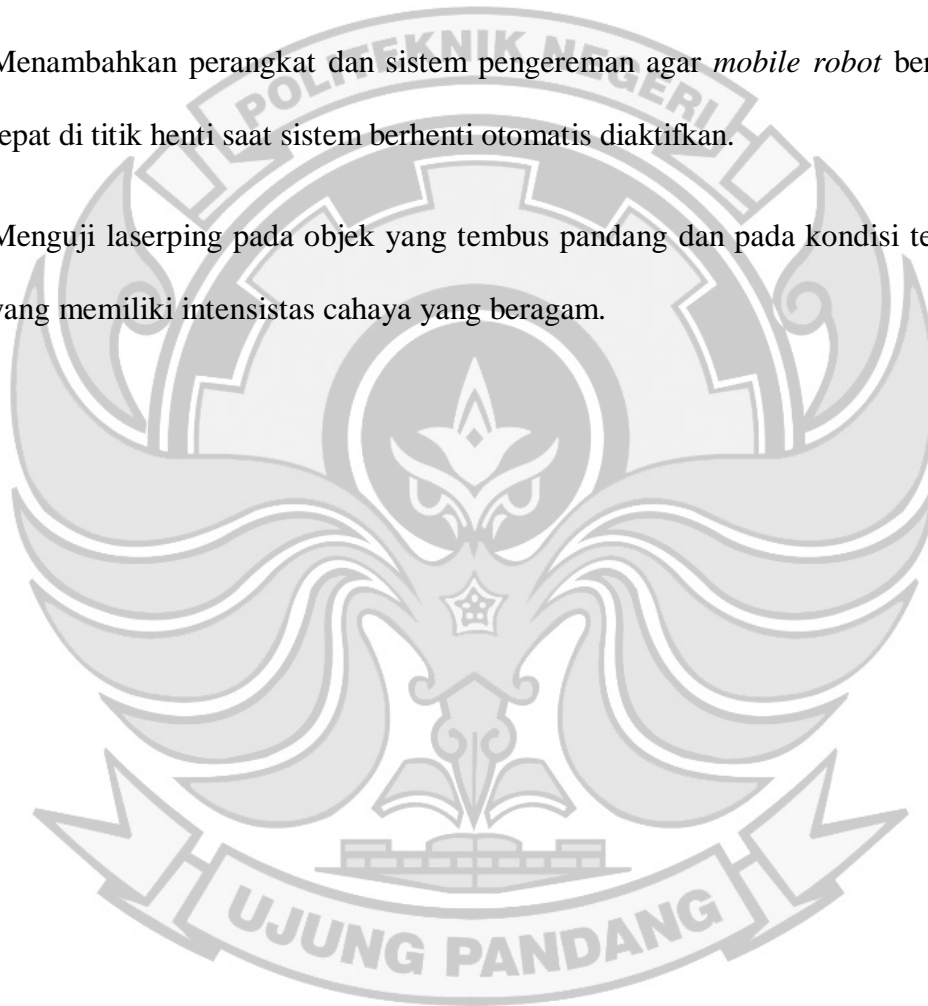
Perancangan sistem pendeteksi objek menggunakan sensor laserping pada *mobile robot* melibatkan penggunaan berbagai komponen perangkat keras dan perangkat lunak yang saling terintegrasi. Dalam penelitian ini, perangkat keras yang digunakan meliputi Arduino Uno sebagai pusat kendali, sensor laserping 2m range finder untuk deteksi objek, motor servo SG90 untuk mengarahkan sensor, motor DC untuk menggerakkan roda, driver motor L298N sebagai pengendali motor, dan LCD untuk menampilkan hasil pengukuran. Sensor laserping bekerja dengan memancarkan sinar laser dan mengukur waktu yang diperlukan untuk sinar tersebut memantul kembali setelah mengenai objek, memungkinkan penghitungan jarak yang akurat hingga 200 cm dengan tingkat kesalahan yang rendah. Sistem dirancang agar data dari sensor laserping dibaca oleh Arduino Uno, yang kemudian mengirimkan perintah kepada komponen lain untuk bertindak berdasarkan data yang diterima. Software yang digunakan untuk mengendalikan seluruh sistem ini adalah Arduino IDE dengan beberapa library pendukung seperti AFMotor.h, I2C LCD, dan Servo.h. Dengan diagram alir yang jelas, software ini memungkinkan *robot* untuk bergerak maju, berhenti secara otomatis saat mendeteksi objek dalam jarak yang ditentukan, dan berbelok ke arah yang aman jika ada halangan di depannya.

Implementasi sistem pendeteksi objek menggunakan sensor laserping pada *mobile robot* dilakukan melalui serangkaian pengujian untuk memastikan bahwa sistem bekerja dengan baik dalam berbagai kondisi. Pengujian pertama difokuskan pada akurasi sensor laserping dalam mendeteksi jarak objek. Hasil pengujian menunjukkan bahwa sensor laserping memiliki akurasi yang tinggi dengan tingkat kesalahan yang rendah untuk jarak di bawah 200 cm, namun ketepatan pengukuran menurun untuk jarak lebih dari 200 cm. Ini menunjukkan bahwa sistem ini ideal untuk penggunaan dalam lingkungan terbatas di mana jarak objek relatif dekat. Pengujian berikutnya melibatkan sistem berhenti otomatis pada *mobile robot*, yang bertujuan untuk memastikan bahwa *mobile robot* dapat berhenti pada jarak yang telah diprogram dari objek. Hasil pengujian menunjukkan bahwa *mobile robot* sering melewati titik berhenti yang telah ditentukan karena inersia komutator motor DC yang masih berputar meskipun sistem telah menghentikan motornya. Hal ini mengindikasikan bahwa meskipun sistem deteksi bekerja dengan baik, ada kebutuhan untuk peningkatan pada sistem mekanik, seperti menambahkan mekanisme pengereman, agar *mobile robot* dapat berhenti dengan lebih tepat. Implementasi ini membuktikan bahwa sensor laserping dapat digunakan secara efektif untuk mendeteksi objek dan menghindari tabrakan, meskipun masih ada ruang untuk perbaikan lebih lanjut.

5.2 Saran

Demikian laporan tugas akhir “Rancang Bangun Sistem Pendeteksi Objek Berbasis Laser Pada *Mobile Robot*” yang telah dibuat. Penelitian ini dapat dikembangkan lebih lanjut dengan melakukan hal-hal berikut:

1. Menambahkan perangkat dan sistem pengereman agar *mobile robot* berhenti tepat di titik henti saat sistem berhenti otomatis diaktifkan.
2. Menguji laserping pada objek yang tembus pandang dan pada kondisi tempat yang memiliki intensitas cahaya yang beragam.



DAFTAR PUSTAKA

- Ariawan, K. U. 2019. Design of Hexapod Robot Movement based on Arduino Mega 2560. *Journal of Physics: Conference Series*.
- Fikri, Muhammad. Rivai, Muhammad. 2019. *Sistem Penghindar Halangan Dengan Metode LIDAR Pada Unmanned Surface Vehicle*. Departemen Teknik Elektro Institut Teknologi Sepuluh Nopember (ITS).
- Ilham Efendi. 2014. Pengertian dan Kelebihan Arduino. *IT Jurnal*.
- Jazar, R. N. 2020. Inverse Kinematics. *Journal Robotics*.
- Julianto, D. 2017. *Media Pembelajaran Trainer Motor DC, Brushless Servo, dan Stepper Dengan Kendali Mikrokontroler Arduino Uno Pada Mata Pelajaran Teknik Mikroprosesor di SMK Negeri 2 Depok Yogyakarta*. Universitas Negeri Yogyakarta.
- Khorassani, S. M. H., dkk. 2015. Arduino Documentation. *Journal Electronic Engineering*.
- Louis, L. 2016. Working Principle of Arduino and Using it as a Tool for Study and Research, *International Journal of Control, Automation, Communication, and System*.
- Laserping Rangefinder Module. (Online), (<https://www.parallax.com>, diakses 17 Mei 2023)
- Putra, I Wayan Krisna Eka Putra. 2016. *Sistem Kerja Laser pada LIDAR*. Jurusan Pendidikan Geografi, FHIS, UNDIKSHA.
- Salim, A. I. 2020. Implementasi Motor Servo SG 90 Sebagai Penggerak Mekanik Pada E. I. Helper (ELECTRONICS INTEGRATION HELMET WIPER). *Ectro Luceat*.

LAMPIRAN

Lampiran 1 Program Sistem *Mobile Robot* menggunakan Arduino IDE



```
FIX - last.ino | Arduino IDE 2.3.0
File Edit Sketch Tools Help

Select Board

FIX.ino last2.ino 120cm.ino 40cm.ino last.ino
1 // #include "AFMotor.h"
2 #include <Servo.h>
3
4 #define pingPin 7
5 #include <LiquidCrystal_I2C.h>
6 LiquidCrystal_I2C lcd(0x27,16,2);
7
8 const int SCLPin = A5;
9 const int SDAPin = A4;
10
11 Servo myservo;
12
13 const int motor1 = 5;
14 const int motor2 = 4;
15 const int motor3 = 8;
16 const int motor4 = 6;
17 const int pinSpeed = 9;
18 // inialisasi variabel penampung nilai kecepatan
19 int Speed,cm;
20

Ln 7, Col 1 | No board selected
21 //=====
22
23 int distance_L, distance_F, distance_R;
24 long distance,duration;
25
26 int set = 20;
27
28 void setup() {
29     Serial.begin(9600); // Initialize serial port
30     Serial.println("Start");
31
32     myservo.attach(10);
33     myservo.write(90);
34
35
36     pinMode(motor1, OUTPUT);
37     pinMode(motor2, OUTPUT);
38     pinMode(motor3, OUTPUT);
39     pinMode(motor4, OUTPUT);
40     //=====
```

```
FIX - lastino | Arduino IDE 2.3.0
File Edit Sketch Tools Help

Select Board

FIX.ino last2.ino 120cm.ino 40cm.ino last.ino

41 lcd.init(); // initialize the lcd
42 lcd.init();
43 // initialize
44 lcd.backlight();
45 //delay (1000);
46 lcd.clear();
47 lcd.setCursor(1,0);
48 lcd.print("Pendeteksi");
49 lcd.setCursor(1,1);
50 lcd.print("Objek");
51 delay (1000);
52 lcd.clear();
53 lcd.setCursor(1,0);
54 lcd.print("Berbasis");
55 lcd.setCursor(1,1);
56 lcd.print("Laser");
57 delay (1000);
58 lcd.clear();
59 lcd.setCursor(1,0);
60 lcd.print("Pada Mobile");

Ln 7, Col 1 x No board selected
21:37
14/01/2023
```

```
FIX - lastino | Arduino IDE 2.3.0
File Edit Sketch Tools Help

Select Board

FIX.ino last2.ino 120cm.ino 40cm.ino last.ino

61 lcd.setCursor(1,1);
62 lcd.print("Robot");
63 delay (1000);
64
65 }
66 //-----
67 // Main
68 //-----
69 void loop() {
70   Speed = constrain(Speed, 0,255);
71   // inialisasi nilai kecepatan
72   // nilai kecepatan dapat diubah, batasannya antara 0 - 255
73   Speed = 50
74   ;
75   // menuliskan nilai speed pada pinSpesed
76   analogWrite(pinSpeed, Speed);
77   //-----
78   if(cm >= 20 & cm <199){
79     lcd.clear();
80     lcd.setCursor(1,0);

Ln 7, Col 1 x No board selected
21:37
14/01/2023
```

```
FIX - lastino | Arduino IDE 2.3.0
File Edit Sketch Tools Help

Select Board

FIX.ino last2.ino 120cm.ino 40cm.ino last.ino
81 lcd.print( "Distance: " );
82 lcd.print(cm);
83 lcd.print("cm");
84 lcd.setCursor(1,1);
85 lcd.print( "IN RANGE" );
86 delay(1000);
87 }
88 if(cm >= 200){
89   lcd.clear();
90   lcd.setCursor(1,0);
91   lcd.print( "Distance: " );
92   lcd.print(cm);
93   lcd.print("cm");
94   lcd.setCursor(1,1);
95   lcd.print( "OUT OF RANGE" );
96   delay(1000);
97 }
98
99 //-----
100 distance_F = data();
```

Ln 7, Col 1 x No board selected
21:37 14/01/2023

```
FIX - lastino | Arduino IDE 2.3.0
File Edit Sketch Tools Help

Select Board

FIX.ino last2.ino 120cm.ino 40cm.ino last.ino
101 Serial.print("S=");
102 Serial.println(distance_F);
103 if (distance_F > set){
104   digitalWrite(motor1, HIGH);
105   digitalWrite(motor2, LOW);
106   digitalWrite(motor3, HIGH);
107   digitalWrite(motor4, LOW);
108   Serial.println("Forward");
109 }
110 else{
111   cm= data();
112   lcd.clear();
113   lcd.setCursor(1,0);
114   lcd.print( "Distance: " );
115   lcd.print(cm);
116   lcd.print("cm");
117   lcd.setCursor(1,1);
118   lcd.print( "BELOW OF RANGE" );
119   delay(1000);
120   hc_sr4();
```

Ln 7, Col 1 x No board selected
21:37 14/01/2023

```
FIX - last.ino | Arduino IDE 2.3.0
File Edit Sketch Tools Help

Select Board

FIX.ino last2.ino 120cm.ino 40cm.ino last.ino
121 |   }
122 | }
123 |
124 |
125 | long data(){
126 |   long distance;
127 |   digitalWrite(pingPin, LOW);
128 |   delayMicroseconds(2);
129 |   digitalWrite(pingPin, HIGH);
130 |   delayMicroseconds(5);
131 |   digitalWrite(pingPin, LOW);
132 |   pinMode(pingPin, INPUT);
133 |   duration = pulseIn(pingPin, HIGH);
134 |   // cm = microsecondsToCentimeters(duration);
135 |   distance = duration / 29 / 2 - 7;
136 |   return distance;
137 | }
138 |
139 |
140 |
```

Ln 7, Col 1 X No board selected 21:38 14/01/2023

```
FIX - last.ino | Arduino IDE 2.3.0
File Edit Sketch Tools Help

Select Board

FIX.ino last2.ino 120cm.ino 40cm.ino last.ino
141 | void compareDistance(){
142 |   if (distance_L > distance_R){
143 |     kiri();
144 |     delay(350);
145 |   }
146 |   else if (distance_R > distance_L){
147 |     kanan();
148 |     delay(390);
149 |   }
150 |   else{
151 |     scan();
152 |     delay(390);
153 |     kiri();
154 |     delay(500);
155 |   }
156 | }
157 |
158 | void hc_sr4(){
159 |   Serial.println("Stop");
160 |   scan();//stop
```

Ln 7, Col 1 X No board selected 21:38 14/01/2023

```
FIX - last.ino | Arduino IDE 2.3.0
File Edit Sketch Tools Help

Select Board

FIX.ino last2.ino 120cm.ino 40cm.ino last.ino
161 myservo.write(0);
162 delay(300);
163 distance_R = data();
164 delay(100);
165 myservo.write(170);
166 delay(500);
167 distance_L = data();
168 delay(100);
169 myservo.write(90);
170 delay(300);
171 compareDistance();
172 }
173
174 //-----
175 void maju(){
176   digitalWrite(motor1, HIGH);
177   digitalWrite(motor2, LOW);
178   digitalWrite(motor3, HIGH);
179   digitalWrite(motor4, LOW);
180 }
```

Ln 7, Col 1 No board selected

21:38 14/01/2025

```
FIX - last.ino | Arduino IDE 2.3.0
File Edit Sketch Tools Help

Select Board

FIX.ino last2.ino 120cm.ino 40cm.ino last.ino
181 void mundur(){
182   digitalWrite(motor1, LOW);
183   digitalWrite(motor2, HIGH);
184   digitalWrite(motor3, LOW);
185   digitalWrite(motor4, HIGH);
186 }
187 void kiri(){
188   digitalWrite(motor1, LOW);
189   digitalWrite(motor2, HIGH);
190   digitalWrite(motor3, HIGH);
191   digitalWrite(motor4, LOW);
192 }
193 void kanan(){
194   digitalWrite(motor1, HIGH);
195   digitalWrite(motor2, LOW);
196   digitalWrite(motor3, LOW);
197   digitalWrite(motor4, HIGH);
198 }
199
200 void scan(){
```

Ln 7, Col 1 No board selected

21:38 14/01/2025

```
FIX - last.ino | Arduino IDE 2.3.0
File Edit Sketch Tools Help

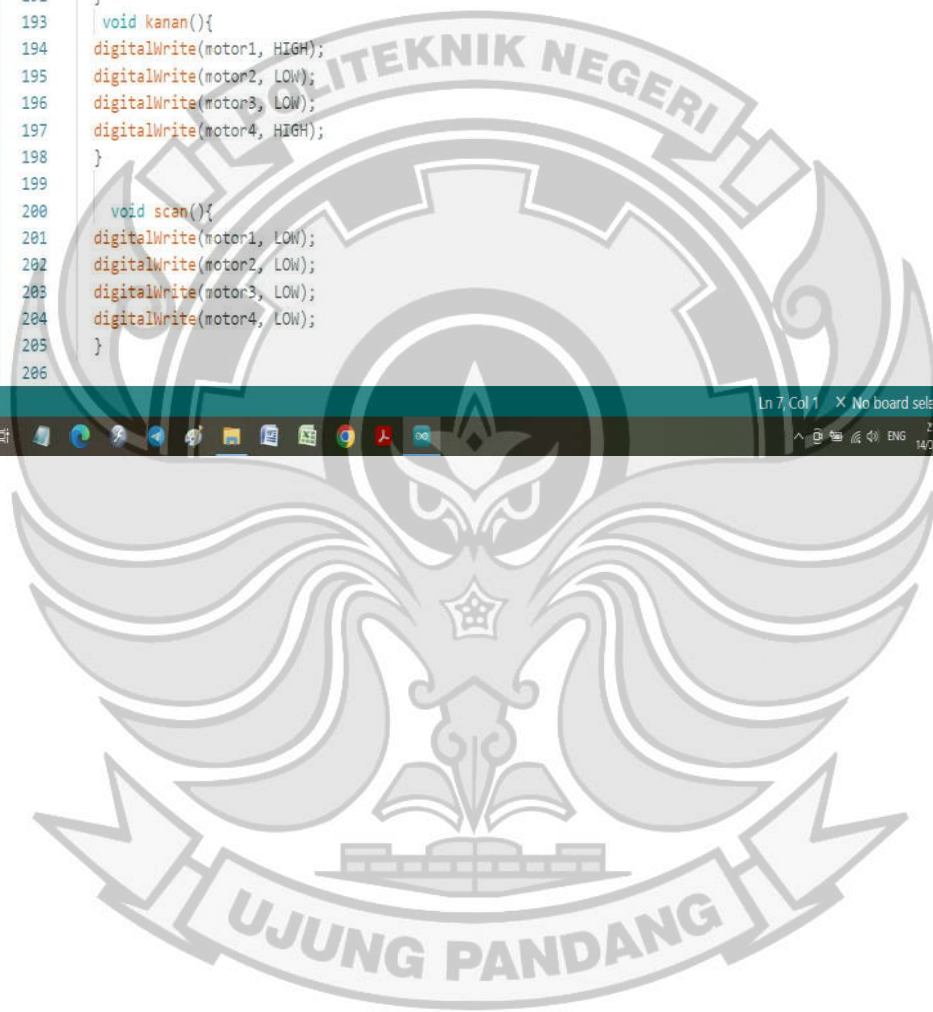
Select Board

FIX.ino last2.ino 120cm.ino 40cm.ino last.ino

187 void kiri(){
188   digitalWrite(motor1, LOW);
189   digitalWrite(motor2, HIGH);
190   digitalWrite(motor3, HIGH);
191   digitalWrite(motor4, LOW);
192 }
193 void kanan(){
194   digitalWrite(motor1, HIGH);
195   digitalWrite(motor2, LOW);
196   digitalWrite(motor3, LOW);
197   digitalWrite(motor4, HIGH);
198 }
199
200 void scan(){
201   digitalWrite(motor1, LOW);
202   digitalWrite(motor2, LOW);
203   digitalWrite(motor3, LOW);
204   digitalWrite(motor4, LOW);
205 }
206
```

Ln 7, Col 1 x No board selected

21:38
14/11/2025



Lampiran 2 Pengujian Sistem Berhenti Otomatis Pada *Mobile Robot*



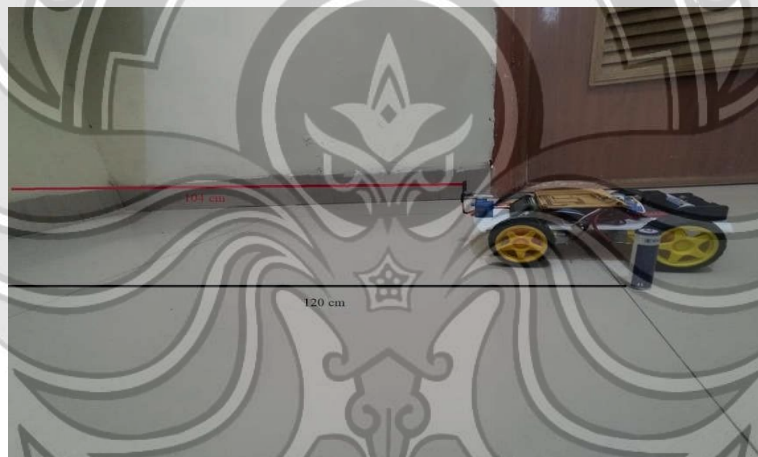
Pengujian Sistem Berhenti Otomatis dengan Batas Jarak Objek 10 cm dari *Mobile Robot*



Pengujian Sistem Berhenti Otomatis dengan Batas Jarak Objek 20 cm dari *Mobile Robot*



Pengujian Sistem Berhenti Otomatis dengan Batas Jarak Objek 40 cm dari *Mobile Robot*



Pengujian Sistem Berhenti Otomatis dengan Batas Jarak Objek 120 cm dari *Mobile Robot*

Lampiran 3 Datasheet Arduino Uno

1 The Board

1.1 Application Examples

The UNO board is the flagship product of Arduino. Regardless if you are new to the world of electronics or will use the UNO R3 as a tool for education purposes or industry-related tasks, the UNO R3 is likely to meet your needs.

First entry to electronics: If this is your first project within coding and electronics, get started with our most used and documented board: UNO. It is equipped with the well-known ATmega328P processor, 14 digital input/output pins, 6 analog inputs, USB connections, ICSP header and reset button. This board includes everything you will need for a great first experience with Arduino.

Industry-standard development board: Using the UNO R3 board in industries, there are a range of companies using the UNO R3 board as the brain for their PLCs.

Education purposes: Although the UNO R3 board has been with us for about ten years, it is still widely used for various education purposes and scientific projects. The board's high standard and top quality performance makes it a great resource to capture real time from sensors and to trigger complex laboratory equipment to mention a few examples.

1.2 Related Products

- Arduino Starter Kit
- Arduino UNO R4 Minima
- Arduino UNO R4 WiFi
- Tinkerkit Braccio Robot

2 Ratings

2.1 Recommended Operating Conditions

Symbol	Description	Min	Max
	Conservative thermal limits for the whole board:	-40 °C (-40 °F)	85 °C (185 °F)

NOTE: In extreme temperatures, EEPROM, voltage regulator, and the crystal oscillator, might not work as expected.

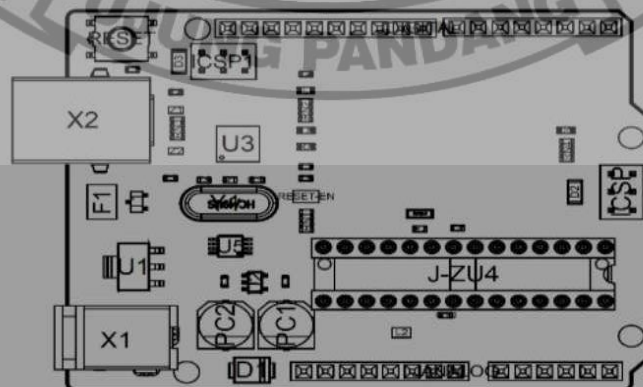
2.2 Power Consumption

Symbol	Description	Min	Typ	Max	Unit
VINMax	Maximum input voltage from VIN pad	6	-	20	V
VUSBMax	Maximum input voltage from USB connector	-	-	5.5	V
PMax	Maximum Power Consumption	-	-	xx	mA

3 Functional Overview

3.1 Board Topology

Top view



Board topology

Ref.	Description	Ref.	Description
X1	Power jack 2.1x5.5mm	U1	SPX1117M3-L-5 Regulator
X2	USB B Connector	U3	ATMEGA16U2 Module
PC1	EEE-1EA470WP 25V SMD Capacitor	U5	LMV358LIST-A.9 IC
PC2	EEE-1EA470WP 25V SMD Capacitor	F1	Chip Capacitor, High Density
D1	CGRA4007-G Rectifier	ICSP	Pin header connector (through hole 6)
J-ZU4	ATMEGA328P Module	ICSP1	Pin header connector (through hole 6)
Y1	ECS-160-20-4X-DU Oscillator		

3.2 Processor

The Main Processor is a ATmega328P running at up to 20 MHz. Most of its pins are connected to the external headers, however some are reserved for internal communication with the USB Bridge coprocessor.



Lampiran 4 Datasheet Laserping

LaserPING Rangefinder Module (#28041)

The LaserPING 2m Rangefinder provides an easy method of distance measurement. This near-infrared, time-of-flight (TOF) sensor is ideal for taking measurements between moving or stationary objects.

A single I/O pin is used to both query the LaserPING sensor for its latest distance measurement, and to read the reply.

The LaserPING 2m Rangefinder can be used with nearly any microcontroller, utilizing its PWM mode or optional serial mode. It is designed to be circuit- and code-compatible with the PING))) Ultrasonic Distance Sensor, making applications adaptable where different environmental conditions need to be considered. Measurements can even be taken through an acrylic window to protect the sensor.

The sensor's built-in co-processor ensures the right logic levels. Its I/O connections operate at the same voltage supplied to the VIN pin, for compatibility with 3.3V and 5V microcontrollers.

Features

- Non-contact distance measurement with a 2–200 cm range
- Factory pre-calibrated for accuracy with 1 mm resolution
- Eye-safe invisible near-infrared (IR) illumination using a class 1 laser emitter
- Reverse polarity protection if VIN and GND are accidentally swapped
- Onboard microprocessor handles complex sensor code
- Compatible with 3.3V and 5V microcontrollers
- Breadboard-friendly 3-pin SIP form-factor with mounting hole

Application Ideas

- Physics studies
- Security systems
- Interactive animated exhibits
- Robotics navigation and parking assistant systems
- Interactive applications such as hand detection and 1D gesture recognition
- Volume or height detection in process control systems

Key Specifications

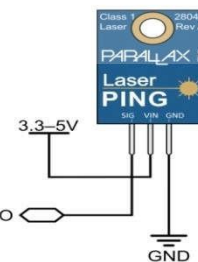
- Laser: 850 nm VCSEL (Vertical Cavity Surface Emitting Laser)
- Range: 2–200 cm
- Resolution: 1 mm
- Typical refresh rate: 15 Hz PWM mode, 22 Hz serial mode
- Power requirements: +3.3V DC to +5 VDC; 25 mA
- Communication: PWM (idle low) or serial 9600 baud (idle high); logic level = VIN
- Operating temperature: +14 to +140 °F (-10 to +60 °C)
- Laser eye safety: near-infrared Class 1 laser product
- Field of illumination: 23° degrees
- Field of view: 55° degrees
- Form factor: 3-pin male header with 0.1" spacing
- PCB dimensions: 22 x 16 mm

Getting Started

Connect the LaserPING sensor's pins to power, ground, and your microcontroller's I/O pin as shown in the diagram. Note that the diagram shows the back of the sensor; point the component side toward your target object.

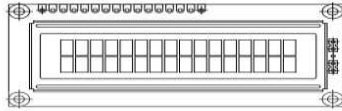
The LaserPING sensor is supported by BlocklyProp blocks, Propeller C libraries, and example code for the BASIC Stamp and Arduino Uno. It is circuit- and code-compatible with applications for the PING))) Ultrasonic Distance Sensor (#28015).

Look for downloads and tutorial links on the sensor's product page; search "28041" at www.parallax.com.



Lampiran 5 Datasheet *Liquid Crystal Display (LCD)*

16 x 2 Character LCD



FEATURES

- Type: Character
- Display format: 16 x 2 characters
- Built-in controller: ST 7066 (or equivalent)
- Duty cycle: 1/16
- 5 x 8 dots includes cursor
- + 5 V power supply
- LED can be driven by pin 1, pin 2, or A and K
- N.V. optional for + 3 V power supply
- Optional: Smaller character size (2.95 mm x 4.35 mm)
- Material categorization: For definitions of compliance please see www.vishay.com/doc?99912



MECHANICAL DATA		
ITEM	STANDARD VALUE	UNIT
Module Dimension	80.0 x 36.0 x 13.2 (max.)	
Viewing Area	66.0 x 16.0	
Dot Size	0.55 x 0.65	mm
Dot Pitch	0.60 x 0.70	
Mounting Hole	75.0 x 31.0	
Character Size	2.95 x 5.55	

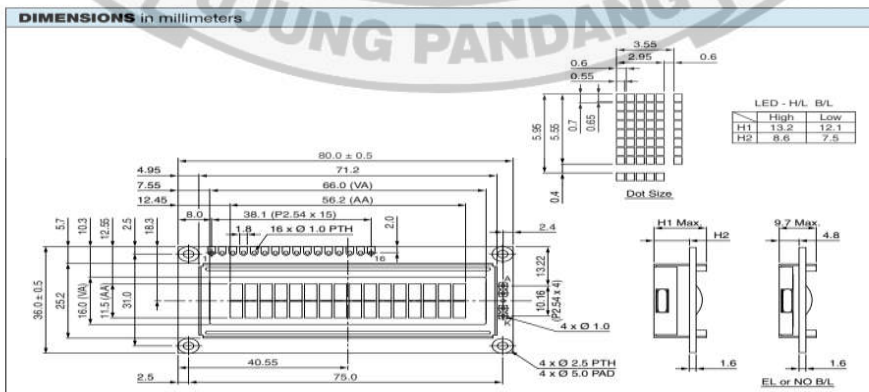
ABSOLUTE MAXIMUM RATINGS					
ITEM	SYMBOL	STANDARD VALUE			UNIT
		MIN.	TYP.	MAX.	
Power Supply	V_{DD} to V_{SS}	- 0.3	-	13	V
Input Voltage	V_I	V_{SS}	-	V_{DD}	

Note
 • $V_{SS} = 0$ V, $V_{DD} = 5.0$ V

ELECTRICAL CHARACTERISTICS						
ITEM	SYMBOL	CONDITION	STANDARD VALUE			UNIT
			MIN.	TYP.	MAX.	
Input Voltage	V_{DD}	$V_{DD} = +5$ V	4.5	5.0	5.5	V
Supply Current	I_{DD}	$V_{DD} = +5$ V - 20 °C	1.0	1.2	1.5	mA
Recommended LC Driving Voltage for Normal Temperature Version Module	V_{DD} to V_0	0 °C	-	-	-	
		25 °C	-	3.7	-	V
		50 °C	-	-	-	
		70 °C	3.1	-	-	
LED Forward Voltage	V_F	25 °C	-	4.2	4.6	V
LED Forward Current - Array	I_F	25 °C	-	100	-	mA
LED Forward Current - Edge			-	20	40	mA
EL Power Supply Current	I_{EL}	$V_{EL} = 110$ V _{AC} ; 400 Hz	-	-	5.0	mA

DISPLAY CHARACTER ADDRESS CODE																
Display Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DD RAM Address	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
DD RAM Address	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F

INTERFACE PIN FUNCTION		
PIN NO.	SYMBOL	FUNCTION
1	V_{SS}	Ground
2	V_{DD}	Supply voltage for logic
3	V_0	Operating voltage for LCD
4	RS	H: Data/L: Instruction code
5	R/W	H: Read (MPU → Module)/L: Write (MPU → Module)
6	E	H → L chip enable signal
7	DB0	Data bus line
8	DB1	Data bus line
9	DB2	Data bus line
10	DB3	Data bus line
11	DB4	Data bus line
12	DB5	Data bus line
13	DB6	Data bus line
14	DB7	Data bus line
15	A	Supply power for LED+
16	R	Supply power for Red-
17	G	Supply power for Green-
18	B	Supply power for Blue-



Lampiran 6 Datasheet *Inter-Integrated Circuit (I2C)*

I2C Serial Interface 1602 LCD Module

This is I2C interface 16x2 LCD display module, a high-quality 2 line 16 character LCD module with on-board contrast control adjustment, backlight and I2C communication interface. For Arduino beginners, no more cumbersome and complex LCD driver circuit connection. The real significance advantages of this I2C Serial LCD module will simplify the circuit connection, save some I/O pins on Arduino board, simplified firmware development with widely available Arduino library.



SKU: [DSP-1182](#)

Brief Data:

- Compatible with Arduino Board or other controller board with I2C bus.
- Display Type: Negative white on Blue backlight.
- I2C Address: 0x38-0x3F (0x3F default)
- Supply voltage: 5V
- Interface: I2C to 4bits LCD data and control lines.
- Contrast Adjustment: built-in Potentiometer.
- Backlight Control: Firmware or jumper wire.
- Board Size: 80x36 mm.

Setting Up:

Hitachi's HD44780 based character LCD are very cheap and widely available, and is an essential part for any project that displays information. Using the LCD piggy-back board, desired data can be displayed on the LCD through the I2C bus. In principle, such backpacks are built around PCF8574 (from NXP) which is a general purpose bidirectional 8 bit I/O port expander that uses the I2C protocol. The PCF8574 is a silicon CMOS circuit provides general purpose remote I/O expansion (an 8-bit quasi-bidirectional) for most microcontroller families via the two-line bidirectional bus (I2C-bus). Note that most piggy-back modules are centered around PCF8574T (SO16 package of PCF8574 in DIP16 package) with a default slave address of 0x27. If your piggy-back board holds a PCF8574AT chip, then the default slave address will change to 0x3F. In short, if the piggy-back board is based on PCF8574T and the address connections (A0-A1-A2) are not bridged with solder it will have the slave address 0x27.



Lampiran 7 Datasheet Motor Servo SG90

SERVO MOTOR SG90
DATA SHEET



Tiny and lightweight with high output power. Servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds but smaller. You can use any servo code, hardware or library to control these servos. Good for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. It comes with a 3 horns (arms) and hardware.



Dimensions & Specifications	
A (mm) :	32
B (mm) :	23
C (mm) :	28.5
D (mm) :	12
E (mm) :	32
F (mm) :	19.5
Speed (sec) :	0.1
Torque (kg-cm) :	2.5
Weight (g) :	14.7
Voltage :	4.8 - 6

Position "0" (1.5 ms pulse) is middle, "90" (~2ms pulse) is middle, is all the way to the right, "-90" (~1ms pulse) is all the way to the left.

PWM = Orange ()

Vcc = Red (+)

Ground = Brown (-)





1 - 2 ms
Duty Cycle

4.8 V (~5 V)
Power
and Signal

20 ms (50 Hz)
PWM Period

Lampiran 9 Datasheet Driver Motor L298N

L298N Dual H-Bridge Motor Driver

This dual bidirectional motor driver, is based on the very popular L298 Dual H-Bridge Motor Driver Integrated Circuit. The circuit will allow you to easily and independently control two motors up to 2A each in both directions. It is ideal for robotic applications and well suited for connection to a microcontroller requiring just a couple of control lines per motor. It can also be interfaced with simple manual switches, TTL logic gates, relays, etc. This board equipped with power LED indicators, on-board +5V regulator and protection diodes.

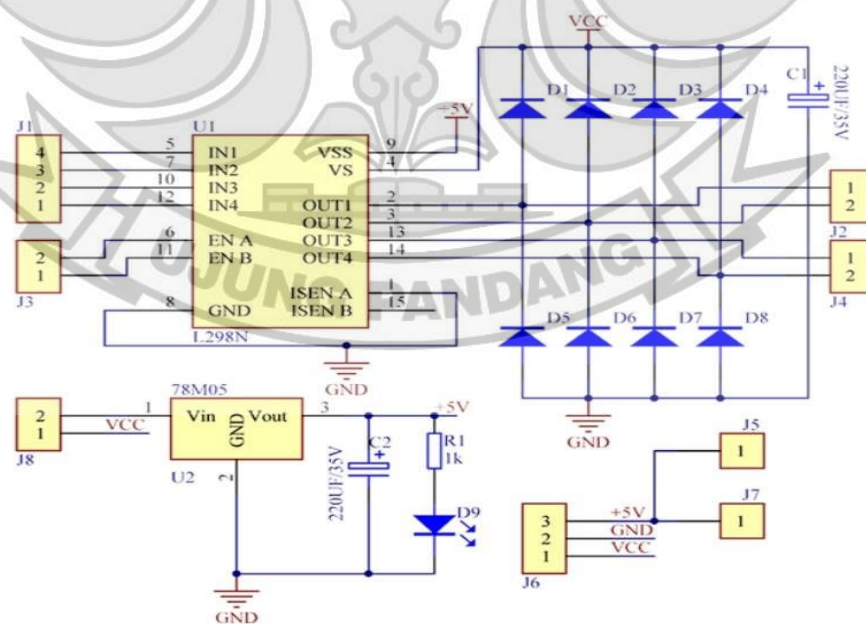


SKU: MDU-1049

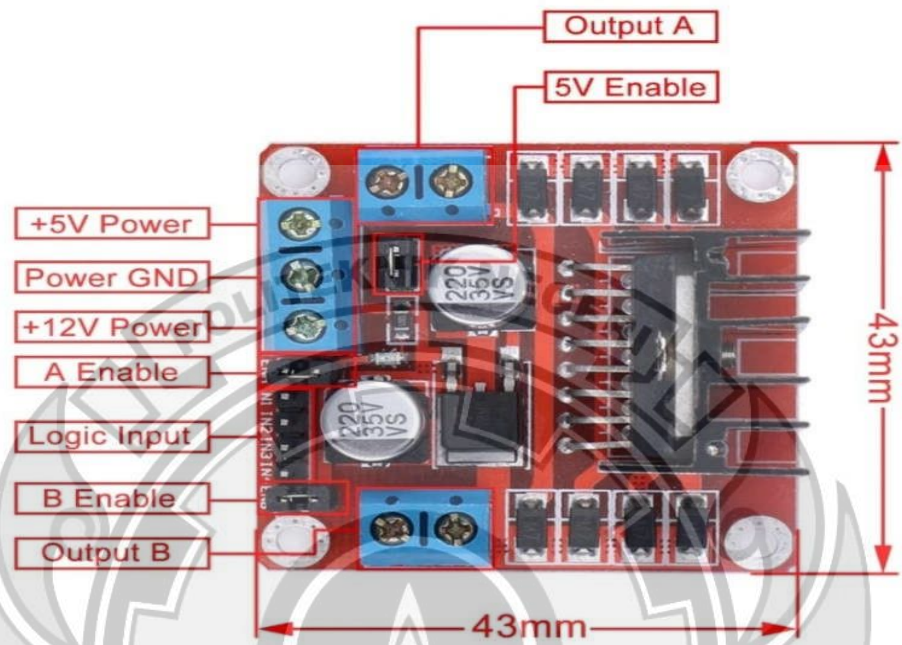
Brief Data:

- Input Voltage: 3.2V~40Vdc.
- Driver: L298N Dual H Bridge DC Motor Driver
- Power Supply: DC 5 V - 35 V
- Peak current: 2 Amp
- Operating current range: 0 ~ 36mA
- Control signal input voltage range :
 - Low: $-0.3V \leq V_{in} \leq 1.5V$.
 - High: $2.3V \leq V_{in} \leq V_{ss}$.
- Enable signal input voltage range :
 - Low: $-0.3 \leq V_{in} \leq 1.5V$ (control signal is invalid).
 - High: $2.3V \leq V_{in} \leq V_{ss}$ (control signal active).
- Maximum power consumption: 20W (when the temperature $T = 75^\circ C$).
- Storage temperature: $-25^\circ C \sim +130^\circ C$.
- On-board +5V regulated Output supply (supply to controller board i.e. Arduino).
- Size: 3.4cm x 4.3cm x 2.7cm

Schematic Diagram:



Board Dimension & Pins Function:



Connection Examples:

Controlling 2-DC Motor with +5V Arduino onboard Power Supply:

Below is the circuit connection use the on-board +5V power supply from Arduino board, and should be done without the 5V Enable Jumper on (Active 5V). This connection can drive two 5V DC motors simultaneously.

