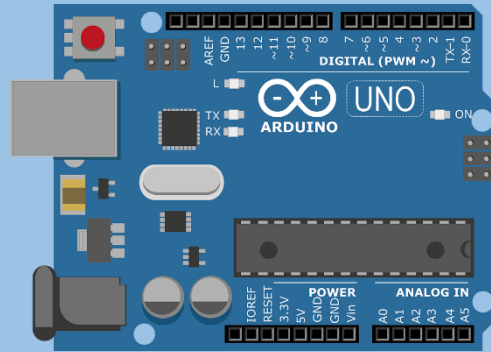
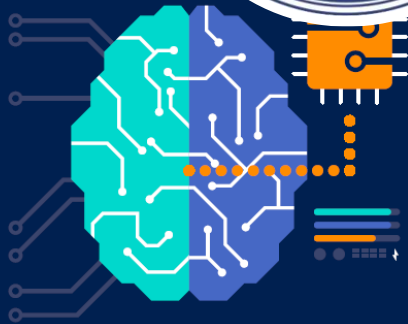


# MODUL PRAKTIKUM



## SISTEM ANTARMUKA KOMPUTER



UNIVERSITAS TEKNOLOGI YOGYAKARTA  
2019

NAMA : \_\_\_\_\_

NPM : \_\_\_\_\_

# **MODUL PRAKTIKUM SISTEM ANTARMUKA KOMPUTER**

Oleh:  
MS HENDRIYAWAN ACHMAD  
ARI SUGIHARTO  
DANANJAYA ARIATEJA

**UNIVERSITAS TEKNOLOGI YOGYAKARTA  
2019**

# Modul Praktikum Sistem Antarmuka Komputer

@ 2019

ISBN

- 978-623-7746-09-6

Oleh :

MS Hendriyawan Achmad, S.T., M.Eng., Ph.D.

Ari Sugiharto, S.Si., M.Eng.

Dananjaya Ariateja, S.T., M.Eng.

-

Edisi ke-1

Cetakan Pertama, 2019

Diterbitkan oleh :

Universitas Teknologi Yogyakarta

Jl. Siliwangi, Jombor, Sleman, Yogyakarta

Email : publikasi@uty.ac.id

Website : uty.ac.id

Hak Cipta @2019 pada penulis,

Dilarang memperbanyak atau memindahkan sebagian atau seluruh isi buku dalam bentuk apapun tanpa ijin dari penulis.

## KATA PENGANTAR

Era modern saat ini sudah mulai meninggalkan teknologi elektronika digital tradisional dan berpindah menuju teknologi digital terprogram, yaitu sistem elektronika yang menerapkan kombinasi antara perangkat keras dan perangkat lunak terprogram yang melibatkan piranti mikroprosesor. Sistem antarmuka komputer berkembang sangat signifikan setelah ditemukannya teknologi pemroses mikro (microprocessor) yang memiliki kemampuan pengolahan aritmatika (arithmetic logic unit – ALU) dan pengendali proses (control unit – CU) yang menggantikan peran sistem antarmuka elektronik konvensional. Sistem benam merupakan komputer super mini yang mengintegrasikan fasilitas-fasilitas komputer skala besar dalam satu keping rangkaian terintegrasi. Dengan adanya fasilitas tersebut dan dengan kesederhanaan implementasinya, mempermudah antarmuka komputer baik untuk proses akuisisi data maupun proses pengendalian. Terlebih saat ini dunia sudah mulai menerapkan isu Industrial 4.0 yaitu konsep penerapan teknologi IoT – Internet of Thing untuk proses industri yang melibatkan sistem antarmuka komputer. Dengan fakta tersebut di atas, maka pembelajaran secara akademik tentang teknik antarmuka komputer modern sangat bermanfaat bagi mahasiswa perguruan tinggi. Upaya pembelajaran sistem antarmuka komputer akan memberikan nilai tambahan pengetahuan bagi mahasiswa untuk mempersiapkan diri bekerja di Industri modern yang sudah menerapkan konsep Industrial 4.0 untuk proses produksi.

Modul praktikum ini disusun oleh tim penulis yaitu MS Hendriyawan Achmad, Ari Sugiharto, dan Dananjaya Ariateja dengan menyajikan pembahasan terdiri ulasan teori dan praktik meliputi: pengantar pemrograman arduino dan delphi, antarmuka komunikasi serial pc – Arduino, akses input – output digital arduino oleh pc, akses input – output analog arduino oleh pc, pemrograman serial i2c: antarmuka rtc ds1307, pemrograman serial uart: antarmuka modul gprs, pemrograman serial spi: antarmuka ethernet, akuisisi data sensor suhu lm35 dan membuat gui menggunakan aplikasi delphi kendali motor dc dengan teknik pwm dan membuat gui menggunakan aplikasi delphi, kendali motor servo dengan teknik pwm dan membuat gui menggunakan aplikasi delphi..

Sebagai penutup, kami akan senantiasa memperbaiki kualitas isi penyampaian materi agar selalu sinkron dengan target capaian pembelajaran dari waktu ke waktu. Saran dan koreksi bisa dikirimkan melalui e-mail penulis utama: [hendriyawanachmad@uty.ac.id](mailto:hendriyawanachmad@uty.ac.id)

Yogyakarta, 2019

Tim Penulis

## DAFTAR ISI

HALAMAN SAMPUL	i
HALAMAN JUDUL	ii
PERNYATAAN HAK CIPTA	iii
KATA PENGANTAR	iv
DAFTAR ISI	v
MODUL 1 Pengantar Pemrograman Arduino Dan Delphi	1
MODUL 2 Antarmuka Komunikasi Serial PC – Arduino	11
MODUL 3 Akses Input – Output Digital Arduino Oleh PC	17
MODUL 4 Akses Input – Output Analog Arduino Oleh PC	22
MODUL 5 Pemrograman Serial I2C: Antarmuka RTC DS1307	31
MODUL 6 Pemrograman Serial UART: Antarmuka Modul GPRS	35
MODUL 7 Pemrograman Serial SPI: Antarmuka Ethernet	39
MODUL 8 Akuisisi Data Sensor Suhu LM35 Menggunakan Aplikasi Delphi	42
MODUL 9 Kendali Motor DC Dengan Teknik PWM Menggunakan Aplikasi Delphi	47
MODUL 10 Kendali Motor Servo Dengan Teknik PWM Menggunakan Aplikasi Delphi	52

# MODUL 1

## PENGANTAR PEMROGRAMAN ARDUINO DAN DELPHI

### 1.1 Kompetensi Dasar

- Mengajarkan kepada mahasiswa tentang cara menggunakan Arduino IDE dan memahami cara mengembangkan perangkat lunak untuk sistem elektronik benam (embedded) berbasis bahasa program C.
- Mengajarkan kepada mahasiswa tentang cara menggunakan Delphi IDE dan memahami cara mengembangkan perangkat lunak untuk antarmuka pengguna berbasis grafis (GUI – *graphical user interface*) berbasis bahasa program Pascal.

### 1.2 Indikator

- Mahasiswa mampu mengoperasikan Arduino IDE untuk membuat proyek sederhana, dan berhasil menjalankan aplikasi pada papan Arduino.
- Mahasiswa mampu membuat aplikasi GUI menggunakan Delphi IDE untuk menampilkan informasi visual pada pengguna

### 1.3 Uraian Materi

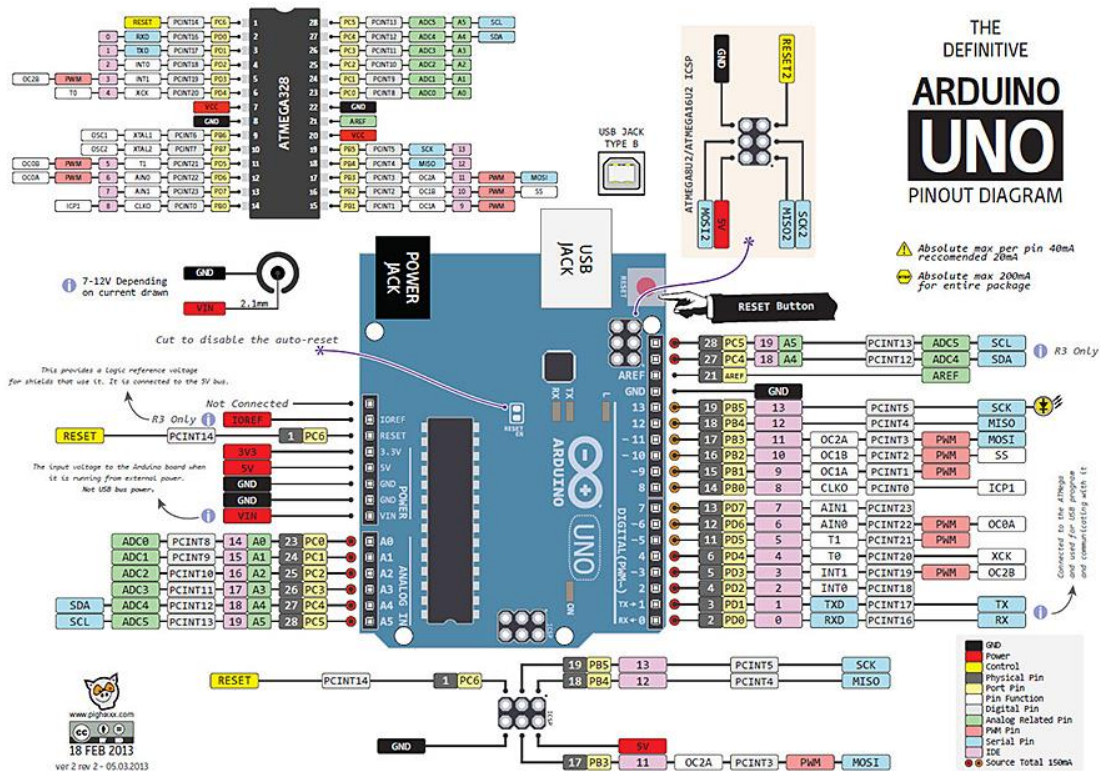
#### 1.3.1 Arduino

Arduino adalah perangkat keras elektronik yang terdapat sistem benam di dalamnya. Terdapat beberapa varian papan elektronik Arduino yang umum digunakan seperti ditunjukkan oleh Gambar 1.1 berikut ini.



**Gbr. 1.1** Variasi papan elektronik Arduino. (Nano – Uno – Mega)

Pada umumnya sistem benam berbasis Arduino menggunakan jenis pengendali mikro ATMEL AVR 8-bit, antara lain: ATmega8, ATmega32, ATmega168, ATmega328, ATmega1280, dan ATmega2560. Selain itu, Arduino juga menggunakan jenis pengendali mikro 32-bit dari beberapa vendor, yaitu: SAM3X8E ARM Cortex-M3 produksi ATMEL, dan Galileo serta Curie produksi INTEL. Perkembangan perangkat keras Arduino sangatlah dinamis mengikuti keinginan pengguna yang menghendaki sebuah piranti elektronik serbaguna, mudah diprogram, harganya murah, dan memiliki fleksibilitas integrasi dengan piranti lain.



Gbr. 1.2 Tata letak dan penamaan pin masukan dan keluaran Arduino UNO

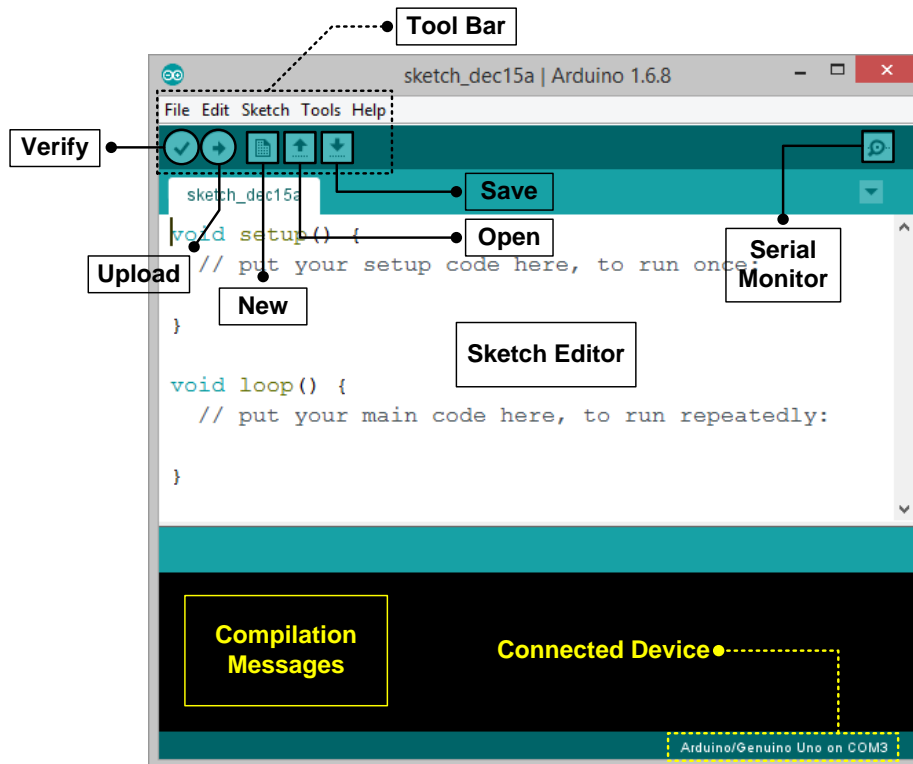
Tata letak dan penamaan pin masukan dan keluaran papan Arduino memiliki standar tersendiri, seperti yang ditunjukkan oleh Gambar 1.3 yang menunjukkan tata letak dan penamaan papan Arduino UNO. Pada dasarnya penamaan pin Arduino berdasarkan karakteristiknya, yaitu: input/output digital, dan input/output analog. Pin input/output A0 – A7 secara khusus digunakan untuk membaca data analog, namun secara umum juga bisa digunakan untuk input/output data digital. Selain itu terdapat pin 0, 1, 2, ... , n yang secara umum digunakan untuk input/output data digital. Maksimal nilai penomoran pin digital sesuai dengan jenis pengendali mikronya. Untuk papan Arduino UNO, pin input/output digital antara 0 – 13. Pada beberapa pin digital terdapat penambahan tanda ~ di awal nomor, hal ini menunjukkan bahwa pin juga bisa digunakan untuk output analog (PWM – pulse width modulation). Adapun fungsi-fungsi khusus lain bisa dilihat secara jelas pada Gambar 1.2.

### 1.3.2 Arduino IDE

Sistem Arduino tidak hanya sebatas perangkat keras elektronik saja, namun juga terdapat perangkat lunak pengembang nya, dikenal dengan Arduino IDE (Integrated Development Environment). Arduino IDE merupakan perangkat lunak pengembang berbasis platform Java yang digunakan untuk melakukan beberapa hal utama, antara lain:

- Menulis/mengubah kode program Arduino (sketch)
- Melakukan kompilasi kode program Arduino menjadi kode mesin (\*.hex)
- Mengisikan kode mesin hasil kompilasi ke dalam pengendali mikro Arduino

Selain melakukan tiga hal utama di atas, Arduino IDE juga memiliki fasilitas pendukung lain, antara lain: pembaharuan library secara daring yang akan memudahkan dalam penyusunan kode program, monitor komunikasi data serial, dan pengaturan koneksi dengan perangkat keras Arduino. Gambar 1.3 menunjukkan tampilan Arduino IDE.



**Gbr. 1.3** Tampilan Arduino IDE dengan keterangan fungsi tiap bagiannya.

Kode-kode program yang ditulis pada editor teks Arduino IDE disebut dengan Sketch, dengan menggunakan platform dasar bahasa pemrograman C yang sudah disesuaikan untuk keperluan pengembangan sistem benam berbasis Arduino. Status dan informasi proses kompilasi dapat dipantau pada bagian bagian bawan Arduino IDE, meliputi kesalahan yang terdeteksi, tahapan kompilasi, dan hasil kompilasi.

```

void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}

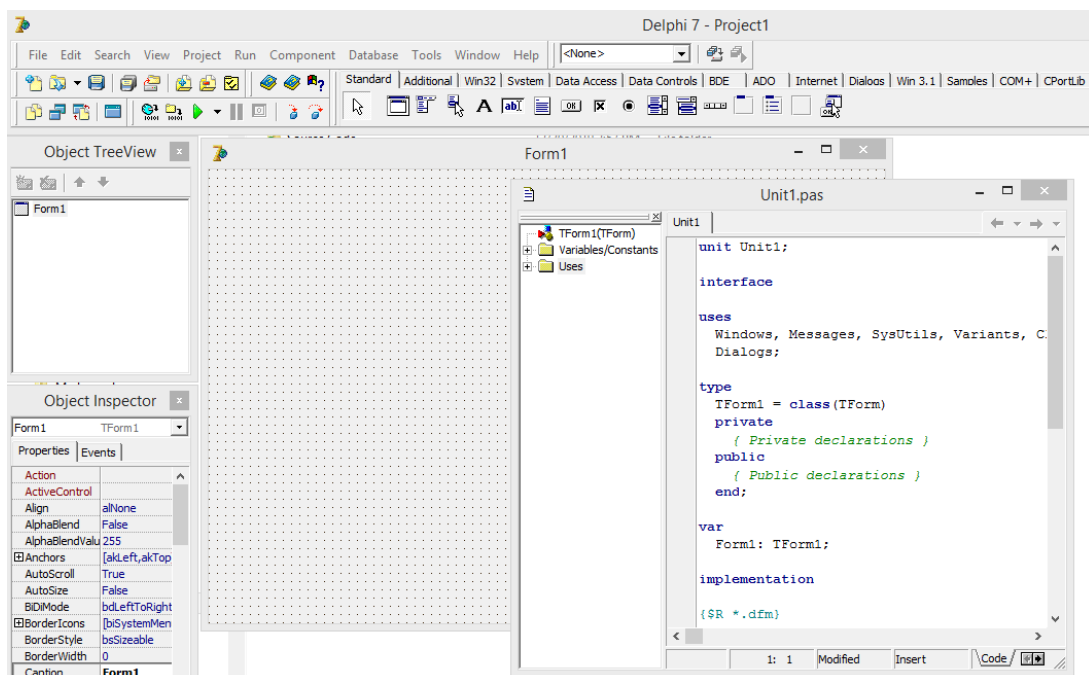
```

Struktur dasar pemrograman Arduino secara umum terbagi menjadi dua bagian, yaitu setup dan loop. setup digunakan untuk mendefinisikan fungsi dan karakteristik fasilitas yang akan digunakan dan hanya dieksekusi sekali saja, sedangkan loop digunakan untuk menyatakan sekuensial kerja yang akan dilakukan secara berulang terus menerus. Di luar setup dan loop dapat digunakan untuk deklarasi variabel dan fungsi, deklarasi library, definisi pin input/output, dll. Di bawah ini contoh struktur dasar penyusunan kode program pada Arduino IDE.



### 1.3.3 Delphi IDE

Delphi adalah suatu bahasa pemrograman (development language) yang digunakan untuk merancang suatu aplikasi program visual (GUI – graphical user interface). Terdapat beberapa kelebihan dari Delphi, yaitu: IDE (Integrated Development Environment) atau lingkungan pengembangan aplikasi sendiri adalah satu dari beberapa keunggulan delphi, di dalamnya terdapat menu – menu yang memudahkan kita untuk membuat suatu proyek program. Kemudian proses kompilasi cepat, pada saat aplikasi yang kita buat dijalankan pada Delphi, maka secara otomatis akan dibaca sebagai sebuah program, tanpa dijalankan terpisah. Selain itu Delphi mudah digunakan, source code delphi yang merupakan turunan dari pascal, sehingga tidak diperlukan suatu penyesuaian lagi. Dan terakhir adalah bersifat multi purphase, artinya bahasa pemrograman Delphi dapat digunakan untuk mengembangkan berbagai keperluan pengembangan aplikasi. Gambar 1.4 menunjukkan tampilan utama Delphi IDE versi 7.



Gbr. 1.4 Tampilan utama Delphi IDE versi 7.

Delphi adalah salah satu pemrograman berbasis obyek (*Object Oriented Programming – OOP*). OOP adalah metode pemrograman dengan membentuk sebuah aplikasi yang mendekati keadaan dunia yang sesungguhnya. Hal itu bisa dilakukan dengan cara mendisain object untuk menyelesaikan masalah.

Tiga unsur OOP:

#### 1. *Encapsulation* atau pemodelan.

Encapsulation adalah konsep penggabungan data dengan operator. Dalam konsep pemodelan data dan operasi menjadi satu kesatuan yang disebut object. Encapsulation juga disebut dengan penyembunyian informasi (information hiding). Contoh: ketika kita menyalakan sebuah TV kita tidak tahu apa yang terjadi dengan proses dan percakapan antar alat yang berhubungan dengan TV tersebut sehingga mampu menampilkan sebuah gambar, atau ketika melakukan klik pada sebuah object dalam suatu aplikasi program kita tidak tahu apa yang dilakukan program sehingga ditampilkan hasil yang sesuai. Dari dua contoh kasus tersebut terdapat kesamaan proses mengenai information hiding yang tidak diketahui oleh user sampai hasil ditampilkan.

## 2. *Inheritance* atau penurunan.

Inheritance adalah sebuah object yang dapat diturunkan menjadi object yang baru dengan tidak menghilangkan sifat asli dari object tersebut. Contoh: TV merupakan salah satu media elektronik yang digunakan untuk menampilkan gambar dengan tujuan untuk memberikan informasi kepada konsumen. Secara umum TV mempunyai cara kerja yang sama dengan media elektronik yang lain dalam proses penyampaian informasi, tetapi mempunyai sifat yang unik yang dapat membedakan dengan media elektronik yang ada, atau dalam aplikasi program kita sering menggunakan command button untuk beberapa perintah yang berbeda.

## 3. *Polymorphism* atau Polimorfisme.

Polymorphism merupakan penggunaan berbagai macam object yang berbeda tetapi secara fungsi bergantung pada satu object sebagai induk, dengan cara pelaksanaan yang berbeda – beda. Contoh: TV dan radio adalah media elektronik yang mempunyai sistem yang sama tentang bagaimana menyebarkan suatu informasi, tetapi cara kerja dari masing – masing sistem pasti berbeda. Atau object Simpan dan Update adalah icon yang berasal dari induk yang sama yaitu command button tetapi cara kerja tersebut berbeda – beda.

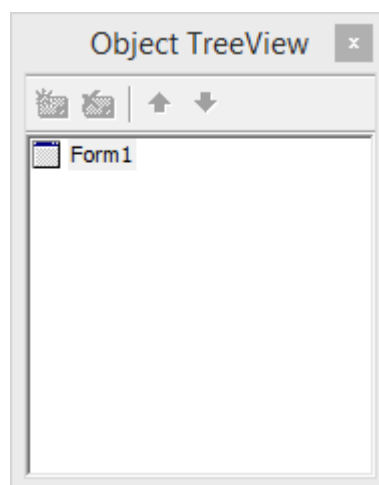
Secara default ketika kita merancang suatu aplikasi program, tanpa kita sadari bahwa kita telah menerapkan OOP, walaupun secara teori kita kurang memahami OOP dalam arti yang sebenarnya. Contoh dalam bahasa program adalah sebagai berikut, ketika merancang suatu label di form maka secara otomatis Delphi akan menuliskan label tersebut dalam jendela code editor tentang turunan dari label tersebut.

```
Type  
  Tform = class(tform)  
    Label1 = TLabel  
  End;
```

Adapun bagian – bagian jendela Delphi IDE adalah sebagai berikut:

### 1. Object Tree View

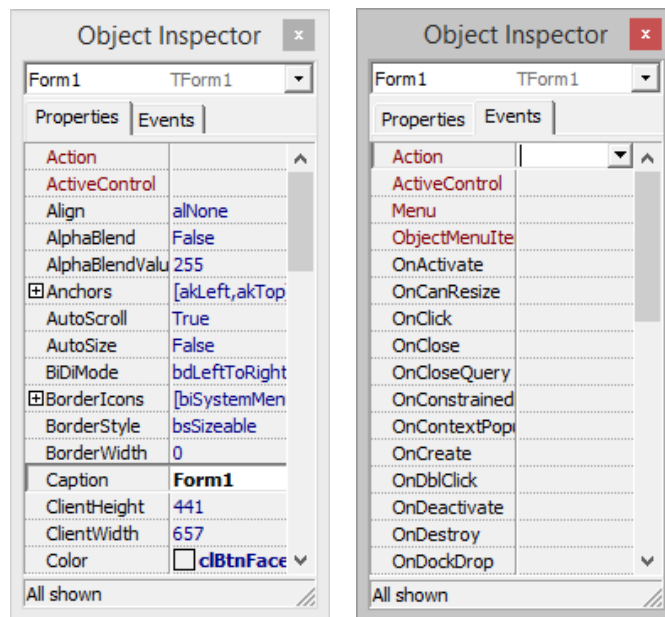
Merupakan sebuah diagram pohon yang menggambarkan hubungan logis menghubungkan semua komponen yang terdapat dalam suatu proyek program. Komponen tersebut meliputi form, modul atau frame. Fungsinya digunakan untuk menampilkan seluruh daftar komponen program dalam sebuah aplikasi program sesuai dengan penempatannya.



Gbr. 1.5 Tampilan Object Tree View

## 2. Object Inspector

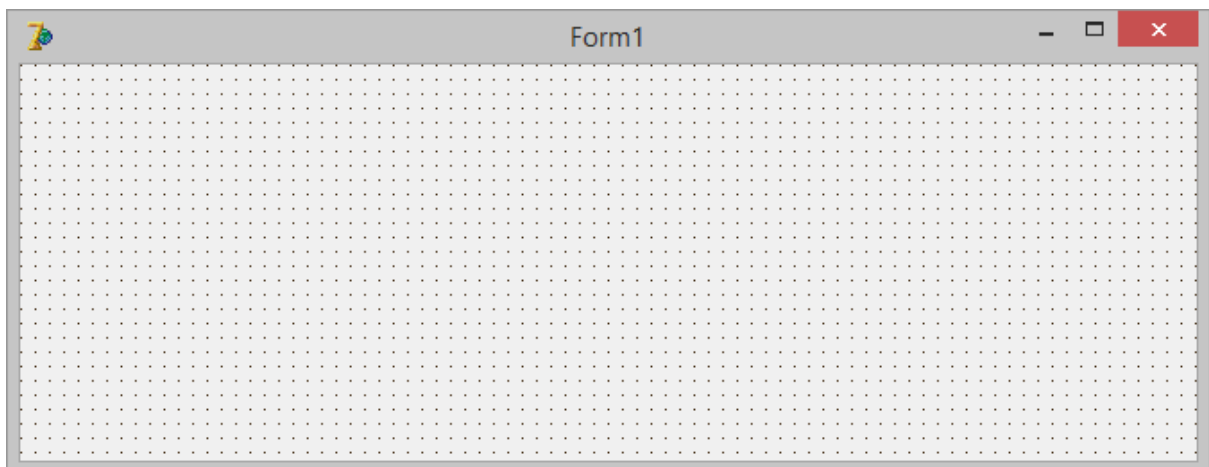
Merupakan jendela yang digunakan untuk mengatur tampilan komponen pada form, misalnya bagaimana mengubah tulisan button pada command button menjadi Simpan, atau menghapus tulisan pada label dan mengganti nama menjadi Nama Mahasiswa atau memberikan perintah tertentu pada sebuah komponen sehingga ada interaksi ketika program dijalankan. Secara Umum Object Inspector terbagi menjadi 2, yaitu Properties dan Events. Properties digunakan untuk mengatur tampilan pada sebuah komponen baik itu meliputi penggantian nama, warna, jenis huruf, border dan lain-lain. Sedangkan Events digunakan untuk memberikan fungsi yang lebih detail dari fungsi sebenarnya. Misalnya ketika tombol Simpan di klik maka program akan menjalankan perintah penyimpanan data. Dari kalimat tersebut ada event klik untuk mengeksekusi sebuah tombol simpan. Perintah event klik tersebut dapat diberikan melalui jendela events.



Gbr. 1.6 Tampilan Object Inspector

## 3. Form Designer

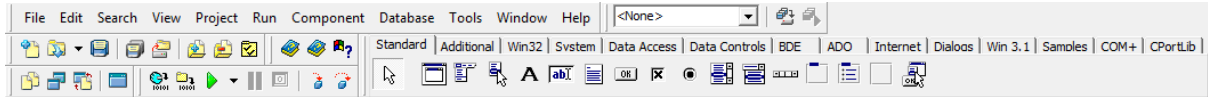
Merupakan tempat yang digunakan untuk merancang semua aplikasi program yang diambil dari komponen palette.



Gbr. 1.7 Tampilan Form Designer

#### 4. Component Pallette

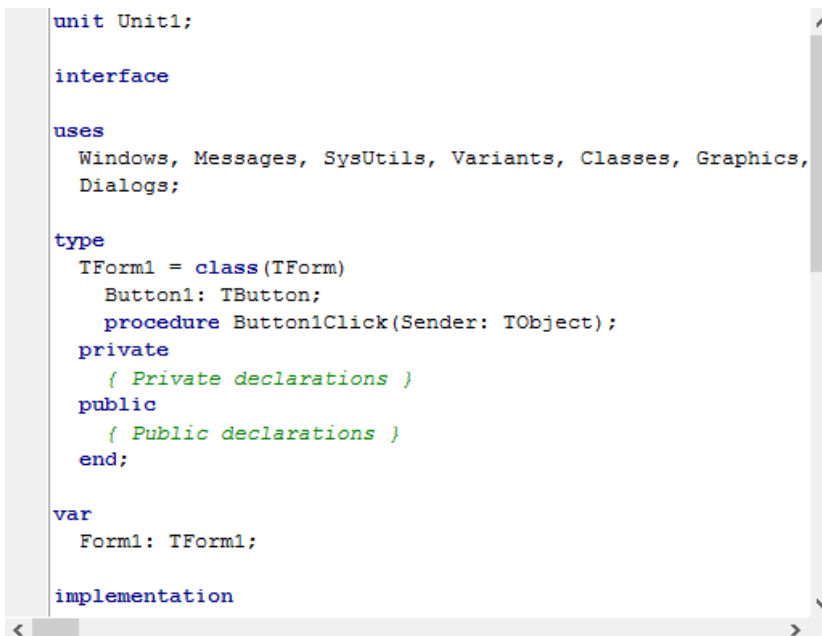
Merupakan kumpulan icon yang digunakan untuk merancang suatu aplikasi pada Form Designer untuk membentuk sebuah aplikasi graphical user interface (GUI). Dalam komponen pallette, semua icon dikelompokan dalam berbagai komponen sesuai dengan fungsi dan kegunaannya.



Gbr. 1.8 Tampilan Component Pallette

#### 5. Code Editor

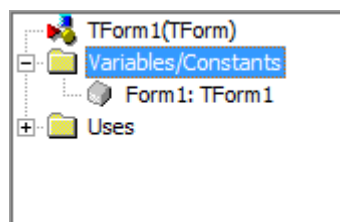
Bagian dari delphi yang digunakan untuk menuliskan kode program. Pada bagian code editor terdapat 3 bagian utama yaitu bagian paling kiri yang berisi berupa angka menunjukkan baris dan kolom. Keterangan modified menunjukkan bahwa telah terjadi modifikasi terhadap baris program. Dan yang paling kanan menunjukkan status keyboard tentang tombol insert atau overwrite.



Gbr. 1.9 Tampilan Code Editor

#### 6. Code Explorer

Jendela yang digunakan untuk menampilkan seluruh variabel, type, dan routine yang didefinisikan pada sebuah unit.



Gbr. 1.10 Tampilan Code Explorer

## 1.4 Peralatan Praktikum

- A. 1 PC
- B. 1 Trainer Sistem Antarmuka Komputer
- C. Software Arduino IDE
- D. Software Delphi 7

## 1.5 Langkah Praktikum

### A. Pemrograman Arduino IDE

Lakukan langkah – langkah seperti berikut ini:

1. Menghubungkan papan Arduino dengan PC/Laptop melalui kabel USB.

Pada tahap ini papan Arduino tidak perlu diberikan daya listrik melalui konektor power supply karena saat terhubung dengan PC/Laptop melalui kabel USB maka papan Arduino secara langsung mendapatkan daya listrik. Kecuali jika papan Arduino terhubung dengan beban listrik yang besar (misal: motor servo, dll) maka harus diberikan daya listrik eksternal.

2. Membuka aplikasi Arduino IDE.

Setelah aplikasi Arduino IDE berhasil dijalankan, selanjutnya membuka contoh kode program (Blink.ino) yang sudah disediakan oleh Arduino IDE. Langkahnya:

Sorot: **File** → **Examples** → **01.Basics** → **Blink**

3. Mengatur koneksi Arduino IDE dengan papan Arduino.

Hal pertama yang harus dilakukan adalah memilih jenis papan Arduino yang terhubung dengan PC/Laptop. Langkahnya:

Sorot: **Tools** → **Board:** “ ? “ dan pilih jenis papan Arduino yang digunakan.

Tahap selanjutnya adalah memilih port USB PC/Laptop yang sudah terhubung dengan papan Arduino. Langkahnya:

Sorot: **Tools** → **Port:** “ ? “ dan pilih nama port pada PC/Laptop.

4. Melakukan kompilasi kode program dan upload kode mesin menuju papan Arduino.

Setelah contoh kode program tertampil pada Sketch Editor, selanjutnya lakukan kompilasi kode program dengan menekan ikon kompilasi seperti yang terlihat pada Gambar 1.3. Jika tidak ada notifikasi kesalahan selanjutnya dilakukan proses upload kode mesin hasil kompilasi menuju papan Arduino yang sudah dihubungkan ke PC/Laptop dengan menekan ikon upload seperti yang terlihat pada Gambar 1.3.

5. Mengamati apa yang terjadi pada papan Arduino.

### B. Pemrograman Delphi IDE

Lakukan langkah – langkah sebagai berikut:

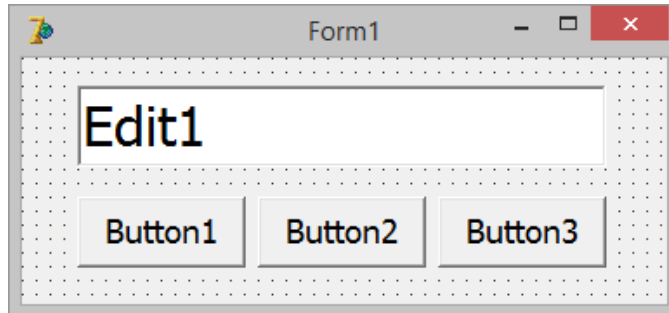
1. Membuka aplikasi Delphi IDE

Setelah aplikasi Delphi IDE berhasil dijalankan, selanjutnya silahkan anda buat aplikasi GUI. Adapun langkahnya:

Sorot: **File** → **New** → **Application**

## 2. Mendisain form GUI

Selanjutnya merancang GUI dengan memilih komponen yang diinginkan pada component palette (gambar 1.8) dan diletakkan pada form designer (gambar 1.7) sesuai dengan keperluannya, misal seperti yang ditunjukkan oleh gambar berikut.



Terdapat 4 komponen yang digunakan, yaitu 1 object edit dan 3 object button. Setelah object terpasang, selanjutnya aturlah nilai properti dari masing-masing object sesuai dengan tabel berikut:

Object	Font Size	Caption
Edit1	Tahoma, 14	-
Button1	Tahoma, 14	Mode1
Button2	Tahoma, 14	Mode2
Button3	Tahoma, 14	Tutup

Kemudian untuk setiap button object, tulis kode program berikut pada bagian events di Object Inspector nya.

### Events Button1 → action: **OnClick**

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  Edit1.Text := 'hello world';
end;
```

### Events Button2 → action: **OnClick**

```
procedure TForm1.Button2Click(Sender: TObject);
begin
  Edit1.Text := 'apa kabar dunia';
end;
```

### Events Button3 → action: **OnClick**

```
procedure TForm1.Button3Click(Sender: TObject);
begin
  close;
end;
```

## 3. Mengkompilasi dan menjalankan aplikasi

Selanjutnya lakukan kompilasi dengan cara menekan tombol **RUN** (F9) pada component palette dan jika tidak ada kesalahan maka akan muncul aplikasi GUI sesuai dengan desain yang sudah dibuat

## 4. Mengamati hasil GUI Delphi

## 1.6 Tugas

### A. Arduino IDE.

1. Buatlah variasi nilai waktu tunda (delay) pada file contoh Blink.ino kemudian amati apa perbedaannya.
2. Catatlah nilai *memory footprint* (RAM dan ROM) pada bagian *compilation message* (gambar 1.3) dari hasil kompilasi kode program Arduino oleh komputer.

### B. Delphi IDE

1. Tambahkan satu komponen **button** untuk fungsi menghapus teks pada bagian **Edit**.
2. Tambahkan satu komponen **label** untuk menampilkan teks yang ada di bagian **Edit**.

## MODUL 2

### ANTARMUKA KOMUNIKASI SERIAL PC - ARDUINO

#### 2.1 Kompetensi Dasar

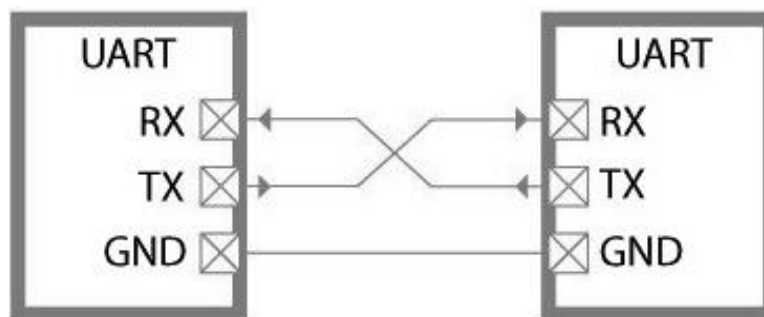
- Mengajarkan kepada mahasiswa tentang cara komunikasi data USB/serial antara papan Arduino dengan PC dan memahami cara mengembangkan perangkat lunak untuk komunikasi data serial dua arah.
- Mengajarkan kepada mahasiswa tentang cara mengembangkan perangkat lunak untuk akuisisi data melalui komunikasi serial dan menampilkan informasi dalam bentuk GUI.

#### 2.2 Indikator

- Mahasiswa mampu mengirim atau menerima informasi dari atau menuju papan Arduino melalui komunikasi data serial/USB.
- Mahasiswa mampu mengirim atau menerima informasi dari atau menuju PC melalui komunikasi data serial/USB.

#### 2.3 Uraian Materi

Komunikasi data serial sangat banyak diterapkan dalam teknologi digital modern karena faktor kecepatan dan kesederhanaan bus komunikasi datanya jika dibandingkan dengan komunikasi data digital secara paralel yang membutuhkan bus data besar dan tidak efektif untuk komunikasi jarak jauh. Komunikasi data digital secara serial terbagi menjadi dua jenis, komunikasi sinkron (synchronous) dan tak sinkron (asynchronous). Komunikasi data digital serial sinkron membutuhkan jalur khusus untuk sinkronisasi pewaktuan transmisi data serial sehingga pola komunikasi bersifat half duplex, yaitu komunikasi dilakukan secara bergantian. Sedangkan komunikasi data digital serial tak sinkron tidak memerlukan sinkronisasi pewaktuan transmisi data dengan asumsi bahwa dua bagian yang berkomunikasi memiliki kecepatan transmisi data (baud rate) yang sama, sehingga komunikasi dapat dilakukan secara bersamaan (full duplex).



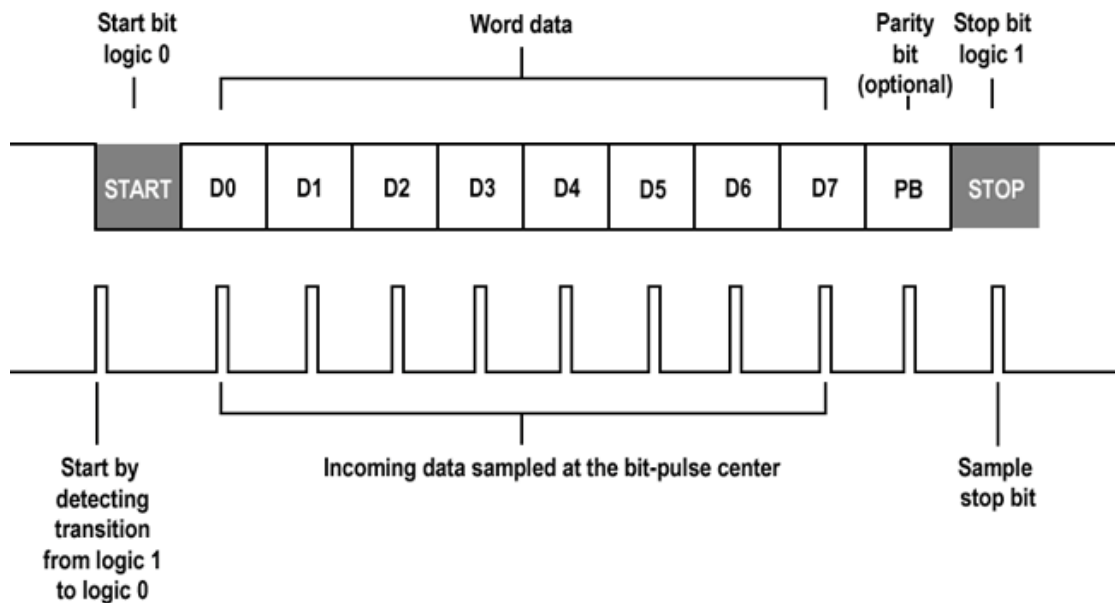
**Gbr. 2.1** Bus komunikasi digital serial tak sinkron UART

Gambar 6.1 di atas menunjukkan bus komunikasi serial tak sinkron antara dua piranti yang hanya memerlukan 3 jalur pengkawatan dalam bus komunikasi. Hal ini lah yang membuat komunikasi data serial tak sinkron banyak digunakan dari pada komunikasi data serial sinkron.



Dengan karakteristik bus data komunikasi data tak sinkron seperti yang ditunjukkan oleh Gambar 2.1 memungkinkan untuk diterapkan pada komunikasi nir-kabel dalam jarak yang jauh.

Baik PC maupun Arduino menerapkan komunikasi data serial jenis tak sinkron, dengan pertimbangan fleksibilitas dan kesederhanaan pola komunikasinya. Komunikasi data serial tak sinkron memerlukan format data setiap paket transmisi agar dua piranti yang saling berkomunikasi bisa memahami paket data yang dikirim/diterima. Gambar 2.2 menunjukkan format paket transmisi setiap Byte data pada komunikasi serial tak sinkron.



**Gbr. 2.2** Format paket transmisi data pada komunikasi serial tak sinkron.

Berdasarkan Gambar 2.2 dapat dipahami bahwa periode setiap bit data pada komunikasi tak sinkron harus disepakati oleh dua piranti yang berkomunikasi. Lebar bit data menentukan kecepatan transmisi serial yang dikenal dengan nama baud rate. Jika nilai baud rate tidak disepakati oleh dua piranti maka komunikasi data serial tidak akan berjalan dengan benar.

## 2.4 Peralatan Praktikum

- A. 1 PC
- B. 1 Trainer Sistem Antarmuka Komputer
- C. Software Arduino IDE
- D. Software Delphi 7

## 2.5 Langkah Praktikum

### A. Pemrograman Arduino IDE

Lakukan langkah – langkah seperti berikut ini:

1. Menghubungkan papan Arduino dengan PC/Laptop melalui kabel USB.

Pada tahap ini papan Arduino tidak perlu diberikan daya listrik melalui konektor power supply karena saat terhubung dengan PC/Laptop melalui kabel USB maka papan Arduino secara langsung mendapatkan daya listrik. Kecuali jika papan Arduino terhubung dengan beban listrik yang besar (misal: motor servo, dll) maka harus diberikan daya listrik eksternal.

2. Membuka aplikasi Arduino IDE.

Setelah aplikasi Arduino IDE berhasil dijalankan, selanjutnya buat kode program baru seperti listing program berikut:

```
void setup() {  
    Serial.begin(9600);  
}  
  
void loop() {  
    if (Serial.available()) {  
        Serial.write(Serial.read());  
    }  
}
```

3. Mengatur koneksi Arduino IDE dengan papan Arduino.

Hal pertama yang harus dilakukan adalah memilih jenis papan Arduino yang terhubung dengan PC/Laptop. Langkahnya:

Sorot: **Tools** → **Board**: “ ? “ dan pilih jenis papan Arduino yang digunakan.

Tahap selanjutnya adalah memilih port USB PC/Laptop yang sudah terhubung dengan papan Arduino. Langkahnya:

Sorot: **Tools** → **Port**: “ ? “ dan pilih nama port pada PC/Laptop.

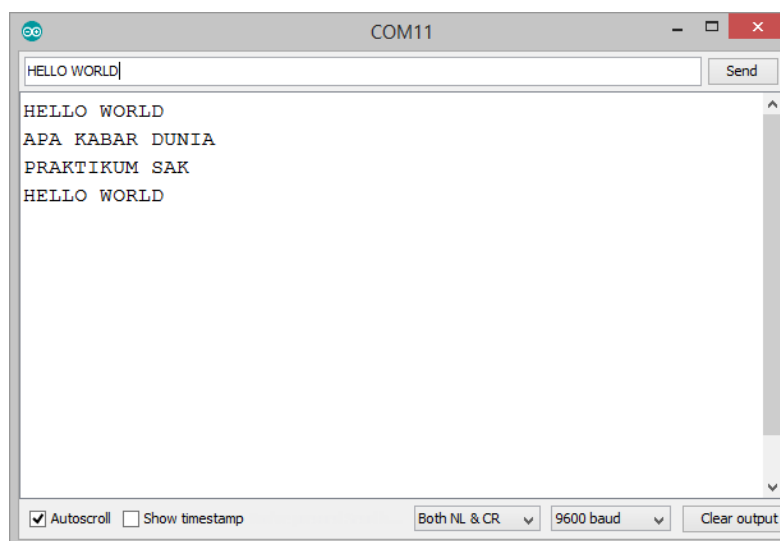
4. Melakukan kompilasi kode program dan upload kode mesin menuju papan Arduino.

Setelah contoh kode program tertampil pada Sketch Editor, selanjutnya lakukan kompilasi kode program dengan menekan ikon kompilasi seperti yang terlihat pada Gambar 1.3. Jika tidak ada notifikasi kesalahan selanjutnya dilakukan proses upload kode mesin hasil kompilasi menuju papan Arduino yang sudah dihubungkan ke PC/Laptop dengan menekan ikon upload seperti yang terlihat pada Gambar 1.3.

5. Mengamati komunikasi data serial, dengan cara:

Sorot: **Tools** → **Serial Monitor**

Ketik teks dan klik send jika tidak ada kesalahan, maka apapun teks yang dikirimkan ke Arduino akan dikirimkan balik dan ditampilkan pada PC seperti yang ditunjukkan oleh gambar berikut ini.



## B. Pemrograman Delphi IDE

Lakukan langkah – langkah sebagai berikut:

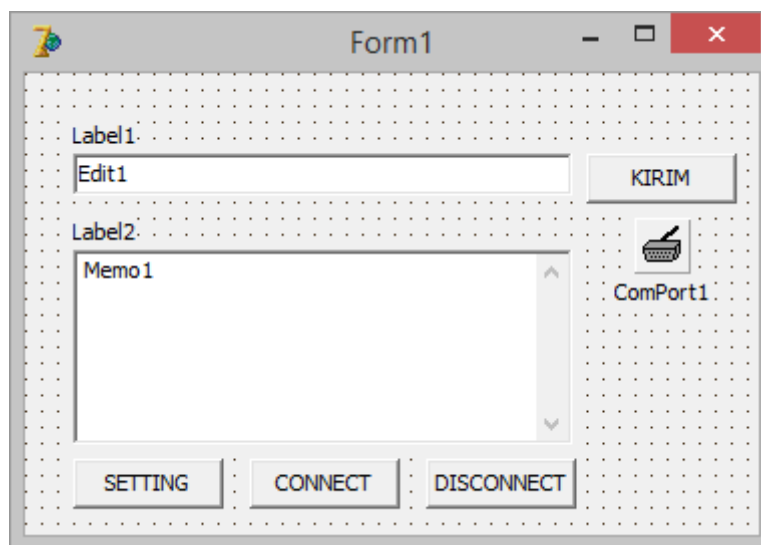
### 1. Membuka aplikasi Delphi IDE

Setelah aplikasi Delphi IDE berhasil dijalankan, selanjutnya silahkan anda buat aplikasi GUI. Adapun langkahnya:

Sorot: **File** → **New** → **Application**

### 2. Mendisain form GUI

Selanjutnya merancang GUI dengan memilih komponen yang diinginkan pada component pallette (gambar 1.8) dan diletakkan pada form designer (gambar 1.7) sesuai dengan keperluannya, misal seperti yang ditunjukkan oleh gambar berikut.



Terdapat 7 komponen yang digunakan, yaitu 2 object **label**, 1 object **edit**, 1 object **memo** dan 4 object **button**, dan 1 object **ComPort** (library Delphi untuk melayani komunikasi data serial). Setelah object terpasang, selanjutnya aturlah nilai properti dari masing-masing object sesuai dengan tabel berikut:

Object	Font Size	Caption
Label1	Tahoma, 14	PERINTAH
Label2	Tahoma, 14	RESPON
Edit1	Tahoma, 14	-
Memo1	Tahoma, 14	-
Button1	Tahoma, 14	KIRIM
Button2	Tahoma, 14	SETTING
Button3	Tahoma, 14	CONNECT
Button4	Tahoma, 14	DISCONNECT
ComPort1	-	-

Kemudian untuk setiap button object, tulis kode program berikut pada bagian events di Object Inspector nya. Tidak semua object dibuat prosedur event, misal: object **label**, **edit**, dan **memo** tidak memerlukan prosedur. Prosedur dibuat hanya untuk object yang ada interaksi dengan pengguna, yaitu **button**.

**Events Form1 → action: OnCreate**

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  Button1.Enabled := False;
  Button2.Enabled := True;
  Button3.Enabled := True;
  Button4.Enabled := False;
  Edit1.Clear;
  Memo1.Clear;
end;
```

**Events Button1 → action: OnClick**

```
procedure TForm1.Button1Click(Sender: TObject);
var
  mydata : string;
begin
  mydata := Edit1.Text + #13#10;
  ComPort1.WriteStr(mydata);
  Edit1.Clear;
end;
```

**Events Button2 → action: OnClick**

```
procedure TForm1.Button2Click(Sender: TObject);
begin
  ComPort1.ShowSetupDialog;
end;
```

**Events Button3 → action: OnClick**

```
procedure TForm1.Button3Click(Sender: TObject);
begin
  ComPort1.Open;
  Button1.Enabled := True;
  Button2.Enabled := False;
  Button3.Enabled := False;
  Button4.Enabled := True;
end;
```

**Events Button4 → action: OnClick**

```
procedure TForm1.Button4Click(Sender: TObject);
begin
  ComPort1.Close;
  Button1.Enabled := False;
  Button2.Enabled := True;
  Button3.Enabled := True;
  Button4.Enabled := False;
end;
```

**Events ComPort1 → action: OnRxChar**

```
procedure TForm1.ComPort1RxChar(Sender: TObject; Count: Integer);
var
  data_masuk : string;
begin
  ComPort1.ReadStr(data_masuk , Count);
  Memo1.Text := Memo1.Text + data_masuk;
end;
```

3. Mengkompilasi dan menjalankan aplikasi

Selanjutnya lakukan kompilasi dengan cara menekan tombol **RUN** (F9) pada component pallete dan jika tidak ada kesalahan maka akan muncul aplikasi GUI sesuai dengan desain yang sudah dibuat

4. Mengamati hasil GUI Delphi

## 2.6 Tugas

### A. Arduino IDE.

1. Buatlah program pada Arduino yang hanya akan merespon dengan mengirimkan data serial ke PC jika sudah menerima data serial yang sesuai dengan **keyword** atau **perintah**.
2. Catatlah nilai *memory footprint* (RAM dan ROM) pada bagian *compilation message* (gambar 1.3) dari hasil kompilasi kode program Arduino oleh komputer.

### B. Delphi IDE

1. Tambahkan satu komponen **button** untuk menutup aplikasi.
2. Tambahkan satu komponen **ListBox** yang berisi teks-teks yang bisa dipilih untuk dikirimkan ke port serial.

## MODUL 3

### AKSES INPUT – OUTPUT DIGITAL ARDUINO OLEH PC

#### 3.1 Kompetensi Dasar

- Mengajarkan kepada mahasiswa tentang cara komunikasi data USB/serial antara papan Arduino dengan PC dan memahami cara mengembangkan perangkat lunak untuk komunikasi data serial dua arah.
- Mengajarkan kepada mahasiswa tentang cara mengembangkan perangkat lunak untuk akses input – output digital pada Arduino oleh PC melalui komunikasi serial dan menampilkan informasi dalam bentuk GUI.

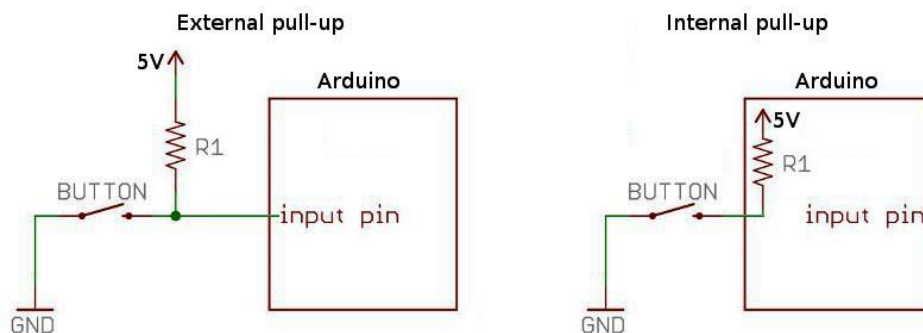
#### 3.2 Indikator

- Mahasiswa mampu mengirim atau menerima informasi dari atau menuju papan Arduino melalui komunikasi data serial/USB.
- Mahasiswa mampu mengakses input – output digital yang terbaca atau terkendali oleh PC melalui komunikasi data serial/USB.

#### 3.3 Uraian Materi

##### 3.3.1 Input Digital

Arduino memiliki salah satu fasilitas penting selain ADC, yaitu GPIO (General Purpose Input Output). Fasilitas ini memiliki fungsi serbaguna yaitu bisa menerima input atau mengeluarkan output digital secara bergantian tergantung pada pengaturan fungsinya oleh kode program perangkat lunak. Berbeda dengan ADC yang menerima masukan analog, pin masukan digital Arduino hanya mampu mengenali data digital saja, yaitu kondisi HIGH (+5 volt) dan LOW (0 volt). Pada papan Arduino UNO, pin masukan digital tidak hanya pin 1 – 13 saja, namun pin masukan analog juga bisa digunakan sebagai pin masukan digital. Secara umum pin masukan digital Arduino memiliki resistor pull\_up internal yaitu sebuah resistor yang menarik arus dari catu positif menuju catu negatif, namun secara default resistor ini tidak diaktifkan. Sehingga perlu dipertimbangkan penggunaannya berdasarkan karakteristik masukan dari piranti eksternal.

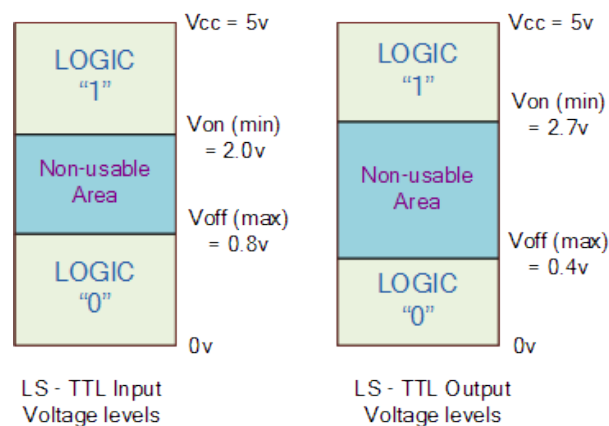


Gbr. 3.1 Penggunaan resistor pull up pada masukan digital Arduino

Jika piranti masukan digital yang terhubung dengan papan Arduino mampu mengalirkan arus listrik menuju pin masukan digital Arduino maka resistor pull\_up tidak diperlukan, namun jika piranti masukan digital hanya bisa menyedot arus listrik dari pin masukan digital Arduino maka resistor pull\_up internal harus digunakan agar terdapat beda potensial yang terbaca sebagai nilai HIGH atau LOW.

### 3.3.2 Output Digital

Keluaran digital merupakan fungsi dasar semua piranti digital seperti halnya masukan digital. GPIO Arduino memberikan fleksibilitas tinggi terhadap penggunaan pin apakah sebagai masukan atau sebagai keluaran dalam satu waktu. Pada dasarnya fungsi keluaran digital menghasilkan logika HIGH dan LOW yang mewakili tegangan 0 volts dan +5 volts. Semua pin GPIO pada papan Arduino UNO dapat dijadikan keluaran digital, termasuk pin masukan analog A0 – A5.



**Gbr. 3.2** Karakteristik digital GPIO Arduino

Berdasarkan Gambar 3.2 yang menunjukkan karakteristik digital GPIO untuk masukan dan keluaran, dapat dilihat bahwa logika keluaran HIGH memiliki rentang tegangan antara 2.7 volts – 5 volts, sedangkan untuk logika keluaran LOW memiliki rentang tegangan antara 0 volts – 0.4 volts. Dan tegangan diantaranya adalah tidak dikenali.

### 3.4 Peralatan Praktikum

- A. 1 PC
- B. 1 Trainer Sistem Antarmuka Komputer
- C. 1 Logic Analyzer
- D. Software Arduino IDE
- E. Software Delphi 7

### 3.5 Langkah Praktikum

#### A. Pemrograman Arduino IDE

Lakukan langkah – langkah seperti berikut ini:

1. Menghubungkan papan Arduino dengan PC/Laptop melalui kabel USB.

Pada tahap ini papan Arduino tidak perlu diberikan daya listrik melalui konektor power supply karena saat terhubung dengan PC/Laptop melalui kabel USB maka papan Arduino secara langsung mendapatkan daya listrik. Kecuali jika papan Arduino terhubung dengan beban listrik yang besar (misal: motor servo, dll) maka harus diberikan daya listrik eksternal.

## 2. Membuka aplikasi Arduino IDE.

Setelah aplikasi Arduino IDE berhasil dijalankan, selanjutnya buat kode program baru seperti listing program berikut:

```
#define LED 13
#define SW1 2
#define SW2 3

void setup() {
  Serial.begin(9600);
  pinMode(LED, OUTPUT);
  pinMode(SW1, INPUT_PULLUP);
  pinMode(SW2, INPUT_PULLUP);
}

void loop() {
  if (Serial.available()) {
    char cmd = Serial.read();
    if (cmd == 'A'){
      digitalWrite(LED, HIGH);
    }
    else if (cmd == 'B'){
      digitalWrite(LED, LOW);
    }
  }

  if (digitalRead(SW1) == LOW){
    Serial.println("Q");
  }
  else{
    Serial.println("W");
  }

  if (digitalRead(SW2) == LOW){
    Serial.println("Z");
  }
  else{
    Serial.println("X");
  }

  delay(100);
}
```

## 3. Mengatur koneksi Arduino IDE dengan papan Arduino.

Hal pertama yang harus dilakukan adalah memilih jenis papan Arduino yang terhubung dengan PC/Laptop. Langkahnya:

Sorot: **Tools** → **Board**: “ ? “ dan pilih jenis papan Arduino yang digunakan.

Tahap selanjutnya adalah memilih port USB PC/Laptop yang sudah terhubung dengan papan Arduino. Langkahnya:

Sorot: **Tools** → **Port**: “ ? “ dan pilih nama port pada PC/Laptop.

## 4. Melakukan kompilasi kode program dan upload kode mesin menuju papan Arduino.

Setelah contoh kode program tertampil pada Sketch Editor, selanjutnya lakukan kompilasi kode program dengan menekan ikon kompilasi seperti yang terlihat pada Gambar 1.3. Jika tidak ada notifikasi kesalahan selanjutnya dilakukan proses upload

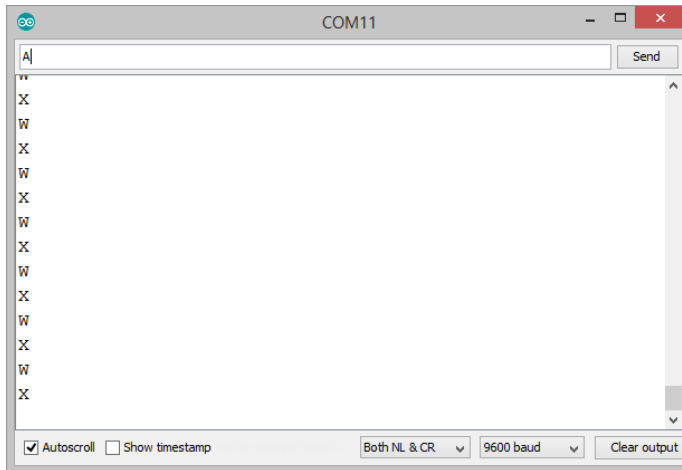


kode mesin hasil kompilasi menuju papan Arduino yang sudah dihubungkan ke PC/Laptop dengan menekan ikon upload seperti yang terlihat pada Gambar 1.3.

5. Mengamati komunikasi data serial, dengan cara:

Sorot: **Tools** → **Serial Monitor**

Ketik **keyword text** (**A** = LED ON & **B** = LED OFF) dan klik send, jika tidak ada kesalahan, maka LED pada papan Arduino akan ON dan OFF sesuai perintah yang diberikan. Dan Arduino akan selalu mengirimkan status input digital kepada PC (**Q/Z** = input digital terpicu & **W/X** = 2 input digital tak terpicu) seperti yang ditunjukkan oleh gambar berikut ini.



## B. Pemrograman Delphi IDE

Lakukan langkah – langkah sebagai berikut:

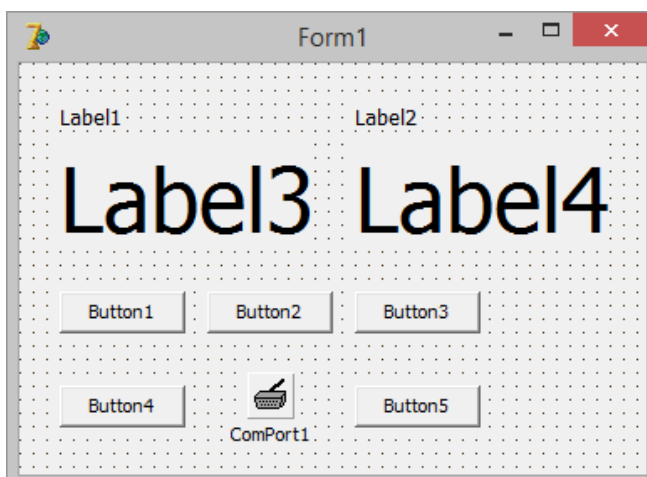
1. Membuka aplikasi Delphi IDE

Setelah aplikasi Delphi IDE berhasil dijalankan, selanjutnya silahkan anda buat aplikasi GUI. Adapun langkahnya:

Sorot: **File** → **New** → **Application**

2. Mendisain form GUI

Selanjutnya merancang GUI dengan memilih komponen yang diinginkan pada component palette (gambar 1.8) dan diletakkan pada form designer (gambar 1.7) sesuai dengan keperluannya, misal seperti yang ditunjukkan oleh gambar berikut.



Terdapat 10 komponen yang digunakan, yaitu 4 object **label**, 5 object **button**, dan 1 object **ComPort** (library Delphi untuk melayani komunikasi data serial). Setelah object terpasang, selanjutnya aturlah nilai properti dari masing-masing object sesuai dengan tabel berikut:

Object	Font Size	Caption
Label1	Tahoma, 14	INPUT-1
Label2	Tahoma, 14	INPUT-2
Label3	Tahoma, 40	?
Label4	Tahoma, 40	?
Button1	Tahoma, 14	SETTING
Button2	Tahoma, 14	CONNECT
Button3	Tahoma, 14	DISCONNECT
Button4	Tahoma, 14	LED ON
Button5	Tahoma, 14	LED OFF
ComPort1	-	-

Kemudian untuk setiap button object, tulis kode program berikut pada bagian events di Object Inspector nya. Tidak semua object dibuat prosedur event, misal: object **label**, **edit**, dan **memo** tidak memerlukan prosedur. Prosedur dibuat hanya untuk object yang ada interaksi dengan pengguna, yaitu **button**.

#### Events Form1 → action: **OnCreate**

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  Button1.Enabled := True;
  Button2.Enabled := True;
  Button3.Enabled := False;
  Button4.Enabled := False;
  Button5.Enabled := False;
end;
```

#### Events Button1 → action: **OnClick**

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  ComPort1.ShowSetupDialog;
end;
```

#### Events Button2 → action: **OnClick**

```
procedure TForm1.Button2Click(Sender: TObject);
begin
  ComPort1.Open;
  Button1.Enabled := True;
  Button2.Enabled := False;
  Button3.Enabled := True;
  Button4.Enabled := True;
  Button5.Enabled := True;
end;
```

**Events Button3 → action: **OnClick****

```
procedure TForm1.Button3Click(Sender: TObject);
begin
  ComPort1.Close;
  Button1.Enabled := True;
  Button2.Enabled := True;
  Button3.Enabled := False;
  Button4.Enabled := False;
  Button5.Enabled := False;
end;
```

**Events Button4 → action: **OnClick****

```
procedure TForm1.Button4Click(Sender: TObject);
begin
  ComPort1.WriteStr('A');
end;
```

**Events Button5 → action: **OnClick****

```
procedure TForm1.Button5Click(Sender: TObject);
begin
  ComPort1.WriteStr('B');
end;
```

**Events ComPort1 → action: **OnRxChar****

```
procedure TForm1.ComPort1RxChar(Sender: TObject; Count: Integer);
var
  buffer : string;
  cmd : char;
  i : integer;
begin
  ComPort1.ReadStr(buffer, Count);
  for i:=1 to Count do
  begin
    cmd := buffer[i];
    case cmd of
      'Q' : Label3.Caption := 'LOW';
      'W' : Label3.Caption := 'HIGH';
      'Z' : Label4.Caption := 'LOW';
      'X' : Label4.Caption := 'HIGH';
    end;
  end;
end;
```

3. Mengkompilasi dan menjalankan aplikasi

Selanjutnya lakukan kompilasi dengan cara menekan tombol **RUN** (F9) pada component pallete dan jika tidak ada kesalahan maka akan muncul aplikasi GUI sesuai dengan desain yang sudah dibuat

4. Hubungkan pin 3 dan 4 pada papan Arduino menuju GROUND menggunakan kabel jumper secara bergantian, kemudian amati aplikasi GUI Delphi yang sudah di jalankan.

### 3.6 Tugas

#### A. Arduino IDE.

1. Ubah keyword untuk kendali LED pada papan Arduino (selain “A”-“B”).
2. Tambahkan 2 lagi pin input digital (selain pin 2-3) dan juga keyword untuk status kondisi logika input digital nya (selain “Q”-“W”-“Z”-“X”).
3. Catatlah nilai *memory footprint* (RAM dan ROM) pada bagian *compilation message* (gambar 1.3) dari hasil kompilasi kode program Arduino oleh komputer.

#### B. Delphi IDE

1. Sesuaikan kode program Delphi dengan perubahan yang dilakukan pada papan Arduino.
2. Tambahkan 2 komponen **Label** untuk menampilkan status logika input digital yang baru dari papan Arduino.

## MODUL 4

### AKSES INPUT – OUTPUT ANALOG ARDUINO OLEH PC

#### 4.1 Kompetensi Dasar

- Mengajarkan kepada mahasiswa tentang cara komunikasi data USB/serial antara papan Arduino dengan PC dan memahami cara mengembangkan perangkat lunak untuk komunikasi data serial dua arah.
- Mengajarkan kepada mahasiswa tentang cara mengembangkan perangkat lunak untuk akses input – output analog pada Arduino oleh PC melalui komunikasi serial dan menampilkan informasi dalam bentuk GUI.

#### 4.2 Indikator

- Mahasiswa mampu mengirim atau menerima informasi dari atau menuju papan Arduino melalui komunikasi data serial/USB.
- Mahasiswa mampu mengakses input – output analog yang terbaca atau terkendali oleh PC melalui komunikasi data serial/USB.

#### 4.3 Uraian Materi

##### 4.3.1 Input Analog

Arduino memiliki salah satu fasilitas penting, yaitu ADC (*Analog to Digital Converter*). Fasilitas ini bertugas untuk mengubah besaran input analog menjadi digital, karena pada dasarnya pengendali mikro adalah piranti digital yang tidak bisa mengenali data yang bersifat analog. Besaran alam (contoh: suhu, tekanan udara, kelembaban, dll) dapat dibaca menggunakan sensor yang mengeluarkan arus atau tegangan secara analog, padahal pengendali mikro hanya bisa mengenali data digital. Untuk itulah fungsi ADC diperlukan, agar pengendali mikro bisa membaca besaran alamiah yang terbaca oleh sensor analog.

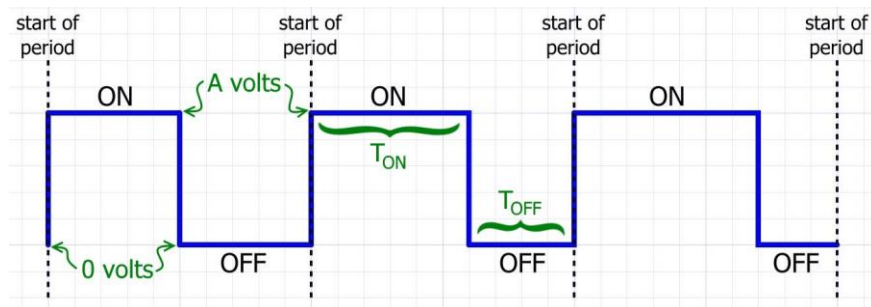
Fasilitas ADC yang digunakan pada papan Arduino 8-bit adalah ADC 10-bit sebanyak 6 kanal yang terdapat pada pin A0, A1, A2, A3, A4, A5. Selain pin masukan analog, pada papan Arduino juga terdapat pin masukan tegangan referensi untuk ADC yaitu pin AREF. Secara default, tegangan referensi ADC Arduino adalah +5.0 volt, namun dengan menggunakan pin AREF bisa diberikan masukan bervariasi. Kalkulasi konversi analog ke digital oleh ADC mengikuti kaidah yang ditunjukkan oleh Persamaan 4.1 berikut ini.:

$$Digital_{OUT} = \frac{2^N \times V_{masukan\_analog}}{V_{Referensi}} \quad (4.1)$$

Data digital keluaran memiliki satuan bit, sedangkan N menunjukkan jumlah bit ADC yang dimiliki oleh papan Arduino, yaitu N=10 untuk jenis Arduino dengan pengendali mikro 8-bit.  $V_{analog}$  adalah tegangan analog yang terima oleh ADC, sedangkan  $V_{referensi}$  adalah tegangan referensi yang digunakan oleh papan Arduino pada pin VREF (default = +5.0 volt).

### 4.3.2 Output Analog

PWM atau lebih dikenal dengan modulasi lebar pulsa, yaitu sebuah metode pengaturan lebar pulsa secara variabel untuk sebuah nilai frekwensi yang tetap. Berbeda halnya dengan modulasi frekwensi (FM) atau modulasi amplitudo (AM) yang mengatur variasi frekwensi atau tegangan, PWM menggunakan gelombang kotak (square wave) dengan amplitudo dan frekwensi yang tetap namun rasio periode ON terhadap total periode gelombang atau disebut dengan DUTY CYCLE bisa diatur dalam satuan persen (%). Gambar 4.1 menunjukkan bentuk gelombang PWM.



Gbr. 4.1 Bentuk gelombang PWM

Berdasarkan Gambar 4.1 yang menunjukkan bentuk gelombang kotak PWM, dapat dilihat bahwa periode ON pada satu siklus gelombang bisa melebar dan menyempit tanpa merubah frekwensi dan amplitudonya. Variasi lebar pulsa PWM dapat dihitung secara matematis mengikuti kaidah yang ditunjukkan oleh Persamaan 4.2 berikut.

$$duty\ cycle\ (\%) = \frac{T_{ON}}{T_{ON} + T_{OFF}} \quad (4.2)$$

Pada papan Arduino jenis 8-bit terdapat beberapa pin yang bisa dijadikan sebagai keluaran PWM, yaitu: 3, 5, 6, 9, 10, 11. Dan terdapat dua jenis frekwensi PWM yang berbeda, yaitu: 488 Hz untuk pin 3, 9, 10, 11, dan 977 Hz untuk pin 5, 6. Rentang amplitudo PWM Arduino hampir sama dengan tegangan catu komponen digital, yaitu: 0 – 5 volts. Sedangkan rentang nilai digital yang merepresentasikan *duty cycle* pada Arduino yaitu: 0 – 255 untuk 0% – 100% rasio.

### 4.4 Peralatan Praktikum

- A. 1 PC
- B. 1 Trainer Sistem Antarmuka Komputer
- C. 1 Logic Analyzer
- D. Software Arduino IDE
- E. Software Delphi 7

### 4.5 Langkah Praktikum

#### A. Pemrograman Arduino IDE

Lakukan langkah – langkah seperti berikut ini:

1. Menghubungkan papan Arduino dengan PC/Laptop melalui kabel USB.

Pada tahap ini papan Arduino tidak perlu diberikan daya listrik melalui konektor power supply karena saat terhubung dengan PC/Laptop melalui kabel USB maka papan Arduino secara langsung mendapatkan daya listrik. Kecuali jika papan

Arduino terhubung dengan beban listrik yang besar (misal: motor servo, dll) maka harus diberikan daya listrik eksternal.

2. Membuka aplikasi Arduino IDE.

Setelah aplikasi Arduino IDE berhasil dijalankan, selanjutnya buat kode program baru seperti listing program berikut:

```
#define LED 13
#define AIN A0
#define PWM 3

int adc;
char txt;
String val;

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

void loop() {
  if (Serial.available()) {
    txt = Serial.read();
    if (txt == '*'){
      val = "";
    }
    else if (txt == '#'){
      analogWrite(val.toInt(),PWM);
      val = "";
    }
    else{
      val = val + txt;
    }
  }

  adc = analogRead(AIN);
  Serial.print('*');
  Serial.print(String(adc));
  Serial.print('#');
  delay(100);
}
```

3. Mengatur koneksi Arduino IDE dengan papan Arduino.

Hal pertama yang harus dilakukan adalah memilih jenis papan Arduino yang terhubung dengan PC/Laptop. Langkahnya:

Sorot: **Tools** → **Board:** “ ? “ dan pilih jenis papan Arduino yang digunakan.

Tahap selanjutnya adalah memilih port USB PC/Laptop yang sudah terhubung dengan papan Arduino. Langkahnya:

Sorot: **Tools** → **Port:** “ ? “ dan pilih nama port pada PC/Laptop.

4. Melakukan kompilasi kode program dan upload kode mesin menuju papan Arduino.

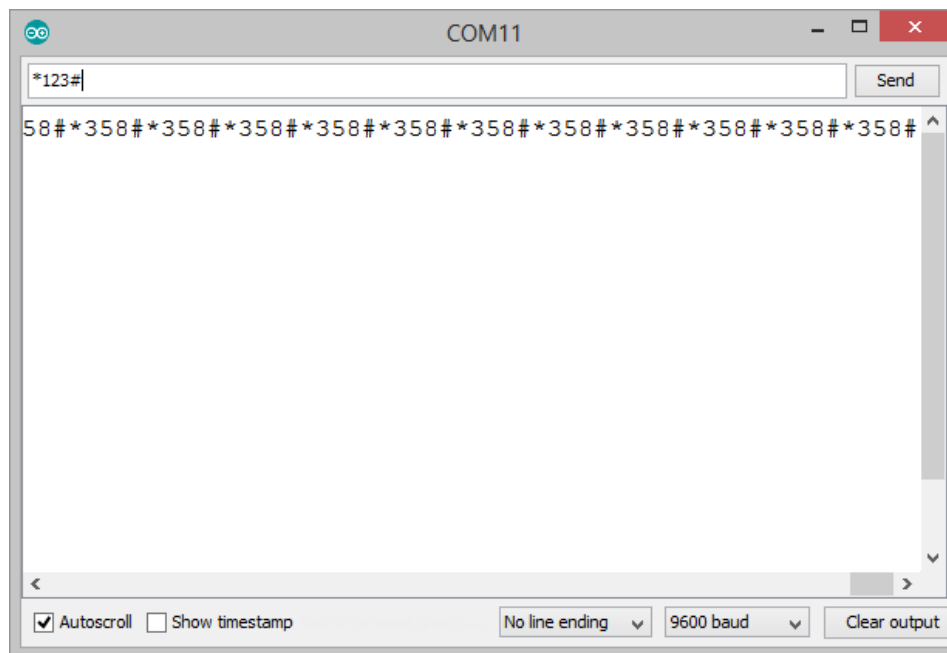
Setelah contoh kode program tertampil pada Sketch Editor, selanjutnya lakukan kompilasi kode program dengan menekan ikon kompilasi seperti yang terlihat pada

Gambar 1.3. Jika tidak ada notifikasi kesalahan selanjutnya dilakukan proses upload kode mesin hasil kompilasi menuju papan Arduino yang sudah dihubungkan ke PC/Laptop dengan menekan ikon upload seperti yang terlihat pada Gambar 1.3.

5. Mengamati komunikasi data serial, dengan cara:

Sorot: **Tools** → **Serial Monitor**

Ketik **keyword text** (*\*nilai\_pwm#*) dan klik send, contoh: \*123#. Jika tidak ada kesalahan, maka pin 3 pada papan Arduino akan mengeluarkan sinyal PWM dengan nilai duty cycle sesuai dengan yang diberikan (dibuktikan menggunakan piranti Logic Analyzer). Dan Arduino akan selalu mengirimkan nilai input analog yang terbaca oleh ADC Arduino kepada PC (antara **0 - 1023**) seperti yang ditunjukkan oleh gambar berikut ini.



## B. Pemrograman Delphi IDE

Lakukan langkah – langkah sebagai berikut:

1. Membuka aplikasi Delphi IDE

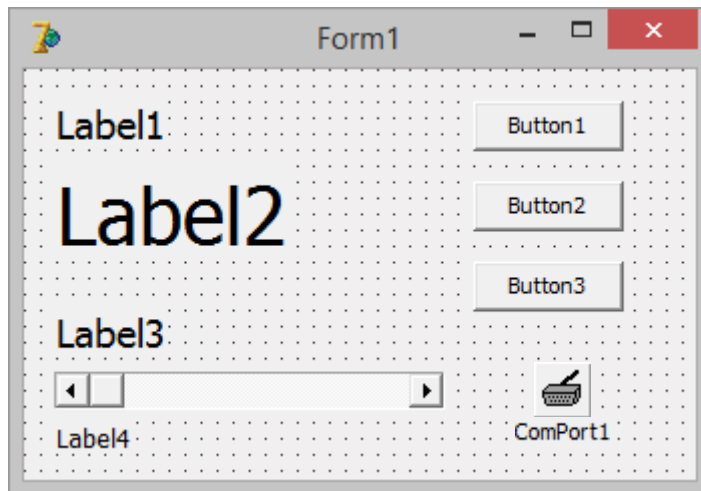
Setelah aplikasi Delphi IDE berhasil dijalankan, selanjutnya silahkan anda buat aplikasi GUI. Adapun langkahnya:

Sorot: **File** → **New** → **Application**

2. Mendisain form GUI

Selanjutnya merancang GUI dengan memilih komponen yang diinginkan pada component pallette (gambar 1.8) dan diletakkan pada form designer (gambar 1.7) sesuai dengan keperluannya, misal seperti yang ditunjukkan oleh gambar berikut.





Terdapat 9 komponen yang digunakan, yaitu 4 object **label**, 3 object **button**, 1 object **ScrollBar**, dan 1 object **ComPort** (library Delphi untuk melayani komunikasi data serial). Setelah object terpasang, selanjutnya aturlah nilai properti dari masing-masing object sesuai dengan tabel berikut:

Object	Font Size	Caption
Label1	Tahoma, 10	ANALOG INPUT
Label2	Tahoma, 30	?
Label3	Tahoma, 10	ANALOG OUTPUT
Label4	Tahoma, 10	?
ScrollBar	-	-
Button1	Tahoma, 14	SETTING
Button2	Tahoma, 14	CONNECT
Button3	Tahoma, 14	DISCONNECT
ComPort1	-	-

Kemudian untuk setiap button object, tulis kode program berikut pada bagian events di Object Inspector nya. Tidak semua object dibuat prosedur event, misal: object **label**, **edit**, dan **memo** tidak memerlukan prosedur. Prosedur dibuat hanya untuk object yang ada interaksi dengan pengguna, yaitu **button**.

#### Events Form1 → action: **OnCreate**

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  Button1.Enabled := True;
  Button2.Enabled := True;
  Button3.Enabled := False;
  ScrollBar1.Enabled := False;
end;
```

#### Events Button1 → action: **OnClick**

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  ComPort1.ShowSetupDialog;
end;
```

**Events Button2 → action: `OnClick`**

```
procedure TForm1.Button2Click(Sender: TObject);
begin
  ComPort1.Open;
  Button1.Enabled := False;
  Button2.Enabled := False;
  Button3.Enabled := True;
  ScrollBar1.Enabled := True;
end;
```

**Events Button3 → action: `OnClick`**

```
procedure TForm1.Button3Click(Sender: TObject);
begin
  ComPort1.Close;
  Button1.Enabled := True;
  Button2.Enabled := True;
  Button3.Enabled := False;
  ScrollBar1.Enabled := False;
end;
```

**Events ScrollBar1 → action: `OnChange`**

```
procedure TForm1.ScrollBar1Change(Sender: TObject);
var
  pwm : string;
begin
  Label4.Caption := inttostr(ScrollBar1.Position);
  pwm := '*' + Label4.Caption + '#';
  ComPort1.WriteString(pwm);
end;
```

**Events ComPort1 → action: `OnRxChar`**

```
//-----
var
  val : string;          //buat variabel ini sebagai variabel GLOBAL
//-----
procedure TForm1.ComPort1RxChar(Sender: TObject; Count: Integer);
var
  buffer : string;
  hdr : char;
  i : integer;
begin
  ComPort1.ReadStr(buffer, Count);
  for i:=1 to Count do
  begin
    hdr := buffer[i];
    if hdr = '*' then
      val := ''
    else if hdr = '#' then
      begin
        Label2.Caption := val;
        val := '';
      end
    else
      val := val + hdr;
  end;
end;
```

3. Mengkompilasi dan menjalankan aplikasi  
Selanjutnya lakukan kompilasi dengan cara menekan tombol **RUN** (F9) pada component palette dan jika tidak ada kesalahan maka akan muncul aplikasi GUI sesuai dengan desain yang sudah dibuat
4. Hubungkan pin **A0** Arduino dengan sumber analog (output potensiometer pembagi tegangan), kemudian putar-putar potensiometer dan amati aplikasi GUI Delphi yang sudah di jalankan.
5. Pada aplikasi Delphi, geserlah ke kanan atau ke kiri Scroll Bar dan amati sinyal PWM yang dikeluarkan oleh pin 3 Arduino menggunakan piranti Logic Analyzer atau Oscilloscope.

#### 4.6 Tugas

##### A. Arduino IDE.

1. Ubah karakter delimiter data masuk maupun data keluar (selain '\*' dan '#'), dan update kode program Arduino.
2. Tambahkan lagi 1 pin masukan analog (selain pin A0) dan pindahkan pin keluaran analog (selain pin 3), dan update kode program Arduino.
3. Catatlah nilai *memory footprint* (RAM dan ROM) pada bagian *compilation message* (gambar 1.3) dari hasil kompilasi kode program Arduino oleh komputer.

##### B. Delphi IDE

1. Sesuaikan kode program Delphi dengan perubahan yang dilakukan pada papan Arduino.
2. Tambahkan 1 komponen **Label** untuk menampilkan nilai input analog yang baru dari papan Arduino.

## MODUL 5

### PEMROGRAMAN SERIAL I2C: ANTARMUKA RTC DS1307 (BACA DAN SET WAKTU)

#### 5.1 Kompetensi Dasar

Setelah mengikuti praktikum ini, mahasiswa mampu:

- Memahami dasar sistem komunikasi serial I2C (*Inter-integrated Circuit*).
- Membuat program komunikasi serial I2C pada *platform* Arduino UNO pada mode *Synchronous* antara Arduino dan RTC DS1307.
- Mengirimkan data serial I2C 8-bit antara Arduino dengan RTC DS1307.

#### 5.2 Indikator

- Keterampilan menyambungkan perangkat keras (*hardware*) standar komunikasi serial I2C.
- Keterampilan membuat program pembacaan data waktu dari RTC DS1307.
- Keterampilan membuat program pengaturan data waktu ke RTC DS1307.

#### 5.3 Uraian Materi

Serial I2C (*Inter-integrated Circuit*) adalah standar komunikasi serial dua arah menggunakan dua saluran yang didesain 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 pengendalinya. Piranti yang dihubungkan dengan sistem I2C Bus dapat dioperasikan sebagai Master dan Slave. Master adalah piranti yang memulai transfer data pada I2C Bus dengan membentuk sinyal Start, mengakhiri transfer data dengan membentuk sinyal Stop, dan membangkitkan sinyal clock. Slave adalah piranti yang dialamati master. Pada *platform* Arduino UNO terdapat fitur I2C yang dapat digunakan sebagai media komunikasi serial 8-bit antar devais yang diletakkan pada pin A4 dan A5 berturut – turut untuk pin SCL dan SDA.

Modul RTC (*Real Time Clock*) yang digunakan pada praktikum ini memiliki akurasi dan presisi yang sangat tinggi dalam mencacah waktu (*extremely accurate temperature compensated* RTC). DS1307 memiliki kristal internal dan rangkaian kapasitor *tuning* di mana suhu dari kristal dimonitor secara berkesinambungan dan kapasitor disetel secara otomatis untuk menjaga kestabilan detak frekuensi.

Modul ini juga sudah dilengkapi dengan IC AT24C32 yang memberikan EEPROM tambahan sebesar 4 KB (32.768 bit) yang dapat digunakan untuk berbagai keperluan, misalnya untuk menyimpan jadwal (*time schedule*), menyimpan setelan waktu *alarm*, menyimpan data hari libur pada kalender, merekam absensi, dsb. Alamat dari EEPROM ini dapat disetel dengan menghubungkan-singkatkan pad A0, A1, dan A2 (8 pilihan alamat), secara *default* tersetel di alamat 0x57.

Potongan program untuk melakukan pembacaan data waktu dari RTC terdapat pada Gambar 5.1. Sedangkan potongan program untuk melakukan penulisan data waktu pada RTC terdapat pada Gambar 5.2.

```

#include "Wire.h"
#define DS1307_ADDRESS 0x68

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

void loop(){
  printDate();
  delay(1000);
}

byte bcdToDec(byte val) {
  // Convert binary coded decimal to normal decimal numbers
  return ( (val/16*10) + (val%16) );
}

void printDate(){
  // Reset the register pointer
  Wire.beginTransmission(DS1307_ADDRESS);
  byte zero = 0x00;
  Wire.write(zero);
  Wire.endTransmission();
  Wire.requestFrom(DS1307_ADDRESS, 7);
  int second = bcdToDec(Wire.read());
  int minute = bcdToDec(Wire.read());
  int hour = bcdToDec(Wire.read() & 0b111111); //24 hour time
  int weekDay = bcdToDec(Wire.read()); //0-6 -> sunday - Saturday
  int monthDay = bcdToDec(Wire.read());
  int month = bcdToDec(Wire.read());
  int year = bcdToDec(Wire.read());

  //print the date EG 3/1/11 23:59:59
  Serial.print(month);
  Serial.print("/");
  Serial.print(monthDay);
  Serial.print("/");
  Serial.print(year);
  Serial.print(" ");
  Serial.print(hour);
  Serial.print(":");
  Serial.print(minute);
  Serial.print(":");
  Serial.println(second);
}

```

Gambar 5.1 Potongan Program Pembacaan Waktu Pada RTC

```

#include "Wire.h"
#define DS1307_ADDRESS 0x68

byte zero = 0x00; //workaround for issue #527

void setup(){
  Wire.begin();
  Serial.begin(9600);
  setDateTime(); //MUST CONFIGURE IN FUNCTION
}

void loop(){
  printDate();
  delay(1000);
}

void setDateTime(){
  byte second = 45; //0-59
  byte minute = 40; //0-59
  byte hour = 0; //0-23
  byte weekDay = 2; //1-7
  byte monthDay = 1; //1-31
  byte month = 3; //1-12
  byte year = 11; //0-99
  Wire.beginTransmission(DS1307_ADDRESS);
  Wire.write(zero);
  Wire.write(decToBcd(second));
  Wire.write(decToBcd(minute));
  Wire.write(decToBcd(hour));
  Wire.write(decToBcd(weekDay));
  Wire.write(decToBcd(monthDay));
  Wire.write(decToBcd(month));
  Wire.write(decToBcd(year));
  Wire.write(zero); //start
  Wire.endTransmission();
}

byte decToBcd(byte val){
  // Convert normal decimal numbers to binary coded decimal
  return ( (val/10*16) + (val%10) );
}

byte bcdToDec(byte val) {
  // Convert binary coded decimal to normal decimal numbers
  return ( (val/16*10) + (val%16) );
}

void printDate(){
  // Reset the register pointer
  Wire.beginTransmission(DS1307_ADDRESS);
  Wire.write(zero);
  Wire.endTransmission();
  Wire.requestFrom(DS1307_ADDRESS, 7);
  int second = bcdToDec(Wire.read());
  int minute = bcdToDec(Wire.read());
  int hour = bcdToDec(Wire.read() & 0b111111); //24 hour time
  int weekDay = bcdToDec(Wire.read()); //0-6 -> sunday - Saturday
  int monthDay = bcdToDec(Wire.read());
  int month = bcdToDec(Wire.read());
  int year = bcdToDec(Wire.read());

  //print the date EG 3/1/11 23:59:59
  Serial.print(month);
  Serial.print("/");
  Serial.print(monthDay);
  Serial.print("/");
  Serial.print(year);
  Serial.print(" ");
  Serial.print(hour);
  Serial.print(":");
  Serial.print(minute);
  Serial.print(":");
  Serial.println(second);
}

```

Gambar 5.2 Potongan Program Penulisan Waktu Pada RTC

## 5.4 Peralatan Praktikum

- A. Seperangkat komputer
- B. Koneksi internet
- C. Perangkat lunak Arduino IDE
- D. Arduino UNO
- E. Modul Komunikasi I2C
- F. Modul RTC beserta EEPROM

## 5.5 Skema Praktikum

Mahasiswa menyambungkan kabel jumper pada sambungan pin I2C Arduino UNO. Memastikan sambungan sesuai panduan yaitu dengan sambungan langsung tanpa persilangan antara pin SDA Arduino dengan SDA devais I2C RTC DS1307 maupun EEPROM AT24Cxx, serta memastikan sambungan *ground* dan VCC terhubung dengan baik. Mahasiswa memprogram bahasa C/C++ pada Arduino IDE dan mencari kesalahan pada *syntax program*.

## 5.6 Langkah Praktikum

1. Pelajari materi mengenai serial I2C dari beberapa pustaka yang ada.
2. Pastikan telah *download library* untuk komunikasi I2C pada arduino (*wire.h*)
3. Menyambungkan kabel penghubung pada pin serial I2C Arduino UNO yaitu pin A4 dan A5 berturut untuk SCL dan SDA.
4. Ikuti langkah – langkah pemrograman serial I2C yang akan dipandu pada saat praktikum.
5. Mengkompilasi program dan melihat susunan *syntax program* apakah ada kesalahan atau tidak.
6. Mengunggah file komunikasi serial I2C ke Arduino UNO.

## 5.7 Pertanyaan

1. Lakukan deskripsi singkat mengenai Komunikasi Serial I2C.
2. Lakukan deskripsi singkat menurut anda mengenai teknik komunikasi serial I2C antara PC dengan RTC.
3. Cari tahu mengenai teknik komunikasi dengan EEPROM pada RTC, dan berikan deskripsi singkat langkah-langkahnya.

## MODUL 6

### PEMROGRAMAN SERIAL UART: ANTARMUKA MODUL GPRS

#### 6.1 Kompetensi Dasar

Setelah mengikuti praktikum ini, mahasiswa mampu:

- Memahami dasar sistem komunikasi serial UART.
- Mengirimkan data serial UART mode AT pada *GPRS Module*.
- Mengirimkan dan menerima SMS (*Short Message Service*) dari perangkat Arduino melalui modul GPRS.

#### 6.2 Indikator

- Keterampilan menyambungkan perangkat keras (*hardware*) standar komunikasi serial UART.
- Keterampilan mengirimkan data serial berupa AT *command* pada modul GPRS.
- Keterampilan membuat pengaturan pengiriman dan penerimaan SMS (*Short Message Service*) ke perangkat HP.

#### 6.3 Uraian Materi

##### 6.3.1 Komunikasi Serial UART

Serial UART (*Universal Asynchronous Receiver Transceiver*) merupakan mode komunikasi serial yang banyak digunakan pada teknologi antarmuka antar perangkat hingga saat ini. Jenis komunikasi ini memiliki sistem deret dengan sebuah sistem denyut (*clock*). UART 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. Fitur UART saat ini telah disertakan di dalam beberapa teknologi mikrokontroler termasuk keluarga AVR yang menjadi bagian inti pada *platform* Arduino.

UART adalah protokol komunikasi yang umum digunakan dalam pengiriman data serial antara device satu dengan yang lainnya. Pada *platform* Arduino UNO terdapat 1 kanal UART yang dapat digunakan sebagai media komunikasi serial 8-bit antar devais maupun dengan komputer yang diletakkan pada pin 0 dan 1.

##### 6.3.2 AT Command

*AT Command* merupakan kepanjangan dari *Attention Command*, dan selalu digunakan untuk memulai pengiriman baris perintah dari *Terminal Equipment* (TE) kepada *Terminal Adaptor* (TA). Untuk memulai suatu perintah *AT Command*, diperlukan prefiks “AT” atau “at” dalam setiap perintah *AT Command*, dan diakhiri dengan ”” (= 0x0D). Beberapa perintah *AT Command* dapat dituliskan pada baris yang sama dengan hanya menggunakan satu prefiks “AT” atau “at”, kemudian antar perintah dibatasi oleh karakter “;”.



Beberapa perintah *AT Command* yang digunakan untuk keperluan SMS (pengiriman, penerimaan) seperti pada Tabel 6.1. Potongan program untuk memeriksa koneksi dari modul GPRS dengan PC seperti pada Gambar 6.1. Potongan program untuk mengirim SMS dari modul GPRS dengan serial monitor pada PC ke nomor HP seperti pada Gambar 6.2.

Tabel 6.1 Contoh Perintah AT Command

<b>AT Command</b>	<b>Keterangan</b>
AT	Mengecek apakah Handphone telah terhubung
AT+CMGF	Untuk menetapkan format mode dari terminal
AT+CSCS	Untuk menetapkan jenis encoding
AT+CNMI	Untuk mendeteksi pesan SMS baru masuk secara otomatis
AT+CMGL	Membuka daftar SMS yang ada pada SIM Card
AT+CMGS	Mengirim pesan SMS
AT+CMGR	Membaca pesan SMS
AT+CMGD	Menghapus pesan SMS
ATE1	Mengatur ECHO
ATV1	Mengatur input dan output berupa naskah
AT+CGMI	Mengecek Merek HP
AT+CGMM	Mengecek Seri HP
AT+CGMR	Mengecek Versi Keluaran HP
AT+CBC	Mengecek Baterai
AT+CSQ	Mengecek Kualitas Sinyal
AT+CCLK?	Mengecek Jam (waktu) pada HP
AT+CALM=   	Mengecek Suara/dering HP saat di Telepon (ada Telepon Masuk)  ‘n’ adalah adalah angka yang menunjukkan jenis dering  0 = berdering  1 dan 2 = Silent (Diam)
AT^SCID	Mengecek ID SIM CARD
AT+CGSN	Mengecek Nomor IMEI
AT+CLIP=1	Menampilkan nomor telepon pemanggil
AT+CLCC	Menampilkan nomor telepon yang sedang memanggil
AT+COPN	Menampilkan Nama Sumua Operator di dunia

AT+COPS?	Menampilkan nama operator dari SIM yang digunakan
AT+CPBR=  'n'	Membaca nomor telepon yang disimpan pada buku telepon (SIM CARD)  'n' adalah nomor urut penyimpanan
AT+CPMS=  'md'	Mengatur Memori dari HP  'md' adalah memori yang digunakan  ME = Memori HP  SM = Memori SIM CARD

```
#include <SoftwareSerial.h>
SoftwareSerial SIM800L(2, 3); // RX | TX
// Connect the SIM800L TX to Arduino pin 2 RX.
// Connect the SIM800L RX to Arduino pin 3 TX.

char c = ' ';

void setup()
{
  Serial.begin(9600);
  while(!Serial);
  Serial.println("Arduino with SIM800L is ready");

  SIM800L.begin(9600);
  Serial.println("SIM800L started at 9600"); delay(1000);
  Serial.println("Setup Complete! SIM800L is Ready!");
}

void loop()
{
  if (SIM800L.available())
  {
    c = SIM800L.read();
    Serial.write(c);
  }

  if (Serial.available())
  {
    c = Serial.read();
    SIM800L.write(c);
  }
}
```

Gambar 6.1 Potongan Program Untuk Tes Koneksi

```
#include <SoftwareSerial.h>
SoftwareSerial SIM800L(2, 3); // RX | TX
// Connect the SIM800L TX to Arduino pin 2 RX.
// Connect the SIM800L RX to Arduino pin 3 TX.

void setup() {
  Serial.begin(9600);
  while(!Serial);
  Serial.println("Arduino with SIM800L is ready");
  SIM800L.begin(9600);
  Serial.println("SIM800L started at 9600"); delay(1000);
  Serial.println("Setup Complete! SIM800L is Ready!");
  Serial.println("Set format SMS ke ASCII");
  SIM800L.write("AT+CMGF=1\r\n");
  delay(1000);
  Serial.println("SIM800L Set SMS ke Nomor Tujuan");
  SIM800L.write("AT+CMGS=\"089666699999\"\r\n");
  delay(1000);
  Serial.println("SIM800L Send SMS content");
  SIM800L.write("Testing Kirim SMS via SIM800L");
}
```

Gambar 6.2 Potongan Program Untuk Kirim SMS

#### 6.4 Peralatan Praktikum

- A. Seperangkat komputer
- B. Koneksi internet
- C. Perangkat lunak Arduino IDE
- D. Arduino UNO
- E. Modul Komunikasi UART
- F. Modul GPRS

#### 6.5 Skema Praktikum

Mahasiswa menyambungkan kabel jumper pada sambungan pin UART Arduino UNO. Memastikan sambungan sesuai panduan yaitu dengan menyilangkan/*cross* pin Tx dengan Rx dan sebaliknya antar kedua devais serta memastikan sambungan *ground* bersama terhubung dengan baik. Mahasiswa memprogram bahasa C/C++ pada Arduino IDE dan mencari kesalahan pada *syntax program*.

#### 6.6 Langkah Praktikum

1. Pelajari materi mengenai serial UART dari beberapa pustaka yang ada.
2. Pastikan telah mengunduh *library* koneksi serial UART (*softwareserial.h*)
3. Menyambungkan kabel penghubung pada pin serial UART Arduino UNO yaitu pin 0 dan 1.
4. Ikuti langkah – langkah pemrograman serial UART yang akan dipandu pada saat praktikum.
5. Mengkompilasi program dan melihat susunan *syntax program* apakah ada kesalahan atau tidak.
6. Mengunggah file komunikasi serial UART ke Arduino UNO.
7. Amati hasilnya pada terminal serial dengan menyesuaikan baudrate yang telah ditentukan.
8. Lakukan pengiriman dan penerimaan SMS dan amati hasilnya pad Serial Monitor pada PC.

#### 6.7 Pertanyaan

1. Lakukan deskripsi singkat mengenai Komunikasi Serial UART.
2. Lakukan deskripsi singkat menurut anda mengenai teknik komunikasi serial UART pada Arduino UNO.

## MODUL 7

### PEMROGRAMAN SERIAL SPI: ANTARMUKA ETHERNET

#### 7.1. Kompetensi Dasar

Setelah mengikuti praktikum ini, mahasiswa mampu:

- Memahami dasar sistem komunikasi serial SPI (*Serial Peripheral Interface*).
- Membuat program komunikasi serial SPI pada *platform* Arduino UNO pada mode *Synchronous* antara Arduino dan *wizNET W5100 Ethernet Controller*.
- Mengirimkan data serial SPI 8-bit antara Arduino dengan *wizNET W5100 Ethernet Controller*.

#### 7.2. Indikator

- Keterampilan menyambungkan perangkat keras (*hardware*) standar komunikasi serial SPI.
- Keterampilan membuat program pengiriman data serial SPI 8-bit antara Arduino dengan *wizNET W5100 Ethernet Controller*.

#### 7.3. Uraian Materi

Serial SPI (*Serial Peripheral Interface*) merupakan mode komunikasi serial yang banyak digunakan pada teknologi antarmuka antar perangkat hingga saat ini. Jenis komunikasi ini memiliki sistem deret dan terpisah antara jalur pengiriman (dari *Master* ke *Slave* dan sebaliknya) dengan sebuah sistem denyut (*clock*) pada jalur tersendiri dan juga adanya 1 jalur yang digunakan untuk mengaktifkan *slave* mana yang akan dikirimkan data. SPI adalah bagian perangkat keras pada Arduino yang menerjemahkan bit-bit serial. SPI biasanya berupa sirkuit terintegrasi yang digunakan untuk komunikasi serial perangkat periperal. Transfer data dengan SPI digunakan antara *Master* (Arduino) dgn *slave* (Arduino lainnya atau beberapa peripheral SPI seperti MMC card, SPI ADC, dll.) dalam jarak dekat dan kecepatan cukup tinggi.

Komunikasi serial data antara master dan slave pada SPI diatur melalui 4 buah pin yang terdiri dari SCLK, MOSI, MISO, dan SS sebagai berikut:

- MOSI jalur data dari master dan masuk ke dalam slave
- MISO jalur data keluar dari slave dan masuk ke dalam master
- SLK dari master ke slave yang berfungsi sebagai clock
- SS (*slave select*) merupakan pin yang berfungsi untuk mengaktifkan slave

Pada *platform* Arduino UNO terdapat fitur SPI yang dapat digunakan sebagai media komunikasi serial 8-bit antar devais yang diletakkan pada konektor ICSP. Pada Gambar 7.1 diberikan contoh pengiriman karakter melalui modul ethernet. Kombinasi penekanan *push button* pada modul akan mengirimkan pembacaan karakter yang berbeda-beda pada klien.

```

#include <Ethernet.h>

EthernetClient client;
const int PINsw1 = 2;
const int PINsw2 = 3;

void setup() {
  pinMode(PINsw1,INPUT_PULLUP);
  pinMode(PINsw2,INPUT_PULLUP);
  Serial.begin(9600);
  uint8_t mac[6] = {0x00,0x01,0x02,0x03,0x04,0x05};
  Ethernet.begin(mac);
}

char tx_data;
void loop() {
  int sw1 = digitalRead(PINsw1);
  int sw2 = digitalRead(PINsw2);
  if(client.connect(IPAddress(192,168,0,6),80))
  {
    if ( sw1 == 0 )client.print("A");
    if ( sw2 == 0 )client.print("B");
    if ( sw1 == 1 )client.print("C");
    if ( sw2 == 1 )client.print("D");
    client.stop();
  }
  delay(1000);
}

```

Gambar 7.1 Potongan Program Mengirim Karakter Via Modul Ethernet

#### 7.4. Peralatan Praktikum

- A. Seperangkat komputer
- B. Koneksi internet
- C. Perangkat lunak Arduino IDE
- D. Arduino UNO
- E. Modul Komunikasi SPI
- F. Modul Ethernet

#### 7.5. Skema Praktikum

Mahasiswa menyambungkan kabel jumper pada sambungan pin SPI Arduino UNO. Memastikan sambungan sesuai panduan yaitu dengan sambungan langsung tanpa persilangan antara pin MOSI Arduino dengan MOSI devais SPI *wizNET W5100 Ethernet Controller*, begitu pula pada sambungan MISO dan SCK. Mahasiswa memprogram bahasa C/C++ pada Arduino IDE dan mencari kesalahan pada *syntax program*.

#### 7.6. Langkah Praktikum

1. Pelajari materi mengenai serial SPI dari beberapa pustaka yang ada.
2. Pastikan telah mengunduh *library* untuk etherne (ethernet.h)
3. Menyambungkan kabel penghubung pada pin serial SPI Arduino UNO untuk pin

MOSI, MISO, dan SCK.

4. Ikuti langkah – langkah pemrograman serial SPI yang akan dipandu pada saat praktikum.
5. Mengkompilasi program dan melihat susunan *syntax program* apakah ada kesalahan atau tidak.
6. Mengunggah file komunikasi serial SPI ke Arduino UNO.
7. Lakukan pengiriman dan penerimaan teks melalui modul ethernet dan serila monitor pada PC.

### **7.7. Pertanyaan**

1. Lakukan deskripsi singkat mengenai Komunikasi Serial SPI.
2. Lakukan deskripsi singkat menurut anda mengenai teknik komunikasi serial SPI pada Arduino UNO.
3. Lakukan deskripsi singkat mengenai komunikasi menggunakan modul ethernet dan berikan analisa anda.

## MODUL 8

### AKUISISI DATA SENSOR SUHU LM35 OLEH PC

#### 8.1 Kompetensi Dasar

Setelah mengikuti praktikum ini, mahasiswa mampu:

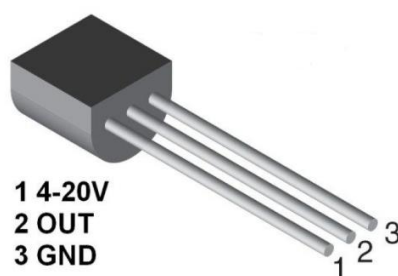
- Membuat program untuk membaca sensor suhu LM35 dan menampilkannya ke dalam GUI yang dibuat menggunakan Delphi 7.
- Memahami proses komunikasi serial antara Arduino UNO dan Komputer.

#### 8.2 Indikator

- Keterampilan membuat pengaturan dan instalasi sistem aplikasi program Arduino IDE.
- Keterampilan mengoperasikan program aplikasi Arduino IDE dan Delphi 7.
- Keterampilan mengoperasikan perangkat Arduino UNO dan Sensor LM35

#### 8.3 Uraian Materi

Akuisisi data merupakan salah satu bagian proses kontrol dalam kegiatan perindustrian. Dalam akuisisi data terdapat proses pengukuran besaran fisis yang menjadi ujung tombak dalam suatu sistem. Pada praktikum kali ini, kita menggunakan sensor suhu LM 35, dimana sensor ini dapat mengubah besaran suhu menjadi besaran listrik dalam bentuk tegangan. LM 35 merupakan sensor suhu yang teliti dan terkemas dalam bentuk *Integrated Circuit* (IC), dimana output tegangan keluaran sangat linear terhadap perubahan suhu. Sensor ini memiliki koefisien sebesar  $10 \text{ mV } / ^\circ\text{C}$  yang berarti bahwa kenaikan suhu  $1^\circ \text{C}$  maka akan terjadi kenaikan tegangan sebesar  $10 \text{ mV}$ . Fisik sensor LM 35 dapat diamati pada gambar 8.1 di bawah ini.



Gambar 8.1 Sensor Suhu LM 35

Pada suatu sistem kontrol, suhu merupakan salah satu besaran fisis yang sering dipakai untuk mendapatkan informasi, baik untuk sistem monitoring saja atau untuk proses pengendalian lebih lanjut. Dalam kegiatan monitoring, salah satu hal penting adalah bagaimana cara menyajikan data agar dapat dibaca atau dimengerti oleh pengguna. Cara yang efektif untuk menyajikan data bisa menggunakan *Graphic User Interface* (GUI).



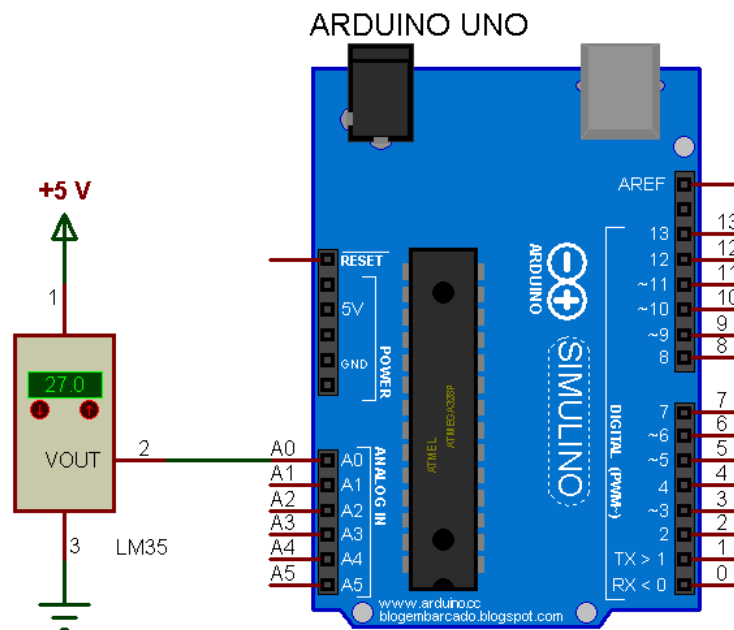
GUI merupakan jenis antarmuka pengguna yang menggunakan metode interaksi pada peranti elektronik secara grafis antara pengguna dan komputer. Dalam praktikum kali ini, GUI yang akan dibangun menggunakan basis pemrograman bahasa Pascal dengan perangkat lunak Delphi 7. Hasil pengukuran atau akuisisi data sensor LM35 akan diolah oleh Arduino UNO, kemudian hasilnya akan dikirimkan melalui komunikasi data serial dan ditampilkan dalam bentuk grafis di aplikasi visual yang dibangun.

### 8.4 Peralatan Praktikum

- A. Seperangkat komputer
- B. Trainer Sistem Antarmuka Komputer
- C. Perangkat lunak Arduino IDE
- D. Perangkat lunak Delphi 7

### 8.5 Langkah Praktikum

1. Perhatikan Gambar 8.2 di bawah ini, hubungkan sensor LM35 ke pin A0 Arduino UNO menggunakan kabel *jumper*.



Gambar 8.2 Rangkaian Arduino dan LM 35

2. Bukalah aplikasi Arduino IDE dan aturlah *Board* dan *Port* Arduino yang digunakan sesuai dengan *Board* yang ada di *trainer* dan *Port* yang terdeteksi di komputer.
3. Buatlah program Arduino seperti di bawah ini:

```

float temp = A0;
float in;

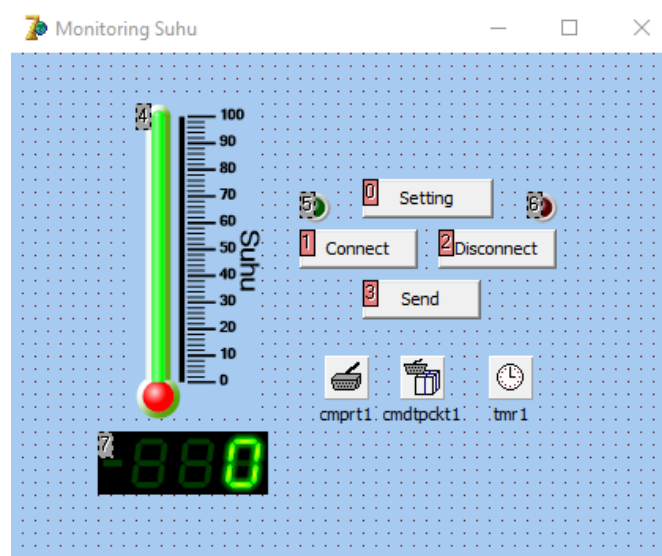
void setup()
{
  Serial.begin(9600);
  pinMode(A0, INPUT);
}

void loop()
{
  if (Serial.available() > 0) {
    int dataIn = Serial.read();
    if(dataIn == '*'){
      suhu();
    }
  }
}

void suhu()
{
  in = analogRead(A0);
  in = in * 0.48828125;
  Serial.print('*');
  Serial.print(in);
  Serial.println('#');
}

```

4. Klik tombol “Verify” untuk mengecek apakah ada kesalahan dalam membuat program.
5. Apabila tidak ada kesalahan tekan tombol “upload” dan tunggu hingga ada keterangan “Done Uploading”.
6. Bukalah terminal serial monitor pada Arduino IDE, ketikkan karakter “\*”, tekan enter. Perhatikan apakah ada nilai yang keluar.
7. Jika nilai suhu sudah keluar maka anda berhasil mengakuisisi data sensor suhu LM 35.
8. Buka perangkat lunak Delphi 7 dan ambillah beberapa komponen yang dibutuhkan seperti Gambar 8.3 di bawah ini.



Gambar 8.3 GUI *Monitoring Suhu*

9. Buatlah program seperti berikut pada *software* Delphi 7 untuk masing-masing komponen yang diambil:

- Variable:

```
data: Real;  
awal:integer;  
akhir:integer;  
temp:String;
```

- Events ComDataPacket:

```
awal:=Pos('*', Str);  
akhir:=Pos('#', Str);  
  
temp:=MidStr(Str, awal+1, 3);  
  
try  
    data:=StrToFloat(temp);  
except  
    on E: EConvertError do  
        begin  
            end;  
        end;  
end;  
  
Form1.alatSuhu.Value:=data;  
Form1.segSuhu1.Value:=data;
```

- Events Setting:

```
Form1.cmpprt1.ShowSetupDialog;
```

- Events Connect:

```
cmpprt1.Open;  
Form1.tmr1.Enabled:=False;  
Form1.led_con1.Value:=True;  
Form1.led_con2.Value:=False;
```

- Events Disconnect

```
cmpprt1.Close;  
Form1.led_con1.Value:=False;  
Form1.led_con2.Value:=True;
```

- Events Send

```
Form1.tmr1.Enabled:=True;  
cmpprt1.WriteString('*');
```

- Events Timer

```
if cmpprt1.Connected then  
begin  
    cmpprt1.WriteString('*');  
end;
```

10. Simpan pekerjaan anda dengan cara *File* → *Save all*.

11. Tekan *Run (F9)* untuk menjalankan aplikasi yang anda buat. Apabila ada pesan *error*, silahkan cek kembali program anda.

## 8.6 Pertanyaan

1. Jelaskan secara singkat cara akuisisi data sensor suhu LM35.
2. Menurut anda, bagaimana teknik pengujian dan *troubleshooting* akuisisi data sensor suhu LM35 dengan GUI yang anda buat?
3. Diskripsikan secara singkat mengenai komunikasi data antara *trainer* sistem antarmuka dengan GUI yang anda buat, khususnya pembacaan data oleh komputer dari *trainer*.

## MODUL 9

### KENDALI MOTOR DC DENGAN TEKNIK PWM OLEH PC

#### 9.1 Kompetensi Dasar

Setelah mengikuti praktikum ini, mahasiswa mampu:

- Membuat program untuk mengendalikan motor DC melalui GUI yang dibuat menggunakan Delphi 7.
- Memahami proses komunikasi serial antara Arduino UNO dan Komputer.

#### 9.2 Indikator

- Keterampilan membuat pengaturan dan instalasi sistem aplikasi program Arduino IDE.
- Keterampilan mengoperasikan program aplikasi Arduino IDE dan Delphi 7.
- Keterampilan mengoperasikan perangkat Arduino UNO dan motor DC

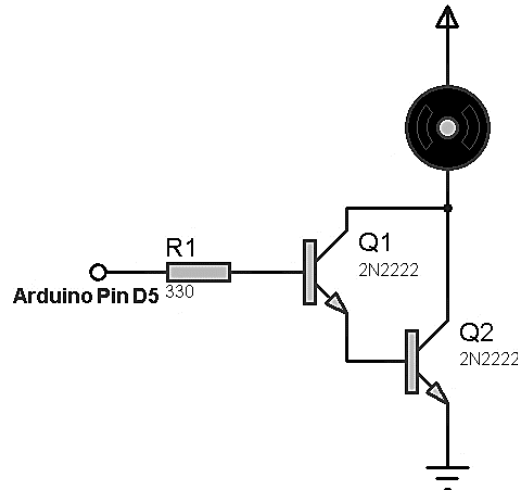
#### 9.3 Uraian Materi

Teknik PWM (*Pulse Width Modulation*) merupakan salah satu metode untuk mengendalikan motor DC. Teknik PWM bekerja dengan cara mengatur gelombang persegi yang digunakan. Gelombang persegi digunakan untuk mensuplai motor DC dan memiliki frekuensi tetap (biasanya max 10 KHz) namun lebar pulsa *high* dan *low* dalam 1 periode akan diatur dengan skala 0 sampai 100%, hal ini sering disebut dengan siklus kerja (*duty cycle*). Perbandingan pulsa *high* terhadap *low* ini akan menentukan jumlah daya yang diberikan ke motor DC. Rumus untuk menghitung siklus kerja ditunjukkan dalam persamaan berikut:

$$\text{Siklus Kerja} = \text{Waktu}_{ON} / (\text{Waktu}_{ON} + \text{Waktu}_{OFF})$$

dimana, Waktu ON adalah waktu dimana tegangan keluaran berada pada posisi tinggi (*high* atau 1). Sedangkan Waktu OFF adalah waktu dimana tegangan keluaran berada pada posisi rendah (*low* atau 0). Kemudian penjumlahan antara Waktu ON dan Waktu OFF merupakan waktu siklus atau periode satu gelombang.

*Driver motor* merupakan sebuah piranti yang bertugas untuk menjalankan motor baik mengatur arah putaran motor maupun kecepatan putar motor. Untuk menjalankan motor DC dengan PWM tidak dapat menggunakan relai, melainkan harus menggunakan rangkaian "*driver motor DC*" elektronik yang lainnya. Rangkaian *driver motor DC* dengan transistor di Gambar 9.1 adalah contoh sederhana. Rangkaian tersebut memiliki komponen utama transistor yang disusun secara Darlington. Transistor Darlington ini dapat berupa dua buah transistor yang terhubung secara Individu ataupun satu perangkat tunggal yang dibuat secara komersial dalam satu kemasan paket dengan standar tiga kaki yaitu Basis, Emitor dan Kolektor yang diintegrasikan dalam satu *chip* atau komponen. Transistor yang dipakai dapat berupa transistor jenis NPN tipe misal 2N222.



Gambar 9.1 Driver Motor DC

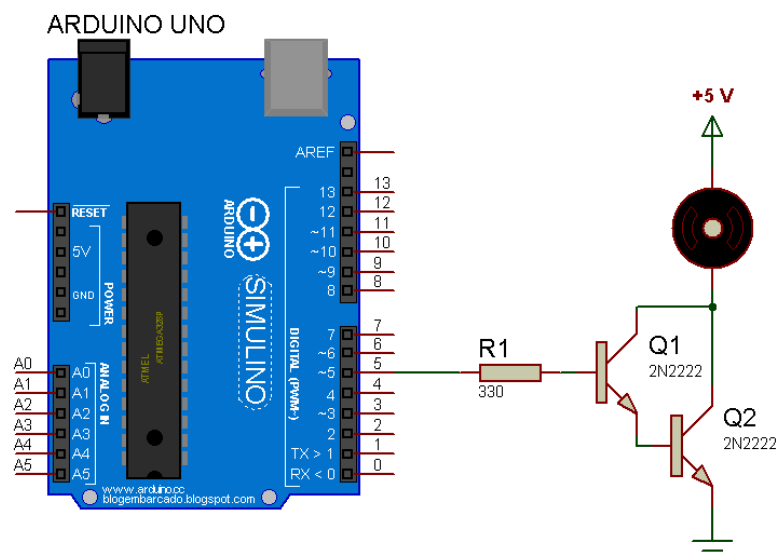
Dengan menghubungkan 2 buah transistor secara paralel maka besarnya kemampuan transistor mengalirkan arus akan menjadi 2 kali lebih besar. Kelebihan dari Transistor Darlington ini antarlain *Gain* arus dari transistor ini tinggi, impedansi input dari rangkaian ini tinggi, dan konfigurasi rangkaian yang mudah.

#### 9.4 Peralatan Praktikum

- A. Seperangkat komputer
- B. Trainer Sistem Antarmuka Komputer
- C. Perangkat lunak Arduino IDE
- D. Perangkat lunak Delphi 7

#### 9.5 Langkah Praktikum

1. Perhatikan Gambar 9.2 di bawah ini. Rangkaian Arduino dan Motor DC di dalam trainer terhubung seperti berikut:



Gambar 9.2 Rangkaian Arduino dengan Motor DC

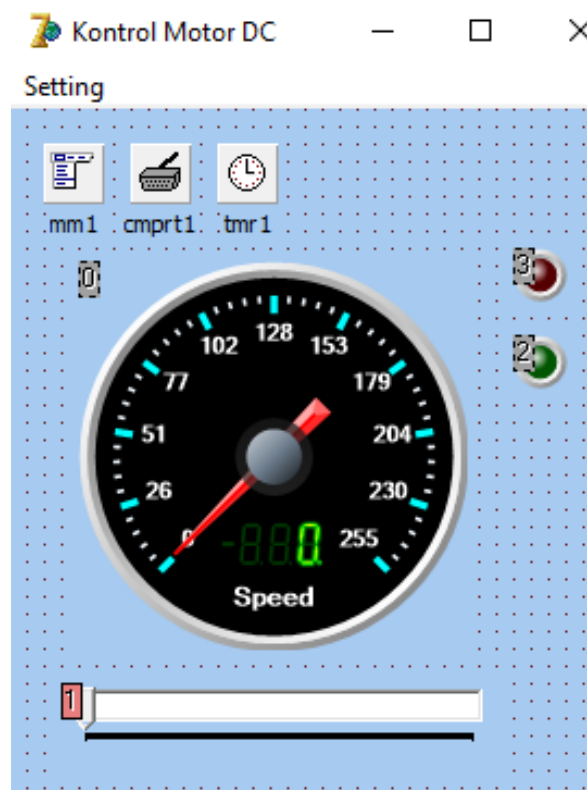
2. Buka aplikasi Arduino IDE dan aturlah *Board* dan *Port* Arduino yang digunakan sesuai dengan *Board* yang ada di *trainer* dan *Port* yang terdeteksi di komputer.
3. Buatlah program Arduino seperti di bawah ini:

```
const int motorPin = 5;
int Speed;

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

void loop()
{
  if (Serial.available() > 0) {
    Speed = Serial.parseInt();
    int dataIn = Serial.read();
    if(dataIn == '*'){
      analogWrite(motorPin, Speed);
    }
  }
}
```

4. Klik tombol “*Verify*” untuk mengecek apakah ada kesalahan dalam membuat program. Apabila tidak ada kesalahan tekan tombol “*upload*” dan tunggu hingga ada keterangan “*Done Uploading*”.
5. Buka perangkat lunak Delphi 7 dan ambil beberapa komponen yang dibutuhkan seperti Gambar 9.3 di bawah ini:



Gambar 9.3 GUI Kendali Motor DC

6. Buatlah program seperti berikut pada software Delphi 7 untuk masing-masing komponen yang diambil:

- Events ComDataPacket:

```
awal:=Pos('*', Str);
akhir:=Pos('#', Str);

temp:=MidStr(Str, awal+1, 3);

try
  data:=StrToFloat(temp);
except
  on E:EConvertError do
  begin
    end;
end;

Form1.alatSuhu.Value:=data;
Form1.segSuhu1.Value:=data;
```

- Events Com Port:

```
Form1.cmprt1.ShowSetupDialog;
```

- Events Connect:

```
cmprt1.Open;
Form1.tmr1.Enabled:=True;
led_con1.Value:=True;
led_con2.Value:=False;
```

- Events Disconnect

```
cmprt1.Close;
Form1.led_con1.Value:=False;
Form1.led_con2.Value:=True;
Form1.trckbr1.Position:=0;
Form1.motor_speed.Value:=0;
```

- Events Timer

```
if cmprt1.Connected then
begin
  cmprt1.WriteString('*'+IntToStr(Form1.trckbr1.Position));
  form1.motor_speed.Value:=Form1.trckbr1.Position;
end;
```

7. Simpan pekerjaan anda dengan cara *File* → *Save all*.
8. Tekan *Run (F9)* untuk menjalankan aplikasi yang anda buat. Apabila ada pesan *error*, silahkan cek kembali program anda.



## 9.6 Pertanyaan

1. Jelaskan prosedur penggunaan teknik PWM untuk pengendalian motor DC menggunakan Arduino dan PC.
2. Menurut anda bagaimana cara pengujian dan *troubleshooting* pengendalian PWM motor DC dengan GUI?
3. Jelaskan secara singkat tentang komunikasi data antara *trainer* sistem antarmuka dengan GUI yang anda buat, khususnya untuk pengiriman data dari laptop ke trainer.

## MODUL 10

### KENDALI MOTOR SERVO DENGAN TEKNIK PWM OLEH PC

#### 10.1 Kompetensi Dasar

Setelah mengikuti praktikum ini, mahasiswa mampu:

- Membuat program untuk mengendalikan motor Servo melalui GUI yang dibuat menggunakan Delphi 7.
- Memahami proses komunikasi serial antara Arduino UNO dan Komputer.

#### 10.2 Indikator

- Keterampilan membuat pengaturan dan instalasi sistem aplikasi program Arduino IDE.
- Keterampilan mengoperasikan program aplikasi Arduino IDE dan Delphi 7.
- Keterampilan mengoperasikan perangkat Arduino UNO dan motor Servo

#### 10.3 Uraian Materi

Motor servo adalah sebuah motor DC yang dilengkapi rangkaian kendali dengan sistem *closed feedback*. Motor servo mampu bekerja dua arah (searah jarum jam dan berlawanan arah jarum jam) dimana arah dan sudut pergerakan rotornya dapat dikendalikan dengan memberikan variasi lebar pulsa (*duty cycle*) sinyal PWM pada bagian pin kontrolnya. Motor servo disusun dari sebuah motor DC, *gearbox*, *variabel resistor* (VR) atau potensiometer dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas maksimum putaran sumbu (axis) motor servo. Sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang pada pin kontrol motor servo.

Motor servo dikendalikan oleh sebuah pulsa selebar  $\pm 20$  ms, dimana lebar pulsa antara 0.5 ms dan 2 ms menyatakan akhir dari range sudut maksimum. Apabila motor servo diberikan pulsa dengan besar 1.5 ms mencapai gerakan  $90^\circ$ , maka bila kita berikan pulsa kurang dari 1.5 ms maka posisi mendekati  $0^\circ$  dan bila kita berikan pulsa lebih dari 1.5 ms maka posisi mendekati  $180^\circ$ .

Jenis motor servo yang digunakan dalam praktikum ini adalah motor servo standard  $180^\circ$ . Motor servo jenis ini hanya mampu bergerak dua arah CW (*Clock Wise*) dan CCW (*Counter Clock Wise*) dengan defleksi masing-masing sudut mencapai  $90^\circ$  sehingga total defleksi sudut dari kanan – tengah – kiri adalah  $180^\circ$ . Motor Servo yang digunakan dalam praktikum ini tergambar pada Gambar 10.1 di bawah ini.



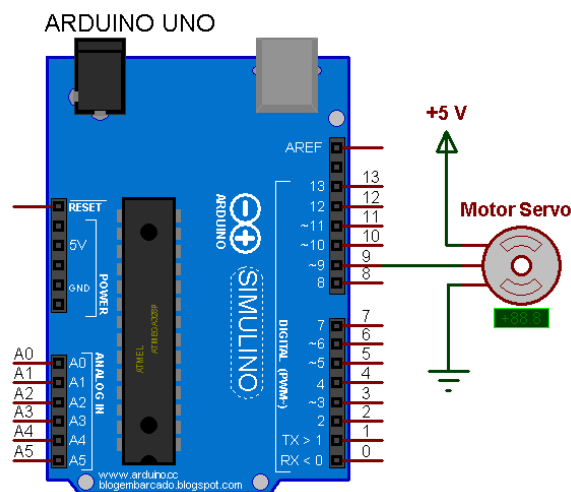
Gambar 10.1 Motor Servo

## 10.4 Peralatan Praktikum

- A. Seperangkat komputer
- B. Trainer Sistem Antarmuka Komputer
- C. Perangkat lunak Arduino IDE
- D. Perangkat lunak Delphi 7

## 10.5 Langkah Praktikum

1. Perhatikan Gambar 10.2 di bawah ini. Rangkaian Arduino dan Motor Servo di dalam trainer terhubung seperti berikut:



Gambar 10.2 Rangkaian Arduino dan Motor Servo

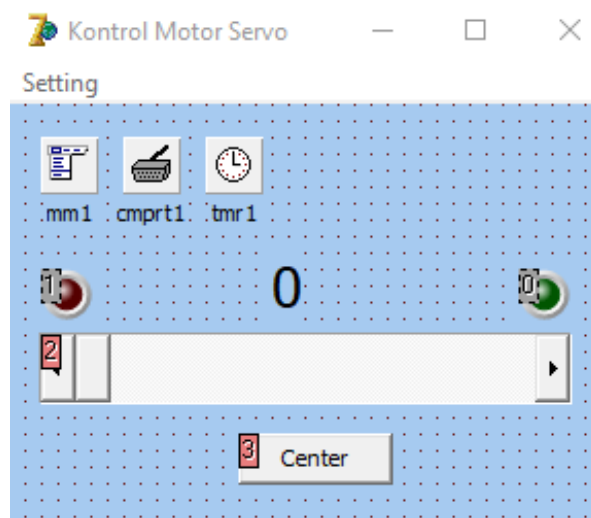
2. Buka aplikasi Arduino IDE dan aturlah *Board* dan *Port* Arduino yang digunakan sesuai dengan *Board* yang ada di *trainer* dan *Port* yang terdeteksi di komputer.
3. Buatlah program Arduino seperti di bawah ini:

```
#include <Servo.h>
Servo myservo;
int sudut;

void setup()
{
  Serial.begin(9600);
  myservo.attach(9);
  myservo.write(0);
}

void loop()
{
  if (Serial.available() > 0) {
    sudut = Serial.parseInt();
    int dataIn = Serial.read();
    if(dataIn == '*'){
      myservo.write(sudut);
      delay(15);
    }
  }
}
```

4. Klik tombol “Verify” untuk mengecek apakah ada kesalahan dalam membuat program. Apabila tidak ada kesalahan tekan tombol “upload” dan tunggu hingga ada keterangan “Done Uploading”.
5. Buka perangkat lunak Delphi 7 dan ambillah beberapa komponen yang dibutuhkan seperti Gambar 10.3 di bawah ini:



Gambar 10.3 GUI Kendali Motor Servo

6. Buatlah program seperti berikut pada software Delphi 7 untuk masing-masing komponen yang diambil:

- Events ComDataPacket:

```

awal:=Pos('*', Str);
akhir:=Pos('#', Str);

temp:=MidStr(Str, awal+1, 3);

try
  data:=StrToFloat(temp);
except
  on E:EConvertError do
  begin
    end;
  end;
end;

Form1.alatSuhu.Value:=data;
Form1.segSuhu1.Value:=data;

```

- Events Com Port:

```
Form1.cmprt1.ShowSetupDialog;
```

- Events Center:

```

cmprt1.WriteString('*'+'90');
Form1.ScrollBar1.Position:=90

```

- Events Connect:

```
cmpprt1.Open;
Form1.tmr1.Enabled:=True;
led_con1.Value:=True;
led_con2.Value:=False;
```

- Events Disconnect

```
cmpprt1.Close;
Form1.led_con1.Value:=False;
Form1.led_con2.Value:=True;
Form1.trckbr1.Position:=0;
Form1.motor_speed.Value:=0;
```

- Events Timer

```
if cmpprt1.Connected then
begin
    cmpprt1.WriteString('*'+IntToStr(Form1.ScrollBar1.Position));
    Form1.lbl_servo.Caption:=IntToStr(Form1.ScrollBar1.Position);
end;
```

7. Simpan pekerjaan anda dengan cara *File* → *Save all*.
8. Tekan *Run (F9)* untuk menjalankan aplikasi yang anda buat. Apabila ada pesan *error*, silahkan cek kembali program anda.

## 10.6 Pertanyaan

1. Menurut anda, bagaimana teknik PWM untuk pengendalian motor Servo dengan Arduino dan PC?
2. Jelaskan cara pengujian dan *troubleshooting* pengendalian PWM motor Servo dengan GUI yang anda buat.
3. Diskripsikan tentang cara komunikasi data antara *trainer* sistem antarmuka dengan GUI yang anda buat, khususnya untuk pengiriman data dari laptop ke trainer.

**@2019**

**Diterbitkan oleh:**

**Universitas Teknologi Yogyakarta**

**Jl. Siliwangi, Jombor, Sleman, Yogyakarta**

**Email : [publikasi@uty.ac.id](mailto:publikasi@uty.ac.id)**

**Website : [uty.ac.id](http://uty.ac.id)**