

MIKROKONTROLER

PROGRAM STUDI
TEKNIK TELKOMUNIKASI
Semester 3

Akuwan Saleh, MT

1. Silabus/SAP/RPS

2. Tool / Program

- Arduino & ESP32; - Pemrograman C
- IDE Arduino; - Simulator Proteus;
- Virtual Serial Port Emulator (**VSPE**).

3. Tata Tertib (TaTib)

- Ijin atau sakit → Surat ijin/sakit
- Tidak Hadir (A), Kehadiran < 90 % ➡ UP;
- Toleransi keterlambatan 10 menit
- Nilai C boleh ikut UP

PENILAIAN

⇒ UTS	= 30%
⇒ UAS	= 40%
⇒ Tugas-1+Absensi	= 15%
⇒ Tugas-2+Absensi	= 15%

REFERENSI

- Julien Bayle, “C Programming for Arduino”, Packt Publishing Ltd, Birmingham, May 2013.
- James Floyd K & Harold T , “Arduino Adventure Escape from Gemini Station”, Apress, 2013.
- Famosa Studio Arduino Starter Kit Manual – V1.0, Famosa Studio, 2013.
- Martin E, Joshua N, & Jordan H, “Arduino in Action“,Manning Publications.Co, USA, 2013.
- Jack Purdum, “Beginning C for Arduino, Learn C Programming for the Arduino and Compatible Microcontrollers”, Apress, 2012.
- John-David Warren, Josh Adams, and Harald Molle, “Arduino Robotics”, Springer, New York, 2011.
- _____, Sistem minimum Arduino Uno/ATmega328, Instruction Manual, 2010.

MATERI

- 1. PENDAHULUAN**
- 2. ANTARMUKA MIKROKONTROLER DENGAN LIGHT EMITTING DIODE (LED)**
- 3. ANTARMUKA MIKROKONTROLER DENGAN SAKLAR**
- 4. ANTARMUKA MIKROKONTROLER DENGAN LED DOT MARIK**
- 5. ANTARMUKA MIKROKONTROLER DENGAN KEYPAD**
- 6. ANTARMUKA MIKROKONTROLER DENGAN SEVEN SEGMENT (7-S)**
- 7. ANTARMUKA MIKROKONTROLER DENGAN LCD 2x16**
- 8. PEMROGRAMAN MELODY**
- 9. ANALOG INPUT (ADC)**
- 10. KOMUNIKASI SERIAL**
- 11. ANTARMUKA MIKROKONTROLER DENGAN LM 35**
- 12. ANTARMUKA MIKROKONTROLER DENGAN LDR**
- 13. ANTARMUKA MIKROKONTROLER DENGAN LAMPU AC 220V**
- 14. ANTARMUKA MIKROKONTROLER DENGAN MOTOR DC**
- 15. ANTARMUKA MIKROKONTROLER DENGAN SENSOR ULTRASONIC**
- 16. Demo Tugas Proyek Semester**

1. PENDAHULUAN

1.1. Pengantar Mikroprosesor

- **Mikroprosesor** : bagian CPU dari sebuah komputer, tanpa *Memory*, I/O, dan peripheral lain.
 - Supaya dapat bekerja, mikroprosesor memerlukan perangkat pendukung seperti RAM, ROM dan I/O.
- **Mikrokomputer** : kombinasi mikroprosesor dengan I/O dan *Memory* (RAM/ROM).
- **Mikrokontroler** : Mikrokomputer yang dibuat dalam bentuk *Single Chip* yaitu *Single Chip Microcomputer* (SCM).

□ Sejarah mikroprosesor

- **Alat penghitung**

- Menghitung biji jagung dengan menggunakan jari (digit)

- ***Sipoa***

- Tahun 1642 **Blaise Pascal** : mesin kalkulasi digital pertama kali

- Tahun 1671, **Gottfried Wilhelm von Leibniz** mesin hitung untuk operasi perkalian.

- Tahun 1820, **Thomas Colmar (Charles Xavier Thomas)** kalkulator mekanik mampu tambah, kurang, kali, & bagi.

- Tahun 1812, **Charles Babbage**, kalkulator otomatis menjadikan ciri dari komputer elektronik masa kini.

Tahun 1822 membangun model aktif kecil untuk demonstrasi.

- 1941 kalkulator untuk membangkitkan *mathematical table*. Kalkulator elektronik pertama adalah ENIAC (for Electrical Numerical Integrator And Calculator), menggunakan 18000 tabung ruang hampa, 1800 kaki luas lantai bujur sangkar, dengan *power* 180 kW.
- Tabung ruang hampa digantikan dengan transistor pada tahun 1955, digantikan oleh *Integrated Circuit* (IC) di tahun 1964.
- Tahun 1971 *Central Processing Unit* (CPU) - sekarang

❑ Perbedaan mikrokomputer dengan mikrokontroler (SCM) :

- Penggunaan perangkat I/O dan media penyimpan program

Mikrokomputer : menggunakan ***disket*** atau ***Harddisk***.

Mikrokontroler : menggunakan EPROM

❑ Keuntungan mikrokontroler dibandingkan dengan mikroprosesor

Mikrokontroler terdapat memori (RAM/ROM) dan peralatan I/O

Perbedaan *MIKROPROSESSOR* dan *MIKROKONTROLLER*

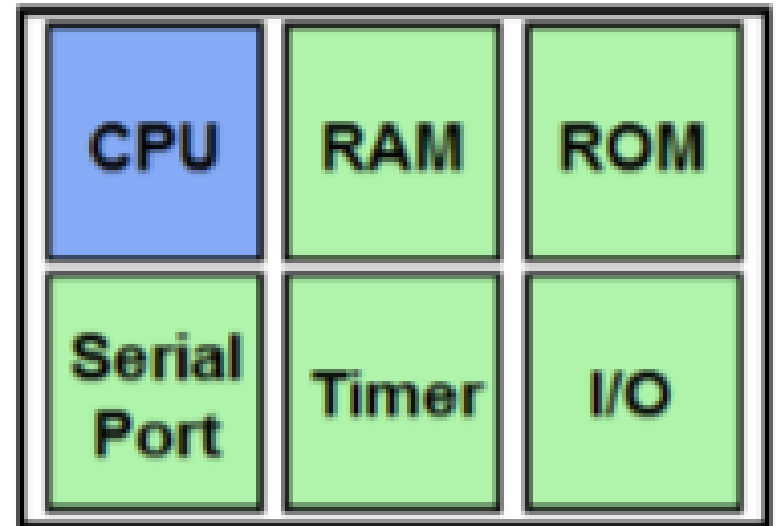
A. MIKROPROSESSOR

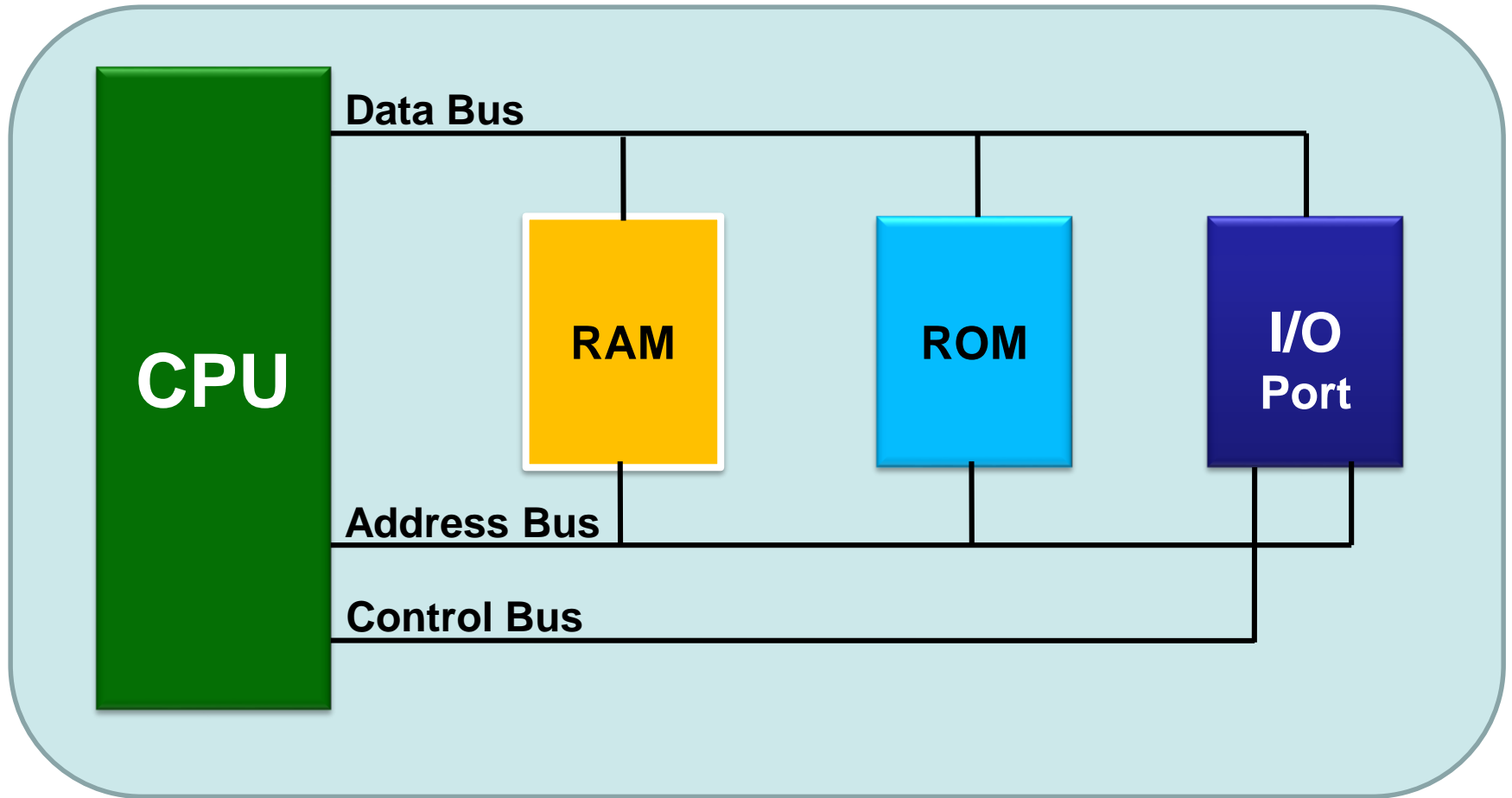
- ✓ Sebuah single chip yang berisi CPU (Central Processing Unit).
- ✓ Untuk membentuk sebuah minimum system mikrokomputer dibutuhkan peralatan pendukung:
 - RAM (Random Acces Memory)
 - ROM (Read Only Memory)
 - I/O (Unit Input / Output)

B. MIKROKONTROLLER

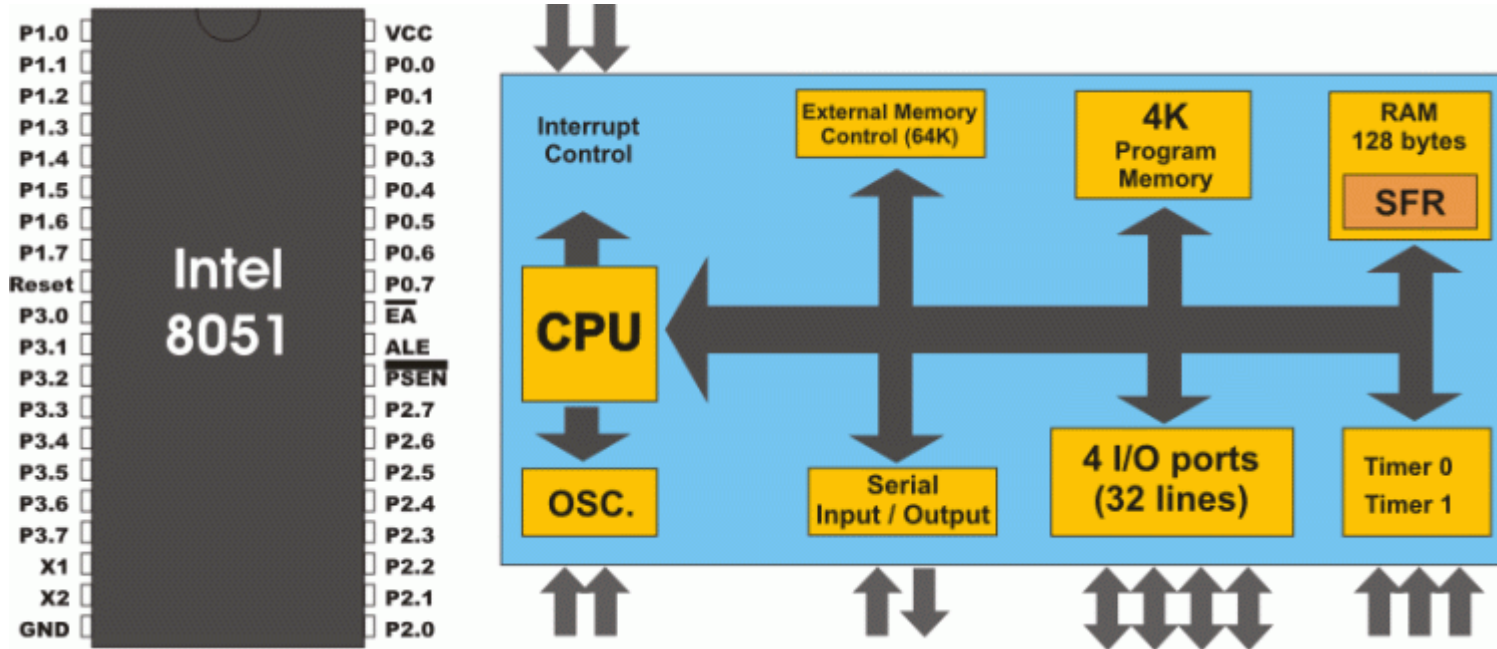
Sebuah single chip yang di dalamnya sudah berisi

- CPU (Central Processing Unit)**
- RAM (Random Acces Memory)**
- ROM (Read Only Memory)**
- I/O (Unit Input / Output)**
- Timer/Counter**
- Serial Port.**



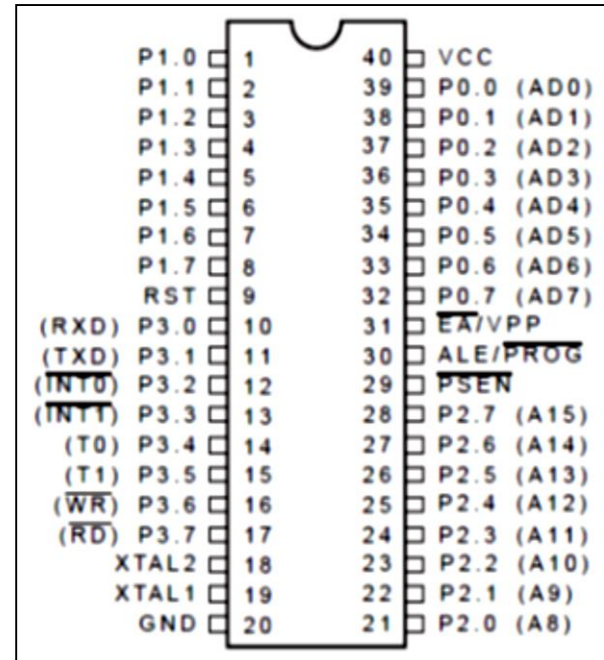


Blok Diagram MIKROPROSESSOR

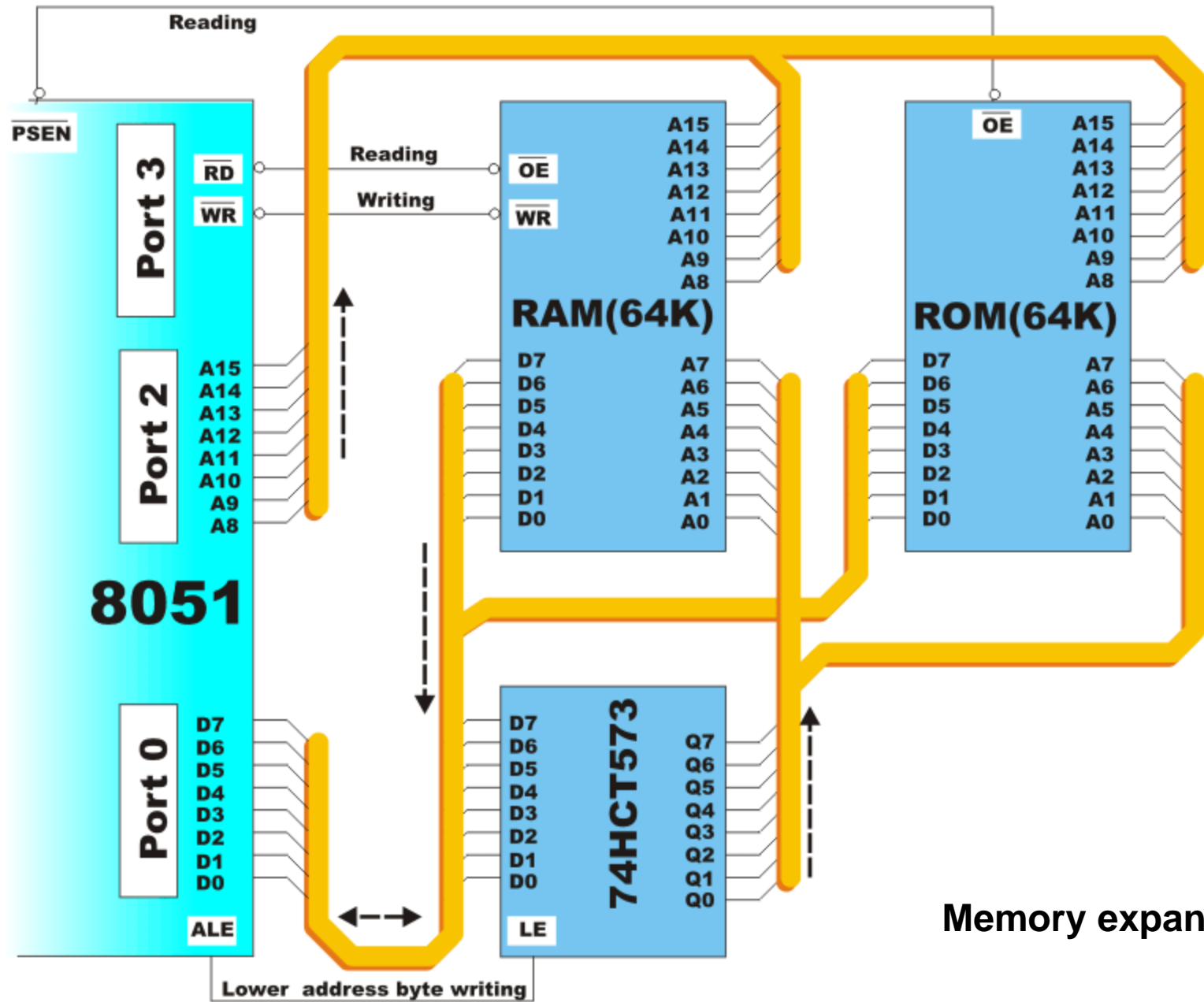


- ROM = 4 Kb
- RAM = 128b
- 4 ports = 32 jalur

- Keseluruhan sistem komputer yang dimasukkan ke dalam satu chip (SCM)



Blok Diagram MIKROKONTROLER



Memory expansion

Prinsip Kerja Mikrokontroler

- **Sebuah mikrokontroler tidak dapat bekerja bila tidak diberikan program kepadanya.**
- **Sistem kerja mikrokontroler dapat dirubah setiap saat sesuai dengan program yang diberikan kepadanya.**
- **Instruksi-instruksi perangkat lunak berbeda untuk masing-masing jenis mikrocontroller.**
- **Sebuah mikrokontroler tidak dapat memahami instruksi-instruksi yang berlaku pada mikrokontroler jenis lain.**

Contoh:

Mikrokontroler buatan Intel dengan mikrokontroler buatan Zilog memiliki perangkat instruksi yang berbeda

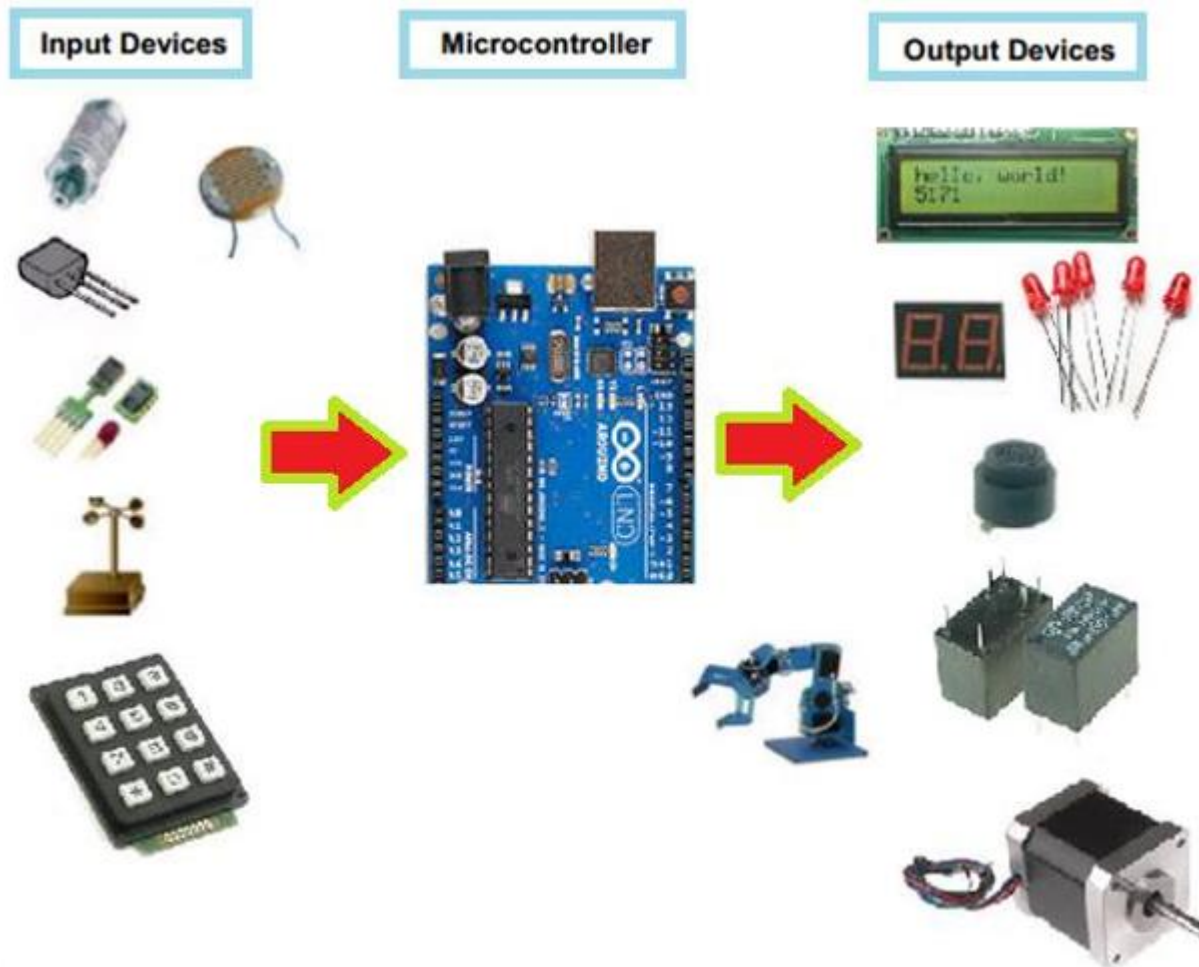
LD A,E → Z80

MOV AL,DL → Intel 80x86/88

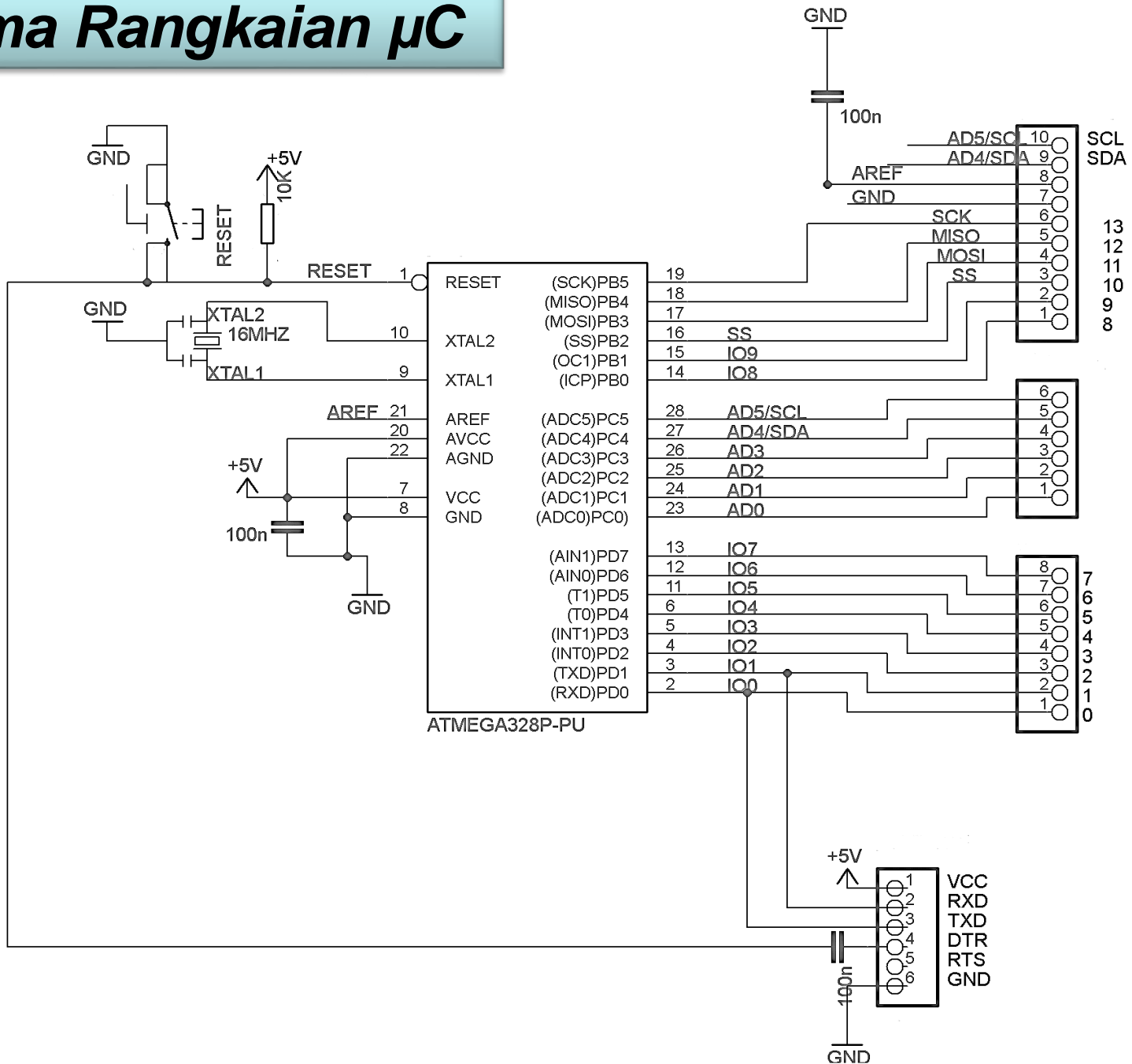
MOV R0,A → MCS-51

- Instruksi-instruksi pada mikrokontroler dikenal sebagai bahasa pemrograman sistem mikrokontroler
- Sebuah Bahasa pemrograman untuk beberapa jenis mikrokontroler misalnya: Pemrograman Bahasa C.

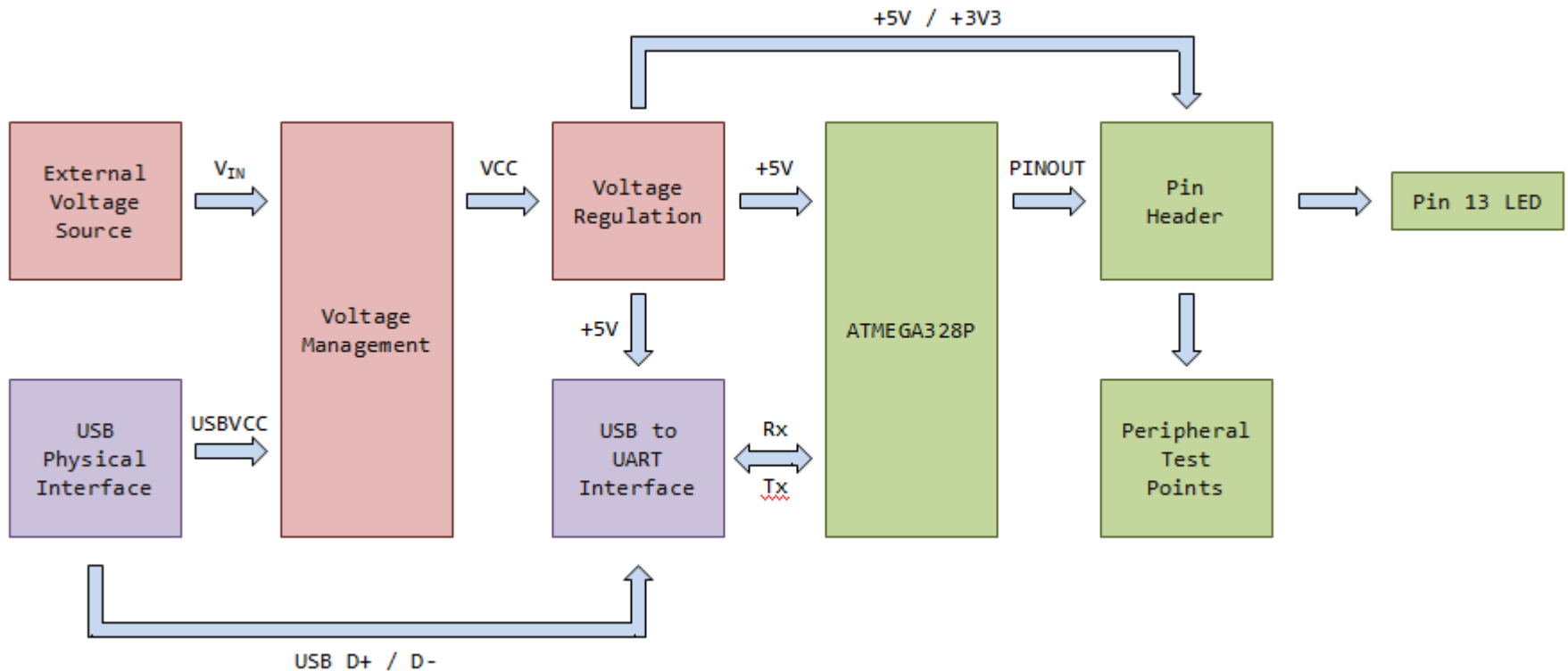
Teknik Antarmuka Menggunakan Mikrokontroler (μC)



Skema Rangkaian μC



Blok Diagram μC



- Bahasa pemrograman yang digunakan: Pemrograman Bahasa C.

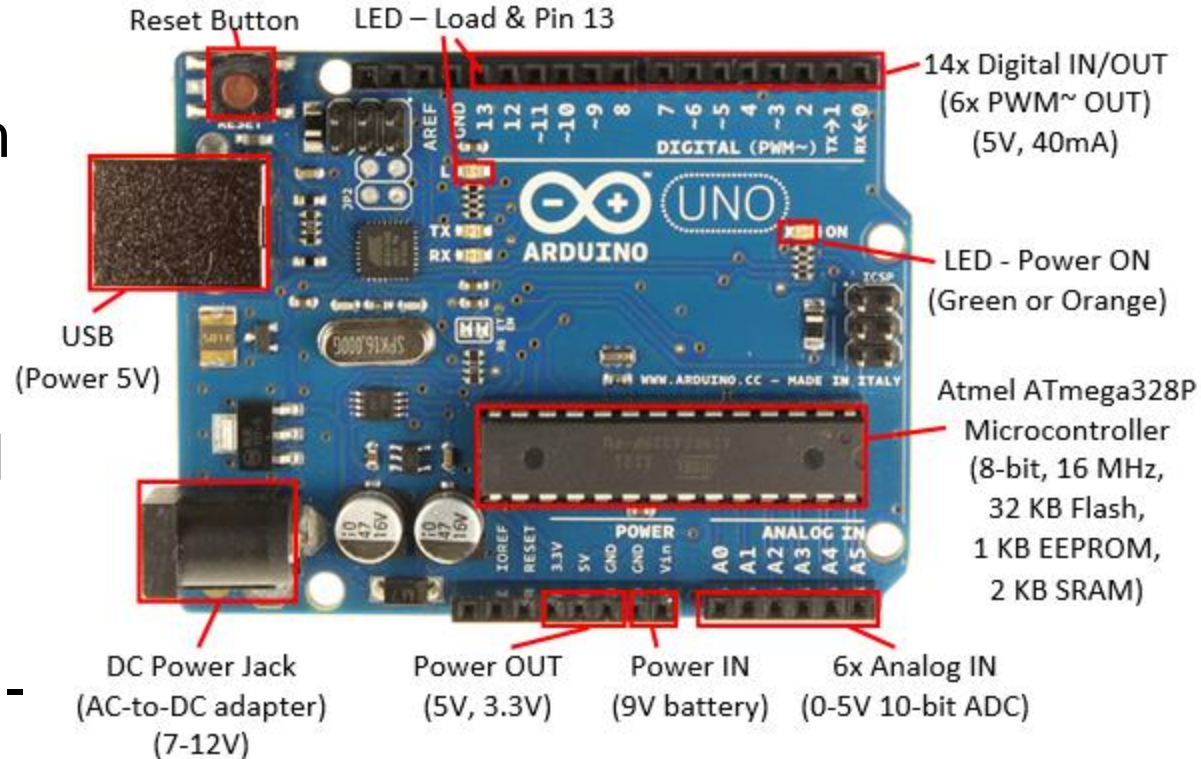
2. ANTARMUKA MIKROKONTROLER DENGAN LIGHT EMITTING DIODE (LED)

TUJUAN

- Dapat membuat proses kontrol display LED sebagai output
- Membuat Rangkaian Antarmuka Mikrokontroler dengan LED

DASAR TEORI

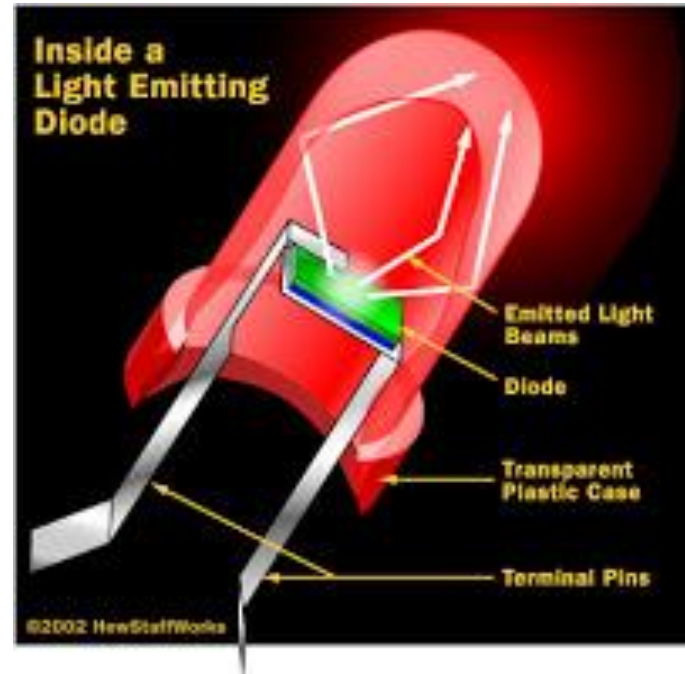
- Pada papan Arduino Uno terdapat sebuah LED kecil yang terhubung ke pin digital 13.
- Untuk menguji board arduino user dapat membuat program menyalakan LED pin-13.



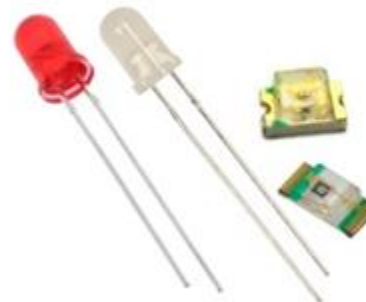
DASAR TEORI

LED

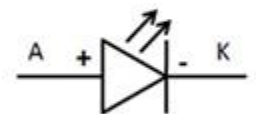
- LED (Light Emitting Diode) adalah sebuah sumber cahaya yang terbuat dari semikonduktor.
- LED digunakan sebagai lampu indikator dalam beberapa piranti, banyak digunakan sebagai penerangan/lampu.



Bentuk LED

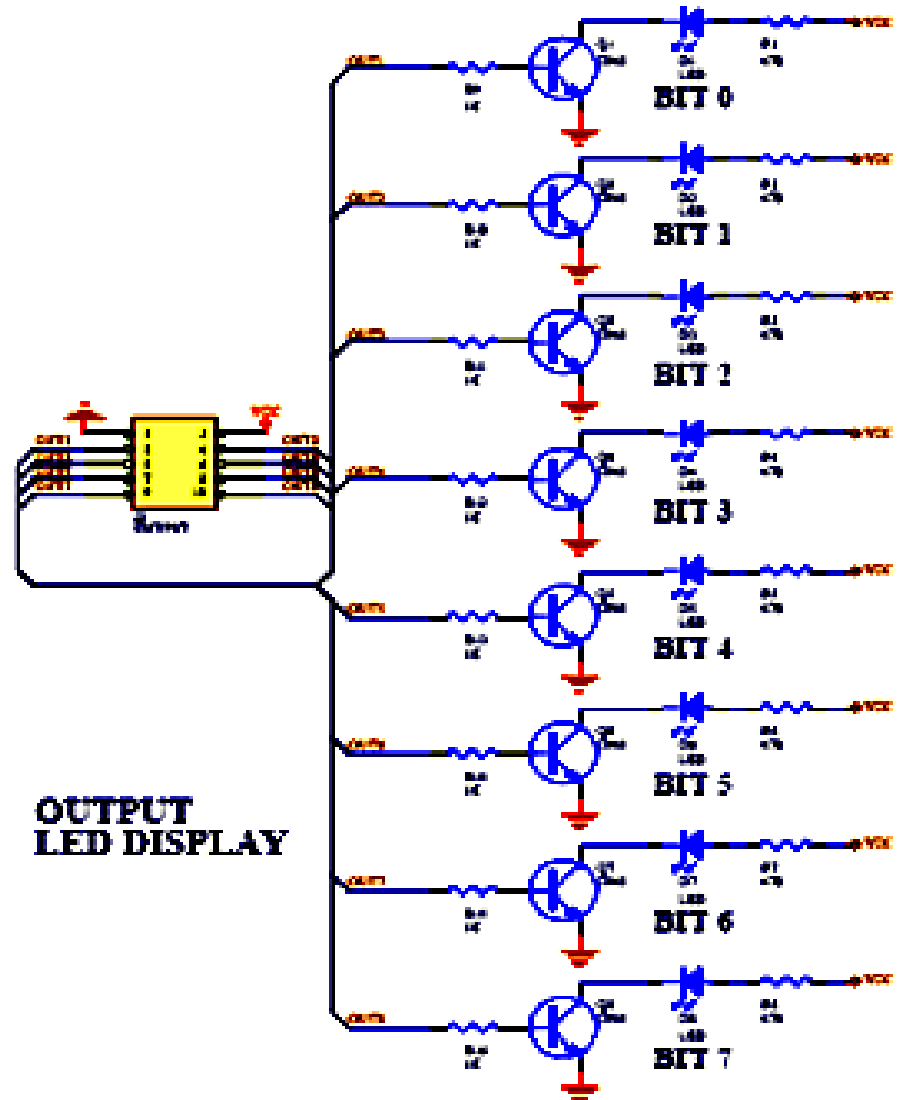


Simbol LED



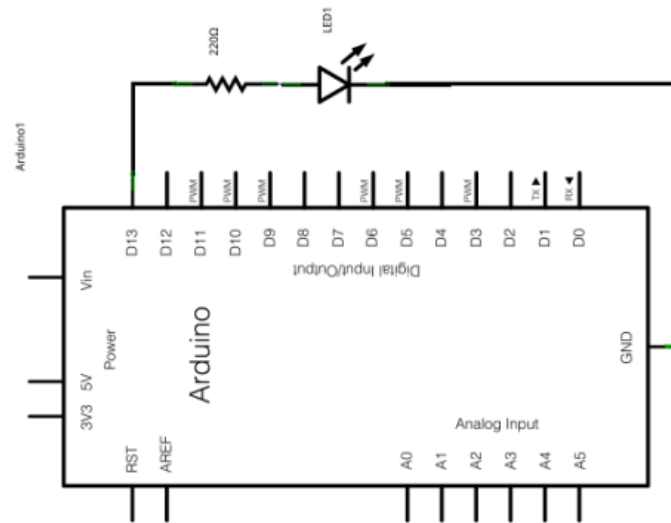
DASAR TEORI

- Untuk menyalakan sebuah LED perlu rangkaian tambahan rangkaian yang berupa sebuah transistor sebagai saklar dan dua buah resistor untuk pembatas arus.



DASAR TEORI

- Dalam modul I/O yang dipakai dalam praktikum, rangkaian LED tersebut dihubungkan ke sebuah pin pada board arduino.



DASAR TEORI

- Penulisan program dengan bahasa C,

- ✓ Fungsi Setup

```
void setup()  
{  
  pinMode(13,OUTPUT);  
  Serial.begin(9600);  
}
```

**set digital pin 13
sebagai output dan
mengkonfigurasi
komunikasi serial
dengan baud rate 9600.**

DASAR TEORI

- Penulisan program dengan bahasa C,

- ✓ Fungsi Setup

```
void setup(){  
  // nothing to setup  
}
```

Tidak melakukan setting apapun, diperlukan ketika terdapat kesalahan proses saat verifikasi atau upload sketsa

DASAR TEORI

- Penulisan program dengan bahasa C,

- ✓ Fungsi loop

```
void loop()  
{  
digitalWrite(13, HIGH);  
delay(1000);  
digitalWrite(13,LOW);  
delay(1000);  
}
```

- Loop u/ LED berkedip
- rutinitas atau fungsi perulangan disebut proses perulangan sampai kondisi terpenuhi

DASAR TEORI

PERINTAH DASAR MENGELUARKAN DATA

- Penulisan program dengan bahasa C,
 - ✓ `int led = 13;`
 - ✓ `pinMode(led, OUTPUT);`
 - ✓ `digitalWrite(led, HIGH);`
 - ✓ `digitalWrite(led, LOW);`

DASAR TEORI

PERINTAH DASAR MENGELUARKAN DATA

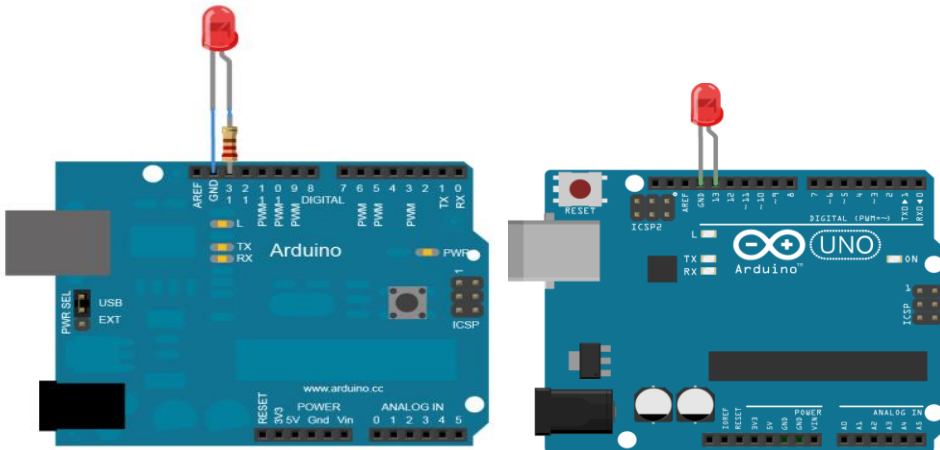
- Penulisan program dengan bahasa C,
 - ✓ Menggunakan for () loops untuk menjalankan sejumlah baris kode berulang-2
 - ✓ Menggunakan arrays[] untuk mengatur penggunaan variabel dengan lebih mudah (array = grup dari variabel).

Hardware :

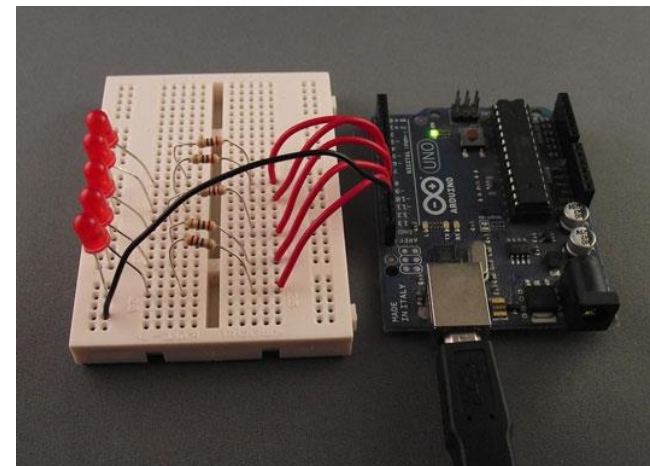
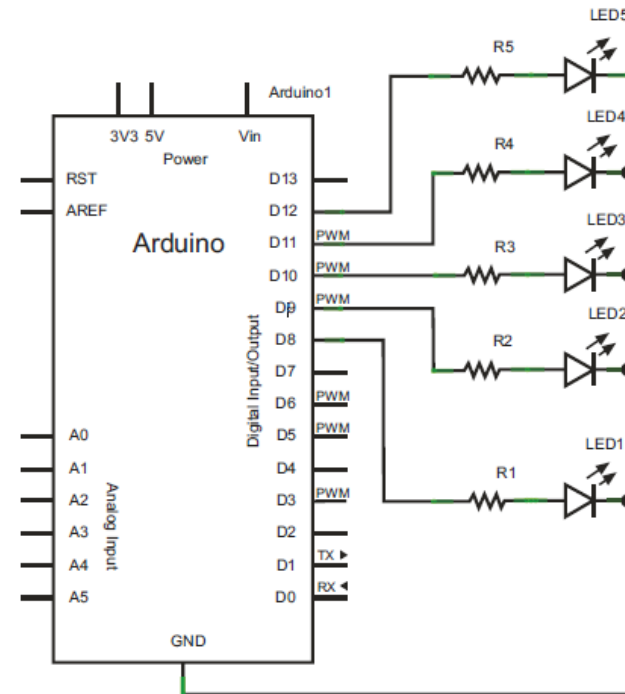
- Arduino Uno Board
- 8 buah LED
- Kabel jumper

Rangkaian :

A. Antarmuka 1 LED



B. Antarmuka 5 LED

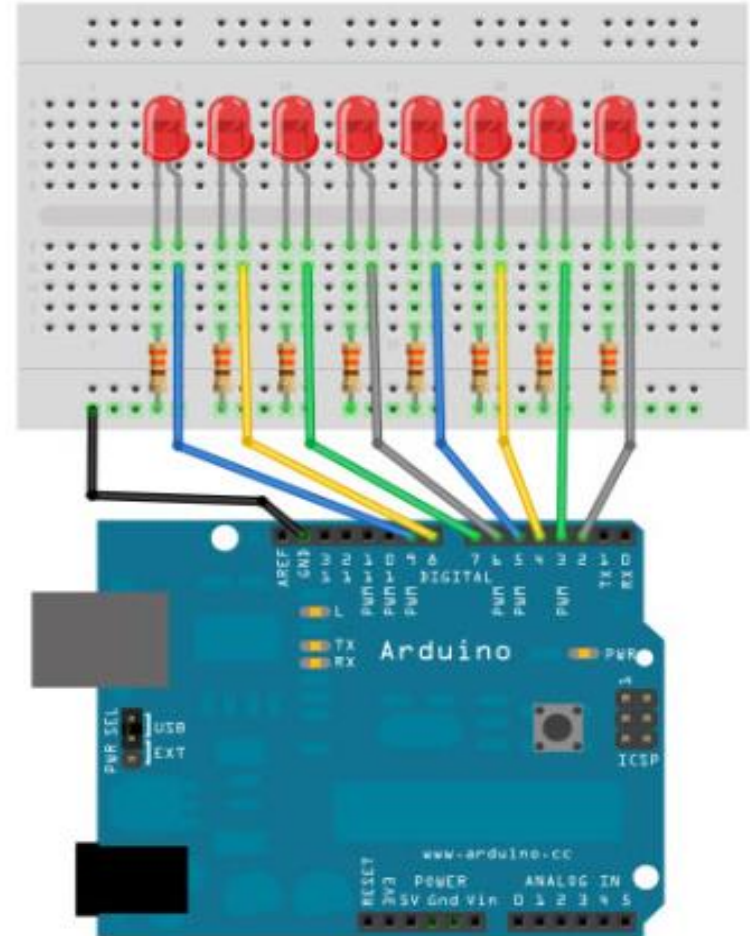
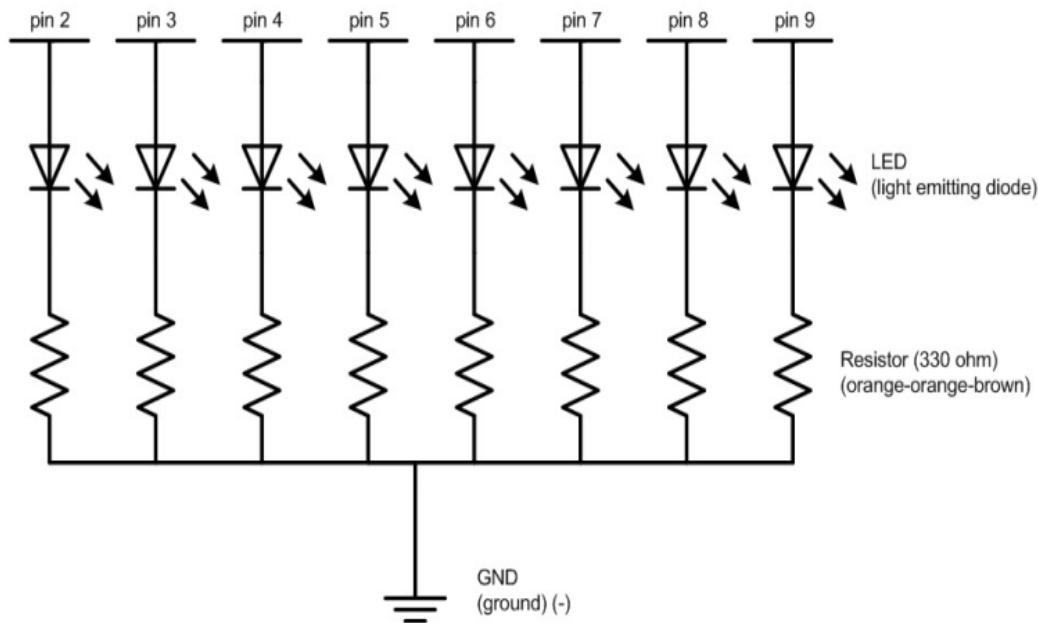


Hardware :

- Arduino Uno Board
- 8 buah LED
- Kabel jumper

Rangkaian :

C. Antarmuka 8 LED



PROGRAM: (1 LED)

```
int led = 13;

void setup() {
  pinMode(led, OUTPUT);
}

void loop() {
  digitalWrite(led, HIGH);
  delay(1000); // (1000ms=1s )
  digitalWrite(led, LOW);
  delay(1000);
}
```

```
int led = 9;
int brightness = 0;
int fadeAmount = 5;

void setup() {
  pinMode(led, OUTPUT);
}

void loop() {
  analogWrite(led, brightness);
  brightness = brightness + fadeAmount;
  if (brightness == 0 || brightness == 255)
    { fadeAmount = -fadeAmount ; }
  delay(30);
}
```

PROGRAM: (5 LED)

```
int ledArray[] = {8, 9, 10, 11, 12};  
int count = 0;  
int timer = 75;  
void setup(){  
  for (count=0;count<5;count++){  
    pinMode(ledArray[count], OUTPUT);  
  }  
}  
void loop(){  
  for (count=0;count<5;count++){  
    digitalWrite(ledArray[count], HIGH);  
    delay(timer);  
    digitalWrite(ledArray[count], LOW);  
    delay(timer);  
  }  
}
```

LED array

A for loop

digitalWrite writes LOW or HIGH

The diagram consists of a rectangular box containing C++ code. To the right of the box, there are three text labels with red arrows pointing to specific lines of code. The first label, 'LED array', has an arrow pointing to the first line of code: 'int ledArray[] = {8, 9, 10, 11, 12};'. The second label, 'A for loop', has an arrow pointing to the first line of the 'setup()' function's for loop: 'for (count=0;count<5;count++){'. The third label, 'digitalWrite writes LOW or HIGH', has two arrows: one pointing to the 'digitalWrite(ledArray[count], HIGH);' line and another pointing to the 'digitalWrite(ledArray[count], LOW);' line in the 'loop()' function.

PROGRAM: (8LED)

//LED Pin Variables

```
int ledPins[] = {2,3,4,5,6,7,8,9};
```

```
void setup()
```

```
{
```

```
  for(int i = 0; i < 8; i++){
```

```
    pinMode(ledPins[i],OUTPUT);
```

```
  }
```

```
}
```

```
void loop()
```

```
{
```

```
  oneAfterAnotherLoop();
```

```
}
```

```
void oneAfterAnotherLoop() {
```

```
  int delayTime = 100;
```

```
  for(int i = 0; i <= 7; i++) {
```

```
    digitalWrite(ledPins[i], HIGH);
```

```
    delay(delayTime);
```

```
  }
```

```
  for(int i = 7; i >= 0; i--) {  
    digitalWrite(ledPins[i], LOW);  
    delay(delayTime);
```

```
  }
```

```
}
```

```
void oneOnAtATime() {
```

```
  int delayTime = 100;
```

```
  for(int i = 0; i <= 7; i++) {
```

```
    int offLED = i - 1;
```

```
    if(i == 0) {
```

```
      offLED = 7;
```

```
    }
```

```
    digitalWrite(ledPins[i], HIGH);
```

```
    digitalWrite(ledPins[offLED],  
    LOW);
```

```
    delay(delayTime);
```

```
  }
```

```
}
```

PROGRAM: (8LED)

```
void inAndOut() {
  int delayTime = 100;
  for(int i = 0; i <= 3; i++) {
    int offLED = i - 1;
    if(i == 0) {
      offLED = 3;
    }
    int onLED1 = 3 - i;
    int onLED2 = 4 + i;
    int offLED1 = 3 - offLED;
    int offLED2 = 4 + offLED;
    digitalWrite(ledPins[onLED1], HIGH);
    digitalWrite(ledPins[onLED2], HIGH);
    digitalWrite(ledPins[offLED1], LOW);
    digitalWrite(ledPins[offLED2], LOW);
    delay(delayTime);
  }
}
```

```
for(int i = 3; i >= 0; i--) {
  int offLED = i + 1;
  if(i == 3) {
    offLED = 0; }
  int onLED1 = 3 - i;
  int onLED2 = 4 + i;
  int offLED1 = 3 - offLED;
  int offLED2 = 4 + offLED;
  digitalWrite(ledPins[onLED1], HIGH);
  digitalWrite(ledPins[onLED2], HIGH);
  digitalWrite(ledPins[offLED1], LOW);
  digitalWrite(ledPins[offLED2], LOW);
  delay(delayTime);
}
}
```

Latihan :

1. Buatlah program untuk mengontrol nyala LED-1 di pin 13 dan LED-2 di pin 9.

Jika LED-1 nyala, LED-2 padam

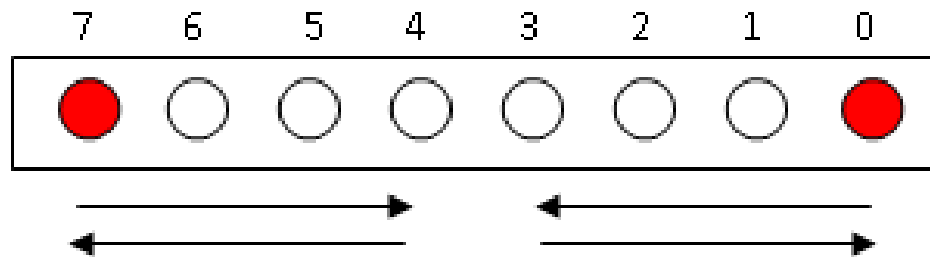
Jika LED-1 padam, LED-2 nyala.

LED-1 dan LED-2 nyala dan padam bergantian secara berulang-ulang.

2. Buatlah program untuk mengontrol nyala 5 LED bergerak dari LED-1 ke LED-5 dan sebaliknya dari LED-5 ke LED-1 berulang 3x kemudian 5 LED berkedip 2x.

Latihan :

3. Buatlah program untuk menyalakan LED dengan tampilan sebagai berikut :



4. Buatlah program untuk menyalakan LED 0 sampai LED 7 melompat 2, nyala LED dimulai dari LED 0, kemudian LED 2, 4, 6 kembali ke LED 0 berulang terus menerus.