

ABSTRAK

BRIAN ARTHA ARDIANSYAH, PERANCANGAN PROTOTIPE SISTEM PENGONTROL KECEPATAN UDARA BERBASIS MIKROKONTROL ARDUINO UNO UNTUK APLIKASI PADA LEMARI ASAM. Skripsi. Jakarta, Program Studi Pendidikan Teknik Elektronika, Fakultas Teknik Universitas Negeri Jakarta, 2016. Dosen Pembimbing EFRI SANDI, S.Pd., MT dan MUHAMMAD YUSRO, S.Pd., MT.

Penelitian bertujuan untuk merancang, membuat, dan menguji sebuah Prototipe Sistem Pengontrol Kecepatan Udara dengan sensor angin sebagai indikatornya dan menggunakan sistem kendali Arduino Uno dengan *output* Motor *driver*, TFT LCD, *LED*, dan *buzzer* yang diprogram menggunakan program aplikasi Arduino IDE.

Penelitian ini menggunakan metode *Research and Development* yang meliputi perencanaan, perancangan, pembuatan dan pengujian alat yang dilakukan dengan pembuatan sistem alat dengan input berupa sensor angin dengan menggunakan sistem kendali arduino uno dimana bertujuan untuk mengatur kecepatan putaran kipas dan mendeteksi kecepatan udara yang melaju pada lemari dengan menggunakan sistem mikrokontroler. Pendeteksian sensor juga dapat diintegrasikan dengan output diantaranya Motor *driver*, TFT LCD, *LED*, dan *buzzer*. Data dikumpulkan dan dianalisis secara sistematis melalui pengukuran data input dan output dengan Avo Meter dan serial monitor, untuk mendapatkan data yang lebih valid.

Dari hasil percobaan yang telah dilakukan, kecepatan udara pada sistem pengontrol kecepatan udara ini adalah berkisar antara 0,4 m/s sampai dengan 0,6 m/s.

Kata Kunci : Lemari Asam, Pengontrol kecepatan udara, Arduino Uno, Sensor angin.

ABSTRACT

BRIAN ARTHA ARDIANSYAH, PROTOTYPE OF AIR VELOCITY CONTROL SYSTEM BY MICROCONTROL ARDUINO UNO FOR APPLICATION ON FUMEHOOD CABINET. Final Essay. Jakarta, Study Program of Electronics Engineering, Faculty of Engineering Universitas Negeri Jakarta, 2016. Supervisors EFRI SANDI S.Pd, MT and MUHAMMAD YUSRO, S.Pd., MT.

This research aims to design, create, and testing an Air Velocity Control System with Windcensor as an indicator and using Arduino Uno as a control system with output are motor *driver*, TFT LCD, *LED*, and *Buzzer* that programmed using Arduino IDE Application.

This research Apply R and D (Research and Development) which cover plan, design, create, and testing tools that done by creating system tool with wind censor as input using arduino uno as control system, which aims to adjust fan rotation speeds and detected airspeed that drove to the cabinet using microcontroller system. Detection censor can also be integrated with output like motor *driver*, TFT LCD, *LED*, and *Buzzer*. Data were collected and analyze systematically through measurement of input and to obtain more valid data.

From the result of experient, air speed that controlled by air velocity control system is between 0.4 m/s to 0.6 m/s.

Keyword : Fumehood Cabinet, Air Velocity Control, Arduino Uno, Wind censor.

LEMBAR PENGESAHAN SKRIPSI

Nama Dosen	Tanda Tangan	Tanggal
<u>Efri Sandi, S.Pd, MT</u> (Dosen Pembimbing I)
<u>Muhammad Yusro, S.Pd, MT</u> (Dosen Pembimbing II)

PENGESAHAN PANITIA UJIAN SKRIPSI

NAMA DOSEN	TANDA TANGAN	TANGGAL
<u>Drs.Pitoyo Yuliatmojo, MT</u> (Ketua)
<u>Drs.Wisnu Djatmiko, MT</u> (Sekretaris)
<u>Aodah Diamah, M.Eng</u> (Dosen Ahli)

Dinyatakan Lulus Tanggal : 22 JULI 2016

HALAMAN PERNYATAAN

Dengan ini saya menyatakan bahwa :

1. Skripsi ini merupakan karya asli dan belum pernah diajukan untuk mendapatkan gelar akademik sarjana, baik di Universitas Negeri Jakarta maupun perguruan tinggi lain.
2. Skripsi ini murni gagasan, rumusan dan penelitian penulis sendiri dengan arahan dosen pembimbing.
3. Skripsi ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila terdapat penyimpangan dan ketidakbenaran, maka saya bersedia menerima sanksi akademik sesuai dengan norma yang berlaku di Universitas Negeri Jakarta.

Jakarta, 24 Juni 2016

Yang Membuat Pernyataan

Brian ArthaArdiansyah

NIM: 5215097015

KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadirat Allah SWT yang memberikan limpahan rahmat dan karunia-Nya ini akhirnya penulis dapat menyelesaikan penulisan skripsi dengan judul ” *Perancangan Prototipe Sistem Pengontrol Kecepatan Udara Berbasis Mikrokontrol Arduino Uno Untuk Aplikasi Pada Lemari Asam* ”. Shalawat beserta salam semoga selalu tercurah atas Nabi Muhammad SAW, beserta keluarga, sahabat dan pengikutnya hingga akhir jaman.

Skripsi ini ditulis sebagai bagian dalam mendapatkan gelar sarjana pendidikan dan syarat penyelesaian massa studi di Jurusan Teknik Elektro Fakultas Teknik Universitas Negeri Jakarta. Bersamaan dengan ini penulis ingin mengucapkan terima kasih sebesar-besarnya kepada semua pihak yang terlibat dalam penulisan skripsi, terutama kepada yang terhormat :

1. Bapak Drs. Pitoyo Yuliatmojo, MT sebagai Ketua Program Studi Jurusan Teknik Elektronika, Fakultas Teknik, Universitas Negeri Jakarta.
2. Bapak Efri Sandi, S.pd., MT sebagai Dosen Pembimbing I yang telah memberikan bimbingan dan arahan selama menyusun skripsi ini.
3. Bapak Muhammad Yusro, S.pd., MT sebagai Dosen Pembimbing II yang telah memberikan bimbingan dan masukan serta saran demi kesempurnaan skripsi ini.
4. Bapak dan Ibu Dosen Jurusan Teknik Elektro yang tidak bisa penulis sebutkan satu persatu, yang telah memberikan ilmu yang bermanfaat bagi penulis selama masa perkuliahan di Jurusan Teknik Elektro Universitas Negeri Jakarta.
6. Ayah dan Ibu serta kakak yang telah memberikan segenap do'a, motivasi, arahan, serta nasihat yang sangat berarti.
7. Rekan-rekan seperjuangan Pendidikan Teknik Elektronika Fakultas Teknik Elektronika 2009 dan semua pihak yang telah membantu.

Penulis menyadari masih banyaknya kekurangan dalam penulisan dan penyelesaian penelitian ini, oleh karena itu kritik dan saran yang bersifat membangun dari segenap pembaca sangat diharapkan demi kemajuan dan kesempurnaan dalam penelitian ini.

Jakarta, 24 Juni 2016
Penulis,

Brian Artha Ardiansyah
NIM. 5215097015

DAFTAR ISI

ABSTRAK	i
ABSTRACT	ii
LEMBAR PENGESAHAN SKRIPSI	iii
HALAMAN PERNYATAAN.....	iv
KATA PENGANTAR.....	v
DAFTAR ISI.....	vi
DAFTAR TABEL	x
DAFTAR GAMBAR.....	xi
DAFTAR LAMPIRAN	xii
BAB I.....	1
PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Identifikasi Masalah	2
1.3. Pembatasan Masalah	3
1.4. Rumusan Masalah	4
1.5. Tujuan Penelitian.....	4
1.6. Kegunaan Penelitian.....	4
BAB II	5
KERANGKA TEORITIS DAN KERANGKA BERFIKIR.....	5
2.1. Kerangka Teoritis	5
2.1.1. Prototipe	5
2.1.2. Sistem Kontrol	6
2.1.2.1 Definisi Sistem Kontrol	6
2.1.2.2 Sistem Kontrol Lup Terbuka	7
2.1.2.3 Sistem Kontrol Lup Tertutup.....	7

2.1.3. Air Flow Control System	9
2.1.4. Lemari Asam.....	9
2.1.4.1. Definisi Lemari Asam.....	9
2.1.3.2 Sejarah Lemari Asam.....	10
2.1.3.3 Karakteristik Lemari Asam.....	11
2.1.5. Mikrokontroler	12
2.1.6. Arduino Uno.....	12
2.1.7. TFT LCD Display	14
2.1.8. Sensor	15
2.1.9. Motor <i>Driver</i>	17
2.1.10. Kipas <i>Exhaust</i>	17
2.1.11. <i>LED</i>	18
2.1.12. <i>BUZZER</i>	19
2.1.13. Catu Daya	19
2.1.14. Arduino IDE	20
2.2. Kerangka Berfikir.....	20
BAB III.....	24
METODOLOGI PENELITIAN	24
3.1. Tempat dan Waktu Penelitian	24
3.2. Metode Penelitian.....	24
3.2.1. Tahap Penelitian dan Pengumpulan Informasi	26
3.2.2. Tahap Perencanaan	26
3.2.3. Tahap Pengembangan Produk.....	27
3.2.4. Tahap Uji Coba	28
3.2.5. Tahap Perbaikan Produk	28
3.3. Rancangan Prototipe Sistem Pengontrol Kecepatan Udara.....	28

3.3.1. Perancangan Desain Lemari Asam	28
3.3.2. Perancangan Catu Daya	30
3.3.3. Perancangan TFT LCD <i>Touchscreen</i>	30
3.3.4. Perancangan Sensor Angin (<i>Wind Sensor</i>)	31
3.3.5. Perancangan Motor <i>Driver</i>	32
3.3.6. Perancangan <i>Buzzer</i> dan <i>LED</i>	32
3.3.7. Perancangan Program Arduino	33
3.4. Instrumen Penelitian	34
3.5. Teknik Analisis Data	35
3.5.1. Kriteria Pengujian Rangkaian Catu Daya	35
3.5.2. Kriteria Pengujian Sensor Angin	36
3.5.3. Kriteria Pengujian Motor <i>Driver</i>	38
3.5.4. Kriteria Pengujian Tombol <i>Touchscreen</i>	39
3.5.5. Kriteria Pengujian <i>Buzzer</i> dan <i>LED</i>	39
3.5.6. Kriteria Pengujian Kecepatan Udara pada Lemari Asam	40
BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....	41
4.1. Hasil Penelitian.....	41
4.1.1. Hasil Pengujian Rangkaian Catu Daya	41
4.1.2. Hasil Pengujian Sensor Angin	41
4.1.3. Hasil Pengujian Motor <i>Driver</i>	44
4.1.4. Hasil Pengujian Tombol <i>Touchscreen</i>	45
4.1.5. Hasil Pengujian <i>Buzzer</i> dan <i>LED</i>	45
4.1.6. Hasil Pengujian Kecepatan Udara pada Lemari Asam	46
4.2. Pembahasan	46
BAB V.....	50
KESIMPULAN DAN SARAN	50

5.1. Kesimpulan.....	50
5.2. Saran.....	50
DAFTAR PUSTAKA.....	52
LAMPIRAN.....	53
Lampiran 1. Spesifikasi dan Gambar Prototipe Lemari Asam	53
Lampiran 2. Skema Rangkaian Sistem Pengontrol Kecepatan Udara..	54
Lampiran 3. Gambar Hasil Pengujian	55
Lampiran 4. Program Arduino Uno.....	58
Lampiran 5. Datasheet Resistor Thermal.....	68
Datasheet IC L298.....	72
Datasheet Arduino Uno.....	84
Lampiran 6. Daftar Riwayat Hidup	92

DAFTAR TABEL

Tabel 2.1 Karakteristik Arduino Uno	14
Tabel 3.1 Penggunaan Pin TFT LCD Pada Arduino	31
Tabel 3.2 Kriteria Pengujian Catu Daya	36
Tabel 3.3 Pengujian Error Kecepatan Sensor Angin	36
Tabel 3.4 Kriteria pengujian Sensor Angin	37
Tabel 3.5 Kriteria Pengujian Delay Kecepatan Sensor Saat Dibuka	38
Tabel 3.6 Kriteria Pengujian Delay Saat Sensor Ditutup	38
Tabel 3.7 Kriteria Pengujian Motor <i>Driver</i>	39
Tabel 3.8 Kriteria Pengujian Button Touchscreen	39
Tabel 3.9 Kriteria Pengujian <i>Buzzer</i>	39
Tabel 3.10 Kriteria Pengujian <i>LED</i>	39
Tabel 3.11 Kriteria Pengujian Kecepatan Udara Pada Lemari Asam	40
Tabel 4.1 Hasil pengujian Catu Daya	41
Tabel 4.2 Hasil Pengujian <i>Wind Censor</i>	41
Tabel 4.3 Hasil Pengujian Sensor Angin Pada Lemari Asam	41
Tabel 4.4 Hasil Pengujian Tegangan Sensor	43
Tabel 4.5 Hasil Pengujian Delay Kecepatan Sensor Saat Dibuka	43
Tabel 4.6 Hasil Pengujian Delay Kecepatan Sensor Saat Ditutup	43
Tabel 4.7 Hasil Pengujian Motor <i>Driver</i>	44
Tabel 4.8 Hasil Pengujian Tombol Touchscreen	45
Tabel 4.9 Hasil Pengujian <i>Buzzer</i>	46
Tabel 4.10 Hasil Pengujian <i>LED</i>	46
Tabel 4.11 Hasil Pengujian Kecepatan Udara pada Lemari Asam	46

DAFTAR GAMBAR

Gambar 2.1 Sistem Kontrol Lup Terbuka	7
Gambar 2.2 Sistem Kontrol Lup Tertutup.....	8
Gambar 2.3 Sistem Kontrol Lup Tertutup Pada Prototipe Lemari Asam.....	8
Gambar 2.4 Lemari Asam (Fume Hood Cabinet)	10
Gambar 2.5 Arduino Uno.....	13
Gambar 2.6 TFT LCD	15
Gambar 2.7 Wind Sensor (Sensor Angin)	16
Gambar 2.8 Modul Rangkaian Motor <i>Driver</i>	17
Gambar 2.9 Kipas Exhaust	18
Gambar 2.10 <i>LED</i> (Light Emiting Diode)	18
Gambar 2.11 <i>Buzzer</i>	19
Gambar 2.12 Catu Daya.....	20
Gambar 2.13 Tampilan Arduino IDE	20
Gambar 2.14 Blok Diagram Sistem Pengontrol Kecepatan Udara.....	22
Gambar 2.15 Flowchart Sistem Kerja Sistem Pengontrol Kecepatan Udara	23
Gambar 3.1 Flowchart Penelitian.....	25
Gambar 3.2 Sub-Flowchart Inisialisasi.....	26
Gambar 3.3 Sub-Flowchart Pembacaan Sensor Angin.....	27
Gambar 3.4 Prototipe Lemari Asam Tampak Dari Arah Depan.....	29
Gambar 3.5 Prototipe Lemari Asam Tampak Dari Arah Samping.....	29
Gambar 3.6 Catu Daya 5 Volt.....	30
Gambar 3.7 Rangkaian Sensor Angin.....	32
Gambar 3.8 Rangkaian Motor <i>Driver</i>	32
Gambar 3.9 Rangkaian Indikator <i>Buzzer</i> dan <i>LED</i>	33

DAFTAR LAMPIRAN

Lampiran 1. Spesifikasi dan Gambar Prototipe Lemari Asam	53
Lampiran 2. Skema Rangkaian Sistem Pengontrol Kecepatan Udara.....	54
Lampiran 3. Gambar Hasil Pengujian.....	55
Lampiran 4. Program Arduino Uno	58
Lampiran 5. Datasheet Resistor Thermal.....	68
Datasheet IC L298	72
Datasheet Arduino Uno	84
Lampiran 6. Daftar Riwayat Hidup	92

BAB I

PENDAHULUAN

1.1. Latar Belakang

Sejalan dengan perkembangan teknologi yang semakin berkembang disegala aspek kehidupan, mulai dari bidang pendidikan dan juga elektronika membuat manusia mencari solusi dari tantangan teknologi yang mengacu pada sistem-sistem yang praktis dan efisien. Tujuannya adalah mempermudah pekerjaan yang pada awalnya dilakukan manual (tanpa sistem) menjadi pekerjaan yang lebih mudah dengan hasil keakuratan yang lebih baik serta efisiensi di beberapa aspek. Teknologi yang dipakai atau diterapkan harus memiliki syarat-syarat seperti dari aspek hasil kerja alat tersebut, sistematis prinsip kerja, harga, kemudahan perawatan dan kemudahan servis. Bila semua aspek tersebut ada pada alat tersebut, maka alat tersebut bisa menjadi alternatif teknologi. Salah satunya adalah perkembangan teknologi di bidang industri serta instrumentasi dan kendali, yang semakin modern dengan berbagai inovasi-inovasi yang dikembangkan.

Salah satu teknologi instrumentasi kendali yang menjadi topik menarik oleh para peneliti adalah mikrokontroler. Mikrokontroler merupakan sebuah bagian yang menjadi ciri khas dari sistem instrumentasi kendali. Berbagai jenis Mikrokontroler telah banyak dibuat dan dikembangkan untuk beragam aplikasi seperti bidang industri, sistem keamanan, pengukuran dan lain-lain. Seiring dengan kemajuan teknologi yang semakin berkembang dengan pesat, dibuktikan dengan semakin canggihnya peralatan-peralatan industri di dunia, khususnya di Indonesia, dan barang-barang hasil produksi yang sangat baik, serta berteknologi tinggi dengan bantuan mesin-mesin canggih serba otomatis.

Lemari asam yang banyak dipakai pada laboratorium-laboratorium di Indonesia masih banyak yang menggunakan sistem penghisap udara (Air Flow System) secara manual, dikarenakan harga perangkat pengontrol otomatis yang tersedia saat ini masih terbilang cukup mahal. Lemari asam dengan perangkat pengontrol udara otomatis memiliki beberapa kelebihan, salah satunya adalah efisiensi energi listrik yang dihasilkan untuk memutar kipas exhaust lebih baik daripada perangkat lemari asam tanpa kontrol udara otomatis. Perangkat sistem pengontrol kecepatan udara yang biasa digunakan di Indonesia saat ini masih didatangkan dari luar negeri, sehingga dalam biaya servis dan perawatannya juga menjadi kendala. Untuk mengatasi beberapa kendala tersebut di atas, maka dirancanglah sebuah prototipe berupa sistem pengontrol kecepatan udara menggunakan Mikrokontroler Arduino, karena Mikrokontroler Arduino bersifat Open Source yang dapat dengan bebas dikembangkan dan juga harganya yang terjangkau.

1.2. Identifikasi Masalah

Berdasarkan latar belakang yang telah diuraikan, maka masalah yang dapat diidentifikasi adalah sebagai berikut:

1. Bagaimanakah merancang dan membuat Prototipe sistem pengontrol kecepatan udara berbasis Mikrokontroler Arduino Uno untuk aplikasi pada lemari asam agar dapat bekerja dengan baik ?
2. Bagaimana menggunakan sensor angin untuk mendeteksi kecepatan angin pada prototipe sistem pengontrol kecepatan udara ini ?
3. Bagaimana menggunakan *Touchscreen* sebagai pengganti tombol ?

4. Bagaimana menggunakan motor *driver*, TFT LCD, *LED*, dan *buzzer* yang berfungsi sebagai aktuator pada prototipe sistem pengontrol kecepatan udara ini ?
5. Bagaimana menggunakan Prototipe *sistem pengontrol kecepatan udara* dengan system kendali Arduino Uno?

1.3. Pembatasan Masalah

Sesuai dengan masalah yang diidentifikasi di atas, maka penulis membatasi masalah pada Prototipe sistem pengontrol kecepatan udara berbasis Mikrokontroler Arduino Uno, sebagai berikut :

1. Membuat Prototipe sistem pengontrol kecepatan udara berbasis Mikrokontroler Arduino Uno menggunakan sensor kecepatan angin dengan beberapa output diantaranya : motor *driver*, LCD, *LED* dan *buzzer*.
2. Membuat sensor angin untuk mengukur kecepatan angin yang melewati pintu (*sash door*).
3. Membuat *touchscreen* sebagai peranti pengganti tombol manual dan tombol lampu penerangan.
4. Membuat motor *driver* sebagai pengatur kecepatan kipas *exhaust*, TFT LCD sebagai *display*, *LED* dan *buzzer* sebagai indikator alarm, *LED* sebagai lampu penerangan.
5. Membuat Prototipe sistem pengontrol kecepatan udara dengan menggunakan sistem kendali Arduino Uno.

1.4. Rumusan Masalah

Berdasarkan latar belakang masalah yang telah diuraikan, maka dalam penelitian ini perumusan masalah yang dapat di tuangkan adalah Bagaimana mendesain Prototipe sistem pengontrol kecepatan udara dengan kecepatan udara sebagai indikatornya dan menggunakan sistem kendali Arduino Uno serta dapat diintegrasikan dengan *output* motor *driver*, TFT LCD, *LED*, dan *buzzer* yang diprogram menggunakan program aplikasi Arduino IDE ?

1.5. Tujuan Penelitian

Sesuai masalah yang telah dirumuskan, maka tujuan yang hendak dicapai dari penelitian pada pembuatan skripsi ini adalah peneliti mampu merancang sebuah Prototipe sistem pengontrol kecepatan udara dengan kecepatan udara sebagai indikatornya dan menggunakan sistem kendali Arduino Uno dengan *output* Motor *driver*, TFT LCD, *LED*, dan *buzzer* yang diprogram menggunakan program aplikasi Arduino IDE.

1.6. Kegunaan Penelitian

Manfaat pada penelitian ini adalah untuk mengembangkan teknologi khususnya dalam bidang teknik Elektronika medis dan Laboraturium furniture serta dalam bidang *Safety Management System* dengan membuat suatu Prototipe sistem pengontrol kecepatan udara berbasis Mikrokontroler Arduino Uno untuk aplikasi pada lemari asam yang dapat bekerja dengan baik untuk mencapai tujuannya, dan dapat diaplikasikan pada lemari asam yang sesungguhnya.

BAB II

KERANGKA TEORITIS DAN KERANGKA BERFIKIR

2.1. Kerangka Teoritis

2.1.1. Prototipe

Definisi Prototipe Menurut Sommerville (2003: 163) Prototipe merupakan versi awal dari sistem perangkat lunak yang dipakai untuk mendemonstrasikan konsep, mencoba pilihan desain dan umumnya menemukan lebih banyak mengenai masalah-masalah dan solusinya.

Dalam Kamus Besar Bahasa Indonesia Prototipe memiliki arti model yang mula-mula (model asli) yang menjadi contoh. Jadi prototip hanya sebuah bentuk awal yang memungkinkan dapat dikembangkan menjadi skala yang lebih besar. Terdapat 3 pendekatan utama prototipe,yaitu:

- a. ***Throw-Away***, Prototipe dibuat dan dites.Pengalaman yang diperoleh dari pembuatan prototipe digunakan untuk membuat produk akhir(final), kemudian prototipe tersebut dibuang (tak dipakai).
- b. ***Incremental***, Produk finalnya dibuat sebagai komponen-komponen yang terpisah.Desain produk finalnya secara keseluruhan haya ada satu tetapi dibagi dalam komonen- komponen lebih kecil yang terpisah (independent).
- c. ***Evolutionary***, Pada metode ini, prototipenya tidak dibuang tetapi digunakan untuk iterasi desain berikutnya. Dalam hal ini, sistem atau produk yang sebenarnya dipandang sebagai evolusi dari versi awal yang sangat terbatas menuju produk final atau produk akhir.

2.1.2. Sistem Kontrol

2.1.2.1 Definisi Sistem Kontrol

Dalam proses industri, sering dibutuhkan besaran-besaran yang memerlukan kondisi atau persyaratan yang khusus, seperti ketelitian yang tinggi, harga yang konstan untuk selang waktu yang tertentu, nilai yang bervariasi dalam suatu rangkaian tertentu, perbandingan yang tetap antara 2 (dua) variabel, atau suatu besaran sebagai fungsi dari besaran lainnya. Jelas, kesemuanya itu tidak cukup dilakukan hanya dengan pengukuran saja, tetapi juga memerlukan suatu cara pengontrolan agar syarat-syarat tersebut dapat dipenuhi. Karena alasan inilah diperkenalkan suatu konsep pengontrolan yang disebut Sistem Kontrol.

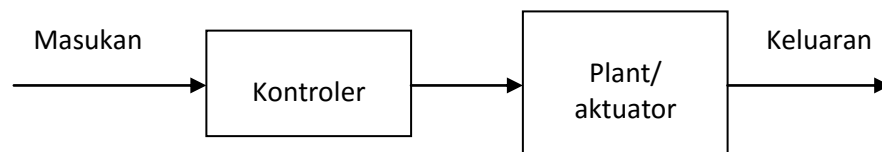
Ada beberapa definisi yang harus dimengerti untuk lebih memahami Sistem Kontrol secara keseluruhan, yaitu: Sistem, Proses, Kontrol dan Sistem Kontrol. Definisi dari beberapa istilah tersebut adalah sebagai berikut (Ogata, 1970):

1. **SISTEM:** Sistem adalah kombinasi dari beberapa komponen yang bekerja bersama-sama melakukan sesuatu untuk sasaran tertentu.
2. **PROSES:** Proses adalah perubahan yang berurutan dan berlangsung secara kontiniu dan tetap menuju keadaan akhir tertentu.
3. **KONTROL:** Kontrol adalah suatu kerja untuk mengawasi, mengendalikan, mengatur dan menguasai sesuatu
4. **SISTEM KONTROL** (*Control System*): Sistem Kontrol adalah proses pengaturan atau pengendalian terhadap satu atau beberapa besaran

(*variabel* atau *parameter*) sehingga berada pada suatu harga atau *range* tertentu. Contoh variabel atau parameter fisik, adalah: tekanan (*pressure*), (*flow*), suhu (*temperature*), ketinggian (*level*), pH, kepadatan (*viscosity*), kecepatan (*velocity*), dan lain-lain. (Dr.Ir.Andi Adriansyah, 2012).

2.1.2.2 Sistem Kontrol Lup Terbuka

Sistem control lup terbuka adalah sistem kontrol yang keluarannya tidak terpengaruh pada aksi pengontrolan. Jadi pada sistem lup terbuka, keluaran tidak diukur atau diumpam-balikkan untuk dibandingkan dengan masukan. Gambar 2.1 berikut ini menunjukkan hubungan masukan-keluaran untuk sistem control lup terbuka.

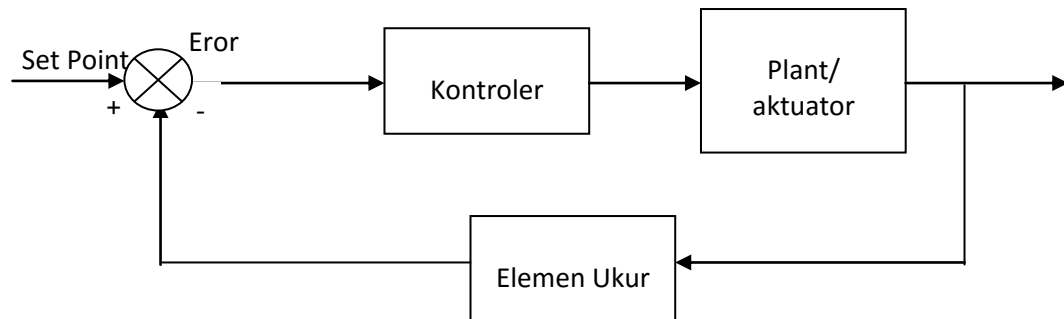


Gambar 2.1 Sistem Kontrol Lup Terbuka

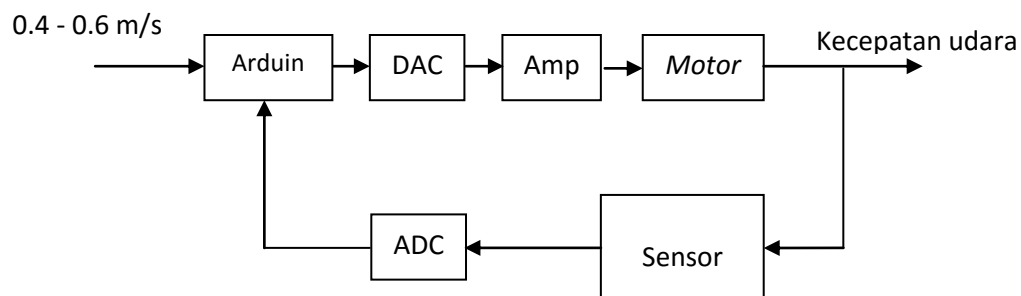
2.1.2.3 Sistem Kontrol Lup Tertutup

Sistem kontrol lup tertutup adalah sistem kontrol yang sinyal keluarannya mempunyai pengaruh langsung pada aksi pengontrolan. Jadi, sistem kontrol lup tertutup adalah sistem kontrol berumpan-balik (Ogata, 1970). Sinyal kesalahan yang merupakan selisih antara sinyal masukan dan sinyal umpan balik (yang dapat berupa sinyal keluaran dan turunannya), diumpankan ke kontroler untuk memperkecil kesalahan dan membuat agar keluaran system mendekati harga yang diinginkan. Dengan kata lain, istilah “lup tertutup” berarti menggunakan aksi umpan-balik untuk memperkecil kesalahan system. Gambar 2.2 berikut ini

menunjukkan sistem control lup tertutup dan gambar 2.3 menunjukkan hubungan masukan-keluaran dari sistem yang dirancang.



Gambar 2.2 Sistem Kontrol Lup Tertutup



Keterangan :

Set Point : 0.4 – 0.6 m/s
 Kontroler : Arduino
 Aktuator : Motor
 Elemen Ukur : Sensor

Gambar 2.3 Sistem Kontrol Lup Tertutup Pada Prototipe Lemari Asam

2.1.3. Air Flow Control System

Air Flow adalah jumlah udara per unit waktu yang mengalir melalui suatu perangkat tertentu (anonim, 1984). Jumlah udara dapat diukur dengan satuan volume atau massa. Instrumen yang mengukur jumlah udara ini disebut *Airflow meter*. Jadi, berdasarkan beberapa definisi yang telah di jabarkan pada halaman sebelumnya, dapat dijelaskan bahwa *Air Flow Control System* adalah suatu sistem pengaturan atau pengendalian jumlah udara per unit waktu (m^3/s) yang mengalir melalui suatu perangkat tertentu, yang dikendalikan oleh sebuah pengendali (mikrokontrol).

2.1.4. Lemari Asam

2.1.4.1. Definisi Lemari Asam

Lemari Asam adalah peralatan keamanan laboratorium berupa lemari / kabinet yang digunakan untuk menyimpan bahan kimia atau larutan asam yang berbahaya. Lemari asam berfungsi untuk melindungi operator / analis dari paparan asam, gas berbahaya, uap beracun, ataupun debu yang dihasilkan dari reaksi bahan-bahan kimia yang disimpan selama proses pengujian, riset, maupun pembelajaran dilaboratorium. Lemari asam biasa juga disebut dengan *fume hood cabinet*.



Gambar 2.4 Lemari Asam (*Fume Hood Cabinet*)

2.1.4.2 Sejarah Lemari Asam

Pada sekitar tahun 1900an, Thomas Alfa Edison merupakan salah seorang ilmuwan yang sudah peduli mengenai ventilasi ruang laboratorium. Dia menggunakan cerobong asap perapian dilaboratoriumnya untuk membuang gas dan bau dari hasil penelitiannya.

Di tahun 1936 Labconco, satu dari beberapa perusahaan ternama yang membuat lemari asam mulai memproduksi perangkat ini untuk komersil. Dengan bentuk sudah memakai pembatas vetrikal dan operator bisa memasukan tangannya kedalam kabinet.

Pada masa perang dunia ke II, kemajuan teknologi lemari asam semakin pesat karena tingginya kekhawatiran paparan bahan kimia beracun dan radioaktif. Sehingga dibutuhkan lemari asam yang aman. awalnya, lemari asam lebih banyak dibuat dengan bahan kayu, kemudian sejak tahun 70-an dan 80-an, secara perlahan lemari asam dengan bahan pelat yang dilapisi epoxy mulai banyak dipakai. Kemudian, pada tahun 90-an, muncul lagi bahan phenolic resin (merupakan plastik laminasi dan serat kayu) yang tahan kimia dan panas api.

2.1.4.3 Karakteristik Lemari Asam

Lemari asam yang baik setidaknya memenuhi persyaratan sebagai berikut (jasakalibrasi, 2013) :

1. Mempunyai blower yang berfungsi dengan baik
2. Kondisi pintu vertikal yang tidak terlepas atau jatuh saat dirubah posisinya
3. Terhubung dengan sumber air
4. Disertai dengan alarm yang berfungsi dengan baik jika terjadi kerusakan pada lemari asam tersebut
5. Memenuhi persyaratan ergonomis dimana tinggi lemari asam tersebut nyaman digunakan khususnya oleh orang indonesia.
6. Mudah dibersihkan sehingga selalu terbebas dari kotoran.

Dalam perencanaan dan pembuatan lemari asam harus berdasarkan standar dan aturan-aturan yang sudah ditetapkan oleh lembaga-lembaga yang mempunyai kewenangan. Tujuannya adalah menetapkan persyaratan minimal untuk mendesain sebuah lemari asam dan untuk mempertahankan udara yang cukup pada lemari asam. Sehingga lemari asam dapat digunakan dengan aman. Setiap lemari asam diuji sesuai dengan ketentuan dan standar yang memenuhi standar internasional keselamatan, kinerja, pencahayaan, efisiensi, kebisingan, ergonomis, dan estetika. Berikut adalah lembaga-lembaga atau agensi dan beberapa aturan-aturan standarnya :

- ***The National Fire and Protection Agency(NFPA)***

Section 6-4.5 states, "Face velocities of 0.4 m/sec to 0.6 m/sec (80 fpm to 120 fpm) generally provide containment if the hood location requirements and laboratory ventilation criteria of this standard are met.

- **National Research Council (NRC)**

recommends face velocities between 80 and 100fpm. 120fpm is recommended for substances with very high toxicity or where outside influences adversely influence hood performance. Face velocities approaching or exceeding 150 fpm should not be used.

2.1.5. Mikrokontroler

Menurut Syahwil (2013:53) mikrokontroler adalah sebuah sistem komputer fungsional dalam sebuah *chip*. Di dalamnya terkandung sebuah inti prosesor, memori (sejumlah kecil RAM, memori program, atau keduanya), dan perlengkapan *input-output*. Sinyal *input* mikrokontroler berasal dari sensor yang merupakan informasi dari lingkungan.

Jadi secara sederhana mikrokontroler dapat diibaratkan sebagai otak dari suatu perangkat/produk yang mampu berinteraksi dengan lingkungan sekitarnya. Mikrokontroler memiliki kelebihan dalam hal efisiensi dan efektivitas biaya untuk mengontrol berbagai macam peralatan elektronik secara otomatis seperti mainan, *remote control*, mesin kantor, peralatan rumah tangga hingga pengendalian robot.

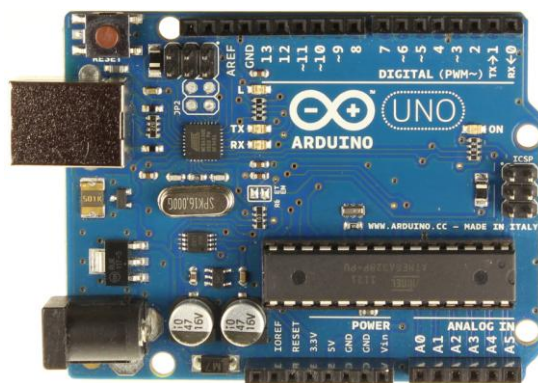
2.1.6. Arduino Uno

Arduino adalah sebuah modul komputasi yang bersifat *open source* yang didasarkan atas papan masukan / keluaran (I/O) sederhana dan juga merupakan sebuah modul pengembangan yang mengimplementasikan bahasa pemrograman (Banzi, 2008). Sedangkan menurut effendi (2014) Arduino adalah kit elektronik atau papan rangkaian elektronik *open source* yang di dalamnya terdapat komponen utama yaitu sebuah chip mikrokontroler dengan jenis AVR dari

perusahaan Atmel. Arduino bisa digunakan untuk mengembangkan objek interaktif yang berdiri sendiri atau bisa juga dihubungkan dengan *software* di komputer.

Dewasa ini arduino sangat banyak diminati oleh pemula dalam mempelajari tentang elektronika dan robotika karena bahasa yang digunakan dalam pemrograman arduino tergolong mudah, yaitu menggunakan bahasa C dan dipermudah lagi dengan penyederhanaan menggunakan bantuan *libraries*. Arduino memiliki berbagai macam tipe dengan spek dan karakteristik yang berbeda-beda.

Arduino Uno adalah mikrokontroler berbasis ATmega328 yang memiliki 14 pin digital I/O dimana 6 pin diantaranya dapat digunakan sebagai *output* PWM, sepasang pin komunikasi serial dan juga memiliki 6 pin masukan analog. Arduino Uno adalah mikrokontrol yang paling umum digunakan, terutama untuk pemula karena ukurannya tidak terlalu besar dan tidak terlalu kecil dan harganya pun relatif murah.



Gambar 2.5 Arduino Uno

Gambar 2.5 merupakan tampilan dari arduino uno dimana pada penelitian ini, peneliti menggunakan arduino uno sebagai bagian dari salah satu sistem input, yaitu pada sistem TFT LCD 2.4 inch. Hal tersebut dikarenakan TFT LCD 2.4 inch

yang dipakai membutuhkan pemroses yang berdiri sendiri dan selain itu juga karena TFT LCD 2.4 lebih kompatibel pada arduino uno. Karakteristik dari arduino uno sendiri dapat dilihat pada tabel 2.1 berikut ini.

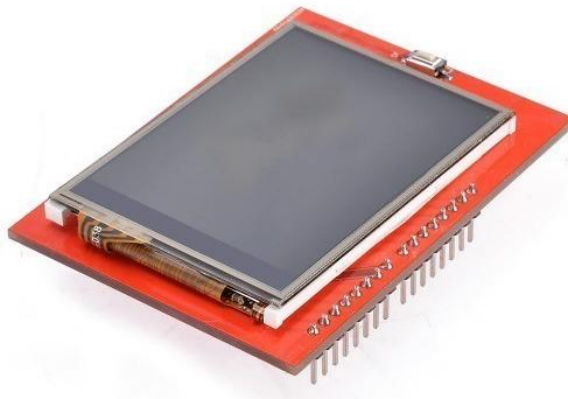
Tabel 2.1 Karakteristik Arduino Uno

NO	Name	Value
1	Microcontroller	Atmega328
2	Operating Voltage	5V
3	Input Voltage (recommended)	7-12V
4	Input Voltage (limits)	6-20V
5	Dijital Pin I/O	14 (of which 6 provide PWM output)
6	Pin Analog Input	6
7	DC Current per Pin I/O	40 mA
8	DC Current for 3.3V Pin	50 mA
9	Flash Memory	32 KB of which 0,5 KB used by bootloader
10	SRAM	2 KB
11	EEPROM	1 KB
12	Clock Speed	MHz

Sumber: www.arduino.cc

2.1.7. TFT LCD Display

TFT adalah singkatan dari *Thin Film Transistor* dan merupakan jenis teknologi yang digunakan untuk meningkatkan kualitas gambar dari LCD. Setiap piksel pada TFT-LCD memiliki transistor sendiri pada kaca itu sendiri, yang menawarkan lebih banyak kontrol atas gambar dan warna (Cassavoy, 2016). Gambar 2.6 menunjukkan bentuk gambar dari TFT LCD 2.4 inch.



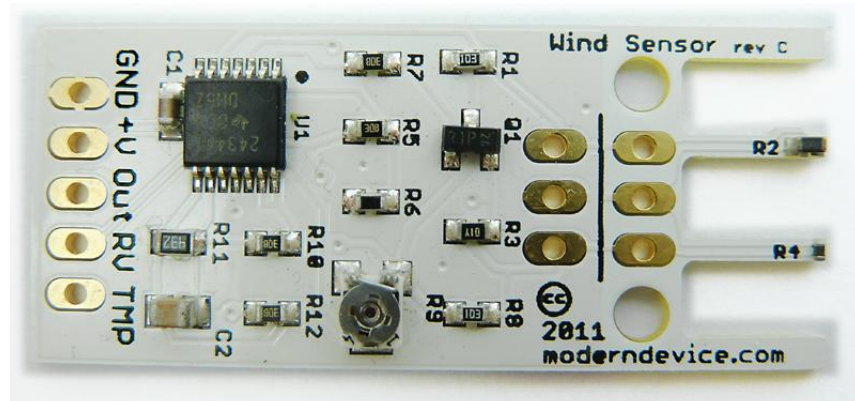
Gambar 2.6 TFT LCD

TFT LCD ini dapat digunakan sebagai *display* maupun sebagai perintah tambahan melalui *touchscreen*. *Touchscreen* adalah jenis lapisan ditempatkan pada tampilan layar yang digunakan berinteraksi dengan sentuhan pada layar. Layar sentuh (*touchscreen*) bukan jenis layar, melainkan komponen yang dapat ditambahkan ke layar yang ada. Layar sentuh menggunakan dua metode yang berbeda untuk mendapat interaksi sentuh yang disebut "resistif" dan "kapasitif," yang merujuk pada tekanan dan menyentuh sensitivitas masing-masing. Layar resistif bekerja dengan mengukur perbedaan tegangan yang disebabkan oleh jari atau stylus tekanan pada layar, sedangkan layar kapasitif bekerja dengan mengukur gangguan arus (Stone, 2016). Fungsi dari TFT LCD disini adalah sebagai tampilan kecepatan udara dan juga sebagai kontrol manual dari operator.

2.1.8. Sensor

Sensor adalah suatu peralatan yang berfungsi untuk mendeteksi gejala-gejala atau sinyal-sinyal yang berasal dari perubahan suatu energi seperti energi listrik, energi fisika, energi kimia, energi biologi, energi mekanik dan sebagainya. Contoh : kamera sebagai sensor penglihatan, telinga sebagai sensor pendengaran,

kulit sebagai sensor peraba, LDR (*light dependent resistance*) sebagai sensor cahaya, dan lainnya (Elektronika-dasar, 2015).



Gambar 2.7 Wind Sensor (Sensor Angin)

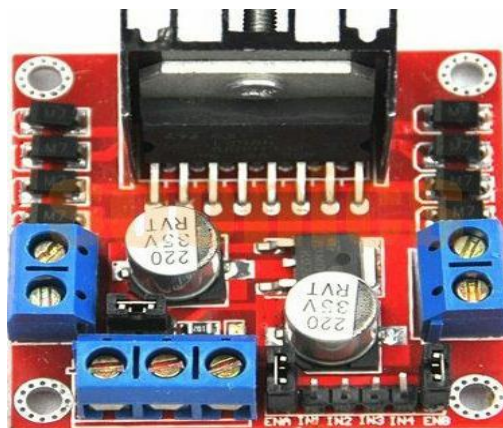
Sensor angin adalah sebuah sensor yang mampu mendeteksi adanya pergerakan udara. Sensor ini mempunyai sensitivitas yang tinggi karena mampu mendeteksi adanya pergerakan udara yang sangat kecil hingga 0,1 m/s (meter per sekon). Sensor Angin ini menggunakan resistor *thermal* sebagai komponen utamanya untuk mengukur kecepatan angin. Teknik ini disebut teknik "*hot-wire*", yang bekerja dengan cara memanaskan elemen pada suhu konstan dan kemudian mengukur daya listrik yang diperlukan untuk mempertahankan suhu ideal tersebut yang berubah karena pergerakan angin (Badger, 2011).

Spesifikasi sensor angin :

- Dimensi : 0.68 x 1.590 x 0.25 inchi
- Suplai Tegangan : 4 – 10 Volts
- Suplai Arus : 20 – 40 mA
- Sinyal keluaran : Analog, 0 - VCC

2.1.9. Motor Driver

Motor *driver* adalah sebuah modul / perangkat elektronika yang berfungsi untuk mengendalikan arah putaran sebuah motor listrik, kecepatan putaran dan menambah daya pada motor. Motor *driver* ini menggunakan sebuah IC dengan tipe L298 sebagai komponen utamanya. IC L298 mampu mengeluarkan *output* tegangan untuk motor dc dan motor *stepper* sebesar 50 Volt. IC L298 terdiri dari transistor-transistor logik (TTL) yang memudahkan dalam menentukan arah putaran suatu motor dc dan motor *stepper*. IC L298 dapat mengendalikan dua motor dc namun hanya dapat mengendalikan satu motor *stepper*. Pada penelitian ini, modul motor *driver* digunakan untuk mengatur kecepatan kipas *exhaust* agar dapat berubah sesuai dengan pembacaan sensor angin.



Gambar 2.8 Modul Rangkaian Motor Driver

2.1.10. Kipas Exhaust

Kipas *exhaust* adalah merupakan sebuah kipas yang fungsinya menyedot udara yang kotor, asap atau debu yang ada di dalam suatu ruangan agar udara yang ada didalam ruangan tersebut mengalir dan digantikan dengan udara yang bersih. Dalam pengaplikasian kipas *exhaust* di suatu ruangan haruslah memiliki celah atau ventilasi supaya udara dapat mengalir. Pengaplikasian kipas *exhaust*

banyak kita temukan dalam kehidupan sehari-hari, biasanya terdapat di dapur, kamar mandi dan ruangan-ruangan yang membutuhkan udara yang selalu segar. Dalam penelitian ini kipas *exhaust* digunakan untuk menyedot / mengalirkan udara berbahaya yang ada di dalam kabinet agar tidak menyebar kedalam ruangan. Berikut ini adalah gambar dari kipas *exhaust*.



Gambar 2.9 Kipas Exhaust

2.1.11. LED

Light Emitting Diode atau sering disingkat dengan *LED* adalah komponen elektronika yang dapat mengeluarkan cahaya. *LED* merupakan keluarga Dioda yang terbuat dari bahan semikonduktor. Perbedaan jenis semikonduktor akan menghasilkan warna cahaya yang berbeda pula. Gambar 2.10 menunjukkan gambar dari *LED (Light Emitting Diode)*.



Gambar 2.10 LED (*Light Emitting Diode*)

2.1.12. BUZZER

Buzzer adalah sebuah komponen elektronika yang berfungsi untuk mengubah getaran listrik menjadi getaran suara. *Buzzer* akan bekerja jika mendapatkan tegangan antara 5 Volt sampai dengan 12 Volt DC.

Prinsip kerja *buzzer* pada dasarnya hampir sama dengan *loud speaker*, perbedaannya *buzzer* hanya beroperasi untuk satu frekuensi tertentu. *Buzzer* biasa digunakan sebagai indikator bahwa proses telah selesai atau terjadi suatu kesalahan pada sebuah alat (*alarm*). Gambar 2.11 menunjukkan bentuk fisik dari *buzzer*.



Gambar 2.11 Buzzer

2.1.13. Catu Daya

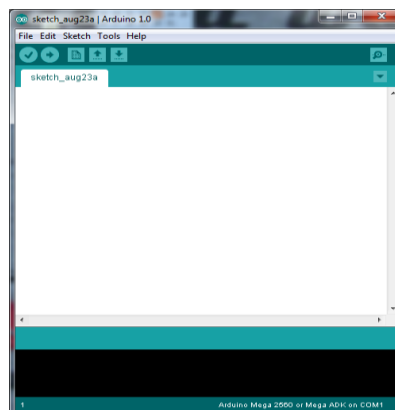
Catu daya adalah sumber tenaga yang digunakan untuk menyuplai tegangan suatu komponen elektronik maupun rangkaian elektronik. Catu daya juga sering disebut sebagai *Power Supply* atau sumber tegangan atau adaptor. Catu daya yang digunakan dalam suatu rangkaian elektronik bisa memiliki lebih dari satu besaran tegangan. Catu daya yang digunakan pada sistem pengontrol kecepatan udara ini yang memiliki sebuah besaran tegangan yaitu 5VDC. Catu daya 5VDC dibutuhkan untuk mensuplai arduino, sensor dan kipas *exhaust*.



Gambar 2.12 Catu Daya

2.1.14. Arduino IDE

Arduino IDE merupakan *software* khusus untuk memprogram *board* arduino dengan bahasa C sebagai dasar pemrograman. Arduino IDE dibuat khusus untuk memudahkan dalam pembuatan *sintaks* program arduino yang sifatnya *open source* dengan menyediakan berbagai *library* yang dapat di *download* secara gratis di situs resmi arduino, yaitu www.arduino.cc. Gambar 2.13 menunjukkan gambar dari tampilan Arduino IDE.



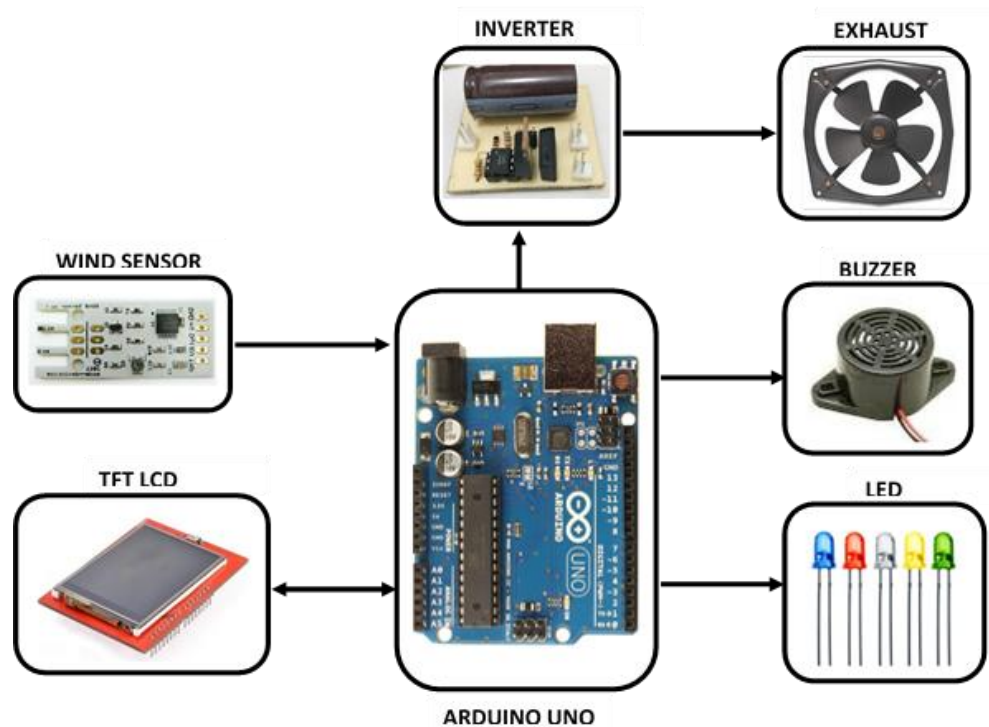
Gambar 2.13 Tampilan Arduino IDE

2.2. Kerangka Berfikir

Cara kerja dari sistem pengontrol kecepatan udara ini adalah, udara atau uap kimia yang ada di dalam ruang lemari asam akan dihisap oleh kipas *exhaust* dengan kecepatan minimal yang sudah ditetapkan yaitu 0,4 – 0,6 m/s (NFPA,

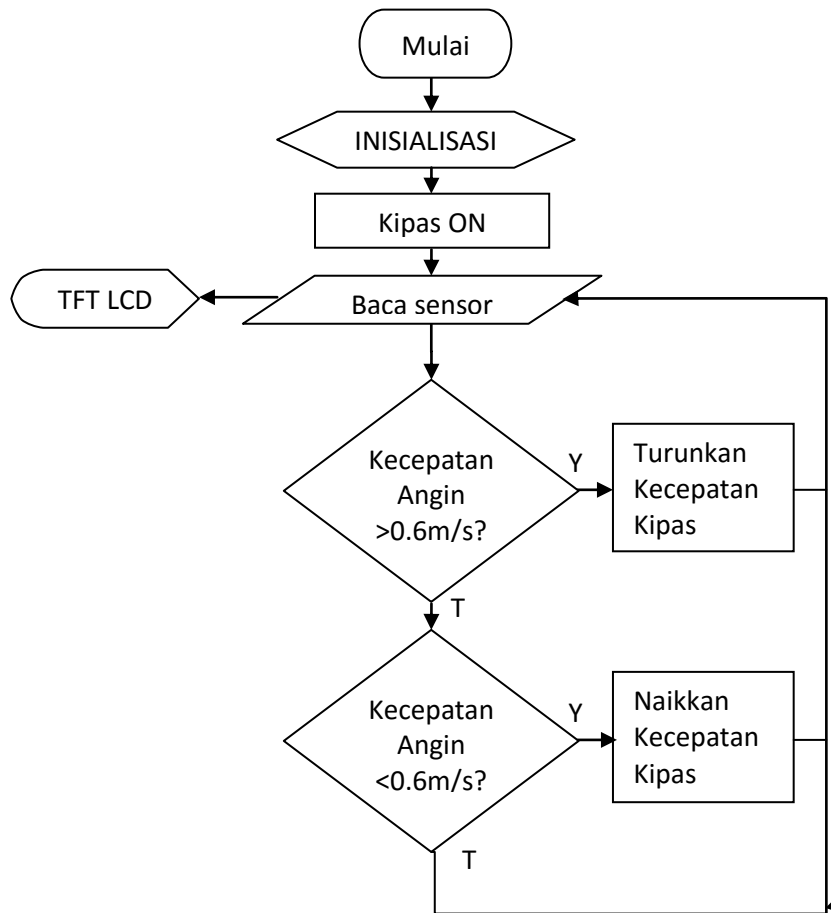
2002). Arduino akan membaca data kecepatan udara pada permukaan pintu, kemudian data yang diperoleh tersebut akan diolah. Apabila data kecepatan yang diperoleh kurang dari batas standar 0,4 m/s maka arduino akan membunyikan *alarm* dan memerintahkan kepada motor *driver* untuk mempercepat laju putaran kipas *exhaust*, sehingga diperoleh data yang stabil yaitu 0,4 m/s – 0,6 m/s. Begitu pula sebaliknya, jika data kecepatan yang diperoleh lebih besar dari 0,6 m/s maka arduino akan memerintahkan kepada motor *driver* untuk mengurangi laju putaran kipas *exhaust* sehingga kembali diperoleh data yang stabil. pada penelitian kali ini peneliti menggunakan kecepatan batas atas 0,6 m/s dengan asumsi agar mendapat efisiensi daya listrik yang lebih baik, sehingga dapat menghemat pemakaian listrik untuk perangkat ini.

Apabila dalam keadaan darurat atau operator menginginkan kecepatan hisapan kipas yang lebih besar maka dapat menggunakan tombol *MANUAL MODE* yang ada di menu layar. Manual mode akan memerintahkan kepada motor *driver* untuk memutar kipas *exhaust* dengan kecepatan penuh tanpa dipengaruhi sensor.



Gambar 2.14 Blok Diagram sistem pengontrol kecepatan udara

Dalam blok proses ini, menggunakan sistem kendali Arduino Uno yang berfungsi untuk memproses data yang dikirim oleh perangkat *input* untuk kemudian diolah dan dikeluarkan melalui *port output* untuk menggerakkan peralatan *output*. TFT LCD dalam blok diatas mempunyai fungsi sebagai *input* melalui *touchscreen* dan juga sebagai *output* tampilan.



Gambar 2.15 Flowchart Sistem Pengontrol Kecepatan Udara

Dalam *flowchart* pada gambar 2.15 menunjukkan sistem kerja sistem pengontrol kecepatan udara berbasis Arduino Uno. Diawali dengan pembacaan sensor lalu dilanjutkan dengan proses oleh arduino dan diakhiri dengan perubahan kecepatan *exhaust*.

BAB III

METODOLOGI PENELITIAN

3.1. Tempat dan Waktu Penelitian

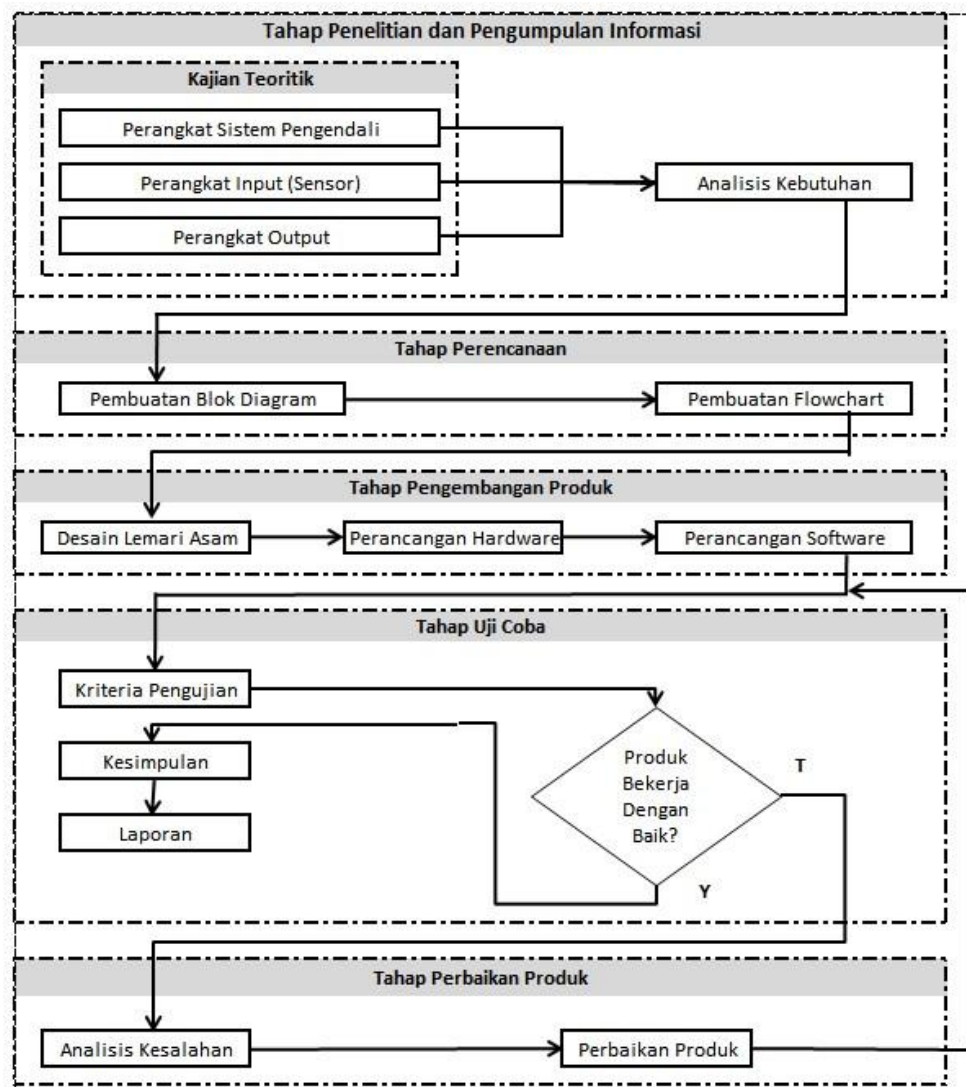
Adapun untuk tempat penelitian ini dilaksanakan di Program Studi Pendidikan Teknik Elektronika Fakultas Teknik Universitas Negeri Jakarta dengan waktu penelitian yang dilaksanakan pada bulan November 2015 sampai Juni 2016.

3.2. Metode Penelitian

Metodologi penelitian adalah langkah-langkah yang dilakukan dalam penelitian ilmiah yang bertujuan untuk mendapatkan hasil sehingga tujuan dari penelitian tersebut dapat terpenuhi. Untuk metode penelitian ini peneliti menggunakan metode *Research and Development* (R&D). Menurut Sugiyono (2009:407) metode *Research and Development* adalah metode penelitian dan pengembangan yang digunakan untuk menghasilkan produk tertentu, dan menguji keefektifan produk tersebut.

Untuk langkah – langkah dalam penelitian, Borg & Gall (1983: 775) mengungkapkan bahwa siklus *Research and Development* tersusun dalam beberapa langkah penelitian sebagai berikut: penelitian dan pengumpulan informasi, perencanaan, pengembangan produk awal, ujicoba pendahuluan, perbaikan produk operasional, ujicoba operasional, perbaikan produk akhir, diseminasi dan pendistribusian.

Langkah-langkah dalam penelitian ini mengacu pada langkah-langkah yang dikemukakan oleh Borg & Gall yang kemudian dimodifikasi oleh peneliti menjadi lima tahap yaitu tahap penelitian dan pengumpulan informasi (*research and information collecting*), tahap perencanaan (*planning*), tahap pengembangan produk (*develop of product*), tahap ujicoba (*field test*) dan tahap perbaikan produk (*product revision*). Langkah-langkah jelasnya dapat dilihat pada gambar dibawah ini:



Gambar 3.1 Flowchart Penelitian

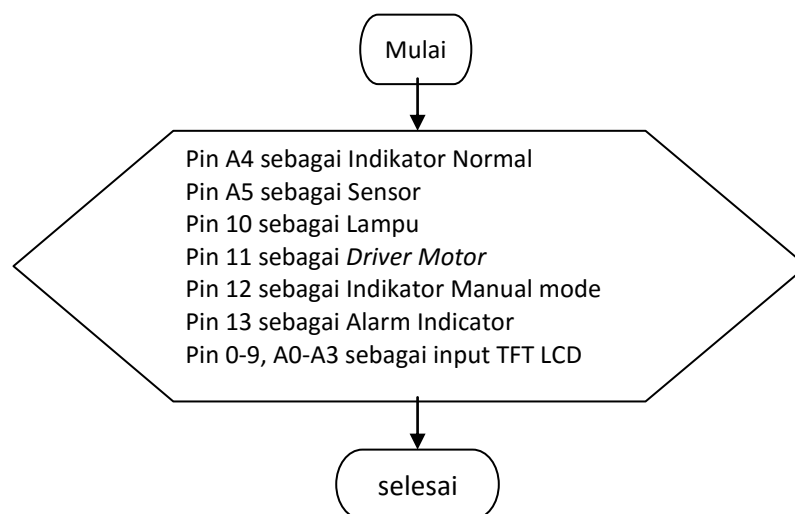
3.2.1. Tahap Penelitian dan Pengumpulan Informasi

Tahap penelitian dan pengumpulan informasi disini merupakan analisis kebutuhan sistem. Kebutuhan suatu sistem pada umumnya yaitu perangkat *input* dan *output* yang digunakan dalam sistem tersebut.

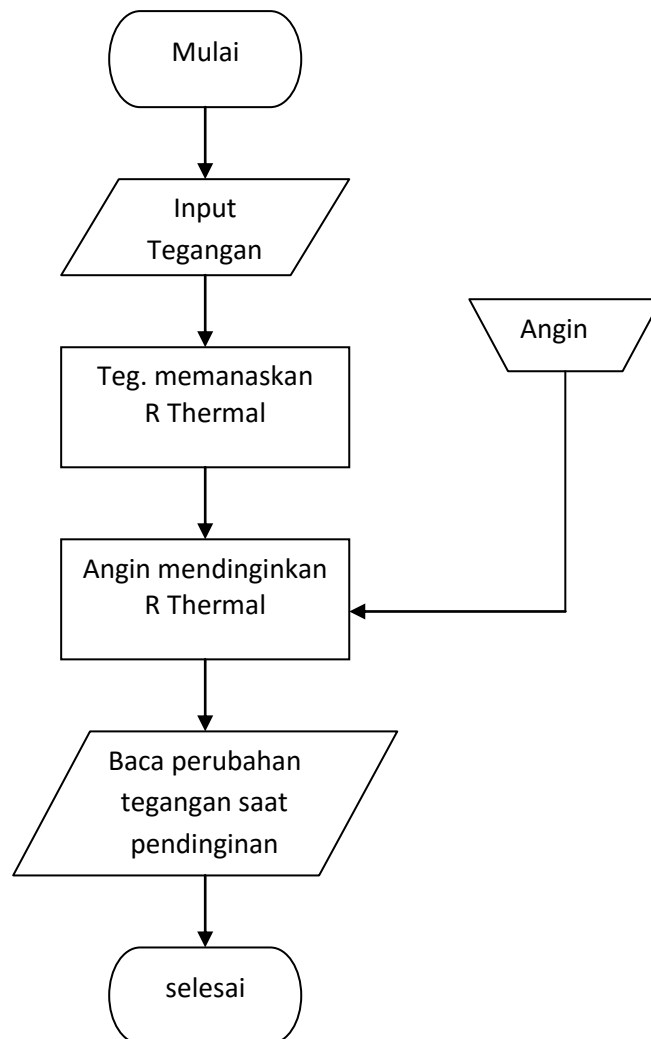
Pada penelitian ini, perangkat-perangkat yang dibutuhkan untuk membuat Prototipe sistem pengontrol kecepatan udara telah dijelaskan pada bab sebelumnya yang meliputi perangkat pendukung sistem pengendali yaitu Arduino Uno, sebuah sensor angin, *touchscreen* dan perangkat out yang digunakan dalam prototipe.

3.2.2. Tahap Perencanaan

pada tahap ini berisi kerangka berpikir peneliti dalam pembuatan prototipe sistem pengontrol kecepatan udara yang telah dijelaskan pada bab sebelumnya meliputi blok diagram yang dapat dilihat pada gambar 2.11, dan *flowchart* keseluruhan sistem yang terdapat pada gambar 2.12. Pada tahap perencanaan ini akan dijelaskan cara kerja masing masing bagian system dengan sub-routine dari *flowchart* keseluruhan sistem yang terdapat pada pada bab II. Gambar 3.2, 3.3, berikut adalah Sub-Flowchart dari bagian Inisialisasi dan Pembacaan Sensor.



Gambar 3.2 Sub-Flowchart Inisialisasi



Gambar 3.3 Sub-Flowchart Pembacaan Sensor Angin

3.2.3. Tahap Pengembangan Produk

Tahap pengembangan produk adalah merupakan tahap perancangan sistem dari prototipe sistem pengontrol kecepatan udara yang akan dibuat yang meliputi perancangan desain Lemari asam, perancangan sensor, perancangan *driver* dan perancangan perangkat lunak.

3.2.4. Tahap Uji Coba

Tahap uji coba dilakukan setelah prototipe selesai dibuat. Pada tahap ini dilakukan pengujian kelayakan cara kerja Lemari asam untuk mengetahui kelayakan produk yang dihasilkan sesuai dengan kriteria yang telah ditentukan.

3.2.5. Tahap Perbaikan Produk

Tahap perbaikan produk dilakukan ketika hasil uji coba tidak sesuai dengan perencanaan yang bertujuan untuk mencari kesalahan dan kekurangan pada sistem agar dapat diperbaiki sehingga mendapatkan hasil yang sesuai dengan perencanaan dan kriteria yang telah ditentukan.

3.3. Rancangan prototipe sistem pengontrol kecepatan udara

Rancangan penelitian ini merupakan suatu rencana atau gagasan yang mempunyai suatu tujuan yang terarah dalam melakukan suatu penelitian untuk menghasilkan atau meningkatkan karya yang dibuat.

3.3.1. Perancangan Desain Lemari Asam

Pada rancangan desain prototipe lemari asam harus bersifat ergonomis dan fleksibel sehingga dapat memudahkan operator dalam pengoperasiannya. Untuk Spesifikasi prototipe alat ini adalah sebagai berikut :

Bahan dan Dimensi : PVC (*polyvinyl chloride*) 30x30x50 cm

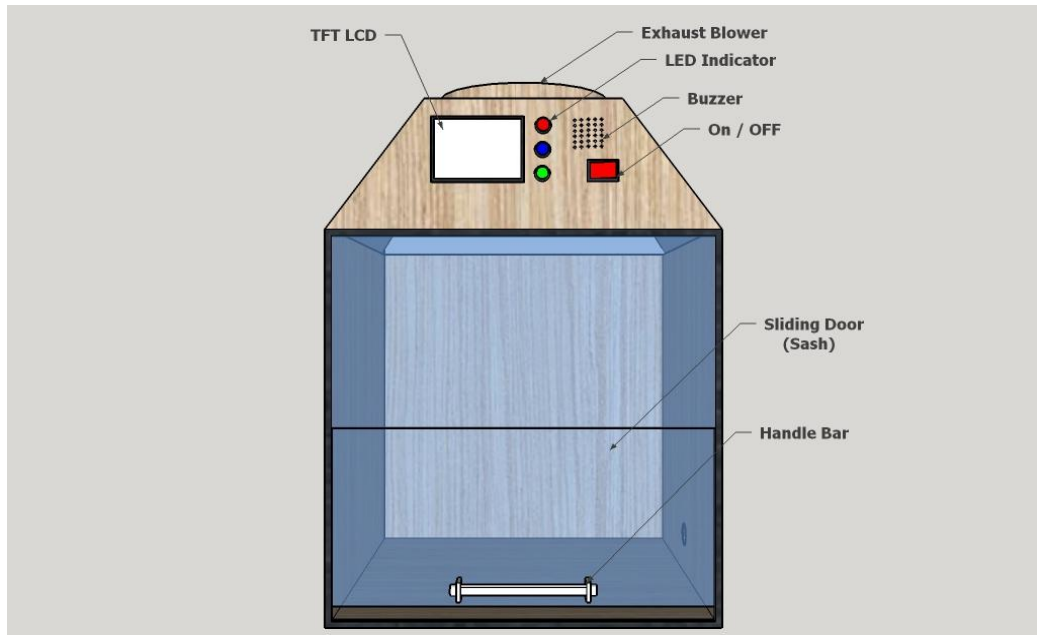
Volume : 430 m³

Tegangan Kerja : 220 VAC

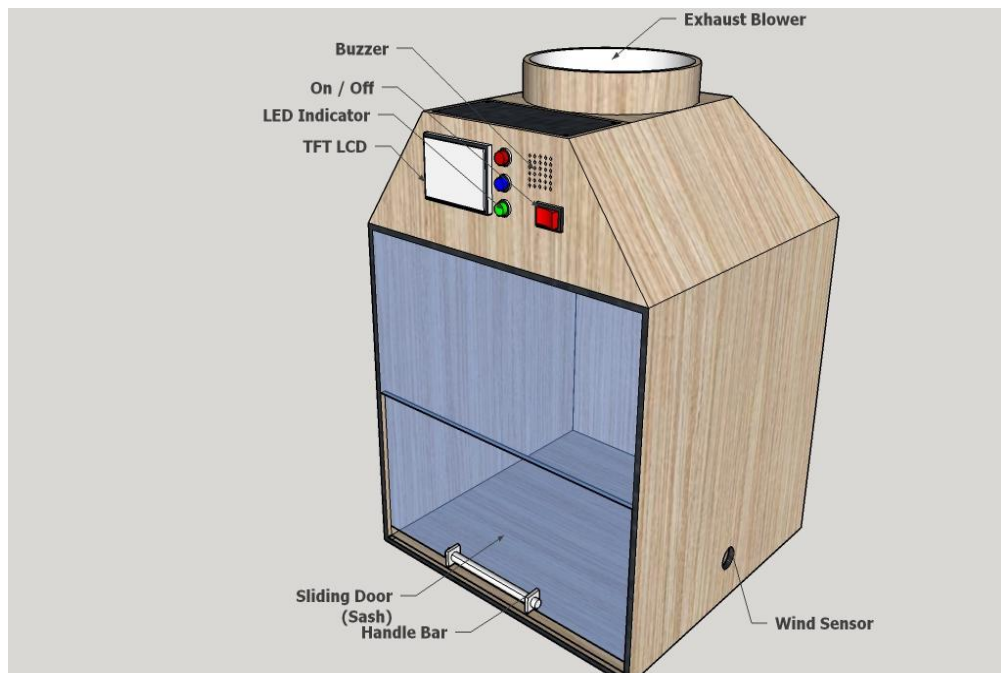
Kecepatan Minimum Angin : 0,4 meter per detik

Kecepatan Maksimum Angin : 2,6 meter per detik (saat manual mode)

Suhu Ideal Ruangan : 18 – 21 °C



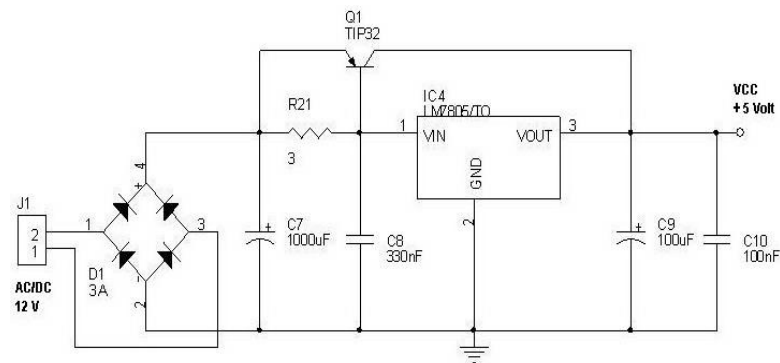
Gambar 3.4 Prototipe Lemari Asam Tampak Dari Arah Depan



Gambar 3.5 Prototipe Lemari Asam Tampak Dari Arah Samping

3.3.2. Perancangan Catu Daya

Pada penelitian ini, peneliti menggunakan Catu daya untuk mensuplai tegangan ke setiap rangkaian dan komponen utama atau pun pendukungnya yang digunakan dalam prototipe ini. Catu daya yang digunakan pada prototipe ini adalah 5 Volt DC. Tegangan 5 Volt ini digunakan untuk mensuplai tegangan pada Arduino, sensor, dan digunakan untuk mensuplai kipas *exhaust*. Untuk menghasilkan tegangan 5 Volt DC diperlukan sebuah rangkaian khusus untuk menghasilkan arus listrik yang stabil. Berikut adalah gambar rangkaian dari catu daya 5 Volt.



Gambar 3.6 Catu Daya 5 Volt

3.3.3. Perancangan TFT LCD *Touchscreen*

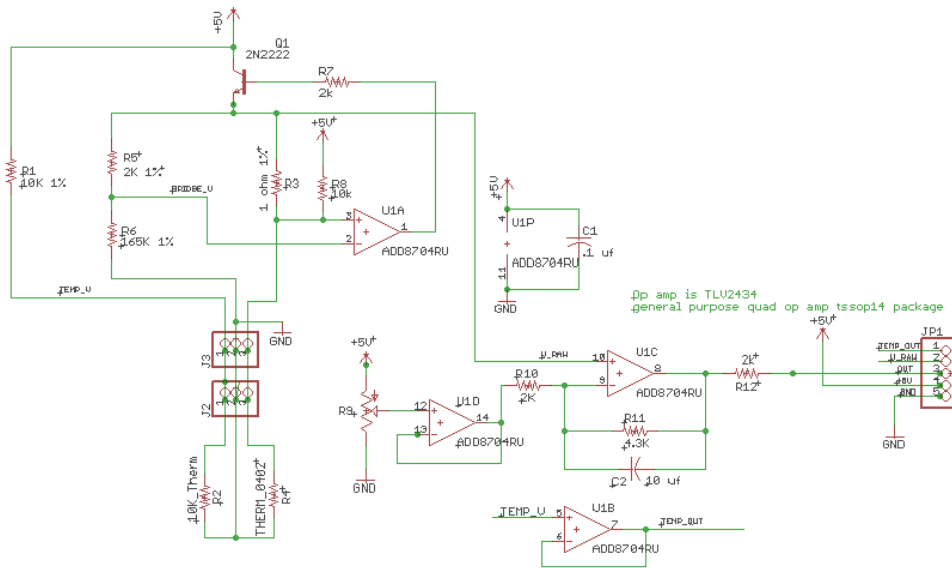
TFT LCD *Touchscreen* yang digunakan pada penelitian ini adalah jenis TFT LCD *shield* dari Arduino Uno, sehingga dalam menghubungkan antara layar TFT LCD ini dengan mikrokontroler Arduino tidak diperlukan lagi sambungan kabel atau jumper. Untuk penggunaan pin pada Arduino, dapat dilihat pada tabel 3.1 berikut ini.

Tabel 3.1 Penggunaan Pin TFT LCD Pada Arduino Uno

Pin Arduino Uno	Pin Perangkat
A0	LCD_CS
A1	LCD_RS
A2	LCD_WR
A3	LCD_RD
2	LCD_D2
3	LCD_D3
4	LCD_D4
5	LCD_D5
6	LCD_D6
7	LCD_D7
8	LCD_00
9	LCD_01
10	Lampu
11	Motor <i>Driver</i>
12	Indikator Manual Mode
13	Indikator Alarm
A4	Indikator Normal Mode
A5	Sensor

3.3.4. Perancangan Sensor Angin (*Wind Sensor*)

Modul sensor angin MD0550 bekerja dengan 3 pin, *VCC*, *Ground* dan *Output* keluaran untuk data. Keluaran yang diperoleh dari sensor angin ini adalah berupa tegangan 0 – *VCC*. Perubahan tegangan output yang terjadi akibat adanya pergerakan angin akan dibaca oleh Arduino kemudian diproses menjadi perintah kepada *driver* untuk mempercepat atau memperlambat laju putaran kipas. Berikut adalah gambar rangkaian dari sensor angin MD0550.

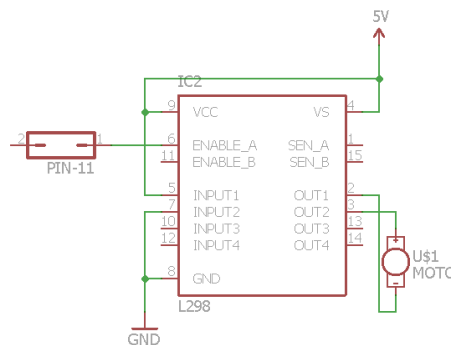


License: CC sharealike license
<http://creativecommons.org/licenses/by-nc-sa/3.0/us/>

Gambar 3.7 Rangkaian Sensor Angin

3.3.5. Perancangan Motor Driver

Rangkaian *driver* pada prototipe sistem pengontrol kecepatan udara ini menggunakan sebuah komponen yang penting, yaitu IC L298. IC ini berfungsi untuk mengkopel daya listrik pada motor yang tidak dapat dilakukan oleh Arduino secara langsung. Gambar 3.9 merupakan rangkaian dari motor *driver*

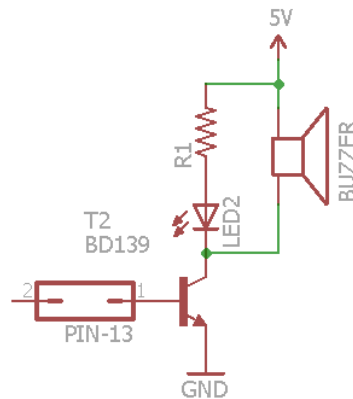


Gambar 3.8 Rangkaian Motor Driver

3.3.6. Perancangan Buzzer dan LED

Rangkaian indikator yang dipakai dalam penelitian ini adalah *Buzzer* dan *LED*. Posisi *Buzzer* dengan *LED* sengaja di pasang secara parallel agar *Buzzer* dan

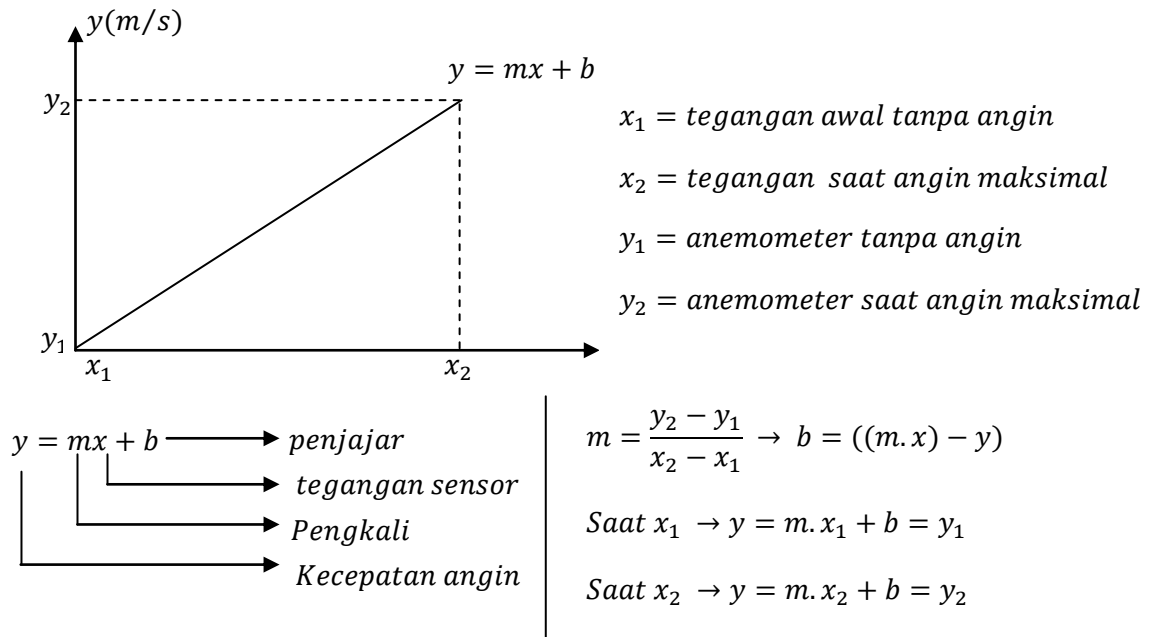
LED dapat bekerja secara bersamaan. Rangkaian ini menggunakan transistor bekerja sebagai *pull up* karena arus dan tegangan yang dihasilkan Arduino kurang mencukupi untuk mengaktifkan *Buzzer* dan *LED* secara bersamaan.



Gambar 3.9 Rangkaian Indikator *Buzzer* dan *LED*

3.3.7. Perancangan Program Arduino

Sebelum merancang program Arduino ini, terdapat beberapa bagian penting yang harus dipahami, salah satunya dalam proses penjumlahan atau menentukan kebenaran nilai penunjukan sensor (*wind censor*) terhadap standar ukur (*anemometer*). Karena sensor yang digunakan dalam penelitian ini menghasilkan perubahan tegangan setiap terjadi perubahan kecepatan udara yang diukur dengan anemometer maka perlu ditentukan persamaan linearnya. Untuk menentukan persamaan linearnya, maka harus diketahui nilai batas atas dan batas bawah dari masing masing parameter (sensor dan anemometer) kemudian dimasukkan kedalam persamaan berikut :



Dalam hal ini parameter yang diukur adalah tegangan output sensor dan kecepatan yg dibaca oleh anemometer. Perubahan tegangan yang dibaca oleh mikrokontrol akan dikonversikan menjadi kecepatan udara dalam meter per sekon.

3.4. Instrumen Penelitian

Untuk mempermudah dalam mendapatkan data yang akurat dan presisi, maka diperlukan instrumen penelitian. Instrumen yang digunakan dalam penelitian ini terdiri dari:

1. *Software* Pendukung
 - a. *Software* Arduino IDE yang digunakan untuk memprogram papan *Arduino*.
 - b. *Software* Eagle yang digunakan untuk membuat gambar skematik dan *layout* rangkaian pada PCB.
 - c. *Software* Sketch Up PRO yang digunakan untuk mendesain gambar prototipe

- d. Microsoft Word 2007 yang digunakan untuk penulisan.
2. *Hardware* Pendukung
- a. Laptop dan Komputer yang digunakan untuk membuat dan mentransfer program.
 - b. Kabel *downloader* yang digunakan untuk men-*download* program kedalam modul Arduino dari laptop atau computer.
 - c. Las Uap yang digunakan untuk membuat desain prototipe berbahan PVC.
 - d. Solder Listrik.
3. Alat Ukur
- a. AVO Meter yang digunakan untuk mengukur besar tegangan, arus, hambatan, dan mengecek komponen maupun jalur rangkaian.
 - b. Serial Monitor Arduino IDE 1.6, digunakan untuk mengukur tegangan yang masuk kedalam Arduino melalui serial monitor.
 - c. Anemometer, digunakan untuk mengkalibrasi pembacaan sensor.

3.5. Teknik Analisis Data

Teknik analisis data pada prototipe ini yaitu dengan pengujian tiap komponen atau perangkat yang digunakan. Pengujian komponen atau perangkat dalam penelitian ini bertujuan untuk mendapat hasil penelitian yang tepat. Pengujian yang akan dilakukan di beberapa komponen atau perangkat prototipe sistem pengontrol kecepatan udara berbasis mikrokontrol Arduino, antara lain:

3.5.1. Kriteria Pengujian Rangkaian Catu Daya

Kriteria pengujian Catu daya ini yaitu dengan mengukur tegangan keluaran catu daya yang ditunjukkan pada tabel 3.2.

Tabel 3.2 Kriteria Pengujian Catu Daya

Kondisi	7805 (5v)
OFF	
ON	

3.5.2. Kriteria Pengujian Sensor Angin

Pengujian sensor angin ini mengacu pada data yang dikeluarkan dari pin data MD0550. Pin out yang ada pada sensor ini menghasilkan rentang tegangan 0 – VCC. Kepresisian sensor ini akan diuji dengan anemometer sebagai pembandingnya. Tabel 3.3. menunjukkan tabel kriteria pengujian sensor angin dengan membandingkan kecepatan angin yang terdeteksi sensor dengan kecepatan angin yang terbaca oleh Anemometer.

Tabel 3.3. Pengujian *Wind Control*

<i>Fan PWM</i>	<i>V-Raw</i>	<i>V-out</i>	<i>V-Temp</i>	<i>Wind Speed m/s</i>
25				
30				
35				
40				
45				
50				
55				
60				
65				
70				
75				
80				
85				
90				
95				
100				

110				
120				
130				
140				
150				
160				
180				
200				
220				
240				
255				

Tabel 3.4. Pengujian *Error* Kecepatan Sensor Angin

No.	Bukaan Sash	Anemometer (m/s)	Sensor Angin (m/s)	Error (m/s)
1	0%			
2	15%			
3	25%			
4	50%			
5	75%			
6	100%			
Rata-rata				

Tabel 3.5. Kriteria Pengujian Tegangan Sensor Angin (*manual mode*)

No.	bukaan	Kecepatan Angin m/s	Tegangan Sensor (v)
1	0		
2	15%		
3	25%		
4	50%		
5	75%		
6	100%		

Tabel 3.6. Kriteria Pengujian Delay Kecepatan Sensor Saat Dibuka

No.	Bukaan Sash	Delay Waktu (t)
1	0-15%	
2	0-25%	
3	0-50%	
4	0-75%	
5	0-100%	

Tabel 3.7. Kriteria Pengujian Delay Kecepatan Sensor Saat Ditutup

No.	Bukaan Sash	Delay Waktu (t)
1	100-0%	
2	75-0%	
3	50-0%	
4	25-0%	
5	15-0%	

3.5.3. Kriteria Pengujian Motor *Driver*

Pengujian motor *driver* ini berdasar dari seberapa besar bukaan pintu *sash* dari lemari asam, kemudian dibandingkan dengan tegangan keluaran yang dihasilkan oleh motor *driver*.

Tabel 3.8. Kriteria Pengujian Motor *Driver*

No.	Bukaan Pintu Sash	PWM Input	Tegangan Motor (v)
1	0%		
2	15%		
3	25%		
4	50%		
5	75%		
6	100%		

3.5.4. Kriteria Pengujian Tombol *Touchscreen*

Pengujian tombol *touchscreen* ini terdiri dari beberapa tombol yaitu: tombol Lampu, tombol Manual, Tombol matikan Alarm. Ketika tombol tidak ditekan maka keluaran tegangan pada tiap pin keluaran ada yang *high output* dan juga ada yang *low output*. Berikut adalah criteria pengukuran tombol *touchscreen* yang ditunjukkan pada tabel 3.9.

Tabel 3.9. Kriteria Pengujian *Button Touchscreen*

No.	Button	Ketika Ditekan (V)	Ketika Tidak Ditekan (V)
1	Tombol Lampu		
2	Tombol Manual		
3	Tombol Alarm		

3.5.5. Kriteria Pengujian *Buzzer* dan *LED*

Kriteria pengujian *Buzzer* dan *LED (Light Emitting Diode)* yaitu dengan mengukur tegangan pada saat kondisi *high* maupun *low* ditunjukkan pada tabel 3.10.dan 3.11.

Tabel 3.10. Kriteria Pengujian *Buzzer*

<i>Buzzer</i>	Tegangan (V)
Hidup	
Mati	

Tabel 3.11. Kriteria Pengujian *LED*

<i>LED</i>	Tegangan (V)
Hidup	
Mati	

3.5.6. Kriteria Pengujian Kecepatan Udara pada Lemari Asam

Pengujian unjuk kerja lemari asam ini menggunakan asap sebagai media pengujiannya. Asap digunakan untuk melihat berapa waktu yang dibutuhkan untuk menyedot semua asap yang ada pada lemari asam. Hasil pengujian ini diukur dalam satuan m^3/s .

Tabel 3.12. Kriteria Pengujian Kecepatan Udara pada Lemari Asam

No.	Volume	Kecepatan motor	Waktu	Kinerja $m^3/detik$
1				
2				

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1. Hasil Penelitian

4.1.1. Hasil Pengujian Rangkaian Catu Daya

Pengujian Catu Daya ini yaitu dengan mengukur tegangan keluaran pada catu daya 5V. Hasil pengujian dari rangkaian catu daya dapat dilihat pada tabel berikut :

Tabel 4.1. Hasil Pengujian Catu Daya

Kondisi	Output (5V)
OFF	0V
ON	5.13V

4.1.2. Hasil Pengujian Sensor Angin




Pengujian sensor angin ini mengacu pada data yang dikeluarkan dari pin data MD0550. Kepresisian sensor ini akan diuji dengan anemometer sebagai pembandingnya.

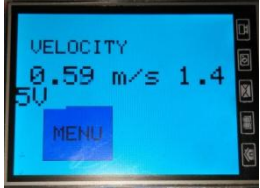

Tabel 4.2. Hasil Pengujian *Wind Censor*

Fan PWM	V-Raw	V-out	V-Temp	Wind Speed m/s
25	2.57	2.87	29.9	1.8
30	2.62	3	30.4	2.3
35	2.63	3.07	30.6	2.6
40	2.66	3.15	31	3.1
45	2.69	3.25	31.3	3,6
50	2.71	3.32	31.5	4,0
55	2.74	3.39	31.8	4,5
60	2.75	3.45	32.1	4,9
65	2.78	3.52	32.2	5,4
70	2.79	3.57	32.5	5.8

75	2.81	3.63	32.7	6.2
80	2.82	3.68	32.9	6.7
85	2.85	3.74	33.1	7.2
90	2.86	3.78	33.2	7.5
95	2.87	3.81	33.4	8.0
100	2.89	3.85	33.5	8.3
110	2.91	3.93	33.8	9.3
120	2.93	3.99	34	10.2
130	2.95	4.08	34.3	11.1
140	2.97	4.14	34.5	11.8
150	2.99	4.19	34.8	12.7
160	3	4.13	34.8	13.6
180	3.03	4.34	35.2	15.1
200	3.07	4.47	35.7	17.1
220	3.1	4.53	36	18.8
240	3.12	4.62	36.2	20.2
255	3.13	4.62	36.4	20.6

Tabel 4.3. Hasil Pengujian Sensor Angin Pada lemari Asam

No.	Level Pintu (%)	Anemo meter (m/s)	Sensor (m/s)	Gambar	Error (m/s)
1	0%	-	-	-	-
2	15%	0.5	0.52		0.02
3	25%	0.5	0.53		0.03
4	50%	0.6	0.67		0.07

5	75%	0.6	0.59		0.01
6	100%	0.7	0.63		0.07

Tabel 4.4. Hasil Pengujian Tegangan Sensor Angin (manual mode)

No.	bukaan	Kecepatan Angin m/s	Tegangan Sensor (v)
1	0	-	-
2	15%	2.6	2
3	25%	1.8	1.6
4	50%	1.3	1.5
5	75%	1.1	1.4
6	100%	0.8	1.3

Tabel 4.5. Hasil Pengujian Delay Kecepatan Sensor Saat Dibuka

No.	Bukaan Sash	Delay Waktu (s)
1	15-25%	1
2	15-50%	3
3	15-75%	5
4	15-100%	6

Tabel 4.6. Hasil Pengujian Delay Kecepatan Sensor Saat Ditutup

No.	Bukaan Sash	Delay Waktu (s)
1	100-15%	15



2	75-15%	11
3	50-15%	10
4	25-15%	7




Tabel 4.2 dan 4.3 diatas menunjukkan tabel hasil pengujian sensor angin dengan membandingkan kecepatan angin yang terdeteksi sensor dengan kecepatan angin yang terbaca oleh Anemometer. Sedangkan tabel 4.4 menunjukkan hasil pengukuran tegangan pada *output* sensor. Gambar 4.5 dan 4.6 menunjukkan delay waktu yang dibutuhkan untuk mencapai keadaan normal (0.4 - 0.6 m/s) dari keadaan bukaan pintu maksimal 100% ke bukaan minimal 15%. Gambar 4.5 menunjukkan delay waktu yang dibutuhkan untuk mencapai keadaan normal (0.4 - 0.6 m/s) dari keadaan bukaan pintu minimal yaitu sebesar 15% ke bukaan maksimal.

4.1.3. Hasil Pengujian Motor Driver

Pengujian motor *driver* ini berdasar dari seberapa besar bukaan pintu sash dari lemari asam, kemudian diukur tegangan output yang dihasilkan oleh motor *driver*.

Tabel 4.7. Hasil Pengujian Motor Driver

No.	Bukaan Pintu Sash	Tegangan Output Driver	Input PWM
1	0%	-	-
2	15%		40
3	25%		71

4	50%		101
5	75%		126
6	100%		171

4.1.4. Hasil Pengujian Tombol *Touchscreen*

Pengujian tombol *touchscreen* ini terdiri dari beberapa tombol yaitu: tombol Lampu, tombol Manual, Tombol matikan Alarm. Ketika tombol tidak ditekan maka keluaran tegangan pada tiap pin keluaran Arduino ada yang *high output* dan juga ada yang *low output*. Tabel 4.8. menunjukkan hasil pengujian tombol *touchscreen*.

Tabel 4.8. Hasil Pengujian Tombol *Touchscreen*

No.	Button	Ketika Ditekan (V)	Ketika Tidak Ditekan (V)
1	Tombol Lampu	0.6	5.06
2	Tombol Manual	0.6	5.05
3	Tombol Alarm	0.6	5.06

4.1.5. Hasil Pengujian *Buzzer* dan *LED*

Hasil pengujian *Buzzer* dan *LED (Light Emitting Diode)* yaitu dengan mengukur tegangan pada saat kondisi *high* maupun *low* ditunjukkan pada tabel 4.9 dan 4.10

Tabel 4.9. Hasil Pengujian Buzzer

<i>Buzzer</i>	Tegangan (V)
Hidup	5.08
Mati	0.9

Tabel 4.10. Hasil Pengujian LED

<i>LED</i>	Tegangan (V)
Hidup	5.08
Mati	0.9

4.1.6. Hasil Pengujian Kecepatan Udara pada Lemari Asam

Pengujian unjuk kerja lemari asam ini menggunakan asap sebagai media pengujiannya. Asap digunakan untuk melihat berapa waktu yang dibutuhkan untuk menyedot semua asap yang ada pada lemari asam. Hasil pengujian ini diukur dalam satuan m³/s.

Tabel 4.11. Hasil Pengujian Kecepatan Udara pada Lemari Asam

No.	Volume	Kecepatan motor	Waktu	Kinerja m³/detik
1	430m³	maks	6 Detik	71.6
2	430m³	min	15 Detik	28.6

4.2. Pembahasan

Setelah hasil dari keseluruhan pengujian dilakukan, maka prototipe sistem pengontrol kecepatan udara ini dapat dikatakan sesuai dengan perencanaan, tetapi ada beberapa keterbatasan pada prototipe sistem pengontrol kecepatan udara ini. Berikut ini adalah pembahasan-pembahasan pada hasil yang didapat.

Pembahasan pertama yaitu cara kerja *wind censor*. *Wind censor* ini bekerja dengan cara membaca perubahan tegangan yang terjadi pada *Resistor Thermal* (lihat gambar rangkaian pada lampiran 2). *Resistor Thermal* dialiri tegangan konstan (VCC) sehingga menghasilkan panas pada resistor tersebut. Ketika ada pergerakan udara yang melewati *Resistor Thermal* tersebut, maka suhu pada *Resistor Thermal* juga mengalami perubahan sebagaimana tegangan yang melewati *Resistor Thermal* tersebut. Perubahan tegangan itulah yang kemudian dibaca oleh mikrokontrol dan kemudian dirubah kedalamsatuan meter per sekon.

Pembahasan selanjutnya yaitu pada pengujian perangkat *input* yang meliputi sensor angin dan tombol *touchscreen*. Pengujian pada sensor angin yaitu dengan membandingkan kecepatan angin yang didapat dari pembacaan program dengan kecepatan yang didapat dari pengukuran manual melalui pengukuran anemoneter. Pengujian ini menghasilkan nilai *output* yang terbaca pada program tidak jauh berbeda dengan pengujian secara manual. Pengujian pada tombol *touchscreen* yaitu dengan mengukur tegangan keluaran dari setiap *button*, pengukuran tegangan ini menghasilkan tegangan 0.6 Volt atau logika *low* ketika tidak ditekan dan 5.6 Volt atau logika *high* ketika ditekan.

Pengujian pada perangkat *output* meliputi motor *driver*, *LED* dan *Buzzer*. Pengujian pertama yaitu pengujian motor *driver*, motor DC ini sudah dibuat sedemikian rupa untuk dapat berputar ke satu arah dengan kecepatan putarannya yang berubah-ubah sesuai bukaan *sash door*. Pengujian ini didapatkan hasil yaitu motor dapat berputar searah jarum jam. Pengujian berikutnya yaitu pengujian pada *LED* yaitu dengan mengukur tegangan *LED* pada saat kondisi *high* maupun pada saat *low*. Hasil yang didapatkan pada pengujian *LED* menunjukkan *LED* dapat

bekerja dengan baik. Pengujian *Buzzer* yaitu dengan mengukur tegangan pada saat hidup dan pada saat mati, dimana pada saat hidup tegangan terukur 5.08 Volt dan pada saat mati 0.9 Volt, pengukuran ini sesuai dengan kriteria pengujian.

Pada hasil pengujian juga didapatkan adanya delay waktu yang cukup lama setiap kali keadaan pintu dirubah. Itu disebabkan karena pergerakan udara yang tidak spontan. Delay juga merupakan kekurangan dari cara kerja sistem kontrol lup tertutup.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan perancangan, hasil dan analisa yang telah dilakukan pada penelitian ini, maka dapat disimpulkan bahwa seluruh bagian dari prototipe sistem pengontrol kecepatan udara ini dapat berfungsi dengan baik. Sensor angin dapat bekerja dengan cukup baik. Tombol touchscreen, *LCD*, motor *driver* juga dapat bekerja sebagaimana mestinya. Alat ini bekerja dengan memanfaatkan sensor yang memindai kecepatan putaran motor pada prototype, kemudian hasil dari sensor diolah pada mikrokontroller untuk selanjutnya output dari mikrokontroller digunakan untuk mengendalikan motor yang berperan menjaga kestabilan kecepatan udara yang melintas tetap berada diantara 0,4 m/s hingga 0,6 m/s. Prototipe ini memiliki kemampuan menghisap udara sebesar 71,6 m³/detik .

5.2. Saran

Berdasarkan penelitian yang telah dilaksanakan dan kesimpulan yang didapatkan, maka saran untuk penyempurnaan dan pengembangan lebih lanjut dari penelitian ini antara lain:

- A. Menggunakan jenis mikrokontroler dengan spesifikasi yang lebih tinggi lagi guna meningkatkan kecepatan dan dapat menyimpan lebih banyak program.
- B. Menambah jenis sensor lainnya yang belum ada didalam prototipe lemari asam untuk peringatan dari kebocoran dan lebih meningkatkan keamanan pengguna.
- C. Membuat sistem bukaan pintu atau *sash door* dengan secara otomatis

D. Membuat sistem keamanan dengan *password* / *ID Card*, sehingga penggunaan lemari asam ini lebih aman dan tidak disalahgunakan.

DAFTAR PUSTAKA

- Anonim. (1984, 11 21). *Omega.com*. Dipetik 07 28, 2016, dari Omega Engineering.
- Anonim. (t.thn.). *praktikum mekatronika*. Dipetik Juli 26, 2016, dari <http://mechatronicscrew.wordpress.com>.
- ASHRAE, NFPA, NRC. (2016, maret 19). *dale sartor*. Dipetik maret 19, 2016, dari <http://ateam.lbl.gov/hightech/fumehood/students/su00/Fox/FHSafety.htm>
- Badger, P. (2011). *wind sensor rev C*. Dipetik april 20, 2016, dari wind sensor: <https://moderndevice.com/contact-modern-device/>
- Banzi, M. (2008). *Getting Started with Arduino*. USA: O'reilly.
- Cassavoy, L. (2016, maret 19). *about tech*. Dipetik mei 1, 2016, dari about.com: <http://cellphones.about.com/od/phoneglossary/g/What-Is-Tft-Lcd.htm>
- Dr.Ir.Andi Adriansyah, M. (2012). Dasar Sistem Kontrol. *Pengantar Sistem Kontrol* , 19.
- Efendi, I. (2014, Juni 18). *IT Jurnal*. Dipetik Juni 18, 2016, dari <http://www.it-jurnal.com/2014/05/pengertian-dan-kelebihan-arduino.html>
- Elektronika-dasar. (2015, Januari 15). Dipetik mei 1, 2016, dari <http://elektronika-dasar.web.id/persyaratan-sensor-dan-tranducer/>
- jasakalibrasi. (2013, september 18). *lemari asam*. Dipetik april 19, 2016, dari <http://jasakalibrasi.net/lemari-asam/>
- NFPA. (2002). *Paten No. 45*. America.
- Ogata, K. (1970). *Teknik Kontrol Automatik*. Engelwood Cliffs, N.J: Prentice-Hall,inc.
- Sommerviile, Ian. 2003. *Software Engineering*. Jakarta: Erlangga.
- Stone, D. (2016, maret 1). *tft touchscreen*. Dipetik mei 1, 2016, dari http://www.ehow.com/facts_7429491_tft-touch-screen_.html
- Tim Penyusun Fakultas Teknik, Universitas Negeri Jakarta. 2015. *Buku Pedoman Skripsi/Karya Inovatif/Komprehensif*. Jakarta: Fakultas Teknik, Universitas Negeri Jakarta.
- Syahwil, M. (2013). *Panduan Mudah Simulasi & Praktek Mikrokontroler Arduino*. yogyakarta: C.V Andi Offset.

LAMPIRAN

Lampiran 1. Spesifikasi dan Gambar Prototipe Lemari Asam

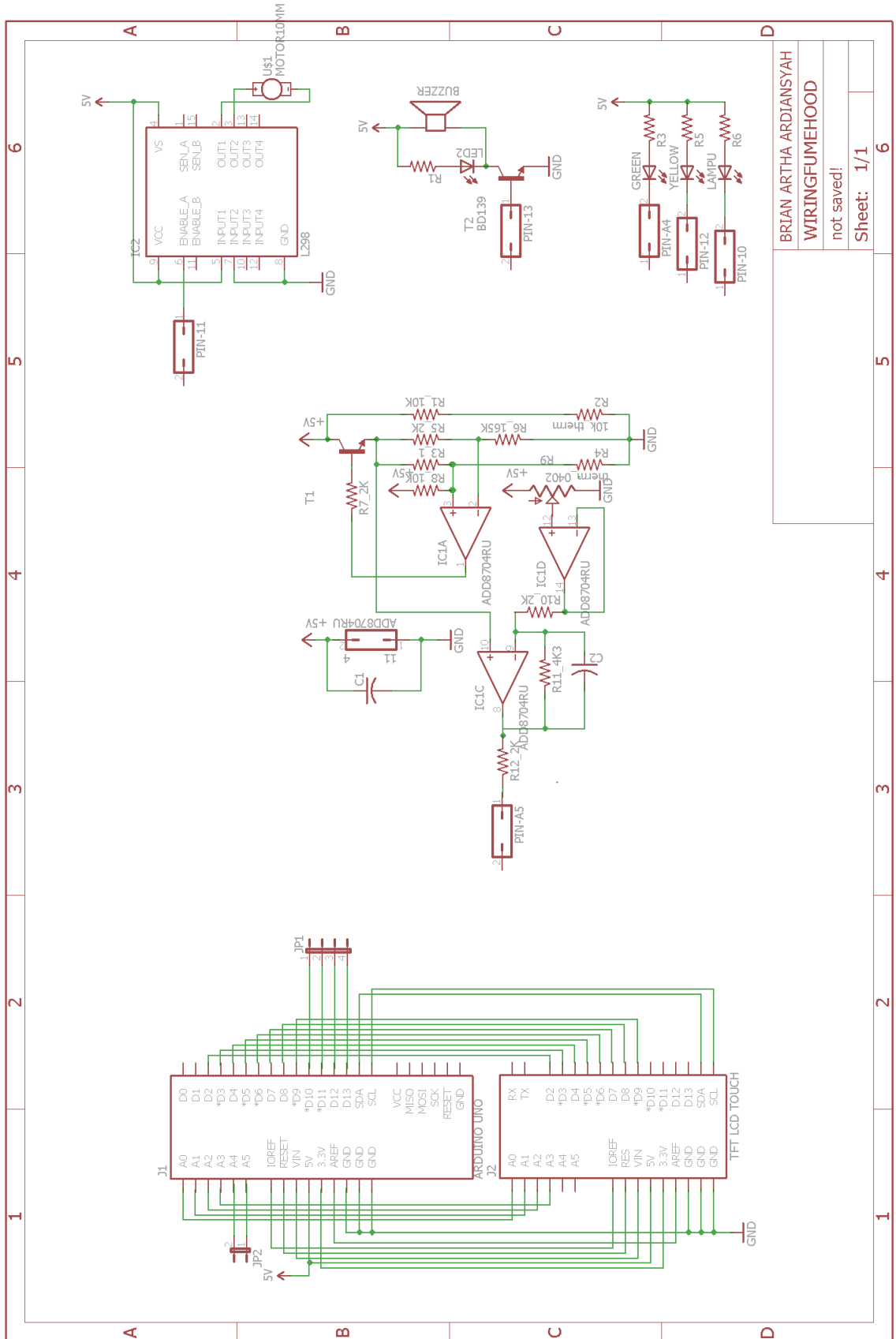
1. Spesifikasi

Dimensi	:	30 cm x 30 cm x 50 cm
Tipe Mikrokontroler	:	Arduino Uno
Tegangan AC	:	220 Volt
<i>Input</i>	:	Sensor Kecepatan Udara, <i>Touchscreen Button</i> .
<i>Output</i>	:	TFT LCD, Motor DC, Lampu <i>LED</i> , Indikator <i>LED</i> , dan <i>Buzzer</i> .

2. Gambar



Lampiran 2. Skema Rangkaian sistem pengontrol kecepatan udara



BRIAN ARTHA ARDIANSYAH	6
WIRINGFUMEHOOD	6
not saved!	6
Sheet: 1/1	6

Lampiran 3. Gambar Hasil Pengujian



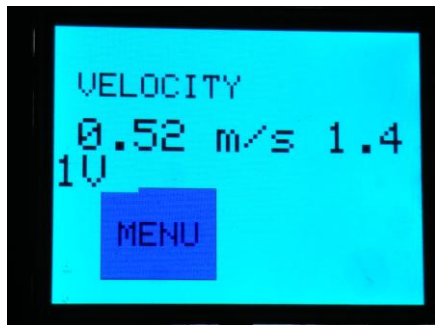
Hasil Pengukuran
Tegangan Catu Daya



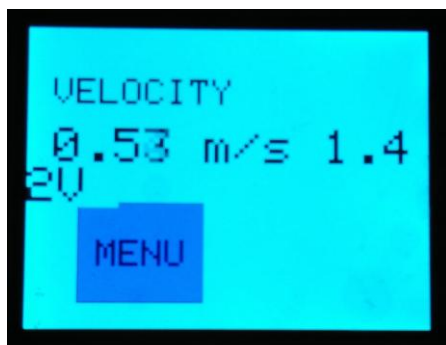
Hasil Pengujian *LED* (High)
dan *Buzzer*



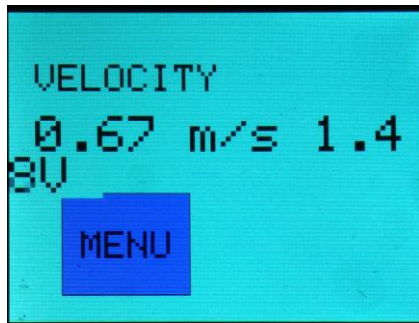
Hasil Pengujian *LED*
Kondisi *Low Output*



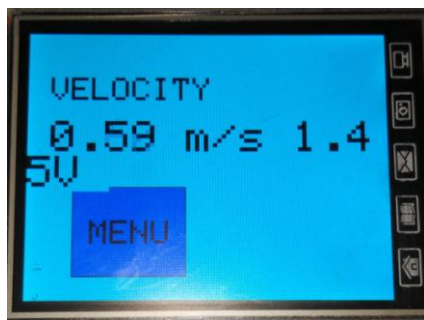
Hasil Pengukuran Kecepatan Udara Dengan Bukaannya Pintu 15%



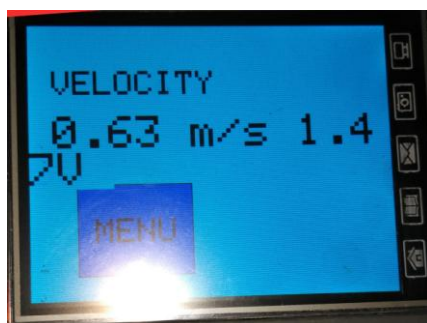
Hasil Pengukuran Kecepatan Udara Dengan Bukaannya Pintu 25%



Hasil Pengukuran Kecepatan Udara Dengan Bukaannya Pintu 50%



Hasil Pengukuran Kecepatan Udara Dengan Bukaannya Pintu 75%



Hasil Pengukuran Kecepatan Udara Dengan Bukaannya Pintu 100%



Hasil Pengukuran Teg Motor DC Pada Bukaannya 100%



Hasil Pengukuran Teg Motor DC Pada Bukaannya 75%



**Hasil Pengukuran Teg Motor DC
Pada Bukaannya 50%**



**Hasil Pengukuran Teg Motor DC
Pada Bukaannya 25%**



**Hasil Pengukuran Teg Motor DC
Pada Bukaannya 15%**

Lampiran 4. Program Arduino Uno

```

#include <Adafruit_GFX.h>    // Core graphics library
#include <MCUFRIEND_kbv.h>
MCUFRIEND_kbv tft;        // hard-wired for UNO shields anyway.
#include <TouchScreen.h>
#include <stdio.h>
#include <stdlib.h>
#include <EEPROM.h>
#include <math.h>

#if defined(__SAM3X8E__)
#undef __FlashStringHelper::F(string_literal)
#define F(string_literal) string_literal
#endif

// most mcufriend shields use these pins and Portrait mode:
uint8_t YP = A1;  // must be an analog pin, use "An" notation!
uint8_t XM = A2;  // must be an analog pin, use "An" notation!
uint8_t YM = 7;   // can be a digital pin
uint8_t XP = 6;   // can be a digital pin
uint8_t SwapXY = 0;

uint16_t TS_LEFT = 132;//916;
uint16_t TS_RT  = 917;//127;
uint16_t TS_TOP = 146;//944;
uint16_t TS_BOT = 944;//149;
char *name = "Unknown controller";

// For better pressure precision, we need to know the resistance
// between X+ and X- Use any multimeter to read it
// For the one we're using, its 300 ohms across the X plate
TouchScreen ts = TouchScreen(XP, YP, XM, YM, 300);
TSPoint tp;

#define MINPRESSURE 20
#define MAXPRESSURE 1000

#define SWAP(a, b) {uint16_t tmp = a; a = b; b = tmp;}

int16_t BOXSIZE;
int16_t PENRADIUS = 0;
uint16_t identifier, oldcolor, currentcolor;
uint8_t Orientation = 1;    //PORTRAIT

// Assign human-readable names to some common 16-bit color values:
#define BLACK    0x0000
#define BLUE     0x001F
#define RED      0xF800
#define GREEN    0x07E0
#define CYAN     0x07FF
#define MAGENTA  0xF81F
#define YELLOW   0xFFE0
#define WHITE    0xFFFF
#define analogPinForRV    A5    // sensor rv

```

```

#define K      11// kipas
#define L      10//lampu dalem
#define M      12//manual mode
#define H      A4// buat lampu ijo

#define B      13//buzzer

char warna;
float
velocity,tes,timer,keluar=0,lampu,mode,kunci,batas_bawah,batas_atas;
float alarm=0,indi_alarm=0;
unsigned warning;
long int kode=123456,pass=0;
long int nilai;
unsigned int i,a,array[50],data[15],cek,dt[25],Volt,c=0,d=0,e=0;
unsigned int nilaiPWM,kali;

int TMP_Therm_ADunits; //temp termistor value from wind sensor
float RV_Wind_ADunits,MPH,m,b; //RV output from wind sensor
float RV_Wind_Volts;
unsigned long lastMillis,lasstMillis,lassstMillis,lasssstMillis;

void menu();
void tap();
void cek_tegangan();
void key_pad(unsigned int pas);
void program_inti();

void tap() // buat ambil koordinat touch screen
{
    tp = ts.getPoint();
    pinMode(XM, OUTPUT);
    pinMode(YP, OUTPUT);
    pinMode(XP, OUTPUT);
    pinMode(YM, OUTPUT);
}

void cek_tegangan() //tampilan data sensor
{
    e=0;
    if(MPH < (batas_bawah-0.01)&& c==0 && warning>=500)
    {
        tft.fillScreen(RED);
        tft.fillRect(40, 140, 100,80, BLUE);
        tft.drawRect(40, 140, 100,80, BLACK);
        tft.setTextSize(3);
        tft.setCursor(55,170);
        tft.setTextColor(BLACK,BLUE);
        tft.print("MENU");
        tft.setTextColor(BLACK,RED);
        c=1;
        d=0;
    }
    else
    {
        if(mode==0 && b==0)
        {

```

```

tft.fillScreen(YELLOW);
tft.fillRect(40, 140, 100,80, BLUE);
tft.drawRect(40, 140, 100,80, BLACK);
tft.setTextSize(3);
tft.setCursor(55,170);
tft.setTextColor(BLACK,BLUE);
tft.print("MENU");
tft.setTextColor(BLACK,YELLOW);
}

if(mode==1 && b==0)
{
tft.fillScreen(CYAN);
tft.fillRect(40, 140, 100,80, BLUE);
tft.drawRect(40, 140, 100,80, BLACK);
tft.setTextSize(3);
tft.setCursor(55,170);
tft.setTextColor(BLACK,BLUE);
tft.print("MENU");
tft.setTextColor(BLACK,CYAN);
}
b=1;
}

if(MPH >=batas_bawah && d==0)
{
if(mode==0 && c==0)
{
tft.fillScreen(YELLOW);
tft.fillRect(40, 140, 100,80, BLUE);
tft.drawRect(40, 140, 100,80, BLACK);
tft.setTextSize(3);
tft.setCursor(55,170);
tft.setTextColor(BLACK,BLUE);
tft.print("MENU");
tft.setTextColor(BLACK,YELLOW);
d=1;
}
if(mode==1 && c==0)
{
tft.fillScreen(CYAN);
tft.fillRect(40, 140, 100,80, BLUE);
tft.drawRect(40, 140, 100,80, BLACK);
tft.setTextSize(3);
tft.setCursor(55,170);
tft.setTextColor(BLACK,BLUE);
tft.print("MENU");
tft.setTextColor(BLACK,CYAN);
d=1;
}
}
d=1;
}

while(1)
{

```

```

program_inti();

if(MPH < (batas_bawah-0.01)&& c==0 && warning>=500)
{
  tft.fillScreen(RED);
  tft.fillRect(40, 140, 100,80, BLUE);
  tft.drawRect(40, 140, 100,80, BLACK);
  tft.setTextSize(3);
  tft.setCursor(55,170);
  tft.setTextColor(BLACK,BLUE);
  tft.print("MENU");
  tft.setTextColor(BLACK,RED);
  c=1;
  d=0;
}
/*
else
{
  if(mode==0 && b==0)
  {
    tft.fillScreen(YELLOW);
    tft.fillRect(40, 140, 100,80, BLUE);
    tft.drawRect(40, 140, 100,80, BLACK);
    tft.setTextSize(3);
    tft.setCursor(55,170);
    tft.setTextColor(BLACK,BLUE);
    tft.print("MENU");
    tft.setTextColor(BLACK,YELLOW);
    b=1;
  }

  if(mode==1 && b==0)
  {
    tft.fillScreen(CYAN);
    tft.fillRect(40, 140, 100,80, BLUE);
    tft.drawRect(40, 140, 100,80, BLACK);
    tft.setTextSize(3);
    tft.setCursor(55,170);
    tft.setTextColor(BLACK,BLUE);
    tft.print("MENU");
    tft.setTextColor(BLACK,CYAN);
    b=1;
  }
  b=1;
}
*/

if(MPH >=batas_bawah && d==0)
{
  if(mode==0 && c==0 && d==0)
  {
    tft.fillScreen(YELLOW);
    tft.fillRect(40, 140, 100,80, BLUE);
    tft.drawRect(40, 140, 100,80, BLACK);
    tft.setTextSize(3);
    tft.setCursor(55,170);
    tft.setTextColor(BLACK,BLUE);
  }
}

```



```

    tft.print("MENU");
    tft.setTextColor(BLACK,YELLOW);
    d=1;
  }
  if(mode==1 && c==0 && d==0)
  {
    tft.fillScreen(CYAN);
    tft.fillRect(40, 140, 100,80, BLUE);
    tft.drawRect(40, 140, 100,80, BLACK);
    tft.setTextSize(3);
    tft.setCursor(55,170);
    tft.setTextColor(BLACK,BLUE);
    tft.print("MENU");
    tft.setTextColor(BLACK,CYAN);
    d=1;
  }

  d=1;
}

tft.setCursor(20,80);
tft.setTextSize(4);
tft.print(String(MPH)+" m/s "+String(nilaiPWM)+" ");

tft.setCursor(20,80);
tft.setTextSize(3);
tft.setCursor(20,40);
tft.print("VELOCITY");

tp = ts.getPoint();
pinMode(XM, OUTPUT);
pinMode(YP, OUTPUT);
pinMode(XP, OUTPUT);
pinMode(YM, OUTPUT);
tft.setCursor(0, 0);

if (tp.x > 620 && tp.x < 730 && tp.y > 660 && tp.y < 730)
{
  tap();
  menu();
  e=0;
  if(MPH < (batas_bawah-0.01)&& c==0 && warning>=500)
  {
    tft.fillScreen(RED);
    tft.fillRect(40, 140, 100,80, BLUE);
    tft.drawRect(40, 140, 100,80, BLACK);
    tft.setTextSize(3);
    tft.setCursor(55,170);
    tft.setTextColor(BLACK,BLUE);
    tft.print("MENU");
    tft.setTextColor(BLACK,RED);
    c=1;
    d=0;
  }
  else
{
  if(mode==0 && b==0)

```

```

{
  tft.fillScreen(YELLOW);
  tft.fillRect(40, 140, 100,80, BLUE);
  tft.drawRect(40, 140, 100,80, BLACK);
  tft.setTextSize(3);
  tft.setCursor(55,170);
  tft.setTextColor(BLACK,BLUE);
  tft.print("MENU");
  tft.setTextColor(BLACK,YELLOW);
}

if(mode==1 && b==0)
{
  tft.fillScreen(CYAN);
  tft.fillRect(40, 140, 100,80, BLUE);
  tft.drawRect(40, 140, 100,80, BLACK);
  tft.setTextSize(3);
  tft.setCursor(55,170);
  tft.setTextColor(BLACK,BLUE);
  tft.print("MENU");
  tft.setTextColor(BLACK,CYAN);
}
b=1;
}

if(MPH >=batas_bawah && d==0)
{
  if(mode==0 && c==0 && d==0)
  {
    tft.fillScreen(YELLOW);
    tft.fillRect(40, 140, 100,80, BLUE);
    tft.drawRect(40, 140, 100,80, BLACK);
    tft.setTextSize(3);
    tft.setCursor(55,170);
    tft.setTextColor(BLACK,BLUE);
    tft.print("MENU");
    tft.setTextColor(BLACK,YELLOW);
    d=1;
  }
  if(mode==1 && c==0 && d==0)
  {
    tft.fillScreen(CYAN);
    tft.fillRect(40, 140, 100,80, BLUE);
    tft.drawRect(40, 140, 100,80, BLACK);
    tft.setTextSize(3);
    tft.setCursor(55,170);
    tft.setTextColor(BLACK,BLUE);
    tft.print("MENU");
    tft.setTextColor(BLACK,CYAN);
    d=1;
  }
}
d=1;
}
}
}

```

```

}

void menu()    //menu
{
  tft.fillScreen(WHITE);
  tap();
  tft.setTextSize(3); tft.setTextColor(BLACK); tft.setCursor(130,5);
tft.print("MENU");

  tft.setTextSize(3); tft.setTextColor(RED); tft.setCursor(40,50);
tft.print("TURN LIGHT");
  tft.setTextSize(3); tft.setTextColor(GREEN); tft.setCursor(40,110);
tft.print("TURN OFF ALARM");
  tft.setTextSize(3); tft.setTextColor(RED); tft.setCursor(40,170);
tft.print("MANUAL MODE");

  tft.fillRect(250,200,70,40,BLACK);
  tft.setTextSize(3); tft.setTextColor(WHITE); tft.setCursor(275,208);
tft.print("<");
  if(lampu==1){tft.fillRect(5,50,20,20,GREEN);digitalWrite(L,
LOW);delay(300);} //lampu on
  if(lampu==0){tft.fillRect(5,50,20,20,RED);digitalWrite(L,
HIGH);delay(300);} //lampu off

  if(mode==0){tft.fillCircle(15,180,10,GREEN);digitalWrite(M,
LOW);delay(300);} //mode on
  if(mode==1){tft.fillCircle(15,180,10,RED);digitalWrite(M,
HIGH);delay(300);} //mode off
  e=1;
  program_inti();
  tap();
  while(1)
  {
    e=1;
    program_inti();

tap();tft.setTextColor(BLACK,BLUE);tft.setTextSize(2);tft.setCursor(0,
0);
    tft.print("x=" + String(tp.x) + " y=" + String(tp.y) + " ");
    if(indi_alarm==0)
    {
      tft.setTextSize(3); tft.setTextColor(RED); tft.setCursor(40,50);
tft.print("TURN LIGHT");
      tft.setTextSize(3); tft.setTextColor(RED); tft.setCursor(40,110);
tft.print("TURN OFF ALARM");
      tft.setTextSize(3); tft.setTextColor(RED); tft.setCursor(40,170);
tft.print("MANUAL MODE");
    }

    if(indi_alarm==1)
    {
      tft.setTextSize(3); tft.setTextColor(RED); tft.setCursor(40,50);
tft.print("TURN LIGHT");
      tft.setTextSize(3); tft.setTextColor(GREEN);
tft.setCursor(40,110); tft.print("TURN OFF ALARM");
      tft.setTextSize(3); tft.setTextColor(RED); tft.setCursor(40,170);
tft.print("MANUAL MODE");
    }
  }
}

```

```

    }

    else
        if(tp.x > 180 && tp.x < 270 && tp.y > 820 && tp.y <
850){tap();delay(150);tap();e=c=d=0;cek_tegangan();break;}//back

        if(tp.x > 550 && tp.x < 650 && tp.y > 700 && tp.y < 730 &&
mode==1)//manual mode on/off
            {mode=0;tft.fillCircle(15,180,10,GREEN); digitalWrite(M,
LOW);tap();delay(300); tap();} //mode on
            if(tp.x > 550 && tp.x < 650 && tp.y > 680 && tp.y < 730 &&
mode==0)//manual mode on/off
            {mode=1;tft.fillCircle(15,180,10,RED); digitalWrite(M,
HIGH);tap();delay(300); tap();} //mode off

            if(tp.x > 250 && tp.x < 870 && tp.y > 520 && tp.y < 570 && MPH <=
(batas_bawah-0.01) && indi_alarm==0) // turn off alarm
            { indi_alarm=1; tap();delay(300); tap();} //lampu on
            if(tp.x > 250 && tp.x < 870 && tp.y > 520 && tp.y < 570 && MPH <=
(batas_bawah-0.01) && indi_alarm==1) // turn on alarm
            {indi_alarm=0; tap();delay(300); tap();} //lampu off

            if(tp.x > 400 && tp.x < 760 && tp.y > 290 && tp.y < 360 &&
lampu==0)// turn LAMPU on off
            { lampu=1;tft.fillRect(5,50,20,20,GREEN); digitalWrite(L,
LOW);tap();delay(300); tap();} //lampu on
            if(tp.x > 400 && tp.x < 760 && tp.y > 290 && tp.y < 360 &&
lampu==1)// turn LAMPU on off
            {lampu=0;tft.fillRect(5,50,20,20,RED); digitalWrite(L,
HIGH);tap();delay(300); tap();} //lampu off
            }
    }

void program_inti()
{
    if (millis() - lastMillis > 300)
    {
        Volt=0;
        for(a=0;a<10;a++)
        {
            dt[a]= analogRead(analogPinForRV);
            Volt=Volt+dt[a];
        }

        RV_Wind_Volts = ((Volt/10) * 0.0048828125);
        //MPH = ((8.297*RV_Wind_Volts));
        //MPH = (MPH - 14.52)-0.1;
        m=(2.9-0)/(2.03-0.58);
        b=(m*0.58);
        MPH = ((m*RV_Wind_Volts));
        MPH = (MPH - b - 1.1);
        if(MPH<=0){MPH=0;}

        lastMillis = millis();
    }

    if(mode==1)

```

```

    {
        if (millis() - lasstMillis > 400) //pwm naik
        {
            if ( MPH >= batas_bawah && MPH <=
batas_atas){} //nilaiPWM=40;}

            if (MPH < (batas_bawah + 0.01) && (nilaiPWM <
240)){nilaiPWM=nilaiPWM+5;} //jika velocity kurang atau sama dengan 0.5

            lasstMillis = millis();
        }

        if (millis() - lasstMillis > 250) //pwm turun
        {
            if (MPH > (batas_atas) && nilaiPWM >
40){nilaiPWM=nilaiPWM-1;} //jika velocity lebih atau sama dengan 0.7
            lasstMillis = millis();
        }
    }
    if(mode==0){nilaiPWM=200;}

    analogWrite(K,nilaiPWM); //jalankan fan1 dengan pwm yang diinginkan

    if(MPH<=(batas_bawah-0.01))
    {
        warning++;
        if(warning>=500)
        {
            alarm++;
            digitalWrite(H, HIGH);
            if(alarm<=3){if(indi_alarm==0){digitalWrite(B,
HIGH);}else{digitalWrite(B, LOW);}}
            if(e==1){tft.fillTriangle(15,110,3,130,27,130,RED);}}
            if(5 > alarm && alarm > 3){if(indi_alarm==0){digitalWrite(B,
LOW);}if(e==1){tft.fillTriangle(15,110,3,130,27,130,GREEN);}}
            if(8 > alarm && alarm > 5){if(indi_alarm==0){digitalWrite(B,
HIGH);}if(e==1){tft.fillTriangle(15,110,3,130,27,130,RED);}}
            if(alarm > 8){if(indi_alarm==0){digitalWrite(B,
LOW);}else{digitalWrite(B,
LOW);}if(e==1){tft.fillTriangle(15,110,3,130,27,130,GREEN);}}
            if(alarm>=15){alarm=0;}
        }

        if(warning>=600){warning=600;}
    }

    if(MPH>(batas_bawah-
0.01)){warning=c=b=0;indi_alarm=0;alarm=0;digitalWrite(H,
LOW);digitalWrite(B,
LOW);if(e==1){tft.fillTriangle(15,110,3,130,27,130,GREEN);}}
}

void setup(void)
{
    uint16_t tmp;

```

```

tft.begin(9600);
tft.reset();
identifier = tft.readID();
pinMode(K, OUTPUT); pinMode(L, OUTPUT);
pinMode(M, OUTPUT); pinMode(H, OUTPUT);
pinMode(B, OUTPUT);

switch (Orientation)
{
    // adjust for different aspects
    case 0: break; //no change, calibrated for PORTRAIT
    case 1: tmp = TS_LEFT, TS_LEFT = TS_BOT, TS_BOT = TS_RT, TS_RT
= TS_TOP, TS_TOP = tmp; break;
    case 2: SWAP(TS_LEFT, TS_RT); SWAP(TS_TOP, TS_BOT); break;
    case 3: tmp = TS_LEFT, TS_LEFT = TS_TOP, TS_TOP = TS_RT, TS_RT
= TS_BOT, TS_BOT = tmp; break;
}

ts = TouchScreen(XP, YP, XM, YM, 300); //call the constructor
AGAIN with new values.
tft.begin(identifier);
tft.setRotation(Orientation);
tes=0;
tft.fillScreen(BLACK);
keluar=lampu=0;
mode=1;
if(lampu==1){digitalWrite(L, LOW);} //lampu on
if(lampu==0){digitalWrite(L, HIGH);} //lampu off

if(mode==0){digitalWrite(M, LOW);} //mode on
if(mode==1){digitalWrite(M, HIGH);} //mode off
digitalWrite(B, LOW);digitalWrite(H, LOW);

batas_bawah=0.4;
batas_atas=0.7;
alarm=indi_alarm=0;
cek=100;
nilaiPWM=100;
mode=1;
b=c=d=e=0;
warning=alarm=0;
}

void loop()
{
    cek_tegangan();
}

```

RIWAYAT HIDUP



Penulis bernama lengkap BRIAN ARTHA ARDIANSYAH,S.Pd, dilahirkan di desa Candimulyo kecamatan Candimulyo kabupaten Magelang Jawa Tengah pada tanggal 03 Agustus 1992. Penulis anak kandung dari pasangan suami istri dengan ayah yang bernama Sularto dan ibu bernama Sujiati. Penulis merupakan anak pertama dari enam bersaudara.

Penulis memulai pendidikan formal di sekolah dasar SD Negeri 02 Menteng Dalam Jakarta pada tahun 2003. Kemudian melanjutkan pendidikan di kota kelahirannya, tepatnya di SLTP Negeri 2 Candimulyo Magelang dan lulus pada tahun 2006. Penulis kemudian melanjutkan pendidikannya di SMK Adipura Magelang dengan kejuruan Teknik Audio Video dan lulus pada tahun 2009. Setelah lulus SMK, penulis melanjutkan pendidikannya ke tingkat perguruan tinggi dan di terima di Fakultas Teknik, Program Studi S1 Pendidikan Teknik Elektronika, Universitas Negeri Jakarta yang berlokasi di Rawamangun, Jakarta Timur. Penulis memperoleh gelar Sarjana Pendidikan dengan judul Skripsi “PERANCANGAN PROTOTIPE SISTEM PENGONTROL KECEPATAN UDARA BERBASIS ARDUINO UNO UNTUK APLIKASI PADA LEMARI ASAM” dan lulus pada tahun 2016.