

BAB II

LANDASAN TEORI

2.1 Sensor Sidik Jari

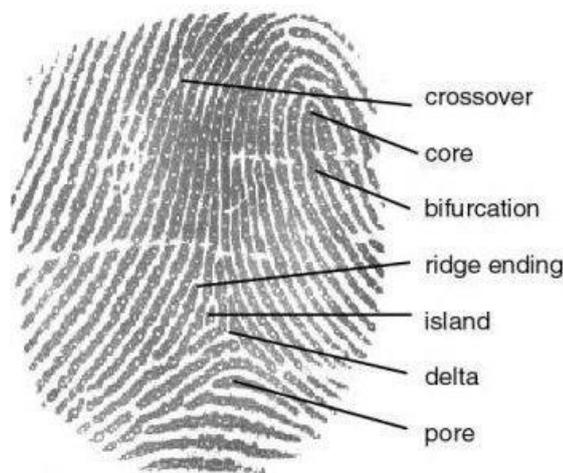
Sistem biometrik sidik jari merupakan sistem yang paling banyak digunakan saat ini, karena memiliki kecenderungan tingkat akurasi yang tinggi dan mudah diterapkan. Sifat yang dimiliki sidik jari antara lain:

1. Perennial nature, yaitu guratan-guratan pada sidik jari yang melekat pada kulit manusia seumur hidup.
2. Immutability, yaitu sidik jari seseorang tidak pernah berubah, kecuali mendapatkan kecelakaan yang serius.
3. Individuality, pola sidik jari adalah unik dan berbeda untuk setiap orang.



Gambar 2.1 Sensor Sidik Jari
(Elma Tri Yulida, 2018)

Ciri khas sidik jari yang digunakan adalah sidik jari yang diidentifikasi dengan cara menganalisis detail dari guratan-guratan sidik jari yang dinamakan “minutiae” (Naslim Lathif, 2001). Minutiae berasal dari bahasa Inggris yang artinya barang tidak berarti atau rincian tidak penting dan terkadang diartikan sebagai detail. Minutiae sebenarnya merupakan rincian sidik jari yang tidak penting bagi kita, tetapi bagi sebuah mesin sidik jari itu adalah detail yang diperhatikan.



Gambar 2.2 Definisi Sidik Jari.
(Bayu, 2017)

Pemindai sidik jari saat ini sudah banyak digunakan, mulai dari absensi, sebagai access control, hingga sebagai identitas pribadi seperti pada SIM atau passport. Seperti halnya bagian tubuh yang lain, sidik jari terbentuk karena faktor genetic dan lingkungan. Kode genetik pada DNA memberi perintah untuk terbentuknya janin yang secara spesifik membentuk hasil secara acak. Demikian juga halnya dengan sidik jari. Sidik jari memiliki bentuk unik bagi setiap orang. Artinya setiap orang memiliki bentuk sidik jari yang berbeda-beda meskipun terlahir kembar. Jadi, walaupun sidik jari terlihat seperti sama bila dilihat sekilas, buat penyidik terlatih atau dengan menggunakan *software* khusus akan terlihat perbedaannya.

Secara umum, sidik jari dapat dibedakan menjadi beberapa tipe menurut *Henry Classification System*, yaitu *loop pattern*, *whorl pattern*, dan *arch pattern*. Hampir 2/3 manusia memiliki sidik jari dengan *loop pattern*, 1/3 lainnya memiliki sidik jari dengan *whorl pattern*, dan hanya 5-10% yang memiliki sidik jari dengan *arch pattern*. Pola-pola sidik jari seperti ini yang digunakan untuk membedakan sidik jari secara umum. Namun, untuk mesin pembaca sidik jari, perbedaan seperti ini tidak cukup. Karena itulah, mesin sidik jari dilengkapi dengan pengenalan lain yang disebut *minutiae*.



Gambar 2.3 *Arch pattern.*
(Bayu, 2017)



Gambar 2.4 *Whorl Pattern.*
(Bayu, 2017)



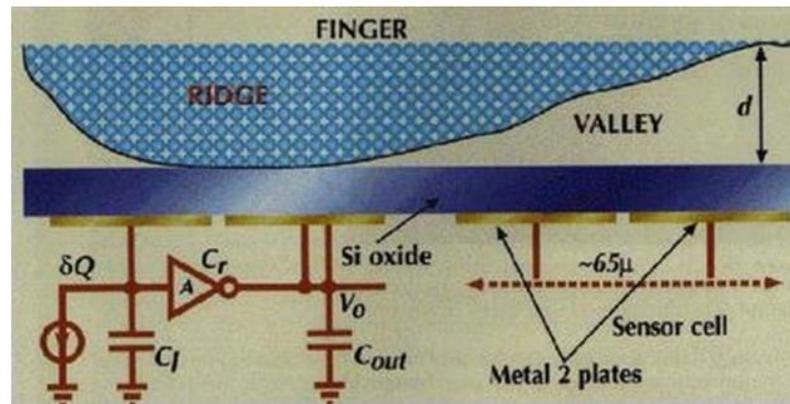
Gambar 2.5 *Loop Pattern.*
(Bayu, 2017)

Untuk lebih jelasnya, *minutiae* pada sidik jari adalah titik-titik yang mengacu kepada *crossover* (persilangan dua garis), *core* (putar-balikan sebuah garis), *bifurcation* (percabangan sebuah garis), *ridge ending* (berhentinya sebuah garis), *island* (sebuah garis yang sangat pendek), *delta* (pertemuan dari tiga buah garis yang membentuk sudut) dan *pore* (percabangan sebuah garis yang langsung diikuti dengan menyatunya kembali percabangan tersebut sehingga membentuk sebuah lingkaran kecil).

Mesin pemindai sidik jari akan mencari titik-titik ini dan membuat pola dengan menghubungkan-hubungkan titik-titik ini. Pola yang didapat dengan menghubungkan titik-titik inilah yang nantinya akan digunakan untuk melakukan pencocokan bila ada jari yang menempel pada mesin sidik jari. Jadi, sebenarnya mesin sidik jari tidak mencocokkan pola yang didapat dari *minutiae*- *minutiae* ini. Mesin pemindai sidik jari bekerja dengan mengambil gambar dari sidik jari tersebut. Sebenarnya banyak cara dapat dilakukan untuk mengambil gambar sidik jari tersebut namun metode umum yang dilakukan adalah dengan 2 cara yaitu dengan sensor optikal dan kapasitansi.

Inti dari sensor optikal adalah dengan adanya CCD (*Charge Couple Device*) yang cara kerjanya sama seperti system sensor yang terdapat pada kamera digital dan *camcorder*. CCD merupakan chip silikon yang terbentuk dari ribuan atau bahkan jutaan *diode fotosensitif* yang disebut *photosites*, *photodelements* atau disebut juga piksel. Tiap *photosite* menangkap suatu titik objek kemudian dirangkai dengan hasil tangkapan *photosite* lain menjadi suatu gambar. 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 gambar ulang untuk ditampilkan kembali. Perekaman gambar yang dilakukan oleh CCD sebenarnya dalam format '*grayscale*' atau monochrome dengan 256 macam intensitas warna dari putih sampai hitam.

Sensor kapasitif bekerja berdasarkan prinsip pengukuran kapasitansi dari material yang dipindai. Material tersebut bisa saja besi, baja, alumunium, tembaga, kuningan bahkan hingga air. Berbeda dengan pemindai optikal yang menggunakan cahaya, pemindai kapasitif menggunakan arus listrik untuk mengukur besarnya kapasitas.

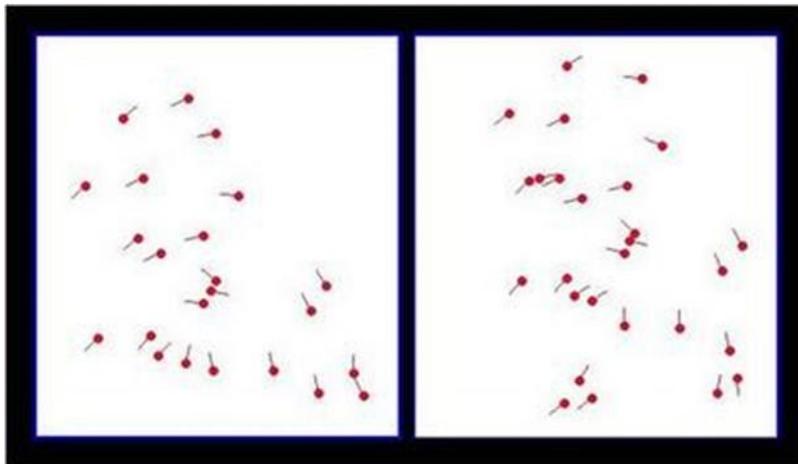


Gambar 2.6 Sensor kapasitif
(Bayu, 2017)

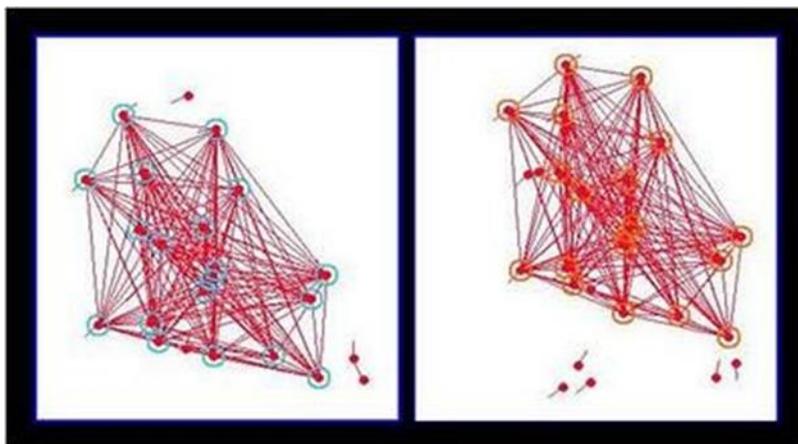
Diagram di atas menunjukkan sebuah sensor kapasitif sederhana. Dimana sensor dibuat dari beberapa chip semi konduktor pada sebuah sel yang tipis. Setiap sel memiliki tempat konduktor yang ditutupi dengan lapisan isolasi. Sensor tersebut terhubung dengan sebuah integrator yang dilengkapi dengan inverter penguat yang dapat menterjemahkan sehingga pada akhirnya akan membentuk sidik jari yang sedang dipindai. Setelah mesin pemindai sidik jari menyimpan *image* atau gambar yang diambil, mesin kemudian melakukan '*searching minutiae*' atau mencari titik-titik *minutiae*.



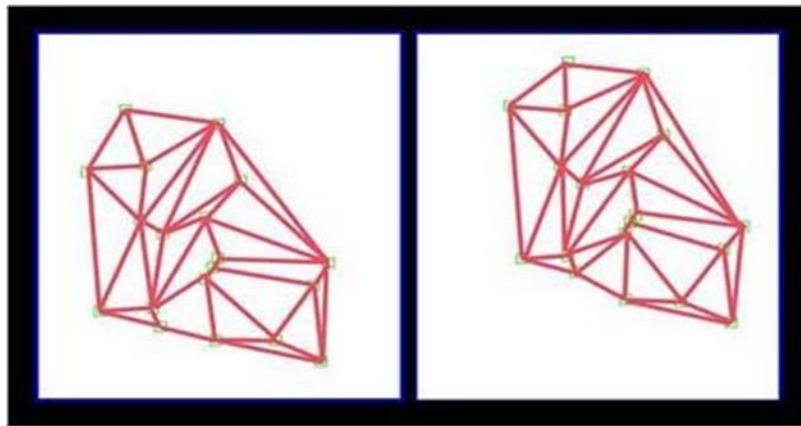
Gambar 2.7 *Searching minutiae.*
(Bayu, 2017)



Gambar 2.8 *Before match.*
(Bayu, 2017)



Gambar 2.9 *Match minutiae.*
(Bayu, 2017)



Gambar 2.10 *Matched result.*
(Bayu, 2017)

Jika mesin sidik jari mendapat pola yang sama maka proses identifikasi sudah berhasil. Tidak semua *minutiae* harus digunakan dan pola yang ditemukan tidak harus sama, maka kita dapat menyimpulkan bahwa posisi jari kita pada saat identifikasi pada mesin sidik jari juga tidak harus persis sama dengan pada saat kita menyimpan data sidik jari kita pertama kali pada mesin tersebut. Pemindai sidik jari optikal dan kapasitif dianggap menghasilkan tingkat keamanan yang tinggi, karena tidak bisa dipalsukan dengan foto copy sidik jari, sidik jari tiruan, atau bahkan dengan cetak lilin yang mendetil dengan guratan-guratan kontur sidik jari sekalipun (Joyner R. Oroh, 2014).

Sidik jari merupakan salah satu sistem biometri yang saat ini tersedia selain retina mata, wajah, dan sistem biometri lainnya. Identifikasi sidik jari telah diakui sebagai identifikasi yang canggih karena sidik jari masing-masing manusia sangat unik dan berbeda satu dengan lainnya. Identifikasi sidik jari sebelumnya telah digunakan untuk bidang forensik, investigasi kriminal, dan identifikasi anggota badan. Kelebihan identifikasi biometrik diharapkan dapat menjadi solusi atas kelemahan proses identifikasi personal, sehingga dapat memberikan pelayanan dan kemudahan.

Proses identifikasi personal memiliki dua tahapan, yaitu proses verifikasi dan proses pengenalan. Proses verifikasi dilakukan dengan memasukkan tokes, seperti kartu paspor, SIM, kartu kredit, ATM, dan sebagainya. Sedangkan pada proses pengenalan, memberikan pengenalan dengan menggunakan *password*, PIN, dan sebagainya. Sebagai contoh kartu ATM, proses identifikasi saat

memasukkan kartu ke dalam mesin ATM. Sedangkan proses pengenalan yaitu pada saat memasukkan PIN ATM. Proses ini memiliki kelemahan saat proses verifikasi yaitu hilang, lupa, dan salah menempatkan. Sedangkan pada tahap pengenalan terjadi kelupaan PIN dan atau PIN tersebut pernah diubah, dapat ditebak orang lain. Hal ini merujuk pada data pencurian ATM sebesar 25%. Pada sistem biometrik, tahapan prosesnya mengidentifikasi individu-individu berdasarkan keunikan pribadi. Seperti sensor sidik jari yang berbeda tiap manusia. Pengidentifikasi biometrik dianggap lebih handal dibandingkan identifikasi personal. Mekanisme awal yaitu fase penyimpanan (enrollment). Pada fase ini masukan akan dipindai oleh sensor biomterik yang merupaka reпреntasi karakteristik digital. Kemudian fase pencocokan. Selanjutnya disimpan dalam database yang diubah menjadi representasi digital. Pada sensor sidik jari yang digunakan, database penyimpanan terdapat pada sensor sidik jari tersebut. Pada fase pengenalan, karakteristik indivdu dibaca oleh sensor. Selanjutnya dikonversi ke format digital. Selanjutnya dicocokkan dengan identifikasi individu. Kelebihan sistem biometrik khususnya sensor sidik jari dibandingkan dengan sistem identifikasi personal yaitu:

1. Bersifat permanen, tidak dapat diubah.
2. Tidak akan hilang, lupa, tertinggal, dan salah menempatkan.
3. Tidak bisa disalahgunakan oleh orang lain.
4. Praktis dan mudah.

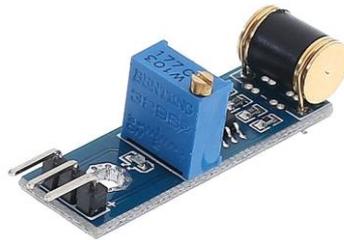
Kekurangan yang dimiliki sensor sidik jari antara lain:

1. Tidak bisa memindai saat kondisi jari basah dan berdebu.
2. Terjadi kesalahan pencocokan dan ketidakcocokan. (Eko Adi Sarwoko, 2006).

2.2 Sensor Getar 801S

Sensor getar adalah suatu perangkat atau device yang mengubah besaran fisis berupa getaran menjadi besaran elektrik yang bisa berupa tegangan maupun arus. Pada umumnya getaran ini diubah menjadi arus karena pertimbangan bahwa jarak antara sensor dengan kontroler tidak dekat, ada kemungkinan jaraknya jauh. Bila getaran tersebut diubah menjadi arus, maka arus yang dihasilkan sensor

dengan arus yang diterima kontroler akan sama besarnya. Hal ini tentunya akan berbeda jika getaran diubah menjadi tegangan. Tegangan yang dihasilkan sensor akan tidak sama dengan tegangan yang diterima kontroler sebagai akibat dari adanya losses.



Gambar 2.11 Sensor Getar 801S
(Elma Tri Yulida, 2018)

Tabel 2.1 Spesifikasi Sensor Getar

Chip utama	LM393, 801S
Tegangan operasi	3V - 5V
Arus operasi	15 mA
Dimensi	10 mm x 30 mm x 20 mm
Karakteristik	- Sensitivitas dapat disesuaikan - Getaran mendeteksi non-directional

2.3 Arduino UNO

Arduino adalah sebuah *board* mikrokontroler yang berbasis ATmega328. Arduino memiliki 14 pin *input/output* yang mana 6 pin dapat digunakan sebagai output PWM, 6 analog input, crystal osilator 16 MHz, koneksi USB, jack power, kepala ICSP, dan tombol reset. Arduino mampu *men-support* mikrokontroler; dapat dikoneksikan dengan komputer menggunakan kabel USB.

Arduino memiliki kelebihan tersendiri dibanding board mikrokontroler yang lain selain bersifat open source, arduino juga mempunyai bahasa pemrogramannya sendiri yang berupa bahasa C. Selain itu dalam board arduino sendiri sudah terdapat *loader* yang berupa USB sehingga memudahkan kita ketika kita memprogram mikrokontroler didalam arduino. Sedangkan pada kebanyakan

board mikrokontroler yang lain yang masih membutuhkan rangkaian *loader* terpisah untuk memasukkan program ketika kita memprogram mikrokontroler. Port USB tersebut selain untuk *loader* ketika memprogram, bisa juga difungsikan sebagai port komunikasi serial.

Arduino menyediakan 20 pin I/O, yang terdiri dari 6 pin input analog dan 14 pin digital input/output. Untuk 6 pin analog sendiri bisa juga difungsikan sebagai output digital jika diperlukan output digital tambahan selain 14 pin yang sudah tersedia. Untuk mengubah pin analog menjadi digital cukup mengubah konfigurasi pin pada program.

Dalam board kita bisa lihat pin digital diberi keterangan 0-13, jadi untuk menggunakan pin analog menjadi output digital, pin analog yang pada keterangan board 0-5 kita ubah menjadi pin 14-19. dengan kata lain pin analog 0-5 berfungsi juga sebagai pin output digital 14-16. Sifat *open source* arduino juga banyak memberikan keuntungan tersendiri untuk kita dalam menggunakan board ini, karena dengan sifat open source komponen yang kita pakai tidak hanya tergantung pada satu merek, namun memungkinkan kita bisa memakai semua komponen yang ada dipasaran.

Bahasa pemrograman arduino merupakan bahasa C yang sudah disederhanakan dengan bahasa pemrogramannya sehingga mempermudah kita dalam mempelajari dan mendalami mikrokontroller.

Tabel 2.2 Spesifikasi Arduino Uno

Chip Mikrokontroler	ATmega328
Tegangan Operasi	5 Volt
Tegangan input (rekomendasi)	7 -12 Volt
Tegangan input (limit)	6 – 20 Volt
Digital I/O pin	14 (6 diantaranya pin <i>PWM</i>)
Memori Flash	32 KB (ATmega328), sekitar 0.5 KB digunakan oleh bootloader
Arus DC per pin I/O	20 mA
Arus DC pin 3.3V	50 mA
Berat	25 gram

2.3.1 Power

Arduino dapat diberikan *power* melalui koneksi USB atau *power supply*. *Power*-nya diselek secara otomatis. *Power supply* dapat menggunakan adaptor DC atau baterai. Adaptor dapat dikoneksikan dengan mencolok *jack* adaptor pada koneksi port input *supply*. Board arduino dapat dioperasikan menggunakan *supply* dari luar sebesar 6 - 20 volt. Jika *supply* kurang dari 7V, kadangkala pin 5V akan menyuplai kurang dari 5 volt dan board bisa menjadi tidak stabil. Jika menggunakan lebih dari 12 V, tegangan diregulator bisa menjadi sangat panas dan menyebabkan kerusakan pada board. Rekomendasi tegangan ada pada 7 sampai 12 volt. Penjelasan pada pin power adalah sebagai berikut :

a. Vin

Tegangan input ke board arduino ketika menggunakan tegangan dari luar (seperti yang disebutkan 5 volt dari koneksi USB atau tegangan yang diregulasikan). Pengguna dapat memberikan tegangan melalui pin ini, atau jika tegangan suplai menggunakan power *jack*, aksesnya menggunakan pin ini.

b. 5V

Regulasi power *supply* digunakan untuk power mikrokontroller dan komponen lainnya pada board. 5V dapat melalui Vin menggunakan regulator pada board, atau *supply* oleh USB atau *supply* regulasi 5V lainnya.

c. 3V3

Suplai 3.3 volt didapat oleh FTDI chip yang ada di board. Arus maximumnya adalah 50mA

d. Pin Ground

Berfungsi sebagai jalur ground pada arduino

2.3.2 Memori

ATmega328 memiliki 32 KB flash memori untuk menyimpan kode, juga 2 KB yang digunakan untuk bootloader. ATmega328 memiliki 2 KB untuk SRAM dan 1 KB untuk EEPROM.

2.3.3 Input & Output

Setiap 14 pin digital pada arduino dapat digunakan sebagai input atau output, menggunakan fungsi `pinMode()`, `digitalWrite()`, dan `digitalRead()`. Input/output dioperasikan pada 5 volt. Setiap pin dapat menghasilkan atau menerima maximum 40 mA dan memiliki internal pull-up resistor (disconnected oleh default) 20-50K Ohm. Beberapa pin memiliki fungsi sebagai berikut :

- a. Serial : 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan mengirim (TX) TTL data serial. Pin ini terhubung pada pin yang koresponding dari USB ke TTL chip serial.
- b. Interrupt eksternal : 2 dan 3. Pin ini dapat dikonfigurasi untuk trigger sebuah interap pada low value, rising atau falling edge, atau perubahan nilai.
- c. PWM : 3, 5, 6, 9, 10, dan 11. Mendukung 8-bit output PWM dengan fungsi `analogWrite()`.
- d. SPI : 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Pin ini mensupport komunikasi SPI, yang mana masih mendukung hardware, yang tidak termasuk pada bahasa arduino.
- e. LED : 13. Ini adalah dibuat untuk koneksi LED ke digital pin 13. Ketika pin bernilai HIGH, LED hidup, ketika pin LOW, LED mati.

2.3.4 Komunikasi

Uno Arduino memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, Arduino lain, atau mikrokontroler lain. ATmega328 ini menyediakan UART TTL (5V) komunikasi serial, yang tersedia pada pin digital 0(RX) dan 1 (TX). *Firmware* Arduino menggunakan USB driver standar COM, dan tidak ada *driver* eksternal yang dibutuhkan. Namun, pada Windows, file. Ini diperlukan. Perangkat lunak Arduino termasuk monitor *serial* yang memungkinkan data sederhana yang akan dikirim ke *board* Arduino. RX dan TX LED di *board* akan berkedip ketika data sedang dikirim melalui chip USB-to-serial dan koneksi USB ke komputer.

2.3.5 Software Arduino

Arduino Uno dapat diprogram dengan perangkat lunak Arduino . Pada ATmega328 di Arduino terdapat *bootloader* yang memungkinkan Anda untuk meng-*upload* kode baru untuk itu tanpa menggunakan *programmer hardware eksternal*.

IDE Arduino adalah *software* yang sangat canggih ditulis dengan menggunakan *Java*. IDE Arduino terdiri dari:

1. Editor program, sebuah window yang memungkinkan pengguna menulis dan mengedit program dalam bahasa *Processing*.
2. *Compiler*, sebuah modul yang mengubah kode program (bahasa *Processing*) menjadi kode biner. Bagaimanapun sebuah mikrokontroler tidak akan bisa memahami bahasa *Processing*. Yang bisa dipahami oleh mikrokontroler adalah kode biner. Itulah sebabnya *compiler* diperlukan dalam hal ini.
3. *Uploader*, sebuah modul yang memuat kode biner dari komputer ke dalam memory didalam papan Arduino.

Sebuah kode program Arduino umumnya disebut dengan istilah *sketch*. Kata “*sketch*” digunakan secara bergantian dengan “kode program” dimana keduanya memiliki arti yang sama.

2.3.6 Bahasa Pemrograman Arduino

Seperti yang telah dijelaskan diatas program Arduino sendiri menggunakan bahasa C. walaupun banyak sekali terdapat bahasa pemrograman tingkat tinggi (*high level language*) seperti pascal, basic, cobol, dan lainnya. Walaupun demikian, sebagian besar dari paraprogramer profesional masih tetap memilih bahasa C sebagai bahasa yang lebih unggul, berikut alasan-alasannya:

- a. Bahasa C merupakan bahasa yang *powerfull* dan *fleksibel* yang telah terbukti dapat menyelesaikan program-program besar seperti pembuatan sistem operasi, pengolah gambar (seperti pembuatan game) dan juga pembuatan kompilator bahasa pemrograman baru.
- b. Bahasa C merupakan bahasa yang *portable* sehingga dapat dijalankan di beberapa sistem operasi yang berbeda. Sebagai contoh program yang kita

tulis dalam sistem operasi windows dapat kita kompilasi didalam sistem operasi linux dengan sedikit ataupun tanpa perubahan sama sekali.

- c. Bahasa C merupakan bahasa yang sangat populer dan banyak digunakan oleh programmer berpengalaman sehingga kemungkinan besar library pemrograman telah banyak disediakan oleh pihak luar/lain dan dapat diperoleh dengan mudah.
- d. Bahasa C merupakan bahasa yang bersifat modular, yaitu tersusun atas rutin-rutin tertentu yang dinamakan dengan fungsi (function) dan fungsi-fungsi tersebut dapat digunakan kembali untuk pembuatan program-program lainnya tanpa harus menulis ulang implementasinya.
- e. Bahasa C merupakan bahasa tingkat menengah (middle level language) sehingga mudah untuk melakukan interface (pembuatan program antar muka) ke perangkat keras.
- f. Struktur penulisan program dalam bahasa C harus memiliki fungsi utama, yang bernama main(). Fungsi inilah yang akan dipanggil pertama kali pada saat proses eksekusi program. Artinya apabila kita mempunyai fungsi lain selain fungsi utama, maka fungsi lain tersebut baru akan dipanggil pada saat digunakan.

Oleh karena itu bahasa C merupakan bahasa prosedural yang menerapkan konsep runtutan (program dieksekusi per baris dari atas ke bawah secara berurutan), maka apabila kita menuliskan fungsi-fungsi lain tersebut dibawah fungsi utama, maka kita harus menuliskan bagian prototipe (prototype), hal ini dimaksudkan untuk mengenalkan terlebih dahulu kepada kompilator daftar fungsi yang akan digunakan di dalam program. Namun apabila kita menuliskan fungsi-fungsi lain tersebut diatas atau sebelum fungsi utama, maka kita tidak perlu lagi untuk menuliskan bagian prototipe diatas.

Selain itu juga dalam bahasa C kita akan mengenal file header, biasa ditulis dengan ekstensi h(*.h), adalah file bantuan yang digunakan untuk menyimpan daftar-daftar fungsi yang akan digunakan dalam program. Bagi anda yang sebelumnya pernah mempelajari bahasa pascal, file header ini serupa dengan

unit. Dalam bahasa C, file header standar yang untuk proses input/output adalah `<stdio.h>`.

Perlu sekali untuk diperhatikan bahwa apabila kita menggunakan file header yang telah disediakan oleh kompilator, maka kita harus menuliskannya didalam tanda '<' dan '>' (misalnya `<stdio.h>`). Namun apabila menggunakan file header yang kita buat sendiri, maka file tersebut ditulis diantara tanda “ dan ” (misalnya “cobaheader.h”). perbedaan antara keduanya terletak pada saat pencerian file tersebut. Apabila kita menggunakan tanda `<>`, maka file tersebut dianggap berada pada direktori default yang telah ditentukan oleh kompilator. Sedangkan apabila kita menggunakan tanda “”, maka file header dapat kita dapat tentukan sendiri lokasinya.

File header yang akan kita gunakan harus kita daftarkan dengan menggunakan directive `#include`. Directive `#include` ini berfungsi untuk memberi tahu kepada kompilator bahwa program yang kita buat akan menggunakan file-file yang didaftarkan. Berikut ini contoh penggunaan directive `#include`.

```
#include<stdio.h>
#include<stdlib.h>
#include"myheader.h"
```

Setiap kita akan menggunakan fungsi tertentu yang disimpan dalam sebuah file header, maka kita juga harus mendaftarkan file headernya dengan menggunakan directive `#include`. Sebagai contoh, kita akan menggunakan fungsi `getch()` dalam program, maka kita harus mendaftarkan file header `<conio.h>`.

2.3.7 Reset Otomatis (Software)

Dari pada mengharuskan sebuah penekanan fisik dari tombol reset sebelum sebuah penguploadan, Arduino Uno didesain pada sebuah cara yang memungkinkannya untuk direset dengan software yang sedang berjalan pada komputer yang sedang terhubung. Salah satu garis kontrol aliran hardware (DTR) dari ATmega8U2/16U2 dihubungkan ke garis reset dari ATmega328 melalui sebuah kapasitor 100 nanofarad. Ketika saluran ini dipaksakan (diambil rendah), garis reset jatuh cukup panjang untuk mereset chip. Software Arduino menggunakan kemampuan ini untuk memungkinkan kita untuk mengupload kode

dengan mudah menekan tombol upload di software Arduino. Ini berarti bahwa bootloader dapat mempunyai sebuah batas waktu yang lebih singkat, sebagai penurunan dari DTR yang dapat menjadi koordinasi yang baik.

Pengaturan ini mempunyai implikasi. Ketika Arduino Uno dihubungkan ke sebuah komputer lain yang sedang running menggunakan OS Mac X atau Linux, Arduino Uno mereset setiap kali sebuah koneksi dibuat dari software (melalui USB). Untuk berikutnya, setengah-detik atau lebih, bootloader sedang berjalan pada Arduino UNO. Ketika Arduino UNO diprogram untuk mengabaikan data yang cacat/salah (contohnya apa saja selain sebuah penguploadan kode baru) untuk menahan beberapa bit pertama dari data yang dikirim ke board setelah sebuah koneksi dibuka.

Jika sebuah sketch sedang berjalan pada board menerima satu kali konfigurasi atau data lain ketika sketch pertama mulai, memastikan bahwa software yang berkomunikasi menunggu satu detik setelah membuka koneksi dan sebelum mengirim data ini. Arduino Uno berisikan sebuah jejak yang dapat dihapus untuk mencegah reset otomatis. Pad pada salah satu sisi dari jejak dapat disolder bersama untuk mengaktifkan kembali. Pad itu diberi label "RESET-RN" Kita juga dapat menonaktifkan reset otomatis dengan menghubungkan sebuah resistor 110 ohm dari tegangan 5V ke garis reset.

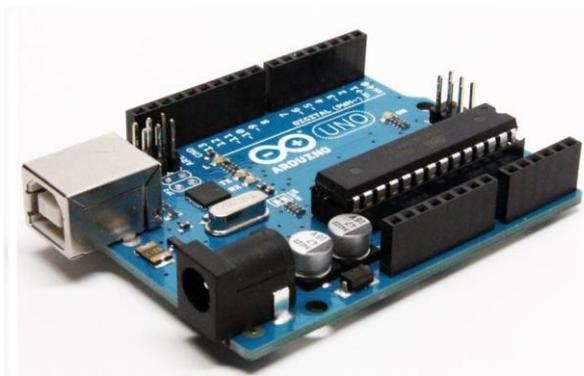
2.3.8 Proteksi Arus lebih USB

Arduino UNO mempunyai sebuah sekering reset yang memproteksi port USB komputer dari hubungan pendek dan arus lebih. Walaupun sebagian besar komputer menyediakan proteksi internal sendiri, sekering menyediakan sebuah proteksi tambahan. Jika lebih dari 500 mA diterima port USB, sekering secara otomatis akan memutuskan koneksi sampai hubungan pendek atau kelebihan beban hilang.

2.3.9 Karakteristik Fisik

Panjang dan lebar maksimum dari PCB Arduino UNO masing-masingnya adalah 2.7 dan 2.1 inci, dengan konektor USB dan power jack yang memperluas dimensinya. Empat lubang sekrup memungkinkan board untuk dipasangkan ke sebuah permukaan atau kotak. Sebagai catatan, bahwa jarak antara pin digital 7

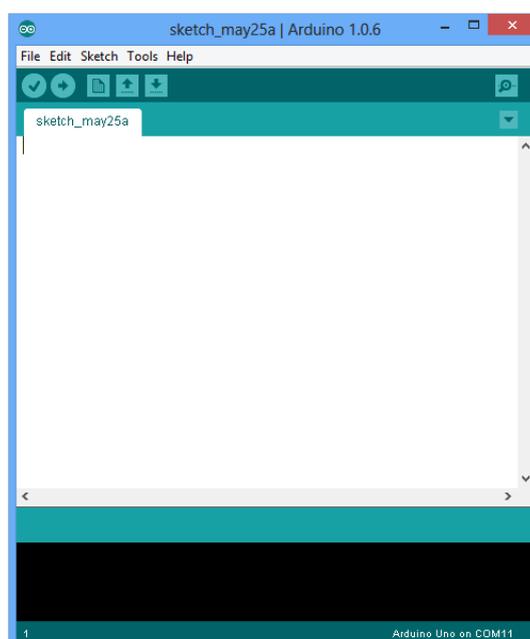
dan 8 adalah 160 mil. (0.16"), bukan sebuah kelipatan genap dari jarak 100 mil dari pin lainnya.



Gambar 2.12 Board Arduino Uno
(Febrianto, 2014)

2.3.10 Arduino *Integrated Development Environment* (IDE)

IDE (*Integrated Development Environment*) adalah sebuah perangkat lunak yang digunakan untuk mengembangkan aplikasi mikrokontroler mulai dari menuliskan *source* program, kompilasi, *upload* hasil kompilasi dan uji coba secara terminal serial. IDE arduino dapat dilihat pada gambar 2.11.



Gambar 2.13 IDE Arduino
(Febrianto, 2014)

- a. Icon menu *verify* yang bergambar ceklis berfungsi untuk mengecek program yang ditulis apakah ada yang salah atau error.
- b. Icon menu upload yang bergambar panah ke arah kanan berfungsi untuk memuat / transfer program yang dibuat di software arduino ke hardware arduino.
- c. Icon menu *New* yang bergambar sehelai kertas berfungsi untuk membuat halaman baru dalam pemrograman.
- d. Icon menu *Open* yang bergambar panah ke arah atas berfungsi untuk membuka program yang disimpan atau membuka program yang sudah dibuat dari pabrikan software arduino.
- e. Icon menu *Save* yang bergambar panah ke arah bawah berfungsi untuk menyimpan program yang telah dibuat atau dimodifikasi.
- f. Icon menu serial monitor yang bergambar kaca pembesar berfungsi untuk mengirim atau menampilkan serial komunikasi data saat dikirim dari hardware arduino.

2.4 Modul (GSM) SIM900A

Modul GSM merupakan perangkat modul yang berfungsi sebagai media komunikasi antara mikrokontroler dengan handphone/mobile *device* yang bekerja pada sistem komunikasi GSM. Modul GSM ini dapat berkomunikasi dan beroperasi dengan mikrokontroler menggunakan perintah ATCommand (*Attention Command*), ATCommand adalah perintah yang dapat diberikan pada modem GSM/CDMA seperti untuk mengirim dan menerima data berbasis GSM/GPRS, atau mengirim dan menerima SMS, maupun perintah lainnya. Dapat dilihat pada tabel 2.3. merupakan contoh dari beberapa perintah ATCommand.

Tabel 2.3 Contoh beberapa perintah ATCommand

AT	Mengecek apakah handphone telah terhubung
AT+CSCS	Menetapkan jenis encoding
AT+CMGR	Membaca pesan
AT+CMGS	Mengirim pesan
AT+CMGD	Menghapus pesan

ATD	Untuk melakukan panggilan
ATH Contoh penggunaan	Untuk menghentikan panggilan ATD08129886435;
AT+CUSD=1 Dan contoh penggunaan	Menjalankan perintah USSD. Misalnya untuk memeriksa jumlah pulsa dan masa aktif kartu SIM. AT+CUSD=1,*888# OK
ATV1	Mengatur input dan output berupa naskah
AT+CALM=	Mengecek suara dering saat panggilan masuk “n” adalah angka yang menunjukkan jenis dering 0= dering 1&2= silent/diam
AT^SCID	Mengecek id simcard

SIM900A adalah modul Dual-band GSM/GPRS menggunakan core IC SIM900A, sehingga ukurannya kecil (24mm x 24mm x 3 mm) dan merupakan solusi yang efektif sebagai modul komunikasi.

SIM900A sudah menerapkan antarmuka standar industri dalam menyediakan fitur komunikasi GSM/GPRS 900/1800MHz untuk voice, SMS, Data, dan Fax. Untuk spesifikasi dan fitur lebih lengkapnya bisa dilihat berikut ini.

Spesifikasi SIM900A adalah sebagai berikut :

- a. GPRS multi-slot class 10/8, kecepatan transmisi hingga 85.6 kbps (*downlink*), mendukung PBCCH, PPP *stack*, skema penyandian CS 1,2,3,4
- b. GPRS *mobile station class B*
- c. Memenuhi standar GSM 2/2 +
- d. SMS (*Short Messaging Service*): point-to-point MO & MT, SMS cell broadcast, mendukung format teks dan PDU (*Protocol Data Unit*)
- e. Dapat digunakan untuk mengirim pesan MMS (*Multimedia Messaging Service*)
- f. Mendukung transmisi faksimili (*fax group 3 class 1*)
- g. *Handsfree mode* dengan sirkit reduksi gema (*echo suppression circuit*)

- h. Dimensi: 24 x 24 x 3 mm
- i. Rentang catu daya antara 7 Volt hingga 12 Volt DC
- j. SIM Application Toolkit
- k. Hemat daya, hanya mengkonsumsi arus sebesar 1 mA pada mode tidur (*sleep mode*)
- l. Rentang suhu operasional: -40 °C hingga +85 °C

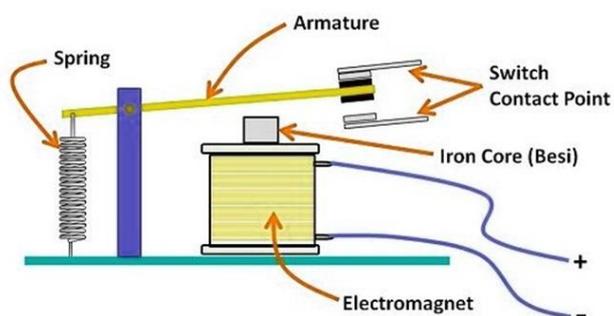


Gambar 2.14 Modul GSM SIM900A
(Radiyah, 2017)

2.5 Relay

Relai merupakan saklar mekanik yang dioperasikan secara elektromagnetik oleh sinyal listrik dengan arus yang kecil untuk menggerakkan saklar penghubung atau pemutus untuk arus beban yang cukup besar. Saklar pada relai akan terjadi perubahan posisi membuka dan atau menutup kontak saklar dalam relai pada saat diberikan energi elektro magnetik.

2.5.1 Cara Kerja Relai



Gambar 2.15 Skema Cara Kerja Relai
(Cahaya Kusuma Ardhi, 2018)

Dari gambar tersebut dapat diketahui bahwa sebuah Besi (Iron Core) yang dililit oleh kumparan Coil, berfungsi untuk mengendalikan Besi tersebut. Apabila Kumparan Coil dialiri arus listrik, maka akan muncul gaya elektromagnetik yang dapat menarik Armature sehingga dapat berpindah dari posisi sebelumnya tertutup (NC) menjadi posisi baru yakni terbuka (NO).

Dalam posisi (NO) saklar dapat menghantarkan arus listrik. Pada saat tidak dialiri arus listrik, Armature akan kembali ke posisi awal (NC). Sedangkan Coil yang digunakan oleh relay untuk menarik Contact Point ke posisi close hanya membutuhkan arus listrik yang relatif cukup kecil.

2.5.2 Modul Relay 1 Channel

Modul relay 1 *channel* merupakan sebuah rangkaian modul relai dengan tegangan masukan 5 Volt dan 1 *channel* keluaran yang memiliki dua buah kontak di dalamnya (1 NO dan 1 NC). Dapat digunakan sebagai saklar elektronik. Untuk mengendalikan perangkat listrik yang membutuhkan daya yang besar. Kompatibel dengan semua mikrokontroler (terutama Arduino, 8051, 8535, AVR, PIC, DSP, ARM, MSP430, TTL logic) maupun Raspberry Pi.

Relai 1 *Channel* ini memerlukan arus minimal 15-20 mA agar dapat mengendalikan. Dengan relai *high-current* sehingga dapat menghubungkan perangkat dengan AC250Volt dan 10Ampere.

Spesifikasi Modul Relai

- a. Jumlah relai : 2
- b. Sinyal pengendali : TTL *level (active low)*
- c. Nilai beban : 10A 250 VAC, 10A 30VDC, 10A 125VAC, 10A 28 VDC
- d. Tunda waktu kontak : 10ms/5ms
- e. Indikator : LED
- f. Ukuran relai : 51 x 41 mm



Gambar 2.16 Bentuk Relay 1 Channel
(Cahya Kusuma Ardhi, 2018)

Pada pembuatan sistem keamanan ini, penulis menggunakan relay Songle SRD-05VDC-SL-C yang mana merupakan relay tipe SPDT (*Single Pole Double Throw*). Gambar 2.6 adalah bentuk fisik dari relay Songle SRD-05VDC-SL-C yang dilengkapi dengan LED SMD yang berfungsi sebagai indikator kerja dari relay tersebut. Pada dasarnya, relay ini menggunakan logika LOW untuk menghubungkan kontak dan mengalirkan listrik.

Kelebihan penggunaan relay ini adalah adanya optocoupler seri 817 yang berfungsi memisahkan hubungan elektris antara mikrokontroler dengan rangkaian relay secara optik. Dengan demikian, apabila terjadi masalah secara elektris pada relay atau perangkat yang dikendalikan, masalah tersebut tidak akan merambat ke rangkaian mikrokontroler atau Arduino yang digunakan.

2.6 Buzzer

Buzzer adalah sebuah komponen elektronika yang berfungsi untuk mengubah getaran listrik menjadi getaran suara. Pada dasarnya prinsip kerja buzzer hampir sama dengan loud speaker, jadi buzzer juga terdiri dari kumparan yang terpasang pada diafragma dan kemudian kumparan tersebut dialiri arus sehingga menjadi elektromagnet, kumparan tadi akan tertarik ke dalam atau keluar, tergantung dari arah arus dan polaritas magnetnya, karena kumparan dipasang pada diafragma maka setiap gerakan kumparan akan menggerakkan diafragma secara bolak-balik sehingga membuat udara bergetar yang akan

menghasilkan suara. Buzzer biasa digunakan sebagai indikator bahwa proses telah selesai atau terjadi suatu kesalahan pada sebuah alat (alarm).



Gambar 2.17 Buzzer (Muhammad Ilham, 2010)

Buzzer elektronik ini dapat diciptakan dengan merangkai beberapa komponen yang pada prinsipnya alat ini dapat menimbulkan pulsa dimana arus listrik adalah sebagai indikator terciptanya pulsa tersebut. Pada pembuatan alat ini, kami menggunakan IC NE555 sebagai sistem pembangkit pulsa yang tentunya arus listrik adalah indikator utama daripada pembangkit sinyalnya. Namun IC NE555 bukanlah satu – satunya komponen yang digunakan. Disini kami menambahkan beberapa komponen pelengkap yang sama bergunanya dalam pembuatan alat ini. Komponen – komponen itu antara lain adalah 1 buah potensiometer 10k, 2 buah kapasitor 0,01uF, 1 buah kapasitor 1uF, 1 buah IC NE555, 1 buah Speaker 8Ω, 0,5watt, 1 buah baterai 9volt, dan 1 buah transistor NPN 9013.