TUGAS AKHIR – TE 145561

GENERIC PROGRAMMABLE PENGHUBUNG PENGGUNA DENGAN *I/O* DARI ARDUINO YANG DITERAPKAN DENGAN *FINGERPRINT* DAN DHT11

Rendra Kurnia .R
NRP 2214030051

Dosen Pembimbing
Ir. Djoko Suprajitno Rahardjo, M.T
Dr. Ir. Achmad Affandi, DEA

PROGRAM STUDI KOMPUTER KONTROL
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017

FINAL PROJECT – TE 145561

GENERIC PROGRAMMABLE PENGHUBUNG PENGGUNA DENGAN *I/O* DARI ARDUINO YANG DITERAPKAN DENGAN *FINGERPRINT* DAN DHT11

Rendra Kurnia .R
NRP 2214030051

Advisor
Ir. Djoko Suprajitno Rahardjo, M.T
Dr. Ir. Achmad Affandi, DEA

COMPUTER CONTROL STUDY PROGRAM
Electrical and Automation Engineering Department
Vocational Faculty
Institut Teknologi Sepuluh Nopember
Surabaya 2017

PERNYATAAN KEASLIAN
TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "***Generic Programmable* penghubung pengguna dengan *I/O* dari arduino yang diterapkan dengan *Fingerprint* dan DHT11**" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya,9 Juni 2017

Rendra Kurnia .R
NRP 2214030051

Setyo Budi Utomo
NRP 2214030052

-----Halaman ini sengaja dikosongkan-----

***GENERIC PROGRAMMABLE* PENGHUBUNG PENGGUNA DENGAN *I/O* DARI ARDUINO YANG DITERAPKAN DENGAN *FINGERPRINT* DAN DHT11**

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Ahli Madya
Pada
Program Studi Komputer Kontrol
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember

Menyetujui :

Dosen Pembimbing I	Dosen Pembimbing II
Ir. Djoko Suprajitno Rahardjo, M.T.	Dr. Ir. Achmad Affandi, DEA.
NIP. 19550622 198701 1 001	NIP. 19651014 199002 1 001

SURABAYA
JUNI, 2017

-----Halaman ini sengaja dikosongkan-----

GENERIC PROGRAMMABLE PENGHUBUNG PENGGUNA DENGAN _I/O_ DARI ARDUINO YANG DITERAPKAN DENGAN _FINGERPRINT_ DAN DHT11

Nama : Rendra Kurnia .R
Nama : Setyo Budi Utomo
Pembimbing 1 : Ir. Djoko Suprajitno Rahardjo, M.T.
Pembimbing 2 : Dr. Ir. Achmad Affandi, DEA

ABSTRAK

Arduino merupakan pengendali mikro *single-board* yang bersifat *open-source*, diturunkan dari *Wiring platform*, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. *Hardware*-nya memiliki prosesor Atmel AVR dan *software*-nya memiliki bahasa pemrograman tersendiri. Maka dari itu, dari beberapa alat atau teknologi yang memakai Arduino sebagai pusat kontrolnya, dibutuhkan kemampuan khusus untuk memahami bahasa pemrograman Arduino tersebut untuk *I/O* yang akan digunakan.

Dalam rancangan alat ini, pada tahap awal perangkaian akan disesediakan *database* program untuk Arduino, yang dimana berisikan *library* atau suatu pemrograman untuk Arduino agar *input* dan *output* dari Arduino dapat dikontrol sesuai dengan kebutuhan sipengguna tanpa harus membutuhkan kemampuan khusus untuk memahami arti perintah atau variabel dalam bahasa pemrograman Arduino tersebut.

Dalam hasil pengujian dan pengukuran dari masing – masing sensor, dari sensor DHT11 didapatkan nilai %Error sebesar 1.38% yang nilainya hampir mendekati nilai %Error dari alat *hygrometer* yaitu 1.108% yang dimana alat ini digunakan sebagai parameter atau tolak ukur untuk nilai suhu dan kelembaban, dan ditentukan nilai minimal maupun maksimal dari nilai suhu dan kelembaban didalam *prototype* data tersebut dilanjutkan ke kipas motor untuk menyala yang bertujuan menjaga atau memonitoring sirkulasi udara agar nilai suhu dan kelembaban didalam *prototype* stabil, serta pengujian dan pengukuran sensor *fingerprint* yang menggunakan metode *boolean* yang didapatkan besar nilai %Error sebesar 2.5%.

Kata Kunci : Arduino,*Fingerprint optical,* DHT11.

-----Halaman ini sengaja dikosongkan-----

CONNECTING WITH USER PROGRAMMABLE GENERIC I / O OF ARDUINO IMPLEMENTED WITH FINGERPRINT AND DHT11

Nama : Rendra Kurnia .R
Nama : Setyo Budi Utomo
Pembimbing 1 : Ir. Djoko Suprajitno Rahardjo, M.T
Pembimbing 2 : Dr. Ir. Achmad Affandi, DEA

ABSTRACT

Arduino is an open-source single-board micro controller, derived from Wiring platform, designed to facilitate the use of electronics in various fields. The hardware has an Atmel AVR processor and the software has its own programming language. Therefore, from some equipment or technology using Arduino as its control center, special abilities are required to understand the Arduino programming language for I / O to be used.

In the design of this tool, in the early stages the series will be provided with a program database for Arduino, which contains a library or a programming for Arduino so that input and output of the Arduino can be controlled according to the needs of the user without the need for special skills to understand the meaning of commands or variables in the language The Arduino programming.

In the test results and measurements of each sensor, from the sensor DHT11 obtained the value of Error 1.38% which value is almost close to the value of Error of the tool hygrometer is 1.108% which where the tool is used as a parameter or benchmark for the value of temperature and humidity, And determined the minimum or maximum value of the temperature and humidity value in the prototype data is continued to the motor fan for the light that aims to maintain or monitor the air circulation so that the temperature and humidity value in the prototype is stable, and testing and measuring fingerprint sensor using boolean method obtained large Error value is 2.5%.

Keywoards: *Arduino, Fingerprint Optical, DHT11*

-----Halaman ini sengaja dikosongkan-----

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga Tugas Akhir ini dapat terselesaikan dengan baik. Shalawat serta salam semoga selalu dilimpahkan kepada Rasulullah Muhammad SAW, keluarga, sahabat, dan umat muslim yang senantiasa meneladani beliau.

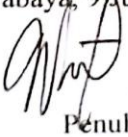
Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan diploma pada Bidang Studi Komputer Kontrol, Program Studi D3 Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya dengan judul:

***GENERIC PROGRAMMABLE* PENGHUBUNG PENGGUNA DENGAN *I/O* DARI ARDUINO YANG DITERAPKAN DENGAN *FINGERPRINT* DAN DHT11**

Penulis mengucapkan terima kasih kepada Ibu dan Bapak penulis yang memberikan berbagai bentuk doa serta dukungan tulus tiada henti, Bapak Ir. Djoko Suprajitno Rahardjo, M.T., dan bapak Dr. Ir. Achmad Affandi, DEA. atas segala bimbingan ilmu, moral, dan spiritual dari awal hingga terselesaikannya Tugas Akhir ini, kedua orang tua yang selalu memberikan doa, semangat, dan dukungannya kepada penulis Penulis juga mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Tugas Akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Tugas Akhir ini. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, 9 Juni 2017



Penulis

-----Halaman ini sengaja dikosongkan-----

DAFTAR ISI

HALAMAN

DAFTAR GAMBAR

HALAMAN

DAFTAR TABEL

HALAMAN

BAB I
PENDAHULUAN

1.1 Latar Belakang

Dunia Telekomunikasi berkembang begitu pesatnya di Indonesia seiring munculnya teknologi telepon tanpa kabel (*Wireless Telephone*). Teknologi telepon tanpa kabel yang dikembangkan di Indonesia terdiri dari dua platform yaitu teknologi GSM (*Globalsystem for Mobile Comunication*) dan CDMA (*Code Division Multiple Access*).

Masing - masing operator telekomunikasi memiliki jaringan yang beragam terkait dengan luas jangkauan areanya. Area terkecil dari layanan telepon tanpa kabel disebut *cell*. Oleh karena itu telepon tanpa kabel disebut juga telepon seluler. Untuk satu area *cell* biasanya terdapat satu perangkat BTS (Base Transceiver Station).

Dalam radius maksimal beberapa kilometer dari pemukiman (terkecuali didaerah pedalaman) bisa dipastikan terdapat sebuah *tower* pemancar dengan cirikhas (cat yang berwarna merah putih) dan pada ketinggian tertentu terpasang beberapa antena. Dan disamping tower tersebut terdapat sebuah bangunan seperti sebuah rumah yang bentuknya hampir sama seperti boks mobil barang yang disebut sebagai *shelter*, itulah salah satu komponen BTS (*Base Transceiver Station*) yang selama ini kerap disebut –sebut.

Jika dilihat dari segi arsitektur sebuah sistem jaringan selular, dimana BTS itu adalah sebagai salah satu sub – sistemnya, berfungsi sebagai pemancar dan penerima yang memberikan pelayanan radio kepada *mobile station* / *handphone*. Dan ada juga yang menyebut perangkat BTS dengan sebutan modem raksasa, sebab menyerupai perangkat *interface* antara *mobile station* dan MSC (*Mobile Switching Centre*).

Base Transceiver Station atau disingkat BTS merupakan sebuah infrastruktur telekomunikasi yang memfasilitasi komunikasi nirkabel antara piranti komunikasi dan jaringan operator. Piranti komunikasi penerima sinyal BTS bisa telepon, telepon seluler, jaringan nirkabel sementara operator jaringan yaitu GSM, CDMA, atau platform TDMA. BTS mengirimkan dan menerima sinyal radio ke perangkat *mobile* dan mengkonversi sinyal – sinyal tersebut menjadi sinyal digital untuk selanjutnya dikirim ke terminal lainnya untuk proses sirkulasi pesan atau

data. Nama lain dari BTS adalah *Base Station* (BS), *Radio Base Station* (RBS), atau node B (eNB). Hingga saat ini masyarakat belum bisa membedakan antara perangkat BTS dan menara BTS padahal menara BTS bukanlah BTS itu sendiri.

Banyaknya BTS yang dimiliki oleh sebuah operator telepon seluler menyebabkan munculnya kesulitan khususnya banyaknya data untuk masing - masing BTS yang terkait kegiatan operasional dan perawatan yang dilakukan dilapangan oleh para pelaksana lapangan. Data tersebut misalnya, lokasi BTS, Tipe kunci site, nama/no telepon Penjaga BTS, no telepon PLN bagian gangguan BTS dan nomor ID Pelanggan PLN untuk BTS. Pelaksana lapangan tidak mungkin menghafal semua data - data yang terkait dengan masing - masing BTS pada saat akan dilakukan kegiatan operasional dan perawatan. Hal ini akan berakibat munculnya keterlambatan penanganan gangguan ataupun penggunaan waktu yang tidak efisien sehingga menaikkan biaya operasional. Oleh karena itu diperlukan suatu alat atu teknologi berupa *prototype* yang menyerupai perangkat BTS untuk mengatasi masalah tersebut

Dan melihat dari segi penempatan serta kondisi dimana perangkat BTS tersebut ditempatkan, tentunya terdapat beberapa kekurangan dari sisi tertentu, seperti masalah keamanan dengan sensor *fingerprint* dengan *type optical* dan menjaga suhu dan kelembaban didalam perangkat tersebut agar komponen dapat bekerja secara maksimal serta mengurangi kemungkinan jika terjadi korosi atau pengkaratan yang dapat mengakibatkan masalah atau *troubleshoot* saat perangkat sedang bekerja.

1.2 Permasalahan

Yang menjadi rumusan masalah pada rancangan alat tugas akhir ini ada diantaranya :

- Dibutuhkan *input* dan *output* serta *library* untuk Arduino secara tidak menentu karena disesuaikan dengan kondisi atau kebutuhan dari *user* atau operator *maintenance* dari pihak BTS.
- Membuat *database* dari keseluruhan program untuk *input* dan *ouput* yang dibutuhkan oleh sipengguna meskipun ada beberapa *input* dan *output* yang tidak digunakan.
- Validasi dalam penggabungan *platform* atau aplikasi *generic* untuk pembuatan *input* dan *output* serta *prototype*.

1.3 Batasan Masalah

Dalam perancangan alat tugas akhir ini, adapun beberapa masalah yang memiliki batasan untuk memfokuskan pada pembuatan alat tugas akhir ini yang diantaranya :

- *Plant* yang dibuat adalah alat ukur suhu dan kelembaban serta pemindai pola sidik jari.
- Pola sidik jari yang diidentifikasi adalah pola sidik jari dalam keadaan bersih dan normal, atau tanpa adanya luka dan steril.
- Akuisisi data sidik jari langsung dilakukan oleh alat pemindai, sistem hanya melakukan pengolahan hasil ekstrasi ciri dari data citra yang diperoleh.
- Sensor DHT11 yang digunakan untuk mendeteksi suhu dan kelembaban dengan pusat pengontrolnya adalah Arduino Mega 2560 dan data yang diterima akan diproses untuk menyala tidaknya kipas.
- Diaplikasikan menggunakan *prototype* sederhana dengan objek suatu kondisi ruangan berupa *Equipment Box*.
- Nilai maksimal dan minimal suhu dan kelembaban didalam ruangan ditentukan oleh kebutuhan *user* serta kondisi dari isi didalam ruangan / *prototype*.

1.4 Tujuan

Tujuan dari penyusunan rancangan alat tugas akhir ini ditujukan kepada pihak pelaksana lapangan dalam kegiatan operasional dan perwatan atau *maintenance* jaringan operator seluler atau dikhususkan kepada operator dari pihak maintenance yang mengoperasikan perangkat BTS.

Adapun tujuan dari perancangan tugas akhir ini ialah :

- Membuat rancangan alat berupa *prototype* dari *Box* Pengaman Perangkat BTS (*Base Transceiver Station*) konvensional dengan penambahan pada sistem keamanan serta mengontrol suhu dan kelembaban didalam *box* yang dapat dijalankan secara otomatis.
- Mengimplementasikan rancangan alat yaitu dengan cara menggabungkan keseluruhan sisi dari program yang berisikan *database* untuk *software* serta rangkaian untuk *hardware* yang berisikan komponen *input* dan *output* yang dibutuhkan oleh *user* yang bisa dijalankan dalam waktu yang bersamaan atau *multitasking.*

1.5 Metodologi Penelitian

Dalam rancangan untuk tugas akhir ini yang berupa inovasi dari alat yang sudah ada namun mengotomatisasi yang mengacu pada 3 aspek yakni sektor keamanan, sektor otomatisasi serta sektor monitoring suhu dan kelembaban.

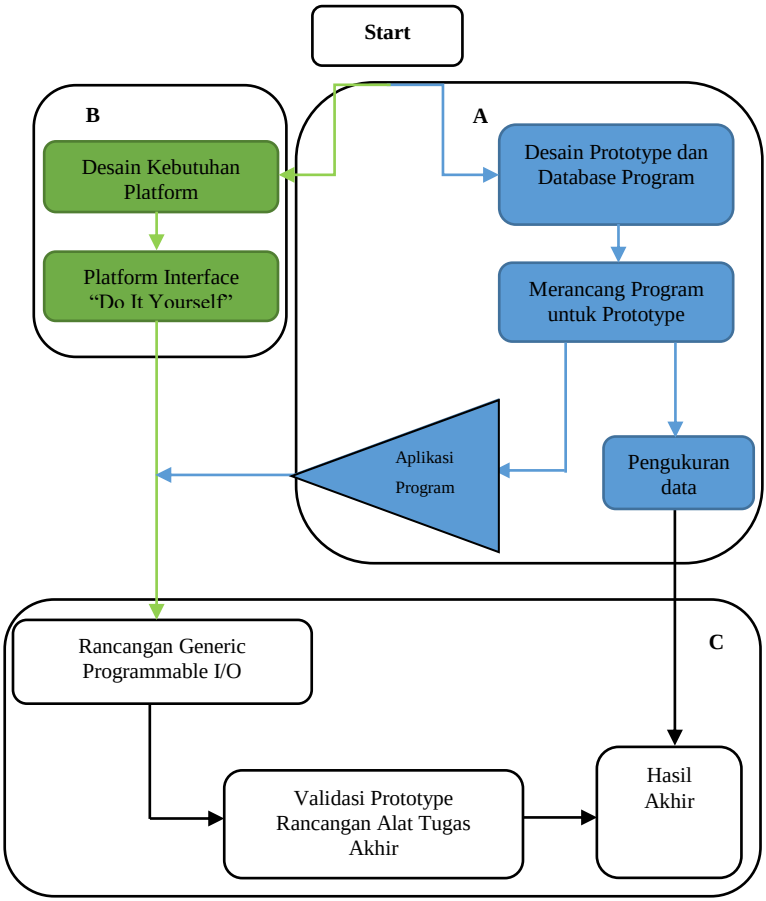
a. Studi Pustaka dan Survey Data Awal :

Pada tahap ini akan dilakukan studi pustaka berkaitan :

1) Artikel – artikel yang berkaitan dengan komponen dan sensor dari *input* dan *output* yang akan digunakan dalam proses pengamanan, suhu dan kelembaban, serta otomatisasi.
2) Karakteristik masing – masing komponen dan sensor dari *input* dan *output*.
3) Membuat rancangan desain alat dan penempatan / penyesuaian komponen dan sensor dari *input* dan *output*.
4) Penggunaan Arduino sebagai pusat pengendali atau kontrol dari alat.

b. Perencaan dan Pembuatan Alat :

Pada tahap ini akan dilakukan perancangan perangkat keras (*hardware*) sesuai dengan data yang telah di dapat dari studi pustaka. Dilanjutkan dengan kebutuhan bahan, komponen, dan peralatan yang diperlukan sesuai dengan perancangan alat. Kemudian disusul dengan membuat media atau objek yang sesuai dengan rancangan atau desain dan juga disertai dengan cara pembuatannya yang diperoleh dari studi pustaka.



Gambar 1.1 Diagram Blok Pembagian Tugas Kelompok

Keterangan :

Dari diagram blok pada Gambar 1.1 diatas dibagi menjadi 2 yang sesuai dengan bagiannya masing – masing untuk anggiota kelompok yang dibedakan dengan Blok "A" dan Blok "B". Blok "C" yang bermaksut memiliki keseluruhan kerja secara tim atau kelompok.

Pertama pada Blok "A" atau alur berwarna biru yang diartikan pada awalnya membuat *prototype* rancangan alat tugas

akhir dan *database* untuk keseluruhan program yang dibutuhkan, lalu langkah selanjutnya pembagian program yang didapatkan dari *database* yang disesuaikan dengan kebutuhan dari *user*. Setelah itu dibagi menjadi dua langkah yaitu pengukuran data awal dari program yang telah diambil dari *database* yang dibutuhkan oleh *user* untuk diproses atau mengevaluasi hasil dari data yang telah diambil agar program menjadi cepat dan tepat. Dan langkah selanjutnya yitu memverifikasi *type input* dan *output* dengan cara mengukur dan menguji coba dengan parameter sebuaha alat yang memiliki prinsip kerja yang sama dengan *type* dari *input* dan *output* tersebut

Lalu dari kedua bagian tersebut digabungkan dan menjadi sebuah *Generic Programmable I/O*. Dimana pada bagian ini rancangan alat yang masih berupa secara "kasar" karena masih uji coba dengan data data yang sebelumnya diambil.

Selanjutnya ada validasi *prototype* rancangan alat tugas akhir yang dimana hasil evaluasi dari proses sebelumnya agar rancangan alat lebih sempurna dan mempunyai data yang tepat sesuai dengan kebutuhan *user* dari rancangan alat.

Dan yang terakhir terdapat bagian hasil akhir atau Blok "C" yang dimana hasil yang sudah tepat dari platform maupun program yang dibutuhkan oleh *user* dan sesuai dengan data yang telah dievaluasi terus menerus hingga terancang dengan benar.

c. Perencanaan dan Pembuatan Software :

Pada tahap ini akan di lakukan perancangan program (*software*) dari bahasa pemrograman Arduino sesuai dengan data yang telah di dapat dan dipelajari dari studi pustaka, dan kemudian program tersebut di terapkan dan diupload ke Arduino.

d. Uji Coba dan Analisis Data :

Pada tahap ini akan di lakukan pengujian program yang dibutuhkan oleh *user* yang telah diambil dari *database*. Selanjutnya akan di lakukan juga pengukuran ketepatan *input* dan *output* berdasarkan data sementara yang telah diambil untuk rancangan alat.

1) Uji coba dilakukan dengan melakukan pengujian bahan, alat, maupun komponen yang dibutuhkan oleh rancangan alat tugas akhir agar data yang didapatkan memiliki hasil yang sesuai dengan perhitungan atau perencanaan sebelumnya.

2) Uji coba atau analisa data dari *input* dan *output* yang diterapkan pada *prototype* sebelum diaplikasikan ke *platform* agar tidak terjadi kesalahan pada saat penerapan.

e. Penyusunan Laporan

Pada tahap ini akan di lakukan penyusunan laporan hasil dari pembuatan rancangan alat dengan *prototype* yang sesuai dengan *platform* yang telah didesain dan dibutuhkan oleh *user* dan hasil akhir dari rancangan alat tersebut.

1.6 Sistematika Laporan

Pembahasan tugas akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut :

Bab I Pendahuluan

Bab ini meliputi latar belakang, permasalahan, tujuan penelitian, metodologi penelitian, sistematika laporan, dan relevansi.

Bab II Teori Dasar

Bab ini menjelaskan tentang tinjauan pustaka, konsep dari Sensor DHT11, Sensor *Fingerprint Optical* R305, Solenoid *Door Lock* 12v, LCD 16x2, Arduino Mega 2560, Kipas Motor DC 5v, dan *power supply.*

Bab III Perancangan Sistem

Bab ini membahas desain dan perancangan alat mekanikal dan elektrikal.

Bab IV Simulasi, Implementasi dan Analisis Sistem

Bab ini memuat hasil simulasi dan implementasi serta analisis data dari hasil tersebut.

Bab V Penutup

Bab ini berisi kesimpulan dan saran dari hasil pembahasan yang telah diperoleh.

1.7 Relevansi

Manfaat dari tugas akhir ini adalah mendukung kemajuan teknologi pada era sekarang ini dengan menerapkannya pada suatu perangkat yang berfungsi untuk menyalurkan jaringan telekomunikasi dari satu medium ke medium lain atau bisa disebut dengan sistem telekomunikasi tanpa kabel (*nircable*) serta membantu pekerjaan dari pihak operasional dan perawatan saat akan melakukan perbaikan dan

perawatan (*maintenance*) jika ada masalah tertentu saat pihak tersebut tidak bisa menangani pada waktu yang sama.

BAB II
TEORI DASAR

2.1 Tinjauan Pustaka

BTS (*Base Transceiver Station*) *box* yang kami buat ini bermaksud untuk pengaman komponen komponen yang berada pada selter BTS (Base Transceiver Station) BTS juga disebut sebagai RDS (*Radio Base Station)* memiliki beberapa komponen utama di dalamnya diantaranya adalah :

- Core atau Inti (COBA / COSA)
  COBA merupakan prosesor yang bekerja di mana dikendalikan oleh *software*. Modul atau perangkat *Core* pada BTS Siemens digunakan sebagai tepat penyimpanan *software* dan *database*, pembangkit waktu (*Clock Generator*), Alarm Handling, jalur antarmuka (*Interface Link*) serta menangani dan mengolah pesan.
- *Carrier Unit*
  CU (*Carrier Unit*) berfungsi untuk pemrosesan seluruh sinyal analog dan sinyal digital termasuk pengaturan RF pada suatu Carrier, sedangkan daya yang dipancarkan tergantung tipe band frekuensi operasi yang digunakan.
- DUAMCO (*Duplexer Amplifier Multi Coupler*)
  DUAMCO atau ACOM (*Combiner*) berisi *duplexer* untuk routing antara pemancar dan penerima pada antena yang sama dan beroperasi pada frekuensi yang sama juga terdapat filter untuk pemancar dan penerima sinyal pula. Pada GSM 900 dinamakan DUAMCOG dan untuk GSM 1800 (DCS) dinamakan DUAMCO. Umumnya hanya dapat meng-handle satu sektor saja untuk satu DUAMCO. Dapat juga sebagai penyaring untuk sinyal pemancar dan penerima.

Dari deskripsi di atas dapat di lihat bahwa komponen diatas merlukan sebuah panel *box* yang aman dan juga memiliki suhu dan kelembaban yang pas sehingga komponen eletronika yang ada di dalamnya dapt bekerja dan berfungsi dengan baik. Sedangkan *box* panel BTS (*Base Transceiver Station*) yang ada hanya berupa panel konvensional yang pengamanya hanya menggunakan kunci tuas dan tidak ada pengatur suhu dan kelembaban seperti pada Gambar 2.1.

Gambar 2.1 Contoh *Box* Perangkat BTS (*Base Transceiver Station*) Konvensional

Dari kondisi seperti box perangkat BTS pada Gambar 2.1 diatas, maka akan ditambahkan *box* pengaman untuk perangkat BTS (*Base Transceiver Station*) dengan menggunakan sensor DHT11 yaitu sensor suhu dan kelembaban untuk mempertahankan kinerja komponen didalamnya agar dapat awet dan bekerja sesuai fungsinya dengan adanya beberapa komponen yang terbuat dari karbon jika menjadi lembab, panas, dapat merusak komponen sedangkan yang bersasal dari bahan *film carbon* dapat bekerja pada suhu di antara -55°C hingga 155°C. Selain itu juga di lengkapi dengan sensor *fingerprint* dengan *type optical* sebagai keamanan karena keamanan *box* panel perangkat BTS yang ada saat ini hanyalah menggunakan kunci tuas. Kekurangan kunci tuas adalah komponennya yang relatif lebih mudah untuk diutak – atik, sehingga lebih rentan jika dibuka secara paksa. Maka dari itu,untuk sektor pengamanannya, pada rancangan akan diberikan sensor *fingerprint* yang dimana pada *box* panel perangkat BTS hanya dapat dibuka menggunakan sidik jari yang telah dikenali atau sidik jari penggunanya mengingat tempat dan lokasi *box* panel perangkat BTS yang seringkali berada pada daerah yang jauh dari keramaian sehingga keamanan sangat di perlukan di sini, selain itu komponen yang berada dalam *box* panel perangkat BTS merupakan komponen – komponen yang mempunyai nilai finansial yang tinggi. Pengamanan yang di maksud untuk menggantikan kunci manual yang ada ialah pada saat sensor *fingerprint* telah mengenali sidik jari sipengguna maka solenoid

door lock akan terbuka sehingga pintu dari *box* panel perangkat BTS dapat dibuka.

2.2 Sensor Fingerprint Optical

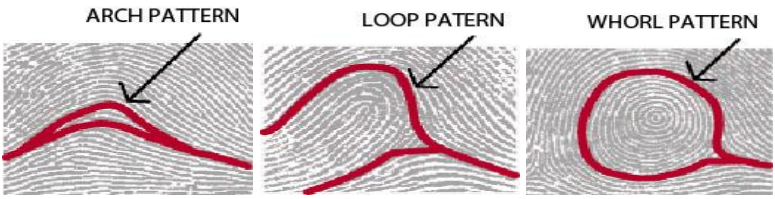
Sebagai pengaman dari akses "jalan masuk dan keluar" kedalam ruang atau *prototype* tersebut, maka digunakan Sensor *Fingerprint Optical* yang berfungsi untuk mengidentifikasi atau memindai pola sidik jari dari *user* atau pengguna yang akan mengakses prototype tersebut. Secara sederhana sensor *fingerprint* bekerja dengan cara "merekam" sidik jari seseorang, lalu menyimpan pola khasnya. Identifikasi dilakukan dengan mencocokkan data yang telah disimpan tersebut. Jika dinyatakan sama, akses untuk *prototype* otomatis akan terbuka.

Seperti halnya bagian tubuh yang lain, sidik jari terbentuk karena faktor genetik dan lingkungan. Kode genetik pada DNA memberi perintah untuk terbentuknya janin yang secara spesifik membentuk hasil secara *random* (acak) [3].

Sidik jari terdiri dari banyak garis menonjol yang cenderung melingkar – lingkar. Hal ini bisa terlihat jelas, salah satunya ketika kita membuat cap jari menggunakan tinta pada surat – surat resmi. Dari sini bisa dilihat, satu sidik jari saja memiliki banyak pola rumit. Jika semua pola ini digunakan, proses identifikasi sidik jari akan memakan waktu terlalu lama. Sebaliknya, jika pola yang diambil terlalu sederhana, kemungkinan pemindaian kurang akurat [4]. Jadi, walaupun sidik jari terlihat sama bila dilihat sekilas, bagi penyelidik terlatih atau dengan *software* khusus akan terlihat perbedaannya. Sebelum kita berbicara tentang alat pemindai sidik jari, kita akan berbicara tentang sidik jari tersebut.

- Pola Sidik Jari

  Secara umum, sidik jari dapat dibedakan menjadi beberapa tipe menurut *Henry Classification System*, yaitu *loop pattern*, *whorl pattern*, dan *arch pattern* [3] seperti pada Gambar 2.2 berikut ini.



Gambar 2.2 Macam – macam Pola Sidik Jari Manusia.

Sekedar informasi, hampir 2/3 manusia memiliki sidik jari dengan tipe *loop pattern*, 1/3 lainnya memiliki tipe *whorl pattern* dan hanya 5-10% yang memiliki tipe *arch pattern* [3].

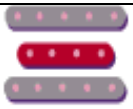
Sidik jari adalah gurat – gurat yang terdapat di kulit ujung jari. Sidik jari berfungsi untuk memberi gaya gesek lebih besar agar jari dapat memegang benda – benda lebih erat. Sistem pengamanan dengan menggunakan sidik jari sudah mulai dipergunakan di Amerika oleh seorang bernama *E. Henry* pada tahun 1902. *Henry* menggunakan metode sidik jari untuk melakukan identifikasi pekerja dalam rangka mengatasi pemberian upah ganda. Sistem *Henry* menggunakan pola *ridge* (*Ridge* = punggung alur pada kulit, baik pada tangan atau kaki), yang terpusat pola jari tangan, jari kaki, khususnya telunjuk. Untuk memperoleh gambar pola *ridge*, dilakukan dengan cara menggulung jari yang diberi tinta pada suatu kartu cetakan hingga dihasilkan suatu pola *ridge* yang unik bagi masing- masing individu. Para pakar membuktikan bahwa tidak ada dua individu yang mempunyai pola *ridge* yang serupa. Pola *ridge* tidaklah diwariskan. Pola *ridge* dibentuk waktu embrio, dan tidak pernah berubah seumur hidup. Perubahan ridge hanya dapat terjadi akibat trauma, missal akibat luka-luka, terbakar, penyakit, atau penyebab lainnya. Sistem biometrika sidik jari merupakan sistem yang paling banyak digunakan saat ini karena memiliki tingkat akurasi yang tinggi dan mudah untuk diterapkan [3]. Dari hasil penelitian, ditemukan 9 macam pola utama *pappilary ridge*, antara lain :

1. *Loop* : Terdiri dari satu atau lebih kurva bebas dari *ridge* dan sebuah *delta*.
2. *Arch* : Membentuk pola dengan *ridge* berada diatas *ridge* yang lain dalam bentuk lengkungan umum.
3. *Whorl* : Pola ini terdiri dari satu atau lebih kurva bebas *ridge* dan dua buah delta.
4. *Tented Arch* : Pola ini terdiri dari paling tidak sebuah *ridge* yang melengkung keatas yang kemudian bercabang menjadi dua *ridge*.
5. *Double Loop* : Pola ini membentuk dua formasi lengkungan yang lalu berpisah, dengan dua titik *delta*.
6. *Central Pocket Loop* : Terdiri dari satu atau lebih kurva *ridge* dan dua titik *delta*.

7. *Accidental* : Pola ini mempunyai dua titik *delta*. Satu *delta* akan berhubungan dengan lengkungan keatas, dan *delta* yang lain terhubung dengan lengkungan yang lain.
8. *Composite* : Terdiri dari gabungan dua atau lebih pola yang berbeda.
9. *Lateral Pocket Loop* : Pola ini terdiri dari dua lengkungan yang terpisah. Ada dua titik dua *delta*.

Dan pada Tabel 2.1 berikut ini merupakan gambar ukuran – ukuran karakteristik anatomi pola tersebut.

Tabel 2.1 Variasi Pola *Ridge*

	**Ridge**	Mempunyai ketegasan jarak ganda dari permulaan ke-akhir, sebagai lebar *ridges* satu dengan lainya.
	**Evading Ends**	Dua *ridge* dengan arah berbeda berjalan sejajar satu sama lain kurang dari 3mm.
	**Bifurcation**	Dua *ridge* dengan arah berbeda berjalan sejajar satu sama lain kurang dari 3mm.
	**Dot**	Bagian *ridges* adalah tidak lagi sebanding dengan *ridges* yang berdekatan.
	**Eye**	*Ridges* merobek dan menggabungkan lagi di dalam 3mm.
	**Island**	*Ridges* merobek dan tidak bergabung lagi, kurang dari 3mm dan tidak lebih dari 6mm. Area yang terlampir adalah *Ridge.*

	**Enclosed Ridge**	*Ridges* tidak lebih panjang dibanding 6mm antara dua ridges.
	**Enclosed Loop**	Yang tidak mempola menentukan pengulangan antara dua atau lebih *ridges* parallel.
	**Specialties**	*Rare ridge* membentuk seperti tanda tanya dan sangkutan pemotong.
	**Fork**	Dua *ridges* dihubungkan oleh sepertiga *ridges* tidak lebih panjang dibanding 3mm.
	**Hook**	*Ridges* merobek; satu *ridges* tidaklah lebih panjang dibanding 3mm.

Pola – pola sidik jari seperti inilah yang digunakan untuk membedakan sidik jari secara umum. Namun untuk mesin pembaca sidik jari, pembedaan seperti ini belumlah cukup. Karena itulah mesin pemindai sidik jari dilengkapi dengan metode pengenalan lain yang disebut *Minutiae* [3].

- *Minutiae*

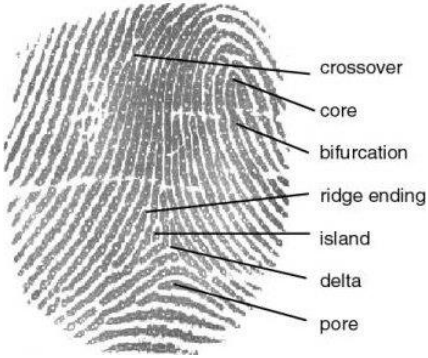
Minutiae berasal dari bahasa Inggris yang berarti "barang yang tidak berarti" atau "rincian yang tidak penting". Seperti artinya, *minutiae* sebenarnya merupakan rincian sidik jari yang tidak penting bagi kita, tetapi bagi sebuah mesin pemindai sidik jari, itu adalah detail yang sangat diperhatikan [3].

Untuk lebih jelasnya, *minutiae* pada sidik jari adalah titik – titik yang mengacu pada :

1. *crossover* (persilangan dua garis)
2. *core* (putar-balikan sebuah garis)
3. *bifurcation* (percabangan sebuah garis)
4. *ridge ending* (berhentinya sebuah garis)

5. *island* (sebuah garis yang sangat pendek)
6. *delta* (pertemuan dari tiga buah garis yang membentuk sudut)
7. *pore* (percabangan sebuah garis yang langsung diikuti dengan menyatunya kembali percabangan tersebut sehingga membentuk sebuah lingkaran kecil).

Atau lebih jelasnya seperti yang terdapat pada Gambar 2.3 dibawah ini.



Gambar 2.3 Minutiae pada Sidik Jari

Area papillary ridge kadang – kadang dikenal sebagai *pattern area*. Masing – masing pola *papillary ridge* menghasilkan suatu bentuk pola area yang berbeda. Pusat gambar jari mencerminkan pola area, dikenal sebagai inti atau *core point*. Bagian *ridges* yang berwujud dua parallel yang berbeda mengelilingi pola area inti disebut *type lines*. Titik awal percabangan dua *ridge* disebut *delta*. Proses perpecahan sebuah garis menjadi dua garis *ridge* disebut *bifurcation*. Banyaknya persimpangan *ridge* di dalam pola area disebut suatu *ridge count*. Komputer *Tormography* dapat digunakan untuk mendeteksi titik – titik tersebut berdasarkan sumbu koordinat x – y. (EkoNugroho, 2009) [4].

Mesin pemindai sidik jari akan mencari titik – titik ini dan membuat pola dengan menghubungkan titik-titik tersebut. Pola yang didapat dari menghubungkan titik – titik inilah yang nantinya akan digunakan untuk melakukan pencocokan bila ada jari yang dipindai. Jadi, sebenarnya mesin sidik jari tidak mencocokkan gambar, tapi mencocokkan pola yang didapat dari *minutiae* ini.

Mesin pemindai sidik jari bekerja dengan mengambil gambar dari sidik jari dan membedakan setiap pola atau alur dari sidik jari tersebut.

Setelah mengetahui bagaimana pola dari sidik jari beserta rinciannya, maka dilanjutkan dengan cara kerja dari sensor *fingerprint optical*. Yang dimana inti dari sensor optikal adalah adanya CCD (*Charge Couple Device*) yang cara kerjanya sama seperti sistem sensor yang terdapat pada kamera digital atau *camcorder*. CCD merupakan chip silikon yang terbentuk dari ribuan bahkan jutaan dioda fotosensitif yang disebut *photosites*, *photodelements*, atau disebut juga *pixel*. Tiap *photosite* menangkap satu titik objek, kemudian dirangkai dengan hasil tangkapan *photosite* lain menjadi satu gambar seperti pada Gambar 2.4 dibawah ini.

Gambar 2.4 Contoh Pola *Loop Pattern*

Bila mengambil contoh pada kamera, saat menekan tombol "*capture*" pada kamera digital, sel pengukur intensitas cahaya akan menerima dan merekam setiap cahaya yang masuk menurut intensitasnya. Dalam waktu yang sangat singkat, tiap titik *photosite* akan merekam cahaya yang diterima dan diakumulasikan dalam sinyal elektronis.

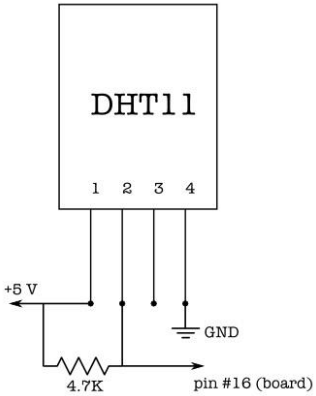
Gambar yang sudah dikalkulasikan dalam gambar yang sudah direkam dalam bentuk sinyal elektronis akan dikalkulasi untuk kemudian disimpan dalam bentuk angka – angka digital. Angka tersebut akan digunakan untuk menyusun ulang gambar untuk ditampilkan kembali. Perekaman gambar yang dilakukan oleh CCD sebenarnya dalam format *grayscale* atau *monochrome* dengan 256 macam intensitas warna dari putih sampai hitam.

Pemindai sidik jari optikal dianggap menghasilkan tingkat keamanan yang tinggi, karena tidak bisa dipalsukan dengan fotokopi

sidik jari, sidik jari tiruan, atau bahkan dengan cetak lilin yang mendetil dengan guratan – guratan kontur sidik jari sekalipun.

2.3 Sensor DHT11

DHT11 adalah sensor Suhu dan Kelembaban, yang memiliki keluaran sinyal digital yang dikalibrasi dengan sensor suhu dan kelembaban yang kompleks. Teknologi ini memastikan keandalan tinggi dan sangat baik stabilitasnya dalam jangka panjang. mikrokontroler terhubung pada kinerja tinggi sebesar 8 bit. Sensor ini termasuk elemen resistif dan perangkat pengukur suhu NTC. Memiliki kualitas yang sangat baik, respon cepat, kemampuan anti – gangguan dan keuntungan biaya tinggi kinerja [5]. Berikut merupakan tampilan pinout dari sensor DHT11 pada Gambar 2.5 dibawah ini.



Gambar 2.5 Pinout Sensor DHT11

Setiap sensor DHT11 memiliki fitur kalibrasi sangat akurat dari kelembaban ruang kalibrasi. Koefisien kalibrasi yang disimpan dalam memori program OTP, sensor internal mendeteksi sinyal dalam proses, kita harus menyebutnya koefisien kalibrasi. Sistem antarmuka tunggal – kabel serial terintegrasi untuk menjadi cepat dan mudah. Kecil ukuran, daya rendah, sinyal transmisi jarak hingga 20 meter, sehingga berbagai aplikasi dan bahkan aplikasi yang paling menuntut. Produk ini memiliki 4 pin baris paket tunggal. Dan berikut pada Tabel 2.2 merupakan spesifikasi dari sensor DHT11

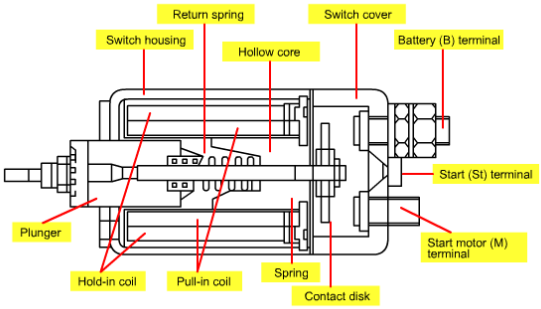
Tabel 2.2 Spesifikasi Sensor DHT11

Model	DHT11
Sumber Tegangan	3-5.5V DC
Sinyal Keluaran	Sinyal digital melalui single-bus
Unsur Perasa (Komponen)	Resistor polimer
Batas Minimal dan Maksimal Pengukuran	Suhu : 0-50 Celcius. Kelembaban : 20-90%RH.
Ketepatan Pengukuran	Suhu : ± 2.0 Celcius. Kelembaban : ± 4%RH (Maks. ± 5%RH).
Nilai Resolusi atau Kepekaan	Suhu : 0.1 Celcius. Kelembaban : 1%RH.
Nilai Pengulangan	Suhu : ± 1 Celcius. Kelembaban : ± 1%RH.
Nilai Histeresis Kelembaban	±1%RH.
Lama Waktu Kestabilan	±0.5%RH/tahun.
Periode Perasa (Komponen)	Rata – rata 2 detik.
Dimensi atau Ukuran	12mmx15.5mmx5.5mm.

2.4 Solenoid Door Lock

Solenoid *Door Lock* adalah salah satu solenoid yang difungsikan khusus sebagai solenoid untuk pengunci pintu secara elektronik. Solenoid ini mempunyai dua sistem kerja, yaitu *Normally Close* (NC) dan *Normally Open* (NO).

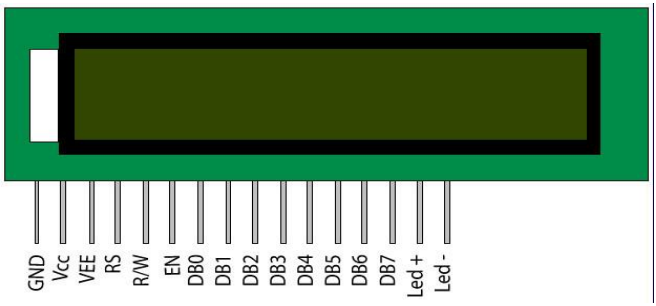
Perbedaanya adalah jika cara kerja solenoid NC apabila diberi tegangan, maka solenoid akan memanjang (menutup). Dan untuk cara kerja dari Solenoid NO merupakan kebalikan dari cara kerja Solenoid NC. Biasanya kebanyakan solenoid *door lock* membutuhkan input atau tegangan kerja 12V DC tetapi ada juga solenoid *door lock* yang hanya membutuhkan input tegangan 5V DC dan sehingga dapat langsung bekerja dengan tegangan *output* dari pin IC digital. Namun jika menggunakan solenoid *door lock* yang 12V DC, maka dibutuhkan *power supply* 12V dan sebuah relay untuk mengaktifkannya. Gambar 2.6 dibawah ini merupakan bagian – bagian dari Solenoid.



Gambar 2.6 Solenoid Door Lock

2.5 LCD 16x2

LCD (*Liquid Crystal Display*) adalah suatu jenis media tampil yang menggunakan kristal cair sebagai penampil utama. LCD sudah digunakan diberbagai bidang misalnya alat – alat elektronik seperti televisi, kalkulator, atau pun layar komputer. LCD berfungsi sebagai media untuk menampilkan suatu data, baik karakter, huruf ataupun grafik. Tampilan pinout LCD 16x2 yang terdapat pada Gambar 2.7 di bawah ini.



Gambar 2.7 Pinout LCD 16x2

Untuk keperluan antar muka suatu komponen elektronika dengan mikrokontroller, perlu diketahui fungsi dari setiap kaki yang ada pada komponen tersebut.

- **Kaki 1 (GND)** : Kaki ini berhubungan dengan tegangan 0 Volt (*Ground*).

- **Kaki 2 (VCC)** : Kaki ini berhubungan dengan tegangan +5 Volt yang merupakan tegangan untuk sumber daya.
- **Kaki 3 (VEE/VLCD)** : Tegangan pengatur kontras LCD, kaki ini terhubung pada cermet. Kontras mencapai nilai maksimum pada saat kondisi kaki ini pada tegangan 0 Volt.
- **Kaki 4 (RS)** : *Register Select*, kaki pemilih register yang akan diakses. Untuk akses ke Register Data, logika dari kaki ini adalah 1 dan untuk akses ke Register Perintah, logika dari kaki ini adalah 0.
- **Kaki 5 (R/W)** : Logika 1 pada kaki ini menunjukan bahwa modul LCD sedang pada mode pembacaan dan logika 0 menunjukan bahwa modul LCD sedang pada mode penulisan. Untuk aplikasi yang tidak memerlukan pembacaan data pada modul LCD, kaki ini dapat dihubungkan langsung ke *Ground*.
- **Kaki 6 (E)** : *Enable Clock* LCD, kaki mengaktifkan *clock* LCD. Logika 1 pada kaki ini diberikan pada saat penulisan atau pembacaan data.
- **Kaki 7 – 14 (D0 – D7)** : Data bus, kedelapan kaki LCD ini adalah bagian dimana aliran data sebanyak 4 bit ataupun 8 bit mengalir saat proses penulisan maupun pembacaan data.
- **Kaki 15 (Anoda)** : Berfungsi untuk tegangan positif dari *backlight* LCD sekitar 4,5 volt (hanya terdapat untuk LCD yang memiliki *backlight*).
- **Kaki 16 (Katoda)** : Tegangan negatif backlight LCD sebesar 0 Volt (hanya terdapat pada LCD yang memiliki *backlight*).

2.6 Arduino Mega 2560

Microcontroller 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* dan *output*. Dengan kata lain, *Microcontroller* adalah suatu alat elektronika digital yang mempunyai masukkan dan keluaran serta kendali dengan program yang bisa ditulis dan dihapus dengan cara khusus, cara kerja *microcontroller* sebenarnya membaca dan menulis data. Sekedar contoh, bayangkan diri Anda saat mulai belajar membaca dan menulis, ketika Anda sudah bisa melakukan hal itu Anda bisa membaca tulisan apapun baik buku, cerpen, artikel dan sebagainya, dan Andapun bisa pula menulis hal – hal sebaliknya. Begitu pula jika Anda sudah mahir membaca dan menulis data maka Anda

dapat membuat program untuk membuat suatu sistem pengaturan otomatik menggunakan *microcontroller* sesuai dengan keinginan Anda. *Microcontroller* merupakan komputer didalam chip yang digunakan untuk mengontrol peralatan elektronik, yang menekankan efisiensi dan efektifitas biaya. Secara harfiahnya bisa disebut "pengendali kecil" dimana sebuah sistem elektronik yang sebelumnya banyak memerlukan komponen – komponen pendukung seperti IC TTL dan CMOS dapat direduksi / diperkecil dan akhirnya terpusat serta dikendalikan oleh *microcontroller* ini.

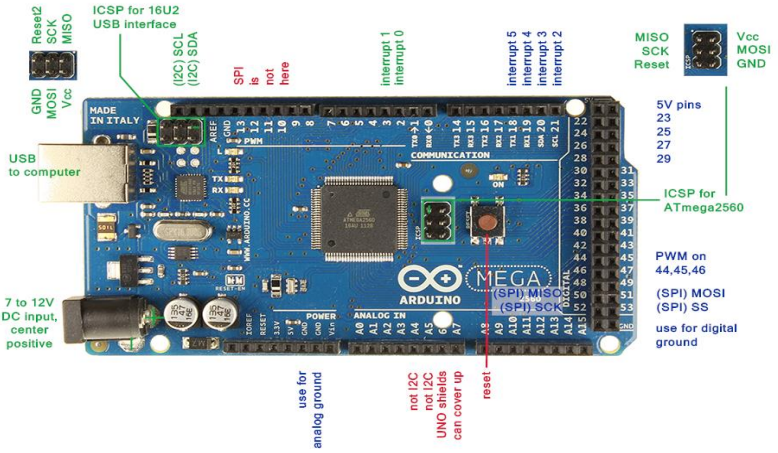
Microcontroller digunakan dalam produk dan alat yang dikendalikan secara automatis, seperti sistem kontrol mesin, *remote control*, mesin kantor, peralatan rumah tangga, alat berat, dan mainan. Dengan mengurangi ukuran, biaya, dan konsumsi tenaga dibandingkan dengan mendesain menggunakan mikroprosesor memori, dan alat *input / output* yang terpisah, kehadiran mikrokontroler membuat kontrol elektrik untuk berbagai proses menjadi lebih ekonomis. Dengan penggunaan *microcontroller* ini maka :

- Sistem elektronik akan menjadi lebih ringkas.
- Rancang bangun sistem elektronik akan lebih cepat karena sebagian besar dari sistem adalah perangkat lunak yang mudah dimodifikasi.
- Pencarian gangguan lebih mudah ditelusuri karena sistemnya yang kompak.

Agar sebuah *microcontroller* dapat berfungsi, maka *microcontroller* tersebut memerlukan komponen eksternal yang kemudian disebut dengan sistem minimum. Untuk membuat sistem minimal paling tidak dibutuhkan sistem *clock* dan reset, walaupun pada beberapa *microcontroller* sudah menyediakan sistem *clock* internal, sehingga tanpa rangkaian eksternal pun *microcontroller* sudah beroperasi.

Arduino Mega 2560 adalah papan pengembangan *microcontroller* yang berbasis Arduino dengan menggunakan chip ATmega2560. Board ini memiliki pin *I/O* yang cukup banyak, sejumlah 54 buah digital *I/O* pin (15 pin diantaranya adalah PWM), 16 pin analog *input*, 4 pin UART (serial port *hardware*). Arduino Mega 2560 dilengkapi dengan sebuah *oscillator* 16 Mhz, sebuah port USB, power jack DC, ICSP header, dan tombol reset. Board ini sudah sangat lengkap, sudah memiliki segala sesuatu yang dibutuhkan untuk sebuah *microcontroller*. Dengan penggunaan yang cukup sederhana, sipengguna hanya tinggal

menghubungkan power dari USB ke PC, atau melalui adaptor AC/DC ke jack-DC [8]. Berikut ini pada Gambar 2.8 merupakan tampilan pinout dari Arduino Mega 2560 :



Gambar 2.8 Pinout Arduino Mega 2560

Dan pada Tabel 2.3 berikut ini merupakan spesifikasi dari Arduino Mega 2560 :

Tabel 2.3 Spesifikasi dari Arduino Mega 2560

Chip Microcontroller	**ATmega 2560**
Tegangan Operasi	5V
Input Voltage (yang direkomendasikan, via jack-DC)	7-12V
Input Voltage (limit, via jack-DC)	6-20V
Jumlah pin *I/O* digital	54 buah (6 pin digunakan sebagai output PWM)
Jumlah pin *input* analog	16 buah
Arus DC tiap pin *I/O*	40 mA
Arus DC untuk pin 3.3V	50 mA
Flash Memory	256 KB (8 KB digunakan

	untuk *bootloader*)
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
Dimensi	101.5 mm x 53.4 mm
Berat	37 g

Arduino Mega dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal. Sumber daya dipilih secara otomatis. Sumber daya eksternal (non-USB) dapat berasal baik dari adaptor AC – DC atau baterai. Adaptor dapat dihubungkan dengan mencolokkan steker 2,1 mm yang bagian tengahnya terminal positif ke *jack* sumber tegangan pada papan. Jika tegangan berasal dari baterai dapat langsung dihubungkan melalui *header* pin GND dan pin Vin dari konektor POWER [8].

Papan Arduino ATmega 2560 dapat beroperasi dengan pasokan daya eksternal 6 Volt sampai 20 Volt. Jika diberi tegangan kurang dari 7 Volt, maka pin 5 Volt mungkin akan menghasilkan tegangan kurang dari 5 Volt dan ini akan membuat papan menjadi tidak stabil. Jika sumber tegangan menggunakan lebih dari 12 Volt, regulator tegangan akan mengalami panas berlebihan dan bisa merusak papan. Rentang sumber tegangan yang dianjurkan adalah 7 Volt sampai 12 Volt [8]. Pin tegangan yang tersedia pada papan Arduino adalah sebagai berikut :

- **VIN** : Adalah *input* tegangan untuk papan Arduino ketika menggunakan sumber daya eksternal (sebagai "saingan" tegangan 5 Volt dari koneksi USB atau sumber daya ter – regulator lainnya). Anda dapat memberikan tegangan melalui pin ini, atau jika memasok tegangan untuk papan melalui *jack power*, kita bisa mengakses / mengambil tegangan melalui pin ini.
- **5V** : Sebuah pin yang mengeluarkan tegangan ter – regulator 5 Volt, dari pin ini tegangan sudah diatur (ter – regulator) dari regulator yang tersedia (built-in) pada papan. Arduino dapat diaktifkan dengan sumber daya baik berasal dari *jack power* DC (7-12 Volt), konektor USB (5 Volt), atau pin Vin pada board (7-12 Volt). Memberikan tegangan melalui pin 5V atau 3,3V secara langsung tanpa melewati regulator dapat merusak papan Arduino.
- **3V3** : Sebuah pin yang menghasilkan tegangan 3,3 Volt. Tegangan ini dihasilkan oleh regulator yang terdapat pada papan (on-board). Arus maksimum yang dihasilkan adalah 50 mA.
- **GND** : Pin *Ground* atau Massa.

- **IOREF** : Pin ini pada papan Arduino berfungsi untuk memberikan referensi tegangan yang beroperasi pada *microcontroller*. Sebuah perisai (*shield*) dikonfigurasi dengan benar untuk dapat membaca pin tegangan IOREF dan memilih sumber daya yang tepat atau mengaktifkan penerjemah tegangan (*voltage translator*) pada output untuk bekerja pada tegangan 5 Volt atau 3,3 Volt.

Untuk memori, Arduino ATmega 2560 memiliki 256 KB *flash memory* untuk menyimpan kode (8 KB digunakan untuk *bootloader*), 8 KB SRAM dan 4 KB EEPROM (yang dapat dibaca dan ditulis dengan perpustakaan EEPROM).

Sedangkan *input* dan *output* dari Arduino Atmega 2560, masing – masing dari 54 digital pin pada Arduino Mega dapat digunakan sebagai *input* atau *output*, menggunakan fungsi *pinMode(), digitalWrite()*, dan *digitalRead().* Arduino Mega beroperasi pada tegangan 5 volt. Setiap pin dapat memberikan atau menerima arus maksimum 40 mA dan memiliki resistor pull-up internal (yang terputus secara *default*) sebesar 20 – 50 kΩ. Selain itu, beberapa pin memiliki fungsi khusus, antara lain :

- Serial yang digunakan untuk menerima (RX) dan mengirimkan (TX) data serial TTL, atau lebih jelasnya pembagian pinout seperti pada Tabel 2.4 berikut ini

Tabel 2.4 Tabel Pin Serial RX dan TX

Nomor Pin	Nama Pin	Peta Nama Pin
2	PE0 (RXD0/PCINT8)	Digital pin 0 (RX0)
3	PE1 (TXD0)	Digital pin 1 (TX0)
12	PH0 (RXD2)	Digital pin 17 (RX2)
13	PH1 (TXD2)	Digital pin 16 (TX2)
45	PD2 (RXDI/INT2)	Digital pin 19 (RX1)
46	PD3 (TXD1/INT3)	Digital pin 18 (TX1)
63	PJ0 (RXD3/PCINT9)	Digital pin 15 (RX3)
64	PJ1	Digital pin 14

	(TXD3/PCINT10)	(TX3)

- Eksternal Interupsi : Pin ini dapat dikonfigurasi untuk memicu sebuah interupsi pada nilai yang rendah, meningkat atau menurun, atau perubah nilai, atau lebih jelasnya pembagian pinout seperti pada Tabel 2.5 berikut ini

Tabel 2.5 Tabel Pin Eksternal Interupsi

Nomor Pin	Nama Pin	Peta Nama Pin
6	PE4 (OC3B/INT4)	Digital pin 2 (PWM)
7	PE5 (OC3C/INT5)	Digital pin 3 (PWM)
43	PD0 (SCL/INT0)	Digital pin 21 (SCL)
44	PD1 (SDA/INT1)	Digital pin 20 (SDA)
45	PD2 (RXDI/INT2)	Digital pin 19 (RX1)
46	PD3 (TXD1/INT3)	Digital pin 18 (TX1)

- **SPI** : Pin ini mendukung komunikasi SPI menggunakan SPI library. Pin SPI juga terhubung dengan header ICSP, yang secara fisik kompatibel dengan Arduino *Uno*, Arduino *Duemilanove* dan Arduino *Diecimila*, untuk lebih jelasnya pembagian pinout seperti pada Tabel 2.6 berikut ini

Tabel 2.6 Tabel Pin *SPI*

Nomor Pin	Nama Pin	Peta Nama Pin
19	PB0 (SS/PCINT0)	Digital pin 53 (SS)
20	PB1 (SCK/PCINT1)	Digital pin 52 (SCK)
21	PB2 (MOSI/PCINT2)	Digital pin 51 (MOSI)
22	PB3 (MISO/PCINT3)	Digital pin 50 (MISO)

- **LED** : Pin 13. Tersedia secara built-in pada papan Arduino ATmega LED terhubung ke pin digital 13. Ketika pin diset bernilai HIGH, maka LED menyala (ON), dan ketika pin diset bernilai LOW, maka LED padam (OFF).
- **TWI** : Pin 20 (SDA) dan pin 21 (SCL). Yang mendukung komunikasi TWI menggunakan *Wire Library*. Perhatikan bahwa pin ini tidak di lokasi yang sama dengan pin TWI pada Arduino *Duemilanove* atau Arduino *Diecimila*.

Arduino Mega 2560 memiliki 16 pin sebagai analog *input*, yang masing – masing menyediakan resolusi 10 bit (yaitu 1024 nilai yang berbeda). Secara default pin ini dapat diukur / diatur dari mulai *Ground* sampai dengan 5 Volt, juga memungkinkan untuk mengubah titik jangkauan tertinggi atau terendah mereka menggunakan pin AREF dan fungsi Analog *Reference().* Ada beberapa pin lainnya yang tersedia, antara lain :

- AREF : Referensi tegangan untuk *input* digunakan dengan fungsi Analog *Reference().*
- RESET: Jalur LOW ini digunakan untuk me – reset (menghidupkan ulang) *microcontroller*. Jalur ini biasanya digunakan untuk menambahkan tombol reset pada *shield* yang menghalangi papan utama Arduino.

Arduino Mega 2560 memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, dengan Arduino lain, atau dengan *microcontroller* lainnya. Arduino ATmega 328 menyediakan 4 hardware komunikasi serial UART TTL (5 Volt). Sebuah chip ATmega 16U2 (ATmega 8U2 pada papan Revisi 1 dan Revisi 2) yang terdapat pada papan digunakan sebagai media komunikasi serial melalui USB dan muncul sebagai COM Port Virtual (pada Device komputer) untuk berkomunikasi dengan perangkat lunak pada komputer, untuk sistem operasi Windows masih tetap memerlukan file inf, tetapi untuk sistem operasi OS X dan Linux akan mengenali papan sebagai port COM secara otomatis. Perangkat lunak Arduino termasuk didalamnya serial monitor memungkinkan data tekstual sederhana dikirim ke dan dari papan Arduino. LED RX dan TX yang tersedia pada papan akan berkedip ketika data sedang dikirim atau diterima melalui chip USB-to-serial yang terhubung melalui USB komputer (tetapi tidak untuk komunikasi serial seperti pada pin 0 dan 1) [8].

Sebuah *Serial Llibrary Software* memungkinkan untuk komunikasi serial pada salah satu pin digital Mega 2560. ATmega 2560 juga

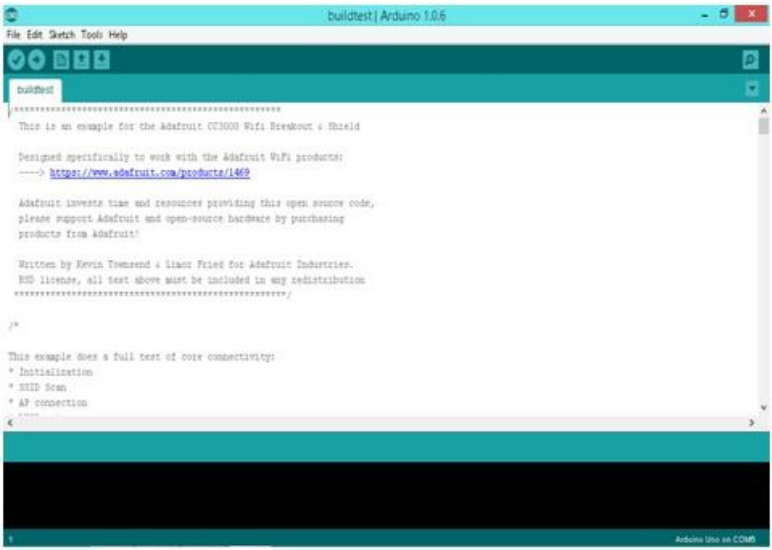
mendukung komunikasi TWI dan SPI. Perangkat lunak Arduino termasuk *Wire Library* digunakan untuk menyederhanakan penggunaan bus TWI. Untuk komunikasi SPI, menggunakan SPI *library*.

IDE (*Integrated Development Environment*) Arduino merupakan aplikasi yang mencakup editor, *compiler*, dan *uploader* dapat menggunakan semua seri modul keluarga Arduino, seperti Arduino *Duemilanove*, *Uno*, *Bluetooth*, *Mega*. Kecuali ada beberapa tipe board produksi Arduino yang memakai *microcontroller* di luar seri AVR, seperti mikroprosesor ARM. Saat menulis kode program atau mengkompilasi modul *hardware* Arduino tidak harus tersambung ke PC atau *Notebook*, walaupun saat proses unggahan ke board diperlukan modul *hardware*.

IDE Arduino juga memiliki keterbatasan tidak mendukung fungsi *debugging hardware* maupun *software*. Proses kompilasi IDE Arduino diawali dengan proses pengecekkan kesalahan sintaksis *sketch*, kemudian memanfaatkan pustaka *Processing* dan avr – gcc *sketch* dikompilasi menjadi berkas object, lalu berkas – berkas object digabungkan oleh pustaka Arduino menjadi berkas biner. Berkas biner ini diunggah ke chip *microcontroller* via kabel USB, serial port DB9, atau Serial Bluetooth.

Compiler IDE Arduino juga memanfaatkan pustaka *open source* AVRLibc sebagai standar *de-facto* pustaka referensi dan fungsi register *microcontroller* AVR. Pustaka AVRLibc ini sudah disertakan dalam satu paket program IDE Arduino. Meskipun demikian, kita tidak perlu mendefinisikan *directive#include* dari pustaka AVRLibc pada sketch karena otomatis compiler me-link pustaka AVRLibc tersebut [9].

Ukuran berkas biner HEX hasil kompilasi akan semakin besar jika kode sketch semakin kompleks. Berkas biner memiliki ekstensi .hex berisi data instruksi program yang biasa dipahami oleh *microcontroller* target. Selain itu, port pararel juga bisa dipakai untuk mengunggah *bootloader* ke *microcontroller*. Meskipun demikian, cara ini sudah jarang digunakan karena sekarang hampir tidak ada *mainboard* PC yang masih menyediakan port pararel, dan pada notebook juga sudah tidak menyertakan port pararel.



Gambar 2.9 Tampilan *Sketch* di Arduino IDE

Pada Gambar 2.9 diatas, terlihat *button* (tombol) yang ada di IDE Arduino, *button compile* berfungsi untuk kompilasi *sketch* tanpa unggah ke board bisa dipakai untuk pengecekan kesalahan kode sintaks *sketch*. *Button upload* untuk mengunggah hasil kompilasi *sketch* ke board target. Pesan *error* akan terlihat jika board belum terpasang atau alamat port COM belum terkonfigurasi dengan benar [9].

Berkas Pustaka yang tersimpan di dalam direktori yang sama *sketchbook* akan terlihat dalam *Tab sketchbook*. Berkas pustaka yang tersimpan di direktori / Arduino / *libraries* / tidak ditampilkan pada tab *sketch* meskipun bias diakses oleh *sketch* lain.

Daripada menekan tombol reset sebelum upload, Arduino Mega 2560 didesain dengan cara yang memungkinkan Anda untuk me – reset melalui perangkat lunak yang berjalan pada komputer yang terhubung. Salah satu jalur kontrol hardware (DTR) mengalir dari ATmega 8U2 / 16U2 dan terhubung ke jalur reset dari ATmega 2560 melalui kapasitor 100 *nanofarad*. Bila jalur ini di-set rendah / *low*, jalur *resetdrop* cukup lama untuk me-reset chip. Perangkat lunak Arduino menggunakan kemampuan ini untuk memungkinkan Anda meng-upload kode dengan hanya menekan tombol upload pada perangkat lunak Arduino. Ini berarti bahwa *bootloader* memiliki rentang waktu yang lebih pendek, seperti

menurunkan DTR dapat terkoordinasi (berjalan beriringan) dengan dimulainya upload.

Pengaturan ini juga memiliki implikasi lain. Ketika Arduino Mega 2560 terhubung dengan komputer yang menggunakan sistem operasi Mac OS X atau Linux, papan Arduino akan di-reset setiap kali dihubungkan dengan *software* komputer (melalui USB). Dan setengah detik kemudian atau lebih, *bootloader* berjalan pada papan Mega 2560. Proses reset melalui program ini digunakan untuk mengabaikan data yang cacat (yaitu apapun selain meng-upload kode baru), ia akan memotong dan membuang beberapa *byte* pertama dari data yang dikirim ke papan setelah sambungan dibuka. Jika sebuah sketsa dijalankan pada papan untuk menerima satu kali konfigurasi atau menerima data lain ketika pertama kali dijalankan, pastikan bahwa perangkat lunak diberikan waktu untuk berkomunikasi dengan menunggu satu detik setelah terkoneksi dan sebelum mengirim data.

Mega 2560 memiliki trek jalur yang dapat dipotong untuk menonaktifkan fungsi *auto – reset*. Pad di kedua sisi jalur dapat hubungkan dengan disolder untuk mengaktifkan kembali fungsi *auto – reset* Pad berlabel "RESET-EN". Anda juga dapat menonaktifkan *auto – reset* dengan menghubungkan resistor 110 ohm dari 5V ke jalur reset.

Selain itu Arduino Mega 2560 memiliki *polyfuse* reset yang melindungi port USB komputer anda dari hubungan singkat dan arus lebih. Meskipun pada dasarnya komputer telah memiliki perlindungan internal pada port USB mereka sendiri, sekring memberikan lapisan perlindungan tambahan. Jika arus lebih dari 500 mA dihubungkan ke port USB, sekring secara otomatis akan memutuskan sambungan sampai hubungan singkat atau *overload* dihapus / dibuang.

2.7 Power Supply

Power supply merupakan perangkat keras yang mampu menyuplai tegangan listrik secara langsung dari sumber tegangan listrik ke perangkat yang membutuhkan tegangan listrik. *Power supply* memiliki input dari tegangan yang berarus AC dan mengubahnya manjadi arus DC lalu menyalurkannya ke berbagai perangkat keras yang membutuhkannya. Karena arus DC yang dibutuhkan untuk perangkat keras agar dapat beroperasi, arus DC bisa disebut juga sebagai arus yang searah, sedangkan arus AC merupakan arus yang berlawanan. *Power Supply* merupakan komponen yang sangat penting agar perangkat keras yang digunakan bisa berjalan dengan baik dan optimal. Tegangan

keluaran *power supply* yang dibutuhkan dan digunakan pada perangkat keras biasanya 24 Volt, 12 Volt, 9 Volt, dan 5 Volt.

2.8 Boost Converter (Step – up Converter)

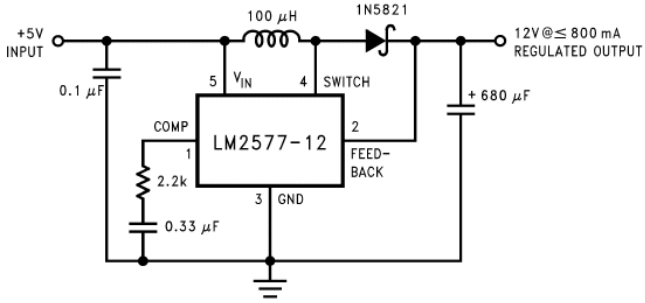
Konverter jenis ini dapat juga disebut sebagai konverter penaik tegangan atau *step – up converter*. Alasan disebut demikian ialah, konverter dari jenis ini mampu untuk menaikkan tegangan masukan. Meskipun konverter jenis ini mampu untuk menaikkan tegangan , namun juga harus mengikuti aturan dari *boost converter* tersebut, yaitu dengan mengatur *Duty Cycle* (D) / siklus kerja.

Gambar 2.10 *Boost Converter* 5V *to* 12V (*Step – up Converter*)

Dari Gambar 2.10 diatas merupakan bentuk sederhana dari *boost converter* atau konverter penaik tegangan. Rangkaian *boost converter* sendiri terdiri dari sumber tegangan DC, induktor, dan *electronic switch* (saklar elektronik), yang dimana pada rangkaian tersebut menggunakan komponen MOSFET, diode, serta kapasitor dan resistor yang juga bertindak sebagai beban (*load*). Untuk mengaktifkan *electronic switch* MOSFET, pada dasarnya ialah dengan menggunakan PWM (*Pulsed Width Modulation*), dimana pengaturan PWM ini sendiri sangat terkait dengan *duty cycle* (D) / siklus kerja. Dan berikut merupakan persamaan dari *duty cycle* (D).

$$\frac{V_0}{V_{dc}} = \frac{1}{1-D}$$

dimana V0 adalah tegangan *output*, Vdc adalah tegangan dc masukan, dan D adalah *duty cycle* atau siklus kerja. Rumus diatas , pada dasarnya dikondisikan bilamana semua parameter dari rangkaian adalah dalam kondisi ideal. Namun, ketika rangkaian tersebut diaplikasikan dengan eksperimen kadang – kadang terdapat suatu perbedaan dengan perhitungan. Perbedaan hasil yang didapat biasanya diakibatkan adanya kerugian energi saat *switching*.Selain itu, adanya *reverse recovery* dari diode yang menyebabkan hilangnya energi.



Gambar 2.11 Skema Rangkaian dari *Boost Converter*

Dari Gambar 2.11 diatas, dapat dijelaskan prinsip kerja dari rangkaian *boost converter* / rangkaian step – up diatas ialah. Siklus I (Saat saklar on) Ketika kondisi dari MOSFET *on* atau nyala, maka siklus tegangan DC / *input* akan mengalir melalui induktor. Selain itu , MOSFET bertindak sebagai konduktor. Sementara itu tidak ada tegangan yang mengalir pada dioda. Siklus II (Saat saklar *off*) Kondisi ini akan mengakibatkan MOSFET dalam kondisi terputus, sehingga tegangan DC yang ada pada induktor akan diteruskan menuju beban (R) dengan melalui dioda. Perlu diketahui bahwa proses *on* dan *off* ini membutuhkan waktu yang sangat cepat sekali, sehingga mendapatkan hasil yang diharapkan.

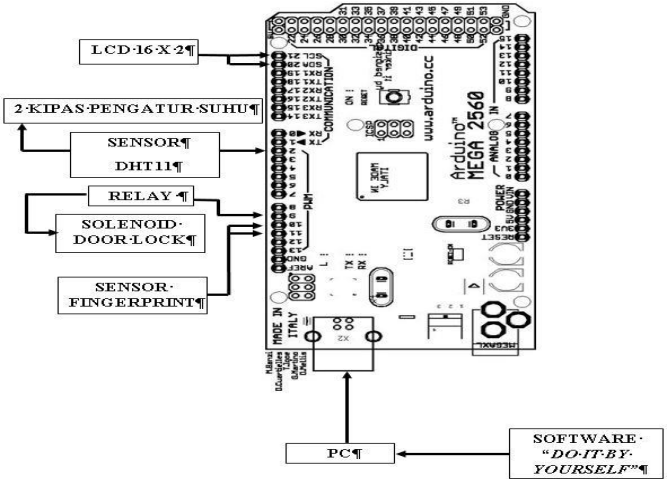
-----Halaman ini sengaja dikosongkan-----

BAB III
PERENCANAAN DAN PEMBUATAN ALAT

Pada bab ini berisi tahapan mengenai tahapan yang dilakukan dalam perencanaan dan pembuatan tugas akhir. Penjelasan diawali dengan blok fungsional sistem secara keseluran yang meliputi proses kerja alat dalam bentuk alur diagram. Perancangan mekanik yang membahas tentang desain dan pembuatan mekanik yang mendukung cara kerja alat. Perancangan elektrik yang membahas perancangan rangkaian elektrik sebagai rangkaian pendukung alat. Serta perancangan perangkat lunak yang meliputi perancangan diagram alur program dan desain *software* "*Do It by Yourself"* menggunakan Microsoft Visual Studio.

3.1 Blok Fungsional Sistem

Sebelum melakukan perancangan perangkat keras dan perangkat lunak, diperlukan sebuah perancangan blok fungsional sistem berupa blok diagram yang menjelaskan sistem kerja secara keseluruhan tugas akhir ini. Secara keseluruhan blok fungsional sistem dapat dilihat pada Gambar 3.1 berikut.



Gambar 3.1 Pinout Arduino Mega 2560

Dari Gambar 3.1 tersebut, maka dapat dijelaskan tentang pemetaan pinout atau *I/O* dari Arduino yang menghubungkan ke masing – masing komponen dari alat maupun sensor serta *software interface* yang digunakan oleh *user* atau orang awam yang menginginkan nilai atau fungsi dari *I/O* Arduino tanpa harus memprogram terlebih dahulu yang tentunya membutuhkan pendalaman pada bahasa pemrograman Arduino tersebut.

Dimana output yang pertama dari output Arduino yaitu sensor *fingerprint*. Cara kerja dari sensor *fingerprint* ini yaitu dengan "merekam" atau mengenali dengan cara membaca pola sidik jari dari *user* yang kemudian akan disimpan oleh Arduino. Lalu selanjutnya proses identifikasi tersebut bisa dilakukan dengan memverifikasi atau mencocokkan data atau pola sidik jari dari *user* tersebut yang sebelumnya telah discan untuk disimpan, dan jika data dinyatakan sama, maka akses akan otomatis terbuka. Kemudian dari proses pengenalan oleh *fingerprint*, maka dilanjutkan ke sensor solenoid *door lock.* Yang dimana sensor ini memiliki cara kerja dimana saat solenoid NC *(Normally Close)* diberi tegangan, maka solenoid akan memanjang atau tertutup. Dan untuk solenoid NO *(Normally Open)* memiliki cara kerja kebalikan dari solenoid NC. Begitu juga dari *output* Arduino atau menerima data dari sensor *fingerprint*. Jika data pengenalan *user* sama dengan inputan dari memori sebelumnya atau pengenalan yang dilakukan senelumnya, maka solenoid akan memendek sehingga *prototype* bisa dibuka, begitu juga dengan kebalikannya jika data atau user berbeda maka akses akan ditolak.

Output dari Arduino selanjutnya yaitu digunakan untuk sensor suhu dan kelembaban atau sensor DHT11. Dimana cara kerja atau fungsi dari sensor DHT11 yaitu mengontrol dan menjaga kestabilan suhu dan kelembaban yang dibutuhkan serta mengatur sirkulasi dan kebersihan udara di dalam *prototype* rancangan alat tugas akhir ini. Karena masing – masing komponen maupun isi di dalam *prototype* alat perlu dijaga suhu dan kelembabannya karena rata – rata material atau bahan dari komponen terbuat dari logam yang tentunya memiliki sifat mudah korosif atau berkarat. Setelah data nilai atau variabel dari suhu dan kelembaban yang dibutuhkan dari dalam "ruang" *prototype* tersebut disimpan yang memiliki suhu dan kelembaban minimal maupun maksimal. Jika pada saat nilai suhu di dalam *prototype* tersebut melebihi maupun mendekati nilai maksimum atau pada saat suhu di dalam *prototype* tersebut dingin, maka secara otomatis akan menyalakan dua

kipas yang ditempatkan diantara dalam ruang dari *prototype* serta luar ruang dari *prototype*, sehingga bisa menjaga sirkulasi udara demi meminimalisir proses korosif atau berkarat karena adanya nilai kelembaban yang tinggi.

Selanjutnya, dari 3 data diatas akan ditampilkan di tampilan LCD 16x2 yang ditempatkan di bagian depan dari *prototype* rancangan alat tugas akhir ini. Dimana untuk tampilan dari sensor *fingerprint* yang menampilkan data dari *user* yang sebelumnya telah disimpan atau dikenali dan akses untuk bisa membuka dan akses *error* atau ketika akses ditolak karena data tidak sesuai dengan pengenalan dari *user*.

Selain itu juga LCD 16x2 diprogram untuk menampilkan data dari sensor DHT11 yang memonitoring suhu maupun kelembaban didalam *prototype* rancangan tugas akhir.

Dari keseluruhan output Arduino yang telah dijelaskan diatas yang mencakup cara kerja maupun fungsi dari masing – masing komponen, tentunya dibutuhkan program dari Arduino atau dalam bahasa pemrograman Arduino biasa disebut dengan *library*, dimana *library* tersebut disesuaikan dengan *output* dari Arduino yang berupa sensor maupun komponen – komponen tertentu.

Dalam *prototype* rancangan tugas akhir ini, telah dibuat *database* program dalam bahasa pemrograman Arduino atau dengan kata lain seperti yang dijelaskan tadi yaitu *library*. Tentunya untuk melakukan pemrograman ke dalam Arduino membutuhkan pendalaman terlebih dahulu terhadap bahasa pemrograman Arduino, dan juga tentunya tidak semua orang bisa melakukan pemrograman dari *hardware* Arduino ini.

Maka dari itu, dalam rancangan tugas akhir ini dibuatlah suatu *software* dengan menggunakan bahasa pemrograman Visual Basic yang dimana dirancang untuk mempermudah penggunaan rancangan tugas akhir ini kepada para pengguna yang rata – rata orang awam yang belum tahu seluk beluk dari bahasa pemrograman Arduino yang ingin menentukan sendiri secara bebas atau sesuai keinginan fungsi dari *I/O* dari Arduino yang telah dirancang dalam rancangan *prototype* alat tugas akhir ini atau biasa disebut dengan *"Do It by Yourself"*.

3.2 Kelebihan dan Kekurangan Komponen

Pada sub bab ini akan dibahas mengenai Kelebihan dan Kekurangan pada spesifikasi masing – masing komponen yang dipakai untuk *prototype* rancangan alat tugas akhir ini.

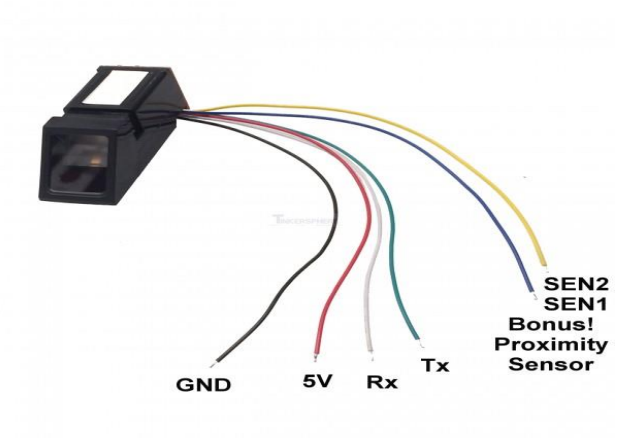
- Sensor *Fingerprint Optical* R305

Sensor *fingerprint* memiliki jenis – jenis tersendiri yang mempunya spesifikasi masing – masing dan tentunya memiliki kelebihan dan kekurangan tersendiri. Didalam *prototype* rancangan alat tugas akhir ini memakai sensor *fingerprint optical* R305 yang memiliki kelebihan sebagai berikut.

1. Pembaca sidik jari biometrik optik dengan fitur lebih canggih dan bisa disematkan ke dalam berbagai produk akhir, seperti: kontrol akses, kehadiran, brankas, kunci pintu mobil.
2. Pengambilan gambar yang tidak terintegrasi dan chip algoritma bersama-sama, *ALL-in-One.*
3. Pembaca sidik jari bisa melakukan pengembangan sekunder, bisa disematkan ke dalam berbagai produk akhir.
4. Konsumsi daya rendah, biaya rendah, ukuran kecil, performa prima.
5. Teknologi optik profesional, teknik pembuatan modul yang tepat.
6. Kemampuan pengolahan gambar yang bagus, bisa berhasil menangkap gambar hingga resolusi 500 DPI (Dots per Inch).

Dan berikut merupakan kelebihan sensor *fingerprint optical* SM630 yang memiliki spesifikasi maupun fisik yang hampir sama dengan sensor *fingerprint* R305 serta pada Gambar 3.2 merupakan contoh dari sensor *fingerprint* dari tipe SM630 beserta pinoutnya.

1. Memiliki intelektual tersendiri (kemampuan yang dibutuhkan untuk melakukan berbagai aktivitas yang dibutuhkan pada aplikasi alat seperti membaca / merekam pola sidik jari, menyimpan data, mengupload, dll.).
2. Berbagai macam aplikasi sidik jari dengan kualitas berbeda.
3. Algoritma peningkatan yang sangat besar.
4. Penggunaannya lebih fleksibel.
5. Mudah digunakan dan dapat dikembangkan pengaplikasiannya.
6. Konsumsi daya rendah.



Gambar 3.2 Sensor Fingerprint Optical SM630

Dari perbandingan spesifikasi dan kelebihan dari kedua sensor fingerprint tersebut, dapat diambil kesimpulan yaitu sama – sama berjenis sensor *fingerprint optical* dan memiliki konsumsi daya rendah serta pengimplementasian dari kedua sensor tersebut dapat dikembangkan sesuai dengan kebutuhan dari pengguna sensor tersebut, hanya saja sensor fingerprint R305 memiliki harga yang lebih terjangkau dibandingkan dengan sensor *fingerprint* SM630.

- Sensor DHT11

  Dalam *prototype* rancangan alat dibutuhkan sensor yang bisa mengontrol maupun memonitoring suhu dan kelembaban didalam "ruang" *prototype* rancangan alat tersebut. Dan untuk sensor suhu dan kelembaban sendiri memiliki berbegai macam dan perbedaan dari fisik maupun spesifikasinya, tetapi dari cara kerjanya, semua sensor ini memiliki beberapa persamaan. Contoh dari sensor suhu dan kelembaban yaitu seperti DHT11, DHT22, serta sensor suhu dan kelembaban dari tipe SHT seperti SHT10, SHT11, SHT15, SHT21, SHT25, SHT71, dan SHT75. Dan berikut pada Tabel 3.1 merupakan spesifikasi masing – masing sensor dari tipe SHT.

Tabel 3.1 Spesifikasi Masing – masing Sensor dari Tipe Sensor SHT

Humidity Sensor		Packaging	Max. RH tolerance*	Max. T tolerance*	Sensor Output
SHT10		SMD	±4.5%RH	±0.5°C	Digital Sbus
SHT11		SMD	±3%RH	±0.4°C	Digital Sbus
SHT15		SMD	±2%RH	±0.3°C	Digital Sbus
SHT21		DFN	±3%RH	±0.4°C	I^2C, PWM, SDM
SHT25		DFN	±2%RH	±0.3°C	I^2C
SHT71		Pins	±3%RH	±0.4°C	Digital Sbus
SHT75		Pins	±1.8%RH	±0.3°C	Digital Sbus

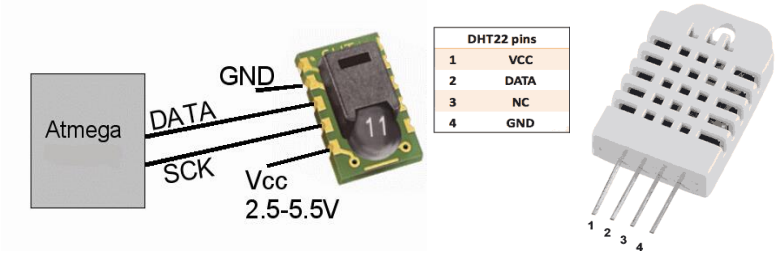
DHT11 memiliki *output* digital yang sudah terkalibrasi. Sensor ini terdiri dari komponen pengukur kelembaban tipe *resistive* dan pengukuran suhu via NTC serta terhubung dengan 8 bit uC sehingga memberikan hasil yang cukup baik, kecepatan respon yang cukup, memiliki ketahanan yang baik terhadap interferensi dan cukup murah dalam harga. *Interface* yang digunakan adalah s*ingle write, serial interface* yang cukup cepat dan mudah. Ukuran sensor yang kecil, kebutuhan komsumsi daya yang rendah dan mampu mentransmisikan outputnya dalam jarak 20 meter.

Kelemahan dari sensor DHT11 ini adalah akurasi yang kurang, selain itu range pengukuran suhunya hanya 0 sd 50 derajat celcius tapi dengan harga yang cukup murah sensor ini bisa menjadi alternatif untuk dipakai dalam pengukuran suhu dan kelembaban yang tidak terlalu memerlukan akurasi yang bagus. Misalnya pengukuran suhu kamar,pengukuran suhu dan kelembaban mesin penetas dan lain-lain.

Catu daya yang diperlukan DHT11 ini berkisar 3.5V sampai 5V. Akses ke sensor hanya diperbolehkan lebih dari 1 detik setelah catu daya pertama kali diberikan. Perlu pula ditambahkan kapasitor 100nF diantara pin VCC dan GND untuk filter catu daya.

Sedangkan untuk sensor DHT22 memiliki range pengukuran yang luas yaitu 0 sampai 100% untuk kelembaban dan -40 derajat celcius sampai 125 derajat celcius untuk suhu. Sensor ini juga memiliki output digital (*single-bus*) dengan akurasi yang tinggi.

Lalu untuk sensor SHT11 memiliki spesifikasi yang lebih baik daripada kedua sensor diatas. Sensor ini dapat mengukur suhu dari -40C hingga +123,8C, atau dari -40F hingga +254,9F dan kelembaban relatif dari 0%RH hingga 1%RH dan memiliki ketetapan (akurasi) pengukuran suhu hingga 0,5C pada suhu 25C dan ketepatan (akurasi) pengukuran kelembaban relatif hingga 3,5%RH, serta membutuhkan catu daya +5V DC dengan konsumsi daya rendah30 µW. Modul ini memiliki faktor bentuk 8 pin DIP 0,6sehingga memudahkan pemasangannya. Pada Gambar 3.3 merupakan contoh dari sensor SHT11 (kiri) dan DHT22 (kanan) beserta pinout yang akan dihubungkan dengan Arduino.



Gambar 3.3 Pinout Sensor SHT11 dan DHT22

Dari beberapa sensor suhu dan kelembaban yang telah dijelaskan mulai dari spesifikasi serta kelebihan dan kekurangan masing – masing. Dari *type* SHT, memang memiliki spesifikasi dan cara kerja lebih mumpuni daripada *type* DHT dan juga dari segi fisik, *type* SHT lebih fleksibel karena lebih kecil sehingga tidak memakan tempat dan lebih efisien. Tetapi, harga yang dipatok dipasaran untuk *type* SHT lebih mahal serta jarang ditemui di indonesia, karena memang *type* SHT sebagian besar diproduksi di luar negeri, sehingga *type* DHT lah yang digunakan sebagai alternatif yang diterapkan pada alat – alat teretentu, karena selain mudah ditemui dipasaran, *type* DHT ini memiliki harga lebih terjangkau dibandingkan dengan *type* SHT meskipun memiliki fisik yang sedikit lebih besar daripada *type* SHT.

- Arduino Mega 2560

Dari jenis – jenis Arduino, Arduino memiliki banyak *type* yang mempunyai sedikit perbedaan antara satu sama lain, mulai dari jumlah pinout untuk I/O (*Input* dan *Output*) Digital dan pinout untuk *I/O* Analog, serta dari segi fisik atau dimensi dan jenis input ke sumber atau koneksi yang digunakan untuk terhubung ke PC.

Jenis – jenis Arduino yang ada sekarang ini yaitu seperti Arduino *Uno*, Arduino *Due*, Arduino *Mega*, Arduino *Leonardo*, Arduino *Fio*, Arduino *Lilypad*, Arduino *Nano*, Arduino *Mini*, Arduino *Micro*, Arduino *Ethernet*, Arduino *Esplora*, dan Arduino *Robot* yang memiliki kelebihan dan kekurangan masing – masing.

Dari sekian banyak jenis – jenis Arduino tersebut, hanya beberapa yang paling sering ditemui teerutama dikalangan mahasiswa atau di indonesia untuk diterapkan pada alat – alat canggih yang membutuhkan proses berulang – ulang dan otomatis seperti robot, *conveyor*, dan sistem – sistem kontrol atau projek – projek tertentu. Seperti dari masing – masing typenya seperti Arduino *Uno* R3, Arduino *Nano* 3, Arduino *Pro Mini*, dan Arduino *Mega* 2560 yang memiliki beberapa persamaan maupun perbedaan atau seperti yang tertera pada Tabel 3.2 berikut ini.

Tabel 3.2 Spesifikasi dari Tipe – tipe Arduino

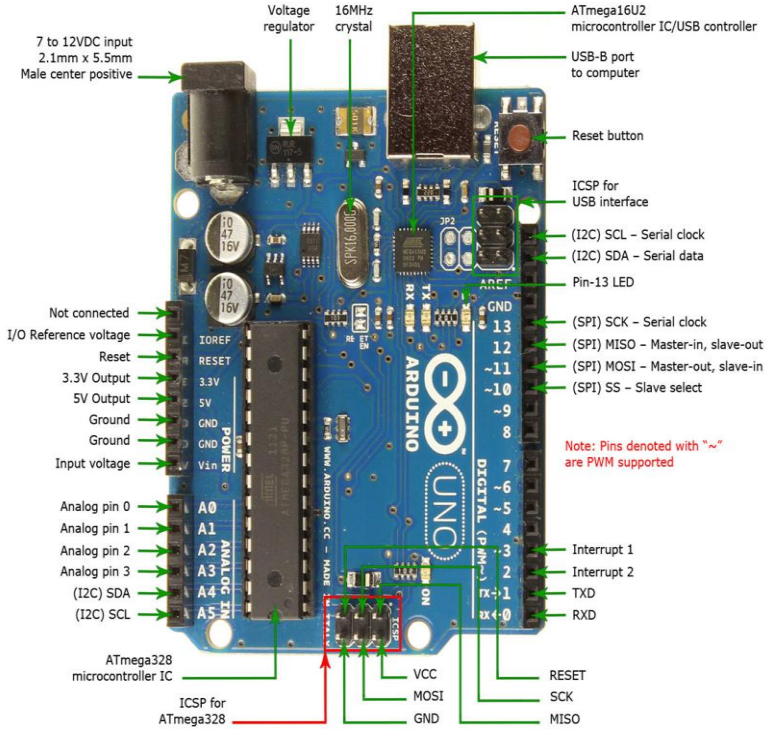
Deskripsi	**Arduino Uno R3**	**Arduino Nano 3**	**Arduino Pro Mini**	**Arduino Mega 2560**
Chip Mikrokontroller	ATMega328P	ATMega328P	ATMega328P	ATMega2560
Tegangan Operasi	5V	5V	5V atau 3.3V tergantung model	
Tegangan *Input* (Rekomendasi, via Jack-DC)	7V – 12V	-	-	7V – 12V
Tegangan *Input* (*Limit*, via Jack-DC)	6V – 20V	-	-	6V – 20V
Digital I/O	14 Buah, 6	14 Buah, 6	14 Buah, 6	54 Buah,

Pin	diantaranya PWM	diantaranya PWM	diantaranya PWM	6 diantaranya PWM
Analog *Input* Pin	6 Buah	6 Buah	6 Buah	16 Buah
Arus DC per Pin *I/O*	40 mA	20 mA	40 mA	20 mA
Arus DC pin 3.3V	50 mA	-	-	50 mA
Memori *Flash*	32 KB, 0.5 KB digunakan *bootloader*	32 KB, 0.5 KB digunakan *bootloader*	32 KB, 0.5 KB digunakan *bootloader*	256 KB, 8KB digunakan *bootloader*
SRAM	2 KB	2 KB	2 KB	8 KB
EEPROM	1 KB	1 KB	1 KB	4 KB
Clock Speed	16 Mhz	16 Mhz	16 Mhz	16 Mhz
Dimensi	68.6 mm x 53.4 mm	45 mm x 18 mm	33 mm x 18mm	101.5 mm x 53.4 mm
Berat	25 g	5 g	5 g	37 g

Jika memperhatikan tabel perbandingan dari ke empat Arduino tersebut, maka bisa diketahui bahwa secara spesifikasi teknis tidak terlalu banyak perbedaan antara Arduino *Uno* R3, Arduino *Nano*, dan Arduino *Pro Mini*. Ketiga nya memiliki chip *micocontroller* yang sama (ATmega328), jumlah pin yang sama, *clock speed* yang sama, dan jumlah memori yang sama. Jadi dari sudut pandang "tenaga", ketiga board tersebut boleh di bilang sama saja. Contoh dari Arduino *Uno* R3 beserta pinoutnya dapat dilihat pada Gambar 3.4, serta Arduino *Pro Mini* beserta pinoutnya pada Gambar 3.5, dan Arduino *Nano* beserta pinoutnya pada Gambar 3.6.

Jadi apakah yang menjadi perbedaan antara Arduino *Uno* R3, *Nano*, dan *Pro Mini* jika secara *utility* ketiganya sama? Perbedaan utamanya terletak pada ketersediaan koneksi USB dan ketersediaan Jack-DC. Arduino *Uno* dan Arduino *Nano* sama – sama memiliki koneksi USB, sementara Arduino *Pro Mini* tidak dilengkapi chip untuk koneksi USB ke komputer, sehingga harus menggunakan board FTDI atau USB to Serial. Arduino *Uno* memiliki Jack-DC, sedangkan untuk Arduino *Nano* tidak.

Perbedaan lain dari ketiga board tersebut adalah kemudahan untuk koneksi kabel *jumper*. Hanya Arduino *Uno* yang memudahkan penggunaannya untuk dapat langsung mencobanya dalam melakukan eksperimen mikrokontroller, hanya menggunakan kabel *jumper* ke *female* terminal yang tersedia.

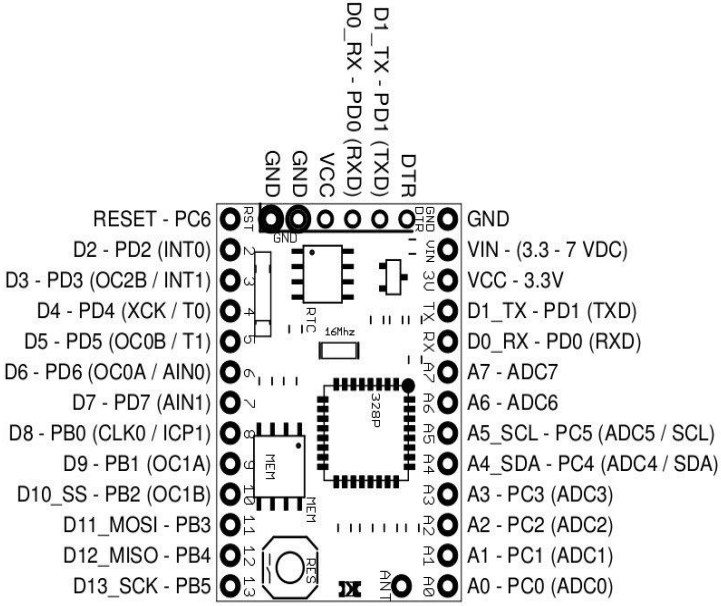


Gambar 3.4 Pinout Arduino *Uno* R3

Hanya Arduino *Uno* juga dari ketiga board tersebut yang memungkinkan untuk menggunakan *shield*. Arduino *Nano* dan *Pro Mini* tidak bisa menggunakan *shield* secara langsung. Untuk mulai mencoba Arduino *Nano* dan *Pro Mini*, pengguna minimal seharusnya telah memiliki breadboard dan kabel *jumper*.

Lalu bagaimana dengan Arduino Mega 2560?. Jadi board yang satu ini memiliki semua kelebihan dari Arduino. *Dev Board* Arduino Mega 2560 memiliki jumlah pin yang paling banyak dari

kesemua board arduino, memiliki seluruh fasilitas yang ada, dan memiliki memori yang paling besar

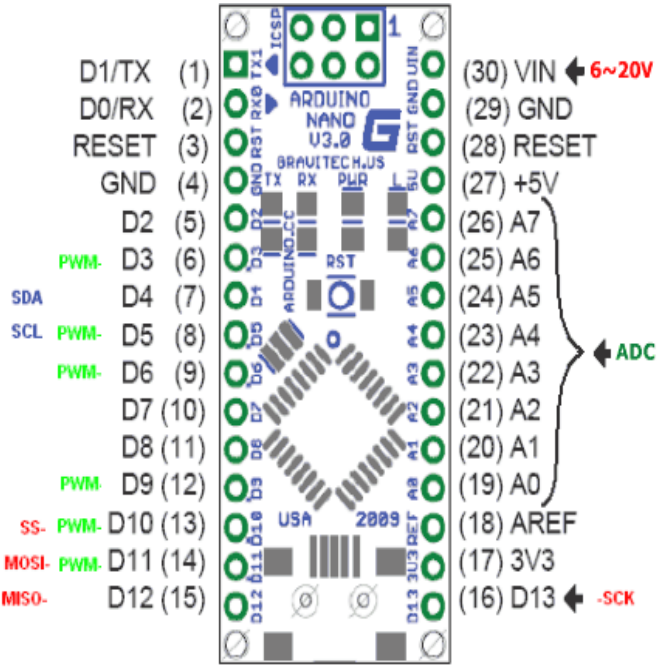


Gambar 3.5 Pinout Arduino *Pro Mini*

Atau bisa dibilang Arduino Mega 2560 adalah yang paling lengkap.

Bahan pertimbangan selanjutnya dalam pemilihan Arduino yaitu dari segi ukuran atau dimensi dan kebutuhan pengguna untuk projek apa yang akan dikerjakan. Arduino Mega 2560 memang memiliki semua apa yang dibutuhkan oleh penggunanya, tetapi apakah harus membutuhkan sebanyak 54 pin *I/O* untuk kebutuhan eksperimen atau projek yang akan dikerjakan, mengingat harga Arduino Mega 2560 adalah yang paling mahal diantara ke – empat board Arduino. Dengan pin sebanyak itu, Arduino Mega memiliki ukuran yang paling besar dari semua board. Ukuran board juga dapat menjadi penentu sebesar apa ukuran box eksperimen atau projek yang akan dikerjakan oleh sipengguna Arduino Mega 2560 tersebut. Jika seseorang sedang berencana untuk membuat sebuah project elektronika yang berukuran kecil, tentunya Arduino *Nano*

atau *Pro Mini* adalah pilihan yang tepat dibanding Arduino Mega dan *Uno*.



Gambar 3.6 Pinout Arduino Nano

Dari penjelasan tersebut, maka dapat ditarik kesimpulan yaitu Arduino Mega 2560 lah yang bisa dikatakan "sempurna" dari segi fisik maupun spesifikasinya untuk diterapkan dalam *prototype* rancangan alat ini.

- LCD 16x2

  Display LCD merupakan sebuah *liquid crystal* atau perangkat elektronik yang dapat digunakan untuk menampilkan angka atau teks. Ada dua jenis utama layar LCD yang dapat menampilkan numerik (digunakan dalam jam tangan, kalkulator, dll) dan menampilkan teks *alfanumerik* (sering digunakan pada mesin fotokopi dan telepon genggam).

  Dalam menampilkan numerik ini, kristal yang dibentuk menjadi bar, dan dalam menampilkan *alfanumerik* kristal hanya diatur kedalam pola titik. Setiap kristal memiliki sambungan listrik

individu sehingga dapat dikontrol secara independen. Ketika kristal *off* (yakni tidak ada arus yang melalui kristal), maka cahaya kristal terlihat sama dengan bahan latar belakangnya, sehingga kristal tidak dapat terlihat. Namun ketika arus listrik melewati kristal, itu akan merubah bentuk dan menyerap lebih banyak cahaya. Hal ini membuat kristal terlihat lebih gelap dari penglihatan mata manusia, sehingga bentuk titik atau bar dapat dilihat dari perbedaan latar belakang.

Sangat penting untuk menyadari perbedaan antara layar LCD dan layar LED. Sebuah LED *display* (sering digunakan dalam radio dan jam) yang terdiri dari sejumlah LED yang benar – benar mengeluarkan cahaya (dan dapat dilihat dalam gelap). Sebuah layar LCD hanya mencerminkan cahaya, sehingga tidak dapat dilihat dalam gelap.

Fungsi dari *display* sendiri dalam suatu aplikasi *microcontroller* sangat penting sekali, yang diantaranya untuk :

1. Memastikan data atau program yeng telah diinputkan sudah valid (benar) atau tidak.
2. Mengetahui hasil dari suatu proses atau projek.
3. Memonitoring suatu proses atau sistem.
4. Mendebug program.
5. Menampilkan pesan yang diinginkan atau diprogram sebagai hasil *interface* dengan alat atau komponen lain.
6. Dan juga hal lainnya yang memiliki proses mekanikal atau elektrikal, terutama elektronika.

Dan jenis atau type dari LCD yang digunakan untuk projek atau aplikasi *microcontroller* tersedia berbagai ukuran seperti 2x8, 2x20, 4x20, dan 4x40 dan semuanya memiliki spesifikasi yang hampir sama, serta kelebihan dan kekurangan satu sama lain

Dari berbagai macam *type* LCD tersebut, hanya terdapat sedikit perbedaan antara satu sama lain, seperti jumlah karakter yang akan ditampilkan. Contohnya LCD 16x2 yang akan menampilkan karakternya dengan dimensi 16 untuk kolomnya dan 2 untuk barisnya yang dimana karakter – karakter yang akan ditampilkan bisa disesuaikan dengan program yang diinputkan atau seperti pada Tabel 3.3 berikut ini.

Tabel 3.3 Karakter pada LCD 16x2

Lower 4 Bits \ Upper 4 Bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)			0	@	P	`	p				ー	タ	ミ	α	p
xxxx0001	(2)		!	1	A	Q	a	q			。	ア	チ	ム	ä	q
xxxx0010	(3)		"	2	B	R	b	r			「	イ	ツ	メ	ß	θ
xxxx0011	(4)		#	3	C	S	c	s			」	ウ	テ	モ	ε	∞
xxxx0100	(5)		$	4	D	T	d	t			、	エ	ト	ヤ	μ	Ω
xxxx0101	(6)		%	5	E	U	e	u			・	オ	ナ	ユ	σ	Ü
xxxx0110	(7)		&	6	F	V	f	v			ヲ	カ	ニ	ヨ	ρ	Σ
xxxx0111	(8)		'	7	G	W	g	w			ァ	キ	ヌ	ラ	g	π
xxxx1000	(1)		(	8	H	X	h	x			ィ	ク	ネ	リ	√	x̄
xxxx1001	(2)		)	9	I	Y	i	y			ゥ	ケ	ノ	ル	⁻¹	y
xxxx1010	(3)		*	:	J	Z	j	z			ェ	コ	ハ	レ	j	千
xxxx1011	(4)		+	;	K	[	k	{			ォ	サ	ヒ	ロ	ˣ	万
xxxx1100	(5)		,	<	L	¥	l	|			ャ	シ	フ	ワ	¢	円
xxxx1101	(6)		−	=	M	]	m	}			ュ	ス	ヘ	ン	£	÷
xxxx1110	(7)		.	>	N	^	n	→			ョ	セ	ホ	゛	ñ	
xxxx1111	(8)		/	?	O	_	o	←			ッ	ソ	マ	°	ö	

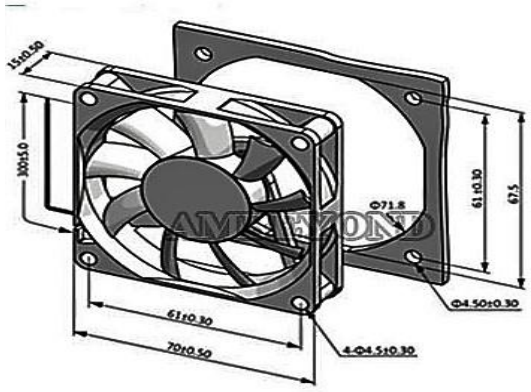
Jadi sebenarnya semua *type* LCD memiliki spesifikasi yang sebagian hampir sama, hanya saja dimensi atau jumlah keseluruhan karakter yang akan ditampilkan, mulai dari kolom dan baris. Dengan itu, pada *prototype* rancangan alat ini, memakai LCD berukuran 16x2 karena selain harga yang dipatok lebih terjangkau, spesifikasinya juga cukup mumpuni yang sesuai dengan kebutuhan untuk menampilkan data dari *input* dan *output* pada *prototype*.

- Kipas Motor DC 5V

  Sebenarnya semua Kipas Motor DC 5V memiliki spesifikasi, fungsi, maupun cara kerja yang hampir sama. Hanya saja yang

membedakan dari semua tipenya yaitu dimensinya atau ukuran keseluruhan dari wadah dari kipas dan juga kipas itu sendiri. *Type* atau dimensi dari kipas motor dc 5V yaitu 20x20x6mm, 25x25x10mm, 40x40x10mm, 50x50x10mm, 60x60x25mm, 70x70x15mm, dan 80x80x25mm.

Motor DC sendiri merupakan motor listrik yang memerlukan suplai tegangan arus searah pada kumparan medan untuk diubah menjadi energi gerak mekanik. Sedangkan pada *prototype* rancangan alat ini, kipas motor dc 5Vhanya digunakan untuk menjaga sirkulasi udara didalam "ruang" *prototype* tersebut, dengan cara mengganti atau membuang udara kotor didalam "ruang" *prototype* yang tentunya mengurangi nilai kelembaban yang dapat menyebabkan korosi dengan udara bersih dari luar. Maka dari itu dibutuhkan kipas motor dc yang berukuran besar atau bisa dibilang cukup agar sesuai dengan kondisi atau kebutuhan dari *prototype* rancangan alat dan juga memiliki sumber 5V yang dimana bisa terhubung ke rangkaian Arduino, dan dimensi kipas yang sesuai yaitu 70x70x15mm, contohnya seperti pada Gambar 3.7 dibawah ini.



Gambar 3.7 Gambar Visual Kipas Motor DC 5V

- Relay DC 5V

  Relay adalah Saklar (*Switch*) yang dioperasikan secara listrik dan merupakan komponen *Electromechanical* (Elektromekanikal) yang terdiri dari 2 bagian utama yakni Elektromagnet (*Coil*) dan Mekanikal (seperangkat Kontak Saklar / *Switch*). Relay

menggunakan Prinsip Elektromagnetik untuk menggerakkan Kontak Saklar sehingga dengan arus listrik yang kecil (*low power*) dapat menghantarkan listrik yang bertegangan lebih tinggi.

Seperti saklar, relay juga dibedakan berdasar *pole* dan *throw* yang dimilikinya.

1. *Pole* : banyaknya *contact* yang dimiliki oleh relay
2. *Throw* : banyaknya kondisi (*state*) yang mungkin dimiliki *contact*

Dan berikut ini penggolongan relay berdasar jumlah *pole* dan *throw* :

1. DPST (*Double Pole Single Throw*)
2. SPST (*Single Pole Single Throw*)
3. SPDT (*Single Pole Double Throw*)
4. DPDT (*Double Pole Double Throw*)
5. 3PDT (*Three Pole Double Throw*)
6. 4PDT (*Four Pole Double Throw*)

Jenis – jenis relay :

1. *Timing* relay adalah jenis relay yang khusus. Cara kerjanya ialah sebagai berikut : jka *coil* dari timing relay ON, maka beberapa detik kemudian, baru *contact* relay akan ON atau OFF (sesuai jenis NO / NC *contact*).
2. *Latching* relay ialah jenis relay digunakan untuk *latching* atau mempertahankan kondisi aktif input sekalipun *input* sebenarnya sudah mati. Cara kerjanya ialah sebagai berikut : jika *latch coil* diaktifkan, ia tidak akan bisa dimatikan kecuali *unlatch coil* diaktifkan.

Jadi dari sekian banyak *type* atau jenis relay, yang prinsip kerja dari semua jenis juga hampir sama. Hanya saja dari sekian banyak jenis relay, Relay DC 5V lah yang paling sering dijumpai atau mudah dicari terutama diindonesia, karena pada umunya projek elektronika menggunakan komponen relay 5v sebagai pelengkap rangkaiannya, selain fungsinya cukup membantu, harganya juga cukup terjangkau atau lebih murah dibandingkan kenis relay yang lain.

- Solenoid *Door Lock* 12V

Solenoid memiliki 2 jenis, yaitu solenoid *valve* dan solenoid *door lock*. Solenoid *valve* merupakan solenoid atau sebuah katup yang dioperasikan secara elektromekanik. Katup dikendalikan oleh arus listrik melalui solenoid: dalam kasus katup dua port, aliran

dinyalakan atau dimatikan; Dalam kasus katup tiga port, arus keluar akan dialihkan di antara dua port *output*. Beberapa katup solenoida dapat ditempatkan bersamaan pada bermacam – macam jenis pipa.

Katup solenoid adalah bagian atau part kontrol yang paling sering digunakan dalam bidang perairan atau sejenisnya. Fungsi dari solenoid *valve* ini adalah mematikan, melepaskan, mengatur jumlah debit air yang dibutuhkan, mendistribusikan atau mencampur cairan. Hal ini dapat ditemukan di banyak projek – projek yang berhubungan. Solenoida dapat membantu dengan cara perpindahan yang cepat dan aman, kehandalan yang tinggi, umur pemakaian yang panjang, kompatibilitas media yang baik dari bahan yang digunakan, daya kontrol rendah dan desain yang terpadu.

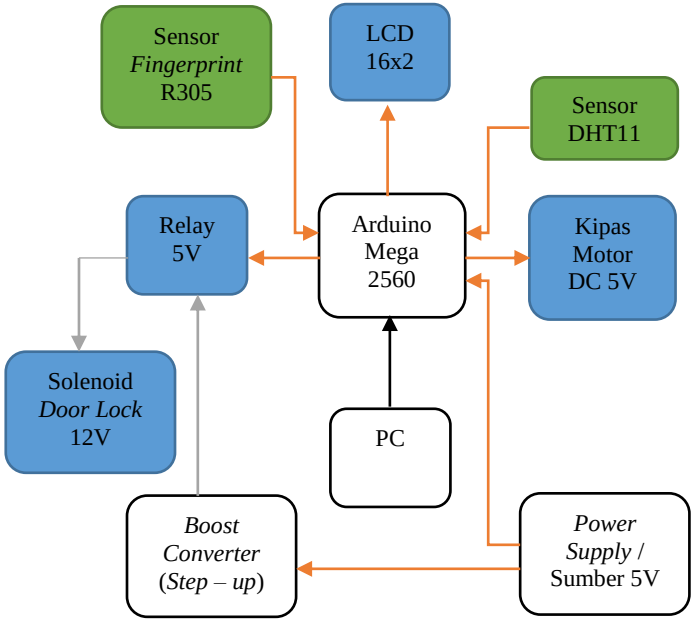
Selain aktuator yang bertipe penyelam (jenis solenoid *valve* yang terdapat didalam air) yang paling sering digunakan, aktuator rotor dan aktuator yang berputar juga digunakan.

Solenoid *door lock*, atau solenoid yang umumnya digunakan sebagai pengaman dalam konstruksi sebuah bangunan (rumah, hotel, gerbang, dan sejenisnya), kendaraan (mobil, motor, dan sejenisnya), ataupun projek – projek yang lain yang memungkinkan pengelolaan akses kontrol secara lebih terpadu, lebih aman, dan lebih efisien.

Dalam prototype ini menggunakan solenoid berjenis *door lock* yang membutuhkan suplai tegangan 12V. Karena selain mudah didapatkan dan memiliki harga terjangkau, solenoid *door lock* hanya ada 2 jenis yaitu solenoid *door lock* 12V dan solenoid *door lock* 24V yang memiliki banyak kesamaan.

3.3 Blok Diagram Arsitektur

Dalam sub bab ini akan dijelaskan alur proses perancangan alat seperti di blok diagram pada Gambar 3.8.



: Garis ini menunjukkan laju tegangan dari sumber 5v

: Garis ini menunjukkan laju tegangan 12v dari sumber setelah proses *step-up*

: Box biru ini merupakan komponen yang digunakan dari output Arduino

: Box hijau ini merupakan komponen atau sensor yang digunakan dari Input Arduino

Gambar 3.8 Blok Diagram Arsitektur Perancangan Alat

Keterangan :

Dari Gambar 3.8 diatas, dapat dijelaskan bagaimana proses rangkaian *electrical* dari prototype rancangan alat ini. Dimana Arduino sebagai "otak" atau pusat kontrol dari keseluruhan *input* dan *output* yang telah disesuaikan dengan kebutuhan *user* alat tersebut. Tetapi disini akan dijelaskan alur dari cara kerja dalam rancangan alat ini.

Di dalam gambar blok diatas, terdapat komponen atau sensor *fingerprint* yang merupakan *input* dari Arduino. Seperti yang dijelaskan pada bab sebelumnya, sensor *fingerprint* memiliki cara kerja "merekam"

identitas seseorang dengan cara menentukan citra atau pola sidik jari dari *user* yang akan memakai suatu alat yang membutuhkan keamanan "khusus" dan dapat dilakukan secara berulang serta memiliki proses otomatis.

Dalam rancangan alat ini, sensor *fingerprint* awalnya akan "merekam" pola sidik jari dari *user* yang akan memakai rancangan alat ini lalu data dari sensor *fingerprint* tersebut akan disimpan dalam memori Arduino.

Setelah proses verifikasi dari identitas tersebut telah selesai, maka proses akan dilanjutkan oleh komponen solenoid *door lock* yang melengkapi sistem keamanan dari rancangan alat ini, dn juga sebagai *output* dari Arduino. Dimana cara kerja daru solenoid *door lock* ini akan memendek atau membuka pengaman dari *prototype* rancangan alat tersebut jika data dinyatakan sama dengan data yang sebelumnya telah disimpan oleh memori Arduino sebelumnya. Begitu juga kebalikannya, ketika proses verifikasi data dinyatakan salah atau tidak sama dengan data yang tersimpan oleh memori Arduino, maka akses akan ditolak dan solenoid masih memanjang atau "pintu" akses untuk membuka *prototype* masih tertutup. Dan didalam gambar blok diatas, solenoid membutuhkan sumber tegangan tersendiri karena spesifikasi dari solenoid tersebut memang membutuhkan sumber 12V serta sebuah komponen elektronik lainnya yaitu Relay agar solenoid tersebut dapat bekerja.

Didalam blok diatas, terdapat komponen atau sensor DHT11 yang merupakan *input* dari Arduino juga. Dimana fungsi dari sensor DHT11 ini yaitu untuk memonitoring serta mengontrol suhu dan kelembaban didalam "ruang" dari *prototype* rancangan alat ini. Dan untuk komponen sebagai *output* dari Arduino yang melengkapi cara kerja dari sensor DHT11 ini yaitu kipas motor DC 5v yang akan membantu dalam menjaga kestabilan sirkulasi udara didalam *prototype* rancangan alat ini. Dimana data yang ditentukan untuk sensor DHT11 ini, jika suhu didalam "ruang" *prototype* tersebut rendah atau dingin tentu saja memiliki nilai kelembaban yang tinggi, dan secara otomatis kipas motor ini akan menyala untuk menjaga kestabilan suhu dan kelembaban udara didalam "ruang" tersebut. Dan jika saat suhu terlalu panas atau lebih tinggi dari yang telah ditentukan oleh kebutuhan *user* maka kipas juga akan menyala sampai suhu maupun nilai kelembaban hingga normal kembali atau stabil agar tidak mengganggu kinerja dari komponen –

komponen atau alat yang terdapat didalam *prototype* yang merupakan pengaplikasian dari BTS box tersebut.

Dari kedua sensor atau *input* Arduino tersebut, data akan ditampilkan di layar LCD 16x2. Yang dimana LCD ini akan menampilkan seperti monitoring suhu dan nilai kelembaban dari sensor DHT11 dan monitoring pengenalan atau akses verifikasi dari sensor *fingerprint*.

Dari keseluruhan proses *electrical* dari prototype rancangan alat ini bisa diubah – ubah komponen atau sensor untuk *Input* dan *Output*-nya sesuai dengan kebutuhan atau keinginan dari *user*. Hanya dalam rancangan alat tugas akhir ini saja menggunakan komponen atau sensor DHT11 dan sensor *fingerprint* yang digunakan sebagai *Input* untuk Arduino.

3.4 Pembagian Tugas Pekerjaan Kelompok

Dalam mengerjakan Tugas Akhir ini pembagian tugas untuk masing – masing orang per kelompok dibagi menjadi beberapa rincian seperti yang akan kami jelaskan seperti di alur pengerjaan pada Gambar 1.1, dimana proses pengerjaan dibagi menjadi 2 blok (sesuai dengan gambar blok diagram pada Bab I) yaitu blok "A", blok "B", dan Blok "C". Dimana blok pertama atau blok "A" akan dibagi tugaskan kepada Rendra Kurnia .R sebagai mahasiswa Pertama. Pada area blok "A" yang diartikan pada awalnya membuat *prototype* rancangan alat tugas akhir dan *database* untuk keseluruhan program dari input dan output yang akan digunakan, lalu langkah selanjutnya pembagian program yang didapatkan dari *database* yang disesuaikan dengan kebutuhan dari user. Setelah itu dibagi menjadi dua langkah yaitu pengukuran data awal dari program yang telah diambil dari *database* program untuk *input* dan *output* yang dibutuhkan oleh *user* untuk diproses dan mengevaluasi hasil dari data yang telah diambil agar program menjadi cepat dan tepat. Dan langkah lainnya yaitu aplikasi *software* dari beberapa program yang diambil dari *database* dapat digunakan sesuai *interface* yang dibutuhkan *user*.

Kemudian blok yang kedua atau pada area blok "B" yang diberikan tugas kepada Setyo Budi Utomo sebagai mahasiswa kedua yang membuat Desain kebutuhan platform yang dimaksudkan yaitu desain *hardware*, yang dimana dapat menjalankan sistem atau program agar dapat berjalan sesuai dengan rancangan. Selanjutnya *Platform*

Interface "*Do It by Yourself*" dimana dari hal ini dibutuhkan sistem yang dapat dijalankan sesuai dengan kebutuhan *user* tanpa harus mengetahui secara detail bagaimana alur dari jalannya program yang dibutuhkan.

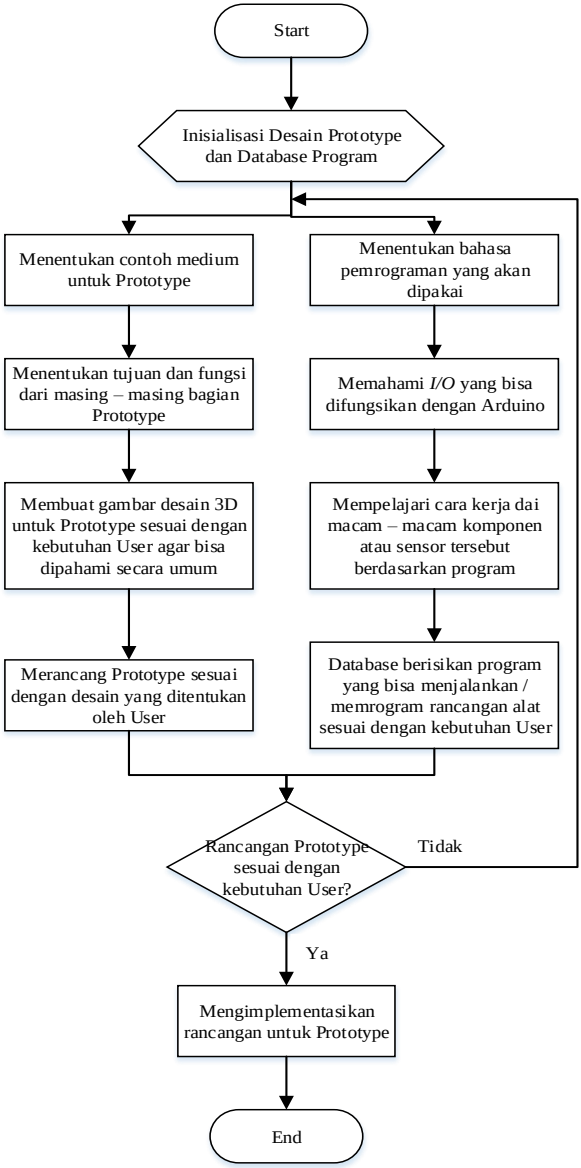
Lalu dari kedua bagian tersebut digabungkan jadi satu dalam area blok "C" akan menjadi sebuah *Generic Programmable I/O*. Dimana pada bagian ini setelah *software"Do It by Yourself"* yang telah di terapkan dalam memprogram *prototype* secara otomatis dengan *input* dan *output* yang di tentukan oleh *user* itu sendiri. Selanjutnya ada validasi *prototype* rancangan alat tugas akhir yang dimana hasil evaluasi dari proses sebelumnya agar rancangan alat lebih sempurna dan mempunyai data yang tepat sesuai dengan kebutuhan *user* dari rancangan alat. Dan yang terakhir terdapat bagian hasil akhir yang dimana hasil yang sudah tepat dari *platform* maupun program yang dibutuhkan oleh *user* dan sesuai dengan data yang telah dievaluasi terus menerus hingga terverifikasi dengan benar. Untuk lebih jelasnya akan dijelaskan pada sub bab 3.4.1.

3.4.1 Proses Pengerjaan Blok "A"

Proses pengerjaan dimulai dengan membuat desain *Prototype box* pengaman perangkat BTS *(Base Transceiver Station)* dengan mendesainya menggunakan software desain 3 dimensi agar maksud desain yang diinginkan dan dapat dimengerti serta dipahami oleh pembaca.

- Merancang desain *prototype* dan *database* program.

  Pada tahap awal. disiapkan atau membuat desain *prototype* terlebih dahulu yang berupa box pengaman untuk perangkat BTS. Dimana pada *prototype* akan diberikan beberapa komponen maupun sensor / *I/O* yang dibutuhkan oleh *user prototype* tersebut dengan Arduino sebagai pusat kontrolnya. Setelah *I/O* dari *prototype* telah ditentukan, maka dibuatlah *database* program, dalam hal ini, Arduino lah yang digunakan sebagai pusat pengontrolnya, sehingga bahasa pemrograman yang digunakan yaitu bahasa pemrograman dari Arduino itu sendiri. Dari masing – masing *I/O* yang merupakan komponen maupun sensor, beberapa diantaranya telah memiliki "*library*" tersendiri didalam bahasa pemrograman Arduino. Hanya saja dalam "*library*" tersebut hanya terdapat dasar – dasar perintah yang akan mengaktifkan cara kerja dari beberapa *I/O*, atau lebih jelasnya dapat dilihat di *flowchart* pada Gambar 3.9



Gambar 3.9 *Flowchart* Merancang desain *Prototype* & *Database* Program

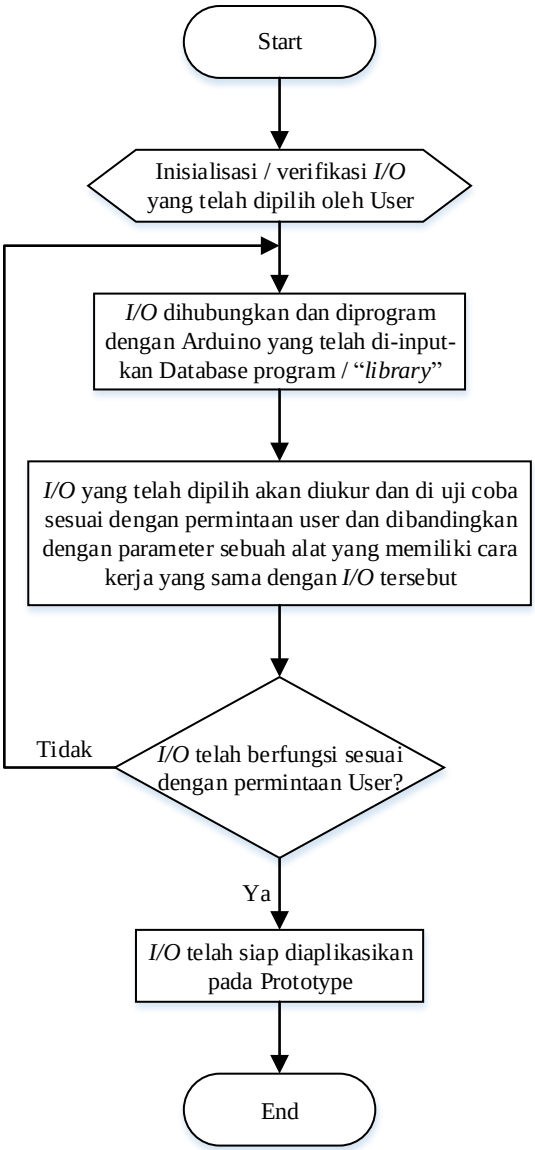
Maka dari itu dibuatlah *database* program yang dimana isi dari *database* tersebut akan mencakup semua "*library*" dari bahasa pemrograman Arduino untuk beberapa jenis *I/O* (yang memang diproduksi atau memiliki pin / kaki yang bisa dihubungkan dengan Arduino atau memiliki sejenis IC yang memiliki *memory* yang bisa menyimpan program dari Arduino), tetapi disini akan lebih ditujukan untuk *I/O* tertentu yang dibutuhkan oleh user saat akan menggunakan *prototype* ini.

- Merancang program untuk *prototype*

  Pada bagian ini, t6elah disiapkan *database* program yang berisi kumpulan data dari bahasa pemrograman Arduino / "*library*" untuk *I/O* yang telah dipilih oleh *user* untuk *prototype*. Dalam kasus ini, *I/O* yang dipilih yaitu sensor *Fingerprint* R305 yang digunakan sebagai pengaman untuk akses *prototype* dan sensor DHT11 sebagai pengontrol atau memonitoring suhu dan kelambaban didalam *prototype* untuk menjaga kestabilan suhu dan kelembaban agar komponen didalam *prototype* dapat bekerja secara maksimal (karena suhu dan kelembaban mempengaruhi cara kerja dari beberapa komponen dan juga telah dijelaskan pada bab sebelumnya). Dan *flowchart* dari program untuk sensor *Fingerprint* R305 terdapat pada Gambar 3.15 serta *flowchart* program untuk DHT11 pada Gambar 3.14.

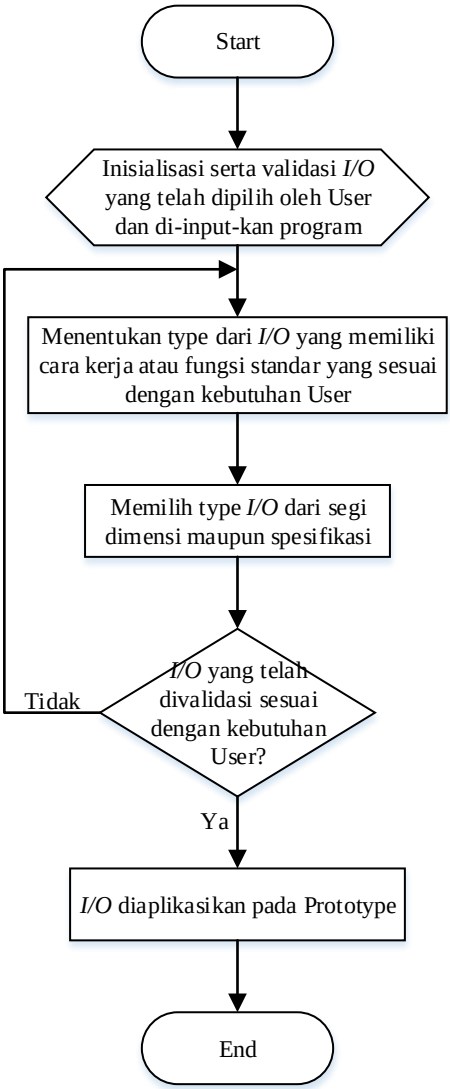
- Pengukuran data

  Dalam bagian ini, *I/O* yang telah dipilih atau ditentukan oleh user akan diukur atau diuji coba agar penghitungan untuk parameter masing – masing *I/O* dapat bekerja secara cepat dan tepat, atau seperti yang tertera pada *flowchart* di Gambar 3.10 berikut ini.



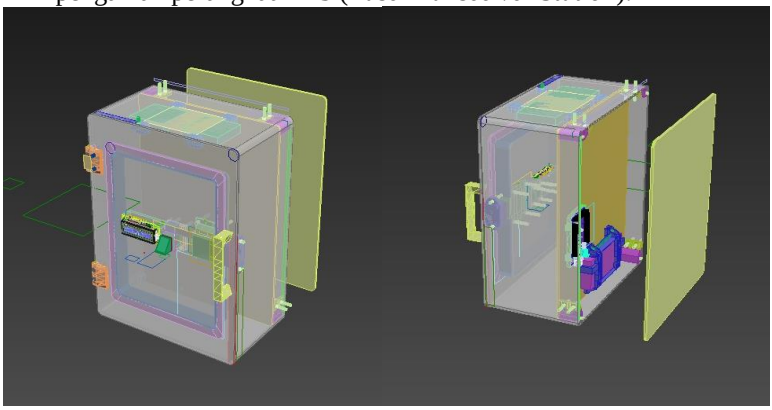
Gambar 3.10 *Flowchart* Pengukuran Data

- Aplikasi program

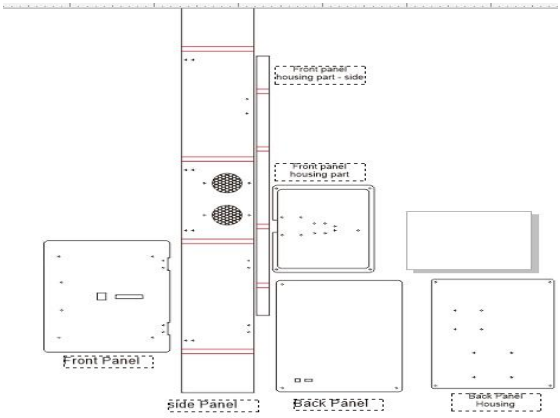


Gambar 3.11 *Flowchart* Aplikasi Program

Untuk bagian ini atau seperti yang tertera pada *flowchart* dari Gambar 3.11, *I/O* yang terdiri dari komponen – komponen atau sensor yang telah dipilih oleh *user* (contoh yang telah dipilih yaitu sensor *fingerprint type* R305 dan DHT11) dan telah disiapkan beberapa bagiannya mulai dari *database* program untuk cara kerja tertentu yang dibutuhkan oleh *user* dan fungsi maupun penempatan atau penyesuaian *I/O* tersebut untuk *prototype* agar bisa berfungsi secara efisien dan praktis serta tidak menggangu komponen eksternal dari bagian *prototype* saat diaplikasikan sebagai *box* pengaman perangkat BTS (*Base Transceiver Station*).



Gambar 3.12 Gambar 3D Visualisasi Desain *Prototype*



Gambar 3.13 Gambar Pola Potong dari Desain *Prototype*

Setelah membuat desain *prototype* secara 3 dimensi seperti pada Gambar 3.12 dan pemetaan atau gambar pola potong dari *prototype* seperti pada Gambar 3.13, langkah selanjutnya yang akan dilakukan yaitu menyiapkan *database* program yang nantinya akan dipakai oleh software "*Do It by Yourself*" dan tentunya cocok dengan *input* maupun *output* yang akan di sediakan, dalam pengartian *input* dan *output* yang di sediakan di batasi karena akan difokuskan penerapannya sesuai dengan konsep utama yaitu pada masalah keamanan serta kontrol suhu dan kelembaban didalam *prototype*. Pada langkah pertama, *list* dari *input* dan *output* yang akan digunakan adalah.

- Sensor *finger*print
- Sensor DHT11
- Arduino Mega 2560
- LCD 16X2
- Kipas motor DC 5V
- Solenoid *door lock*

Setelah melakukan pendataan *I/O* yang akan dipakai atau diimplementasikan ke *prototype*, pada langkah selanjutnya akan disiapkan *database* program untuk Arduino yang disesuaikan untuk setiap *I/O* pada *prototype* atau seperti di bawah ini.

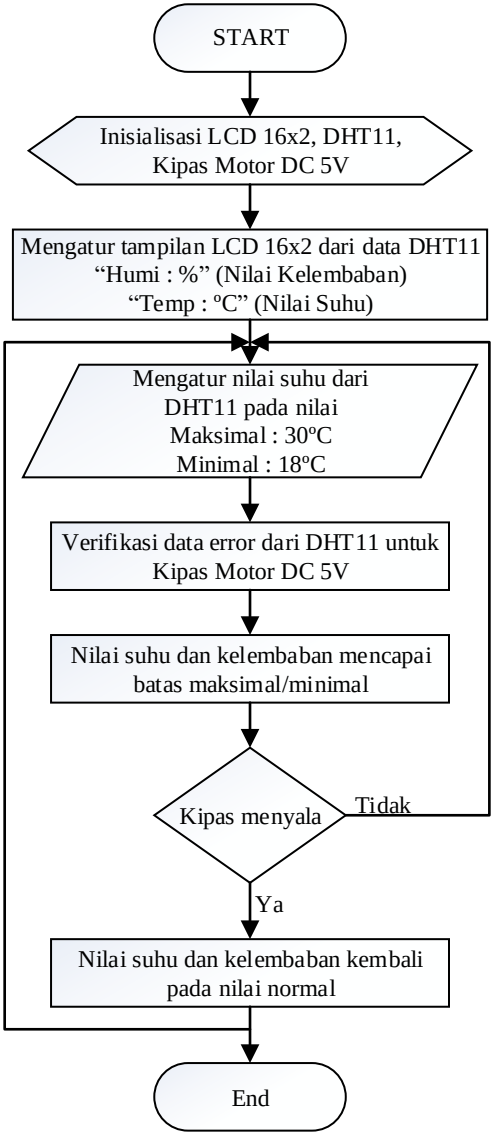
- Program untuk LCD 16x2

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x3f ,2,1,0,4,5,6,7,3, POSITIVE);
        (inisialisasi LCD)
lcd.createChar(0, degree);
  Serial.begin(9600);
  Serial.println("DHTxx test!");
  lcd.begin (16,2); //LCD untuk ukuran 16x2
  dht.begin();
  pinMode (12 ,OUTPUT);          ( pengaturan pin LCD dan
kecerahanya )
lcd.setCursor(0, 0); //baris pertama
  lcd.print("Humi : ");
  lcd.print(h);
  lcd.print(" %");
  lcd.setCursor(0, 1); //baris kedua
  lcd.print("Temp : ");
  lcd.print(t);
```

```
  lcd.print(" C");
  if (h >35 ){
    digitalWrite ( 12, HIGH); (pengaturan penempatan awal
mula karakter yang akan di tampilkan)
```

- Program untuk DHT11

```
#include "DHT.h"(inisialisasi sensor DHT11)
#define DHTPIN 2 (inisialisasi pin yang akan di pakai)
#define DHTTYPE DHT11 (type sensor DHT)
byte degree[8] = {
  0b00110,
  0b00000,
  0b00000,
  0b00000,
  0b00000,
  0b00000,
  0b00000,
  0b00000
};
DHT dht(DHTPIN, DHTTYPE); (pengaturan byte sensor
bekerja)
delay(100);
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  float f = dht.readTemperature(true);(pengaturan dellay
sensor DHT11)
if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println
    return; (pengaturan apabila gagal mengambil data dari
DHT sensor!);
float hif = dht.computeHeatIndex(f, h);
  float hic = dht.computeHeatIndex(t, h, false); (Digunakan
untuk angka desimal (floating point). Memakai 4 byte (32
bit) dari RAM dan mempunyai rentang dari -3.4028235E+38
dan 3.4028235E+38 dan menentukan output dari sensor high
atau low)
Serial.println("fan on");
    digitalWrite (9, LOW);
    delay(2000);
    digitalWrite (9, HIGH);
```



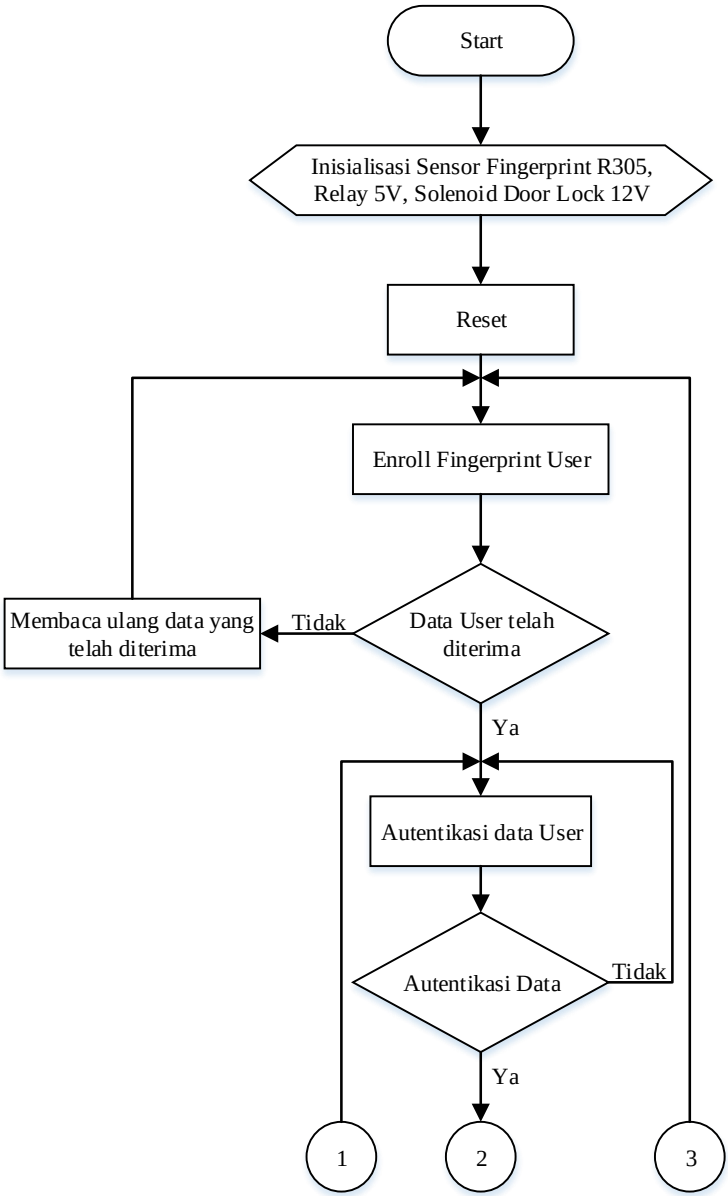
Gambar 3.14 *Flowchart* Program DHT11

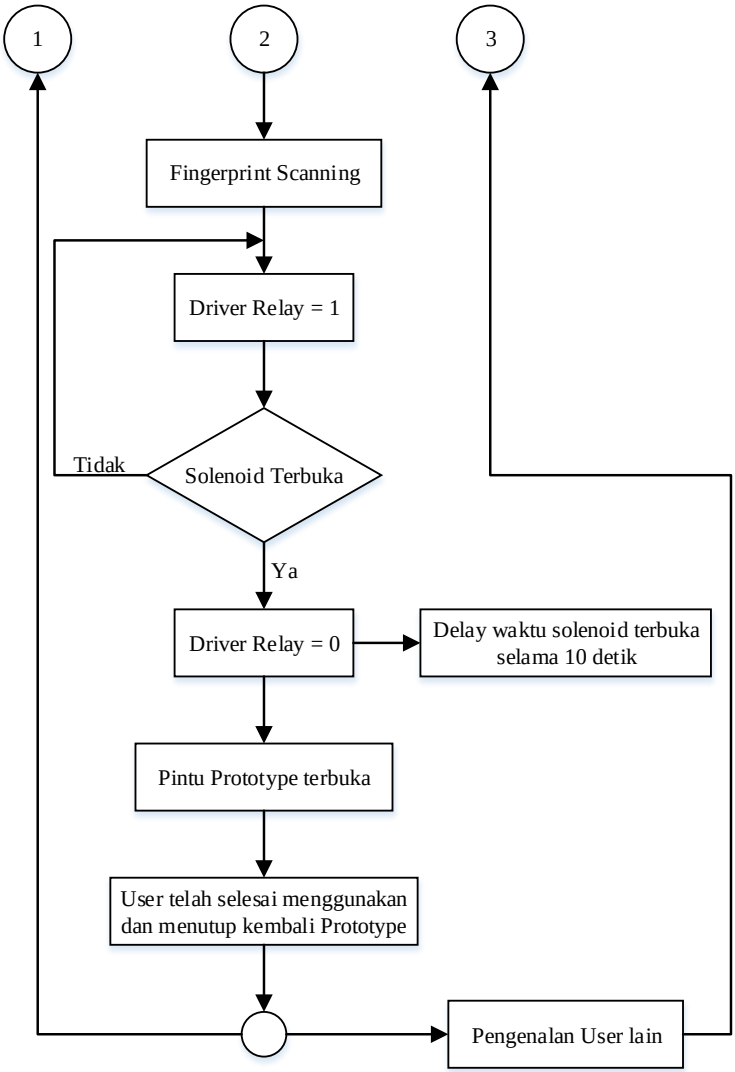
- Program untuk *fingerprint*

```
#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>( inisialisasi sensor fingerprint0

int getFingerprintIDez();

SoftwareSerial mySerial(10, 11);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
(inisialisasi pin yang akan di gunakan)

while (!Serial);  // For Yun/Leo/Micro/Zero/...

  Serial.begin(9600);
  Serial.println("Adafruit finger detect test");
  pinMode (9 ,OUTPUT);
  digitalWrite (9, HIGH); (menentukan menentukan output dari sensor high atau low)

  finger.begin(57600);( mengatur data rate untuk sensor serial port)

  if (finger.verifyPassword()) {
    Serial.println("apabila sensor di detect!");
  } else {
    Serial.println("apabila sensor tidak di detect");
    while (1);
  }
  Serial.println("menunggu sidik jari yang cocok...");

getFingerprintIDez();
  delay(50);            (mengatur kecepatan sensor membaca sidik jari).

int8_t getFingerprintID() { (membaca sidik jari)
  uint8_t p = finger.getImage(); (mengambil data sidik jari)
  switch (p) {
    case FINGERPRINT_OK:
      Serial.println("gambar sidik jari telah di ambil");
```

```
      break;
    case FINGERPRINT_NOFINGER:
      Serial.println(“tidak ada sidik jari yang di baca”);
      return p;
    case FINGERPRINT_PACKETRECIEVEERR:
      Serial.println(“komunikasi error”);
      return p;
    case FINGERPRINT_IMAGEFAIL:
      Serial.println(“gambar sidik jari error”);
      return p;
    default:
      Serial.println(“error yang tidak di ketahui”);
      return p;

/ apabila sukses membaca sidik jari!

  p = finger.image2Tz();
  switch (p) {
    case FINGERPRINT_OK:
      Serial.println("gambar sidik jari di convert");
      break;
    case FINGERPRINT_IMAGEMESS:
      Serial.println(“gambar sidik jari kurang jelas”);
      return p;
    case FINGERPRINT_PACKETRECIEVEERR:
      Serial.println(“komunikasi  error”);
      return p;
    case FINGERPRINT_FEATUREFAIL:
      Serial.println(“tidak dapat menemukan fingerprint
features”);
      return p;
    case FINGERPRINT_INVALIDIMAGE:
      Serial.println(“tidak dapat menemukan fingerprint
features”);
      return p;
    default:
      Serial.println(“error yang tidak di ketahui”);
      return p;
// apabila berhasil di convert!
```

```
  p = finger.fingerFastSearch();
  if (p == FINGERPRINT_OK) {
    Serial.println(“berhasil menenmukan kecocokan!”);
  } else if (p == FINGERPRINT_PACKETRECIEVEERR) {
    Serial.println(“komunikasi error”);
    return p;
  } else if (p == FINGERPRINT_NOTFOUND) {
    Serial.println(“tidak menemukan kecocokan sidik jari”);
    return p;
  } else {
    Serial.println(“error yang tidak di ketahui “);
    return p;

menemukan kecocokan sidik jari!
  Serial.print(“menemukan kecocokan dengan ID data sidik jari yang telah disimpan sebelumnya #”); Serial.print(finger.fingerID);
  Serial.print(“kecocokan data sidik jari 100%”); Serial.println(finger.confidence);
}
Apabila sidik jari masih belom cocok!
// returns -1 if failed, otherwise returns ID #
int getFingerprintIDez() {
  uint8_t p = finger.getImage();
  if (p != FINGERPRINT_OK)  return -1;

  p = finger.image2Tz();
  if (p != FINGERPRINT_OK)  return -1;
  p = finger.fingerFastSearch();
  if (p != FINGERPRINT_OK)  return -1;
  //apabila sidik jari sudah cocok
  Serial.print("menemukan kecocokan dengan ID data sidik jari yang telah di simpan sebelumnya "); Serial.print(finger.fingerID);
  Serial.print(" kecocokan data sidik jari 100% "); Serial.println(finger.confidence);
  return finger.fingerID;
```


Start
Inisialisasi Sensor Fingerprint R305, Relay 5V, Solenoid Door Lock 12V
Reset
Enroll Fingerprint User
Data User telah diterima
Tidak
Membaca ulang data yang telah diterima
Ya
Autentikasi data User
Autentikasi Data
Tidak
Ya
1
2
3



Gambar 3.15 *Flowchart* Program Sensor *Fingerptint*

Setelah perancangan program untuk software langkah yang harus di lakukan selanjutnya adalah pengaplikasian program pada

software yang telah di rancang yaitu salah satu caranya adalah dengan melakukan komunikasi serial dengan langkah – langkah sebagai berikut.
Langkah membuat program sketch komunikasi serial

1. Set *baud rate* misal 9600 dengan fungsi serial.begin(9600) dalam fungsi void setup(). Speed yg tersedia antara lain 300, 1200, 2400, 4800, 9600, 14400, 19200, 28800,38400, 57600, 115200 (liat dokumentasi tiap jenis arduino).
2. Untuk menerima data cek apakah ada data di *Rx Buffer* dengan fungsi serial.avalable()
   - jika data tersedia return value = true
   - jika data kosong return value = false.

   if(serial.available() > 0)
3. Mengambil data dari buffer penerima : serial.read() , return valuenya adalah 1 byte data pertama yg ada di RX Buffer .
   misal byte pertama RX buffer berisi char 2, maka
   - char data = serial.read(); //data = '2' –> karakter
   - byte data = serial.read(); // data = 50 –> numerik
4. Untuk mengirim data bisa menggunakan
   - serial.print (data yang dikirim) : mengirim data asci
   - serial.println (data yang dikirim), ditambahkan kode enter (CR dan LF)
   - serial.write (data yang dikirim) : mengirim data byte

   Contoh serial.print('A') akan mengirim huruf A
   - serial.print(65) akan mengirim 2 byte berisi code Ascii '6' dan '5' (aktual yg dikirim 54 dan 53)
   - serial.write(65) akan mengirim 1 byte 65 (aktual yang dikirim numerik 65)
   - serial.write(data) adalah pengganti syntax Serial.print(data, BYTE) versi sebelumnya. Parameter data pada serial.write adalah angka 1 byte (0-255).

dimana fungsi – fungsi yang tersedia untuk komunikasi Arduino adalah sebagai berikut

- if (Serial) : Untuk mengecek apakah Port sudah siap.
- Serial.available() : Untuk mengecek apakah data sudah ada di buffer penerima.
- Serial.begin() : untuk mengeset kecepatan transmisi data.

- serial.end() : Untuk menon-aktifkan pin rx dan tx sbg fungsi serial dan kembali sebagai pin *I/O*.
- Serial.find() : mencari string dalam buffer data.
- Serial.findUntil() : mencari buffer data sampai data dengan panjang / terminatornya yg diberikan ditemukan.
- Serial.flush() : menunggu data terkirim semua.
- Serial.parseFloat() : mengambil data *float* pertama dari data di buffer serial.
- serial.parseInt() : mengambil data *integer* pertama dari data di buffer serial.
- Serial.peek() : mengambil data berikutnya di bufer penerima.
- Serial.print() : mengirim data ASCII.
- Serial.println() : mengirimdata ASCII + CR,LF (kode enter).
- Serial.read() : membaca data yg diterima.
- Serial.readBytes() : membaca data byte yg diterima.
- Serial.readBytesUntil()
- Serial.setTimeout() : mengeset batas maksimum waktu tunggu (*timeout*) transmisi data.
- Serial.write() : mengirim data byte (numerik).
- Serial.serialEvent() : fungsi ini akan dipanggil jika data validating / diterima.berlaku seperti interupsi serial.

Setelah melakukan hal di atas maka *database* program yang telah disiapkan akan dapat dipanggil oleh *software* "*Do It by Yourself*". Langkah selanjutnya ialah pengukuran data yang akan dibahas lebih lanjut pada bab 4.

3.4.2 Proses Penggabungan Alur Kerja (Blok "C")

Setelah masing – masing blok, baik dari area blok "A" yang ditugaskan kepada mahasiswa pertama dan area blok "B" yang ditugaskan kepada mahasiswa kedua maka proses pengerjaan masing – masing mahasiwa di gabungkan dalam area blok "C". Dalam bagian ini, *database* program yang telah disiapkan berdasarkan *list* kebutuhan dari *platform* untuk *prototype* yang telah di list oleh mahasiswa yang diutgakan untuk area blok "B" akan di terapkan dalam *software* "*Do It by Yourself*" agar pada saat proses validasi program Arduino yang sebelumnya disiapkan untuk masing – masing *input* dan *output* secara manual serta pengaturan pada pinout juga diverifikasi agar dapat di

lakukan secara otomatis sehingga dapat memudahkan *user* yang notabene-nya adalah orang awam agar dengan mudah membuat sebuah rangkaian elektronika berbasis Arduino tanpa adanya kendala dengan bahasa pemrograman Arduino yang cukup rumit. Setelah di jelaskan pada subab 3.4.1 cara penggabungan maka terbentuklah *Generic Programmable* Penghubung Pengguna Dengan *I/O* Dari Arduino Yang Diterapkan Pada *Box* Pengaman Perangkat Bts (*Base Transceiver Station*) .

Selanjutnya proses validasi akan di bahas lebih lanjut di bab 4 dimana proses ini ialah proses pengambilan data saat kedua blok area (Blok “A” dan Blok “B”) sudah digabungkan menjadi satu.

-----Halaman ini sengaja dikosongkan-----

BAB IV
PENGUJIAN DAN ANALISA DATA

Setelah membuat rancangan untuk perencanaan dan pembuatan alat, mulai dari komponen, sensor, program, *hardware*, maupun *I/O* yang lain, yang dihubungkan dengan Arduino sebgai pusat kontrol dari rancangan *prototype*. Maka perlu dilakukan pengujian alat yang meliputi pengujian *hardware* (perangkat keras) dan *software* (perangkat lunak). Pengujian ini dilkaukan pada rancangan alat untuk mengetahui kesesuaian antara teori dengan hasil perancangan, yaitu dengan mengetahui hasil pengukuran pada setiap perangkat yang telah dibuat.

4.1 Pengujian LCD 16x2

LCD 16x2 digunakan untuk menampilkan nilai dari kondisi suhu dan kelembaban didalam rancangan *prototype* yang dihasilkan oleh sensor DHT11 seperti pada Gambar 4.1 dibawah ini.

Gambar 4.1 Nilai Suhu dan Kelembaban yang Ditampilkan dengan LCD 16x2

4.2 Pengukuran dan Pengujian Suhu dan Kelembaban Menggunakan Hygrometer

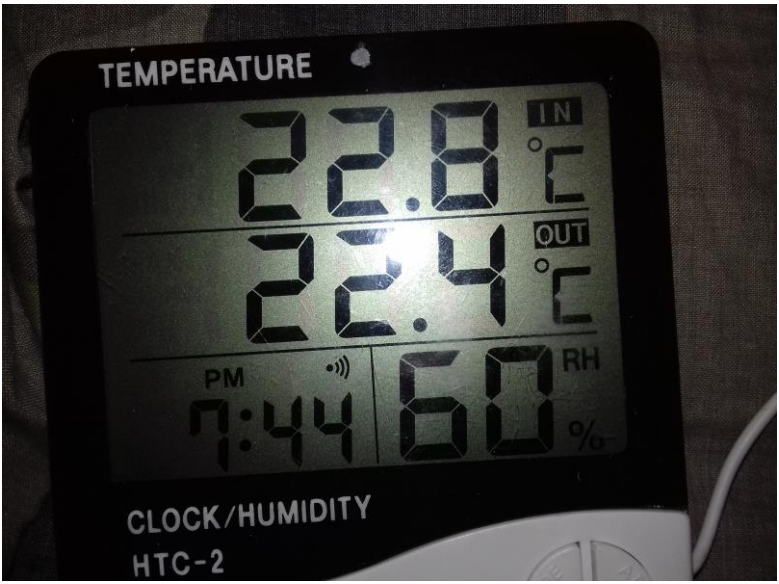
Hygrometer merupakan alat yang digunakan untuk memonitoring secara terus menerus kondisi suhu dan kelembaban disekitarnya atau suatu medium tertentu yang akan diukur besar nilai suhu dan kelembabannya secara akurat. Dalam kasus ini, pengukuran dilakukan pada suatu ruangan yang dapat ditentukan sendiri suhunya dengan parameter nilai suhu minimum sebesar 18ºC dan nilai suhu maksimum sebesar 30ºC yang masing – masing dilakukan sebanyak 3 kali agar mendapatkan nilai yang valid. Secara lengkap data pengujian dapat dilihat pada Tabel 4.1 dibawah ini untuk pengujian suhu dan kelembaban.

Tabel 4.1 Pengujian Nilai Suhu dan Kelembaban dengan *Hygrometer*

Suhu ruangan yang diatur	IN	OUT	RH
18ºC	23.2ºC	22.9ºC	63%
18ºC	23.4ºC	22.9ºC	64%
19ºC	23.6ºC	23.3ºC	61%
19ºC	23.5ºC	23.2ºC	61%
20ºC	23.1ºC	22.9ºC	60%
20ºC	22.9ºC	22.7ºC	61%
21ºC	23.0ºC	22.5ºC	61%
21ºC	23.1ºC	22.5ºC	62%
22ºC	22.8ºC	22.4ºC	60%
22ºC	22.7ºC	22.6ºC	60%
23ºC	22.5ºC	22.0ºC	61%
23ºC	22.5ºC	22.2ºC	62%
24ºC	22.9ºC	23.4ºC	64%
24ºC	23.5ºC	23.2ºC	66%
25ºC	23.7ºC	23.9ºC	69%
25ºC	23.9ºC	24.1ºC	70%
26ºC	24.6ºC	25.9ºC	79%
26ºC	24.5ºC	25.9ºC	75%
27ºC	25.8ºC	27.3ºC	77%
27ºC	25.9ºC	27.5ºC	76%
28ºC	26.5ºC	28.0ºC	82%
28ºC	26.9ºC	28.2ºC	81%
29ºC	28.8ºC	29.4ºC	79%
29ºC	29.2ºC	29.6ºC	85%

30ºC	29.9ºC	31.4ºC	85%
30ºC	30.1ºC	31.6ºC	83%

Setelah mengambil data seperti diatas, dapat dijelaskan yaitu besar nilai suhu dan kelembaban dalam suatu ruangan yang ruangan tersebut bisa diatur kondisi suhunya. Dari nilai suhu dan kelembaban pada *hygrometer* terdapat dua jenis nilai suhu yaitu In dan Out. Nilai dari tabel In diambil dari kondisi suhu disekitar alat *hygrometer* dan nilai dari tabel Out diambil dari nilai suhu suatu medium yang diukur oleh sensor termokopel yang merupakan bagian dari alat *hygrometer*, dan pada Gambar 4.2 dibawah merupakan kondisi dari *hygrometer* yang menampilkan nilai suhu dan kelembaban diruangan ketika ruangan diatur dengan suhu 22 ºC



Gambar 4.2 Nilai Suhu dan Kelembaban yang Diukur oleh *Hygrometer*

Dari Tabel 4.1 diatas, maka dapat diambil nilai Error agar bisa ditentukan nilai atau data yang benar – benar valid, dan sebagai perbandingannya akan mengambil nilai suhu dari In atau kondisi suhu disekitar alat *hygrometer*, atau seperti pada Tabel 4.2 berikut ini.

Tabel 4.2 Pengukuran Nilai Suhu dengan *Hygrometer*

Awal	Akhir	Nilai Suhu ruangan	%Error
23.2ºC	23.4ºC	18ºC	1.2
23.6ºC	23.5ºC	19ºC	0.9
23.1ºC	22.9ºC	20ºC	0.8
23.0ºC	23.1ºC	21ºC	1.1
22.8ºC	22.7ºC	22ºC	0.9
22.5ºC	22.5ºC	23ºC	1
22.9ºC	23.2ºC	24ºC	1.3
23.7ºC	23.9ºC	25ºC	1.2
24.6ºC	24.5ºC	26ºC	0.9
25.8ºC	25.9ºC	27ºC	1.1
26.5ºC	26.9ºC	28ºC	1.4
28.8ºC	29.2ºC	29ºC	1.4
29.9ºC	30.1	30ºC	1.2

Berdasarkan pengukuran nilai suhu didalam ruangan tersebut dengan menggunakan *hygrometer*, maka didapatkan rata – rata %Error sebanyak 1.108% yang artinya alat atau hygrometer tersebut masih sesuai dengan standardnya.

% Error didapatkan dari rumus :

$$\%Error = \frac{(Awal - Akhir) - Suhuruangan}{Suhuruangan} x100\%$$

4.3 Pengukuran dan Pengujian Suhu dan Kelembaban Menggunakan Sensor DHT11

Pengujian nilai suhu dan kelembaban ini dilakukan pada ruang dan medium yang sama dengan alat *hygrometer*. Pengujian dan pengukuran ini dilakukan dengan bertujuan untuk membandingkan besar nilai yang valid atau nilai %Error agar bisa sesuai dengan nilai yang dibutuhkan untuk penyesuain rancangan alat atau *prototype*

Tabel 4.3 Pengujian Nilai Suhu dan Kelembaban dengan Sensor DHT11

Suhu ruangan yang diatur	Suhu / *Temperature*	RH / Kelembaban
18ºC	18ºC	59%

18ºC	18ºC	58%
19ºC	18ºC	61%
19ºC	19ºC	60%
20ºC	19ºC	64%
20ºC	20ºC	63%
21ºC	21ºC	67%
21ºC	20ºC	69%
22ºC	21ºC	70%
22ºC	21ºC	67%
23ºC	22ºC	69%
23ºC	23ºC	71%
24ºC	23ºC	70%
24ºC	23ºC	71%
25ºC	24ºC	73%
25ºC	25ºC	70%
26ºC	26ºC	75%
26ºC	25ºC	74%
27ºC	26ºC	77%
27ºC	26ºC	76%
28ºC	27ºC	79%
28ºC	28ºC	76%
29ºC	28ºC	79%
29ºC	29ºC	80%
30ºC	29ºC	85%
30ºC	30ºC	83%

Dari Tabel 4.3 diatas, telah didapatkan data nilai dari kondisi suhu dan kelembaban pada ruangan yang sama dengan pengambilan data *hygrometer* diambil. Setelah data diatas lengkap, maka dapat diambil nilai %Error untuk mendapatkandata atau nilai yang valid. Dan berikut ini pada Gambar 4.3 merupakan kondisi suhu dan kelembaban diruangan yang diukur dengan DHT11 dan ditampilkan di LCD 16x2.

Gambar 4.3 Nilai Suhu dan Kelembaban yang Diukur oleh DHT11

Tabel 4.4 Pengukuran Nilai Suhu dan Kelembaban dengan Sensor DHT11

Awal	Akhir	Nilai Suhu ruangan	%Error
18ºC	18ºC	18ºC	1
18ºC	19ºC	19ºC	2
19ºC	20ºC	20ºC	2
21ºC	20ºC	21ºC	0
21ºC	21ºC	22ºC	1
22ºC	23ºC	23ºC	2
23ºC	23ºC	24ºC	1
24ºC	25ºC	25ºC	2
26ºC	25ºC	26ºC	0
26ºC	26ºC	27ºC	1
27ºC	28ºC	28ºC	2
28ºC	29ºC	29ºC	2
29ºC	30ºC	30ºC	2

Berdasarkan pengukuran nilai suhu didalam ruangan tersebut atau pada Tabel 4.4 diatas dengan menggunakan sensor DHT11, maka didapatkan rata – rata %Error sebanyak 1.38% yang artinya sensor DHT11 tersebut masih sesuai dengan standardnya.

4.4 Pengujian Sensor Fingerprint Optical R305

Pengujian pada sensor *fingerprint* ini dilakukan untuk melihat syarat atau kondisi dan ketentuan yang berlaku pada kondisi jari si pengguna yang akan mengakses rancangan alat atau *prototype*. Berikut ini merupakan tabel data yang dibutuhkan *fingerprint* untuk kondisi masing – masing jari dari kedua *user*. Sebelumnya akan dijelaskan beberapa parameter untuk pengukuran dari sensor *fingerprint*, yang terdapat 4 kondisi jari yang ditentukan yaitu bersih, kotor, basah,dan luka.

- Bersih

  Disini yang dimaksud dengan arti atau kondisi bersih itu sendiri pada jari yang akan dipakai yaitu di area sekitar *core* atau inti dari pola sidik jari yang berbentuk suatu pusat putar – balikan sebuah garis pada guratan –guratan sidik jari seperti yang tertera pada Gambar 2.3 tidak terdapat benda ataupun hal – hal kecil seperti kotoran, hewan / serangga, dan sesuatu yang besifat cairan yang dapat menyebabkan lengket, berminyak, dan sebagainya. Contoh dari kondisi jari yang bersih terdapat pada Gambar 4.4.

Gambar 4.4 Contoh Kondisi Jari yang Bersih (Ibu Jari)

- Kotor

  Pada bagian ini, yang dimaksud dengan kondisi jari yang kotor yaitu kondisi dimana seluruh area *minutiae* atau di sekitar area *core*, *pore*, dan *delta* seperti yang tertera pada Gambar 2.3 terdapat hal – hal yang menutupi area tersebut seperti serangga / hewan, bercak, atau kotoran lainnya yang bisa menutupi sinar atau cahaya yang dipancarkan oleh sensor *fingerprint* ke jari yang akan digunakan. Contoh dari kondisi jari yang kotor yaitu seperti pada Gambar 4.5.

Gambar 4.5 Contoh Kondisi Jari yang Kotor (Jari Manis)

- Basah

  Dalam bagian ini, yang dimaksudkan dengan kondisi jari yang basah yaitu kondisi dimana jari tersebut berkeringat atau terkena benda cair yang lain yang bersifat berminyak, lengket, berwarna, licin, ataupun benda cair lainnya yang dapat meninggalkan bercak atau dapat merusak pola *ridge*. Contoh untuk kondisi jari yang seperti ini terdapat pada Gambar 4.6.

Gambar 4.6 Contoh Kondisi Jari yang Basah (Jari Kelingking)

- Luka

  Dibagian ini, yang dimaksudkan dengan kondisi jari terluka yaitu pada saat jari tersayat, robek, luka bakar, bernanah, penyakit seperti kutu air dan sebagainya sehingga menyebabkan rusaknya pola ridge atau minutiae pada sidik jari yang akan digunakan. Berikut ini pada Gambar merupakan contoh dari kondisi jari yang terluka.

Gambar 4.7 Contoh Kondisi Jari yang Luka (Jari Telunjuk)

Dan berikut ini merupakan pengujian dan pengukuran sensor *fingerprint* yang dilakukan dengan 4x percobaan seperti pada Tabel 4.5, 4.6, 4.7, 4.8 agar bisa ditentukan nilai %Error dari sensor *fingerprint*.

Tabel 4.5 Pengujian Sensor *Fingerprint* dengan Kondisi Jari dari *User* Pertama pada Percobaan Pertama

User 1	Jari yang Digunakan	Kondisi Jari pada saat *Enroll*				Kondisi Jari pada saat Mengakses				Kondisi Solenoid
		Bersih	Kotor	Basah	Luka	Bersih	Kotor	Basah	Luka	
	Ibu Jari	✓				✓				On
	Jari Telunjuk				✓					Off
	Jari Tengah	✓				✓				On
	Jari Manis		✓							Off
	Jari Kelingking			✓						Off

Tabel 4.6 Pengujian Sensor *Fingerprint* dengan Kondisi Jari dari *User* Pertama pada Percobaan Kedua

User 1	Jari yang Digunakan	Kondisi Jari pada saat *Enroll*				Kondisi Jari pada saat Mengakses				Kondisi Solenoid
		Bersih	Kotor	Basah	Luka	Bersih	Kotor	Basah	Luka	
	Ibu Jari	✓				✓				On
	Jari Telunjuk				✓					Off
	Jari Tengah	✓				✓				On
	Jari Manis		✓							Off
	Jari Kelingking			✓				✓		On

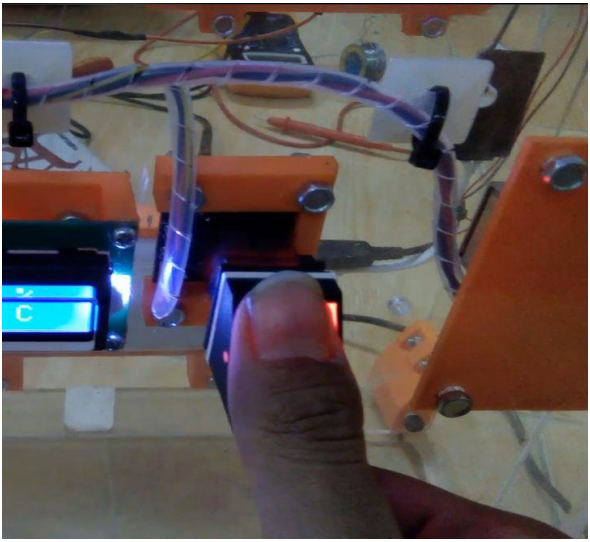
Tabel 4.7 Pengujian Sensor *Fingerprint* dengan Kondisi Jari dari *User* Pertama pada Percobaan Ketiga

User 1	Jari yang Digunakan	Kondisi Jari pada saat *Enroll*				Kondisi Jari pada saat Mengakses				Kondisi Solenoid
		Bersih	Kotor	Basah	Luka	Bersih	Kotor	Basah	Luka	
	Ibu Jari	✓				✓				On
	Jari Telunjuk				✓					Off
	Jari Tengah	✓				✓				On
	Jari Manis		✓							Off
	Jari Kelingking			✓				✓		On

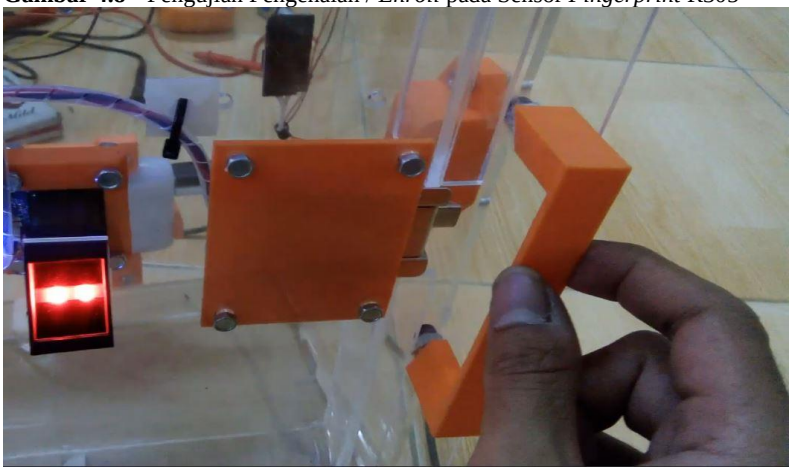
Tabel 4.8 Pengujian Sensor *Fingerprint* dengan Kondisi Jari dari *User* Pertama pada Percobaan Keempat

User 1	Jari yang Digunakan	Kondisi Jari pada saat *Enroll*				Kondisi Jari pada saat Mengakses				Kondisi Solenoid
		Bersih	Kotor	Basah	Luka	Bersih	Kotor	Basah	Luka	
	Ibu Jari	✓				✓				On
	Jari Telunjuk				✓					Off
	Jari Tengah	✓				✓				On
	Jari Manis		✓							Off
	Jari Kelingking			✓				✓		On

Selanjutnya, pada Gambar 4.8, merupakan pengujian oleh user yang pertama dengan menggunakan ibu jari sebagai pengenalan / *enroll* dan juga digunakan saat mengakses sehingga pintu dari *prototype* akan terbuka seperti pada Gambar 4.9.

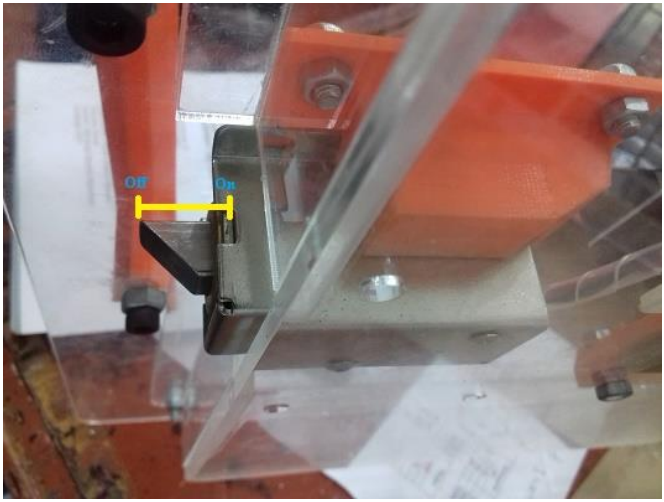


Gambar 4.8 Pengujian Pengenalan / *Enroll* pada Sensor *Fingerprint* R305



Gambar 4.9 Akses Diterima dan Pintu akan Membuka

Pada tabel pengukuran dan pengujian sensor *fingerprint* diatas, terdapat kolom “Kondisi Solenoid” yang berisikan keadaan atau kondisi solenoid tersebut. Solenoid akan memendek pada saat kondisi “On” jika akses atau pola sidik jari diterima sesuai atau sama dengan pola sidik jari pada saat pengenalan / *enroll* sehingga pintu dapat dibuka, dan solenoid akan masih tetap memanjang pada saat “Off” jika akses ditolak atau pola sidik jari yang diberikan tidak sesuai atau tidak terdeteksi dengan kondisi jari saat *enroll* sehingga pintu tidak dapat dibuka. Berikut pada Gambar akan menjelaskan kondisi solenoid yang digunakan.



Setelah mendapatkan data untuk sensor *fingerprint*, langkah selanjutnya yaitu melakukan pengujian dan pengukuran yang sama kepada user yang kedua seperti yang tercantum pada Tabel

Tabel 4.9 Pengujian Sensor *Fingerprint* dengan Kondisi Jari dari *User* Kedua pada Percobaaan Pertama

User 2	Jari yang Digunakan	Kondisi Jari pada saat *Enroll*				Kondisi Jari pada saat Mengakses				Kondisi Solenoid
		Bersih	Kotor	Basah	Luka	Bersih	Kotor	Basah	Luka	
	Ibu Jari		✓							Off
	Jari Telunjuk	✓				✓				On
	Jari Tengah			✓						Off
	Jari Manis				✓					Off
	Jari Kelingking	✓				✓				On

Tabel 4.10 Pengujian Sensor *Fingerprint* dengan Kondisi Jari dari *User* Kedua pada Percobaaan Kedua

User 2	Jari yang Digunakan	Kondisi Jari pada saat *Enroll*				Kondisi Jari pada saat Mengakses				Kondisi Solenoid
		Bersih	Kotor	Basah	Luka	Bersih	Kotor	Basah	Luka	
	Ibu Jari		✓							Off
	Jari Telunjuk	✓				✓				On
	Jari Tengah			✓						Off
	Jari Manis				✓					Off
	Jari Kelingking	✓				✓				On

Tabel 4.11 Pengujian Sensor *Fingerprint* dengan Kondisi Jari dari *User* Kedua pada Percobaaan Ketiga

User 2	Jari yang Digunakan	Kondisi Jari pada saat *Enroll*				Kondisi Jari pada saat Mengakses				Kondisi Solenoid
		Bersih	Kotor	Basah	Luka	Bersih	Kotor	Basah	Luka	
	Ibu Jari		✓							Off
	Jari Telunjuk	✓				✓				On
	Jari Tengah			✓						Off
	Jari Manis				✓					Off
	Jari Kelingking	✓				✓				On

Tabel 4.12 Pengujian Sensor *Fingerprint* dengan Kondisi Jari dari *User* Kedua pada Percobaaan Keempat

User 1	Jari yang Digunakan	Kondisi Jari pada saat *Enroll*				Kondisi Jari pada saat Mengakses				Kondisi Solenoid
		Bersih	Kotor	Basah	Luka	Bersih	Kotor	Basah	Luka	
	Ibu Jari		✓							Off
	Jari Telunjuk	✓				✓				On
	Jari Tengah			✓						Off
	Jari Manis				✓					Off
	Jari Kelingking	✓				✓				On

Berdasarkan kedua tabel dari pengujian untuk masing – masing *user*, data diatas diambil tolak ukur jika sensor *fingerprint* dapat diakses jika kondisi jari dalam keadaan bersih maka akses dapat diterima, selain itu kondisi jari yang lain akan ditolak. Jadi dari data tersebut dapat diambil kesimpulan seperti dalam Tabel 4.13 berikut ini.

Tabel 4.13 Kalkulasi Percobaan dari Hasil Pengujian pada Kedua *User*

User 1	Jari yang Digunakan	Kondisi Jari yang Digunakan	Percobaan yang Dilakukan	Kesalahan Hasil yang Didapatkan dari Percobaan
	Ibu Jari	Bersih	4x	0x
	Jari Telunjuk	Luka	4x	0x
	Jari Tengah	Bersih	4x	0x
	Jari Manis	Kotor	4x	0x
	Jari Kelingking	Basah	4x	1x
User 2	Jari yang Digunakan	Kondisi Jari yang Digunakan	Percobaan yang Dilakukan	Kesalahan Hasil yang Didapatkan dari Percobaan
	Ibu Jari	Kotor	4x	0x
	Jari Telunjuk	Bersih	4x	0x
	Jari Tengah	Basah	4x	0x
	Jari Manis	Luka	4x	0x
	Jari Kelingking	Bersih	4x	0x

Dari kesimpulan hasil Tabel 4.13 diatas dapat dikalkulasikan jika dari *user* 1 dan 2 telah dilakukan 4x percobaan dari masing – masing jari pada tangan kanan yang sebanyak 5 jari, maka:

4 (percobaan) x 10 (jumlah jari) = 40x hasil percobaan

dari percobaan diatas, terdapat beberapa kesalahan pembacaan sejumlah 1x, maka dapat diambil besar nilai %Error yaitu 2.5%

% Error didapatkan dari rumus :

$$\%Error = \frac{KesalahanHasilPercobaan}{JumlahPercobaan} x100\%$$

BAB V
PENUTUP

Setelah melakukan perencanaan, perancangan, dan pengujian alat maka dapat diambil kesimpulan dan memberikan saran demi penyempurnaan Tugas Akhir ini.

5.1 Kesimpulan

Hasil dari pengujian serta analisa data dari masing – masing *I/O* atau sensor yang dipakai pada rancangan alat, maka dapat disimpulan bahwa :

1. Data berupa nilai suhu dan kelembaban dapat ditampilkan secara langsung pada LCD
2. Besar nilai %Error untuk pengukuran nilai suhu dan kelembaban menggunakan alat *hygrometer* didapatkan rata – rata nilai sebesar 1.108%. sedangkan besar nilai %Error untuk pengukuran nilai suhu dan kelembaban menggunakan sensor DHT11 didapatkan rata – rata nilai sebesar 1.38%. Jadi masih terdapat perhitungan yang kurang valid jika menggunakan sensor DHT11 meskipun nilainya sangat kecil.
3. Kondisi jari *user* rancangan alat, sangat mempengaruhi dalam proses mengakses sensor *fingerprint.* Dan besar nilai %Error untuk pengujian dari sensor *fingerprint* yaitu sebesar 2.5% yang dimana didapatkan 1x kesalahan dari 40x percobaan / pengujian

5.2 Saran

Untuk pengembangan alat selanjutnya sebaiknya alat tersebut dibuat dengan bahan yang lebih keras atau tidak mudah rusak, mengingat penempatan dan penggunaan rancangan alat tersebut rawan dengan hal – hal negatif seperti pencurian maupun kerusakan dikarenakan faktor lain.

-----Halaman ini sengaja dikosongkan-----

DAFTAR PUSTAKA

[1] Arief, H. S. , "Perbandingan Akurasi Pengukuran Suhu dan Kelembaban Antara Sensor DHT11 dan DHT22", *Jurnal Infotel* Vol. 6 No. 2, Sekolah Tinggi Teknologi Telematika Telkom, Purwokerto, 2014.

[2] Kadir, A. , "Panduan Praktis Mempelajari Aplikasi Mikrokontroller dan Pemrogramannya Menggunakan Arduino", Edisi 1 ISBN 978-979-29-4017-6, ANDI Publisher, Yogyakarta, 2013.

[3] Kaur, M., Singh, M., Girdar, A. dan Sandhu, P. , "Fingerprint Verification System using Minutiae Extraction Technique", *World Academy of Science, Engineering and Technology 46*, pp. 497-502, 2008.

[4] Ahmad, U. , "Pengolahan Citra Digital dan Teknik Pemrogramannya", Graha Ilmu, Yogyakarta, 2005.

[5] Sapta, A. , "Mengukur Suhu dan Kelembaban Udara dengan Sensor DHT11 dan Arduino", http://saptaji.com/2016/08/10/mengukur-suhu-dan-kelembaban-udara-dengan-sensor-dht11-dan-arduino/, 2016. (diakses tanggal 17 Maret 2017).

[6] Adafruit, "Spring Loaded Electromagnet Solenoid 12V Pull Type", https://www.adafruit.com/product/1512, 2012. (diakses tanggal 25 Maret 2017).

[7] DFRobot , "Digital-output Relative Humidity & Temperature Sensor/Module–DHT11", https://image.dfrobot.com/image/data/KIT0003/DHT11%20datasheet.pdf, 2013. (diakses tanggal 23 Maret 2017).

[8] Ecadio , "Mengenal Arduino Mega 2560", http://ecadio.com/belajar-dan-mengenal-arduino-mega, 2015. (diakses tanggal 26 April 2017).

[9] Arduino , "Arduino Mega 2560", https://www.arduino.cc/en/Main/ArduinoBoardMega2560, 2014. (diakses tanggal 23 Februari 2017).

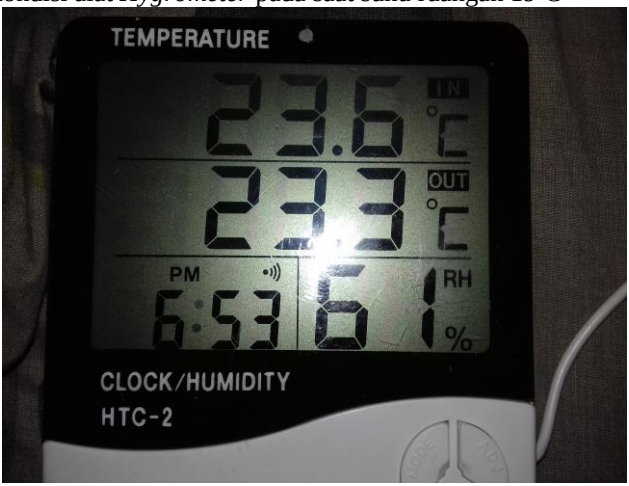
[10] DFRobot , "Karakter LCD 16x2", http://image.dfrobot.com/image/data/FIT0127/datasheet.pdf, 2014. (diakses tanggal 12 Maret 2017).

-----Halaman ini sengaja dikosongkan-----

LAMPIRAN A

A.1 Hasil Pengukuran Suhu dan Kelembaban Menggunakan Hygrometer

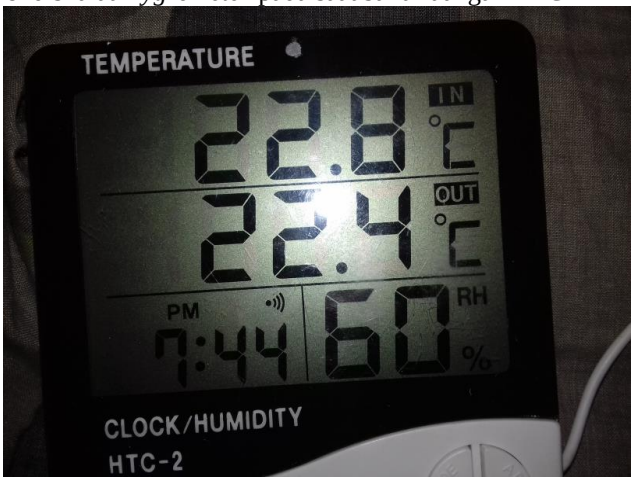
- Kondisi alat *Hygrometer* pada saat suhu ruangan 19ºC



- Kondisi alat *Hygrometer* pada saat suhu ruangan 20ºC

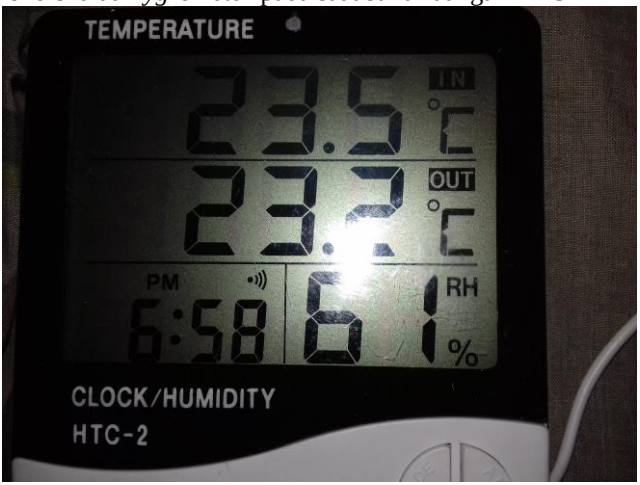
- Kondisi alat *Hygrometer* pada saat suhu ruangan 21ºC

- Kondisi alat *Hygrometer* pada saat suhu ruangan 22ºC

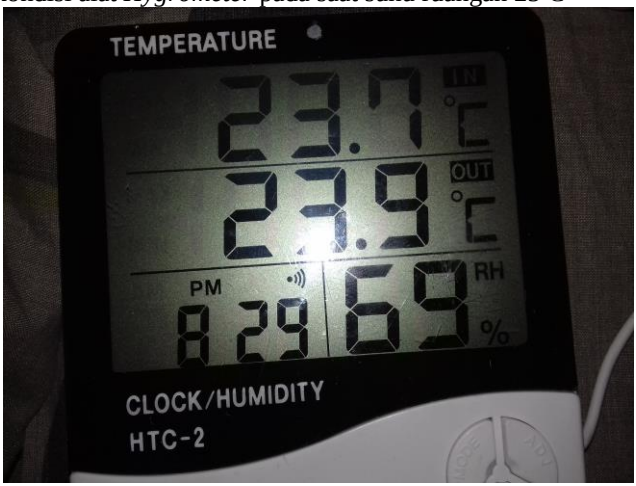


- Kondisi alat *Hygrometer* pada saat suhu ruangan 23ºC

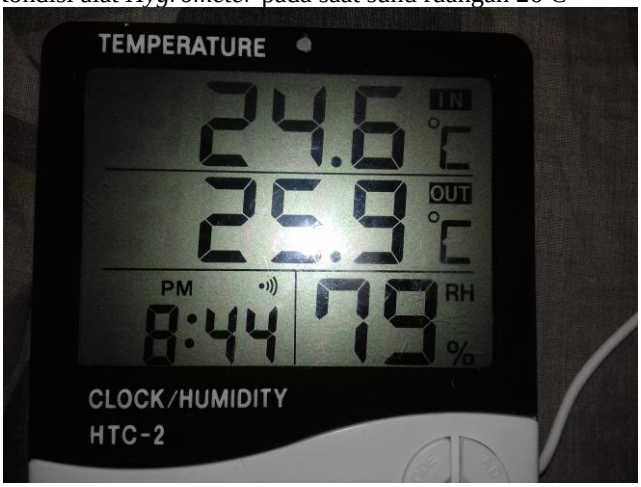
- Kondisi alat *Hygrometer* pada saat suhu ruangan 24ºC



- Kondisi alat *Hygrometer* pada saat suhu ruangan 25ºC



- Kondisi alat *Hygrometer* pada saat suhu ruangan 26ºC



- Kondisi suhu ruangan yang diatur 20ºC

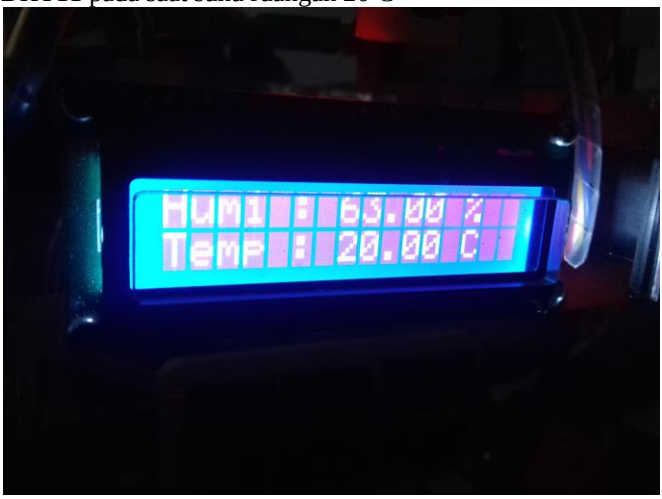
- Kondisi suhu ruangan yang diatur 22ºC

A.2 Hasil Pengukuran Suhu dan Kelembaban Menggunakan DHT11

- Nilai yang ditampilkan oleh LCD 16x2 yang menerima data dari DHT11 pada saat suhu ruangan 18ºC

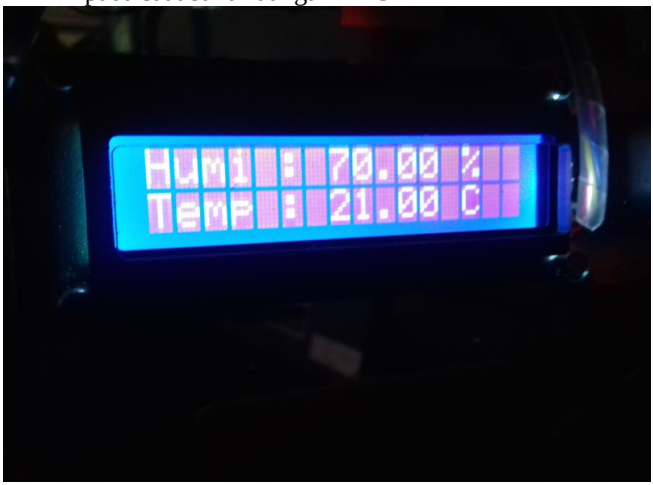
- Nilai yang ditampilkan oleh LCD 16x2 yang menerima data dari DHT11 pada saat suhu ruangan 19ºC

- Nilai yang ditampilkan oleh LCD 16x2 yang menerima data dari DHT11 pada saat suhu ruangan 20ºC



- Nilai yang ditampilkan oleh LCD 16x2 yang menerima data dari DHT11 pada saat suhu ruangan 21ºC

- Nilai yang ditampilkan oleh LCD 16x2 yang menerima data dari DHT11 pada saat suhu ruangan 22ºC



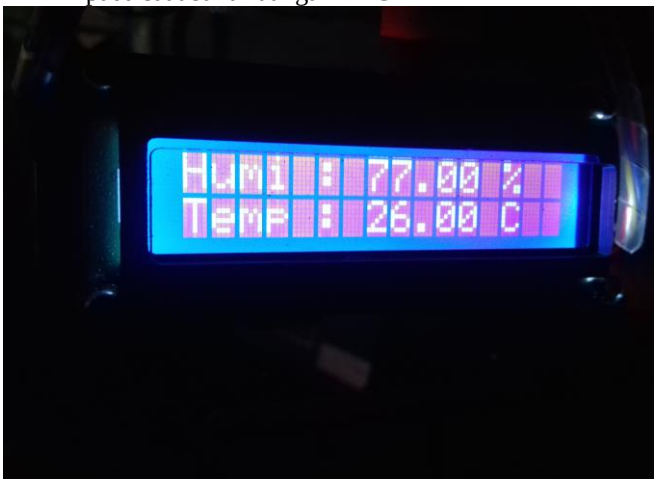
- Nilai yang ditampilkan oleh LCD 16x2 yang menerima data dari DHT11 pada saat suhu ruangan 23ºC

- Nilai yang ditampilkan oleh LCD 16x2 yang menerima data dari DHT11 pada saat suhu ruangan 24ºC

- Nilai yang ditampilkan oleh LCD 16x2 yang menerima data dari DHT11 pada saat suhu ruangan 25ºC

- Nilai yang ditampilkan oleh LCD 16x2 yang menerima data dari DHT11 pada saat suhu ruangan 26ºC

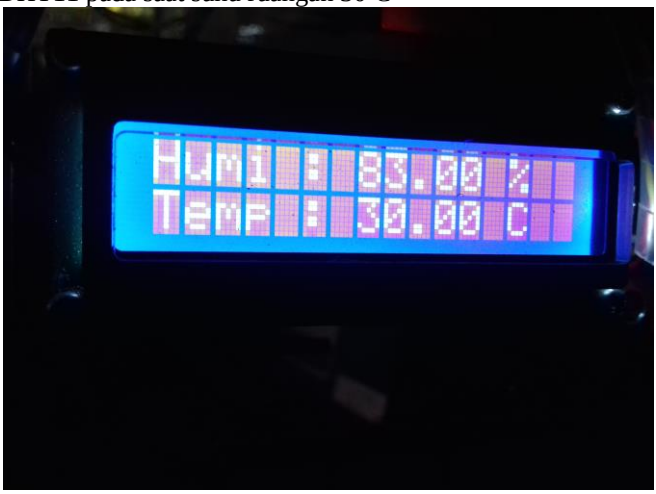
- Nilai yang ditampilkan oleh LCD 16x2 yang menerima data dari DHT11 pada saat suhu ruangan 27ºC



- Nilai yang ditampilkan oleh LCD 16x2 yang menerima data dari DHT11 pada saat suhu ruangan 28ºC

- Nilai yang ditampilkan oleh LCD 16x2 yang menerima data dari DHT11 pada saat suhu ruangan 29ºC

- Nilai yang ditampilkan oleh LCD 16x2 yang menerima data dari DHT11 pada saat suhu ruangan 30ºC



A.3 Database Program untuk Masing-masing Input dan Output

Library Program untuk LCD 16x2 :

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x3f ,2,1,0,4,5,6,7,3, POSITIVE);
        (inisialisasi LCD)

lcd.createChar(0, degree);
  Serial.begin(9600);
  Serial.println("DHTxx test!");
  lcd.begin (16,2); //LCD untuk ukuran 16x2
  dht.begin();
  pinMode (12 ,OUTPUT);          ( pengaturan pin LCD dan
kecerahanya )

lcd.setCursor(0, 0); //baris pertama
  lcd.print("Humi : ");
  lcd.print(h);
  lcd.print(" %");
  lcd.setCursor(0, 1); //baris kedua
```

```
  lcd.print("Temp : ");
  lcd.print(t);
  lcd.print(" C");
  if (h >35 ){
```

digitalWrite (12, HIGH); (pengaturan penempatan awal mula karakter yang akan di tampilkan)

Library Program untuk DHT11 :

```
#include "DHT.h"
```

(inisialisasi sensor DHT11)

```
#define DHTPIN 2
```

(inisialisasi pin yang akan di pakai)

```
#define DHTTYPE DHT11
```

(type sensor DHT)

```
byte degree[8] = {
  0b00110,
  0b00000,
  0b00000,
  0b00000,
  0b00000,
  0b00000,
  0b00000,
  0b00000
};
```

DHT dht(DHTPIN, DHTTYPE); (pengaturan byte sensor bekerja)

```
delay(100);
  float h = dht.readHumidity();
  float t = dht.readTemperature();
```

float f = dht.readTemperature(true);(pengaturan dellay sensor DHT11)

```
if (isnan(h) || isnan(t) || isnan(f)) {
   Serial.println
```

return; (pengaturan apabila gagal mengambil data dari DHT sensor!);

float hif = dht.computeHeatIndex(f, h);
float hic = dht.computeHeatIndex(t, h, false); (Digunakan untuk angka desimal (floating point). Memakai 4 byte (32 bit) dari RAM dan mempunyai rentang dari -3.4028235E+38 dan 3.4028235E+38 dan menentukan output dari sensor *high* atau *low*)

Serial.println("fan on");
digitalWrite (9, LOW);
delay(2000);
digitalWrite (9, HIGH);(apabila sushu melebihi batas yang telah di atur maka kipas akan menyala)

Library Program untuk *Fingerprint* :

#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>(inisialisasi sensor *fingerprint*0

int getFingerprintIDez();

SoftwareSerial mySerial(10, 11);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
(inisialisasi pin yang akan di gunakan)

while (!Serial); // For Yun/Leo/Micro/Zero/...

Serial.begin(9600);
Serial.println("Adafruit finger detect test");
pinMode (9 ,OUTPUT);
digitalWrite (9, HIGH); (menentukan menentukan output dari sensor *high* atau *low*)

finger.begin(57600);(mengatur data *rate* untuk sensor serial *port*)

if (finger.verifyPassword()) {
Serial.println("apabila sensor di *detect*!");

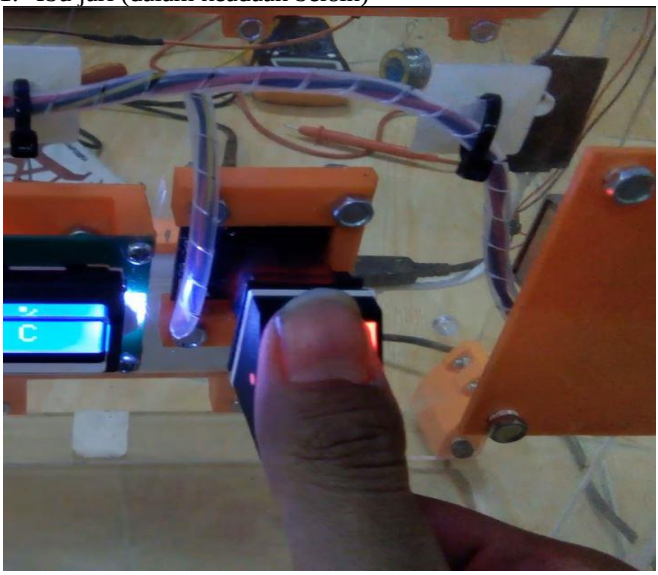
```
  } else {
    Serial.println(“apabila sensor tidak di detect”);
    while (1);
  }
  Serial.println("menunggu sidik jari yang cocok...");

getFingerprintIDez();
  delay(50);                (mengatur kecepatan sensor membaca
sidik jari).

int8_t getFingerprintID() { (membaca sidik jari)
  uint8_t p = finger.getImage(); (mengambil data sidik jari)
  switch (p) {
    case FINGERPRINT_OK:
      Serial.println(“gambar sidik jari telah di ambil”);
      break;
    case FINGERPRINT_NOFINGER:
      Serial.println(“tidak ada sidik jari yang di baca”);
      return p;
    case FINGERPRINT_PACKETRECIEVEERR:
      Serial.println(“komunikasi error”);
      return p;
    case FINGERPRINT_IMAGEFAIL:
      Serial.println(“gambar sidik jari error”);
      return p;
    default:
      Serial.println(“error yang tidak di ketahui”);
      return p;

/ apabila sukses membaca sidik jari!

  p = finger.image2Tz();
  switch (p) {
    case FINGERPRINT_OK:
      Serial.println("gambar sidik jari di convert");
      break;
    case FINGERPRINT_IMAGEMESS:
      Serial.println(“gambar sidik jari kurang jelas”);
      return p;
```

```
    case FINGERPRINT_PACKETRECIEVEERR:
      Serial.println("komunikasi  error");
      return p;
    case FINGERPRINT_FEATUREFAIL:
      Serial.println("tidak dapat menemukan fingerprint
features");
      return p;
    case FINGERPRINT_INVALIDIMAGE:
      Serial.println("tidak dapat menemukan fingerprint
features");
      return p;
    default:
      Serial.println("error yang tidak di ketahui");
      return p;
// apabila berhasil di convert!
  p = finger.fingerFastSearch();
  if (p == FINGERPRINT_OK) {
    Serial.println("berhasil menenmukan kecocokan!");
  } else if (p == FINGERPRINT_PACKETRECIEVEERR) {
    Serial.println("komunikasi error");
    return p;
  } else if (p == FINGERPRINT_NOTFOUND) {
    Serial.println("tidak menemukan kecocokan sidik jari");
    return p;
  } else {
    Serial.println("error yang tidak di ketahui ");
    return p;

menemukan kecocokan sidik jari!
  Serial.print("menemukan kecocokan dengan ID data sidik
jari yang telah disimpan sebelumnya #");
Serial.print(finger.fingerID);
  Serial.print("kecocokan data sidik jari 100%");
Serial.println(finger.confidence);
}
Apabila sidik jari masih belom cocok!
// returns -1 if failed, otherwise returns ID #
int getFingerprintIDez() {
  uint8_t p = finger.getImage();
```

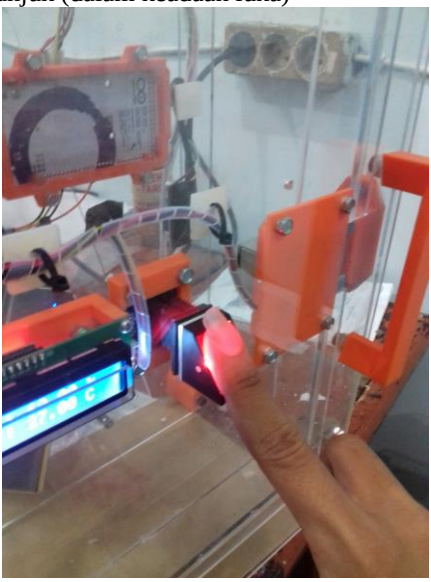
```
  if (p != FINGERPRINT_OK)  return -1;

  p = finger.image2Tz();
  if (p != FINGERPRINT_OK)  return -1;

  p = finger.fingerFastSearch();
  if (p != FINGERPRINT_OK)  return -1;

  //apabila sidik jari sudah cocok

  Serial.print("menemukan kecocokan dengan ID data sidik jari yang telah di simpan sebelumnya "); Serial.print(finger.fingerID);
  Serial.print(" kecocokan data sidik jari 100% "); Serial.println(finger.confidence);
  return finger.fingerID;
```

A.4 Hasil Pengujian Sensor Fingerprint

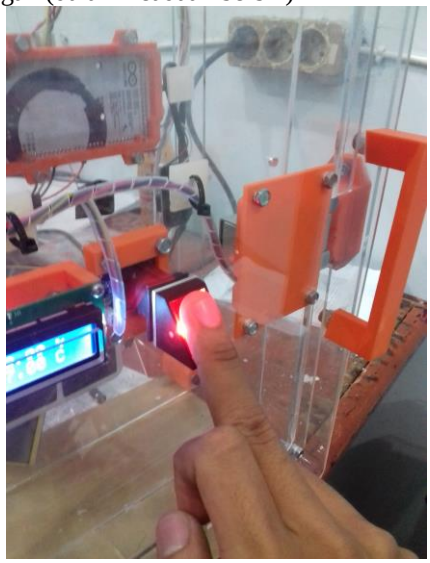
- Kondisi jari yang digunakan untuk *enroll* dan uji coba sensor *fingerprint* oleh *user* pertama
  1. Ibu jari (dalam keadaan bersih)



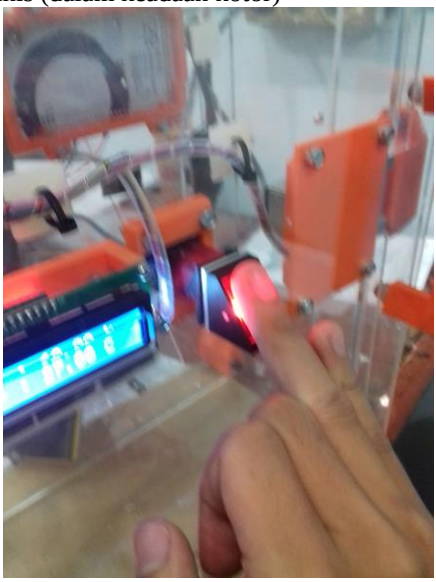
2. Jari telunjuk (dalam keadaan luka)



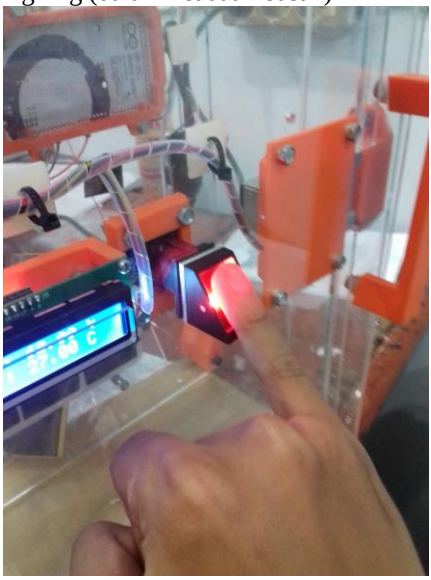
3. Jari tengah (dalam keadaan bersih)



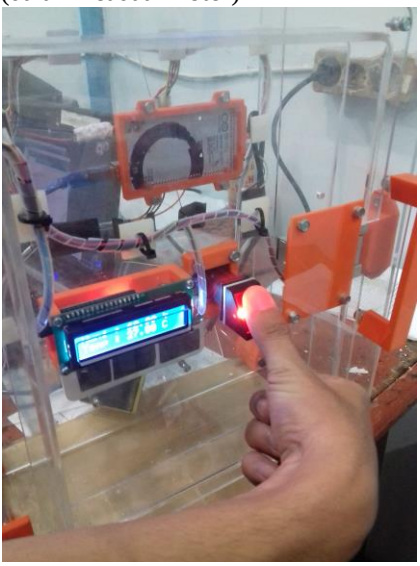
4. Jari manis (dalam keadaan kotor)



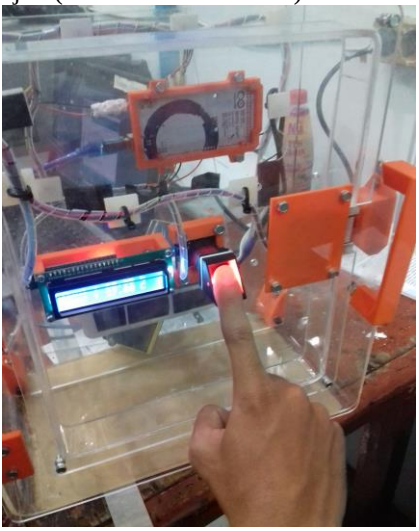
5. Jari kelingking (dalam keadaan basah)



- Kondisi jari yang digunakan untuk *enroll* dan uji coba sensor *fingerprint* oleh *user* kedua
  1. Ibu jari (dalam keadaan kotor)

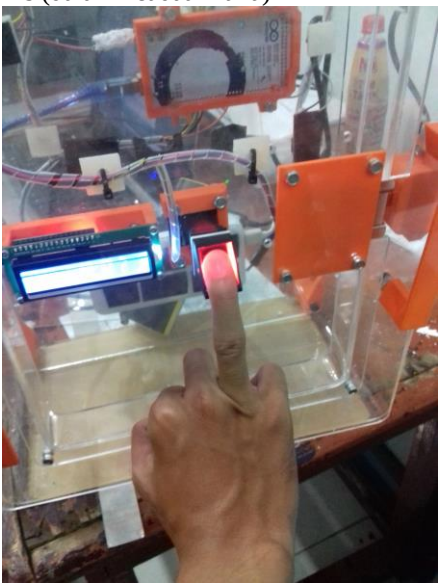


  2. Jari telunjuk (dalam keadaan bersih)



3. Jari tengah (dalam keadaan basah)

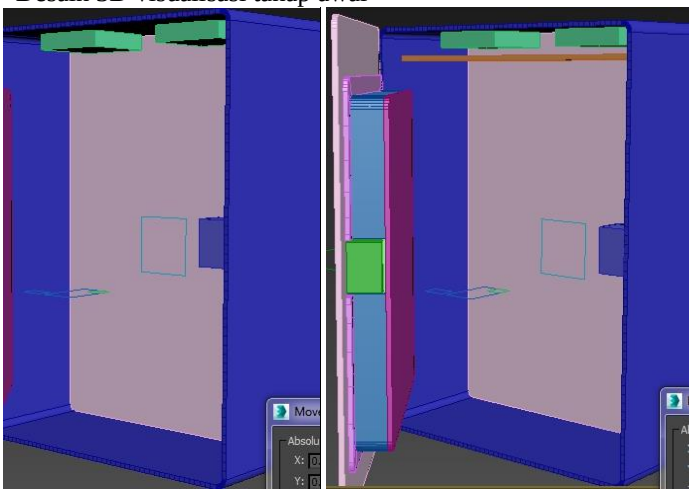
4. Jari manis (dalam keadaan luka)

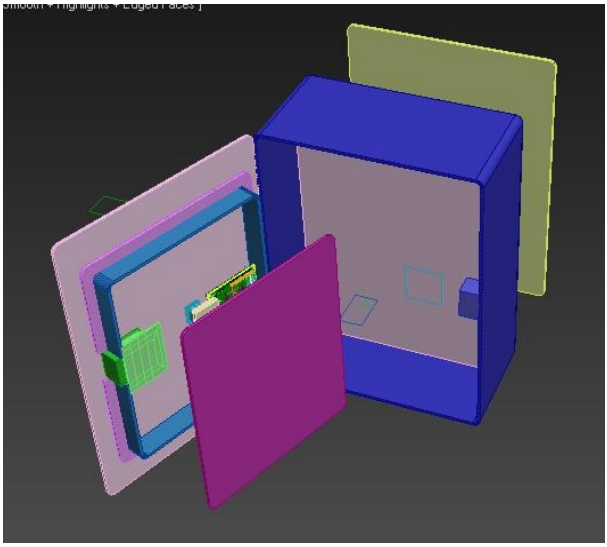


5. Jari kelingking (dalam keadaan bersih)

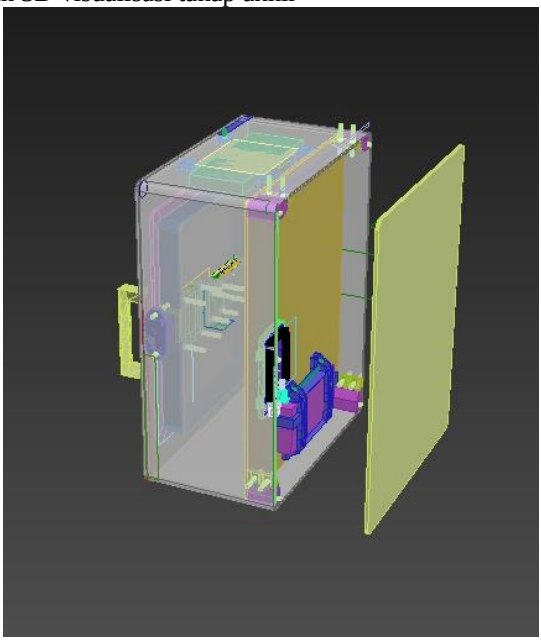
A.5 Desain Rancangan Prototype

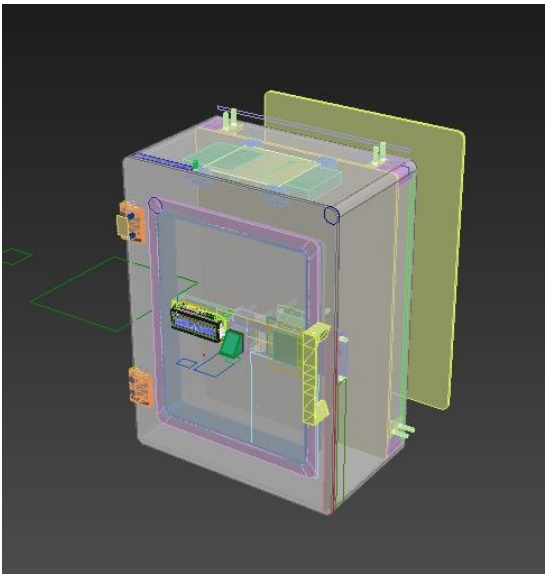
- Desain 3D visualisasi tahap awal



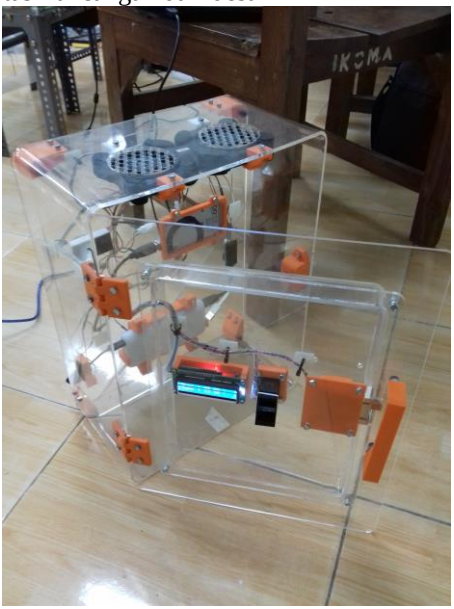


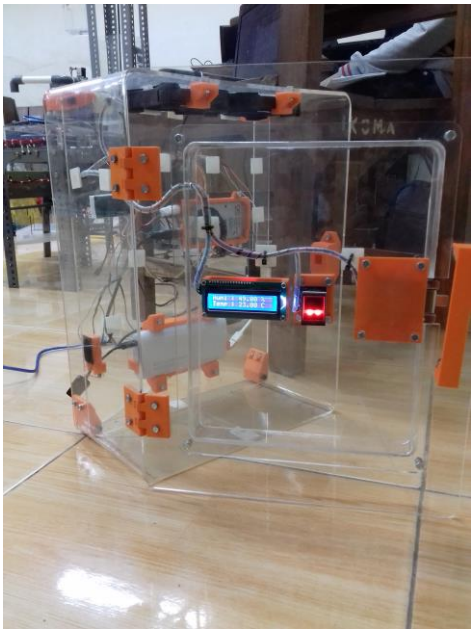
- Desain 3D visualisasi tahap akhir



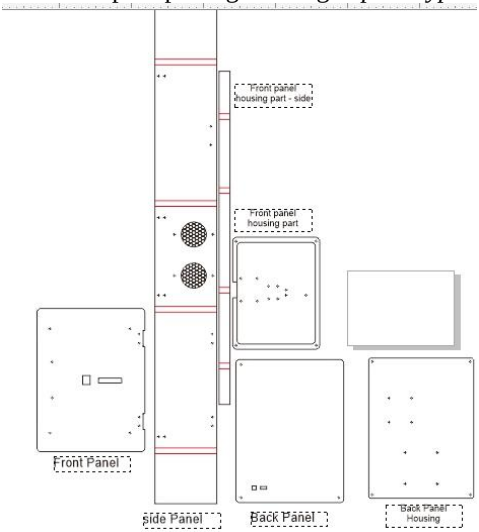


- Implementasi rancangan dari desain





- Desain pemetaan / pola potong rancangan *prototype*



----Halaman ini sengaja dikosongkan-----

DAFTAR RIWAYAT HIDUP

Nama : Rendra Kurnia R.
NRP : 2214030051
Jenis Kelamin : Laki – laki
Tempat, Tanggal Lahir : Surabaya, 31 Oktober 1996
Agama : Islam
Alamat : Jl. Sepanjang Tani RT 08, RW 06 Kec. Taman Kab. Sidoarjo.
No. Hp : 08999585578
Email : rendrakurnia31@gmail.com

RIWAYAT PENDIDIKAN

- 2002 – 2008 : SD Muhammadiyah 1 – 2 Taman
- 2008 – 2011 SMPN 1 Taman
- 2011 – 2014 : SMAN 1 Taman
- 2014 – 2017 : Program Studi Komputer Kontrol, Departemen Teknik Elektro Otomasi, Fakultas Vokasi, ITS Surabaya.