

## Implementasi Teknik I/O Interfacing berbasis Arduino

Zuly Budiarmo, Edy Winarno dan Hersatoto Listiyono

Fakultas Teknologi Informasi, Universitas Stikubank Semarang

email: zbudiarmo@gmail.com; edywinarno3@gmail.com; herlistiyono@gmail.com

### Abstrak

Serangkaian elektronika modern telah mengembangkan sistemnya dengan menggunakan operasi sistem digital, yang menggunakan sinyal dan besaran digital dalam memberikan penggerak / *trigger* maupun menghasilkan sinyal / *signal generator* yang dapat digunakan untuk kelanjutan sistem dalam mata rantai sistem digital dan komputasi. Besaran nilai analog seperti temperatur, tekanan, kecepatan, dan lain sebagainya, yang melibatkan besaran fisis, mampu dikemas dalam sebuah sistem digital, yang masih yang pada umumnya masih bersifat non komputasi. Pengadaan sistem terpadu yang bisa dilakukan untuk menjembatani permasalahan pengolahan data yang terekam dari perangkat elektronik masih memiliki tingkat kesulitan yang tinggi dan pembiayaan yang mahal.

Mikrokontroler yang mewakili suatu figur sistem kendali modern telah dikemas hanya dengan mengimplementasikan sebuah *chip* / rangkaian terpadu, yang kemudian berkembang hingga penggunaan memori dan processor beserta elemen pendukungnya seperti register, *AT command*, *I/O system*, dan lain sebagainya. Bahkan hingga pada saat ini, mikrokontroler telah dikemas dalam sebuah modul aktif kit.

Hasil penelitian ini adalah pengamatan dan analisa terhadap beberapa teknik antarmuka meliputi perancangan dan pemrograman yang terlibat dalam suatu model sistem kendali. Teknik antarmuka pemrograman perangkat lunak dengan bahasa C jauh lebih mudah dimengerti oleh perancang sekalipun untuk tingkat pemula, dengan diketahuinya struktur dasar yaitu *void setup()*, *void loop()* dan beberapa tipe variabel serta konsep percabangan atau kondisional, Sementara itu teknik antarmuka perancangan perangkat keras diperlukan beberapa pengetahuan dasar elektronika dalam hal karakteristik tegangan dan arus dari elemen masukan, yang harus disesuaikan dalam kerangka I/O dari Arduino yang menuntut pengkondisian tegangan dan arus berkisar pada 5-9 volt, dan arus sebesar 500mA.

**Kata kunci :** arduino, mikrokontroler, sistem kendali

### PENDAHULUAN

Sistem kendali telah banyak diimplementasikan dalam berbagai bidang terutama untuk segenap sektor industri yang memerlukan sebuah kontrol terpadu dengan harapan untuk mempertahankan atau bahkan meningkatkan efisiensi kerja apabila dibandingkan dengan menggunakan tenaga kerja manusia. Sistem kendali hingga saat ini mampu menempatkan posisinya sebagai fitur canggih yang mutlak untuk dilakukan. Berbagai model dan bentuk telah dihasilkan mulai dari minikontroler hingga mikrokontroler dan mikroprosesor.

ATMEL telah mengeluarkan salah satu komponen sistem kendali dengan segenap

variasinya seperti ARDUINO. ARDUINO menjadi basis penelitian pada kesempatan kali ini dengan fokus pada pengamatan dan analisa konsep antarmuka pada perangkat modul ARDUINO UNO. ARDUINO UNO memiliki kekuatan prosesor ATMEL328u/pu dengan kemasan yang miniatur dengan kemampuan interfacing dan pemrograman yang mudah bagi pemula hingga bagi para ahli.

Sebuah sistem penghubung / saklar otomatis merupakan implementasi terhadap perkembangan sistem analog yang semula membutuhkan operasi dengan seorang operator, tingkat akurasi dan daya tahan yang terbatas seiring dengan keterbatasan manusia / operator, yang hingga saat ini terus variatif dalam

perancangan sistemnya. Berawal dari sistem kontaktor/saklar manual hingga konsep *switching* pada elemen elektronika seperti transistor, dioda, SCR dan lain sebagainya.

Serangkaian elektronika modern telah mengembangkan sistemnya dengan menggunakan operasi sistem digital, yang menggunakan sinyal dan besaran digital dalam memberikan penggerak / *trigger* maupun menghasilkan sinyal / *signal generator* yang dapat digunakan untuk kelanjutan sistem dalam mata rantai sistem digital dan komputasi.

Mikrokontroler yang mewakili suatu figur sistem kendali modern telah dikemas hanya dengan mengimplementasikan sebuah *chip* / rangkaian terpadu, yang kemudian berkembang hingga penggunaan memori dan processor beserta elemen pendukungnya seperti register, *AT command*, *I/O system*, dan lain sebagainya. Bahkan hingga pada saat ini, mikrokontroler telah dikemas dalam sebuah modul aktif kit.

Perancang sebuah sistem kendali menjadi lebih tertantang dengan hadirnya kemampuan untuk *programming*, sehingga perancang dapat dengan lebih leluasa membuat bentuk maupun model sistem kendali. Kemampuan modul yang bervariasi dengan berbagai aksesorisnya yang siap pakainya telah meningkatkan keinginan lebih lanjut dalam keberagaman modul berbasis ATMEL ini.

Penelitian ini diharapkan akan mampu mengeksplorasi lanjut terhadap implementasi perancangan sistem kendali terpadu dan mengujinya dengan harapan dapat membantu dalam mengangkat pemanfaatan sebuah sistem kendali terpadu berbasis mikrokontroler.

## PERUMUSAN MASALAH

Perkembangan mikrokontroler menuntut perkembangan analisa dalam penggunaan dan mengikuti perkembangan untuk beberapa tipe tertentu dalam sistem kendali, serta diperlukannya suatu kajian analisa tersendiri untuk mencapai suatu simpulan sederhana yang sebenarnya menjadikan topik dalam perkembangan teknologi adalah penyederhanaan sistem dan algoritma sistem kendali modern.

## BATASAN MASALAH

Adapun batasan masalah dalam kesempatan penelitian ini adalah :

- Teknik rancang bangun berbasis sistem kendali dengan inti ATMEL ATMEGA238P-PU yang menjadi tren saat ini.
- Teknik pengamatan dan pengujian *I/O library* dengan memberikan besaran secara simultan dari beberapa sensor
- Teknik *pemrograman I/O* sedemikian rupa dengan berbasis bahasa C

## TUJUAN DAN MAFAAT PENELITIAN

Adapun tujuan dalam penelitian pada kesempatan kali ini adalah :

- Mengimplementasikan sebuah sistem kendali terpadu sebagai simulasi kontrol
- Mengetahui prinsip dasar pemrograman berbasis mikrokontroler Arduino
- Mengetahui teknik *interfacing* komunikasi port dengan PC

Manfaat yang ingin dicapai dalam penelitian ini adalah :

- Memberikan wawasan pemrograman modul sistem kendali
- Memberikan wawasan perancangan modul sistem kendali
- Memberikan wawasan teknik *interfacing* dengan mikrokontroler tipe Arduino dalam berbagai variasi modulnya

## METODE PENELITIAN

Metode yang akan digunakan dalam penelitian ini terdiri dari langkah-langkah sebagai berikut ;

### 1. Studi Literatur

Untuk memperoleh dasar teori berkaitan dengan pemrograman mikrokontroler, pemrograman C baik berasal dari jurnal, buku maupun informasi baku lainnya yang bersumber dari situs-situs di internet, berkaitan dengan perancangan dan

pemrograman I/O mikrokontroler berbasis Arduino.

## 2. Pemrograman Aplikasi

Pemrograman aplikasi ini dimaksudkan untuk menguji sebuah teknik serta mengeksplorasi teknik *interfacing* dan mengimplementasikannya teknik/metode tersebut dalam sebuah aplikasi berbasis Arduino UNO R3, dengan menggunakan software bantu berbasis PC.

## HASIL DAN PEMBAHASAN

Beberapa instruksi program yang diamati dari contoh di atas dapatlah dikupas informasi yang berkaitan dengan hal itu dalam suatu keterangan ringkas sebagai berikut, yang diantaranya adalah berkaitan dengan struktur pemrograman yang berbasis bahasa C. Pembahasan pemrograman ini juga melibatkan beberapa telaah teoritis dari berbagai sumber sebagai rujukan sehingga dapat memperkuat analisa dalam memperoleh informasi yang terbaik.

### Instruksi diawali dengan `setup()` dan `loop()`

Semua program dalam pemrograman arduino ini memiliki komponen utama yaitu `setup()` dan `loop()` dimana instruksi ini dapat diimplementasikan pada contoh berikut ini :

```
void setup()
{
}
void loop()
{
}
```

`Setup()` biasanya digunakan untuk menandai atau mengenalkan atau menginisiasi terminal I/O yang dipakai untuk indikator seperti LED, sensor, motor dan mungkin penggunaan terminal serial (serial port). Instruksi ini seklaigus memberitahukan kepada Arduino bahwa terminal-terminal tersebut akan digunakan selama program aplikasi dijalankan atau dengan kata lain bahwa terminal tersebut akan disiapkan oleh Arduino dalam menjalankan program.

Sementara itu instruksi `loop()` merupakan kode instruksi yang akan mengendalikan

terminal I/O berkaitan dengan instruksi apa yang akan dilakukan selanjutnya, berikutnya dan seterusnya.

Pemrograman dalam Arduino juga memiliki sub-instruksi atau lebih dikenal dengan subrutin, dimana akan lebih berguna saat program memerlukan beberapa fungsi yang akan dipanggil saat instruksi `loop()` dijalankan. Sementara itu dalam penggunaan subrutin ini harus ditentukan terlebih dahulu awal mulai program, yang biasa disebut dengan prototipe sebuah fungsi (*function prototype*).

```
void delayLED();
void setup()
{
}
void loop()
{
}
```

```
void delayLED()
{
}
```

Dimana (1) merupakan contoh sederhana dalam prototipe fungsi dan (2) adalah contoh subrutin.

### Tipe variabel

Beberapa variabel terkadang diperlukan perancangan program untuk memberikan konstanta data berdasarkan tipe variabel yang terlibat dalam pengamatan antarmuka dengan menggunakan perangkat Arduino ini. Variabel digunakan pada saat merancang program yang akan dijalankan dan terdiri dari beberapa bagian / blok, dimana variabel akan digunakan untuk melalukan data-data yang mungkin digunakan selama program dijalankan. Adapun beberapa tipe program diantaranya adalah sebagai berikut:

Tabel 1. Kategori Tipe Variabel

Type Name	Type Value	Type Range
char	'a'	-128 to 127
byte	1011	0 to 255
int	-1	-32,768 to 32,767
unsigned int	5	0 to 65,535
long	512	-2,147,483,648 to 2,147,483,647
unsigned long	3,000,000	0 to 4,294,967, 295
float	2.513	-3.4028235E+38 to 3.4028235E+38
double	2.513	-3.4028235E+38 to 3.4028235E+38

Setelah mengetahui beberapa kelompok tipe variabel, yang perlu diperhatikan adalah bagaimana cara mendeklarasikan variabel-variabel yang kita perlukan dalam sebuah program. Untuk mendeklarasikan variabel tersebut maka diperlukan dasar pengetahuan tentang lingkup variabel yang akan digunakan atau dalam istilahnya adalah variabel global dan lingkup khusus yang akan spesifik atau disebut sebagai variabel lokal.

Variabel lokal hanya akan digunakan pada lingkup tertentu saja, misalnya pada saat program menggunakan instruksi *for* loop, maka variabel tersebut hanya akan berjalan dalam lingkup loop tersebut saja. Sementara variabel global akan digunakan untuk mendefinisikan sebuah variabel saat program pertama kali dijalankan. Sebuah contoh dalam penggunaan variabel dan deklarasinya ada pada program berikut ini.

```

int x;

Voiv setup()
{
}

Void loop()
{
    x = 1 + 2;
    for(int i; i <= 100; i++)
    {
    }
}
    
```

x merupakan variabel global sedangkan i adalah variabel lokal.

Deklarasi dapat pula dibentuk dalam format *array*, dimana variabel akan memuat lebih dari satu konten atau isinya.

```

const int NumOfPins = 3;
int pins[NumOfPins] = {13,9,8};
    
```

**Format kondisional**

Statemen kondisional ini merupakan salah satu upaya pengendalian aliran program pada sebuah aplikasi. Hal ini terlingkupi dalam kondisi statemen *if*, *if-else-if* dan *switch*. Statemen *if* sangat diperlukan jika program memerlukan kondisi prasyarat tertentu, dan statemen ini dapat diberikan pula logika Boolean ataupun membatasi periode pengujian tertentu. Penggunaan tambahan statemen *elseif* pada akhir statemen *if* memungkinkan untuk memberikan keputusan kondisi prasyarat yang terakhir. Sementara itu statemen *switch* diperlukan seandainya program memerlukan banyak kondisi khusus dan lebih ringkas.

```

int i;
if (i < 10)
{
    i++;
} (A)
    
```

```

int i;
if (i < 10)
{
    i++;
}
else if (i > 10)
{
    i--;
} (B)
    
```

```

switch (portValue){
case 500:
    digitalWrite(motorPin,1);

case 501:
    digitalWrite(ledPin,1);
    break;
default:
    digitalWrite(motorPin,0);
    digitalWrite(ledPin,0); (C)
}
    
```

**Penggunaan Looping**

Prancangan program terkadang memerlukan sebuah even yang akan dilakukan secara terus menerus atau berulang sehingga diperlukan konsep looping. *Looping* atau diartikan statemen perputaran program yang berulang, dimana biasanya menggunakan statemen *for*, *while* dan *do ... while*. Perulangan ini akan menyingkat proses, dimana saat program memerlukan sesuatu yang harus dijalankan seandainya sebuah kondisi terpenuhi (*true*).

```
for (int i = 0; i <= 10; i++)
{
} (A)
```

```
int pins[] = {13,9,8};

void setup()
{
  for (int i = 0; i <= 2; i++)
  {
    pinMode(pin[i], OUTPUT);
  }
}
void loop()
{
} (B)
```

```
int potPin = A1;
int motorPin = 9;
int potVal;

void setup()
{
  pinMode(motorPin, OUTPUT);
  pinMode(potPin, INPUT);
}
void loop()
{
  potVal = analogRead(potPin);
  while(potVal <= 100)
  {
    digitalWrite(motorPin,1);
  }
} (C)
```

```
do
```

```
{
  i++;
} while (i <= 100); (D)
```

**Komunikasi digital**

Teknik untuk berkomunikasi pada terminal I/O dengan beberapa tipe yang berbeda melalui pin digital perlu diketahui sebagai seorang perancang program, sehingga dapat dilakukan komunikasi diantaranya, dengan menggunakan instruksi komunikasi pada pin-pin digital yaitu *digitalWrite(pin,HIGH/LOW)* dan *digitalRead(pin)*.

```
int button = 12;
int led = 13;
int buttonState;

void setup()
{
  pinMode(button,INPUT);
  pinMode(led,OUTPUT);
}
void loop()
{
  buttonState = digitalRead(button);
  if(buttonState == 1)
  {
    digitalWrite(led,HIGH);
  }
  Else
  {
    digitalWrite(led,LOW);
  }
}
```

**Teknik berkomunikasi dengan komponen analog**

Komponen analog yang banyak digunakan dalam sistem kendali adalah seperti pada penggunaan sensor maupun motor penggerak yang mampu diatur kecepatannya dengan potensiometer melalui sebuah modulasi yaitu PWM atau pulse width modulation yang telah tersedia pada pin Arduino. Arduino mengenal beberapa fungsi yang digunakan untuk berkomunikasi dengan komponen analog adalah *analogRead(value)* dan *analogWrite(pin,value)*. Nilai analog dari komponen potensiometer akan diskalakan pada rentang 0 - 255.

```
analogWrite(LED,ledValue/4); // 1024/4 = 255
```

Beberapa instruksi serial adalah :

- Serial.begin(baud)

Instruksi ini dituliskan pada struktur setup(), dan memberikan baudrate yang sesuai dengan perangkat untuk berkomunikasi.

```
void setup()
{
    Serial.begin(9600);
}
```

- Serial.println("teks")

Instruksi ini digunakan untuk menuliskan sebuah nilai untuk terminal serial.

```
void loop()
{
    Serial.println("Hello, World");
}

void loop()
{
    Serial.println(potVal);
}
```

- Serial.read()

Instruksi ini akan membaca nilai dari terminal serial. Misalnya dengan menampilkan nilai dari komputer yang akan ditampilkan pada LCD pada arduino.

```
void loop()
{
    char var = Serial.read();
}
```

- Serial.write(binary data)

Instruksi ini akan digunakan untuk menuliskan data biner ke terminal serial.

```
void ();
{
    while(Serial.available() > 0)
    {
        char var = Serial.read();
        Serial.write(var);
    }
}
```

```
}
```

- Serial.available()

Instruksi ini digunakan sebagai fungsi untuk mengecek data jika terdapat data yang masuk melalui serial port.

```
void loop()
{
    while(Serial.available() > 0)
    {
        // Put code here
    }
}
```

- Serial.end()

Instruksi untuk mengakhiri komunikasi serial.

Contoh lengkap komunikasi serial adalah :

```
int incomingByte;
const int ledPin = 13;

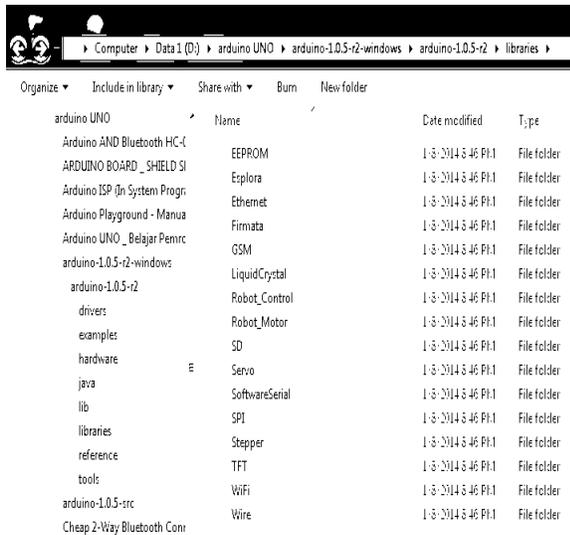
void setup() {
    Serial.begin(9600);
    pinMode(ledPin,OUTPUT);
}

void loop()
{
    while(Serial.available() > 0)
    {
        incomingByte = Serial.read()
        Serial.println(incomingByte, BYTE);
        digitalWrite(ledPin, incomingByte);
    }
}
```

### Instalasi libraries

Ada saatnya saat perancangan program terkadang memerlukan library khusus (Library biasanya sudah tersedia "default") yang TIDAK terdapat pada library Arduino, sehingga perlu menambahkannya, agar program bisa dijalankan.

1. Menentukan lokasi file aplikasi Arduino UNO.
2. Menentukan/pilih folder "libraries"



3. Mengkopikan file library-nya ke folder "libraries"
4. Menyalakan/Restart arduino sketch
5. Mengetikkan header library-nya, misal : `#include <Ethernet.h>`; atau seperti berikut ini :

```
#include <CapSense.h>
void setup() {}
void loop() {}
```

**KESIMPULAN**

Adapun beberapa kesimpulan yang berhasil diperoleh dalam penelitian ini diantaranya adalah sebagai berikut :

- a. Konsep perancangan perangkat keras sistem kendali dengan berbasis mikrokontroler Arduino UNO memerlukan beberapa teknik antarmuka yang telah tersedia beberapa modul terapan aplikatif untuk mengembangkan modul dasar Arduino UNO nya, dan memerlukan pengetahuan/wawasan berkaitan dengan sumber data masukan yang bergantung pada jenis, tipe dan model dari *transducer*, serta model komunikasi yang akan diterapkan pada sistem.
- b. Konsep perancangan perangkat lunak sistem kendali dengan menggunakan ISP Arduino *software* memudahkan perancang dengan mengenali beberapa struktur pemrograman dasar seperti bahasa C, yang meliputi instrumen *void setup()*, *void loop()*,

deklarasi, tipe variabel dan pengembangan dengan melibatkan fungsi percabangan kondisional maupun *looping*.

**DAFTAR PUSTAKA**

Ardi Winoto, 2008, "Mikrokontroler AVR ATmega8/32/16/8535 dan Pemrogramannya dengan Bahasa C pada WinAVR", Penerbit Informatika, Bandung

Endra Pitowarno, 2005, "Mikroprosesor & Interfacing", Penerbit Andi, Yogyakarta

Maik Schmidt, 2011, "Arduino - A Quick Start Guide Quick Start Guide", Pragmatic Programmers, LLC, USA

Teguh Arif Gustaman, Pengendali Pintu Gerbang Menggunakan Bluetooth Berbasis Mikrokontroler Atmega 8, <http://eprints.uny.ac.id/10161/1/Jurnal%20TA%20Teguh%20Arif%20Gustaman%20%2809507131029%29.pdf>

Ferry Agusta Putra, Andi Dharmawan, Triyogatama Wahyu Widodo, Implementasi DuinOS pada Purwarupa Sistem Penyortiran Barang Berbasis Arduino Uno, IJEIS, Vol.2, No.2, <http://jurnal.ugm.ac.id/index.php/ijeis/article/view/2437>

Helmi Guntoro, Yoyo Somantri, Erik Haritman, (2013), Rancang Bangun Magnetic Door Lock Menggunakan Keypad dan Solenoid Berbasis Mikrokontroler Arduino Uno, ELECTRANS, VOL. 12, NO.1, [http://jurnal.upi.edu/file/06\\_helmi\\_guntoro\\_hal\\_39-481.pdf](http://jurnal.upi.edu/file/06_helmi_guntoro_hal_39-481.pdf)

Mahrus Sabang, Rhiza S. Sadjad, Merna Baharuddin, Sistem Parkir Cerdas, <http://pasca.unhas.ac.id/jurnal/files/d649a5b3bc64701cafef244008c6a41.pdf>