

DAFTAR ISI

DAFTAR ISI	i
DAFTAR GAMBAR	iii
LABORATORIUM ELEKTRONIKA INDUSTRI	iv
MODUL I	5
1.1 Pendahuluan	5
1.2 Tujuan	5
1.3 Alat dan Bahan	5
1.4 Dasar Teori	5
1.4.1 General Input Output	5
1.4.2 Arduino	5
1.5 Prosedur Percobaan	7
1. Percobaan 1 ON- OFF LED	7
2. Percobaan 2 Menampilkan Karakter Pada 7-Segment	10
3. Percobaan 3 Menampilkan input keypad pada LCD	12
MODUL II	15
2.1 Pendahuluan	15
2.2 Tujuan	15
2.3 Alat dan Bahan	15
2.4 Dasar Teori	15
2.4.1 Pulse Width Modulation	15
2.4.2 Analog to Digital Converter	16
2.4.3 Arduino	17
2.5 Prosedur Percobaan	18
1. Percobaan 1 Menghidupkan Motor DC	18
2. Percobaan 2 Menampilkan LM35 ke LCD	21
3. Percobaan 3 Mengatur Kecepatan Putar Motor DC	24



MODUL III	28
3.1 Pendahuluan	28
3.2 Tujuan	28
3.3 Alat dan Bahan	28
3.4 Dasar Teori	28
3.4.1 Universal Asynchronous Receiver Transmitter (UART)	28
3.4.2 Serial Peripheral Interface (SPI)	29
3.4.3 Inter Integrated Circuit (I2C)	29
3.4.4 Arduino	30
3.5 Prosedur Percobaan	32
1. Percobaan 1 Komunikasi UART	32
2. Percobaan 2 Komunikasi SPI	34
3. Percobaan 3 Komunikasi I2C	38
MODUL IV	42
4.1 Pendahuluan	42
4.2 Tujuan	42
4.3 Alat dan Bahan	42
4.4 Ketentuan Praktikum	42
4.5 Laporan Praktikum	43



DAFTAR GAMBAR

Gambar 1. Arduino Uno	6
Gambar 2. LED	7
Gambar 3. Rangkaian Percobaan ON-OFF LED	8
Gambar 4. Flowchart Rangkaian Percobaan ON-OFF LED	9
Gambar 5. Seven Segment	10
Gambar 6. Rangkaian Percobaan 7 Segment	10
Gambar 7. Flowchart Rangkaian Percobaan 7 Segment	11
Gambar 8. Keypad	12
Gambar 9. LCD Character	12
Gambar 10. Rangkaian Percobaan Keypad dan LCD	12
Gambar 11. Flowchart Rangkaian Percobaan Keypad dan LCD	14
Gambar 12. Duty Cycle	15
Gambar 13. Siklus Sinyal PWM pada Arduino	16
Gambar 14. Arduino Uno	17
Gambar 15. Motor DC	19
Gambar 16. Driver Motor	19
Gambar 17. Rangkaian Percobaan Motor DC	19
Gambar 18. Flowchart Rangkaian Percobaan Motor DC	21
Gambar 19. Sensor LM35	22
Gambar 20. LCD	22
Gambar 21. Rangkaian Percobaan LM35 dan LCD	22
Gambar 22. Flowchart Rangkaian Percobaan LM35 dan LCD	24
Gambar 23. Motor DC	24
Gambar 24. Potensiometer	25
Gambar 25. Rangkaian Percobaan Mengatur Laju Motor DC	25
Gambar 26. Flowchart Rangkaian Percobaan Mengatur Laju Motor DC	27
Gambar 27. Cara Kerja Komunikasi UART	28
Gambar 28. Cara Kerja Komunikasi SPI	29
Gambar 29. Cara Kerja Komunikasi I2C	30
Gambar 30. Arduino Uno	30
Gambar 31. LED	32
Gambar 32. Push Button	32
Gambar 33. Rangkaian Percobaan Komunikasi UART	32
Gambar 34. Flowchart Rangkaian Percobaan Komunikasi UART	34
Gambar 35. LED	35
Gambar 36. Push Button	35
Gambar 37. Rangkaian Percobaan Komunikasi SPI	35
Gambar 38. Flowchart Rangkaian Percobaan Komunikasi SPI	38
Gambar 39. LED	38
Gambar 40. Potensiometer	39
Gambar 41. Rangkaian Percobaan Komunikasi I2C	39



Gambar 42. Flowchart Rangkaian Percobaan Komunikasi I2C



LABORATORIUM ELEKTRONIKA INDUSTRI

Kepala Laboratorium : Dr. Eng. Muhammad Ilhamdi Rusydi

Sekretaris Laboratorium : Dr. Ir. Darwison, M.T.

Koordinator Asisten : M. Arif Rahman

Koordinator Praktikum : Saidina Usman

Asisten :

1. Farhatul Fityah
2. Niqma Rozalia
3. Nurul Hadian
4. Riko Candra
5. Yoan Winata
6. Indah Irdyana Yemensia
7. M. Arif Rahman
8. M. Farhan
9. Muhammad Dani Anwar
10. Saidina Usman
11. Yolanda Farmayeni



MODUL I

GENERAL INPUT DAN OUTPUT

1.1 Pendahuluan

- a) Asistensi dilakukan 3x dengan lama pertemuan 20 menit (Rabu, Kamis, Jumat)
- b) Praktikum dilakukan 1x dengan lama pertemuan 90 menit (Selasa)
- c) Laporan akhir dikumpulkan pada hari Kamis

1.2 Tujuan

- a) Memahami cara penggunaan input dan output digital pada mikrokontroler
- b) Menggunakan komponen input dan output sederhana dengan Arduino

1.3 Alat dan Bahan

- a) Modul Arduino
- b) Keypad
- c) 7-Segment

1.4 Dasar Teori

1.4.1 General Input Output

Input adalah semua data dan perintah yang dimasukkan ke dalam memori untuk diproses lebih lanjut oleh mikroprosesor. Sebuah perangkat input adalah komponen piranti keras yang memungkinkan user atau pengguna memasukkan data ke dalam mikroprosesor. Output adalah data hasil yang telah diproses. Perangkat output adalah semua komponen piranti keras yang menyampaikan informasi kepada orang-orang yang menggunakannya.

Pada Arduino, pin input/output terdiri dari digital dan analog yang jumlah pin-nya tergantung jenis Arduino yang digunakan. Input digital digunakan untuk mendeteksi perubahan logika biner pada pin tertentu. Adanya input digital memungkinkan mikrokontroler untuk dapat menerjemahkan 0V menjadi logika LOW dan 5V menjadi logika HIGH. Membaca sinyal digital pada Arduino dapat menggunakan sintaks `digitalRead(pin);`

Output digital terdiri dari dua buah logika, yaitu kondisi logika HIGH dan kondisi logika LOW. Untuk menghasilkan output kita dapat menggunakan sintaks `digitalWrite(pin,nilai);` yang sebelumnya pin sudah diset ke mode OUTPUT, lalu parameter kedua adalah set nilai HIGH atau LOW. Apabila pin diset dengan nilai HIGH, maka voltase pin tersebut akan diset ke 5V atau 3.3V dan bila pin diset ke LOW, maka voltase pin tersebut akan diset ke 0V.

1.4.2 Arduino

Arduino adalah kit elektronik atau papan rangkaian elektronik open source yang di dalamnya terdapat komponen utama yaitu sebuah chip mikrokontroler dengan jenis AVR dari perusahaan Atmel. Arduino yang kita gunakan dalam praktikum ini adalah Arduino Uno yang

menggunakan chip AVR ATmega 328P. Dalam memprogram Arduino, kita bisa menggunakan komunikasi serial agar Arduino dapat berhubungan dengan komputer ataupun perangkat lain. Adapun spesifikasi dari Arduino Uno ini adalah sebagai berikut :



Gambar 1. Arduino Uno

Microcontroller	ATmega328P
Operating Voltage	5 V
Input Voltage (recommended)	7 – 12 V
Input Voltage (limit)	6 – 20 V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

A. BAGIAN-BAGIAN ARDUINO UNO

1. POWER USB

Digunakan untuk menghubungkan Papan Arduino dengan komputer lewat koneksi USB.

2. POWER JACK

Supply atau sumber listrik untuk Arduino dengan tipe Jack. Input DC 5 - 12 V.

3. Crystal Oscillator

Kristal ini digunakan sebagai layaknya detak jantung pada Arduino. Jumlah cetak menunjukkan 16000 atau 16000 kHz, atau 16 MHz.

4. Reset

Digunakan untuk mengulang program Arduino dari awal atau Reset.



5. Digital Pins I / O

Papan Arduino UNO memiliki 14 Digital Pin. Berfungsi untuk memberikan nilai logika (0 atau 1). Pin berlabel " ~ " adalah pin-pin PWM (Pulse Width Modulation) yang dapat digunakan untuk menghasilkan PWM.

6. Analog Pins

Papan Arduino UNO memiliki 6 pin analog A0 sampai A5. Digunakan untuk membaca sinyal atau sensor analog seperti sensor jarak, suhu dsb, dan mengubahnya menjadi nilai digital.

7. LED Power Indicator

Lampu ini akan menyala dan menandakan Papan Arduino mendapatkan supply listrik dengan baik.

B. BAGIAN-BAGIAN PENDUKUNG

1. RAM

RAM (Random Access Memory) adalah tempat penyimpanan sementara pada komputer yang isinya dapat diakses dalam waktu yang tetap, tidak memperdulikan letak data tersebut dalam memori atau acak. Secara umum ada 2 jenis RAM yaitu SRAM (Static Random Acces Memory) dan DRAM (Dynamic Random Acces Memory).

2. ROM

ROM (*Read-only Memory*) adalah perangkat keras pada computer yang dapat menyimpan data secara permanen tanpa harus memperhatikan adanya sumber listrik. ROM terdiri dari Mask ROM, PROM, EPROM, EEPROM.

1.5 Prosedur Percobaan

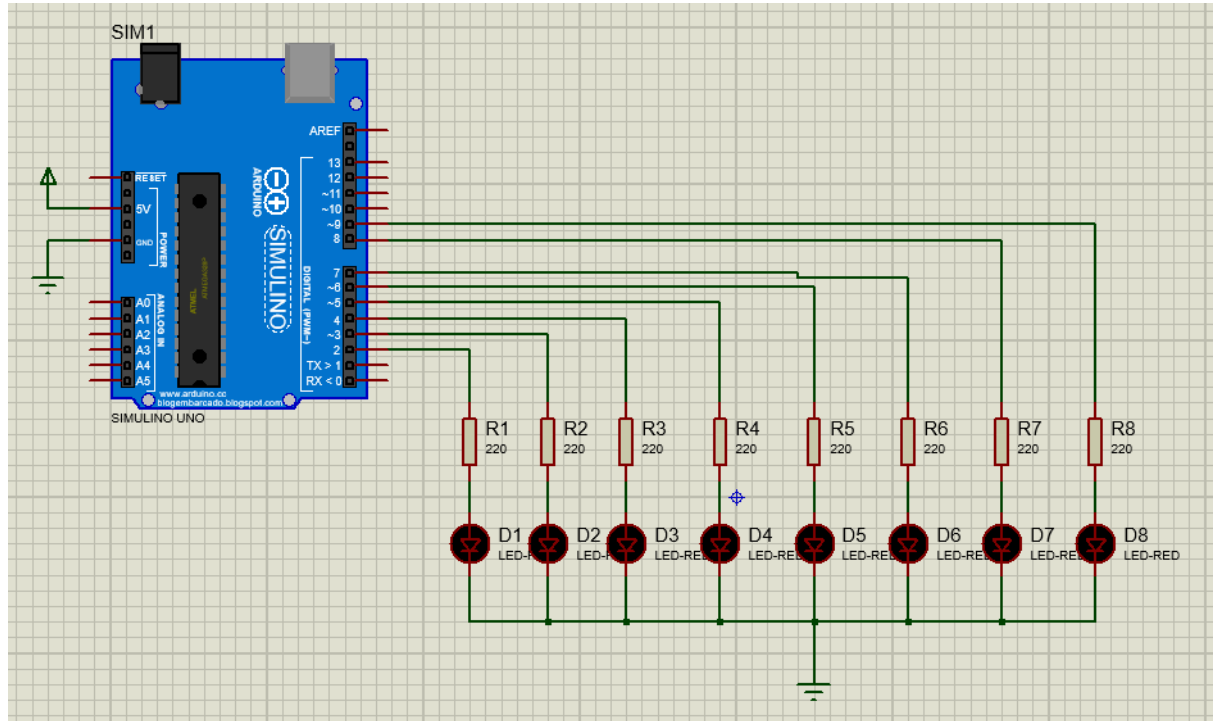
1. Percobaan 1 ON- OFF LED

1) Hardware



Gambar 2. LED

2) Rangkaian Percobaan



Gambar 3. Rangkaian Percobaan ON-OFF LED

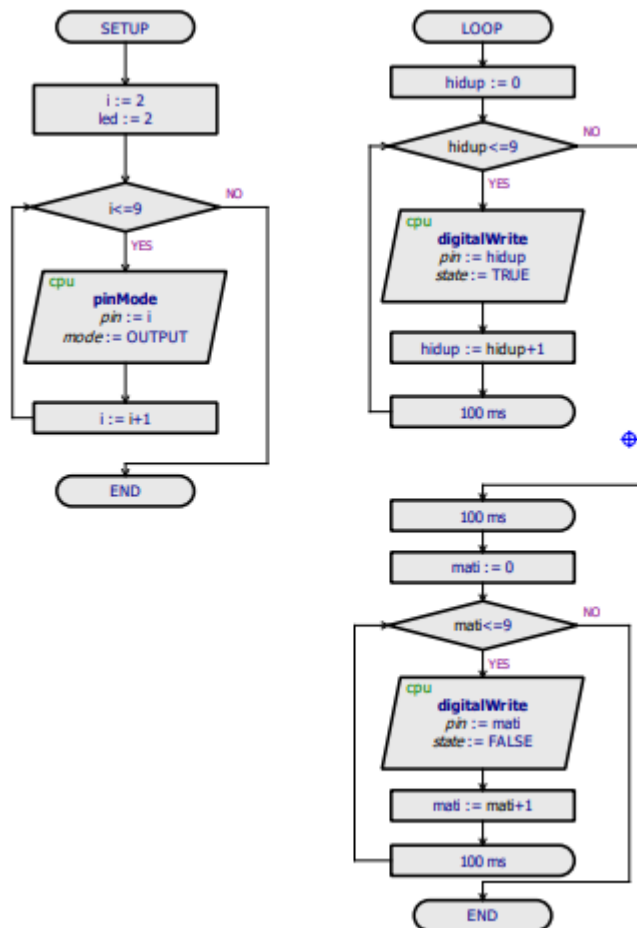
3) Listing Program

```
int led [ ] = {2, 3, 4, 5, 6, 7, 8, 9}; //mendeklarasikan pin 2-9 yang digunakan
void setup()                               //semua kode dalam fungsi ini hanya dibaca sekali
{
    for (int i = 0; i < 8; i++) //perulangan dari 0 dengan nilai batas kecil sama dengan 8;
        increase
    {
        pinMode (led[i], OUTPUT); //menyatakan pin yang dihubungkan ke LED sebagai
        OUTPUT
    }
}
void loop(){                               //Semua kode dalam fungsi ini dibaca berulang
    for (int hidup = 0; hidup < 8; hidup++) //Kondisi perulangan dari 0 dengan batas nilai 8
        untuk menhidupkan LED
    {
        digitalWrite(led[hidup], HIGH); //LED diberi logika 1 sehingga LED menyala
        delay(100);                       //Jarak waktu 100 ms antar LED yang akan menyala
    }
}
```

```

}
delay(100);                //Jarak waktu 100 ms sebelum memulai perulangan
for (int mati = 0; mati < 8; mati++) //Kondisi perulangan dari 0 dengan batas nilai 8; untuk
    mematikan LED
{
    digitalWrite(led[mati], LOW); //LED diberi logika 0 sehingga LED mati
    delay(100);                //Jarak waktu 100 ms antar LED yang akan
    mati
}
delay(100);                //Jarak waktu 100 ms sebelum memulai perulangan
}
    
```

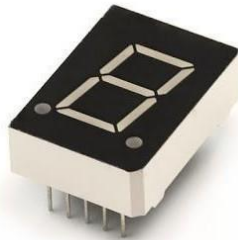
4) Flowchart



Gambar 4. Flowchart Rangkaian Percobaan ON-OFF LED

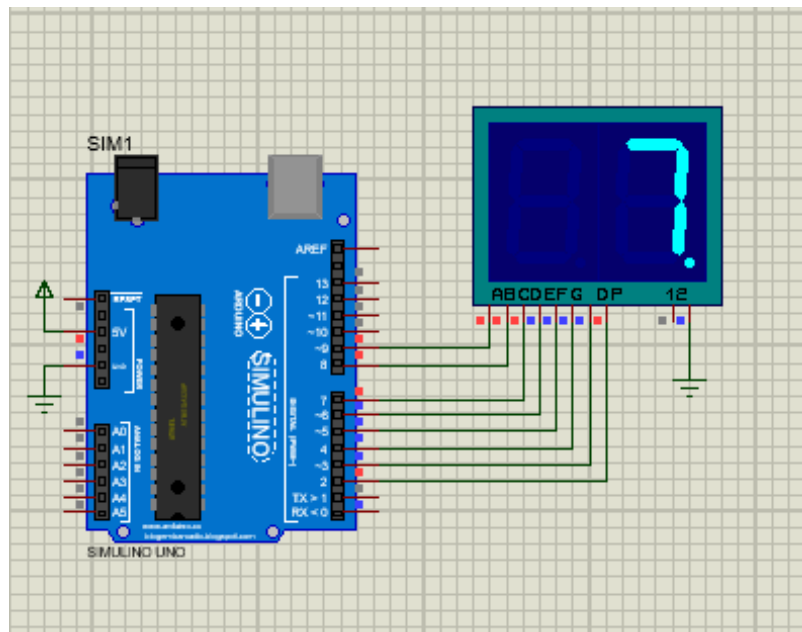
2. Percobaan 2 Menampilkan Karakter Pada 7-Segment

1) Hardware



Gambar 5. Seven Segment

2) Rangkaian Percobaan



Gambar 6. Rangkaian Percobaan 7 Segment

3) Listing Program

```
byte pin[] = {2, 3, 4, 5, 6, 7, 8, 9}; //Deklarasi pin yang digunakan pada 7-segment dengan tipe data byte
void setup() //Semua kode dalam fungsi ini hanya dieksekusi sekali
{ for (int i = 0; i < 9; i++) //Kondisi perulangan dari 0 hingga batas kecil dari 9; Increase
{
pinMode(pin[i], OUTPUT); //Deklarasi pin yang digunakan sebagai OUTPUT
}
}
```

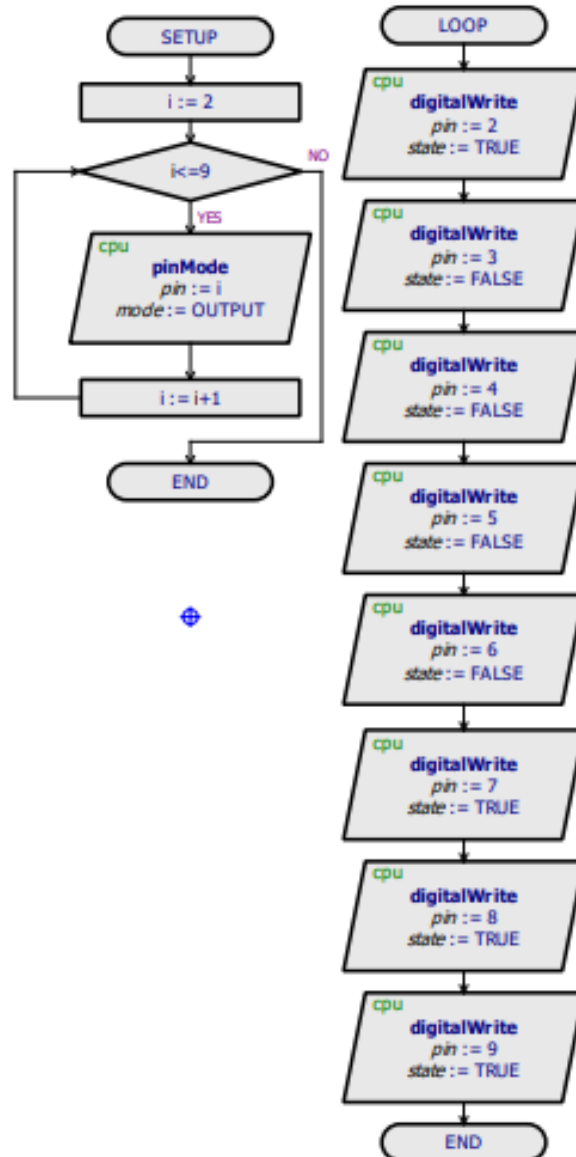


KEMENTRIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
LABORATORIUM ELEKTRONIKA INDUSTRI
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS ANDALAS
Gedung Jurusan Teknik Elektro Lantai III Kampus Unand Limau Manis Padang



```
}  
void loop() //Semua kode dalam fungsi ini di eksekusi berulang  
{  
  digitalWrite(2, HIGH);  
  digitalWrite(3, LOW);  
  digitalWrite(4, LOW);  
  digitalWrite(5, LOW);  
  digitalWrite(6, LOW);  
  digitalWrite(7, HIGH);  
  digitalWrite(8, HIGH);  
  digitalWrite(9, HIGH);  
}
```

4). Flowchart



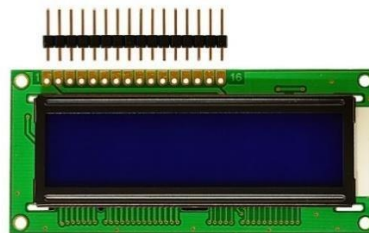
Gambar 7. Flowchart Rangkaian Percobaan 7 Segment

3. Percobaan 3 Menampilkan input keypad pada LCD

1) Hardware

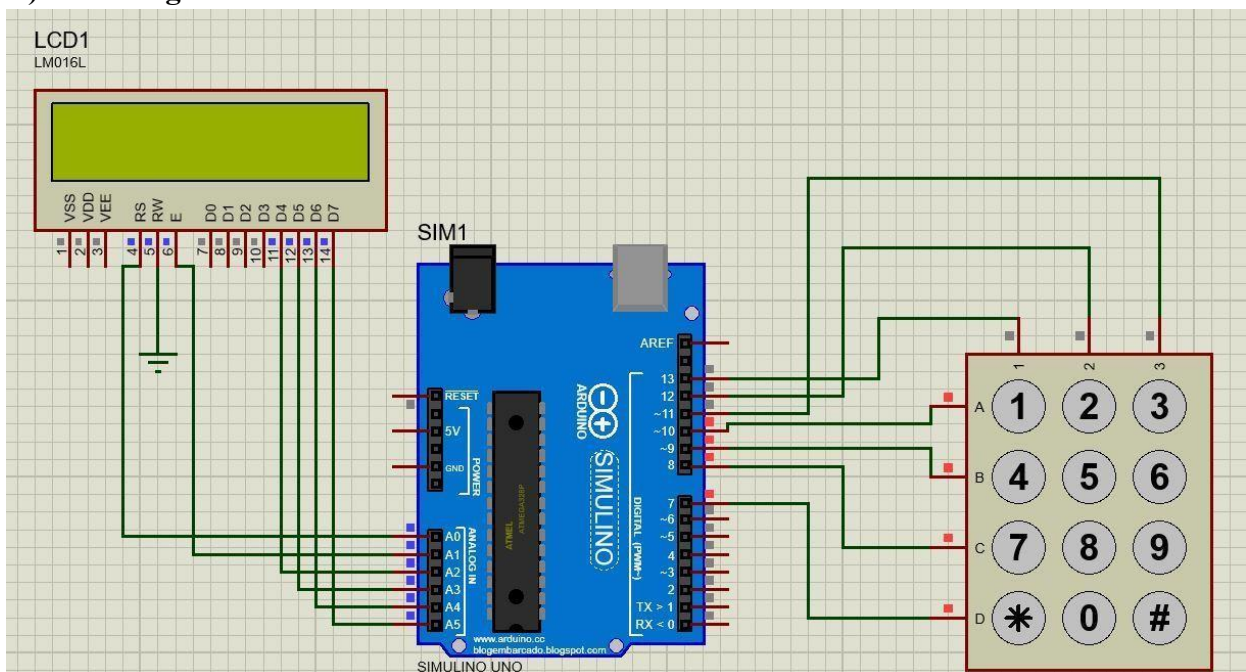


Gambar 8. Keypad



Gambar 9. LCD Character

2) Rangkaian Percobaan



Gambar 10. Rangkaian Percobaan Keypad dan LCD



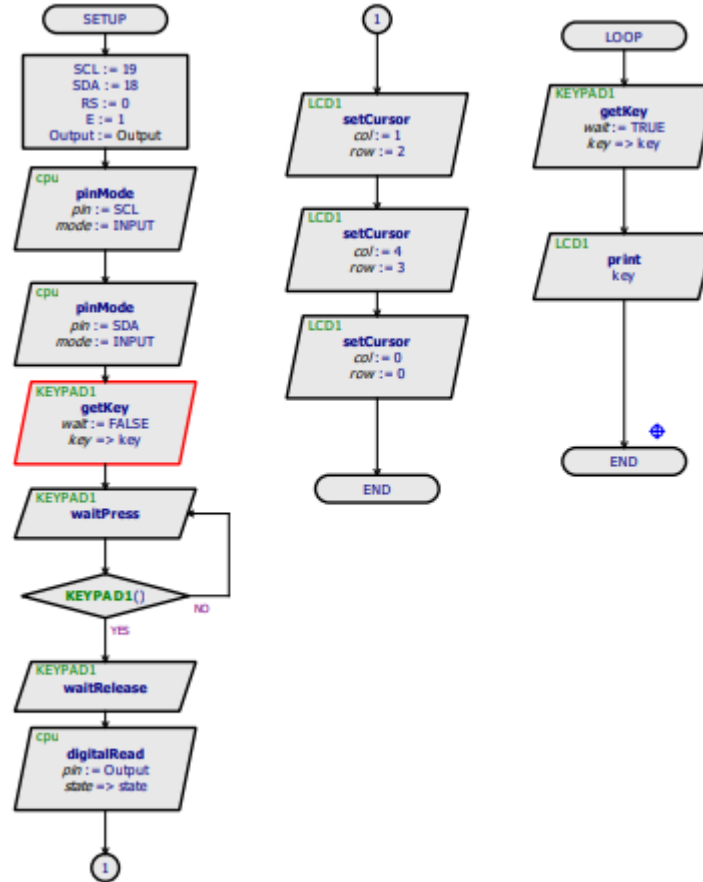
3) Listing Program

```
#include <LiquidCrystal.h> //Deklarasi library LCD
#include <Keypad.h> //Deklarasi library keypad

const byte ROWS = 4; //Deklarasi jumlah baris
const byte COLS = 3; //Deklarasi jumlah kolom
char keys[ROWS][COLS] = {
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'},
  {'*','0','#'}
};
//Inisialisasi keypad matriks 4x3

byte rowPins[ROWS] = {10, 9, 8, 7}; //Deklarasi pin yang digunakan untuk baris
byte colPins[COLS] = {13, 12, 11}; //Deklarasi pin yang digunakan untuk kolom
// Inisialisasi library dengan pin yang digunakan
LiquidCrystal lcd(A0, A1, A2, A3, A4, A5);
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
void setup() {
  //Semua kode dalam fungsi ini akan dieksekusi sekali
  // Set baris dan kolom LCD
  lcd.begin(20, 4);
  lcd.setCursor(1,2);
  lcd.setCursor(4,3);
  lcd.setCursor(0,0);
}
void loop() {
  //Semua kode dalam fungsi ini akan dieksekusi berulang char
  key = keypad.getKey(); //Membaca keypad
  if (key) //Kondisi pilihan jika keypad ditekan
  {
    lcd.print(key); //Menampilkan input dari keypad pada LCD
  }
}
```

4) Flowchart



Gambar 11. Flowchart Rangkaian Percobaan Keypad dan LCD



MODUL II

PWM DAN ADC

2.1 Pendahuluan

- a) Asistensi dilakukan 3x dengan lama pertemuan 20 menit (Rabu, Kamis, Jumat)
- b) Praktikum dilakukan 1x dengan lama pertemuan 90 menit (Selasa)
- c) Laporan akhir dikumpulkan pada hari Kamis

2.2 Tujuan

- a) Memahami prinsip kerja PWM pada mikrokontroler
- b) Memahami prinsip kerja ADC pada mikrokontroler
- c) Menggunakan PWM dan ADC pada Arduino

2.3 Alat dan Bahan

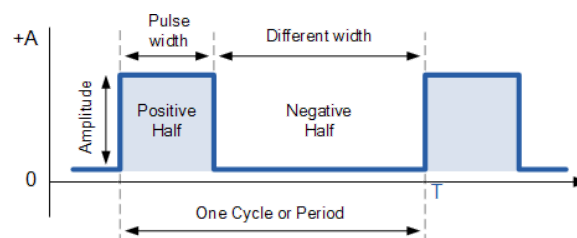
- a) Modul Arduino
- b) Motor DC
- c) LM35

2.4 Dasar Teori

2.4.1 Pulse Width Modulation

PWM (*Pulse Width Modulation*) adalah salah satu teknik modulasi dengan mengubah lebar pulsa (duty cycle) dengan nilai amplitudo dan frekuensi yang tetap. Satu siklus pulsa merupakan kondisi high kemudian berada di zona transisi ke kondisi low. Lebar pulsa PWM berbanding lurus dengan amplitudo sinyal asli yang belum termodulasi.

Duty Cycle adalah perbandingan antara waktu ON (lebar pulsa High) dengan perioda. Duty Cycle biasanya dinyatakan dalam bentuk persen (%).



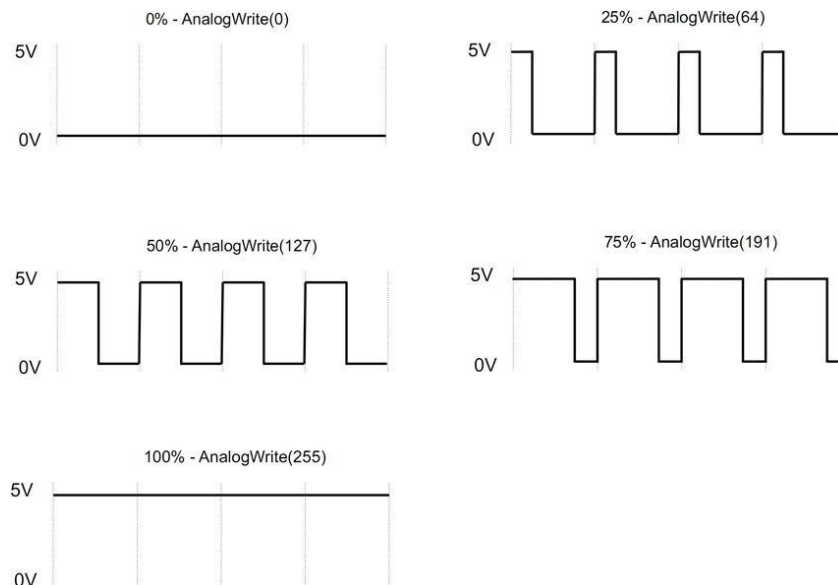
Gambar 12. Duty Cycle

$$\text{Duty Cycle} = t_{\text{ON}} / t_{\text{total}}$$

tON = Waktu ON atau Waktu dimana tegangan keluaran berada pada posisi tinggi (high atau 1)
 tOFF = Waktu OFF atau Waktu dimana tegangan keluaran berada pada posisi rendah (low atau 0)
 ttotal = Waktu satu siklus atau penjumlahan antara tON dengan tOFF atau disebut juga dengan “periode satu gelombang”

Pada board Arduino Uno, pin yang bisa dimanfaatkan untuk PWM adalah pin yang diberi tanda tilde (~), yaitu pin 3, 5, 6, 9, 10, dan pin 11. Pin-pin tersebut merupakan pin yang bisa difungsikan untuk input analog atau output analog. Oleh sebab itu, jika akan menggunakan PWM pada pin ini, bisa dilakukan dengan perintah `analogWrite()`;

PWM pada arduino bekerja pada frekuensi 500Hz, artinya 500 siklus/ketukan dalam satu detik. Untuk setiap siklus, kita bisa memberi nilai dari 0 hingga 255. Ketika kita memberikan angka 0, berarti pada pin tersebut tidak akan pernah bernilai 5 volt (pin selalu bernilai 0 volt). Sedangkan jika kita memberikan nilai 255, maka sepanjang siklus akan bernilai 5 volt (tidak pernah 0 volt). Jika kita memberikan nilai 127 (kita anggap setengah dari 0 hingga 255, atau 50% dari 255), maka setengah siklus akan bernilai 5 volt, dan setengah siklus lagi akan bernilai 0 volt. Sedangkan jika kita memberikan 25% dari 255 ($\frac{1}{4} \times 255$ atau 64), maka $\frac{1}{4}$ siklus akan bernilai 5 volt, dan $\frac{3}{4}$ sisanya akan bernilai 0 volt, dan ini akan terjadi 500 kali dalam 1 detik.



Gambar 13. Siklus Sinyal PWM pada Arduino

2.4.2 Analog to Digital Converter

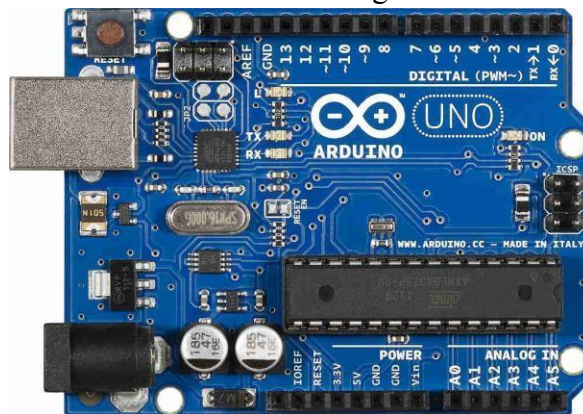
ADC atau Analog to Digital Converter merupakan salah satu perangkat elektronika yang digunakan sebagai penghubung dalam pemrosesan sinyal analog oleh sistem digital. Fungsi

utama dari fitur ini adalah mengubah sinyal masukan yang masih dalam bentuk sinyal analog menjadi sinyal digital dengan bentuk kode-kode digital. Ada 2 faktor yang perlu diperhatikan pada proses kerja ADC yaitu kecepatan sampling dan resolusi.

Kecepatan sampling menyatakan seberapa sering perangkat mampu mengkonversi sinyal analog ke dalam bentuk sinyal digital dalam selang waktu yang tertentu. Biasa dinyatakan dalam *sample per second (SPS)*. Sementara Resolusi menyatakan tingkat ketelitian yang dimiliki. Pada Arduino, resolusi yang dimiliki adalah 10 bit atau rentang nilai digital antara 0 - 1023. Dan pada Arduino tegangan referensi yang digunakan adalah 5 volt, hal ini berarti ADC pada Arduino mampu menangani sinyal analog dengan tegangan 0 - 5 volt. Pada Arduino, menggunakan pin analog input yang diawali dengan kode A(A0- A5 pada Arduino Uno). Fungsi untuk mengambil data sinyal input analog menggunakan `analogRead(pin)`;

2.4.3 Arduino

Arduino adalah kit elektronik atau papan rangkaian elektronik open source yang di dalamnya terdapat komponen utama yaitu sebuah chip mikrokontroler dengan jenis AVR dari perusahaan Atmel. Arduino yang kita gunakan dalam praktikum ini adalah Arduino Uno yang menggunakan chip AVR ATmega 328P. Dalam memprogram Arduino, kita bisa menggunakan komunikasi serial agar Arduino dapat berhubungan dengan komputer ataupun perangkat lain. Adapun spesifikasi dari Arduino Uno ini adalah sebagai berikut :



Gambar 14. Arduino Uno

Microcontroller	ATmega328P
Operating Voltage	5 V
Input Voltage (recommended)	7 – 12 V
Input Voltage (limit)	6 – 20 V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA



KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
LABORATORIUM ELEKTRONIKA INDUSTRI
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS ANDALAS
Gedung Jurusan Teknik Elektro Lantai III Kampus Unand Limau Manis Padang



DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

A. BAGIAN-BAGIAN ARDUINO UNO

1. POWER USB

Digunakan untuk menghubungkan Papan Arduino dengan komputer lewat koneksi USB.

2. POWER JACK

Supply atau sumber listrik untuk Arduino dengan tipe Jack. Input DC 5 - 12 V.

3. Crystal Oscillator

Kristal ini digunakan sebagai layaknya detak jantung pada Arduino. Jumlah cetak menunjukkan 16000 atau 16000 kHz, atau 16 MHz.

4. Reset

Digunakan untuk mengulang program Arduino dari awal atau Reset.

5. Digital Pins I / O

Papan Arduino UNO memiliki 14 Digital Pin. Berfungsi untuk memberikan nilai logika (0 atau 1). Pin berlabel " ~ " adalah pin-pin PWM (Pulse Width Modulation) yang dapat digunakan untuk menghasilkan PWM.

6. Analog Pins

Papan Arduino UNO memiliki 6 pin analog A0 sampai A5. Digunakan untuk membaca sinyal atau sensor analog seperti sensor jarak, suhu dsb, dan mengubahnya menjadi nilai digital.

7. LED Power Indicator

Lampu ini akan menyala dan menandakan Papan Arduino mendapatkan supply listrik dengan baik.

B. BAGIAN-BAGIAN PENDUKUNG

1. RAM

RAM (Random Access Memory) adalah tempat penyimpanan sementara pada komputer yang isinya dapat diakses dalam waktu yang tetap, tidak memperdulikan letak data tersebut dalam memori atau acak. Secara umum ada 2 jenis RAM yaitu SRAM (Static Random Acces Memory) dan DRAM (Dynamic Random Acces Memory).

2. ROM

ROM (*Read-only Memory*) adalah perangkat keras pada computer yang dapat menyimpan data secara permanen tanpa harus memperhatikan adanya sumber listrik. ROM terdiri dari Mask ROM, PROM, EPROM, EEPROM.

2.5 Prosedur Percobaan



KEMENTRIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
LABORATORIUM ELEKTRONIKA INDUSTRI
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS ANDALAS
Gedung Jurusan Teknik Elektro Lantai III Kampus Unand Limau Manis Padang



1. Percobaan 1 Menghidupkan Motor DC

- Pastikan semua supply dalam keadaan off
- Hubungkan jumper seperti gambar dibawah
- Buatlah listing program yang telah ada pada modul
- Hidupkan supply pada modul
- Upload program pada rangkaian
- Tekan tombol reset
- Amati percobaan. Jika tidak sesuai, perbaiki rangkaian atau program
- Jika sesuai, maka selesai dan demokan pada instruktur atau asisten yang mengawas
- Matikan supply pada modul

1) Hardware

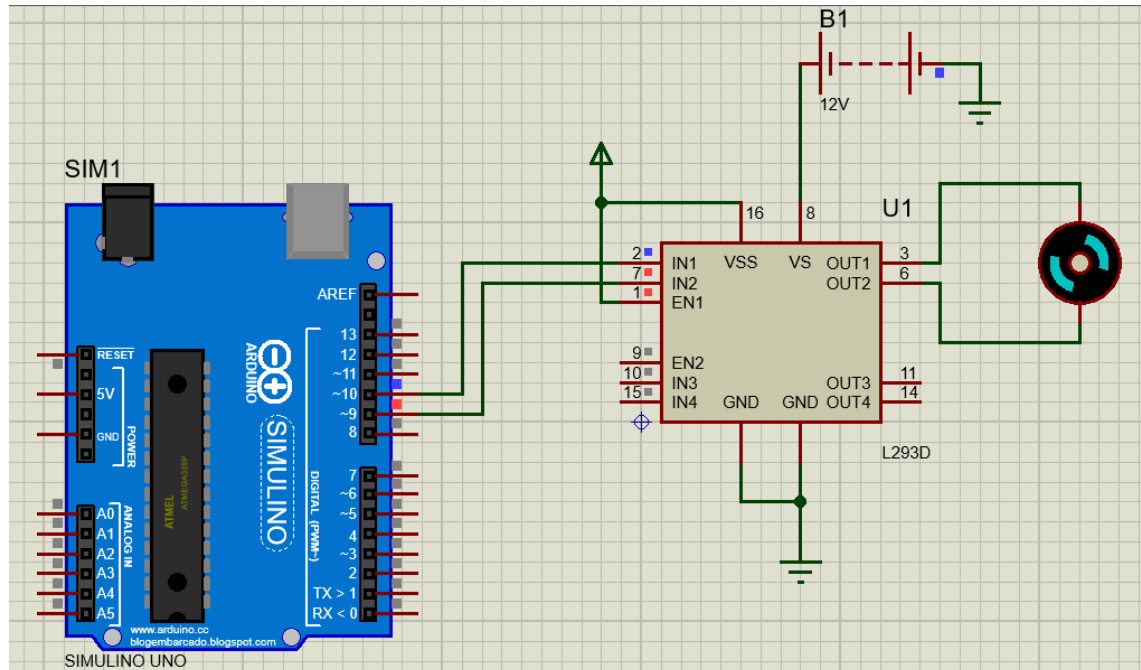


Gambar 15. Motor DC



Gambar 16. Driver Motor

2) Rangkaian Percobaan



Gambar 17. Rangkaian Percobaan Motor DC

3) Listing Program

```
#define in1 10 //Deklarasi pin 10 sebagai input 1
#define in2 9 //Deklarasi pin 9 sebagai input 2
void setup() { //Semua kode dalam fungsi ini di eksekusi sekali
    pinMode(in1, OUTPUT); //Deklarasi in1 sebagai OUTPUT
    pinMode(in2, OUTPUT); //Deklarasi in2 sebagai OUTPUT
}
void loop() { //Semua kode dalam fungsi ini dieksekusi berulang
    digitalWrite(in1, HIGH); //in1 diberi logika 1
    digitalWrite(in2, LOW); //in2 diberi logika 0
    delay(1000); //Jarak waktu 1000 ms setelah input dieksekusi

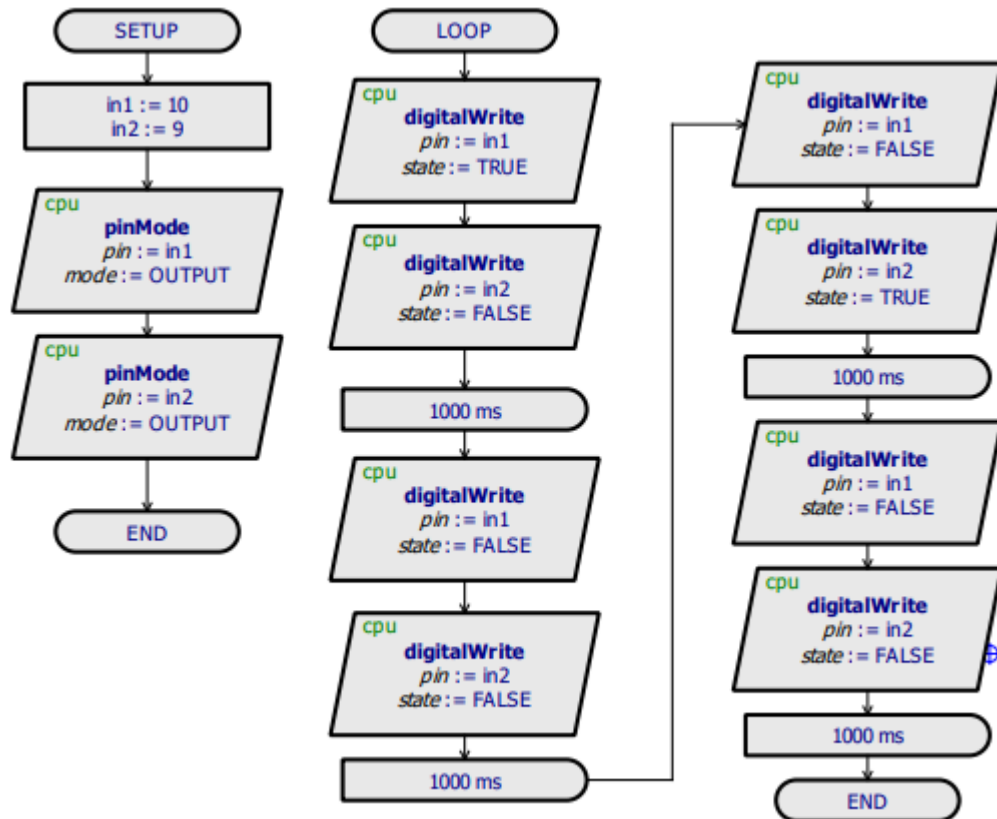
    digitalWrite(in1, LOW); //in1 diberi logika 0
    digitalWrite(in2, LOW); //in2 diberi logika 0
    delay(1000); //Jarak waktu 1000 ms setelah input dieksekusi

    digitalWrite(in1, LOW); //in1 diberi logika 0
    digitalWrite(in2, HIGH); //in2 diberi logika 1
    delay(1000); //Jarak waktu 1000 ms setelah input dieksekusi

    digitalWrite(in1, LOW); //in1 diberi logika 0
```

```
digitalWrite(in2, LOW); //in1 diberi logika 0
delay(1000);           //Jarak waktu 1000 ms setelah input dieksekusi
}
```

4) Flowchart

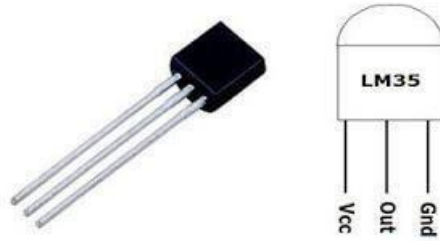


Gambar 18. Flowchart Rangkaian Percobaan Motor DC

2. Percobaan 2 Menampilkan LM35 ke LCD

- Pastikan semua supply dalam keadaan off
- Hubungkan jumper seperti gambar dibawah
- Buatlah listing program yang telah ada pada modul
- Hidupkan supply pada modul
- Upload program pada rangkaian
- Tekan tombol reset
- Amati percobaan. Jika tidak sesuai, perbaiki rangkaian atau program
- Jika sesuai, maka selesai dan demokan pada instruktur atau asisten yang mengawas
- Matikan supply pada modul

1) Hardware

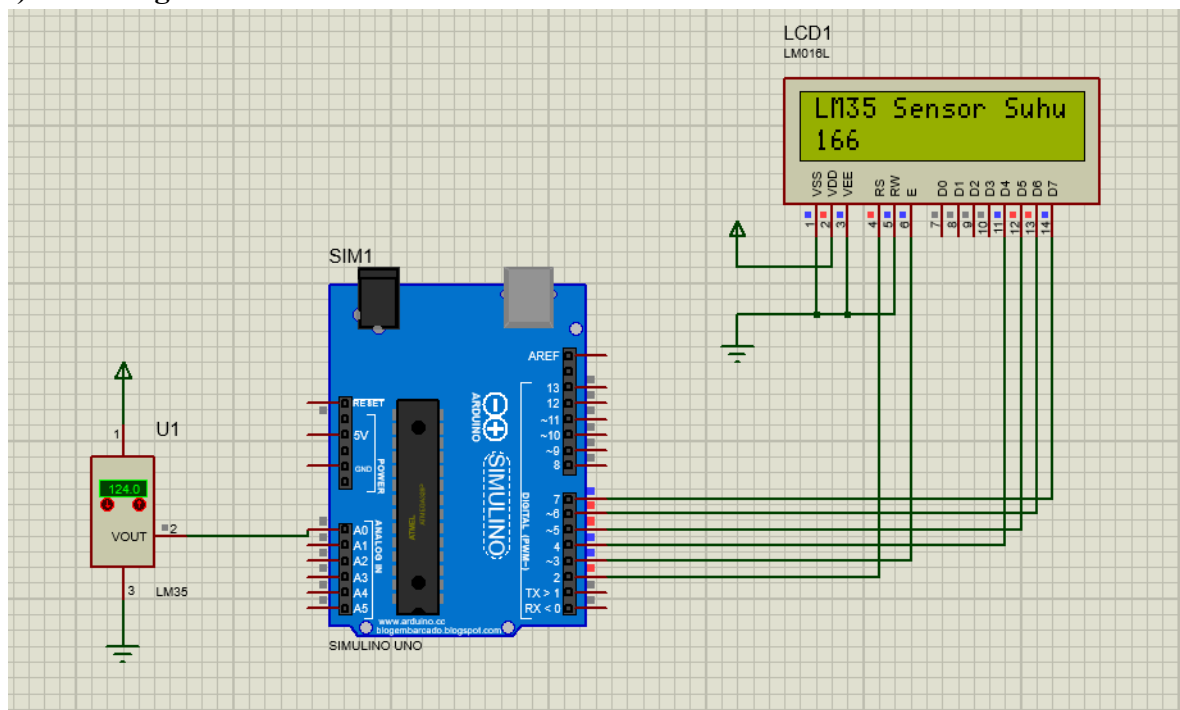


Gambar 19. Sensor LM35



Gambar 20. LCD

2) Rangkaian Percobaan



Gambar 21. Rangkaian Percobaan LM35 dan LCD



KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
LABORATORIUM ELEKTRONIKA INDUSTRI
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS ANDALAS
Gedung Jurusan Teknik Elektro Lantai III Kampus Unand Limau Manis Padang



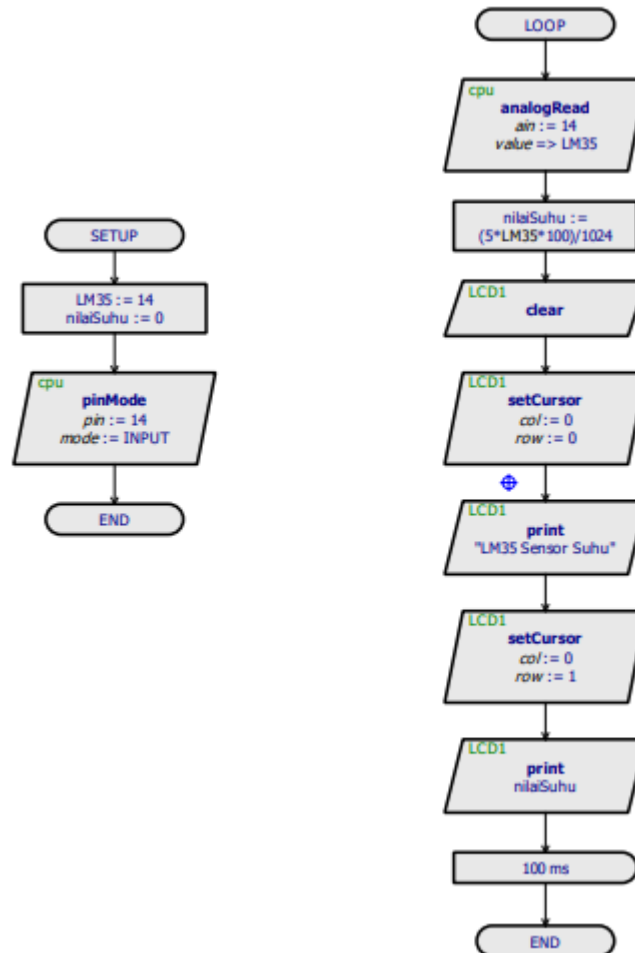
3) Listing Program

```
#include <LiquidCrystal.h>    //Deklarasi library LCD
#define LM35 A0              //Deklarasi pin A0 untuk LM35
LiquidCrystal lcd(2,3,4,5,6,7); //Deklarasi pin 2-7 untuk LCD
int nilaiSuhu;               //Deklarasi variabel nilaiSuhu

void setup()                 //Semua kode dalam fungsi ini dieksekusi sekali
{
  pinMode(A0, INPUT);        //Deklarasi pin A0 sebagai OUTPUT
  lcd.begin(16,2);            //Dimensi LCD yang digunakan
}

void loop()                  //Semua kode dalam fungsi ini dieksekusi berulang
{
  nilaiSuhu=((5*analogRead(LM35)*100.00)/1024); //Mencari nilai Suhu
  lcd.clear();                //Menghapus layar LCD
  lcd.setCursor(0,0);          //Menentukan posisi kursor pada awal penulisan
  lcd.print("LM35 Sensor Suhu"); //Menampilkan text pada LCD
  lcd.setCursor(0,1);          //Menentukan posisi kursor pada awal penulisan
  lcd.print(nilaiSuhu);        //Menampilkan nilaiSuhu pada LCD
  delay(100);                 //Waktu delay 100 ms setelah dieksekusi
}
```

4) Flowchart



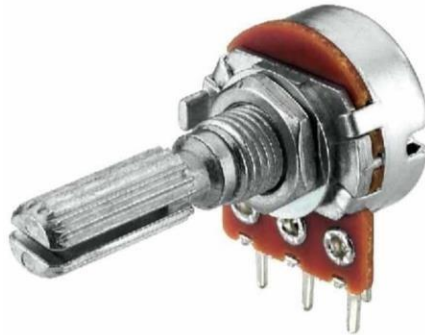
Gambar 22. Flowchart Rangkaian Percobaan LM35 dan LCD

3. Percobaan 3 Mengatur Kecepatan Putar Motor DC

1) Hardware

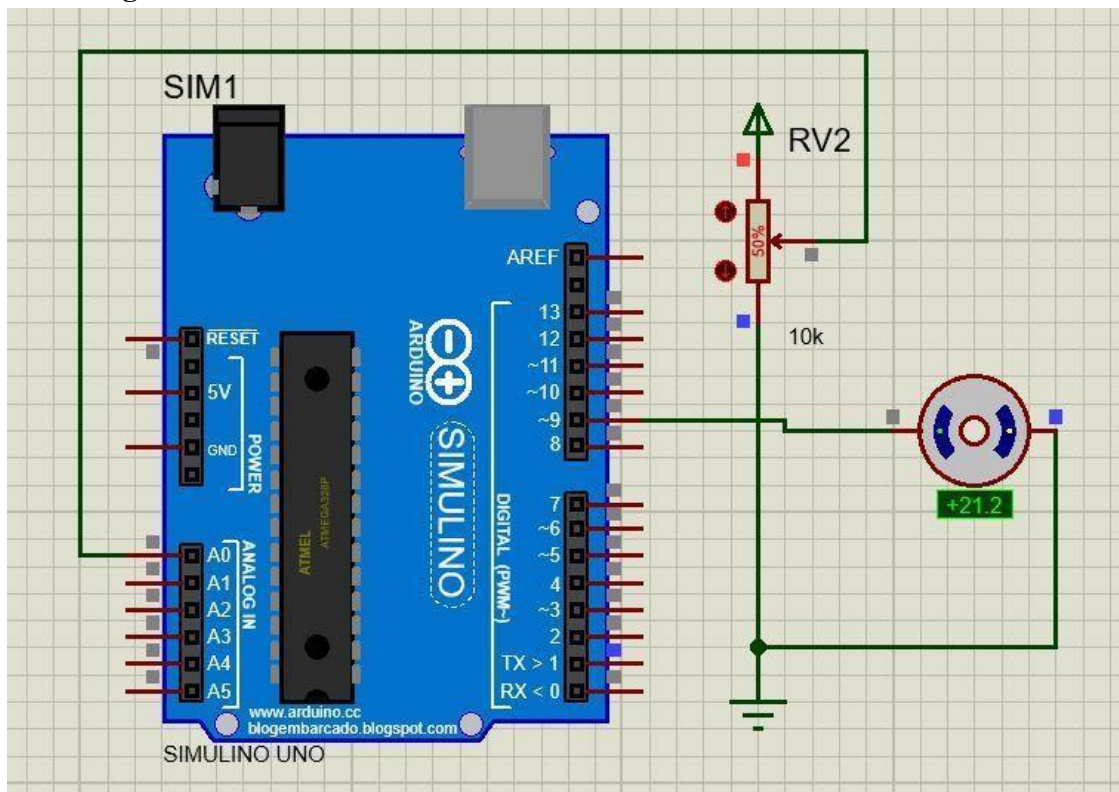


Gambar 23. Motor DC



Gambar 24. Potensiometer

2) Rangkaian Percobaan



Gambar 25. Rangkaian Percobaan Mengatur Laju Motor DC

3) Listing Program

```
byte pot= A0; //Deklarasi pin A0 untuk
potensiometer
```



KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
LABORATORIUM ELEKTRONIKA INDUSTRI
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS ANDALAS
Gedung Jurusan Teknik Elektro Lantai III Kampus Unand Limau Manis Padang



```
byte motor= 9; //Deklarasi pin 9 untuk motor

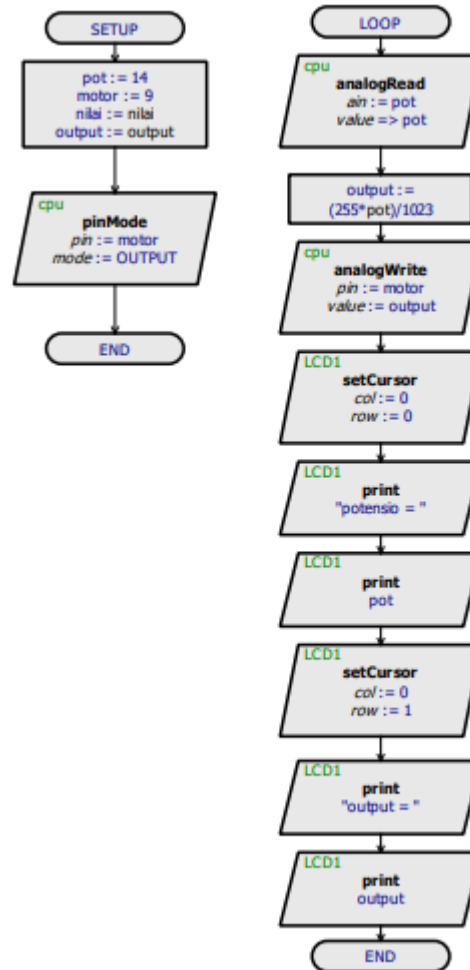
int nilai; //Deklarasi variabel nilai
int output; //Deklarasi variabel output

void setup(){ //Semua kode dalam fungsi ini dieksekusi sekali
pinMode(motor, OUTPUT); //Deklarasi motor sebagai OUTPUT
Serial.begin(9600);          //Set baud rate 9600
}
void loop(){ //Semua kode dalam fungsi ini dieksekusi
berulang
  nilai= analogRead(pot); //Membaca nilai potensiometer
  output= map(nilai, 0, 1023, 0, 255);

  analogWrite(motor, output);

  Serial.print("potensiometer:
");
  Serial.print(nilai);
  Serial.print(" ");
  Serial.print("output: ");
  Serial.print(output);
  delay(2);
}
```

4) Flowchart



Gambar 26. Flowchart Rangkaian Percobaan Mengatur Laju Motor DC

MODUL III

COMMUNICATION

3.1 Pendahuluan

- a) Asistensi dilakukan 3x dengan lama pertemuan 20 menit (Rabu, Kamis, Jumat)
- b) Praktikum dilakukan 1x dengan lama pertemuan 90 menit (Selasa)
- c) Laporan akhir dikumpulkan pada hari Kamis

3.2 Tujuan

- a) Memahami prinsip kerja UART, SPI, dan I²C
- b) Mengaplikasikan protokol komunikasi UART, SPI, dan I²C pada Arduino

3.3 Alat dan Bahan

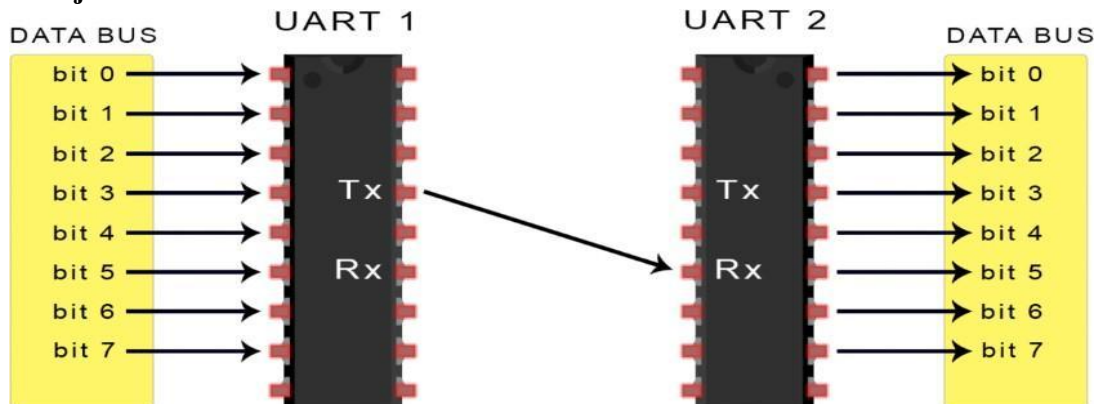
- a) Modul Arduino

3.4 Dasar Teori

3.4.1 Universal Asynchronous Receiver Transmitter (UART)

UART (Universal Asynchronous Receiver-Transmitter) adalah bagian perangkat keras komputer yang menerjemahkan antara bit-bit paralel data dan bit-bit serial. UART biasanya berupa sirkuit terintegrasi yang digunakan untuk komunikasi serial pada komputer atau port serial perangkat periperal.

Cara Kerja Komunikasi UART



Gambar 27. Cara Kerja Komunikasi UART

Data dikirimkan secara paralel dari data bus ke UART1. Pada UART1 ditambahkan start bit, parity bit, dan stop bit kemudian dimuat dalam satu paket data. Paket data ditransmisikan

secara serial dari Tx UART1 ke Rx UART2. UART2 mengkonversikan data dan menghapus bit tambahan, kemudian di transfer secara parallel ke data bus penerima.

3.4.2 Serial Peripheral Interface (SPI)

Serial Peripheral Interface (SPI) merupakan salah satu mode komunikasi serial synchronous kecepatan tinggi yang dimiliki oleh ATmega 328. Komunikasi SPI membutuhkan 3 jalur yaitu MOSI, MISO, dan SCK. Melalui komunikasi ini data dapat saling dikirimkan baik antara mikrokontroler maupun antara mikrokontroler dengan peripheral lain di luar mikrokontroler.

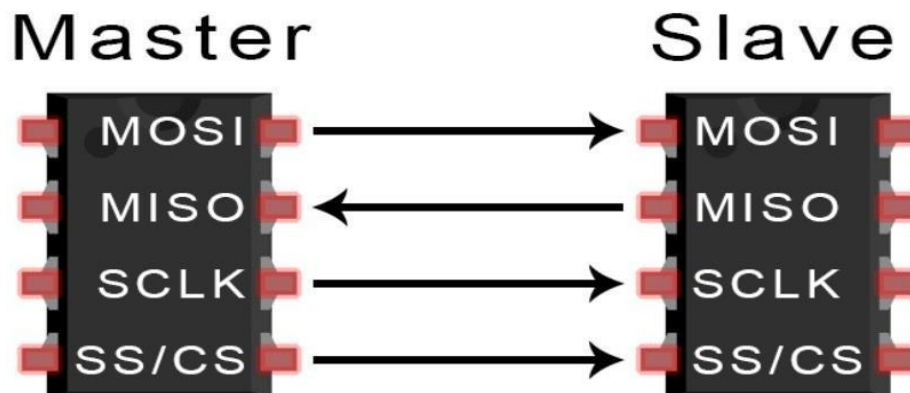
MOSI : Master Output Slave Input Artinya jika dikonfigurasi sebagai master maka pin MOSI sebagai output tetapi jika dikonfigurasi sebagai slave maka pin MOSI sebagai input.

MISO : Master Input Slave Output Artinya jika dikonfigurasi sebagai master maka pin MISO sebagai input tetapi jika dikonfigurasi sebagai slave maka pin MISO sebagai output.

SCLK : Clock Jika dikonfigurasi sebagai master maka pin CLK berlaku sebagai output tetapi jika dikonfigurasi sebagai slave maka pin CLK berlaku sebagai input.

SS/CS : Slave Select/ Chip Select adalah jalur master memilih slave mana yang akan dikirimkan data.

Cara Kerja Komunikasi SPI



Gambar 28. Cara Kerja Komunikasi SPI

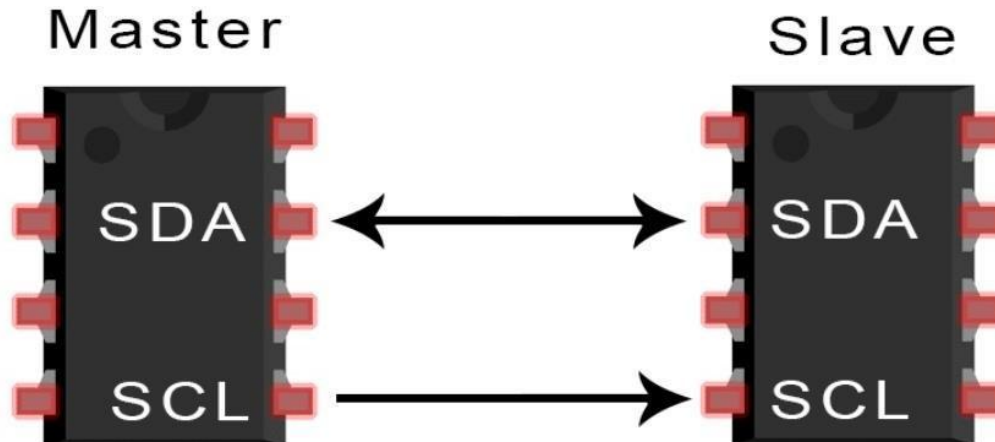
Sinyal clock dialirkan dari master ke slave yang berfungsi untuk sinkronisasi. Master dapat memilih slave mana yang akan dikirimkan data melalui slave select, kemudian data dikirimkan dari master ke slave melalui MOSI. Jika master butuh respon data maka slave akan mentransfer data ke master melalui MISO.

3.4.3 Inter Integrated Circuit (I2C)

Inter Integrated Circuit atau sering disebut I2C adalah standar komunikasi serial dua arah menggunakan dua saluran yang didisain khusus untuk mengirim maupun menerima data.

Sistem I2C terdiri dari saluran SCL (*Serial Clock*) dan SDA (*Serial Data*) yang membawa informasi data antara I2C dengan pengontrolnya.

Cara Kerja Komunikasi I2C

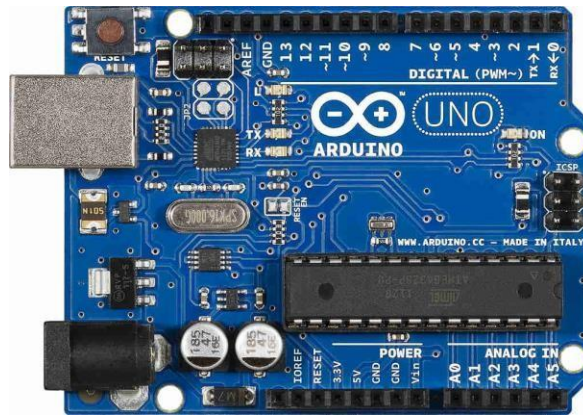


Gambar 29. Cara Kerja Komunikasi I2C

Pada I2C, data ditransfer dalam bentuk message yang terdiri dari kondisi start, Address Frame, R/W bit, ACK/NACK bit, Data Frame 1, Data Frame 2, dan kondisi Stop. Kondisi start dimana saat pada SDA beralih dari logika high ke low sebelum SCL. Kondisi stop dimana saat pada SDA beralih dari logika low ke high sebelum SCL. R/W bit berfungsi untuk menentukan apakah master mengirim data ke slave atau meminta data dari slave. (logika 0 = mengirim data ke slave, logika 1 = meminta data dari slave) ACK/NACK bit berfungsi sebagai pemberi kabar jika data frame ataupun address frame telah diterima receiver.

3.4.4 Arduino

Arduino adalah kit elektronik atau papan rangkaian elektronik open source yang di dalamnya terdapat komponen utama yaitu sebuah chip mikrokontroler dengan jenis AVR dari perusahaan Atmel. Arduino yang kita gunakan dalam praktikum ini adalah Arduino Uno yang menggunakan chip AVR ATmega 328P. Dalam memprogram Arduino, kita bisa menggunakan komunikasi serial agar Arduino dapat berhubungan dengan komputer ataupun perangkat lain. Adapun spesifikasi dari Arduino Uno ini adalah sebagai berikut :



Gambar 30. Arduino Uno

Microcontroller	ATmega328P
Operating Voltage	5 V
Input Voltage (recommended)	7 – 12 V
Input Voltage (limit)	6 – 20 V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

A. BAGIAN-BAGIAN ARDUINO UNO

1. POWER USB

Digunakan untuk menghubungkan Papan Arduino dengan komputer lewat koneksi USB.

2. POWER JACK

Supply atau sumber listrik untuk Arduino dengan tipe Jack. Input DC 5 - 12 V.

3. Crystal Oscillator

Kristal ini digunakan sebagai layaknya detak jantung pada Arduino. Jumlah cetak menunjukkan 16000 atau 16000 kHz, atau 16 MHz.

4. Reset

Digunakan untuk mengulang program Arduino dari awal atau Reset.

5. Digital Pins I / O

Papan Arduino UNO memiliki 14 Digital Pin. Berfungsi untuk memberikan nilai logika (

0 atau 1). Pin berlabel " ~ " adalah pin-pin PWM (Pulse Width Modulation) yang dapat digunakan untuk menghasilkan PWM.

6. Analog Pins

Papan Arduino UNO memiliki 6 pin analog A0 sampai A5. Digunakan untuk membaca sinyal atau sensor analog seperti sensor jarak, suhu dsb, dan mengubahnya menjadi nilai digital.

7. LED Power Indicator

Lampu ini akan menyala dan menandakan Papan Arduino mendapatkan supply listrik dengan baik.

B. BAGIAN-BAGIAN PENDUKUNG

1. RAM

RAM (Random Access Memory) adalah tempat penyimpanan sementara pada komputer yang isinya dapat diakses dalam waktu yang tetap, tidak memperdulikan letak data tersebut dalam memori atau acak. Secara umum ada 2 jenis RAM yaitu SRAM (Static Random Acces Memory) dan DRAM (Dynamic Random Acces Memory).

2. ROM

ROM (*Read-only Memory*) adalah perangkat keras pada computer yang dapat menyimpan data secara permanen tanpa harus memperhatikan adanya sumber listrik. ROM terdiri dari Mask ROM, PROM, EPROM, EEPROM.

3.5 Prosedur Percobaan

1. Percobaan 1 Komunikasi UART

1) Hardware

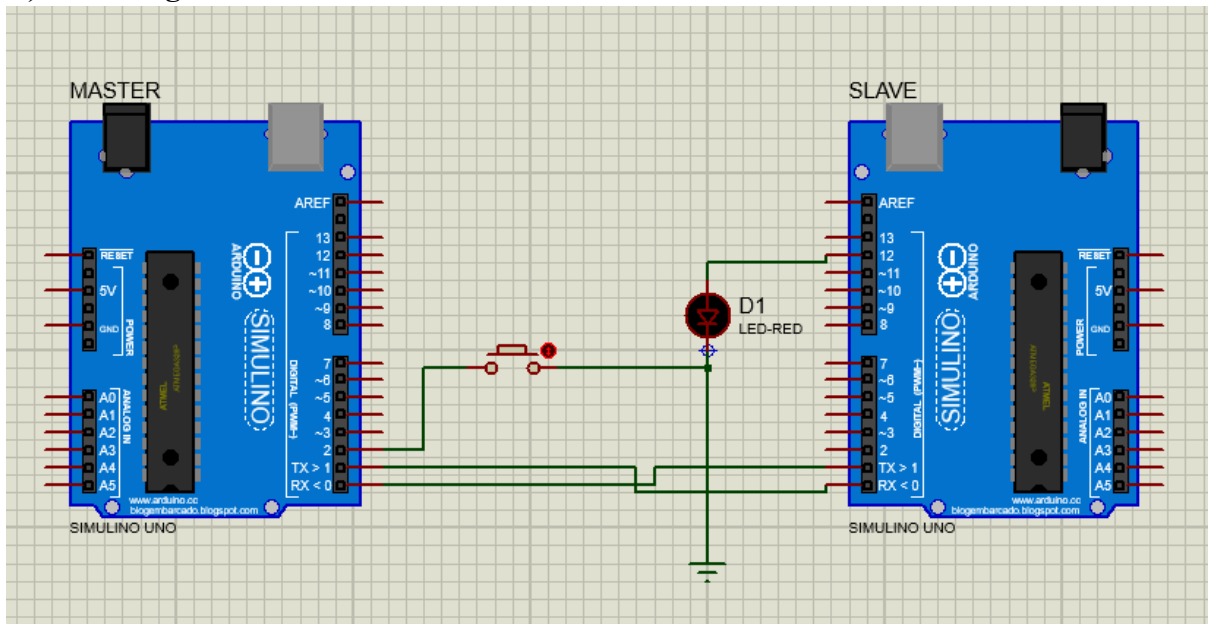


Gambar 31. LED



Gambar 32. Push Button

2) Rangkaian Percobaan



Gambar 33. Rangkaian Percobaan Komunikasi UART



3) Listing Program

//MASTER

```
#define button 2 //Deklarasi pin 2 untuk button

void setup() //Semua kode dalam fungsi ini dieksekusi sekali
{
  pinMode(button,INPUT_PULLUP);
  Serial.begin(9600); //Set baud rate
  9600
}

void loop() //Semua kode dalam fungsi ini dieksekusi berulang
{
  int nilai = digitalRead(button);

  //ditekan
  if(nilai == 0)
  {
    Serial.print("1");
  }
  else
  {
    Serial.print("2");
  }

  delay(200);
}
```

//SLAVE

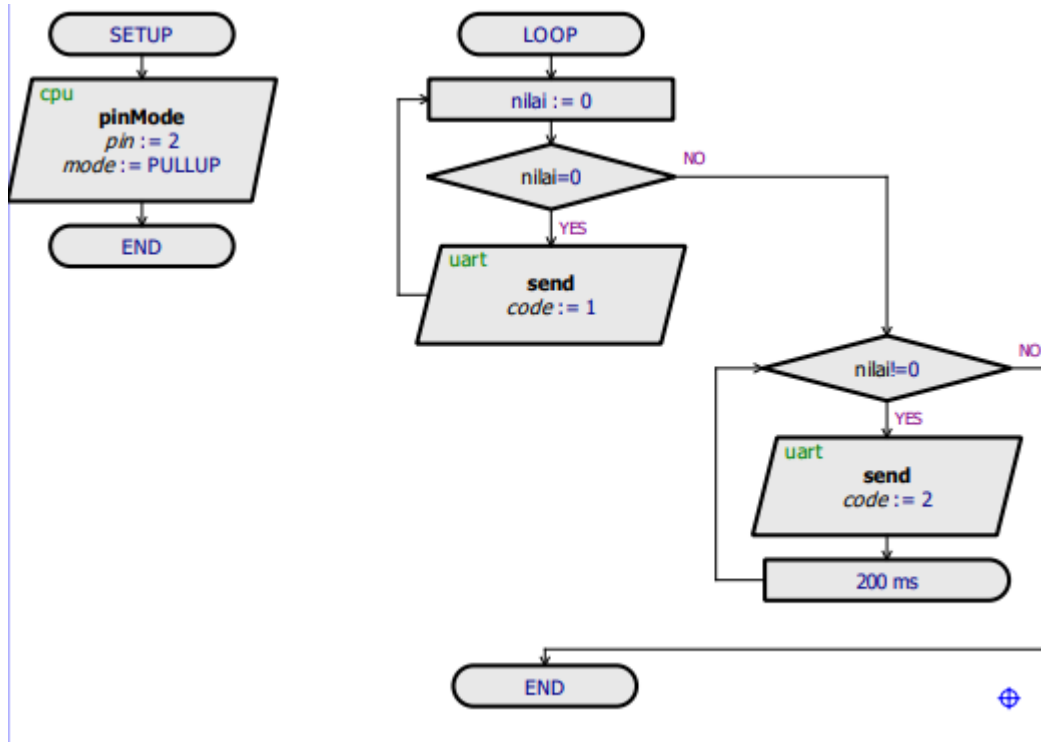
```
#define led 12 //Deklarasi pin 12 untuk LED

void setup() //Semua kode dalam fungsi ini dieksekusi sekali
{
  pinMode(led,OUTPUT); //Deklarasi LED sebagai output
  Serial.begin(9600); //Set baud rate 9600
}

void loop() //Semua program dalam fungsi ini dieksekusi berulang
{
  if(Serial.available()>0)
```

```
{
  int data = Serial.read();
  if(data=='1') //Jika data yang dikirimkan berlogika
  {
    digitalWrite(led,HIGH); //LED menyala
  }
  else
  {
    digitalWrite(led,LOW); //LED mati
  }
}
```

4) Flowchart



Gambar 34. Flowchart Rangkaian Percobaan Komunikasi UART

2. Percobaan 2 Komunikasi SPI



1) Hardware

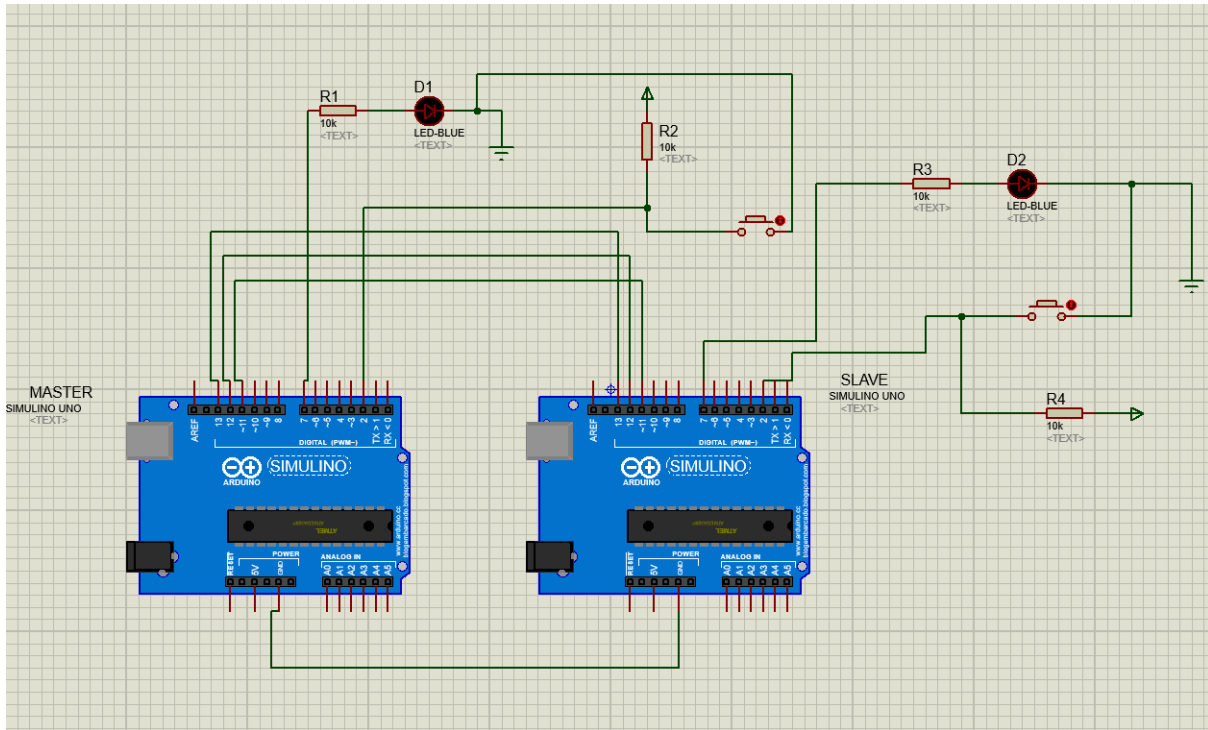


Gambar 35. LED



Gambar 36. Push Button

2) Rangkaian Percobaan



Gambar 37. Rangkaian Percobaan Komunikasi SPI

3) Listing Program

```
//MASTER
#include
<SPI.h> void
setup (void) {
  Serial.begin(115200);
  //set baud rate to 115200 for
  usart digitalWrite(SS,
  HIGH);
  // disable Slave
  Select SPI.begin
  ();
  SPI.setClockDivider(SPI_CLOCK_DIV8); //divide the clock by 8
}

void loop
(void) { char
c;
digitalWrite(SS, LOW); // enable Slave Select
// send test string
```



KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
LABORATORIUM ELEKTRONIKA INDUSTRI
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS ANDALAS
Gedung Jurusan Teknik Elektro Lantai III Kampus Unand Limau Manis Padang

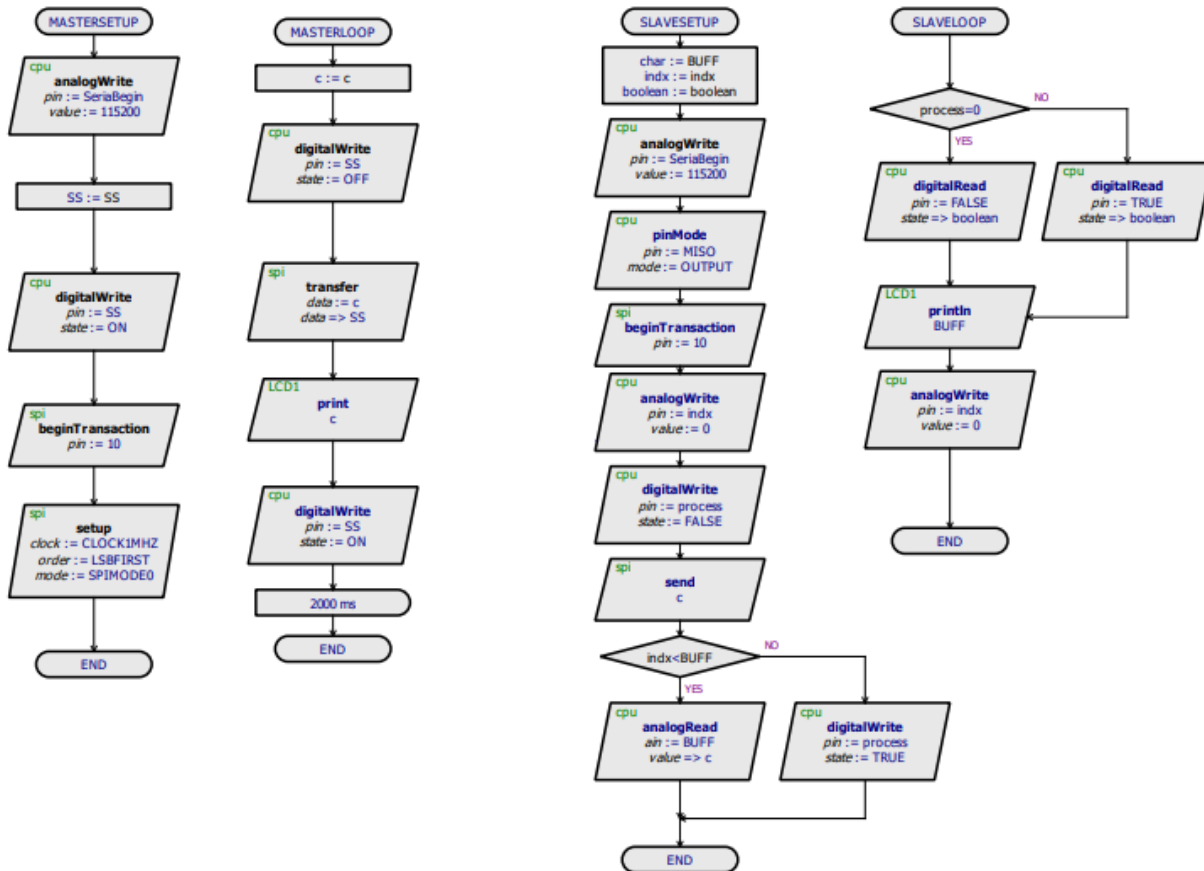


```
for (const char * p = "Hello, world!\r" ; c = *p; p++)
{
    SPI.transfer
    (c);
    Serial.print(c);
}
digitalWrite(SS, HIGH); // disable Slave
Select delay(2000);
}
//SLAVE

#include
<SPI.h> char
buff [50];
volatile byte
indx;
volatile boolean process;

void setup (void) {
    Serial.begin
    (115200);
    pinMode(MISO, OUTPUT); // have to send on master in so it set as output
    SPCR |= _BV(SPE); // turn on SPI in slave mode
    indx = 0; // buffer
    empty process = false;
    SPI.attachInterrupt(); // turn on interrupt
}
ISR (SPI_STC_vect) // SPI interrupt routine
{
    byte c = SPDR; // read byte from SPI Data
    Register if (indx < sizeof buff) {
        buff [indx++] = c; // save data in the next index in the array
        buff if (c == '\r') //check for the end of the word
        process = true;
    }
}
void loop (void)
{ if (process) {
    process = false; //reset the process
    Serial.println (buff); //print the array on serial
    monitor indx= 0; //reset button to zero
}
```

4) Flowchart



Gambar 38. Flowchart Rangkaian Percobaan Komunikasi SPI

3. Percobaan 3 Komunikasi I2C

1) Hardware



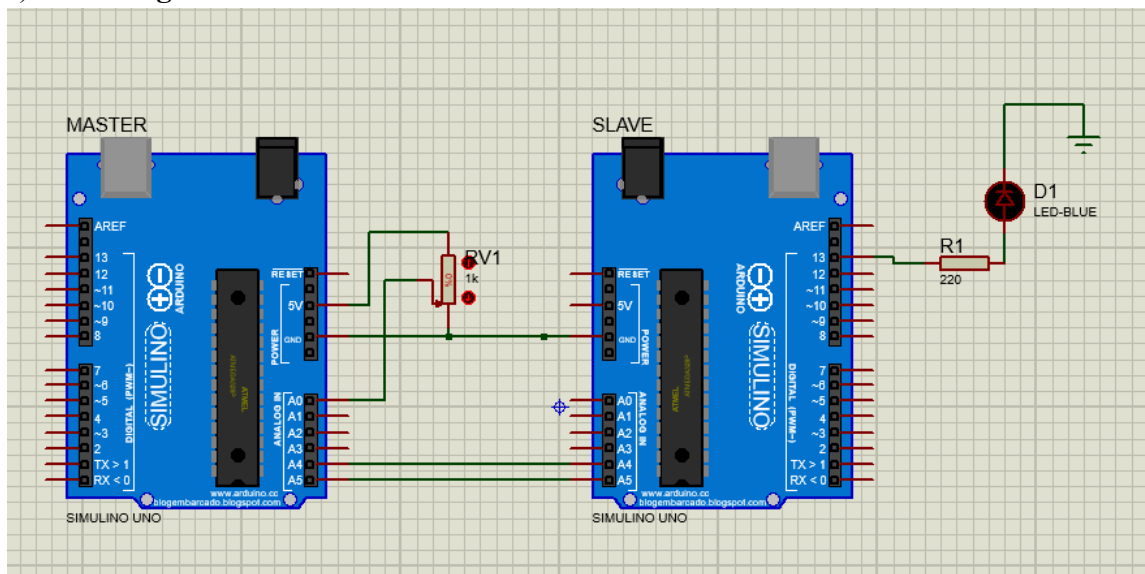


Gambar 39. LED



Gambar 40. Potensiometer

2) Rangkaian Percobaan



Gambar 41. Rangkaian Percobaan Komunikasi I2C



3) Listing Program

```
//MASTER
```

```
#include <Wire.h>
#define MASTER_ADDR 9

int analogPin = 0;
int val = 0;

void setup() {
  Wire.begin();
}

void loop() {
  delay(50);
  val = map(analogRead(analogPin), 0, 1023, 255, 1);

  Wire.beginTransmission(MASTER_ADDR);
  Wire.write(val);
  Wire.endTransmission();
}
```

```
//SLAVE
```

```
#include <Wire.h>
#define SLAVE_ADDR 9

int LED = 13;
int rd;
int br;

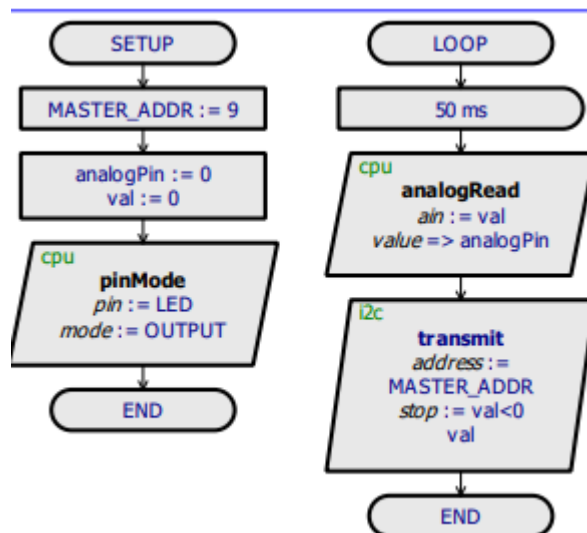
void setup() {
  pinMode(LED,
    OUTPUT);
  Wire.begin(SLAVE_ADDR);
  Wire.onReceive(receiveEvent);

  Serial.begin(9600);
  Serial.println("I2C Slave
    demo");
}
```



```
void receiveEvent() {  
  rd = Wire.read();  
  Serial.println(rd);  
}  
void loop() {  
  delay(50);  
  
  br = map(rd, 1, 255, 100, 2000);  
  
  digitalWrite(LED, HIGH);  
  delay(br);  
  digitalWrite(LED, LOW);  
  delay(br);  
}
```

4) Flowchart



Gambar 42. Flowchart Rangkaian Percobaan Komunikasi I2C



MODUL IV

PROJECT DEMO

4.1 Pendahuluan

- Asistensi dilakukan 3x dengan lama pertemuan 20 menit (daring)
- Praktikum dilakukan 1x dengan lama pertemuan 90 menit (daring)
- Laporan akhir dikumpulkan daring

4.2 Tujuan

- Praktikan dapat merancang sebuah sistem secara detail sehingga rancangan tersebut dapat menjadi sebuah alat
- Praktikan dapat mengkombinasikan berbagai sensor, akuator dan display
- Praktikan dapat membuat sebuah sistem menggunakan mikrokontroler

4.3 Alat dan Bahan

- Mikrokontroler Arduino
- Input : Switch dan sensor
- Output :
 - Display : 7-Segmen, LCD
 - Akuator : Motor

4.4 Ketentuan Praktikum

Praktikan membuat sebuah rancangan sistem secara detail yang ditandai dengan adanya :

- Tujuan dibuat sistem tersebut
 - Landasan teori berdasarkan komponen yang digunakan
 - Listing Program
 - Skema Alat
- Di dalam rancangan sistem tersebut, harus terdapat minimal sebuah input, sebuah output display dan sebuah output aktuator yang telah dipelajari dari modul sebelumnya.
 - Di dalam rancangan sistem tersebut, minimal menggunakan sebuah mikrokontroler Arduino. Praktikan diberi kebebasan untuk memilih Arduino yang akan digunakan.
 - Praktikan membuat simulasi rancangan alat dengan menggunakan perangkat



KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
LABORATORIUM ELEKTRONIKA INDUSTRI
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS ANDALAS
Gedung Jurusan Teknik Elektro Lantai III Kampus Unand Limau Manis Padang



lunak Proteus atau perangkat lunak lainnya.

- Praktikan menjelaskan rancangan sistem, simulasi dan demo alat kepada asisten Labortatorium Elektronika Industri sebelum dapat menunjukkan kepada Dosen Pembimbing.
- Praktikan membuat laporan sesuai dengan kriteria yang ditentukan.
- Tidak dibenarkan untuk meminjam komponen dari Laboratorium Elektronika Industri.

4.5 Laporan Praktikum

Laporan Modul 4 Praktikum Mikrokontroller dan Mikroprosesor di Laboratorium Elektronika Industri, memiliki ketentuan sebagai berikut :

- a. Terdapat tujuan pembuatan alat tersebut.
- b. Terdapat daftar komponen yang akan digunakan dalam pembuatan alat tersebut.
- c. Terdapat landasan teori berdasarkan komponen yang digunakan dalam pembuatan alat tersebut. Sertakan file foto untuk masing-masing komponen tersebut.
- d. Terdapat listing program.
- e. Terdapat foto rangkaian simulasi menggunakan perangkat lunak seperti Proteus.
- f. Terdapat video rangkaian yang disimulasikan.
- g. Terdapat analisa dari rangkaian simulasi yang telah dibuat.
- h. Terdapat tautan untuk mengunduh dokumen berupa video, html dan simulasi.