

APLIKASI SIDIK JARI SEBAGAI PROTEKSI KENDARAAN BERMOTOR BERBASIS MIKROKONTROLLER AVR ATMEGA16

NIKO FAJAR PUTRANTO NIM: 121642006

Jurusan Teknik Elektro, Fakultas Teknologi Industri
Institut Sains & Teknologi AKRIND Yogyakarta
Jln. Kalisahak No. 28 Komplek Balapan Tromol Pos 45 Yogyakarta

Perkembangan teknologi semakin hari akan mengalami peningkatan baik dari segi sistem dan dari segi pengaplikasian penggunaan teknologi tersebut. Salah satu dari berbagai teknologi yang sedang banyak digunakan saat ini adalah sidik jari. Pengaplikasian sidik jari ini membutuhkan rancang bangun arsitektur yang baik karena penggunaan sidik jari ini untuk meningkatkan keamanan dari kasus pencurian. Pada intinya, sidik jari merupakan teknologi yang berfungsi untuk menangkap gambar digital dari pola sidik jari. Dalam hal ini akan menyajikan bagaimana pengaplikasian sidik jari sebagai *proteksi* pada kendaraan bermotor sehingga menambah keamanan pada kendaraan tersebut.

Kata kunci: Sidik Jari, Tag, AVR ATMega 16.

The development of the technology today will increase in terms of both the system and the application in terms of the use of these technology. One of the various technologies that are widely used today is fingerprint. Fingerprint application requires good architectural design because of the use fingerprint to improve the security of the theft. In essence, fingerprint is a technology which use to capture a digital image of the fingerprint pattern. In this case will present how the security bike lock using a Fingerprint.

Keyword: fingerprint, Sensor, AVR ATMega 16.

I PENDAHULUAN

1.1 Latar Belakang

Semakin pesat perkembangan teknologi searah dengan tingginya tingkat kriminalitas. Tingkat kriminalitas di Indonesia mengalami peningkatan dari tahun ke tahun. Jenis kejahatan yang ditemukan juga semakin bertambah, dari pembunuhan, perampokan dan pencurian. Fenomena tersebut tidak diimbangi dengan banyaknya kasus yang berhasil dipecahkan. Sindikat pelaku tindak kriminal juga semakin mahir dalam melakukan aksi kejahatannya, mereka terspesialisasi pada tindak kriminalitas yang spesifik, semisal spesialis pembunuhan, spesialis pencurian, dan lain sebagainya.

Tingkat pencurian kendaraan bermotor masih sering terjadi. Pencurian terjadi karena masih kurangnya sistem keamanan yang terdapat pada kendaraan bermotor, karena hanya menggunakan kunci kontak. Untuk mengatasi masalah pencurian tersebut salah satunya dengan memberikan sistem pengaman ganda pada sepeda motor (Utomo, 2007).

2. Landasan Teori

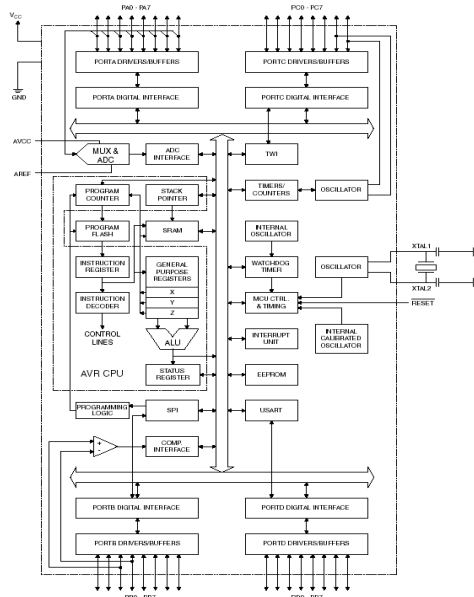
2.1 Mikrokontroler ATMega16

Mikrokontroler adalah sebuah *chip* IC yang berfungsi sebagai pengontrol rangkaian elektronik dan umumnya dapat menyimpan program di dalamnya serta dapat diprogram ulang (*reprogrammable*) (Agfianto, 2004). IC ini dapat diumpamakan sebagai bentuk minimum dari sebuah mikrokomputer yang tersusun dari perangkat keras (*hardware*) dan perangkat lunak (*software*).

Mikrokontroler adalah suatu *chip* dengan kepadatan yang sangat tinggi, dimana semua bagian yang diperlukan untuk suatu kontroler sudah dikemas dalam satu *keping*, biasanya terdiri dari *Central Processsing Unit* (CPU), *Random Access Memory* (RAM), *EEPROM/EPROM/PROM/ROM*, *I/O*, *Timer* dan lain sebagainya. Rata-rata Mikrokontroler memiliki instruksi manipulasi bit, akses ke I/O secara langsung dan mudah, dan proses interupsi yang cepat dan efisien.

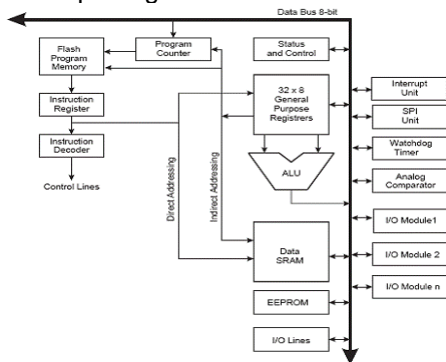
Mikrokontroler sekarang ini sudah banyak dapat kita temui dalam berbagai peralatan elektronik, misalnya peralatan yang terdapat di rumah, seperti telepon digital, *microwave oven*, televisi, dan masih banyak lagi. Mikrokontroler juga dapat kita gunakan

untuk berbagai aplikasi misalnya untuk pengendalian suatu alat, otomasi dalam industri dan lain-lain. Keuntungan menggunakan mikrokontroler adalah harganya murah, dapat diprogram berulang kali, dan dapat diprogram sesuai ditujukan untuk pengukuran cerdas dan aplikasi cerdas.



Gambar 2.1 Diagram blok Arsitektur Mikrokontroler ATmega 16

Arsitektur Mikrokontroler AtMega 16 dapat dilihat pada gambar 2.2



Gambar 2.1 Arsitektur Mikrokontroler ATmega 16

ATMega16 menggunakan arsitektur Harvard dengan memisahkan antara memori dan bus untuk program dan data untuk memaksimalkan kemampuan dan kecepatan. Instruksi dalam memori program dieksekusi dengan *pipelining single level* dimana ketika satu instruksi dieksekusi, instruksi berikutnya diambil dari memori program. Konsep ini mengakibatkan instruksi dieksekusi setiap siklus *clock*. CPU terdiri dari 32x8 bit *general purpose register* yang dapat diakses dengan cepat dalam satu siklus *clock*, yang mengakibatkan operasi *Arithmetic Logic Unit*

(ALU) dapat dilakukan dalam satu siklus. Pada operasi ALU, dua *operand* berasal dari *register*, kemudian operasi dieksekusi dan hasilnya disimpan kembali pada *register* dalam satu siklus *clock*. Operasi aritmatik dan logika pada ALU akan mengubah bit yang terdapat pada *Status Register* (SREG).

2.2 Pemrograman Arduino

Setiap variabel dalam bascom memiliki tipe data yang menunjukkan daya tampungnya. Hal ini berhubungan dengan penggunaan memori mikrokontroler. Berikut adalah tipe data pada arduino.

Arduino dirilis oleh Massimo Banchi dan David Cuartielles pada tahun 2005. Arduino merupakan sistem mikrokontroler yang relatif mudah dan cepat dalam membuat aplikasi elektronika ataupun robotika. Arduino terdiri dari perangkat elektronik atau papan elektronik *open source* yang didalamnya terdapat komponen utama yang berupa *chip* mikrokontroler. Dengan jenis AVR dari perusahaan Atmel. Arduino memiliki program *software* dan *hardware*.



Gambar 2.3 logo Arduino

2.2.1 Hardware

Arduino saat ini telah menggunakan seri chip megaAVR, khususnya ATmega 8, ATmega 168, ATmega 328, ATmega 1280, ATmega 2560. Kebanyakan papan arduino memiliki regulator linier 5 vakt dan 16 MHz osilator kristal (resonator keramik).

Tabel 2.1 Jenis Arduino

No	Jenis Arduino	Processor	Frkuensi (MHz)	Digital IO pins	Analog Input pins	PWM pin
1	ADK	ATmega 2560	16	54	16	14
2	Bluetooth	ATmega 328	16	14	6	4
3	Diecimila	ATmega 168	16	14	6	4
4	Due	AT91SAM	84	54	12	12
5	Duemilan ova	ATmega 168/328 P	16	14	6	6

6	Ethernet	ATmega 328	16	14	6	3
7	Fio	ATmega 328P	8	14	8	6
8	Leonardo	ATmega 32u4	16	14	12	6
9	LilyPad	ATmega 168V atau ATmega 328	8	14	6	6

2.2.2 Software

Arduino diciptakan untuk para pemula bahkan yang tidak memiliki dasar bahasa pemrograman sama sekali menggunakan bahasa C++ yang telah dipermudah melalui *library*. Arduino menggunakan *processing* yang digunakan untuk menulis program kedalam Arduino. *Processing* sendiri merupakan penggabungan antara bahasa C++ dan *Java*. *Software* Arduino ini dapat di-*install* di berbagai *operating system* (OS) seperti, *LINUX*, *Mac OS*, *Windows*, *Software ID*.

Arduino memiliki 3 bagian

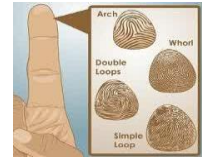
- Editor program untuk menulis dan mengedit program.
- Compiler adalah modul untuk mengubah kode program ke dalam kode biner
- Uploader modul yang berfungsi untuk memasukkan kode biner kedalam mikrokontroler

2.3 Fingerprint

Sidik jari (bahasa Inggris: fingerprint) adalah hasil reproduksi tapak jari baik yang sengaja diambil, dicapkan dengan tinta, maupun bekas yang ditinggalkan pada benda karena pernah tersentuh kulit telapak tangan atau kaki. Kulit telapak adalah kulit pada bagian telapak tangan mulai dari pangkal pergelangan sampai kesemua ujung jari, dan kulit bagian dari telapak kaki mulai dari tumit sampai ke ujung jari yang mana pada daerah tersebut terdapat garis halus menonjol yang keluar satu sama lain yang dipisahkan oleh celah atau alur yang membentuk struktur tertentu. Identifikasi sidik jari, dikenal dengan daktiloskopi adalah ilmu yang mempelajari sidik jari untuk keperluan pengenalan kembali identitas orang dengan cara mengamati garis yang terdapat pada guratan garis jari tangan dan telapak kaki. Daktiloskopi berasal dari bahasa Yunani yaitu *dactylos* yang berarti jari jemari atau garis jari, dan *scopein* yang artinya mengamati atau meneliti. Kemudian dari pengertian itu timbul istilah dalam bahasa Inggris, *dactyloscopy* yang kita kenal menjadi ilmu sidik jari.

Pola sidik jari selalu ada dalam setiap tangan dan bersifat permanen. Dalam artian,

dari bayi hingga dewasa pola itu tidak akan berubah sebagaimana garis tangan. Setiap jari pun memiliki pola sidik jari berbeda. Ada empat pola dasar *Dermatoglyphic* tentang sidik jari yang perlu diketahui, yakni *Whorl* atau *Swirl*, *Arch*, *Loop*, dan *Triradius*. Selain itu hanyalah variasi dari kombinasi keempat pola ini.

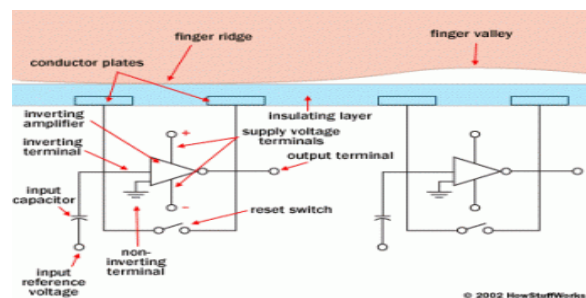


Gambar 2.4 pola Sidik Jari

Seperti scanner optik, scanner kapasitif memproses gambar bukit dan lembah yang membentuk sidik jari seseorang. Tetapi, yang membedakan antara scanner optik adalah, scanner kapasitif mendeteksi gambaran sidik jari menggunakan arus listrik.

Untuk memindai jari, prosesor pemindai membaca voltase output dan menentukan karakteristik bukit dan lembah pada sidik jari. Dengan membaca setiap sel pada kumpulan sensor, prosesor dapat mengumpulkan gambar sidik jari secara keseluruhan.

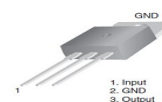
Kelebihan yang mendasar pada scanner kapasitif adalah hanya bisa mendeteksi sidik jari asli, ketimbang mendeteksi pola terang dan gelap yang menjadi prinsip scanner optik. Hal ini akan membuat sistem menjadi lebih sulit untuk ditipu. Selain itu, scanner kapasitif biasanya lebih kompak ukurannya dibanding scanner optik.



Gambar 2.5 Sistem kerja Sidik Jari (<http://arnoldjulians.wordpress.com>)

Komponen – komponen pendukung lain

- Ic Regulator LM 7805 berfungsi untuk menghasilkan tegangan output 5V



Gambar 2.6 IC LM 78xx
 b. Buzzer berfungsi untuk mengubah tegangan listrik menjadi suara.



Gambar 2.7 Buzzer

c. Relay berfungsi untuk memutuskan dan mengalirkan arus listrik.



Gambar 2.8 Relay

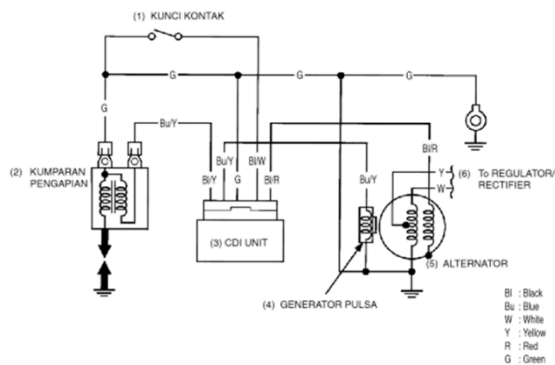
2.4 Sistem pengapian Kendaraan bermotor

Bagian Sepeda Motor yang tak kalah pentingnya dalam memberikan kenyamanan saat berkendara adalah Instalasi kelistrikan yang baik, selain Mesin dan Bodi/ sasis serta penampilan

Bagian kelistrikan pada motor sendiri terdiri dari 2 jalur, yaitu:

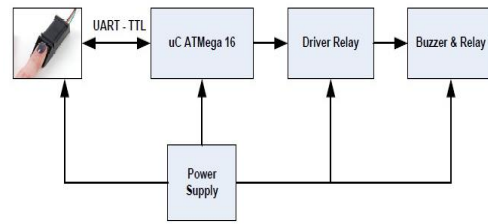
Jalur kelistrikan pengapian

1. Jalur kelistrikan pengapian, yaitu jalur kelistrikan dari spul hingga ke busi yang berfungsi mensuplai Arus listrik yang di gunakan untuk proses pembakaran pada mesin.
2. Jalur kelistrikan *regulator* (DC) yang di gunakan untuk mensuplai arus listrik instrumen kendaraan seperti: klakson, lampu utama, lampu *sein* dan lain sebagainya.



Gambar 2.9 Diagram sistem pengapian (agungsevi.files.wordpress.com)

3. Perancangan Alat 3.1 Diagram Perancangan



Gambar 3.1 Diagram Alir alat

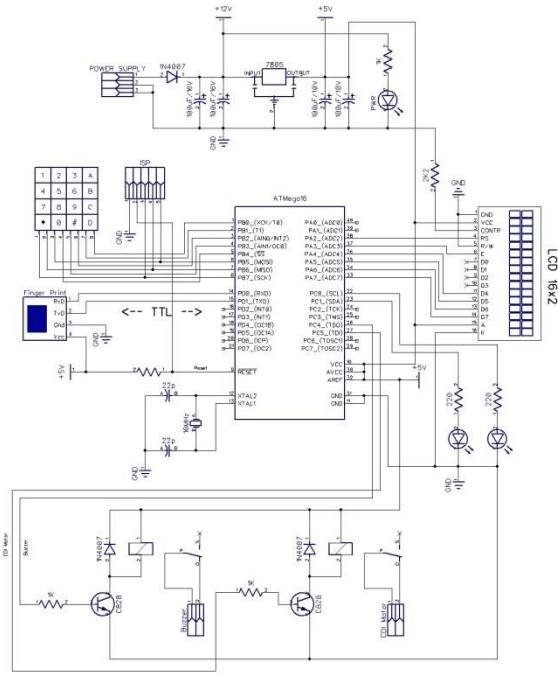
Sistem pengaman kendaraan bermotor dengan menggunakan Sidik Jari (*Fingerprint*) ini akan dipasang pada kendaraan bermotor. Alat ini berfungsi sebagai pengaman pertama pada kendaraan bermotor, karena sebelum mengaktifkan mesin kendaraan bermotor harus mengaktifkan Sidik Jari. Dengan demikian diharapkan keamanan kendaraan bermotor akan lebih terjamin.

Prinsip kerja dari alat pengaman kendaraan bermotor ini dengan memanfaatkan Sidik Jari dan *password* sebagai kunci pertama sebelum kendaraan bermotor diaktifkan. Dengan demikian sebelum mesin diaktifkan maka user diharuskan menempelkan sidik jari yang telah daftar. Tetapi ketika user mengaktifkan Sidik Jari, maka motor akan memberikan peringatan sirine sebagai tanda bahaya.

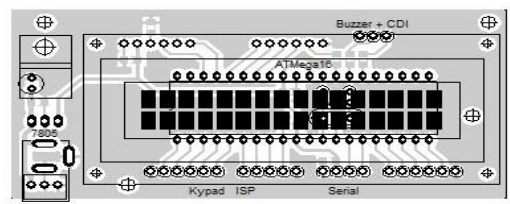
41

Pada alat pengaman kendaraan bermotor ini menggunakan mikrokontroler ATmega16 sebagai pengendali utama (*central processing unit*) dan Sidik Jari *reader* (pembaca *tag* Sidik Jari) sebagai masukan. Sedangkan keluaran atau *output* sistem yaitu *relay* sebagai pengaman kunci.

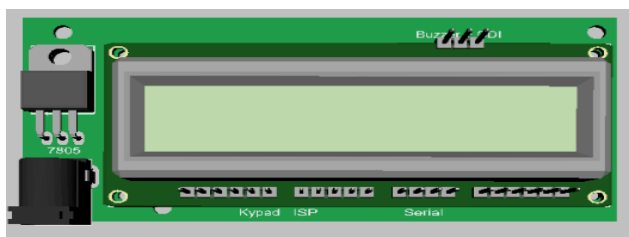
3.2 Rangkaian Lengkap Alat



Gambar 3.4 Skematik alat Lengkap



Gambar 3.3 Board Rangkaian Lengkap alat



Gambar 3.4 Rangkaian Mikro

3.3 Identifikasi Kebutuhan

Dalam perancangan sistem pengaman kendaraan bermotor menggunakan Sidik Jari berbasis Mikrokontroler ATmega 16 dibutuhkan *hardware* dan *software*. Perangkat keras yang dibutuhkan meliputi alat dan bahan, sedang perangkat lunak yang dibutuhkan adalah sebuah *software downloader* ATmega16. Alat yang dibutuhkan antara lain:

- Personal computer (PC)
- Downloader ATmega16
- Power supply
- Multimeter
- Solder
- Solder attractor

• Toolset

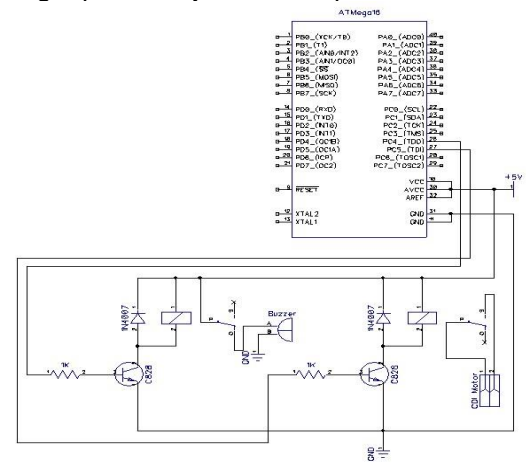
Sedangkan bahan yang digunakan untuk membuat pengaman kendaraan bermotor menggunakan Sidik jari adalah:

- Sistem minimum ATmega16
- Driver Relay
- Power supply 12 Volt

3.4 Perancangan Perangkat Keras

3.4.1 Sistem minimum mikrokontroler

Rangkaian mikrokontroler menggunakan sistem minimum, yaitu sebuah *Crystal* 16 MHz dan dua buah kapasitor sebesar 22pf. Pemakaian *osilator* kristal 16 MHz dimaksudkan agar eksekusi mikrocontroller ke finger print menjadi lebih cepat



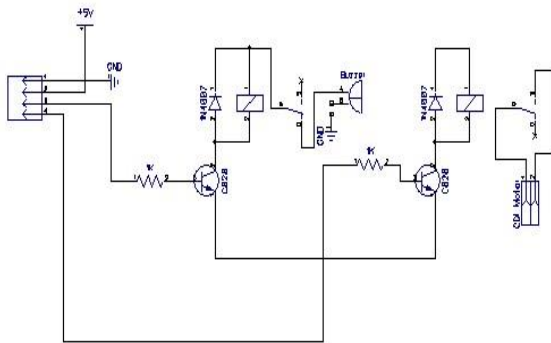
Gambar 3.5 Sistem minimum Mikrokontroler

3.4.2 Driver Relay

Dalam rangkaian *driver relay* ini terdiri dari dua buah rangkaian *switching transistor* yaitu rangkaian saklar 1 (*pin PC5*), rangkaian saklar 2 (*pin PC4*). Rangkaian saklar 1 berfungsi untuk mengendalikan arus yang mengalir ke *buzzer*. Rangkaian saklar 2 berfungsi untuk mengendalikan coil (CDI).. Rangkaian dan komponen pada saklar 1 dan 2 adalah sama, tetapi untuk masing-masing *port* pengendali ke mikrokontroler dan kondisi *relay* difungsikan berbeda. Saklar 1 terhubung ke *pin 27 (PC5)* dan bekerja *Normally Close*, saklar 2 terhubung ke *pin 26 (PC4)* dan bekerja *Normally Open*. Untuk mengontrol rangkaian saklar ini maka mikrokontroler harus mengirimkan sinyal pulsa 1 atau 0. Jika mikrokontroler memberikan data sinyal pulsa 0 maka saklar tidak aktif, dan sebaliknya jika mikrokontroler memberikan data sinyal pulsa 1 maka saklar akan aktif.

Rangkaian saklar 1, 2, dan *relay* indikator kontak terdiri dari komponen resistor, transistor C828, dioda 1N4007 dan relay DC 5 Volt. Resistor pada basis berfungsi untuk membatasi arus yang akan masuk ke

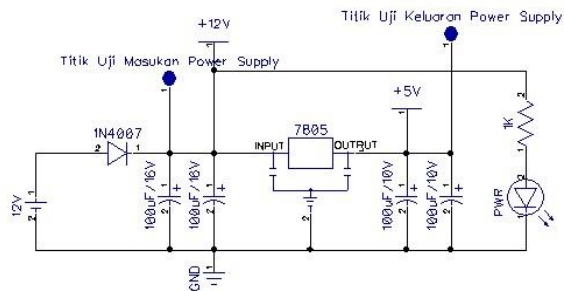
transistor, sedangkan dioda 1N4007 berfungsi untuk menahan tegangan balik dari *relay* pada saat perubahan kondisi dari kondisi aktif ke kondisi tidak aktif. Saat transistor C828 berada pada kondisi jenuh, tegangan pada kolektor-emitor (Vce) mendekati nol. Jika transistor dalam keadaan jenuh, arus akan mengalir menuju *relay* yang menyebabkan saklar akan tertutup dan alat akan terhubung. Untuk rangkaian *driver relay* dapat dilihat pada gambar dibawah ini.



Gambar 3.6 Driver Relay

Gambar 3.4.4 Power Supply

Rangkaian pengamanan kendaraan bermotor ini membutuhkan tegangan 5 volt DC. Rangkaian *regulator* ini berfungsi untuk menurunkan tegangan *accu* 12 volt ke 5 volt. Terdapat IC yang berfungsi untuk menurunkan tegangan dengan keluaran 5 volt, yaitu IC LM 7805, merupakan *regulator* DC yang cukup stabil. Untuk meratakan tegangan *output* dari LM 7805 maka perlu ditambahkan kapasitor *elektrolit* sebesar 100 μ F/16 volt.



Gambar 3.7 Power Supply

Pada rangkaian di atas, dioda 1N407 berfungsi sebagai pengamanan, yaitu untuk menghindari kerusakan pada LM 7805 jika polaritasnya terbalik. Dengan pemasangan dioda, ketika terbalik pemasangan antara VCC dengan GND maka dioda tidak mengalirkan arus, sehingga akan menyelamatkan LM 7805, karena LM 7805 sangat sensitive dengan polaritas.

Rangkaian *power supply* ini tersusun atas beberapa komponen antara lain:

a. Dioda

Dioda merupakan suatu komponen yang berfungsi untuk menyearahkan gelombang AC menjadi gelombang DC yang masih kasar, selain itu juga sebagai pengamanan tegangan.

b. Kapasitor

Kapasitor yang digunakan adalah kapasitor jenis elektrolit, yang berfungsi sebagai perata tegangan setelah disearahkan menggunakan dioda.

c. IC LM 7805

IC LM 7805 ini berfungsi untuk membatasi tegangan agar *output* yang keluar maksimal 5 Volt DC

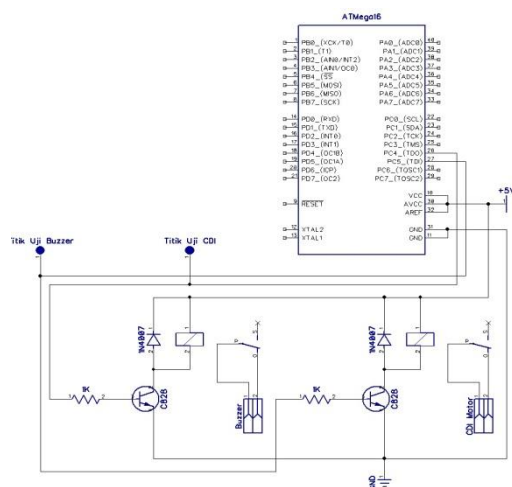
3.5 Komponen Alat.

Tabel 3.1 Komponen Alat

No	Modul	Komponen	Spesifikasi
1	Power Supply	Dioda	1N4007
		Ic Regulator	LM 7805 1A
		Kapasitor	100 μ F / 25 Volt
2	Controller	Kristal	16 MHz
		Kapasitor keramik	22pF, 100nF
		Resistor	5x1K ohm
		AVR ATmega 16	40 pin
		LED	3mm
		Dioda	1N4007
3	Input	Resistor	560 ohm
		Fingerpirnt	ZFM 200series
		Push Button	Normally open
4	Output	Relay	5Volt
		Buzzer	3x5V
		Led	DC6v

4. Pengujian

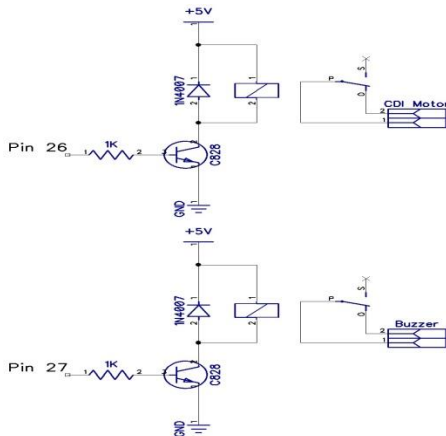
4.1 Pengujian Mikrokontroller



Gambar 4.1 Pengujian Mikrokontroller

No	Titik pengujian ke-	Mikrokontroler (port uC)	Fungsi Port	Tegangan	
				Non Aktif	Aktif
1	1	27	Relay Buzzer	0	4.9
2	2	26	Relay CDI	0	4.9

4.2 Pengujian Driver Relay



Gambar 4.2 Driver Relay

Tabel 4.2 Pengujian relay

No	Masukan		Keluaran	
	Pin 27	Pin 26	Relay Buzzer	Relay CDI
1	0	0	Mati	Mati
2	1	0	Hidup	Mati

4.3 Pengujian Alat

Tabel 4.2 Pengujian Alat

No	Masukan		Keluaran		
	Sidik Jari	Pass word	Relay Buzzer	Relay CDI	Ket
1	0	0	Mati	TA	Mesin off
2	0	1	Hidup	TA	Mesin off tunggu Fingerprint
3	1	0	Mati	TA	Motor off
4	1	1	Mati	A	Motor On

Berdasarkan Tabel 4.4 pengujian diatas ada beberapa kondisi yaitu :

- Saat Sidik Jari yang diregistrasi ditempelkan pada sensor fingerprint maka *buzzer* dan *relay* kontak tidak aktif.
- Saat password diaktifkan dan Sidik Jari tidak ditempelkan maka *relay* CDI tidak menyambung sehingga motor tidak dapat dijalankan.
- Saat Sidik Jari didekatkan dan password tidak sesuai maka *relay* CDI tidak tersambung sehingga motor tidak dapat dijalankan, dan buzzer berbunyi.

- Saat Sidik Jari didekatkan kemudian password sama maka *relay* CDI tersambung sehingga motor dapat dijalankan.

4.4. Software (Skrip)

Perangkat lunak yang digunakan untuk pada sistem *processing* data dan mengendalikan *output* menggunakan *software* Arduino. Untuk lebih jelasnya dapat dilihat pada potongan program di bawah ini:

a. Skrip Sidik Jari

```
#include <SoftwareSerial.h>
#include <Adafruit_Fingerprint.h>
#include <LiquidCrystal.h>
#include "MenuBackend.h"

uint8_t getFingerprintEnroll(uint8_t id);

SoftwareSerial mySerial(A5,A4);

//=====================================================
LiquidCrystal lcd(5, 4, 3, 2, 1, 0);
const int BackLight = 6;

const int buttonPinLeft = A0;
const int buttonPinRight = A1;
const int buttonPinEnter = A2;
const int buttonPinEsc = A3;

int lastButtonPushed = 0;
int lastButtonEnterState = LOW;
int lastButtonEscState = LOW;
int lastButtonLeftState = LOW;
int lastButtonRightState = LOW;

long lastEnterDebounceTime = 0;
long lastEscDebounceTime = 0;
long lastLeftDebounceTime = 0;
long lastRightDebounceTime = 0;
long debounceDelay = 200;

boolean Menu = false;
boolean Mark = false;

Adafruit_Fingerprint finger =
Adafruit_Fingerprint(&mySerial);
//=====================================================
char mode1[] PROGMEM = "[ Setting Time
]";
char mode11 [] PROGMEM = "[ Hour
]";
char mode12 [] PROGMEM = "[ Min
]";
char mode13 [] PROGMEM = "[ Day
]";
char mode14 [] PROGMEM = "[ Date
]";
char mode15 [] PROGMEM = "[ Month
]";
char mode16 [] PROGMEM = "[ Year
]";
char mode2 [] PROGMEM = "[ LCD ]";
char mode3 [] PROGMEM = "[ Value ]";
```

```

MenuBackend menu =
MenuBackend(menuUseEvent,menuChangeEvent);
  Menuitem M1= Menuitem(mode1);
  Menuitem M11 = Menuitem(mode11);
  Menuitem M12 = Menuitem(mode12);
  Menuitem M13 = Menuitem(mode13);
  Menuitem M14 = Menuitem(mode14);
  Menuitem M15 = Menuitem(mode15);
  Menuitem M16 = Menuitem(mode16);
  Menuitem M2 = Menuitem(mode2);
  Menuitem M3 = Menuitem(mode3);
//=====
void menuSetup()
{
  menu.getRoot().add(M3);

M3.addRight(M1).addRight(M2).addRight(M3);

M1.add(M11).addRight(M12).addRight(M13).addRight(M14).addRight(M15).addRight(M16).addRight(M11);
}
inline const __FlashStringHelper *
fromFlash(const char * pStr)
{ return reinterpret_cast <const __FlashStringHelper *> (pStr); }
//=====
void menuUseEvent(MenuUseEvent used)
{
//=====
}
void
menuChangeEvent(MenuChangeEvent changed)
{
  Menuitem newMenuitem = changed.to;

  lcd.clear();
  lcd.print(F(" Menu's "));
  lcd.setCursor(0,1);

  lcd.print(fromFlash(changed.to.getName()));
};

if (newMenuitem.getName() ==
menu.getRoot())
{
  Menu = false;
  lcd.clear();
}
else
  Menu = true;
}

//=====
void setup()
{
  digitalWrite(BackLight,HIGH);
  //Serial.begin(9600);

  menuSetup();

  pinMode(buttonPinLeft,INPUT);
  pinMode(buttonPinRight,INPUT);
  pinMode(buttonPinEnter,INPUT);

  pinMode(buttonPinEsc,INPUT);

  lcd.begin(16,2);
  lcd.setCursor(0,0);
  lcd.print("Fingerprint Test");
  pinMode(BackLight,OUTPUT);
  // put your setup code here, to run once:
  finger.begin(57600);

  if (finger.verifyPassword())
  {
    lcd.clear();
    lcd.setCursor(0,1);
    lcd.print("Found Sensor");
  }
  else
  {
    lcd.clear();
    lcd.setCursor(0,1);
    lcd.print("Didn't find");
    while(1);
  }
}
//=====
void NavigateMenu()
{
  Menuitem currentMenu =
menu.getCurrent();
  if (lastButtonPushed != 0)
  {
    switch (lastButtonPushed)
    {
      case buttonPinRight :
        menu.moveRight();break;
      case buttonPinLeft :
        menu.moveLeft();break;
      case buttonPinEsc :
        menu.toRoot();break;
      // if ((currentMenu.moveUp()) &&
Mark == true) menu.moveUp();
      // else menu.getRoot();break;
      case buttonPinEnter :
        if ((currentMenu.moveDown()))
        menu.moveDown();
        else menu.use();break;
    }
  }
}
//=====
void loop()
{
  ReadButtons();
  NavigateMenu();
  if (Menu == false)
  {
    digitalWrite(BackLight,HIGH);
    lcd.setCursor(0,0);
    lcd.print("test lcd");
    // put your main code here, to run
repeatedly:
  }
}
//=====
void ReadButtons()
{
  int reading;
  int buttonEnterState = LOW;
  int buttonEscState = LOW;

```



```

int buttonLeftState = LOW;
int buttonRightState = LOW;

//enter
reading = digitalRead(buttonPinEnter);
if (reading != lastButtonEnterState)
{
    lastEnterDebounceTime = millis();
}
if ((millis() - lastEnterDebounceTime) >
debounceDelay)
{
    buttonEnterState = reading;
    lastEnterDebounceTime = millis();
}
lastButtonEnterState = reading;

//esc
reading = digitalRead(buttonPinEsc);
if (reading != lastButtonEscState)
{
    lastEscDebounceTime = millis();
}
if ((millis() - lastEscDebounceTime) >
debounceDelay)
{
    buttonEscState = reading;
    lastEscDebounceTime = millis();
}
lastButtonEscState = reading;

//right
reading = digitalRead(buttonPinRight);
if (reading != lastButtonRightState)
{
    lastRightDebounceTime = millis();
}
if ((millis() - lastRightDebounceTime) >
debounceDelay)
{
    buttonRightState = reading;
    lastRightDebounceTime = millis();
}
lastButtonRightState = reading;

//left
reading = digitalRead(buttonPinLeft);
if (reading != lastButtonLeftState)
{
    lastLeftDebounceTime = millis();
}
if ((millis() - lastLeftDebounceTime) >
debounceDelay)
{
    buttonLeftState = reading;
    lastLeftDebounceTime = millis();
}
lastButtonLeftState = reading;

//if pressed
if (buttonEnterState == HIGH)
{ lastButtonPushed = buttonPinEnter; }
else if (buttonEscState == HIGH)
{ lastButtonPushed = buttonPinEsc; }
else if (buttonRightState == HIGH)
{ lastButtonPushed = buttonPinRight; }
else if (buttonLeftState == HIGH)
{ lastButtonPushed = buttonPinLeft; }
else
{ lastButtonPushed = 0; }
}

}

//=====
//=====
a. Keypad Skrip

#include <Keypad.h>
#include <SoftwareSerial.h>
#include <Adafruit_Fingerprint.h>
#include <LiquidCrystal.h>

#define LOCKED 2
#define PASSWORD_OK 1
#define UNLOCKED 0

int LockState = LOCKED;
long StartTime = 0;
int pos = 0;

char* secretCode="1234";

const byte ROWS = 4; //four rows
const byte COLS = 4; //three columns
char keys[ROWS][COLS] = {
    {'1','4','7','*'},
    {'2','5','8','0'},
    {'3','6','9','#'},
    {'A','B','C','D'}
};
byte rowPins[ROWS] = {9,8, 7, 6};
//connect to the row pinouts of the
keypad
byte colPins[COLS] = {13,12, 11, 10};
//connect to the column pinouts of the
keypad

Keypad keypad = Keypad(
makeKeymap(keys), rowPins, colPins,
ROWS, COLS );

int buzz = A2;
int load1 = A0;
int load2 = A1;

int getFingerprintIDez();
int printID = -1;

SoftwareSerial mySerial(A5,A4);
Adafruit_Fingerprint
finger=Adafruit_Fingerprint(&mySerial);

LiquidCrystal lcd(5, 4, 3, 2, 1, 0);
//=====
//=====
void setup()
{
    // put your setup code here, to run once:
    lcd.begin(16,2);

    pinMode(buzz,OUTPUT);
    pinMode(load1,OUTPUT);
    pinMode(load2,OUTPUT);

    // Serial.begin(9600);
    finger.begin(57600);
    /*
    for (int i = 0; i < 5; i++)

```



```

    lcd.setCursor(0,1);
    lcd.print("LOCKED");
    digitalWrite(load2,HIGH);
    //digitalWrite(load1,LOW);
}
else if(state==PASSWORD_OK)
{
// Serial.println("Password_OK");
// digitalWrite(Led,LOW);
lcd.clear();
lcd.setCursor(0,5);
lcd.print("PASS OK");
digitalWrite(load2,LOW);
// digitalWrite(load1,HIGH);
}
else if (state == UNLOCKED)
{
// Serial.println("Unlocked");
// digitalWrite(Led,LOW);
lcd.clear();
lcd.setCursor(0,1);
lcd.print("UNLOCKED");
//digitalWrite(load1,LOW);
digitalWrite(load2,LOW);
}
}

```

Program diatas merupakan program utama aplikasi Sidik Jari sebagai *starter key* elektrik digital berbasis Mikrokontroler AVR Atmega 16. Pada program utama, semua proses dieksekusi, mulai dari program identifikasi, *scanning*, data password sampai keluaran sistem.

4 PENUTUP

5.1 Kesimpulan

Berdasarkan pelaksanaan dan Setelah mengamati dan membahas pengaplikasian Sidik Jari (*Fingerprint*) Sebagai *Starter Key* Elektrik Digital Berbasis Mikrokontroler AVR ATmega 16, maka dapat diambil kesimpulan yaitu:

- Dua pengamanan dalam sidik jari ini, yaitu dengan keypad dan tanpa keypad, untuk pengamanan tingkat tinggi menggunakan keypad.
- Saat pengenalan match maka mesin hidup, saat tidak match mesin diam.
- Output* dari rangkaian *power supply* dalam alat ini sekitar 4,9 Vdc saat diberi *input* 6-12 Vdc.

5.2 Saran

Saran yang dapat penulis berikan untuk pengembangan alat adalah sebagai berikut:

- Untuk mendapatkan hasil yang terbaik dalam pembuatan alat, usahakan menggunakan sistem yang sederhana

dalam rangkaianannya namun memiliki kemampuan atau kualitas maksimal dalam pemilihan.

- Dalam alat ini *buzzer* akan mati bila kunci kontak dimatikan pada saat Sidik Jari tidak diinputkan. Untuk pengembangan lebih lanjut dari alat ini, maka sebaiknya *buzzer* akan tetap menyala sebelum menekan tombol khusus.

Untuk menambah fungsi dari alat ini, maka dapat dikembangkan lagi adanya sistem *monitoring* bagi yang menggunakan kendaraan

DAFTAR PUSTAKA

- Agfianto, 2003, *Belajar Mikrokontroler AT89C51/52/55*. Yogyakarta: Penerbit Gava Media.
- Blocher, R., 2002, *Dasar Elektronika*. Yogyakarta: Penerbit Andi.
- Yuan, A., 2011, *Implementasi Kunci Elektronik Berbasis Mikrokontroler Sebagai Pengaman Tambahan Pada Mobil*. <http://goes-open.blogspot.com>, 2012. *Akses internal EEPROM dengan ATmega 32*. <http://nonot.in>, 2011, *Tutorial Microsoft Word*. <http://www.adafruit.com/datasheets/ZFM%20user%20manualV15.pdf>
- Daftar pustaka sidik jari
- http://elib.unikom.ac.id/files/disk1/528/jbptunikompp-gdl-susanagust-26395-4-unikom_s-i.pdf
- <http://absensi-sidikjari-yogyakarta.blogspot.com/2013/04/4-pola-dasar-sidik-jari-dan-kepribadian.html>
- <http://repository.usu.ac.id/bitstream/123456789/28677/4/Chapter%2011.pdf>
- <http://cnt121.files.wordpress.com/2007/11/kapasitor.pdf>
- <http://elektronika-dasar.web.id/teori-elektronika/definisi-kapasitor/>
- http://eprints.undip.ac.id/41662/8/BAB_2.pdf
- www.circuittoday.com?working-of-fingerprint-scanner
- www.computer.howstuffworks.com/fingerprint-scanner