MODUL 6 Sistem Kendali PID : Kasus P dan D dengan EEPROM dan *Push Button*

1. JUDUL PRAKTIKUM

Sistem Kendali PID Kasus P dan D dengan EEPROM

2. MAKSUD DAN TUJUAN

Maksud dan tujuan dari praktikum ini adalah :

1. Mahasiswa dapat memahami fungsi dan cara kerja PID pada motor DC
2. Mahasiswa dapat membuat program untuk menggunakan EEPROM untuk penyimpanan data sensor yang telah dikalibrasi.
3. Mahasiswa dapat menggunakan peripheral berupa *push button* untuk menambah konstanta Kp dan Kd.

3. PARAMETER PENILAIAN

No.	Parameter	Persentase (%)
1.	Lembar Penilaian Praktikum	40%
2.	Jurnal/Laporan Praktikum	60%

4. PERALATAN DAN BAHAN

Alat dan Bahan :

1. Robot Kit Line Follower
2. Baterai LiPo 2-Cell 1300 mAh
3. Kabel Mini-USB
4. Arduino Nano
5. Battery Checker
6. Battery Balancer

Perangkat Lunak :

1. Software IDE Arduino
2. Software Proteus (untuk simulasi)

5. TEORI DASAR

5.1. Sistem Kendali PD

Teknik kendali proporsional-derivatif (PD) adalah pengendali yang merupakan gabungan antara teknik kendali proporsional (P) dengan teknik kendali derivatif (D). Gambar 1 merupakan gambar diagram blok sistem kendali PD.

Dalam penerapannya di software, kondisi ideal pada robot adalah bergerak maju lurus mengikuti garis, dengan kata lain error = 0 . Dari sini dapat diasumsikan bahwa Set Point (SP) / kondisi ideal adalah saat SP = 0. Nilai sensor yang dibaca oleh sensor disebut *Process Variable* (PV) / nilai aktual pembacaan. Menyimpangnya posisi robot dari garis disebut sebagai *error* (e), yang didapat dari $e = SP - PV$. Dengan mengetahui besar *error*, mikrokontroler dapat memberikan nilai PWM motor kiri dan kanan yang sesuai agar dapat menuju ke posisi ideal (SP = 0). Besarnya nilai PWM ini dapat diperoleh dengan menggunakan kontrol Proporsional (P), dimana $P = e \times Kp$ (Kp adalah konstanta proporsional yang nilainya diset sendiri dari hasil *tuning*/*trial and error*).

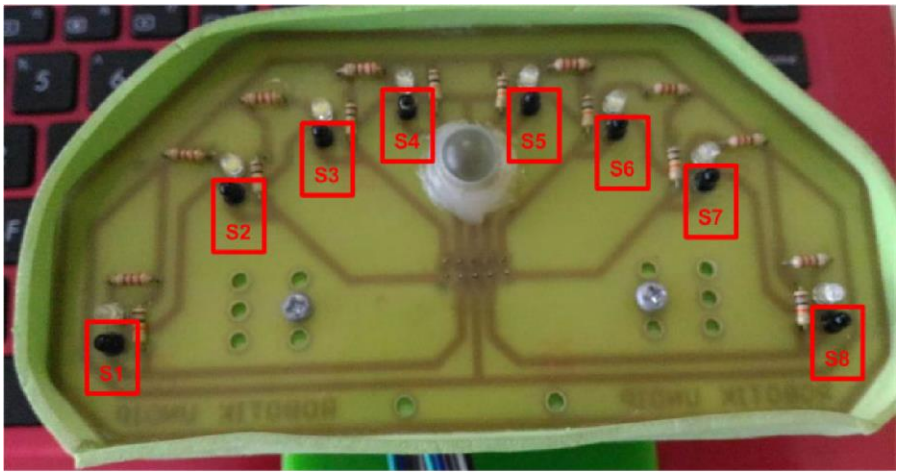
Jika pergerakan robot masih terlihat bergelombang, dapat ditambahkan parameter kontrol Derivatif (D). Kontrol D digunakan untuk mengukur seberapa cepat robot bergerak dari kiri ke kanan atau dari kanan ke kiri. Semakin cepat bergerak dari satu sisi ke sisi lainnya, maka semakin besar nilai D. Konstanta D (K_d) digunakan untuk menambah atau mengurangi imbas dari derivatif. Dengan mendapatkan nilai Kd yang tepat pergerakan sisi ke sisi yang bergelombang akibat dari kontrol proporsional dapat diminimalisasi. Dengan mendapatkan nilai K_d yang tepat pergerakan sisi ke sisi yang bergelombang akibat dari kontrol proporsional bisa diminimalisasi. Nilai D didapat dari $D = Kd/Ts \times rate$, dimana Ts adalah time sampling atau waktu cuplik dan rate = e(n) – e(n-1). Dalam program, nilai error (SP – PV) saat itu menjadi nilai last_error, sehingga rate didapat dari error – last_error

Agar konfigurasi atau hasil kalibrasi sensor tidak hilang ketika robot dimatikan atau kehilangan daya, EEPROM pada Arduino Nano dimanfaatkan untuk menyimpan data tersebut. Arduino Nano dengan mikrokontroler ATmega328 memiliki EEPROM dengan kapasitas 1024 byte. Kemudian untuk mempermudah user dalam memanfaatkan EEPROM untuk menyimpan dan menggunakan data, 4 buah push button yang disediakan pada robot digunakan.

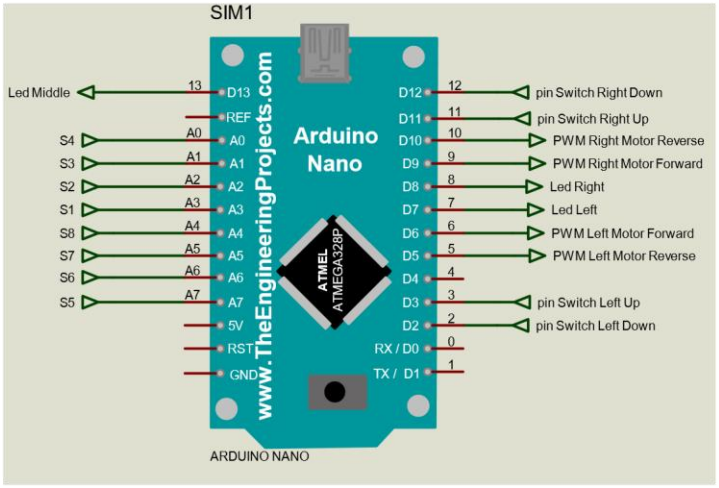
PROSEDUR PRAKTIKUM

A. Percobaan dalam praktikum 1.

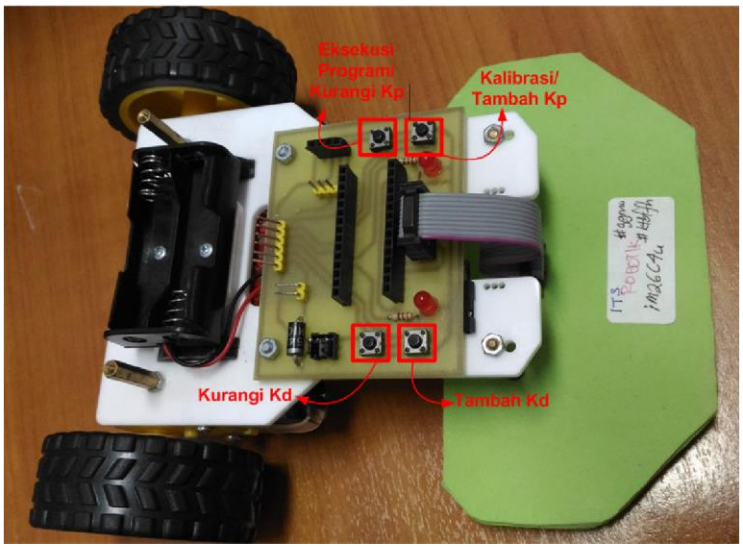
Kasus Percobaan



Gambar 1 Contoh susunan dan urutan sensor pada robot line follower.



Gambar 2 Pin Layout Arduino pada Robot Line Follower.



Gambar 3 Posisi dan fungsi push button pada robot line follower.

a. Modifikasi program pada praktikum sebelumnya dengan variable dan fungsi berikut ini :

- Tambahkan variabel integer dengan nama '**`state`**' dengan nilai awal adalah 0.

- Tambahkan variabel integer dengan nama '**setting**' dengan nilai awal adalah 0.
- Tambahkan variabel integer berupa array dengan nama '**peka**' dari 0 hingga 7 dengan nilai awal adalah 500.
- Tambahkan variabel integer dengan nama '**Kp**' dari 0 hingga 7 dengan nilai awal adalah 20.
- Tambahkan variabel integer dengan nama '**Kd**' dari 0 hingga 7 dengan nilai awal adalah 5.

- Di dalam **void setup** tambahkan :
- Baca nilai eeprom yang terdapat di EEPROM dengan alamat 0 hingga 7 (dengan ketentuan alamat 0 adalah untuk data nilai tengah kalibrasi sensor 1, alamat 1 untuk nilai tengah kalibrasi sensor 2, dan seterusnya hingga sensor 8) dengan perintah **EEPROM.read(0)** hingga **EEPROM.read(7)**, kemudian simpan didalam variabel **peka[0]** hingga **peka[7]** dengan mengkalikan dengan angka 4 (contoh : **peka[0]=EEPROM.read[0]*4**).
- Baca nilai eeprom yang terdapat di EEPROM dengan alamat 8 (ketentuan alamat 8 adalah data nilai Kp), kemudian simpan didalam variabel **Kp**.
- Baca nilai eeprom yang terdapat di EEPROM dengan alamat 9 (ketentuan alamat 9 adalah data nilai Kd), kemudian simpan didalam variabel **Kp**.
- Tampilkan data **peka[0]** hingga **peka[7], Kp** dan **Kd** yang berasal dari data EEPROM ke Serial Monitor.

- Di dalam **void loop** ditambahkan *conditional* sebagai berikut :

Referensi posisi push button dapat dilihat pada

- Jika tombol kiri belakang ditekan dan setting bernilai 0 maka **'state'** akan bernilai 1.
- Jika tombol kiri depan ditekan dan setting bernilai 0 maka **'state'** akan bernilai 2 dan 'setting' akan bernilai 1.
- Jika tombol kiri belakang ditekan dan setting bernilai 1 maka **'state'** akan bernilai 3.
- Jika tombol kiri depan ditekan dan setting bernilai 1 maka **'state'** akan bernilai 4.
- Jika tombol kanan belakang ditekan dan setting bernilai 1 maka **'state'** akan bernilai 5.
- Jika tombol kanan depan ditekan dan setting bernilai 1 maka **'state'** akan bernilai 6.

Perubahan state tersebut akan mengaktifkan sub program berikut :

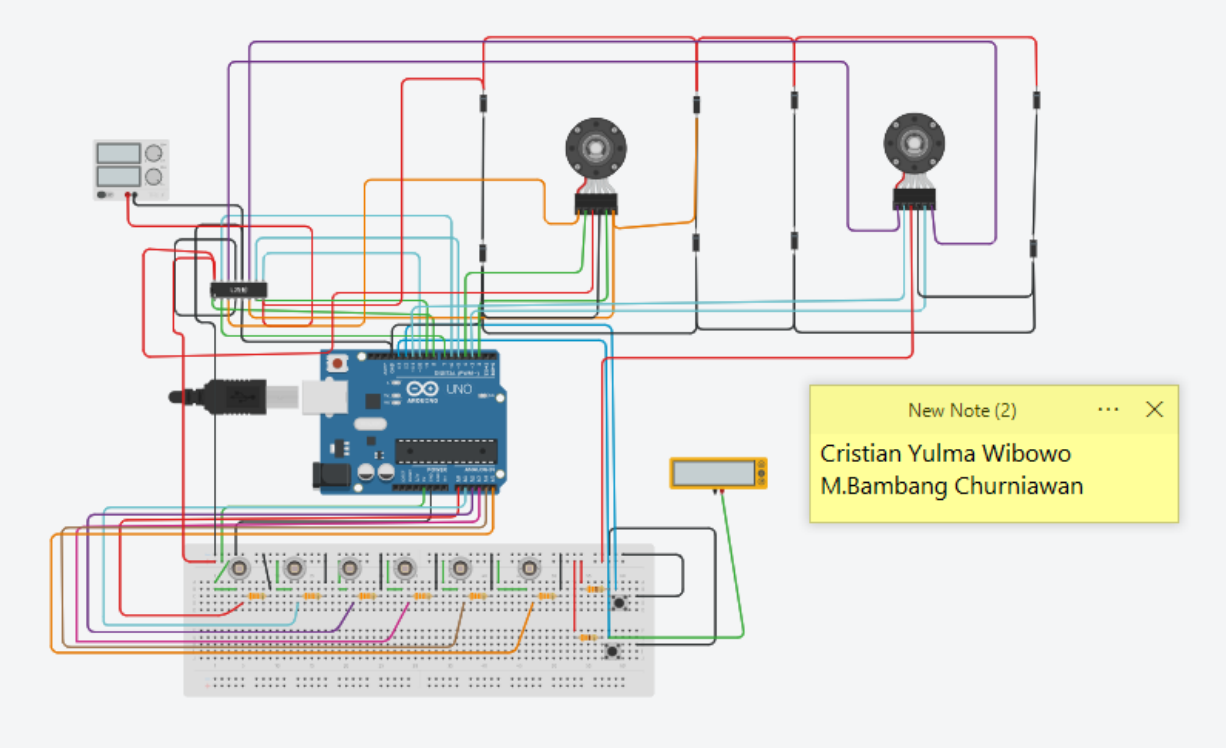
- Jika **'state'** bernilai 1 maka proses *auto calibration* berlangsung.

- Jika **'state'** bernilai 2 maka nilai terakhir variabel **'peka[0]'** hingga **'peka[7]'** disimpan ke dalam EEPROM dengan alamat 0 hingga 7 dan menjalankan perintah proses *line follower* dengan sistem kendali PID kasus P dan D pada praktikum sebelumnya dengan nilai **peka[0]** hingga **peka[7]**, **Kp** dan **Kd** yang diperoleh dari data EEPROM dan atau berasal dari hasil autocallibration.
- Perintah **EEPROM.write(alamat,value)** dengan value yang dibagi dengan 4. Jelaskan mengapa value harus dikali dan dibagi 4!
  Contoh : **EEPROM.write(0, peka[0]/4)**

- Jika '**state**' bernilai 3 maka LED kiri akan menyala dan setelah 1 detik LED kiri akan mati, mengurangi nilai Kp(contoh : Kp=Kp-1;), menyimpan nilai Kp kedalam EEPROM pada alamat 8.
-
- Jika '**state**' bernilai 4 maka LED kiri akan menyala dan setelah 1 detik LED kiri akan mati, menambah nilai Kp(contoh : Kp=Kp+1;), menyimpan nilai Kp kedalam EEPROM pada alamat 8.
-
- Jika **'state'** bernilai 5 maka LED kanan akan menyala dan setelah 1 detik LED kiri akan mati, mengurangi nilai Kd(contoh : Kd=Kd-1;), menyimpan nilai Kp kedalam EEPROM pada alamat 9
-
- Jika **'state'** bernilai 6 maka LED kanan akan menyala dan setelah 1 detik LED kiri akan mati, menambah nilai Kd(contoh : Kd=Kd+1;), menyimpan nilai Kp kedalam EEPROM pada alamat 9

b. Variasikan nilai konstanta Kp dan Kd pada program kemudian ujicoba pada lintasan dan amati hasil perubahan kedua konstanta tersebut pada robot *line follower*! Apa yang menjadi kesimpulan dari hasil percobaan ini?

c. *Screenshoot* keluaran serial monitor untuk setiap kondisi. Cetak dan tempelkan pada buku jurnal praktikum.




```
#include <EEPROM.h>

int ENA =9;
int ENB =10;

//Encoderin Motor 1
//Pin 2 adalah external interrupt Arduino
#define encoderPinA_1 2
#define encoderPinB_1 4

//Encoder PIN motor 2
#define encoderPinA_2 3
#define encoderPinB_2 11

//PIN Kontrol
#define motor_kiri1 7
#define motor_kiri2 8
#define motor_kanan1 5
#define motor_kanan2 6

//Deklarasi Variabel
```

```
//Untuk menampilkan serial port Serial
int encoderMotor1 = 0;
int encoderMotor2 = 0;

int alamat = 0;
int sensor[6];

//Konstanta PID
int baca_sensor[6];

//Initial Speed of Motor
int kecepatanSetPoint = 150;;

//Konstanta PID
float Kp = 1 ;
float Ki = 0 ;
float Kd = 1 ;

float error = 0,P = 0, I = 0 , D = 0 , PID_value = 0;
float lastError = 0;

/* Digunakan untuk Hardware Arduino Uno
int sensorMax[6] = {1023, 1023, 1023, 1023, 1023, 1023};
int sensorMin[6] = {0, 0, 0, 0, 0, 0};
*/

int sensorMax[6] = {687, 687, 687, 687, 687, 687};
int sensorMin[6] = {33, 33, 33, 33, 33, 33};
int sensorMid[6];
// Deklarasi Pin Sensor Photodiode
int sensor1 = A0;
int sensor2 = A1;
int sensor3 = A2;
int sensor4 = A3;
int sensor5 = A4;
int sensor6 = A5;
byte value;

int i;
```

```
//Fungsi external interupt
void doEncoderMotor1(){
        digitalRead(encoderPinA_1)?encoderMotor1--:encoderMotor1++;

}

void setup(){
        Serial.begin(9600);
    pinMode(encoderPinA_1, INPUT_PULLUP);
        pinMode(encoderPinB_1, INPUT_PULLUP);
        pinMode(encoderPinA_2, INPUT_PULLUP);
        pinMode(encoderPinB_2, INPUT_PULLUP);

  pinMode(ENA, OUTPUT);
  pinMode(ENB, OUTPUT);
  digitalWrite(ENA,  HIGH);
  digitalWrite(ENB,  HIGH);

  pinMode(motor_kiri1, OUTPUT);
  pinMode(motor_kiri1, OUTPUT);
  pinMode(motor_kiri2, OUTPUT);
  pinMode(motor_kanan1, OUTPUT);
  pinMode(motor_kanan2, OUTPUT);
  pinMode(13, INPUT);
  pinMode(12, INPUT);

  //Membaca Pin dari encoderPinA == Sinyal yang memberikan notifikasi pada background
  attachInterrupt(digitalPinToInterrupt(encoderPinA_1), doEncoderMotor1, RISING);
}

void kalibrasi(){
        sensor[0] = analogRead(sensor1);
        sensor[1] = analogRead(sensor2);
        sensor[2] = analogRead(sensor3);
        sensor[3] = analogRead(sensor4);
        sensor[4] = analogRead(sensor5);
        sensor[5] = analogRead(sensor6);
```

```
  for (i = 5; i >= 0; i--){
    if(sensor[i] > sensorMin[i]){
        sensorMin[i] = sensor[i];
    }
     if(sensor[i] < sensorMax[i]){
        sensorMax[i] = sensor[i];
    }
    sensorMid[i] = (sensorMin[i] + sensorMax[i]/2);
  }
  Serial.println("-----------------------------");
  Serial.println("| Auto Calibration");
   Serial.println("| SOKHABAT");
  Serial.println("| Kelas D3TK43-03");
  for(int x=0 ; x<=5 ; x++){
    Serial.print("| Sensor ");
    Serial.print(x);
    Serial.print(": ");
    Serial.print(sensor[x]);
    Serial.println(" ");
  }
}

void read_sensor_values(){


  /* Membaca EEFROM kalibrasi
  for(int y=0; y<=5 ;y++){
   baca_sensor[y] = sensor[y];
  }
  */

    baca_sensor[0] = analogRead(sensor1);
    baca_sensor[1] = analogRead(sensor2);
    baca_sensor[2] = analogRead(sensor3);
    baca_sensor[3] = analogRead(sensor4);
    baca_sensor[4] = analogRead(sensor5);
    baca_sensor[5] = analogRead(sensor6);

  // 1 = Gelap 0 = Terang
```

```
  // Case 1 0 0 0 0 0
if (baca_sensor[0] < 34 && baca_sensor[1] > 34 &&
    baca_sensor[2] > 34 && baca_sensor[3] > 34 &&
    baca_sensor[4] > 34 && baca_sensor[5] > 34)
    {
    error = -4;
  Serial.print("\n");
    }

  // Case 1 1 0 0 0 0
else if (baca_sensor[0] < 34 && baca_sensor[1] < 34 &&
    baca_sensor[2] > 34 && baca_sensor[3] > 34 &&
    baca_sensor[4] > 34 && baca_sensor[5] > 34){

    error = -3;
  Serial.print("\n");
    }

  // Case 0 1 0 0 0 0
else if (baca_sensor[0] > 34 && baca_sensor[1] < 34 &&
    baca_sensor[2] > 34 && baca_sensor[3] > 34 &&
    baca_sensor[4] > 34 && baca_sensor[5] > 34){

    error = -2;
  Serial.print("\n");
    }

  // Case 0 1 1 0 0 0
else if (baca_sensor[0] > 34 && baca_sensor[1] < 34 &&
    baca_sensor[2] < 34 && baca_sensor[3] > 34 &&
    baca_sensor[4] > 34 && baca_sensor[5] > 34){

   error = -1;
  Serial.print("\n");
    }

  // Case 0 0 0 1 0 0
else if (baca_sensor[0] > 34 && baca_sensor[1] > 34 &&
    baca_sensor[2] < 34 && baca_sensor[3] > 34 &&
```

```
    baca_sensor[4] > 34 && baca_sensor[5] > 34){

    error = 0;
  Serial.print("\n");
   }

  // Case 0 0 1 1 0 0
else if (baca_sensor[0] > 34 && baca_sensor[1] > 34 &&
    baca_sensor[2] < 34 && baca_sensor[3] < 34 &&
    baca_sensor[4] > 34 && baca_sensor[5] > 34){

    error = 0;
  Serial.print("\n");
   }


  // Case 0 0 0 1 1 0
else if (baca_sensor[0] > 34 && baca_sensor[1] > 34 &&
    baca_sensor[2] > 34 && baca_sensor[3] < 34 &&
    baca_sensor[4] < 34 && baca_sensor[5] > 34){

    error = 1;
  Serial.print("\n");
   }

  // Case 0 0 0 0 1 0
else if (baca_sensor[0] > 34 && baca_sensor[1] > 34 &&
    baca_sensor[2] > 34 && baca_sensor[3] > 34 &&
    baca_sensor[4] < 34 && baca_sensor[5] > 34){

    error = 2;
  Serial.print("\n");
   }

  // Case 0 0 0 0 1 1
else if (baca_sensor[0] > 34 && baca_sensor[1] > 34 &&
    baca_sensor[2] > 34 && baca_sensor[3] > 34 &&
    baca_sensor[4] < 34 && baca_sensor[5] < 34){
```

```
    error = 3;
  Serial.print("\n");
    }

  // Case 0 0 0 0 0 1
else if (baca_sensor[0] > 34 && baca_sensor[1] > 34 &&
    baca_sensor[2] > 34 && baca_sensor[3] > 34 &&
    baca_sensor[4] > 34 && baca_sensor[5] < 34){

    error = 4;
  Serial.print("\n");
    }

}

void hitungPID(){
  P = error;
  I = error + lastError;
  D = error - lastError;

  PID_value = (Kp * P) + (Ki * I) + (Kd * D);

  lastError = error;
}

// Mengatur Kecepatan Motor
void motor_control()
{
  int kecepatanMotorKiri = kecepatanSetPoint - PID_value;
  int kecepatanMotorKanan = kecepatanSetPoint + PID_value;

  // Kecepetan Motor agar tidak melebihhi batas pwm
  kecepatanMotorKiri = constrain(kecepatanMotorKiri, 0, 255);
  kecepatanMotorKanan = constrain(kecepatanMotorKanan, 0, 255);


    Serial.println(kecepatanMotorKanan);
  digitalWrite(ENA, HIGH);
  analogWrite(motor_kiri1, kecepatanMotorKiri);
```

```
  analogWrite(motor_kiri2, 0);

  digitalWrite(ENB, HIGH);
  analogWrite(motor_kanan1, kecepatanMotorKanan);
  analogWrite(motor_kanan2, 0);

}

void loop(){
  if(digitalRead(12) == 0){
  goto bawah;
  }

  if(digitalRead(13) == 0){
    while(true){
      atas:
  kalibrasi();
  if(digitalRead(12) == 0){
    while(true){
      bawah:
  read_sensor_values();
  hitungPID();
   motor_control();
      if(digitalRead(13) == 0){
      goto atas;
      }
    }
  }
    }
  }


}
```

6. Jurnal Praktikum

a. Jurnal pada Buku Praktikum harus memuat konten sebagai berikut :

Judul Praktikum :

- Maksud dan Tujuan Praktikum :
- Peralatan dan Bahan Praktikum :
- Dasar Teori

- Foto Peralatan dan Bahan Praktikum :
- Hasil Praktikum (Tulis tangan kode program yang telah diberi komentar/penjelasan beserta foto hasil percobaan yang telah diberi nama dan

NIM anggota kelompok)

- Kesimpulan Praktikum