

BAB II LANDASAN TEORI

1.1 Teori Sidik Jari

Sidik jari (*Fingerprint*) merupakan hasil reproduksi tapak jari adapun yang sengaja diambil, dicapkan dengan tinta, maupun kesan yang dibiarkan lepas pada benda karena pernah tersentuh kulit telapak tangan atau kaki. Identifikasi sidik jari, dikenal dengan *daktiloskopi* merupakan ilmu yang mempelajari sidik jari kepada keperluan pengenalan kembali identitas orang dengan cara mengamati garis yang terdapat pada guratan garis jari tangan dan telapak kaki. *Daktiloskopi* bermula dari bahasa Yunani yaitu *dactylos* yang faedahnya jari jemari atau garis jari, dan *scopein* yang faedahnya mengamati atau meneliti. Kesudahan dari pengertian itu timbul istilah dalam bahasa Inggris, *dactyloscopy* yang kita tahu dijadikan ilmu sidik jari.[2]



Gambar 2. 1 Sidik Jari

Sidik jari memiliki sifat sifat *Perennial nature*, yaitu guratan-guratan pada sidik jari yang melekat pada kulit manusia seumur hidup. *Immutability*, yaitu sidik jari seseorang tidak pernah berubah, kecuali mendapatkan kecelakaan yang serius, dan *Individuality*, pola sidik jari adalah unik dan berbeda untuk setiap orang.[1] jadi intinya ialah sidik jari merupakan originalitas pemilliknya yang tidak mungkin sama dengan siapapun didunia ini.

Sistem Henry menggunakan pola *Ridge* (*Ridge* = punggung alur pada kulit, baik pada tangan), yang terpusat pola jari tangan, khususnya telunjuk. 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.

Berikut adalah tabel macam – macam variasi pola *Ridge* :

Tabel 2. 1 Variasi Pola *Ridge*

GAMBAR	NAMA POLA	KETERANGAN
 <p>Ridge</p>	<i>Ridge</i>	Mempunyai ketegasan jarak ganda dari permulaan ke akhir, sebagai lebar <i>Ridges</i> satu dengan lainnya.
 <p>Evading Ends</p>	<i>Evading Ends</i>	Dua <i>ridge</i> dengan arah berbeda berjalan sejajar satu sama lain kurang dari 3 mm.
 <p>Bifurcation</p>	<i>Bifurcation</i>	Dua <i>ridge</i> dengan arah berbeda berjalan sejajar satu sama lain kurang dari 3mm.
 <p>Hook</p>	<i>Hook</i>	<i>Ridges</i> merobek; satu <i>Ridges</i> tidaklah lebih panjang disbanding 3mm
 <p>Fork</p>	<i>Fork</i>	Dua <i>Ridges</i> dihubungkan oleh seper tiga <i>Ridges</i> tidak lebih panjang disbanding 3mm
 <p>Dot</p>	<i>Dot</i>	Bagian <i>Ridges</i> tidak lagi dibanding <i>ridges</i> yang berdekatan
 <p>Eye/Island</p>	<i>Eye</i>	<i>Ridges</i> merobek dan menggabungkan lagi di dalam 3mm
 <p>Eye/Island</p>	<i>Island</i>	<i>Ridges</i> merobek dan tidak bergabung lagi, kurang dari 3mm dan tidak lebih dari 6mm. Area yang terlampir adalah <i>Ridge</i> .

 <p>Enclosed Ridge</p>	<p>Enclosed Ridge</p>	<p><i>Ridges</i> tidak lebih panjang dibanding 6mm antara dua <i>Ridges</i>.</p>
 <p>Enclosed Loop</p>	<p>Enclosed Loop</p>	<p><i>Ridges</i> yang tidak mempola dan menentukan pengulangan antar dua atau lebih <i>Ridges parallel</i></p>
 <p>Specialty</p>	<p>Specialties</p>	<p><i>Rare Ridge</i> membentuk seperti tanda tanya dan sangkutan pemotong</p>

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*. [4]

1.2 Sensor Optical Fingerprint AS608

Sensor *Fingerprint* adalah sebuah perangkat teknologi yang memanfaatkan sidik jari sebagai media identifikasi penggunanya. Cara kerja sensor *Fingerprint* adalah dengan merekam gambar *digital* pada pola sidik jari. Dari hasil rekaman tersebut kemudian dijadikan sebagai sebuah template biometrik yang disimpan dan digunakan untuk pencocokan identitas seseorang atau penggunanya. [5] Sensor *Fingerprint* membaca permukaan jari dan merubah pembacaan analog kedalam *digital* melalui sebuah A/D konverter (*Analog ke Digital*), sebuah modul *interface* bertanggung jawab untuk berkomunikasi (mengirim gambar, menerima perintah, dan sebagainya) dengan alat luar (personal computer / PC). [6]



Gambar 2. 2 Sensor *Fingerprint AS608*

AS608 merupakan salah satu jenis sensor sidik jari dengan verifikasi yang sangat sederhana. Modul sensor AS608 ini memiliki chip DSP bertenaga tinggi untuk melakukan *rendering* gambar, perhitungan, pencarian fitur dan pencarian sidik jari yang tersimpan. Modul sensor ini dapat menyimpan 162 sidik jari yang disimpan dalam memori *flash onboard*. Dalam modul ini juga terdapat LED (*light emitting diode*) biru di lensa yang akan menyala selama sensor tersebut bekerja. Spesifikasi dari Sensor AS608 dapat dilihat pada Tabel berikut[7]:

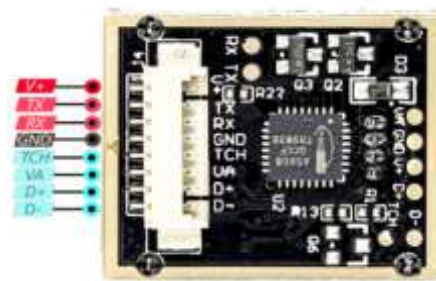
Tabel 2. 2 Spesifikasi Sensor AS608

Nama	Spesifikasi
<i>Voltage supply</i>	3,5 – 6.0VDC
Max operating current	120 mA maks
Peak current	150 mA maks
Max print imaging time	<1,0 s
False accept rate	<0.001%
False reject rate	<1.0%
Intercafe	UART ot TTL serial
Storage capacity	162 <i>fingerprints</i>
Signature file	256 bytes
Template files	512 bytes
Default baud rate	57600
Window area	14mm x 18 mm

Working temperature	-20 to 50 celcius
Working humidity	40% RH-85%RH

Sensor *Fingerprint AS608* ini juga memiliki prinsip kerja yang mencakup dua proses yaitu proses pendaftaran sidik jari dan proses pencocokan sidik jari (dimana pencocokan sidik jari dibagi menjadi perbandingan (1:1) dan pencarian sidik jari (1:N) dua cara).

Dalam mode registrasi, dua gambar sidik jari akan dimasukan kedalam sensor *fingerprint* dan gambar akan diproses dua kali oleh modul. Lalu ketika dalam mode proses pencocokan sensor *fingerprint* digunakan untuk memasukan gambar sidik jari untuk diverifikasi dan diproses lalu dibandingkan kecocokannya dengan data gambar yang sudah ada didalam modul. Jika spesifikasi sidik jari cocok dengan data yang tersimpan maka ini disebut proses *fingerprint comparison mode*, ie, 1:1 mode. Dan jika hasil cocok dengan beberapa sidik jari maka ini dinamakan *fingerprint search mode*, ie 1:N mode, lalu sensor akan mengeluarkan hasil (*pass or fail*).



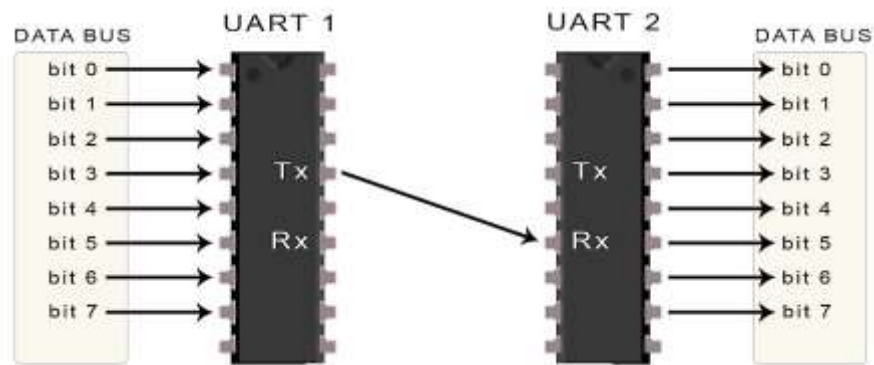
Gambar 2. 3 Pinout *Fingerprint AS608*

Pada sebuah sensor *fingerprint AS608*, sensor ini menggunakan 4 pin utama yang akan terhubung dengan Node MCU ESP32, Pin VCC sebagai *supply* tenaga yaitu sebesar 3v3 VDC, Kemudian GND, Lalu sebagai cara komunikasi sensor ini menggunakan pin RX dan TX. Komunikasi menggunakan RX dan TX disebut juga dengan komunikasi UART.

Universal Asynchronous Receiver Transmitter (UART) adalah metode komunikasi serial yang memungkinkan dua komponen perangkat yang berbeda

untuk berinteraksi satu sama lain tanpa *clock*. UART memiliki cara kerja yang akan mengirimkan data dan akan menerima data dari bus jaringan. Bus data digunakan oleh komputer lain seperti CPU, memori, atau mikrokontroler untuk mengirim data ke UART. Data dilewatkan secara paralel dari bus data ke transmitter UART.

Setelah UART mentransmisikan data paralel dari bus jaringan, paket data dibangkitkan dengan memasukkan bit awal, bit paritas, dan bit stop. Pertama, pin Tx mengeluarkan paket data secara serial, sedikit demi sedikit. UART penerima pada pin Rx-nya membaca paket data sedikit demi sedikit. Kemudian Receiver UART mengubah data kembali menjadi bentuk paralel, menghilangkan bit awal, bit paritas, dan bit stop. Receiver UART akhirnya melewati paket data paralel ke ujung penerima bus data. Berikut ilustrasi bagaimana pin RX dan TX terhubung:



Gambar 2. 4 Pin Rx dan Tx Terhubung

Port UART dapat didasarkan pada perangkat keras atau perangkat lunak. Sebagai contoh, mikrokontroler AT89S52 dan ATMEGA328 Atmel hanya memiliki satu port perangkat keras serial. UART berbasis perangkat lunak diperlukan jika beberapa perangkat perlu dihubungkan melalui UART ke sistem dengan perangkat keras pin UART yang terbatas. Ini juga memberi pengguna fleksibilitas untuk menggunakan pin GPIO sebagai UART jika diperlukan dan menggunakannya di lain waktu untuk tujuan lain. [8]

Sistem Komunikasi UART ini memiliki kelemahan dan kelebihan yaitu:

- Kelebihan : - Tidak ada sinyal *clock*
- Memiliki bit paritas untuk mengoreksi error
 - Struktur paket data dapat dimodifikasi selama kedua sisi dioptimalkan untuk itu.
 - Hanya dua kabel yang digunakan.
- Kekurangan : - Ukuran data frame dibatasi tidak lebih dari 9 bit
- Tidak mendukung banyak master system atau slave
 - Setiap baud rate UART harus berada dalam jarak 10 persen.

1.3 NodeMCU ESP32

NodeMCU adalah sebuah *platform* IoT yang bersifat *opensource*. Terdiri dari perangkat keras berupa *System On Chip* ESP32 buatan *Espressif System*, juga *firmware* yang digunakan, yang menggunakan bahasa pemrograman *scripting Lua*. Istilah NodeMCU secara *default* sebenarnya mengacu pada *firmware* yang digunakan daripada perangkat keras *development kit*[9]. ESP 32 adalah mikrokontroler penerus dari mikrokontroler ESP8266. Untuk tampilan daripada NodeMCU ESP32 dapat dilihat sebagai berikut:



Gambar 2. 5 Mikrokontroler ESP32

Mikrokontroler ESP32 ini dapat diprogram dengan menggunakan C++, C, Python, Lua, dll. untuk menjalankan program mikrokontroler ESP32 ini memerlukan suatu *software* pemrograman, berikut ini adalah contoh *softwrenya* untuk menjalankan program mikrokontroler ESP32, diantaranya sebagai berikut :

1. Arduino Promini.
2. Arduino IDE.
3. Ubuntu 14.04 LTS.
4. ESP-IDF Visual Studio Code Extension.
5. Espressif IoT Development Framework.

Perlu diketahui bahwa ESP32 ini memiliki tegangan operasi 3.3V, berbeda dengan mikrokontroler ATmega pada Arduino Uno, untuk membuat suatu rangkaian elektronik menggunakan ESP32 harus di perhatikan bahwa suplay listrik pada rangkaian tidak boleh lebih dari 5V semisal 9v. Jika tegangan di atas 5v diberikan pada rangkaian yang menggunakan ESP32 tentu akan merusak ESP32 dan membuat rangkaian elektronik tersebut menjadi terbakar. Adapun spesifikasi NodeMCU ESP32 sebagai berikut:

Tabel 2. 3 Spesifikasi NodeMCU ESP32

Spesifikasi	Fitur
<i>MCU</i>	Xtensa Dual-Core 32bit LX6 600DMIPS
802.11 b/g/n <i>Wi-Fi</i>	HT40
<i>Bluetooth</i>	<i>Bluetooth 4.2</i>
<i>Typical Frequency</i>	160Mhz
<i>SRAM</i>	512 <i>kBytes</i>
<i>Flash</i>	SPI
<i>GPIO</i>	36
<i>Hardware/ Software</i> PWM	1/ 16 <i>Channels</i>
SPI/ I2C/ I2S/ UART	4/2/2/2
<i>ADC</i>	12 Bit
<i>CAN</i>	1
<i>Ethernet MAC</i> <i>Interface</i>	1
<i>Touch Sensor</i>	<i>Yes</i>

<i>Temperature Sensor</i>	<i>Yes</i>
<i>Working Temperature</i>	-40°C – 125°C
<i>Current GPIO</i>	12Ma

1.3.1 Memori

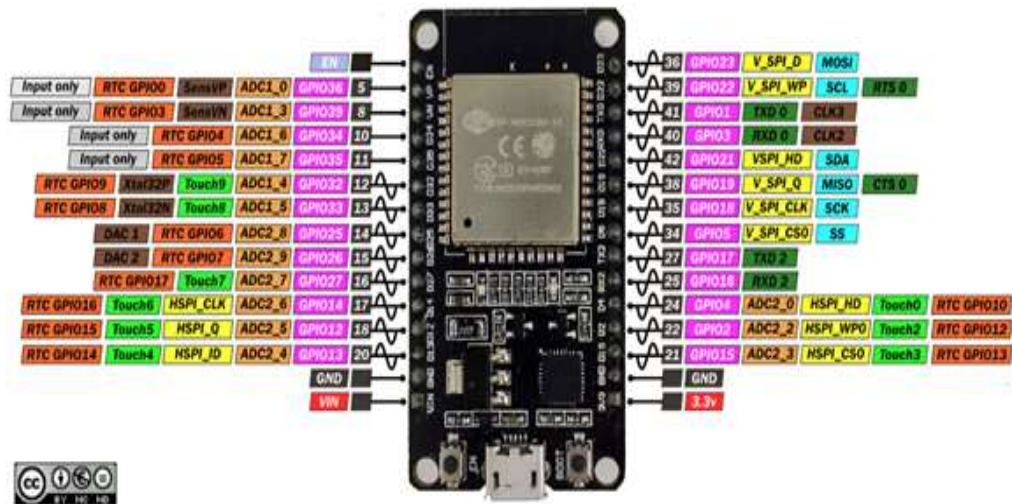
Pada mikrokontroler biasanya terdapat 3 jenis memory yang memiliki fungsi sebagai berikut:

1. *Flash memory* yang berguna sebagai penyimpan data dan program/sketch.
2. SRAM (*Static Random Access Memory*) yang berguna sebagai penyimpanan data variabel sementara.
3. EEPROM (*Electrically Erasable Programmable Read Only Memory*) yang berguna untuk menyimpan data variabel dalam jangka waktu yang lama.

Pada sebuah mikrokontroler ESP32 memiliki *flash memory* sebesar 4 MB untuk menyimpan data kode, lalu SRAM dengan kapasitas 512 KB dan 512 KB EEPROM.[9]

1.3.2 Input & Output

Pada sebuah ESP32 memiliki berbagai macam pin yang berfungsi sebagai berikut:



Gambar 2. 6 PinOut Mikrokontroler ESP32

Untuk fungsi dari pada masing masing pin dapat dikelompokan sebagai berikut :

- **INPUT** : EN, 34, 35, VP(36), VN(39)
- **INPUT/OUTPUT** : 32, 33, 25, 26, 27, 14, 12, 13, 15, 2, 4, RX2(16), TX2(17), 5, 18, 19, 21, RX0, TX0, 22, 23
- **ADC1**: VP(36), VN(39), 34, 35, 32, 33
- **ADC2**: 25, 26, 27, 14, 12, 13, 15, 2, 4
- **DAC**: 25, 26
- **TOUCH 0, 2 – 10**: 4, 2, 15, 13, 12, 14, 27, 33, 32

Perlu juga diperhatikan dalam penggunaan pin masing – masing memiliki fungsinya tersendiri sebagai berikut :

1. Strapping Pin : 2, 4, 5, 12, 15

Strapping pin digunakan untuk memasukan ESP32 ke dalam mode *bootloader* atau mode *flash*. Pada saat melakukan pemrograman, pastikan pin 12 di set dalam keadaan *low* atau 0 pada saat menyalakan ESP32. Akan tetapi, apabila *strapping* pin terhubung dengan beberapa periferal pada saat *upload* program mungkin akan terjadi beberapa kendala. Akan lebih

baik jika menggunakan strapping untuk *input* atau *output*, maka pastikan dalam keadaan *low* atau di set *low*.

2. Pins *High at Boot* : 5, 14, 15

jika ESP32 di reset atau baru di nyalakan, ada beberapa pin yang mengeluarkan logic *HIGH*, dan tentunya jika ada perangkat yang terhubung mungkin dapat mempengaruhi.

3. Pins *High at Boot* : 5, 14, 15

Perlu diketahui bahwa pin pin dengan ADC2 tidak dapat digunakan jika mengaktifkan *Wifi*. Oleh karena itu jika ingin menggunakan *Wifi* maka gunakan pin pin ADC1. Pin ADC ini mempunyai resolusi 12bit, jadi data analog yang akan terbaca bernilai 0 sampai 4095, dimana 0 adalah 0V, dan 4095 adalah 3.3V.

1.4 *Inter Integrated Circuit (I2C)*

I2C merupakan sebuah modul komunikasi serial dua arah yang berfungsi untuk mengirim dan menerima data. Keduanya adalah SCL (*Serial Clock*) dan SDA (*Serial Data*), kedua pin ini tentu sangat menghemat penggunaan pin pada ESP32 jika dibandingkan dengan pemakaian LCD tanpa I2C. Penggunaanya juga mudah dan sudah disediakan *library* yaitu `LiquidCrystal_I2C`

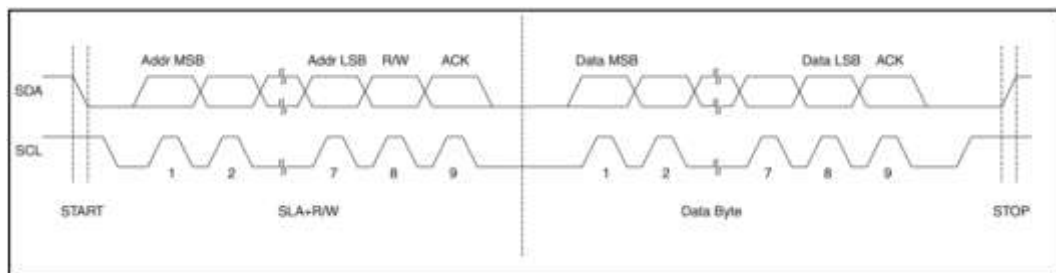
Jenis komunikasi yang dilakukan antar peralatan dengan menggunakan protokol I2C mempunyai sifat serial *synchronous half duplex bidirectional*, dimana yang data ditransmisikan dan diterima hanya melalui satu jalur data *SDA line* (bersifat serial), setiap penggunaan jalur data bergantian antar perangkat (bersifat *half duplex*) dan data dapat ditransmisikan dari dan ke sebuah perangkat (bersifat *bidirectional*).[10] Pada I2C terdapat beberapa pin konfigurasi yang berguna sebagai mana ketentuan berikut :

Pin 1 GND sebagai konektor untuk *Ground*, pin 2 SDA sebagai konektor Serial I2C data, pin 3 SCK sebagai konektor untuk Serial I2C *clock* dan pin 4 VCC sebagai konektor *power 5V DC*.



Gambar 2. 7 Inter Intergrated Circuit (I2C)

Gambar 2 menunjukkan bagaimana satu *frame* tipikal untuk satu sesi transmisi.



Gambar 2. 8 Skema Satu *Frame Transmisi I2C*

Master mengirimkan sinyal *START*, kemudian alamat *slave* yang akan dituju (7 bit), kemudian sinyal R/W yang menandai apakah akan membaca (*HIGH*) atau menulis (*LOW*) ke *slave*. Jika *slave* yang dituju berhasil dicapai, maka *slave* akan memberikan sinyal *ACK* (*acknowledge*). Terhadap adanya sinyal *ACK*, *master* selanjutnya akan mengirim atau menerima data byte (dapat lebih dari satu byte). Pada setiap transmisi data selalu diikuti juga dengan sinyal *ACK*. Jika data sudah komplit ditransmisikan, maka *master* akan menutup sesi transmisi dengan memberikan sinyal *STOP*. [11]

1.5 Modul LCD (*Liquid Crystal Display*)



Gambar 2. 9 *Liquid Crystal Display 20x4*

Display elektronik adalah salah satu komponen elektronika yang berfungsi sebagai tampilan suatu data, baik karakter, huruf ataupun grafik. LCD (*Liquid Cristal Display*) adalah salah satu jenis *display* elektronik yang dibuat dengan teknologi CMOS logic yang bekerja dengan tidak menghasilkan cahaya tetapi memantulkan cahaya yang ada di sekelilingnya terhadap *front-lit* atau mentransmisikan cahaya dari *back-lit*. LCD (*Liquid Cristal Display*) berfungsi sebagai penampil data baik dalam bentuk karakter, huruf, angka ataupun grafik. *Liquid Crystal Display* yang digunakan ini adalah *Liquid Crystal Display 16x4*, artinya LCD terdiri dari 4 baris dan 16 karakter dengan 16 pin konektor.

Pada sebuah LCD terdapat konfigurasi dan deskripsi dari pin-pin LCD antara lain :

1. VSS (Pin 1) : merupakan *power supply* (GND).
2. VCC (Pin2) : merupakan *power supply* (+5V).
3. VEE (Pin 3) : merupakan *input* tegangan kontras LCD.
4. RS *Register Select* (Pin 4) : merupakan register pilihan 0 = *Register Perintah*, 1 = *Register Data*.
5. R/W (Pin 5) : merupakan *read select*, 1 = *Read*, 0 = *Write*.
6. *Enable Clock* LCD (Pin 6) : merupakan masukan logika 1 setiap kali pengiriman atau pembacaan data.
7. D0 sampai D7 (Pin 7 sampai Pin 14) : merupakan data bus 1 sampai 7.

Mikrokontroler pada suatu LCD (*Liquid Cristal Display*) dilengkapi dengan memori dan *register*. Memori yang digunakan mikrokontroler internal LCD adalah:

1. DDRAM (*Display Data Random Access Memory*) merupakan memori tempat karakter yang akan ditampilkan berada.
2. CGRAM (*Character Generator Random Access Memory*) merupakan memori untuk menggambarkan pola sebuah karakter dimana bentuk dari karakter dapat diubah-ubah sesuai dengan keinginan.
3. CGROM (*Character Generator Read Only Memory*) merupakan memori untuk menggambarkan pola sebuah karakter dimana pola tersebut merupakan karakter dasar yang sudah ditentukan secara permanen oleh pabrikan pembuat LCD (*Liquid Cristal Display*) tersebut sehingga pengguna tinggal mengambilnya sesuai alamat memorinya dan tidak dapat merubah karakter dasar yang ada dalam CGROM.[12]

1.6 Keypad

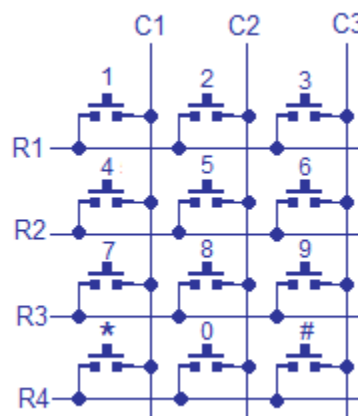
Keypad adalah saklar-saklar *Push button* yang disusun secara matriks yang berfungsi untuk menginput data. *Keypad* berfungsi sebagai *interface* antara perangkat (mesin) elektronik, yaitu mikrokontroler dengan manusia atau dikenal dengan istilah HMI (*Human Machine Interface*). Konfigurasi *Keypad* dengan susunan bentuk matrix ini bertujuan untuk penghematan *port* mikrokontroler karena jumlah *key* (tombol) yang dibutuhkan banyak pada suatu sistem dengan mikrokontroler.[13]



Gambar 2. 10 Keypad Matrix 4x3

Konstruksi matrix *Keypad* 4×3 terdiri dari 4 baris dan 3 kolom dengan *Keypad* berupa saklar push buton yang diletakan disetiap persilangan kolom dan barisnya. Rangkaian matrix *Keypad* terdiri dari 12 saklar push buton dengan konfigurasi 4 baris dan 3 kolom. line yang terdiri dari 4 baris dan 3 kolom tersebut dihubungkan dengan port mikrokontroler. Sisi baris dari *matrix Keypad* ditandai dengan nama *Row1*, *Row2*, *Row3* dan *Row4* kemudian sisi kolom ditandai dengan nama *Col1*, *Col2*, dan *Col3*. Sisi *input* atau *output* dari matrix *Keypad* 4×3 ini tidak mengikat. Kolom dapat dikonfigurasikan sebagai *input* dan baris sebagai *output* ataupun sebaliknya, bergantung pada program yang digunakan.

Berikut adalah konfigurasi dari pada rangkaian dari *Keypad* 4x3 yang dipakai :



Gambar 2. 11 Konfigurasi Keypad Pin Baris & Kolom

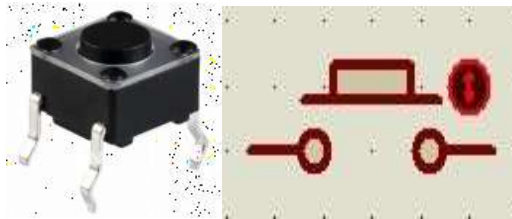
Proses *scanning* untuk membaca penekanan tombol pada *matrix Keypad* 4×3 untuk mikrokontroler dilakukan dengan cara bertahap kolom demi kolom, dari kolom pertama sampai kolom ke 3 dan dari baris pertama hingga baris ke 4. Program *scanning* matrix *Keypad* 4×3 bermacam-macam, meskipun pada intinya sama. Sebagai contoh, kita asumsikan *Keypad* aktif rendah (semua *line* kolom dan baris dipasang *resistor pull-up*) dan dihubungkan ke port mikrokontroler dengan jalur kolom adalah jalur *input* dan jalur *output* maka proses *scanning* matrix *Keypad* 4×3 diatas dapat dituliskan sebagai berikut:

1. Apabila Kolom 1 diberi logika '0', kolom kedua dan kolom ketiga diberi logika '1' maka program akan mengecek tombol 1, 4, 7, dan *, sehingga apabila salah satu baris berlogika '0' maka ada tombol yang ditekan.
2. Apabila Kolom 2 diberi logika '0', kolom pertama dan kolom ketiga diberi logika '1' maka program akan mengecek tombol 2, 5, 8, dan 0, sehingga apabila salah satu baris berlogika '0' maka ada tombol yang ditekan.
3. Apabila Kolom 3 diberi logika '0', kolom pertama dan kolom kedua diberi logika '1' maka program akan mengecek tombol 3, 6, 9, dan #, sehingga apabila salah satu baris berlogika '0' maka ada tombol yang ditekan.
4. Kemudian kembali seperti semula, artinya program *looping* terus mendeteksi data kolom dan data baris, cara ini disebut *scanning* atau pemindaian *Keypad* untuk mendapatkan saklar mana yang ditekan.

1.7 Push button

Push button switch (saklar tombol tekan) adalah perangkat / saklar sederhana yang berfungsi untuk menghubungkan atau memutuskan aliran arus listrik dengan sistem kerja tekan *unlock* (tidak mengunci). Sistem kerja *unlock* disini berarti saklar akan bekerja sebagai *device* penghubung atau pemutus aliran arus listrik saat tombol ditekan, dan saat tombol tidak ditekan (dilepas), maka saklar akan kembali pada kondisi normal.

Sebagai *device* penghubung atau pemutus, *Push button switch* hanya memiliki 2 kondisi, yaitu On dan Off (1 dan 0). Istilah On dan Off ini menjadi sangat penting karena semua perangkat listrik yang memerlukan sumber energi listrik pasti membutuhkan kondisi On dan Off.



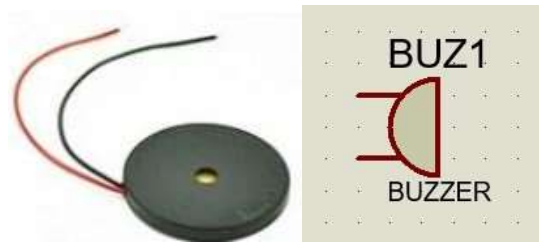
Gambar 2. 12 Push Button

Berdasarkan fungsi kerjanya yang menghubungkan dan memutuskan, *Push button switch* mempunyai 2 tipe kontak yaitu NC (*Normally Close*) dan NO (*Normally Open*).

1. NO (*Normally Open*), merupakan kontak terminal dimana kondisi normalnya terbuka (aliran arus listrik tidak mengalir). Dan ketika tombol saklar ditekan, kontak yang NO ini akan menjadi menutup (*Close*) dan mengalirkan atau menghubungkan arus listrik. Kontak NO digunakan sebagai penghubung atau menyalakan sistem *circuit* (*Push button ON*).
2. NC (*Normally Close*), merupakan kontak terminal dimana kondisi normalnya tertutup (mengalirkan arus listrik). Dan ketika tombol saklar *Push button* ditekan, kontak NC ini akan menjadi membuka (*Open*), sehingga memutus aliran arus listrik. Kontak NC digunakan sebagai pemutus atau mematikan sistem *circuit* (*Push button Off*).[14]

1.8 Modul 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).[15]

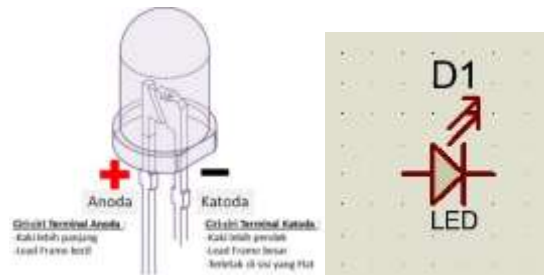


Gambar 2. 13 Buzzer

1.9 LED (*Light Emitting Diode*)

Light Emitting Diode atau biasa disebut dengan LED adalah komponen elektronika yang terbuat dari bahan semikonduktor dan masih termasuk dalam kategori diode. LED mempunyai keistimewaan yaitu dapat memancarkan cahaya seperti lampu. LED strukturnya sama dengan dioda, yaitu menggunakan sambungan P- N untuk menghasilkan emisi cahaya pada semikonduktor, *doping* yang di pakai adalah gallium, arsenic, dan fosforus. Jenis *doping* yang berbeda akan menghasilkan warna yang berbeda. Warna LED yang umum adalah merah, kuning, hijau, biru, dan putih. LED memiliki dua kutub yaitu Anoda dan Katoda. LED akan menyala apabila mendapat *bias forward* atau arus listrik yang mengalir dari Anoda ke Katoda. Dalam rangkaian elektronika, pemasangan kutub LED tidak boleh terbalik, karena apabila terbalik kutubnya maka LED tersebut tidak akan menyala.[16]

Cara kerjanya pun hampir sama dengan Dioda yang memiliki dua kutub yaitu kutub Positif (P) dan Kutub Negatif (N). LED hanya akan memancarkan cahaya apabila dialiri tegangan maju (*bias forward*) dari Anoda menuju ke Katoda. LED terdiri dari sebuah chip semikonduktor yang di doping sehingga menciptakan *junction* P dan N. Yang dimaksud dengan proses doping dalam semikonduktor adalah proses untuk menambahkan ketidak murnian (*impurity*) pada semikonduktor yang murni sehingga menghasilkan karakteristik kelistrikan yang diinginkan.[16]



Gambar 2. 14 *Light Emitting Diode*

Untuk mengetahui polaritas terminal Anoda (+) dan Katoda (-) pada LED. Ciri-ciri Terminal Anoda pada LED adalah kaki yang lebih panjang dan juga *Lead Frame* yang lebih kecil. Sedangkan ciri-ciri Terminal Katoda adalah Kaki yang lebih pendek dengan *Lead Frame* yang besar serta terletak di sisi yang *Flat*. [16]

1.10 Adaptor

Pengertian Adaptor adalah sebuah perangkat elektronik yang berguna untuk dapat mengubah tegangan arus AC (arus bolak-balik) yang tinggi menjadi DC (arus searah) yang rendah, secara prinsip kerja, adaptor ini bisa dikatakan berfungsi sebagai alat catu daya. Adaptor juga sering disebut sebagai pengganti baterai atau aki. Dengan adanya alat tersebut, seluruh perangkat elektronik yang membutuhkan catu daya dapat memanfaatkan adaptor.



Gambar 2. 15 Adaptor USB

Fungsi adaptor adalah sebagai media pengubah tegangan arus listrik tinggi menjadi lebih rendah. Adaptor inilah yang membuat arus tegangan listrik mengalir

sesuai dengan kebutuhan perangkat yang digunakan. Agar dapat melakukan kinerjanya, perangkat tersebut memerlukan komponen – komponen penyusun, yang mana komponen penyusun tersebut ialah bagian dari rangkaian adaptor.



Gambar 2. 16 Bagian Dalam Adaptor

Berikut bagian – bagian apa saja yang menyusun adaptor:

a. *Transformator* (Trafo)

Transformator ini lebih umum disebut sebagai trafo. Dimana trafo ini merupakan salah satu komponen penyusun sebuah adaptor. Trafo berfungsi menurunkan bahkan menaikkan tegangan arus daya sesuai dengan kebutuhan pemakaian.

Pada adaptor, penggunaan trafo yang umum dipakai adalah trafo jenis penurun tegangan (*trafo step down*). meskipun tegangan pada trafo sudah diturunkan. Namun *output* dari trafo masih berbentuk arus AC (arus bolak-balik). Yang mana arus AC ini nantinya akan diproses kembali sesuai dengan kebutuhan.

b. *Rectifier* (Penyearah)

Rectifier lebih umum disebut sebagai penyearah gelombang. Pada rangkaian adaptor arus yang masuk ke dalam perangkat merupakan jenis arus bolak-balik. Agar dapat berfungsi pada peralatan elektronik maka perlu adanya pengubah arus menjadi searah terlebih dahulu atau bisa disebut pengubah arus AC menjadi DC.

Dengan demikian, pada bagian *rectifier* inilah arus AC tersebut diolah dan diubah menjadi arus searah. Di dalam *rectifier*, terdapat juga komponen dioda. Yang mana bagian-bagiannya adalah sebagai berikut :

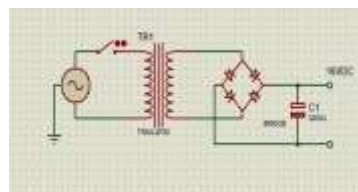
- *Half wave rectifier*, di dalamnya terdapat dioda 1 yang berfungsi sebagai penyearah.
- *Full wave rectifier*, di dalamnya terdapat 2 atau 4 perangkat dioda yang fungsinya penyearah.

c. *Filter* (Penyaring)

Filter atau penyaring adalah komponen penyusun dari adaptor yang sama pentingnya. *Filter* ini berfungsi sebagai penyaring sinyal dari *rectifier*. Komponen yang terdapat didalam filter antara lain adalah kondensator yang berjenis ELCO (*Electrolyte Capacitor*).

d. *Voltage regulator* (Pengatur Tegangan)

Bagian penyusun adaptor yang terakhir ialah *voltage regulator*. Umumnya disebut juga sebagai alat pengatur tegangan. *Voltage regulator* ini berfungsi menstabilkan tegangan arus searah serta melakukan kontrol pada tegangan *output*-nya. Tujuannya yaitu agar tegangan tidak terpengaruh oleh arus beban, suhu, ataupun tegangan *input* yang asalnya dari *output* filter. Di dalam *voltage regulator*, terdapat beberapa komponen. Seperti diantaranya transistor, dioda zener serta IC regulator.[17]

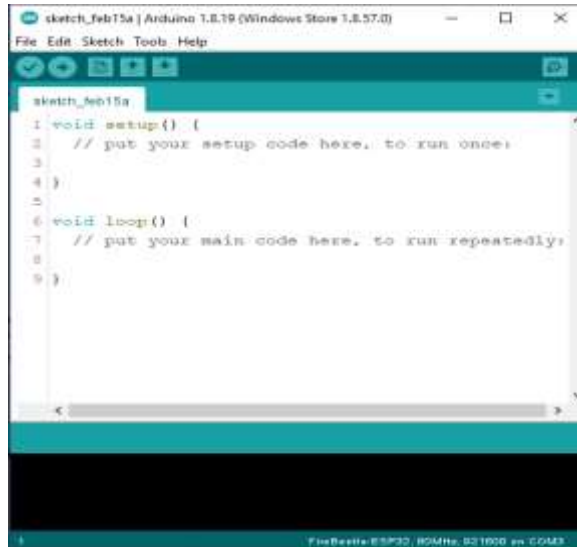


Gambar 2. 17 Skema Rangkaian Adaptor Sederhana

1.11 *Software* Arduino IDE

Arduino IDE adalah *Software* yang digunakan untuk membuat *sketch* pemrograman atau dengan kata lain arduino IDE sebagai media untuk pemrograman pada *board* yang ingin diprogram. Arduino IDE ini berguna untuk mengedit, membuat, meng-*upload* ke board yang ditentukan, dan meng-*coding* program tertentu. Arduino IDE dibuat dari bahasa pemrograman JAVA, yang dilengkapi

dengan *library* C/C++(*Wiring*), yang membuat operasi *input/output* lebih mudah.[18]



Gambar 2. 18 Software Arduino IDE

Setiap program arduino (biasa disebut *sketch*) mempunyai dua buah fungsi yang harus ada dalam setiap program yaitu :

1. *Void setup (){}*

Void setup merupakan fungsi yang hanya menjalankan program yang ada didalam kurung kurawal sebanyak 1 kali.

2. *Void loop (){}*

Fungsi ini akan dijalankan setelah setup (fungsi *void setup*) selesai, setelah dijalankan 1 kali, fungsi ini akan dijalankan lagi dan lagi secara terus menerus sampai catu daya (*power*) dilepaskan.

Arduino IDE juga memiliki syintak dalam penulisan koding seperti berikut:

1. *//* (komentar 1 baris)

Digunakan untuk memberi komentar atau catatan pada kode-kode yang dibuat.

2. */* */* (komentar 2 baris)

Untuk menuliskan catatan pada beberapa baris sebagai komentar.

3. *{ }* (kurung kurawal)

Digunakan untuk mendefinisikan kapan blok program mulai dan berakhir serta digunakan juga pada fungsi dan pengulangan.

4. ; (titik koma)

Setiap baris kode harus diakhiri dengan tanda ; (titik koma), jika ada titik koma yang hilang maka program tidak akan berjalan.

1.12 *Software Bot dalam Aplikasi Telegram*

Telegram adalah sebuah sistem perpesanan yang lintas *platform* dan berpusat pada keamanan kerahasiaan pribadi penggunaannya, sedangkan *bot* adalah program komputer yang melakukan pekerjaan tertentu secara otomatis. *Bot* adalah sebuah mesin, dibuat memudahkan kehidupan keseharian kita tanpa harus terpaku di depan komputer. kelebihan dari Telegram ini adalah adanya landasan untuk menggunakan *Application Programming Interface (API)* untuk masyarakat luas. Salah satu API yang disediakan adalah fitur *bot*. *Bot* Telegram adalah *bot* yang saat ini mulai populer dipergunakan.

Didalam telegram tersedia 2 API yang pertama adalah *API clien*, yang kedua adalah *API bot*, untuk mendapatkan *API client* seorang pengguna telegram hanya perlu *login* ke <https://my.telegram.org/auth>, didalam *my.telegram* seorang pengguna tinggal membuat *API client*, Api jenis kedua adalah *API bot* yang cara membuatnya dengan masuk kedalam *BotFather* terlebih dahulu, baik *API client* ataupun *api bot* memiliki fungsi dan kerja yang sama artinya kedua-keduanya dapat mengirim pesan, video, lokasi, dan dokumen.[19] Adapun cara untuk membuat *bot* pada telegram sebagai berikut: