

**RANCANG BANGUN
ALAT UKUR KONSTANTA PEGAS
MENGUNAKAN ARDUINO UNO
DAN SENSOR ULTRASONIK HC-SR04**

Tugas Akhir

**Diajukan untuk Melengkapi Tugas-tugas
dan Syarat-syarat untuk Mencapai
Gelar Sarjana Teknik**

Oleh :

**ARJUNA TOGATOROP
12.812.0009**



**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS MEDAN AREA
MEDAN
2017**

**RANCANG BANGUN
ALAT UKUR KONSTANTA PEGAS
MENGUNAKAN ARDUINO UNO
DAN SENSOR ULTRASONIK HC-SR04**

SKRIPSI

Oleh :

ARJUNA TOGATOROP
12.812.0009

Disetujui Oleh
Komisi Pembimbing



(Ir. H. Yance Sarif MT)

Pembimbing I



(Faisal Irsan Pasaribu ST, MT)

Pembimbing II



(Prof. Dr. Dadan Ramdan, M.Eng. M.Sc)

Dekan



(Faisal Irsan Pasaribu ST, MT)

Kaprodi

LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : Arjuna Togatorop

Npm : 12.812.0009

Prodi : teknik elektro

Judul skripsi : rancang bangun alat ukur konstanta pegas menggunakan arduino uno dan sensor ultrasonik HC-SR04

Menyatakan dengan sebenarnya bahwa :

1. Skripsi yang saya ajukan adalah sebagai syarat untuk memperoleh gelar sarjana pada fakultas teknik Universitas Medan Area bebas dari plagiat, sesuai kaidah ilmiah, norma akademik dan norma hukum sesuai peraturan menteri pendidikan nasional nomor 17 tahun 2010 tentang penegakan dan penanggulangan plagiat di perguruan tinggi
2. Bilamana terbukti terdapat tindakan plagiat, maka bersedia menerima sanksi sesuai peraturan dan perundangan yang berlaku

Demikian surat pernyataan ini saya buat untuk dapat dipergunakan sebagaimana mestinya

Medan , Juli 2017

Yang memberikan pernyataan



(Arjuna Togatorop)

12.812.0009

ABSTRAK

Telah dibuat alat ukur konstanta pegas berbasis mikrokontroler ATmega 328 sistem minimum Arduino Uno dengan menggunakan sensor ultrasonik HC-SR04. Sistem ini dirancang agar dapat mendeteksi / mengukur panjang pegas dan menampilkan hasil pengukuran pada LCD 2x16 karakter. Sistem ini terdiri atas perangkat keras dan perangkat lunak. Perangkat keras terdiri dari sensor ultrasonik HC-SR04 yang akan memancarkan sinyal ultrasonik melalui transmitter dan ketika sinyal mengenai benda penghalang, maka sinyal ini dipantulkan, dan diterima oleh receiver ultrasonik. Sinyal yang diterima oleh rangkaian receiver dikirimkan ke rangkaian Arduino Uno untuk selanjutnya diolah untuk menghitung konstanta pegas, dan LCD 2x16 karakter sebagai penampil. Perangkat lunak yang digunakan untuk menjalankan atau memproses data pada mikrokontroler ATmega 328 ini adalah bahasa C.

Kata Kunci : pengukur konstanta pegas, Arduino Uno, Sensor Ultrasonik

ABSTRACT

Maternal-based standard for ATmega 328 Arduino Uno microcontroller system has been developed using ultrasonic sensors HC-SR04. This system is designed to be able to detect / measure the length of a spring and display the measurement results on a 2x16-character LCD. This system consists of hardware and software. The hardware consists of ultrasonic HC-SR04 sensors that will emit ultrasonic signals through the transmitter and when the signal is about a barrier object, this signal is reflected, and received by the ultrasonic receiver. The signal received by the receiver circuit is sent to the Arduino Uno circuit for further processing to calculate the spring constant, and the 2x16 character LCD as the viewer. The software used to run or process data on this microcontroller ATmega 328 is C language.

Keywords: spring constant gauge, Arduino Uno, Ultrasonic Sensor

KATA PENGANTAR

Puji dan syukur penulis ucapkan kehadiran Tuhan Yang Maha Esa atas berkat dan karunia yang dilimpahkan-Nya kepada penulis sehingga akhirnya dapat menyelesaikan pendidikan di Fakultas Teknik Universitas Medan Area untuk mendapatkan gelar sarjana saya. adapun judul skripsi yang saya buat adalah **“RANCANG BANGUN ALAT UKUR KONSTANTA PEGAS MENGGUNAKAN ARDUINO UNO DAN SENSOR ULTRASONIK HC-SR04”**

Pada kesempatan kali ini penulis sangat berterima kasih kepada berbagai pihak yang sudah membantu penulis dalam menyelesaikan pendidikan selama di Universitas Medan Area untuk mendapatkan gelar sarjana saya. penulis mengucapkan terima kasih kepada :

1. Bapak Prof.Dr.H.A.Ya'kub Matondang, MA selaku rektor Universitas Medan Area
2. Bapak Prof.Dr.Dadan Ramdan Meng,MSc selaku dekan fakultas teknik Universitas Medan Area
3. Bapak Faisal Irsan Pasaribu ST, MT selaku kaprodi teknik elektro sekaligus sebagai dosen pembimbing II
4. Bapak Ir. H.Yance Syarif,MT selaku dosen pembimbing I
5. Semua bapak dan ibu dosen teknik elektro Universitas Medan Area yang sudah banyak membantu saya dalam mendapatkan ilmu dan gelar sarjana saya yang nantinya bisa saya gunakan bagi diri saya sendiri dan juga bagi nusa dan bangsa.
6. Kedua orangtua saya dan semua keluarga yang selama ini sudah memberikan dukungan dan kepercayaan kepada saya untuk mendapatkan gelar Sarjana Teknik
7. Seluruh mahasiswa/i fakultas teknik terutama teknik elektro Universitas Medan Area yang selama ini menjadi saudara sekaligus teman untuk bertukar pikiran sehingga saya dapat menyelesaikan gelar sarjana

Atas bantuan dan petunjuk dari mereka,secara langsung maupun tidak langsung ,penulis hanya bisa berdoa semoga Tuhan Yang Maha Esa membalas semua kebajikannya.

Dalam penulisan tugas akhir ini, penulis menyadari masih banyak terdapat kekurangan dan kelemahan baik dalam penyajian maupun dalam penulisan, untuk itu penulis mengrapkan saran yang bersifat membangun untuk kesempurnaan skripsi ini.

Medan , Juli 2017

Hormat saya

Arjuna Togatorop



DAFTAR ISI

Lembar pernyataan.....	i
Kata pengantar.....	ii
Abstrak.....	iv
ABSTRACT.....	v
BAB I PENDAHULUAN	
1.1.Latar Belakang.....	1
1.2.PerumusanMasalah.....	1
1.3.BatasanMasalah.....	2
1.4.Tujuan Penelitian.....	2
1.5.ManfaatPenelitian.....	3
1.6.Sistematika Pembahasan.....	3
BAB II TEORI PENUNJANG	
2.1.Hukum Hooke pada Pegas.....	5
2.2. <i>Smart Building</i>	7
2.3. Arduino.....	8
2.3.1. Arduino Uno.....	9
2.3.1.1. Pin Masukan dan Keluaran Arduino Uno.....	11
2.3.1.2. Sumber Daya dan Pin Tegangan Arduino Uno.....	12
2.3.1.3. Peta Memori Arduino Uno.....	13
2.3.1.3.1. Memori Program.....	13
2.3.1.3.2. Memori Data.....	14
2.3.1.3.3. Memori Data EEPROM.....	14
2.3.2. Arduino Ethernet Shield.....	15
2.3.3. Bahasa Pemrograman Arduino.....	16

2.3.3.1. Struktur.....	17
2.3.3.2. Konstanta.....	17
2.3.3.3. Fungsi Masukan dan Keluaran Digital.....	20
2.3.3.4. Progmem.....	21
2.3.4. <i>Arduino Development Environment</i>	24
2.4. Sensor,.....	27
2.4.1. Ultrasonik HC-SR04.....	27
2.5. LCD 2x16 (<i>Liquid Crystal Display</i>).....	30

BAB III METODOLOGI PENELITIAN

3.1. Tempat dan Waktu Penelitian.....	32
3.1.1. Tempat Penelitian.....	32
3.1.2. Waktu Penelitian.....	32
3.2. Metoda Penelitian.....	32
3.2.1. Alat dan Bahan.....	33
3.2.2. Rancangan Struktural.....	34
3.2.3. Rancangan Sistem Elektrikal.....	38
3.2.3.1. Sistem <i>AC-DC</i> Adaptor.....	38
3.2.3.2. Sistem Sensor Ultrasonik HC-SR04.....	39
3.2.3.3. Sistem Penampil Data (LCD 2x16).....	41
3.2.3.4. Sistem <i>Minimum Arduino Uno</i>	42
3.2.3.5. Sistem secara Keseluruhan.....	42
3.3. Pemograman Mikrokontroler ATmega 328 pada Arduino.....	43
3.4. <i>Flowchart</i> Sistem Kerja Alat.....	46

BAB IV PENGUJIAN DAN ANALISA

4.1. Hasil Pembuatan Alat.....	48
--------------------------------	----

4.2. Pengujian Alat dan Analisa.....	49
4.2.1. Pengujian Modul Arduino Uno.....	50
4.2.2. Pengujian Rangkaian Sensor Ultrasonik dengan Arduino Uno.....	52
4.2.3. Pengujian LCD 16x2.....	55
4.2.4. Pengujian Alat secara Keseluruhan.....	57

BAB VKESIMPULAN DAN SARAN

5.1. Kesimpulan.....	64
5.2. Saran.....	65



DAFTAR TABEL

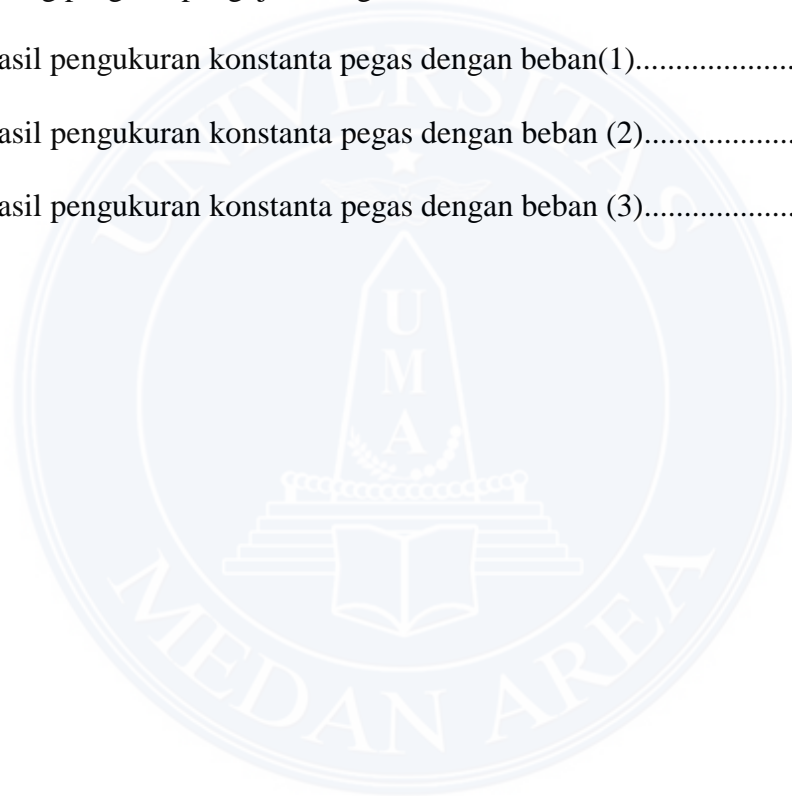
Tabel 2.1 : Deskripsi Ultrasonik HC-SR04.....	30
Tabel 2.2 : Keterangan <i>pin LCD</i> (ArdiWinoto, 2010: 194).....	31
Tabel 3.1 : Penetapan komponen (bahan).....	34
Tabel 4.1 : hasil perbandingan nilai konstanta pegas secara manual dan LCD dengan beban yang berbeda.....	62



DAFTAR GAMBAR

Gambar2.1 :polapengukurankonstantapegas (statis).....	5
Gambar2.2 :Blok diagram arduinoboard.....	9
Gambar2.3 :Arduinouno.....	10
Gambar2.4 :Petamemori program ATmega 328.....	14
Gambar2.5 :Ethernetshield.....	14
Gambar2.6 :Arduinodevelopment environment.....	25
Gambar 2.7: Cara Kerja Sensor Ultrasonik.....	28
Gambar 2.8 : Sensor Ultrasonik HC-SR04.....	29
Gambar2.8 :bentukfisik LCD 2x16.....	30
Gambar 3.1 : Flowchart kerangka berfikir.....	33
Gambar 3.2 : Massa (sebagai beban pegas).....	35
Gambar 3.3 : Desain dan dimensi rangka36.....	36
Gambar 3.4 : Desain tata letak sistem.....	37
Gambar 3.5 : AC-DC adaptor.....	38
Gambar 3.6 : Skema rangkaian AC-DC adaptor.....	39
Gambar 3.7 : Pola penginstalasian sensor ultrasonik HC-SR04.....	40
Gambar 3.8 : Pola instalasi LCD 2x16 pada Arduino Uno.....	41
Gambar 3.9 : Sistem minimum arduino uno.....	42
Gambar 3.10 : Skema rangkaian seluruh sistem.....	43
Gambar 3.11 : Jendela aplikasi penulisan program.....	46
Gambar 3.12: Flowchart sistem pada program aplikasi.....	47
Gambar4.1 :Alatukurkonstantapegas.....	48
Gambar4.2 :Rangkaianpengujianmodularduinodenganindikator led.....	50

Gambar4.3 :Listing program pengujian lampu led.....	51
Gambar4.4 : Listing program pengujian lampu led.....	52
Gambar4.5 :Rangkaianujicoba sensor ultrasonik dengan arduinouno.....	52
Gambar4.6 :Rangkaiannyata sensor ultrasonik dengan arduinouno.....	54
Gambar4.7 :Tampilanhasilpengukuran sensor ultrasonic.....	55
Gambar4.8 : Pengujianrangkaian LCD 16x2 karakter.....	56
Gambar4.9 : Listing program pengujian rangkaian LCD 16x2 karakter.....	56
Gambar4.10 :Hasil pengukuran konstanta pegas dengan beban(1).....	57
Gambar4.11 :Hasil pengukuran konstanta pegas dengan beban (2).....	59
Gambar4.12 :Hasil pengukuran konstanta pegas dengan beban (3).....	60



BAB I PENDAHULUAN

1.2. Latar Belakang

Pengukuran selalu berhubungan dengan besaran dan satuan. Sebagai contoh, saat mengukur panjang buku dan pensil, ini berarti membandingkan panjang buku dengan panjang pensil. Dengan demikian dapat dikatakan bahwa pengertian pengukuran adalah membandingkan sejenis yang ditetapkan sebagai satuan. Kegiatan mengukur memerlukan alat yang digunakan sebagai pembanding dalam pengukuran. Alat yang digunakan untuk mengukur disebut alat ukur.

Dalam pengukuran agar hasil pengukuran yang dilakukan dianggap sebagai satuan standar, maka alat ukur harus memperhatikan syarat-syarat sebagai berikut:

1. Tidak berubah (tetap) dalam keadaan apapun
2. Dapat digunakan secara Internasional
3. Mudah diproduksi (mudah ditiru)

Terkait hal ini sehubungan dengan kemajuan perkembangan teknologi digital saat ini khususnya pada kasus penentuan nilai konstanta pegas, peneliti mencoba mengangkat sebuah judul penelitian dalam konteks “rancang bangun” yaitu membuat alat pengukur konstanta pegas yang berkemampuan cepat dan akurat, sehingga dengan alat ini dapat membantu pada setiap aspek di berbagai bidang yang berkaitan dengan kuantitas konstanta pegas.

1.2. Perumusan Masalah

Adapun yang menjadi rumusan masalah dalam penelitian ini adalah :

1. Bagaimana membuat rancangan alat cerdas dan berkemampuan cepat dalam penentuan nilai konstanta pegas tanpa melakukan penghitungan secara manual lagi?
2. Variabel apa yang dideteksi agar dapat mengetahui kuantitas konstanta pegas?

3. Bagaimana alat ukur yang dibuat agar tidak terbatas untuk penggunaan pengukuran berbagai ukuran pegas ?
4. Alat apa yang digunakan agar dapat melihat dengan mudah hasil kuantitas konstanta pegas yang telah diukur ?

1.3. Batasan Masalah

Yang menjadi batasan masalah dalam penelitian ini adalah :

1. Proyek akhir yang disajikan ini adalah berbentuk studi pembuatan alat yang sumber referensinya adalah jelas dan ilmiah.
2. Ruang lingkup pembahasan meliputi komponen *hardware* (perangkat keras) *software* (perangkat lunak) yang meliputi bahasa pemrograman yaitu menggunakan bahasa "C" dan *software* aplikasi untuk memasukkan program adalah *Arduino*.

1.4. Tujuan Penelitian

Tujuan penelitian ini adalah :

1. Merancang sekaligus membuat secara hardware dan software sebuah alat cerdas yang dapat menghitung secara cepat nilai konstanta pegas yang berbeda-beda.
2. Memanfaatkan karakteristik pendeteksian sensor ultrasonik sebagai pengukur besaran panjang dalam satuan meter selanjutnya dimasukkan ke dalam rumus.
3. Membuat rancangan dimensi dan model alat ukur yang fleksibel agar dapat digunakan untuk pengukuran nilai konstanta pegas yang berbeda-beda.
4. Mengaplikasikan LCD 2x16 sebagai penampil hasil pengukuran nilai konstanta pegas yang diukur.

1.5. Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Dapat memudahkan bagi setiap yang berkepentingan dalam hal pengukuran secara cepat nilai konstanta pegas yang berbeda-beda.
2. Sebagai pembanding pada pengukuran secara manual baik di bidang pendidikan maupun di bidang pekerjaan.
3. Mahasiswa dapat mengasah kemampuan sehingga dapat menciptakan inovasi baru yang konteksnya masih berkaitan dengan masalah ini.

1.6. Sistematika Pembahasan

Sistematika penulisan pada masing-masing bab adalah sebagai berikut :

1. Bab I Pendahuluan

Menjelaskan secara singkat tentang latar belakang penelitian, rumusan masalah penelitian, batasan masalah, tujuan dan manfaat penelitian serta sistematika penulisan.

2. Bab II Teori Penunjang

Bab ini membahas tentang teori-teori pendukung dalam penelitian ini sehingga hasil yang didapatkan lebih optimal.

3. Bab III Metodologi Penelitian

Berisi tentang bagaimana metode penelitian dilakukan, yang meliputi bagaimana cara pengambilan data, dan cara perancangan serta pembuatan alat.

4. Bab IV Hasil dan Pembahasan

Berisi penjelasan tentang hasil perancangan dan pembuatan alat, pengujian beserta pembahasannya.

5. Bab V Penutup

Berisi tentang kesimpulan dan saran yang didapatkan dari penelitian yang dilakukan.



BAB II TEORI PENUNJANG

2.1. Hukum Hooke pada Pegas

Sistem pada tahun 1676, Robert Hooke mengusulkan suatu hukum fisika yang menyangkut pertambahan panjang sebuah benda elastis yang dikenai oleh suatu gaya. Menurut Hooke, pertambahan panjang berbanding lurus dengan yang diberikan pada benda. Secara matematis, hukum Hooke ini dapat dituliskan sebagai

$$F = k \cdot x \dots\dots\dots (2.1)$$

dengan:

F = gaya yang dikerjakan (N)

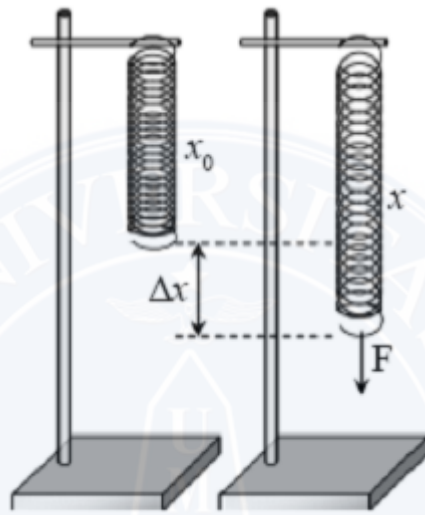
x = pertambahan panjang (m)

k = konstanta gaya (N/m)

(Bob Foster, 2004:122-123)

Pegas merupakan salah satu contoh benda elastis. elastis atau elastisitas adalah kemampuan sebuah benda untuk kembali ke bentuk awalnya ketika gaya luar yang diberikan pada benda tersebut dihilangkan. Jika sebuah gaya diberikan pada sebuah benda yang elastis, maka bentuk benda tersebut berubah. Untuk pegas dan karet, yang dimaksudkan dengan perubahan bentuk adalah pertambahan panjang. Perlu kita ketahui bahwa gaya yang diberikan juga memiliki batas-batas tertentu. Sebuah karet bisa putus jika gaya tarik yang diberikan sangat besar, melawati batas elastisitasnya. Demikian juga sebuah pegas tidak akan kembali ke bentuk semula jika diregangkan dengan gaya yang sangat besar. Jadi benda-benda

elastis tersebut memiliki batas elastisitas. Setiap pegas memiliki panjang alami, jika pada pegas tersebut tidak diberikan gaya. Pada keadaan ini, benda yang dikaitkan pada ujung pegas berada dalam posisi setimbang. Untuk semakin memudahkan pemahaman, berikut Gambar 2.1 yang menampilkan pola pengukuran konstanta pegas.



Gambar 2.1 : pola pengukuran konstanta pegas (statis)

Ada dua metode dalam mencari konstanta pegas yaitu metode dinamis dan statis. Cara statis yaitu variabel manipulasinya adalah massa, variabel kontrolnya adalah panjang pegas mula-mula dan variabel responnya adalah panjang simpangan sedangkan cara dinamis variabel manipulasinya adalah massa, variabel kontrolnya adalah jumlah getaran besar simpangan yang diberikan, dan variabel responnya adalah periode (T) selama 10 kali getaran.

Metode yang digunakan pada pegas serta menentukan tetapan pegas dengan cara statis yaitu mengukur pertambahan panjang dengan menggunakan mistar dengan mengukur panjang mula-mula pegasnya terlebih dahulu. Sedangkan untuk cara dinamis dengan cara menggantungkan beban pada ujung pegas

kemudian digantungkan pada statif, lalu mengukur pertambahan panjang pegas tersebut sebagai panjang mula-mula setelah itu pegas diberi simpangan sepanjang 5 cm sehingga terjadi getaran selaras dengan waktu 10 kali getaran menggunakan stopwatch.

2.2. Smart Building

Perkembangan teknologi melaju dengan sangat pesat. Perkembangan teknologi ini merupakan hasil kerja keras dari rasa ingin tahu manusia terhadap suatu hal yang pada akhirnya diharapkan akan mempermudah kehidupan manusia. Salah satu cara untuk mempermudah kehidupan manusia adalah dengan membangun *smart building*.

Smart building sudah menjadi salah satu topik pembahasan dalam karya tulis fiksi ilmiah selama bertahun-tahun, namun baru mulai dicoba untuk diwujudkan pada awal abad ke 20 ketika pemakaian energi listrik telah meluas hingga ke rumah-rumah dan mulai majunya teknologi informasi.

Smart building mengacu pada penggunaan teknologi informasi dan komputer untuk mengendalikan peralatan yang berada di dalam bangunan tersebut (seperti jedela atau lampu). Sistem yang beroperasi dalam *smartbuilding* dapat berupa sistem kontrol jarak jauh yang sederhana dari lampu-lampu hingga sebuah sistem yang berbasis komputer atau mikrokontroler yang memiliki tingkat kecerdasan yang bervariasi yang secara otomatis mengontrol peralatan yang berada di dalam bangunan.

Elemen dalam sebuah sistem *smart building* terdiri dari sensor (seperti sensor cahaya atau sensor suhu), pengontrol (seperti komputer atau

mikrokontroler) dan aktuator (seperti motor, *relay*, *servo* atau sakelar lampu). Sebuah antarmuka antara manusia dengan mesin diperlukan, agar pemilik bangunan dapat berinteraksi dengan sistem untuk memonitor atau mengontrolnya.

Antarmuka ini dapat berupa terminal khusus atau aplikasi yang berjalan pada telepon genggam atau komputer tablet. Antarmuka tersebut berkomunikasi dengan sistem melalui jaringan kabel atau *wireless* menggunakan satu atau lebih protokol.

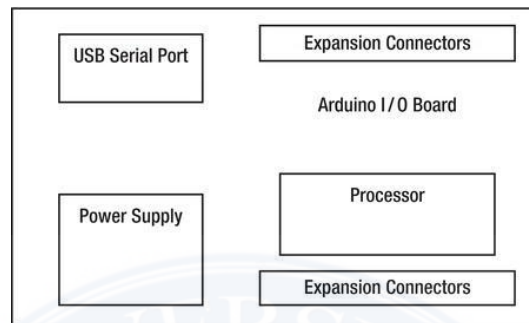
2.3. Arduino

Arduino adalah platform pembuatan prototipe elektronik yang bersifat *open-source hardware* yang berdasarkan pada perangkat keras dan perangkat lunak yang fleksibel dan mudah digunakan. Arduino ditujukan bagi para seniman, desainer, dan siapapun yang tertarik dalam menciptakan objek atau lingkungan yang interaktif.

Arduino pada awalnya dikembangkan di Ivrea, Italia. Nama *arduino* adalah sebuah nama maskulin yang berarti teman yang kuat. *Platformarduino* terdiri dari *arduino board*, *shield*, bahasa pemrograman *arduino*, dan *arduino development environment*. *Arduino board* biasanya memiliki sebuah *chip* dasarmikrokontroler Atmel AVR ATmega8 berikut turunannya. Blok diagram *arduino board* yang sudah disederhanakan dapat dilihat pada Gambar 2.2. *Shield* adalah sebuah papan yang dapat dipasang diatas *arduino board* untuk menambah kemampuan dari *arduino board*.

Bahasa pemrograman *arduino* adalah bahasa pemrograman yang umum

digunakan untuk membuat perangkat lunak yang ditanamkan pada *arduino board*. Bahasa pemrograman arduino mirip dengan bahasa pemrograman C++.



Gambar 2.2 : Blok diagram arduino board

Arduino Development Environment adalah perangkat lunak yang digunakan untuk menulis dan meng-*compile* program untuk arduino. *Arduino Development Environment* juga digunakan untuk meng-*upload* program yang sudah di-*compile* ke memori program arduino board.

2.3.1. Arduino Uno

Arduino Uno adalah arduino board yang menggunakan mikrokontroler ATmega328. Arduino Uno memiliki 14 pin digital (6 pin dapat digunakan sebagai output PWM), 6 input analog, sebuah 16 MHz osilator kristal, sebuah koneksi USB, sebuah konektor sumber tegangan, sebuah header ICSP, dan sebuah tombol reset. Arduino Uno memuat segala hal yang dibutuhkan untuk mendukung sebuah mikrokontroler. Hanya dengan menghubungkannya ke sebuah komputer melalui USB atau memberikan tegangan DC dari baterai atau adaptor AC ke DC sudah dapat membuatnya bekerja. Arduino Uno menggunakan ATmega16U2 yang

diprogram sebagai *USB-to-serial converter* untuk komunikasi serial ke computer melalui port USB. Tampak atas dari arduino uno dapat dilihat pada Gambar 2.3.

Adapun data teknis *board* Arduino Uno R3 adalah sebagai berikut:

1. Mikrokontroler : ATmega328
2. Tegangan Operasi : 5V
3. Tegangan Input (recommended) : 7 - 12 V
4. Tegangan Input (limit) : 6-20 V
5. Pin digital I/O : 14 (6 diantaranya pin PWM)
6. Pin Analog input : 6
7. Arus DC per pin I/O : 40 mA
8. Arus DC untuk pin 3.3 V : 150 mA
9. Flash Memory : 32 KB dengan 0.5 KB digunakan untuk bootloader
10. SRAM : 2 KB
11. EEPROM : 1 KB
12. Kecepatan Pewaktuan : 16 Mhz



Gambar 2.3 : Arduino uno

2.3.1.1. Pin Masukan dan Keluaran Arduino Uno

Masing-masing dari 14 pin digital arduino uno dapat digunakan sebagai masukan atau keluaran menggunakan fungsi `pinMode()`, `digitalWrite()` dan `digitalRead()`. Setiap pin beroperasi pada tegangan 5 volt. Setiap pin mampu menerima atau menghasilkan arus maksimum sebesar 40 mA dan memiliki resistor *pull-up* internal (diputus secara default) sebesar 20-30 KOhm. Sebagai tambahan, beberapa pin masukan digital memiliki kegunaan khusus yaitu:

1. Komunikasi serial: pin 0 (RX) dan pin 1 (TX), digunakan untuk menerima (RX) dan mengirim (TX) data secara serial.
2. External Interrupt: pin 2 dan pin 3, pin ini dapat dikonfigurasi untuk memicu sebuah interrupt pada nilai rendah, sisi naik atau turun, atau pada saat terjadi perubahan nilai.
3. Pulse-width modulation (PWM): pin 3,5,6,9,10 dan 11, menyediakan keluaran PWM 8-bit dengan menggunakan fungsi `analogWrite()`.
4. Serial Peripheral Interface (SPI): pin 10 (SS), 11 (MOSI), 12 (MISO) dan 13 (SCK), pin ini mendukung komunikasi SPI dengan menggunakan SPI library.
5. LED: pin 13, terdapat built-in LED yang terhubung ke pin digital 13. Ketika pin bernilai HIGH maka LED menyala, sebaliknya ketika pin bernilai LOW maka LED akan padam.

Arduino Uno memiliki 6 masukan analog yang diberi label A0 sampai A5, setiap pin menyediakan resolusi sebanyak 10 bit (1024 nilai yang berbeda). Secara

default pin mengukur nilai tegangan dari ground (0V) hingga 5V, walaupun begitu dimungkinkan untuk mengganti nilai batas atas dengan menggunakan pin AREF dan fungsi `analogReference()`. Sebagai tambahan beberapa pin masukan analog memiliki fungsi khusus yaitu pin A4 (SDA) dan pin A5 (SCL) yang digunakan untuk komunikasi *Two Wire Interface* (TWI) atau *Inter IntegratedCircuit* (I2C) dengan menggunakan Wire library.

2.3.1.2. Sumber Daya dan Pin Tegangan Arduino Uno

Arduino uno dapat diberi daya melalui koneksi USB (Universal Serial Bus) atau melalui power supply eksternal. Jika arduino uno dihubungkan ke kedua sumber daya tersebut secara bersamaan maka arduino uno akan memilih salah satu sumber daya secara otomatis untuk digunakan. Power supplay external (yang bukan melalui USB) dapat berasal dari adaptor AC ke DC atau baterai. Adaptor dapat dihubungkan ke soket power pada arduino uno. Jika menggunakan baterai, ujung kabel yang dibubungkan ke baterai dimasukkan kedalam pin GND dan Vin yang berada pada konektor Power.

Arduino uno dapat beroperasi pada tegangan 6 sampai 20 Volt. Jika arduino uno diberi tegangan di bawah 7 Volt, maka pin 5V akan menyediakan tegangan di bawah 5 Volt dan arduino uno mungkin bekerja tidak stabil. Jika diberikan tegangan melebihi 12 Volt, penstabil tegangan kemungkinan akan menjadi terlalu panas dan merusak arduino uno. Tegangan rekomendasi yang diberikan ke arduino uno berkisar antara 7 sampai 12 Volt.

Pin-pin tegangan pada arduino uno adalah sebagai berikut:

1. Vin adalah pin untuk mengalirkan sumber tegangan ke arduino uno ketika

menggunakan sumber daya eksternal (selain dari koneksi USB atau sumber daya yang teregulasi lainnya). Sumber tegangan juga dapat disediakan melalui pin ini jika sumber daya yang digunakan untuk arduino uno dialirkan melalui soket *power*.

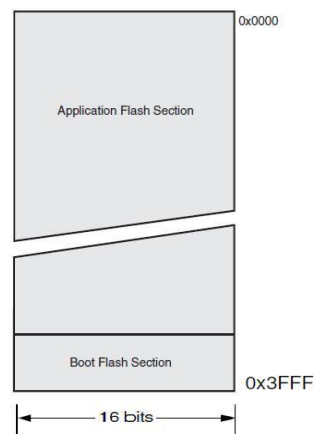
2. 5V adalah pin yang menyediakan tegangan teregulasi sebesar 5 Volt berasal dari *regulator* tegangan pada arduino uno.
3. 3V3 adalah pin yang menyediakan tegangan teregulasi sebesar 3,3 Volt berasal dari regulator tegangan pada arduino uno.
4. GND adalah pin *ground*.

2.3.1.3. Peta Memori Arduino Uno

Arduino Uno adalah arduino *board* yang menggunakan mikrokontroler ATmega328. Maka peta memori arduino uno sama dengan peta memori pada mikrokontroler ATmega328.

2.3.1.3.1. Memori Program

ATMega328 memiliki 32K byte *On-chip In-System ReprogrammableFlash Memory* untuk menyimpan program. Memori *flash* dibagi kedalam duabagian, yaitu bagian program *bootloader* dan aplikasi seperti terlihat pada Gambar 2.4. *Bootloader* adalah program kecil yang bekerja pada saat sistem dimulai yang dapat memasukkan seluruh program aplikasi ke dalam memori prosesor.



Gambar 2.4 : Peta memori program ATmega 328

2.3.1.3.2. Memori Data

Memori data ATmega328 terbagi menjadi 4 bagian, yaitu 32 lokasi untuk register umum, 64 lokasi untuk register I/O, 160 lokasi untuk register I/O tambahan dan sisanya 2048 lokasi untuk data SRAM internal. Register umum menempati alamat data terbawah, yaitu 0x0000 sampai 0x001F. Register I/O menempati 64 alamat berikutnya mulai dari 0x0020 hingga 0x005F. Register I/O tambahan menempati 160 alamat berikutnya mulai dari 0x0060 hingga 0x00FF. Sisa alamat berikutnya mulai dari 0x0100 hingga 0x08FF digunakan untuk SRAM internal.

2.3.1.3.3. Memori Data EEPROM

Arduino uno terdiri dari 1 KByte memori data EEPROM. Pada memori EEPROM, data dapat ditulis/dibaca kembali dan ketika catu daya dimatikan, data terakhir yang ditulis pada memori EEPROM masih tersimpan pada memori ini, atau dengan kata lain memori EEPROM bersifat *nonvolatile*. Alamat EEPROM dimulai dari 0x000 hingga 0x3FF.

2.3.2. Arduino Ethernet Shield

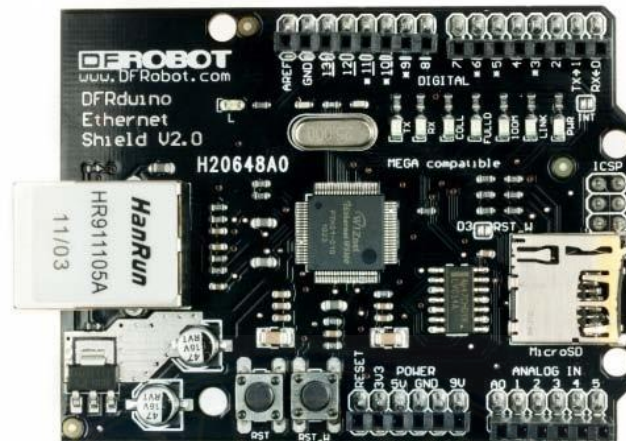
Ethernet Shield menambah kemampuan arduino board agar terhubung ke jaringan komputer. Ethernet shield berbasiskan cip ethernet Wiznet W5100. Ethernet *library* digunakan dalam menulis program agar arduino board dapat terhubung ke jaringan dengan menggunakan arduino ethernet shield.

Pada ethernet shield terdapat sebuah slot micro-SD, yang dapat digunakan untuk menyimpan file yang dapat diakses melalui jaringan. Onboard micro-SD card reader diakses dengan menggunakan *SD library*.

Arduino board berkomunikasi dengan W5100 dan SD card menggunakan bus SPI (Serial Peripheral Interface). Komunikasi ini diatur oleh library SPI.h dan Ethernet.h. Bus SPI menggunakan pin digital 11, 12 dan 13 pada Arduino Uno. Pin digital 10 digunakan untuk memilih W5100 dan pin digital 4 digunakan untuk memilih SD card. Pin-pin yang sudah disebutkan sebelumnya tidak dapat digunakan untuk input/output umum ketika kita menggunakan ethernet shield.

Karena W5100 dan SD card berbagi bus SPI, hanya salah satu yang dapat aktif pada satu waktu. Jika kita menggunakan kedua perangkat dalam program kita, hal ini akan diatasi oleh *library* yang sesuai. Jika kita tidak menggunakan salah satu perangkat dalam program kita, kiranya kita perlu secara eksplisit men-*deselect*-nya. Untuk melakukan hal ini pada SD card, set pin 4 sebagai output dan menuliskan logika tinggi padanya, sedangkan untuk W5100 yang digunakan adalah pin 10. DFRduino Ethernet shield adalah sebuah *clone* dari arduino Ethernet shield yang dibuat oleh DFRobot.

Penampakan DFRduino Ethernet shield dapat dilihat pada Gambar 2.5.



Gambar 2.5 : Ethernetshield

2.3.3. Bahasa Pemrograman Arduino

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 Pemrograman Arduino adalah bahasa pemrograman utama yang digunakan untuk membuat program untuk arduino board. Bahasa pemrograman arduino menggunakan bahasa pemrograman C sebagai dasarnya.

Karena menggunakan bahasa pemrograman C sebagai dasarnya, bahasa pemrograman arduino memiliki banyak sekali kemiripan, walaupun beberapa hal telah berubah.

2.3.3.1. Struktur

Setiap program dalam arduino board terdiri dari dua fungsi utama yaitu

setup() dan loop(). Instruksi yang berada dalam fungsi setup() dieksekusi hanya sekali, yaitu ketika arduino board pertama kali dihidupkan. Biasanya instruksi yang berada pada fungsi setup() merupakan konfigurasi dan inisialisasi dari arduino board. Instruksi yang berada pada fungsi loop() dieksekusi berulang-ulang hingga arduino board dimatikan (catu daya diputus). Fungsi loop() merupakan tugas utama dari arduino board. Jadi setiap program yang menggunakan bahasa pemrograman arduino memiliki struktur sebagai berikut:

```
void setup()
{
  // perintah-perintah untuk konfigurasi dan inisialisasi arduino board
}

void loop()
{
  //perintah-perintah utama arduino board
}
```

Program diatas dapat dianalogikan dalam bahasa C sebagai berikut:

```
void setup(void); // prototipe fungsi setup
void loop(void); // prototipe fungsi loop
int main(void) {
  setup(); //
  while(1) {
    loop(); // ulangi terus menerus
  }
  return 0; //bagian ini tidak akan pernah dieksekusi
}
```

2.3.3.2. Konstanta

Konstanta adalah variable yang sudah ditetapkan sebelumnya dalam bahasa pemrograman arduino. Konstanta digunakan agar program lebih mudah untuk dibaca dan dimengerti. Konstanta dibagi menjadi 3 kelompok yaitu:

1. Konstanta yang digunakan untuk menunjukkan tingkat logika (konstanta Boolean), yaitu *true* dan *false*

2. Konstanta untuk menunjukkan keadaan pin, yaitu HIGH dan LOW
3. Konstanta untuk menunjukkan fungsi pin, yaitu INPUT, INPUT_PULLUP dan OUTPUT

Konstanta yang digunakan untuk menunjukkan benar atau salah dalam bahasa pemrograman arduino adalah *true* dan *false*. *False* lebih mudah didefinisikan daripada *true*. *False* didefinisikan sebagai 0(nol). *True* sering didefinisikan sebagai 1(satu), yang mana hal ini benar, tetapi *true* memiliki definisi yang lebih luas. Setiap integer yang bukan nol adalah *true* dalam pengertian Boolean. Jadi -2, 3 dan -100 semuanya didefinisikan sebagai *true*, juga dalam pengertian Boolean. Tidak seperti konstanta yang lain *true* dan *false* diketik dengan menggunakan huruf kecil.

Ketika membaca atau menulis ke sebuah pin digital, terdapat hanya dua nilai yang dapat diberikan atau diterima, yaitu HIGH dan LOW. HIGH memiliki arti yang berbeda tergantung apakah sebuah pin dikonfigurasi menjadi masukan atau keluaran. Ketika pin dikonfigurasi sebagai masukan dengan fungsi `pinMode()`, lalu kemudian dibaca dengan fungsi `digitalRead()`, mikrokontroler akan melaporkan nilai HIGH jika tegangan yang ada pada pin tersebut berada pada tegangan 3 volt atau lebih.

Ketika sebuah pin dikonfigurasi sebagai masukan, dan kemudian dibuat bernilai HIGH dengan fungsi `digitalWrite()`, maka resistor *pull-up* internal dari chip ATmega akan aktif, yang akan membawa pin masukan ke nilai HIGH kecuali pin tersebut ditarik (*pull-down*) ke nilai LOW oleh sirkuit dari luar.

Ketika pin dikonfigurasi sebagai keluaran dengan fungsi `pinMode()`, dan

diset ke nilai HIGH dengan fungsi `digitalWrite()`, maka pin berada pada tegangan 5 volt. Dalam keadaan ini, pin tersebut dapat memberikan arus, sebagai contoh, untuk menhidupkan LED yang terhubung seri dengan resistor dan ground, atau pin lain yang dikonfigurasi sebagai keluaran dan diberi nilai LOW.

Sama seperti HIGH, LOW juga memiliki arti yang berbeda bergantung pada konfigurasi pin. Ketika pin dikonfigurasi sebagai masukan, maka mikrokontroler akan melaporkan nilai LOW jika tegangan yang terdapat pada pin berada pada tegangan 2 volt atau kurang. Ketika pin dikonfigurasi sebagai keluaran dan diberi nilai LOW maka pin berada pada tegangan 0 volt.

Setiap pin pada arduino dapat dikonfigurasi sebagai masukan, masukan dengan resistor *pull-up* atau keluaran. Untuk mengkonfigurasi fungsi pin pada arduino digunakan konstanta `INPUT`, `INPUT_PULLUP` dan `OUTPUT`. Pin arduino yang dikonfigurasi sebagai masukan dengan fungsi `pinMode()` dikatakan berada dalam kondisi berimpedansi tinggi. Pin yang dikonfigurasi sebagai masukan memiliki permintaan yang sangat kecil kepada sirkuit yang di-sampling-nya, setara dengan sebuah resistor 100 Megaohm dipasang seri dengan pin tersebut. Hal ini membuat pin tersebut berguna untuk membaca sensor, tetapi tidak untuk menhidupkan sebuah *LED*.

Cip ATmega pada arduino memiliki resistor *pull-up* internal (resistor yang terhubung ke sumber tegangan secara internal) yang dapat digunakan. Untuk menggunakan resistor *pull-up* internal ini kita menggunakan konstanta `INPUT_PULLUP` pada fungsi `pinMode()`.

Pin yang dikonfigurasi menjadi sebuah keluaran dikatakan berada dalam kondisi berimpedansi rendah. Hal ini berarti pin tersebut dapat menyediakan

sejumlah besar arus ke sirkuit yang lain. Pin pada ATmega mampu menyediakan arus hingga 40 mA.

2.3.3.3. Fungsi Masukan dan Keluaran Digital

Arduino memiliki 3 fungsi untuk masukan dan keluaran digital pada arduino board, yaitu `pinMode()`, `digitalWrite()` dan `digitalRead()`.

Fungsi `pinMode()` mengkonfigurasi pin tertentu untuk berfungsi sebagai masukan atau keluaran. Sintaksis untuk fungsi `pinMode()` adalah sebagai berikut:

`pinMode(pin, mode)`

Parameter: pin = angka dari pin digital yang akan dikonfigurasi

mode = konfigurasi yang diinginkan (INPUT, INPUT_PULLUP dan OUTPUT).

Fungsi `digitalWrite()` berfungsi untuk memberikan nilai HIGH atau LOW suatu digital pin. Sintaksis untuk fungsi `digitalWrite()` adalah sebagai berikut:

`digitalWrite(pin, value)`

Parameter: pin = angka dari pin digital yang akan dikonfigurasi value = nilai yang diinginkan (HIGH atau LOW).

Fungsi `digitalRead()` bertujuan untuk membaca nilai yang ada pada pin arduino uno. Sintaksis untuk fungsi `digitalRead()` adalah sebagai berikut:

`digitalRead(pin)`

Parameter: pin = angka dari pin digital yang akan dibaca

Berikut ini adalah contoh penggunaan fungsi masukan dan keluaran digital dalam sebuah program:

```
int ledPin = 13; // LED terhubung ke pin      digi
int inPin = 7;  // pushbutton terhubung     ke pin dig
int val = 0;    //          variable untuk menyimpan

void setup()
{
  pinMode(ledPin, OUTPUT); // set pin digital 13 sebagai keluaran
  pinMode(inPin, INPUT);  // set pin digital 13 sebagai masukan
}

void loop()
{
  val = digitalRead(inPin); // baca nilai pin input
  digitalWrite(ledPin, val); // sets LED sesuai dengan nilai val
}
```

2.3.3.4. Progmem

Terkadang program dibuat memerlukan memori yang lebih besar, misalnya kita perlu menyimpan banyak teks dalam sebuah program penampil LCD. Hal ini bisa kita atasi dengan menyimpan data pada flash memori (memori program). Besar memori program pada arduino uno adalah 32 KByte.

Kata kunci `PROGMEM` adalah sebuah *variable modifier*, `PROGMEM` hanya bisa digunakan untuk tipe data yang di definisikan pada *library pgmspace.h*. `PROGMEM` menjelaskan pada kompiler “letakkan informasi berikut pada memori program”, menggantikan SRAM, dimana biasanya informasi tersebut disimpan.

`PROGMEM` adalah bagian dari *library pgmspace.h*, jadi kita perlu memasukkan *library pgmspace.h* ke program kita. Sintaksis untuk menggunakan `PROGMEM` adalah sebagai berikut:

```
dataType variableName[] PROGMEM = {dataInt0, dataInt1, ...};
```

dataType - tipe variable

variableName - nama array data

Karena PROGMEM adalah sebuah *variable modifier*, tidak ada aturan yang baku untuk tempat peletaknya. Kompiler arduino menerima semua definisi yang ditunjukkan di bawah.

```
dataType variableName[] PROGMEM = {}; dataType PROGMEM
variableName[] = {}; PROGMEM dataType variableName[] = {};
```

PROGMEM juga bisa digunakan untuk menyimpan satu variabel, tetapi karena kerepotan dalam penggunaannya, sebaiknya hanya digunakan untuk menyimpan data yang sangat banyak, yang biasanya lebih mudah menyimpannya di dalam array.

Dalam menggunakan PROGMEM kita perlu melakukan 2 langkah prosedur. Pertama adalah menyimpan data pada memori program, kemudian kita menggunakan fungsi khusus, yang juga didefinisikan di *library pgmspace.h*, untuk membaca data tersebut dari memori program dan menuliskannya ke SRAM, sehingga kita dapat melakukan sesuatu dengan data tersebut.

Berikut adalah tipe data yang didukung oleh *library pgmspace.h*:

```
prog_char      - a signed char (1 byte) -127 sampai 128
prog_uchar     - an unsigned char (1 byte) 0 sampai 255
prog_int16_t   - a signed int (2 bytes) -32,767 sampai 32,768
prog_uint16_t  - an unsigned int (2 bytes) 0 sampai 65,535
prog_int32_t   - a signed long (4 bytes) -2,147,483,648 sampai *
                2,147,483,647.
prog_uint32_t  - an unsigned long (4 bytes) 0 sampai 4,294,967,295
```

Berikut ini adalah contoh program untuk menggunakan PROGMEM

```
#include <avr/pgmspace.h>

// simpan beberapa unsigned int
PROGMEM prog_uint16_t charSet[] = { 65000, 32796, 16843, 10,
11234};

// simpan beberapa karakter
prog_uchar signMessage[] PROGMEM = {"UNIVERSITAS SUMATERA
UTARA"};

unsigned int displayInt;
int k; // variabel counter
char myChar;

// baca int
displayInt = pgm_read_word_near(charSet + k)

// baca karakter
myChar = pgm_read_byte_near(signMessage + k);
```

Di bawah ini adalah contoh program untuk menyimpan array string pada program memori. Karena string sendiri adalah sebuah array, contoh di bawah ini juga merupakan contoh penggunaan array dua dimensi.

```
#include <avr/pgmspace.h>
prog_char string_0[] PROGMEM = "String 0";
prog_char string_1[] PROGMEM = "String 1";
prog_char string_2[] PROGMEM = "String 2";
prog_char string_3[] PROGMEM = "String 3";
prog_char string_4[] PROGMEM = "String 4";
prog_char string_5[] PROGMEM = "String 5";

//buat table untuk menunjukkan string.
PROGMEM const char *string_table[] =
{
    string_0,
    string_1,

    string_2, string_3, string_4, string_5 };
char buffer[30]; //pastikan buffer cukup untuk menyimpan string

void setup()
{
```

```

    Serial.begin(9600);
}

void loop()
{
    /*menggunakan table string pada memori program membutuhkan
    fungsi special untuk mengambil data tersebut.
    fungsi strcpy_P ("buffer") menyalin sebuah string dari
    program memori sebuah string di SRAM ("buffer").
    Pastikan bahwa SRAM cukup untuk menampung setiap string yang
    diambil dari memori program*/

    for (int i = 0; i < 6; i++)
    {
        strcpy_P(buffer, (char*)pgm_read_word(&(string_table[i]))); //
        salin string ke buffer
        Serial.println( buffer ); delay( 500 );
    }
}

```

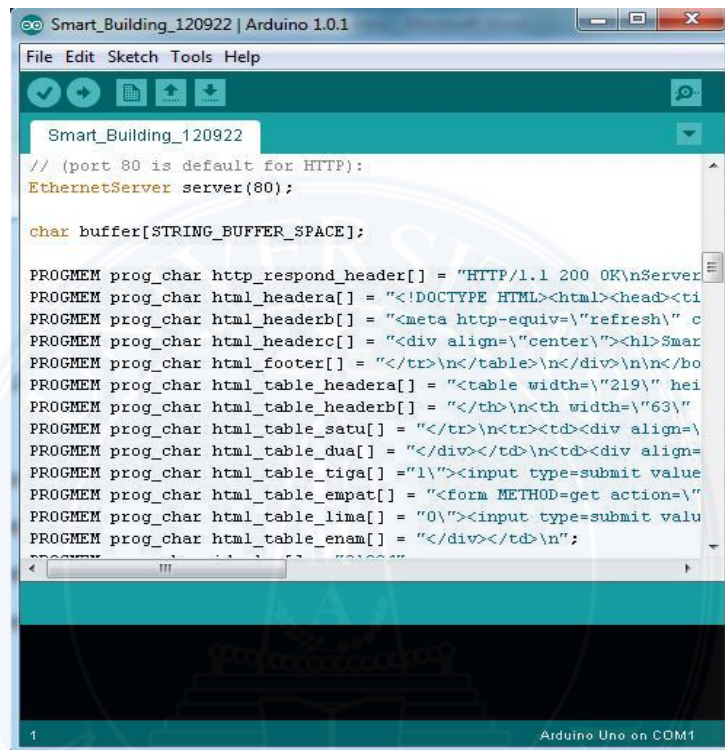
2.3.4. Arduino Development Environment

Arduino *Development Environment* terdiri dari editor teks untuk menuliskode, 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 DevelopmentEnvironment* disebut *sketch*. *Sketch* ditulis pada editor teks. *Sketch* disimpandengan file berekstensi *.ino*. Area pesan memberikan memberikan informasi dan pesan error ketika kita menyimpan atau membuka *sketch*. Konsol menampilkan output teks dari *Arduino Development Environment* dan juga menampilkan pesan





error ketika kita mengkompile *sketch*. Pada sudut kanan bawah dari jendela *Arduino Development Environment* menunjukkan jenis board dan port

seriak yang sedang digunakan. Tombol *toolbar* digunakan untuk mengecek dan mengupload *sketch*, membuat, membuka atau menyimpan *sketch*, dan menampilkan *serialmonitor*.



Gambar 2.6 : Arduinodevelopment environment

Berikut ini adalah tombol-tombol *toolbar* serta fungsinya:

- 
Verify
 mengecek error pada code program.
- 
Upload
 meng-*compile* dan meng-*upload* program ke Arduino board.
- 
New
 membuat sketch baru.
- 
Open
 menampilkan sebuah menu dari seluruh *sketch* yang berada didalam *sketchbook*.



Save
menyimpan sketch.



Serial Monitor
membuka serial *monitor*.

Dalam lingkungan arduino digunakan sebuah konsep yang disebut *sketchbook*, yaitu tempat standar untuk menumpukan program (*sketch*). *Sketch* yang ada pada *sketchbook* dapat dibuka dari menu **File > Sketchbook** atau dari tombol *open* pada *toolbar*. Ketika pertama kali menjalankan arduino developmentenvironment, sebuah direktori akan dibuat secara otomatis untuk tempat penyimpanan *sketchbook*. Kita dapat melihat atau mengganti lokasi dari direktori tersebut dari menu **File > Preferences**.

Serial monitor menampilkan data serial yang sedang dikirim dari arduinoboard. Untuk mengirim data ke board, masukkan teks dan klik tombol *send* atau tekan *enter* pada *keyboard*.

Sebelum meng-*upload* program, kita perlu mensetting jenis board dan port serial yang sedang kita gunakan melalui menu **Tools > Board** dan **Tools > SerialPort**. Pemilihan board berguna untuk mengeset parameter (contohnya: kecepatanmikrokontroler dan *baud rate*) yang digunakan ketika meng-*compile* dan meng-*upload sketch*.

Setelah memilih board dan port serial yang tepat, tekan tombol *upload* pada *toolbar* atau pilih menu **File > Upload**. Arduino board akan me-*reset* secara otomatis dan proses *upload* akan dimulai. Pada kebanyakan board, LED RX dan TX akan berkedip ketika program sedang di-*upload*. Arduino *developmentenvironment* akan menampilkan pesan ketika proses *upload* telah

selesai, atau menampilkan pesan error.

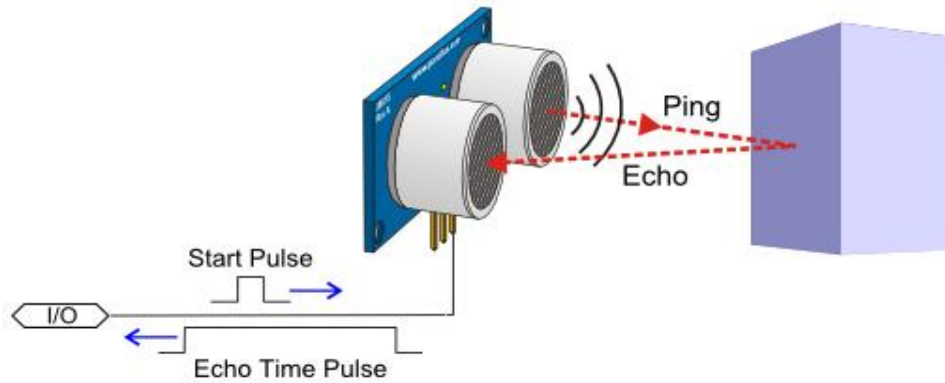
Ketika sedang meng-*upload* program, arduino bootloader sedang digunakan, Arduino bootloader adalah sebuah program kecil yang telah ditanamkan pada mikrokontroler yang berada pada arduino board. Bootloader ini memungkinkan kita meng-*upload* program tanpa menggunakan perangkat keras tambahan.

2.4. Sensor

Sensor adalah alat yang digunakan untuk mendeteksi dan sering berfungsi untuk mengukur magnitudo sesuatu. Dengan menggunakan sensor kita dapat mengubah mekanis, magnetis, panas, cahaya dan kimia menjadi tegangan dan arus listrik. Sensor yang digunakan pada sistem penelitian ini adalah sensor *Ultrasonik HY-SRF05*.

2.4.1. Ultrasonik HC-SR04

Sensor ultrasonik adalah sensor yang bekerja dengan cara memancarkan suatu gelombang dan kemudian menghitung waktu pantulan gelombang tersebut. Gelombang ultrasonik bekerja pada frekuensi mulai 20 kHz hingga 9 sekitar 20 MHz. Frekuensi kerja yang digunakan dalam gelombang ultrasonik bervariasi tergantung pada medium yang dilalui, mulai dari kerapatan rendah pada fasa gas, cair hingga padat.



Gambar 2.7: Cara Kerja Sensor Ultrasonik

Pada Gambar 2.7 digambarkan cara kerja sensor ultrasonik. Sensor ultrasonik terdiri dari rangkaian pemancar ultrasonik yang disebut *transmitter* dan rangkaian penerima ultrasonik yang disebut *receiver*. Sinyal ultrasonik yang dibangkitkan akan dipancarkan dari *transmitter* ultrasonik. Ketika sinyal mengenai benda penghalang, maka sinyal ini dipantulkan, dan diterima oleh *receiver* ultrasonik. Sinyal yang diterima oleh rangkaian *receiver* dikirimkan ke rangkaian mikrokontroler untuk selanjutnya diolah untuk menghitung jarak terhadap benda di depannya (bidang pantul).

Secara matematis gelombang ultrasonik dapat dirumuskan sebagai :

$$s = v.t/2 \dots\dots\dots (2.2)$$

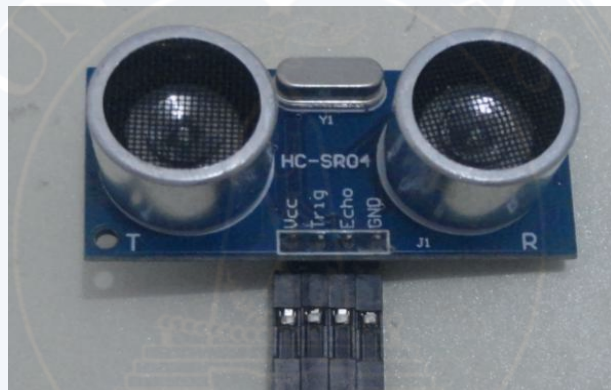
Pada rumus 2.2, s adalah jarak dalam satuan meter, v adalah kecepatan suara yaitu 340 m/detik dan t adalah waktu tempuh dalam satuan detik.

Dijabarkan:

$$s = (34000 * waktu) / 1000000 * 2 \dots\dots\dots (2.3)$$

Pada rumus 2.3, s dan d adalah jarak tempuh dalam satuan cm, waktu adalah waktu tempuh dua kali lintasan ultrasonik hingga permukaan objek dalam satuan mikrosekon maka dikonversi menjadi sekon dibagi dengan 1000000 dan 34000 adalah kecepatan suara dalam satuan cm/s.

Ketika gelombang ultrasonik menumbuk suatu penghalang maka sebagian gelombang tersebut akan dipantulkan sebagian diserap dan sebagian yang lain akan diteruskan. Gelombang yang diserap akan dihitung oleh komparator dan diteruskan menjadi bilangan *binary*.



Gambar 2.8 : Sensor Ultrasonik HC-SR04

Gambar 2.8 merupakan gambaran fisik sensor ultrasonik. Sensor ultrasonik memiliki 4 pin yaitu, pin vcc, pin ground, pin trigger sebagai *input* dan pin echo sebagai *output*. Secara umum sensor ultrasonik digunakan untuk menghitung jarak dari suatu objek yang berada di depan sensor tersebut. Sehingga dengan fungsinya tersebut, sensor ultrasonik biasa digunakan pada perangkat yang membutuhkan perhitungan jarak. Contoh : pendeteksi ketinggian air, pemantauan

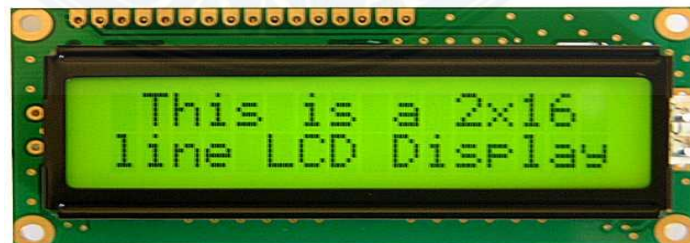
ketersediaan air pada bak penampung, memantau kecepatan kendaraan bermotor dan pendeteksi dini banjir berdasarkan ketinggian dan kecepatan air. Tabel 2.1 adalah deskripsi Ultrasonik HC-SR04.

Tabel 2. 1: Deskripsi Ultrasonik HC-SR04

Tegangan Operasi	5V
Arus DC	15mA
Frekuensi Operasi	40Hz
Maksimum Range	4m
Minimum Range	2cm
Trigger Input	10 μ s
Dimensi	45*20*12mm

2.5. LCD 2x16 (*Liquid Crystal Display*)

Berikut ini adalah Gambar 2.9 yang menampilkan bentuk fisik *LCD* 2 x 16:



Gambar2.9 :Bentuk fisik LCD 16x2

LCD merupakan penampil dengan sistem termodul, yang dapat menampilkan baik karakter, angka atau pun simbol yang tentunya lebih bagus daripada *seven*

segment. Bentuk dan ukuran beragam, salah satu variasi bentuk dan ukuran yang tersedia dan umum digunakan adalah 16x2 karakter (panjang 16, baris 2, karakter 32) dan 16 pin, yaitu delapan jalur hubungan data, tiga jalur hubungan kontrol, tiga jalur catu daya dan pada modul *LCD* dengan fasilitas *backlighting* terdapat dua jalur catu untuk *back lighting* sehingga mereka dapat ditampilkan dalam kondisi cahaya yang kecil. Berikut Tabel 2.2 adalah keterangan dari *pinLCD*:

Tabel2.2 :Keteranganpin LCD (ArdiWinoto, 2010: 194)

NO	Nama Pin	Deskripsi	Port
1	VCC	+ 5V	VCC
2	GND	0 V	GND
3	VEE	Tegangan Kontras LCD	
4	RS	Register Select, 0=Input Instruksi, 1=Input Data	PD7
5	R/W	1= Read ; 0 = Write	PD5
6	E	Enable Clock	PD6
7	D4	Data Bus 4	PC4
8	D5	Data Bus 5	PC5
9	D6	Data Bus 6	PC6
10	D7	Data Bus 7	PC7
11	Anode	Tegangan Positif backlight	
12	Katode	Tegangan Negatif backlight	

BAB III METODOLOGI PENELITIAN

3.1. Tempat dan Waktu Penelitian

3.1.1. Tempat Penelitian

Pembuatan dan pengujian alat ukur konstanta pegas ini dilakukan di :

1. Nama Tempat : Laboratorium Dasar Digital Universitas Medan
Area (UMA)
2. Alamat : Jalan Kolam No. 1 Medan Estate, Medan

3.1.2. Waktu Penelitian

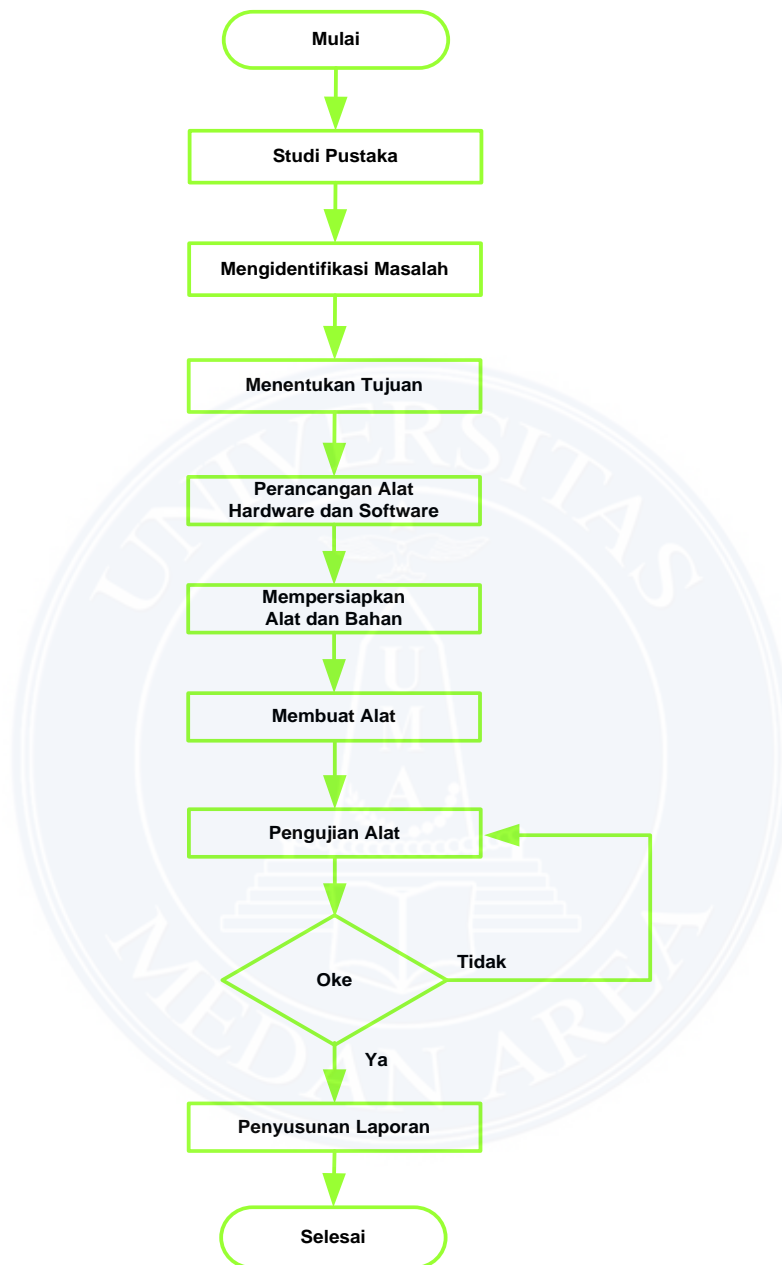
Pembuatan dan pengujian sistem ini membutuhkan waktu dengan rincian sebagai berikut :

1. Penyediaan bahan dan alat : 1 minggu
2. Perancangan dan pembuatan
seluruh sistem : 2 bulan
3. Pengujian sistem : 1 minggu
4. Penyusunan laporan Skripsi : 2 minggu

3.2. Metoda Penelitian

Pelaksanaan penelitian ini dilakukan dalam beberapa tahap untuk mempermudah dan memperjelas arah penelitian. Berikut ini adalah Gambar 3.1, yaitu *flowchart* kerangka berfikir dalam penelitian, dimana berdasarkan *flowchart*

inilah sebagai tahapan-tahapan yang dilakukan oleh peneliti dalam melakukan proses penelitian rancang bangun alat ukur konstanta pegas :



Gambar 3.1 : Flowchart kerangka berfikir

3.2.1. Alat dan Bahan

Alat-alat yang digunakan pada pembuatan sistem ini adalah : 1 set *tools mechanic*, gerinda, bor listrik, solder listrik, mistar, alat tulis. Alat-alat yang digunakan pada uji kinerja alat rancangan antara lain : multimeter, dan testpen.

Bahan elektrik maupun mekanik yang digunakan dalam pembuatan sistem ini secara umum adalah seperti pada Tabel 3.1 berikut :

Tabel 3.1 : Penetapan komponen (bahan)

No.	Komponen	No.	Komponen
1	Arduino Uno	10	Kabel pelangi
2	Sensor <i>Ultrasonik HC-SR04</i>	11	Spicer plastik
3	Kayu model persegi (2,5 cm)	12	Paku kayu
4	Acrelic	13	Plat siku
5	Kayu model persegi panjang (2,5 x 5 cm)	14	Sekrup
6	Pegas	15	LCD 2x16
7	Pilox warna hitam	16	Resistor 2K2
8	AC-DC Adaptor	17	Kawat gantungan
9	Baut	18	Kabel Downloader

3.2.2. Rancangan Struktural

a. Massa (Beban)

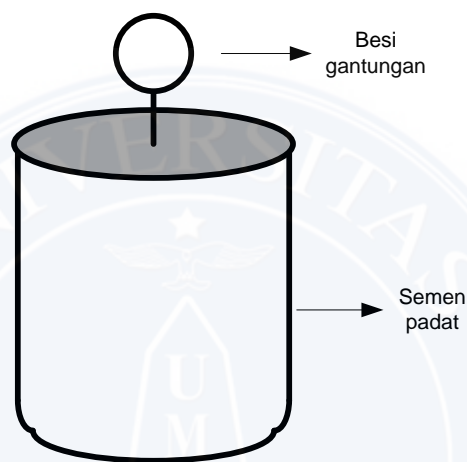
Massa berfungsi sebagai beban yang akan digantung pada ujung pegas yang akan dikur nilai konstantanya. Massa yang digunakan adalah terbuat dari semen dan berbentuk tabung silinder dengan dimensi sebagai berikut :

$$\text{Tinggi (h)} = 9,5 \text{ cm}$$

$$\text{Diameter atas (d}_{(a)}) = 8,5 \text{ cm}$$

$$\text{Diameter bawah (d}_{(b)}) = 7 \text{ cm}$$

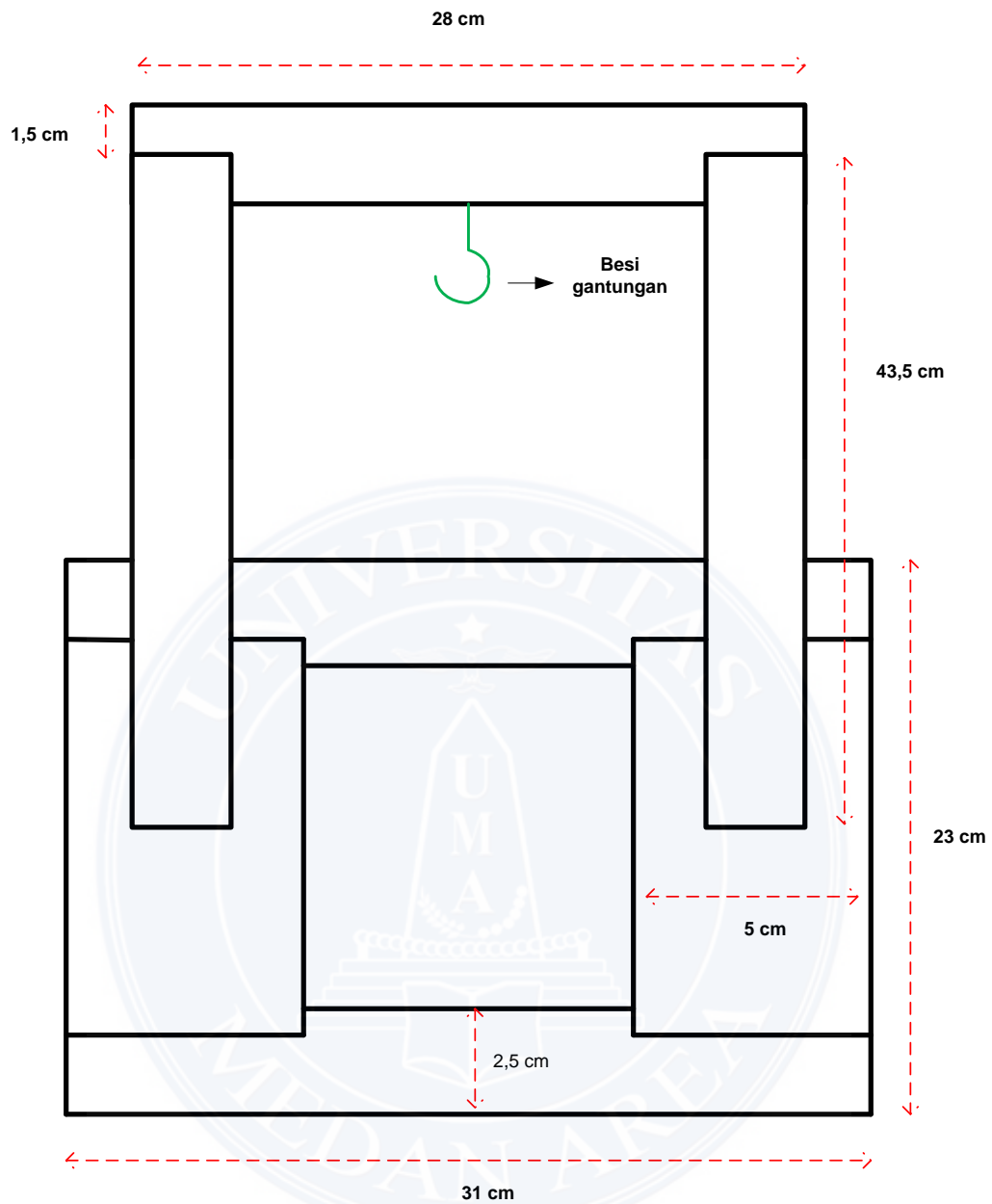
Selanjutnya wadah ini didesain bagian permukaannya dengan tujuan untuk meletakkan besi gantungan agar massa atau beban ini dapat menggantung pada pegas yang diukur konstantanya. Berikut adalah Gambar 3.2 yang menampilkan bentuk dari rancangan massa atau beban (B) pegas pada penelitian ini:



Gambar 3.2 : Massa (sebagai beban pegas)

b. Kerangka

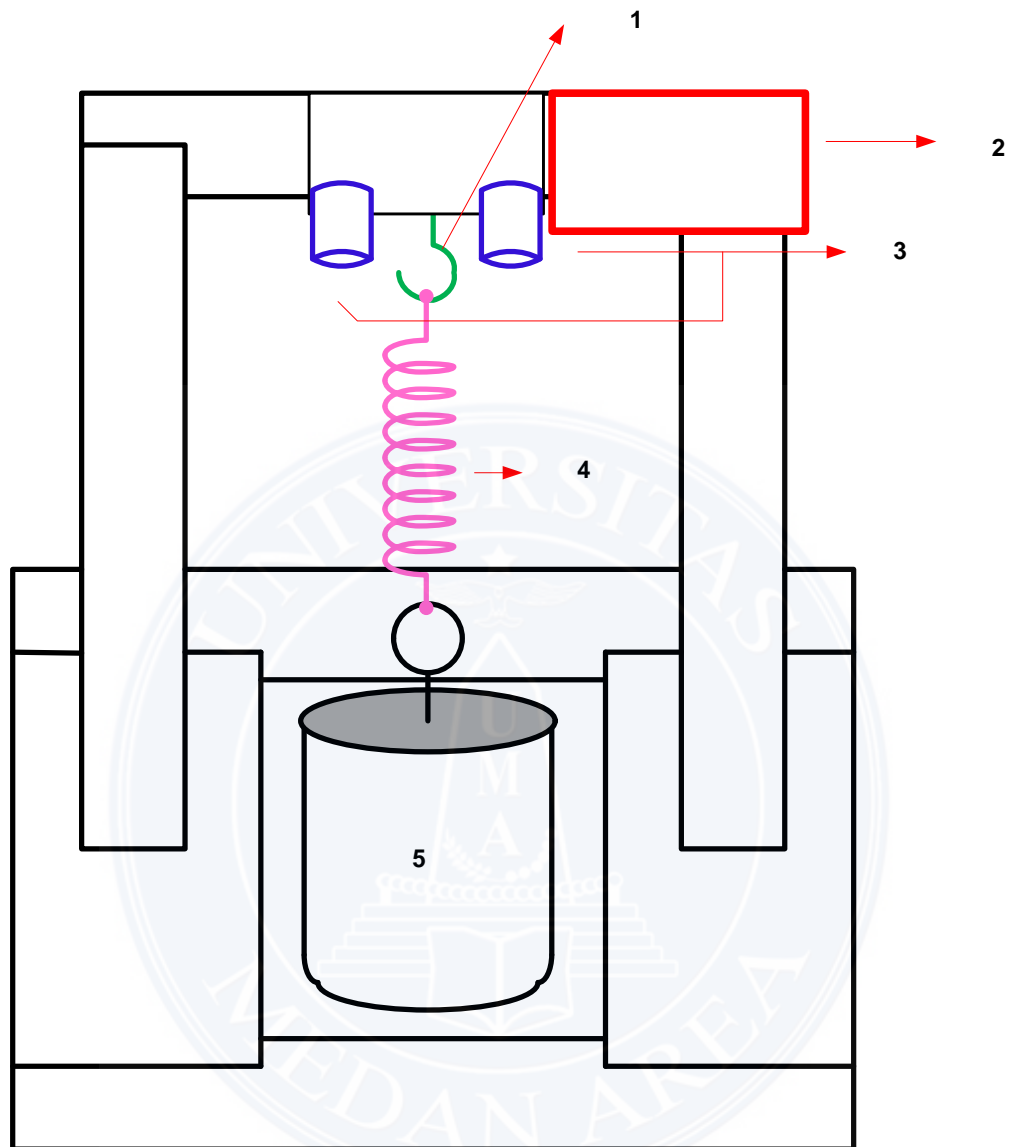
Bagian rangka berfungsi sebagai dudukan setiap sistem-sistem yang dirancang. Rangka terbuat dari kayu model persegi dengan dimensi lebar 2,5 cm, tebal 2,5 cm, kemudiannya kayu model persegi panjang tersebut dengan dimensi lebar 5 cm, tebal 2,5 cm. Kerangka yang dibuat dapat dilihat seperti pada Gambar 3.3 di bawah ini, serta lengkap dengan dimensinya :



Gambar 3.3 : Desain dan dimensi rangka

c. Rancangan Tata Letak Sistem

Berikut ini adalah Gambar 3.4 yang menampilkan tata letak sistem-sistem yang akan dibuat pada penelitian ini :



Gambar 3.4 : Desain tata letak sistem

Keterangan Gambar 3.4 :

- | | |
|------------------------------|------------------|
| 1. Besi Gantungan | 4. Pegas |
| 2. LCD 2x16 | 5. Massa (Beban) |
| 3. Sensor Ultrasonik HC-SR04 | |

3.2.3. Rancangan Sistem Elektrikal

Rancangan sistem elektrikal yang dimaksud adalah meliputi :

1. Sistem AC-DC adaptor
2. Sistem instalasi sensor ultrasonik
3. Sistem minimum Arduino Uno
4. Sistem penampil data yaitu LCD 2x16
5. Sistem secara keseluruhan

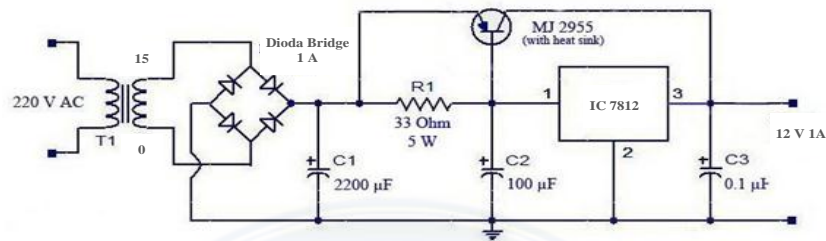
3.2.3.1. Sistem AC-DC Adaptor

AC-DC adaptor yang dirancang adalah cukup menggunakan AC-DC adaptor yang sudah jadi dan lebih simpel yaitu dengan membelinya di toko-toko penjual komponen elektronika, namun hal yang harus diperhatikan adalah spesifikasinya harus sesuai dengan kebutuhan sistem yang dirancang secara keseluruhan. Berikut adalah Gambar 3.5 yang memperlihatkan bentuk fisik AC-DC adaptor yang digunakan beserta spesifikasinya yang tertera di badan adaptor :



Gambar 3.5 : AC-DC adaptor

Pada Gambar 3.5 di atas yang memperlihatkan bentuk fisik AC-DC adaptor dapat juga dilihat skema rangkaian didalamnya seperti Gambar 3.6 di bawah ini :

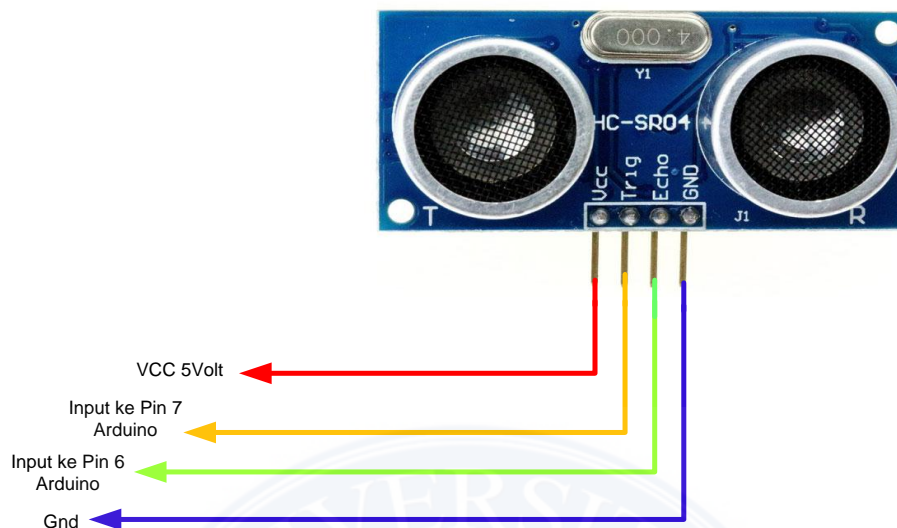


Gambar 3.6 : Skema rangkaian AC-DC adaptor

3.2.3.2. Sistem Sensor Ultrasonik HC-SR04

Sistem ini juga tidak perlu dirancang ataupun dibuat lagi karena sensor ini sudah ada yang siap pakai namun yang perlu dipahami dan diperhatikan adalah bagaimana cara kerja sensor ini dan bagaimana cara penginstalasian pin-pin sensor terhadap sistem pengendali Arduino Uno agar dapat berfungsi sesuai karakteristiknya sebagai pendeteksi ataupun pengukur jarak benda.

Berikut ini akan dijelaskan bagaimana pola penginstalasian pin sensor ultrasonik yang disajikan dalam bentuk Gambar. Berikut Gambar 3.7 yang memperlihatkan pola penginstalasiannya :



Gambar 3.7 : Pola penginstalasian sensor ultrasonik HC-SR04

Keterangan pin Gambar 3.7 :

VCC = 5V Power Supply

Pin sumber tegangan positif sensor atau pin 5V untuk koneksi ke tegangan 5V dc.

Trig = Trigger/Penyulut.

Pin ini yang digunakan untuk membangkitkan sinyal ultrasonik, dengan kata lain Trigger Input dipakai untuk memicu pembangkitan gelombang ultrasonic. Berupa sinyal 'HIGH' selama minimal 100 us.

Echo = Receive/Indikator. Pin ini yang digunakan untuk mendeteksi sinyal pantulan ultrasonik, dengan kata lain Echo Output untuk memantau kondisi logika, apakah gelombang ultrasonic sudah diterima kembali atau belum.

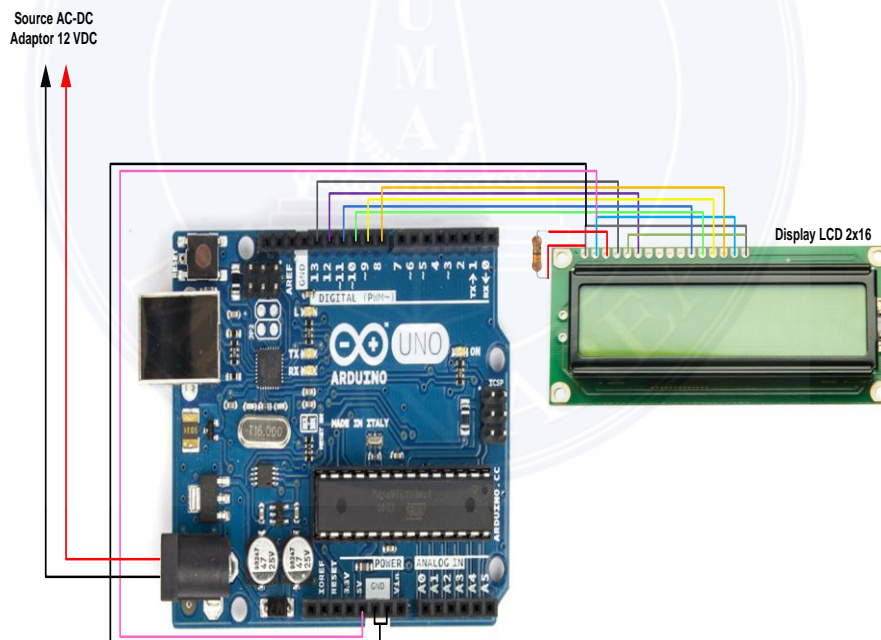
GND = Ground/0V Power Supply

Pin sumber tegangan negatif sensor.

3.2.3.3. Sistem Penampil Data (LCD 2x16)

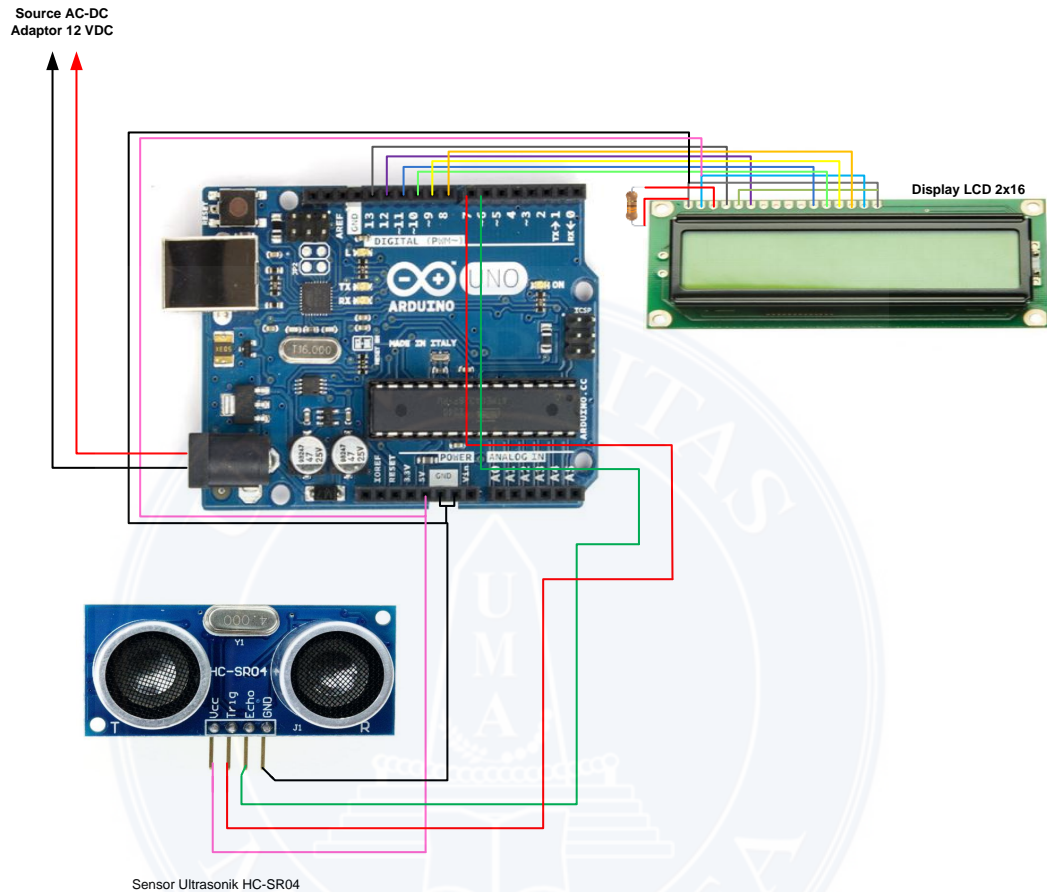
Fungsi dari sistem ini adalah sebagai penampil data berupa informasi ketersediaan nilai konstanta pegas hasil pengukuran dalam bentuk tulisan dan bukanlah dalam bentuk sinyal tegangan atau arus namun telah dirubah olehnya data tegangan menjadi suatu tulisan yang tertampil dilayarnya. Sedangkan tulisan yang tampil tergantung dari tulisan yang dibuat pada program yang dibuat.

Dalam penelitian ini perlu diketahui bagaimana pola penginstalasian LCD 2x16 terhadap pin Arduino Uno agar dapat bekerja sebagai fungsi LCD sebagai penampil data. Berikut adalah Gambar 3.8 yaitu pola penginstalasian LCD 2x16 terhadap Arduino Uno :



Gambar 3.8 : Pola instalasi LCD 2x16 pada Arduino Uno

penggabungan seluruhnya baik dari segi mekanik maupun instalasi listriknya. Berikut Gambar 3.10 yang menampilkan skema rangkaian seluruh sistem.



Gambar 3.10 : Skema rangkaian seluruh sistem

3.3. Pemrograman Mikrokontroler ATmega 328 pada Arduino

Adapun rincian program bahasa “C” yang dimasukkan pada sistem Arduino Uno adalah sebagai berikut :

```
// pin LCD
#include<LiquidCrystal.h> // library
```

```
LiquidCrystal myLcd (13,12,11,10,9,8); // set pin LCD

const float TRIG_PIN = 7;

const float ECHO_PIN = 6;

void setup()
{
myLcd.begin(16, 2);

// inialisasi komunikasi serial:
Serial.begin(9600);
pinMode(TRIG_PIN,OUTPUT);
pinMode(ECHO_PIN,INPUT);

myLcd.clear();
myLcd.setCursor(0,0);
myLcd.print("Nama: ARJUNA TT");
myLcd.setCursor(0,1);
myLcd.print("NPM: 12.812.0009");
delay(4000);
}

void loop()
{
float waktu, s, konstanta;

digitalWrite(TRIG_PIN, LOW);

delayMicroseconds(2);

digitalWrite(TRIG_PIN, HIGH);

delayMicroseconds(10);

digitalWrite(TRIG_PIN, LOW);

waktu = pulseIn(ECHO_PIN,HIGH);
```



```
s = (340 * waktu) / (1000000 * 2);
```

```
konstanta = (1.5/s);
```

```
myLcd.clear();
```

```
myLcd.setCursor(0,0);
```

```
myLcd.print("L : ");
```

```
myLcd.print(s);
```

```
myLcd.print(" meter");
```

```
myLcd.setCursor(0,1);
```

```
myLcd.print("Konst : ");
```

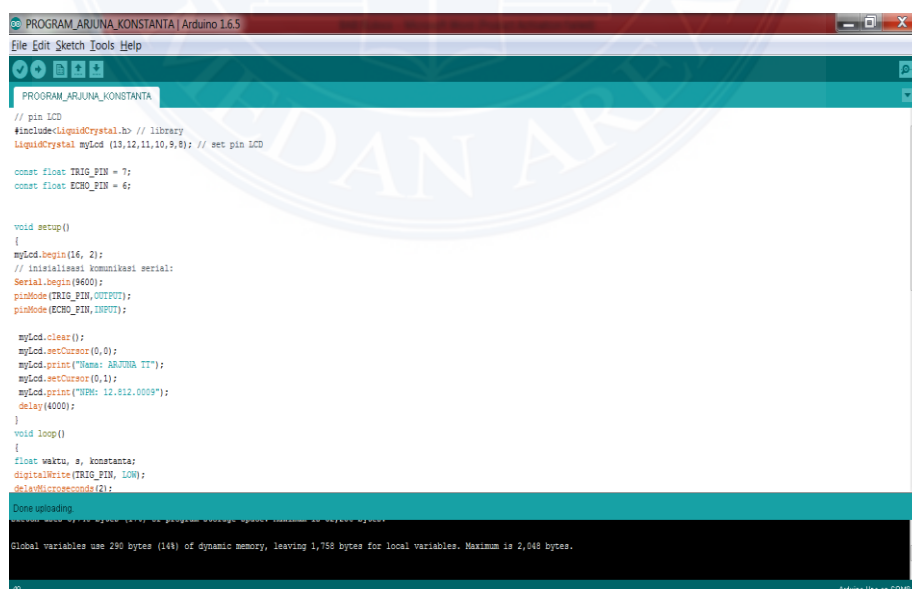
```
myLcd.print(konstanta);
```

```
myLcd.print(" N/m");
```

```
delay(500);
```

```
}
```

Gambar 3.11 berikut ini adalah gambar yang menampilkan model layar jendela aplikasi untuk menuliskan program bahasa “C” di atas.



```
PROGRAM_ARJUNA_KONSTANTA | Arduino 1.6.5
File Edit Sketch Tools Help
PROGRAM_ARJUNA_KONSTANTA
// pin LCD
#include <LiquidCrystal.h> // library
LiquidCrystal myLcd (13,12,11,10,9,8); // set pin LCD

const float TRIG_PIN = 7;
const float ECHO_PIN = 6;

void setup()
{
  myLcd.begin(16, 2);
  // Inisialisasi komunikasi serial:
  Serial.begin(9600);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);

  myLcd.clear();
  myLcd.setCursor(0, 0);
  myLcd.print("Nama: ARJUNA TI");
  myLcd.setCursor(0, 1);
  myLcd.print("Waktu: 12.812.0009");
  delay(4000);
}

void loop()
{
  float waktu, s, konstanta;
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
```

Done uploading.

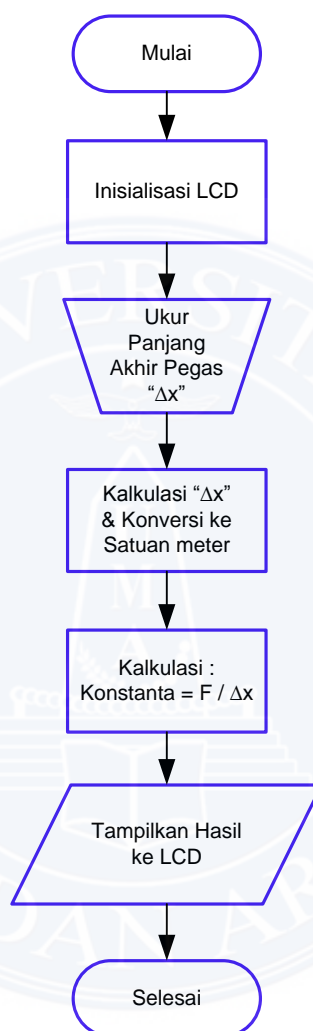
Global variables use 290 bytes (14%) of dynamic memory, leaving 1,758 bytes for local variables. Maximum is 2,048 bytes.

49 Arduino Uno on COM8

Gambar 3.11 : Jendela aplikasi penulisan program

3.4. Flowchart Sistem Kerja Alat

Berikut adalah Gambar 3.13 yang memperlihatkan alur kerja sistem alat ukur konstanta pegas :



Gambar 3.12: Flowchart sistem pada program aplikasi

DAFTAR PUSTAKA

Permana,P,2009,,"*Pembuatan Sistem Monitoring Ketinggian Air dengan Sensor Ultrasonik Berbasis Mikrokontroler Atmega "8535*,JurusanFisika,UniversitasDiponegoro.

Sasongko Hari Bagus. (2012). *Pemrograman Mikrokontroler dengan Bahasa "C"*.Yogyakarta :Penerbit ANDI.

Yuda,K,2010, "*Implementasi Ultrasonik Level Detektor pada Sistem Monitoring Tangki Pendam "pada SPBU*.TeknikElektro.ITS.

Arduino. 2014. *Arduino Uno R3*. <http://arduino.cc/en/Main/ArduinoBoardUno>. Diakses 28 Desember 2016

College, University.2013."College Physics".RiceUniversity.Texas.

<file:///C:/Users/NegaraComp/Downloads/jurnal.pdf>

file:///C:/Users/NegaraComp/Downloads/Pengenalan_ultrasonic_SRF04.pdf

Arduino. (2015). *Arduino UNO*. [online]. Tersedia di <http://www.arduino.cc/en/Main/ArduinoBoardUno>. [Diakses 28 Januari 2017].

jurnal.umrah.ac.id/wp-content/uploads/gravity_forms/1.../2014/08/jurnal-ku.pdf

Tim fisika dasar.panduan praktikum fisika dasar 1 revisi.2011.surabaya:unipress
Unisa