MODUL 1 Pemrograman arduino dasar

A. Tujuan

1. Praktikan mampu melakukan pemrograman arduino dasar
2. Praktikan dapat melakukan pengawatan I/O pada project board
3. Praktikan mampu mengidentifikasi masalah pada pemrograman dasar pada saat dimana parameter waktu menjadi penting .
4. Praktikan mengerti mengapa RTOS dibutuhkan pada pemrograman *microcontroller.*

B. Alat & Bahan

1. Arduino UNO
2. Kit Sensor & Aktuator
3. Project board & jumper

C. Landasan Teori

Arduino is an open source *physical computing* platform for creating interactive objects that stand alone or collaborate with software on your computer. Arduino was designed for artists, designers, and others who want to incorporate physical computing into their designs without having to frst become electrical engineers. The Arduino hardware and software is open source. The opensource philosophy fosters a community that shares its knowledge generously. This is great for beginners as help is often available geographically nearby and always online, at many different skill levels, and on a bewildering array of topics. Example projects are presented not just as pictures of the fnished project, but includ instructions for making your own or as a starting point for incorporation into your derivative or related projects. The Arduino software, known as the Integrated Development Environment (IDE), is free. You can download it from *www.arduino.cc*. The Arduino IDE is based on the Processing language, which was developed to help artists create computer art without having to frst become software engineers. The Arduino IDE can run on Windows, Macintosh, and Linux. The Arduino board is inexpensive (about $30) and quite tolerant of common novice mistakes. If you do somehow manage to damage the main component on the Arduino Uno, it can be replaced for as little as $4. The Arduino project was developed in an educational environment and is a very popular educational tool. The same open source philosophy that created the community which generously shares information, answers, and projects also shares teaching methods, curricula, and other information. Arduino has a special mailing list to facilitate discussion among anyone interested in teaching with or about Arduino._*Banzi, Massimo; Shiloh, Michael Getting started with Arduino*

D. Langkah kerja percobaan & pengamatan

the Arduino program structure and we will learn more new terminologies used in the Arduino world. The Arduino software is open-source. The source code for the Java environment is released under the GPL and the C/C++ microcontroller libraries are under the LGPL.
Sketch − The first new terminology is the Arduino program called "sketch".
Structure − Arduino programs can be divided in three main parts: Structure, Values(variables and constants), and Functions. In this tutorial, we will learn about the Arduino software program, step by step, and how we can write the program without any syntax or compilation error.
Let us start with the Structure. **Software structure consist of two main functions −**

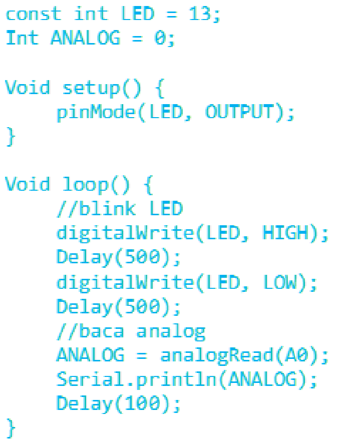
- Setup( ) function
- Loop( ) function

D.1 Percobaan 1 contoh kasus

Pada percobaan ini terdapat pembacaan sensor & blink led (*dua fungsi*) dengan alokasi pengeksekusian waktu tertentu mengandalkan *main fungtions* IDE arduino, yaitu void setup () & void loop().

Contoh kasus : Buatlah program aplikasi arduino untuk keperluan berikut:

- ○ Blink LED setiap 1 detik
- ○ Analog Read setiap 100 ms

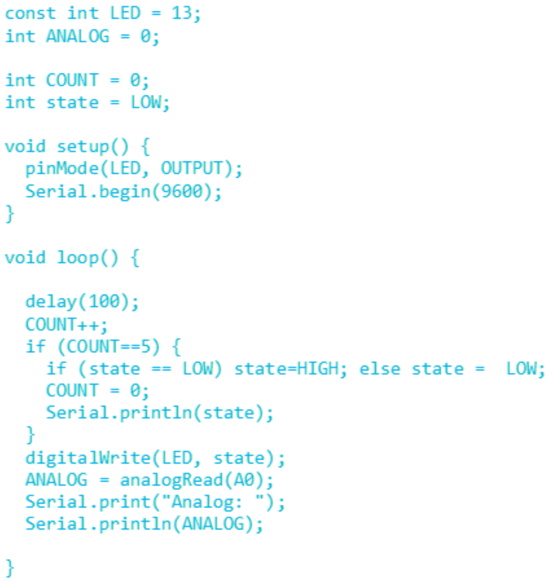


```
const int LED = 13;
Int ANALOG = 0;

Void setup() {
    pinMode(LED, OUTPUT);
}

Void loop() {
    //blink LED
    digitalWrite(LED, HIGH);
    Delay(500);
    digitalWrite(LED, LOW);
    Delay(500);
    //baca analog
    ANALOG = analogRead(A0);
    Serial.println(ANALOG);
    Delay(100);
}
```

Program di atas tidak menyelesaikan masalah pada contoh kasus karena ANALOG read terjadi setiap 1100ms, setelah pembacaan ke dua. Tidak sesuai dengan keperluan analog read setiap 100ms.

D.2 Percobaan 2 alternatif dari contoh kasus percobaan 1 (IF-ELSE statement)

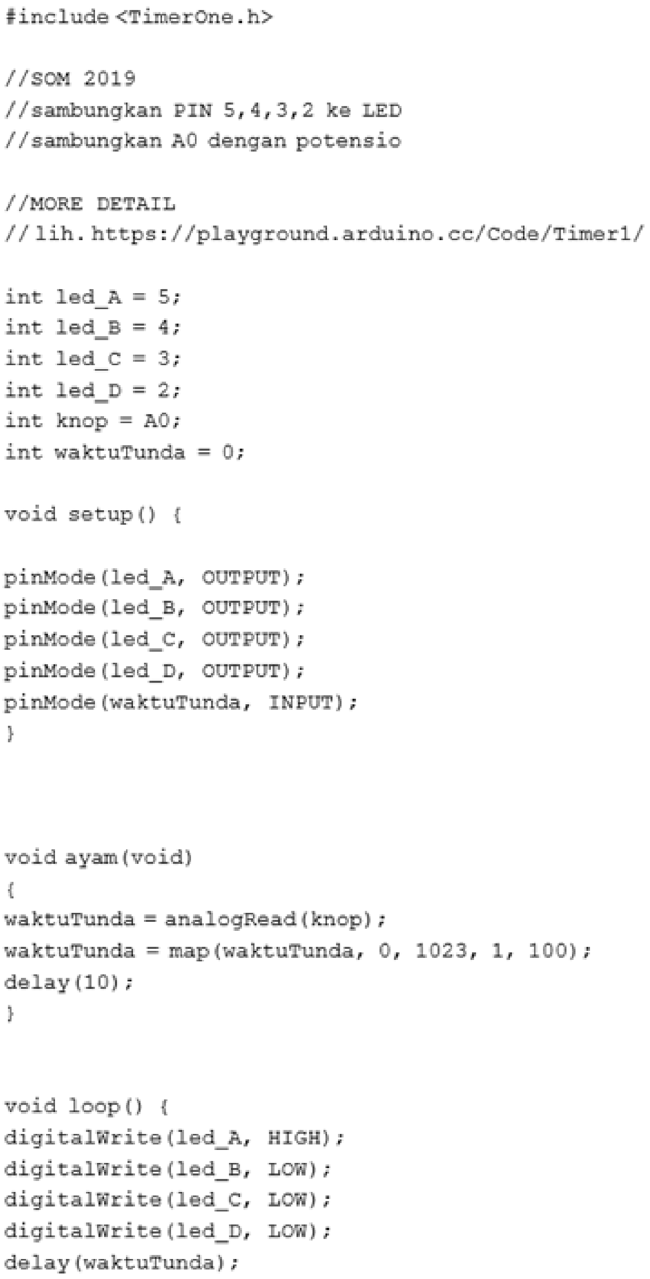
Pada percobaan 2 ini masih mengacu pada contoh kasus percobaan 1, namun mengandalkan IF-ELSE statement untuk memenuhi kebutuhan pembacaan.



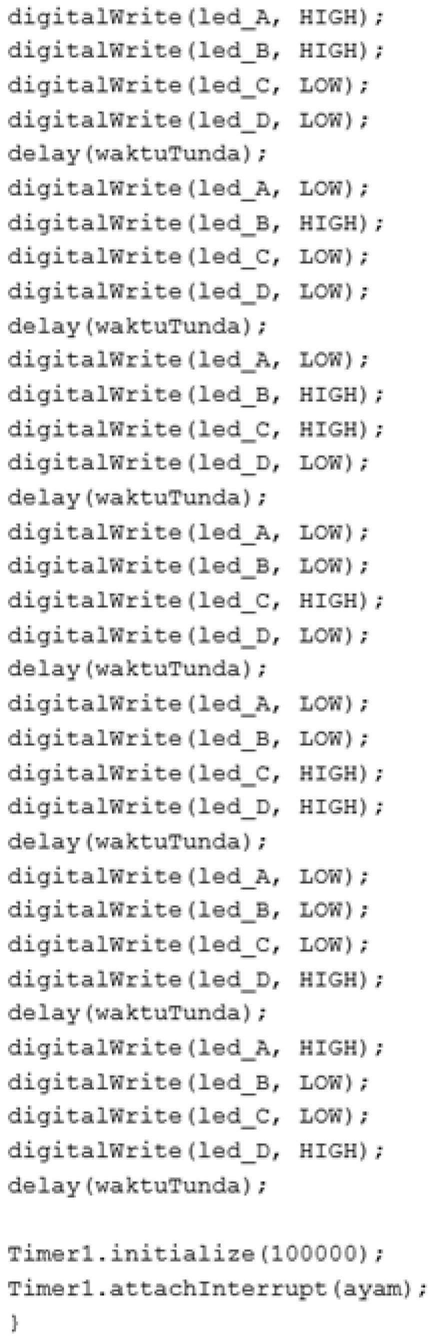
```
const int LED = 13;
int ANALOG = 0;

int COUNT = 0;
int state = LOW;

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

void loop() {

  delay(100);
  COUNT++;
  if (COUNT==5) {
    if (state == LOW) state=HIGH; else state =  LOW;
    COUNT = 0;
    Serial.println(state);
  }
  digitalWrite(LED, state);
  ANALOG = analogRead(A0);
  Serial.print("Analog: ");
  Serial.println(ANALOG);

}
```

Program di atas dapat menyelesaikan masalah **tetapi** mengandung kesalahan waktu pada hitungan kelima.

D.3 Percobaan 3 pengenalan Interupt



```
#include <TimerOne.h>

//SOM 2019
//sambungkan PIN 5,4,3,2 ke LED
//sambungkan A0 dengan potensio

//MORE DETAIL
// lih. https://playground.arduino.cc/Code/Timer1/

int led_A = 5;
int led_B = 4;
int led_C = 3;
int led_D = 2;
int knop = A0;
int waktuTunda = 0;

void setup() {

pinMode(led_A, OUTPUT);
pinMode(led_B, OUTPUT);
pinMode(led_C, OUTPUT);
pinMode(led_D, OUTPUT);
pinMode(waktuTunda, INPUT);
}



void ayam(void)
{
waktuTunda = analogRead(knop);
waktuTunda = map(waktuTunda, 0, 1023, 1, 100);
delay(10);
}


void loop() {
digitalWrite(led_A, HIGH);
digitalWrite(led_B, LOW);
digitalWrite(led_C, LOW);
digitalWrite(led_D, LOW);
delay(waktuTunda);
```



```
digitalWrite(led_A, HIGH);
digitalWrite(led_B, HIGH);
digitalWrite(led_C, LOW);
digitalWrite(led_D, LOW);
delay(waktuTunda);
digitalWrite(led_A, LOW);
digitalWrite(led_B, HIGH);
digitalWrite(led_C, LOW);
digitalWrite(led_D, LOW);
delay(waktuTunda);
digitalWrite(led_A, LOW);
digitalWrite(led_B, HIGH);
digitalWrite(led_C, HIGH);
digitalWrite(led_D, LOW);
delay(waktuTunda);
digitalWrite(led_A, LOW);
digitalWrite(led_B, LOW);
digitalWrite(led_C, HIGH);
digitalWrite(led_D, LOW);
delay(waktuTunda);
digitalWrite(led_A, LOW);
digitalWrite(led_B, LOW);
digitalWrite(led_C, HIGH);
digitalWrite(led_D, HIGH);
delay(waktuTunda);
digitalWrite(led_A, LOW);
digitalWrite(led_B, LOW);
digitalWrite(led_C, LOW);
digitalWrite(led_D, HIGH);
delay(waktuTunda);
digitalWrite(led_A, HIGH);
digitalWrite(led_B, LOW);
digitalWrite(led_C, LOW);
digitalWrite(led_D, HIGH);
delay(waktuTunda);

Timer1.initialize(100000);
Timer1.attachInterrupt(ayam);
}
```

- *void ... Or void ... (void)*
  The void keyword is used only in function declarations. It indicates that the function is expected to return no information to the function from which it was called. More explanation about void, see https://arduino.stackexchange.com/questions/6460/difference-between-void-setup-and-void-setupvoid

- *initialize(period)*
  You must call this method first to use any of the other methods. You can optionally specify the timer's period here (in microseconds), by default it is set at 1 second. Note that this breaks analogWrite() for digital pins 9 and 10 on Arduino.

- attachInterrupt(function, period)
  Calls a function at the specified interval in microseconds. Be careful about trying to execute too complicated of an interrupt at too high of a frequency, or the CPU may never enter the main loop

and your program will 'lock up'. Note that you can optionally set the period with this function if you include a value in microseconds as the last parameter when you call it.

Alternatif ke tiga utuk contoh kasus percobaan 1 dengan menggunakan interrupt : Interrupt-timer (Timer ONe) : hanya satu timer, atau membuat timer interrupt dengan cara memodifikasi REGISTER TIMER dan OUTPUT COMPARE. Bisa menggunakan TIGA TIMER dengan pewaktuan yang berbeda. **dengan Interrupt program akan menjadi rumit** jika banyak hal harus dilakukan pada waktu berbeda-beda. **Solusinya adalah SISTEM OPERASI .** Untuk Arudino salah satu SITEM OPERASI yang dapat digunakan adalah freeRTOS (http://www.freertos.org), Dapat digunakan untuk membuat aplikasi MULTITASKING dalam arduino, akan dibahas lanjut pada modul selajutnya.

D.4 Percobaan 4 contoh pemrograman dengan 5 fungsi

Contoh program dari asisten

D.5 Percobaan 5 tugas mandiri

Buatlah 2 fungsi arduino dimana waktu pembacaan menjadi parameter penting, ubahlah tabel 1.1

Menjadi 5 fungsi pembacaan sensor / penggerakan aktuator, supaya lebih mudah pada percobaan ini,

Silahkan melihat example pada IDE arduino dan example bawaan library sensor yang ingin digunaan pada percobaan.

Tabel 1.1 Tabel Fungsi Percobaan 1

NO.	Sensor / Aktuator	Waktu pembacaan / pengeksekuisan (xx diganti digit NIM anda)
1.	Sensor 1	x000 ms
2.	Sensor 2	1x000 ms

Daftar Pustaka :

- https://github.com/bana-handaga/Sistem-OPerasi-Mikroprosesor-2018
- *Banzi, Massimo; Shiloh, Michael Getting started with Arduino*
- https://playground.arduino.cc/Code/Timer1
- *https://arduino.stackexchange.com/questions/6460/difference-between-void-setup-and-void-setupvoid*

FORMAT LAPORAN MODUL 1

NOMER MODUL
NAMA MODUL

NAMA		:
NIM		:
KELAS		:
TANGGAL PRAKT		:
TANGGAL ACC		:

A. Tujuan

B. Landasan Teori

- ***Jelaskan tentang arduino dan pemrograman dasarnya (wajib mengutip dari* https://www.tutorialspoint.com/arduino*)***
- ***Jelaskan dan carilah literasi datasheet tentang sensor-aktuator yang dipakai pada praktikum modul 1***

  Misal : motor servo 9g

  penjelasan misalnya meliputi gambar sensor / aktuator, Pin IO, Dimensi, spesifikasi sensor (lih. datasheet)

  Buatlah penjelasan secukupnya namun berbobot.
- *Jelaskan dan carilah literasi tentang library-library yang dipakai pada praktikum modul 1*

  Misal : Library arduino **#include <Servo.h> lih. github**

C. Hasil Praktikum

C.1 Percobaan 1 contoh kasus

C.1.1 Skema rangkaian

C.1.2 Foto rangkaian

C.1.3 Script program

C.2 Percobaan 2 alternatif dari contoh kasus percobaan 1 (IF-ELSE statement)

C.2.1 Script program

C.3 Percobaan 3 pengenalan interupt

C.3.1 Skema rangkaian

C.3.2 Script program

C.3.3 Foto hasil (rangkaian saat *running*)

C.4 Percobaan 4 contoh pemrograman dengan 5 fungsi

C.3.1 Skema rangkaian

C.3.2 Script program

C.3.3 Foto hasil (rangkaian saat *running*)

C.5 Percobaan 5 tugas mandiri

1C.5.0 Tabel I/O sensor-aktuator

C.5.1 Skema rangkaian

C.5.3 Script program

C.5.4 Foto hasil (rangkaian saat *running*)

A. Analisa Hasil

D.1 Percobaan 1 contoh kasus

D.1.1 Analisa script perbaris *(berilah komentar pada sketch code yang anda tulis)*

D.1.2 Analisa cara kerja rangkaian

D.2 Percobaan 2 alternatif dari contoh kasus percobaan 1 (IF-ELSE statement)

D.2.1 Analisa script perbaris *(berilah komentar pada sketch code yang anda tulis)*

D.2.2 Analisa cara kerja rangkaian

D.3 Percobaan 3 contoh kasus

D.3.1 Analisa script perbaris *(berilah komentar pada sketch code yang anda tulis)*

D.3.2 Analisa cara kerja rangkaian

D.4 Percobaan 4 contoh kasus

D.4.1 Analisa script perbaris *(berilah komentar pada sketch code yang anda tulis)*

D.4.2 Analisa cara kerja rangkaian

D.5 Percobaan 5 contoh kasus

D.5.1 Analisa script perbaris *(berilah komentar pada sketch code yang anda tulis)*

D.5.2 Analisa cara kerja rangkaian

B. Kesimpulan

Simpulkan praktikum modul 1 min. 3 paragraf secara komperensif