

**LAPORAN PRAKTIK ANTARMUKA  
KENDALI MOTOR STEPPER MENGGUNAKAN LPT1**



Disusun oleh:

NAMA : WRUHANTOJATI

NIM : 14518244013

PROGRAM STUDI PENDIDIKAN TEKNIK MEKATRONIKA

PENDIDIKAN TEKNIK ELEKTRO

FAKULTAS TEKNIK

UNIVERSITAS NEGERI YOGYAKARTA

2016

## I. Tujuan Praktikum

Adapun tujuan praktikum yang dapat diperoleh setelah melakukan praktik yaitu sebagai berikut:

1. Memahami kegunaan Borland Delphi 7 untuk interface kendali motor stepper.
2. Memahami tentang konsep transmisi data paralel.
3. Memahami cara kerja motor stepper.
4. Memahami cara kerja teknik transmisi data paralel lewat port LPT1.
5. Memanfaatkan transmisi data paralel untuk kendali motor stepper.

## II. Kajian Teori

### 1. Delphi 7



Delphi adalah suatu bahasa pemrograman (*development language*) yang digunakan untuk merancang suatu aplikasi program. Delphi termasuk dalam pemrograman bahasa tingkat tinggi (*high level language*). Maksud dari bahasa tingkat tinggi yaitu perintah-perintah programnya menggunakan bahasa yang mudah dipahami oleh manusia. Bahasa pemrograman Delphi disebut bahasa prosedural artinya mengikuti urutan tertentu. Dalam membuat aplikasi perintah-perintah, Delphi menggunakan lingkungan pemrograman visual.

Delphi merupakan generasi penerus dari Turbo Pascal. Pemrograman Delphi dirancang untuk beroperasi dibawah sistem operasi Windows. Program ini

mempunyai beberapa keunggulan, yaitu produktivitas, kualitas, pengembangan perangkat lunak, kecepatan kompilasi, pola desain yang menarik serta diperkuat dengan bahasa pemrograman yang terstruktur dalam struktur bahasa pemrograman

*Object Pascal.*

Sebagian besar pengembang Delphi menuliskan dan mengompilasi kode program di dalam lingkungan pengembang aplikasi atau *Integrated Development Environment* (IDE). Lingkungan kerja IDE ini menyediakan sarana yang diperlukan untuk merancang, membangun, mencoba, mencari atau melacak kesalahan, serta mendistribusikan aplikasi. Sarana-sarana inilah yang memungkinkan pembuatan prototipe aplikasi menjadi lebih mudah dan waktu yang diperlukan untuk mengembangkan aplikasi menjadi lebih singkat.

## 2. Port paralel LPT1/DB25



Port Paralel merupakan pengembangan dari teknologi komputer yang digunakan soket pada personal komputer untuk berkomunikasi dengan peralatan luar seperti printer model lama. Untuk itu, port paralel saat ini sangat berperan penting dalam teknologi modern. Arti istilah paralel yaitu sistem pengiriman data digital, dimana beberapa bit data dikirim sekaligus pada satu saat dengan menggunakan jalur terpisah. Jadi port paralel adalah salah satu jenis soket pada personal komputer untuk berkomunikasi dengan peralatan luar untuk mengirim data digital seperti printer model lama. Karena itu port paralel sering juga disebut printer port. Perusahaan yang memperkenalkan port ini adalah Centronic, maka port ini juga disebut dengan Centronics port. Kesederhanaan port ini dari sisi pemrograman dan antarmuka dengan hardware membuat port ini sering digunakan untuk percobaan -percobaan sederhana dalam perancangan peralatan elektronika.

LPT (*Line Print Terminal*) adalah istilah untuk *port* yang biasanya dihubungkan ke printer. Istilah lain yang sering digunakan untuk menyebut LPT Port adalah Parallel Port. Komunikasi yang digunakan oleh LPT adalah komunikasi paralel yang berjalan lebih cepat dari komunikasi serial karena

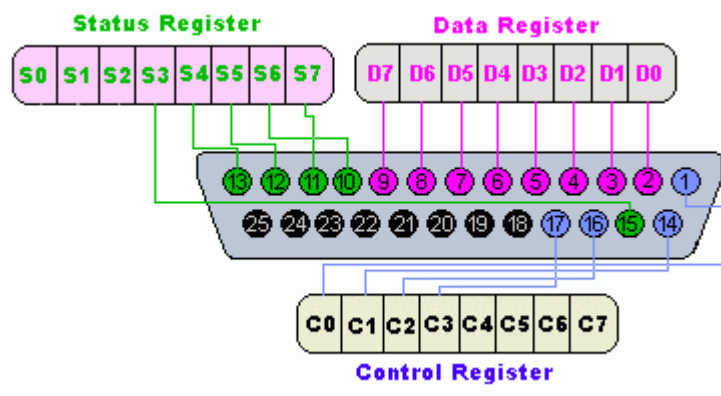
jumlah bit yang dikirimkan bersamaan dalam suatu waktu oleh komunikasi paralel lebih banyak daripada komunikasi serial.

Jumlah kawat yang digunakan oleh komunikasi paralel sebanyak 25 pin yang lebih banyak daripada komunikasi Serial / COM Port yang hanya 9 pin. Meskipun jumlah pin port COM pada komputer umumnya berjumlah sama dengan jumlah pin port LPT, tidak semua pin dalam port COM digunakan.

Port Paralel banyak digunakan dalam berbagai macam aplikasi *Interface*. Port ini membolehkan kita memiliki masukan hingga 8 bit atau keluaran hingga 12 bit pada saat yang bersamaan dengan hanya membutuhkan sedikit rangkaian eksternal sederhana untuk melakukan suatu tugas tertentu.

Port Paralel ini terdiri dari :

- a. Jalur Kontrol dengan alamat \$37A
- b. 5 Jalur Status dengan alamat \$379
- c. 8 Jalur Data. dengan alamat \$378



Berikut daftar pin dari Port Paralel DB-25 (PS = Printer Status, PC = Printer Control) yakni sebagai berikut:

Pin No (DB25)	Signal name	Direction	Register - bit	Inverted
1	nStrobe	Out	Control-0	Yes
2	Data0	In/Out	Data-0	No
3	Data1	In/Out	Data-1	No

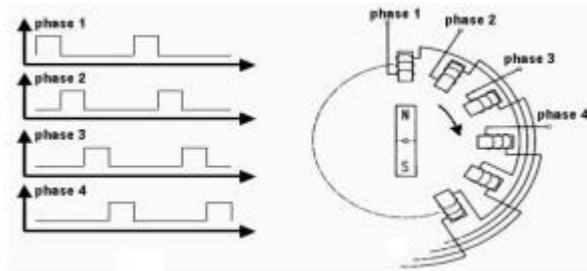
3. Motor	4	Data2	In/Out	Data-2	No
	5	Data3	In/Out	Data-3	No
	6	Data4	In/Out	Data-4	No
	7	Data5	In/Out	Data-5	No
	8	Data6	In/Out	Data-6	No
	9	Data7	In/Out	Data-7	No
	10	nAck	In	Status-6	No
	11	Busy	In	Status-7	Yes
	12	Paper-Out	In	Status-5	No
	13	Select	In	Status-4	No
	14	Linefeed	Out	Control-1	Yes
	15	nError	In	Status-3	No
	16	nInitialize	Out	Control-2	No
	17	nSelect-Printer	Out	Control-3	Yes
	18-25	Ground	-	-	-



Motor stepper adalah suatu motor listrik yang dapat mengubah pulsa listrik yang diberikan menjadi gerakan motor *discret* (terputus) yang disebut *step* (langkah). Satu putaran motor memerlukan  $360^\circ$  dengan jumlah langkah yang tertentu per derajatnya. Ukuran kerja dari motor stepper biasanya diberikan dalam jumlah langkah per-putaran per-detik.

Prinsip kerja motor stepper adalah mengubah pulsa-pulsa input menjadi gerakan mekanis diskrit. Oleh karena itu untuk menggerakkan motor stepper diperlukan pengendali motor stepper yang membangkitkan pulsa-pulsa periodik.

Berikut ini adalah ilustrasi struktur motor stepper sederhana dan pulsa yang dibutuhkan untuk menggerakannya :



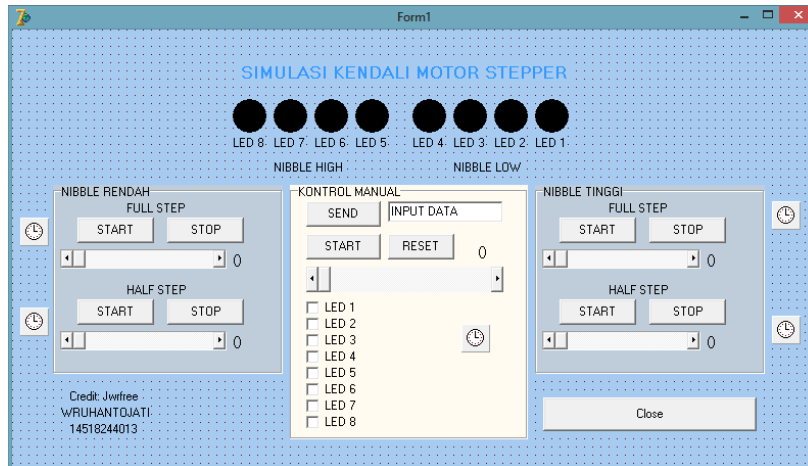
Gambar di atas memberikan ilustrasi dari pulsa keluaran pengendali motor stepper dan penerapan pulsa tersebut pada motor stepper untuk menghasilkan arah putaran yang bersesuaian dengan pulsa kendali. Berdasarkan struktur rotor dan stator motor stepper, maka motor stepper dapat dikategorikan dalam 3 jenis yaitu motor stepper Variable Reluctance (VR), motor stepper Permanent Magnet (PM), motor stepper Hybrid (HB). Berdasarkan metode perancangan rangkaian pengendalinya, motor stepper dapat dibagi menjadi 2 jenis yaitu motor stepper unipolar dan motor stepper bipolar.

### III. Alat dan bahan

1. Modul Praktik motor stepper
2. Komputer dengan Sistem Operasi Windows XP
3. Software Borland Delphi 7
4. Kabel interface LPT1/BD25

### IV. Langkah Praktikum

1. Ikuti petunjuk modul yang ada maupun prosedur dari dosen
2. Menyiapkan alat dan bahan yang diperlukan
3. Jalankan program Borland Delphi 7, dalam form yang aktif buatlah desain tampilan form seperti tampilan di bawah ini:



4. Untuk membuat tampilan kendali motor stepper seperti di atas dibutuhkan beberapa tools yaitu label, shape, edit, button, ScrollBar, Timer, GroupBox, dan CheckBox. Untuk mengatur pilihan warna dapat diatur sesuai selera dan tampilan layout dari form tersebut dapat diatur sesuai yang dikehendaki.
5. Pastikan untuk nama komponen Shape dari LED agar berurutan dari kanan ke kiri yaitu Shape1, Shape2,....,Shape8.
6. Setelah selesai membuat form, langkah selanjutnya adalah membuat program dengan klik tombol Toggle Form/Unit atau tekan tombol F12
7. Ketikkan program di bawah ini pada bagian var untuk mendeklarasikan global variable bertipe integer. Di sini untuk menghidupkan stepper ada yang menggunakan cara biasa atau perkalian, dan juga menggunakan array.

```

var
  Form1: TForm1;
  data : integer;
  nilai : integer;
  rendah : integer;
  half : integer;
  step: integer;
  a : array [1..8] of integer=(16,48,32,96,64,192,128,144); //nibble high half step
  b : array [1..4] of integer=(16,32,64,128); //nibble high, full step
  c : array [1..8] of integer=(1,3,2,6,4,12,8,9); //nibble low, half step

```

8. Ketikkan cuplikan program di bawah ini untuk deklarasi LED yaitu sebagai berikut:

```

procedure TForm1.NYALAKAN8LED;
Begin
  if (data AND 1)=1 then shape1.brush.color := clRED
  else Shape1.Brush.color := clBlack ;
  if (data AND 2)=2 then shape2.brush.color := clRED
  else Shape2.Brush.color := clBlack ;
  if (data AND 4)=4 then shape3.brush.color := clRED
  else Shape3.Brush.color := clBlack ;
  if (data AND 8)=8 then shape4.brush.color := clRED
  else Shape4.Brush.color := clBlack ;
  if (data AND 16)=16 then shape5.brush.color := clRED
  else Shape5.Brush.color := clBlack ;
  if (data AND 32)=32 then shape6.brush.color := clRED
  else Shape6.Brush.color := clBlack ;
  if (data AND 64)=64 then shape7.brush.color := clRED
  else Shape7.Brush.color := clBlack ;
  if (data AND 128)=128 then shape8.brush.color := clRED
  else Shape8.Brush.color := clBlack ;
End;

```

9. Lalu pada bagian private ketikkan deklarasi prosedur NYALAKAN8LED di atas

```

private
  { Private declarations }
  procedure NYALAKAN8LED;
public
  { Public declarations }
end;

```

10. Masukkan prosedur untuk mengeluarkan data pada implementation:

```

implementation

{$R *.dfm}
procedure outportb(portid : integer; value : BYTE) ;
Begin
  asm
    mov edx, portid
    mov al, value
    out dx,al
  end;
end;

```

Dan prosedur untuk memasukkan data:



```

function inportb (portid : integer) : byte;
Var value : byte;

Begin
  asm
    mov edx, portid
    in al, dx
    mov value, al
  end;
  inportb := value;
end;

```

11. Masukkan program untuk menjalankan **motor stepper nibble rendah full step** pada prosedur timer2:

```

procedure TForm1.Timer2Timer(Sender: TObject); //TIMER2
begin
  data := data *2;
  if(data>8) then
    data := 1;
  NYALAKAN8LED;
end;

```

Dan tombol untuk start:

```

procedure TForm1.Button4Click(Sender: TObject); //START2
begin
  data := 1;
  timer2.Enabled := true;
  NYALAKAN8LED;
end;

```

Serta tombol stop:

```

procedure TForm1.Button5Click(Sender: TObject); //STOP2
begin
  timer2.enabled := false;
  edit1.Text := '0';
  data := strtoint(edit1.Text);
  NYALAKAN8LED;
end;

```

Pengaturan ScrollBar dan label interval:

```

procedure TForm1.ScrollBar2Change(Sender: TObject);
begin
  timer2.Interval := scrollbar2.position;
  label8.Caption := inttostr(scrollbar2.position);
end;

```

12. Untuk **nibble rendah half step** menggunakan array, dengan nilai array

```

c : array [1..8] of integer=(1,3,2,6,4,12,8,9); //nibble low, half step

```

Kemudian pada procedure timer3 masukkan program sebagai berikut:

```
procedure TForm1.Timer3Timer(Sender: TObject); //TIMER3 HALF STEP
begin
  if(half>8) then half:=1;
  data := c[half];
  half := half+1;
  NYALAKAN8LED;
  //outporb($37A,data);
end;
```

13. Untuk **nibble tinggi full step** lakukan langkah berikut: masukkan program di bawah ini pada procedure timer4 nibble tinggi. Program ini menggunakan tipe array dengan jumlah data ada 4.

```
procedure TForm1.Timer4Timer(Sender: TObject);
begin
  if rendah>4 then rendah := 1;
  data := b[rendah];
  rendah:= rendah+1;
  NYALAKAN8LED;
end;
```

Untuk pengaturan tombol START dan STOP pada prinsipnya sama seperti sebelumnya, begitu juga pada pengaturan ScrollBar

14. Untuk **nibble tinggi half step** lakukan langkah berikut: masukkan program dibawah ini pada timer nibble tinggi. Program ini menggunakan tipe array dengan nilai:

```
a : array [1..8] of integer=(16,48,32,96,64,192,128,144); //nibble high half step
```

Masukan program pada procedure timer5 di bawah ini:

```
procedure TForm1.Timer5Timer(Sender: TObject); //HALF STEP
begin
  if step>8 then step := 1;
  data := a[step];
  step := step+1;
  NYALAKAN8LED;
end;
```

15. Untuk membuat checkbox lakukan langkah berikut: masukkan program dibawah ini untuk LED 1

```

procedure TForm1.CheckBox1Click(Sender: TObject);
begin
  data := data xor 1;
  NYALAKAN8LED;
end;

```

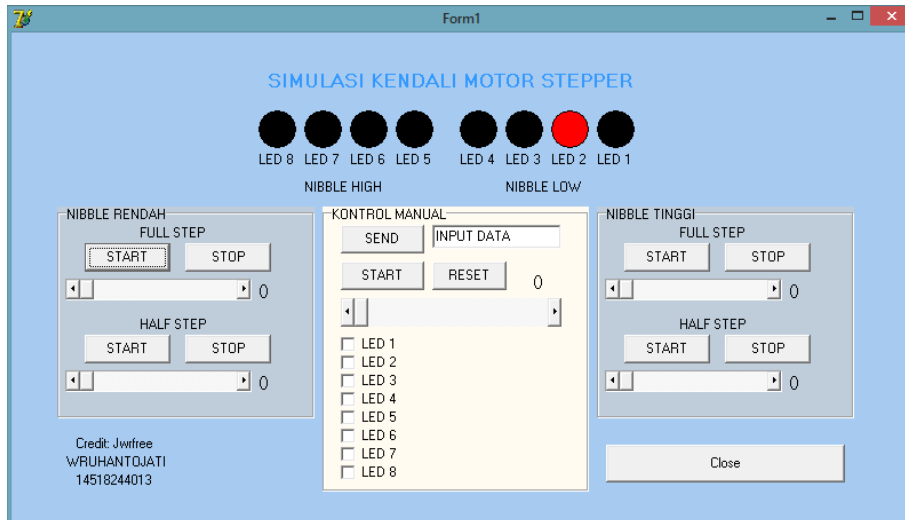
Untuk LED 2 sampai 8 pada prinsipnya sama hanya nilainya yang diganti misal LED 2 maka menjadi: `data := data xor 2;` , lalu 4, 8,..., 128.

16. Setelah selesai memprogram selanjutnya adalah mencoba menjalankan program, jika sudah benar maka dapat dihubungkan dengan modul motor stepper.
17. Hubungkan kabel dengan konektor DB25 pada modul praktik paralel ke port LPT1 di bagian belakang CPU.
18. Pastikan userport.sys sudah ada/terinstal di direktori:  
`C:\Windows\System32\Drivers\`.  
 Jalankan userport.exe dan klik tombol start, jika muncul pesan “*driven already started*” berarti port LPT1 telah terbuka.
19. Uji coba program dengan menekan tombol F9 pada keyboard.
20. Klik tombol start dan amati pergerakan motor stepper, coba atur kecepatan motor stepper dengan menggeser scrollbar. Lakukan langkah diatas untuk nibble rendah baik half maupun full step dan nibble tinggi baik half maupun full step.
21. Catat hasilnya.

## V. Hasil Praktikum

### 1. Full Step Nibble Rendah

Tampilan interface pada Delphi7 yaitu sebagai berikut:

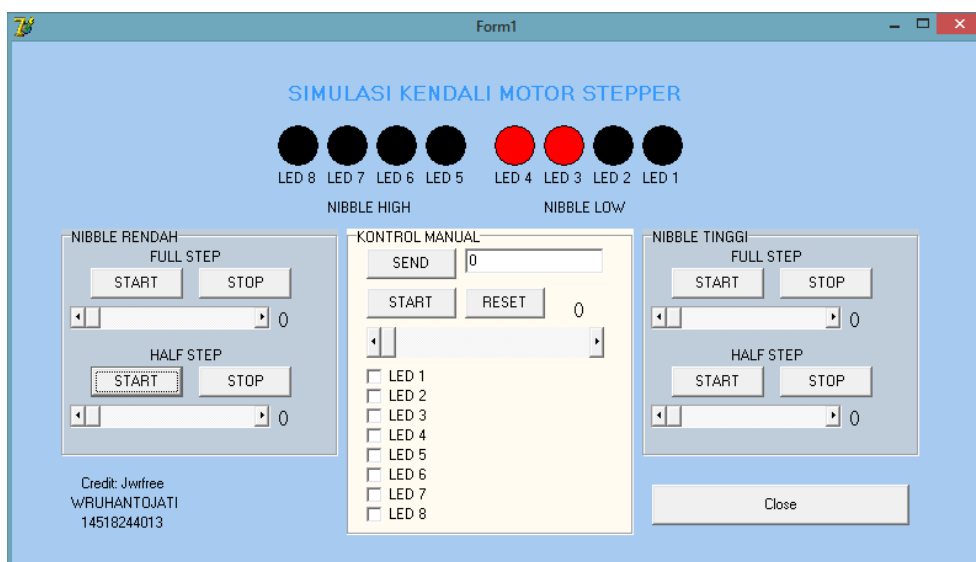


Tabel 1. data nibble rendah full step

FULL STEP							
BERLAWANAN JARUM JAM				SEARAH JARUM JAM			
LED4	LED3	LED2	LED1	LED4	LED3	LED2	LED1
1	0	0	0	0	0	0	1
0	1	0	0	0	0	1	0
0	0	1	0	0	1	0	0
0	0	0	1	1	0	0	0
1	0	0	0	0	0	0	1
0	1	0	0	0	0	1	0
0	0	1	0	0	1	0	0
0	0	0	1	1	0	0	0

## 2. Half Step Nibble Rendah

Tampilan interface pada Delphi7 yaitu sebagai berikut:

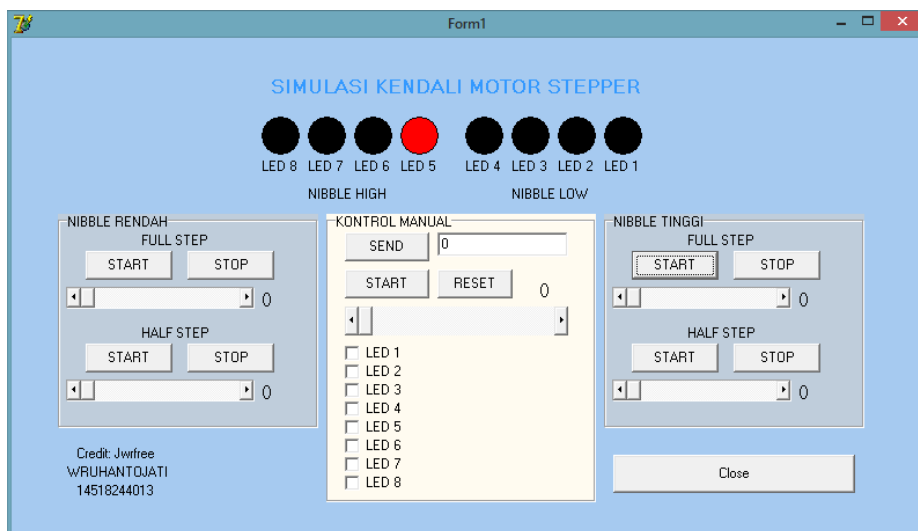


Tabel 2. data nibble rendah half step

HALF STEP							
BERLAWANAN JARUM JAM				SEARAH JARUM JAM			
LED4	LED3	LED2	LED1	LED4	LED3	LED2	LED1
1	0	0	0	0	0	0	1
1	1	0	0	0	0	1	1
0	1	0	0	0	0	1	0
0	1	1	0	0	1	1	0
0	0	1	0	0	1	0	0
0	0	1	1	1	1	0	0
0	0	0	1	1	0	0	0
1	0	0	1	1	0	0	1

### 3. Full Step Nibble Tinggi

Tampilan interface pada Delphi7 yaitu sebagai berikut:

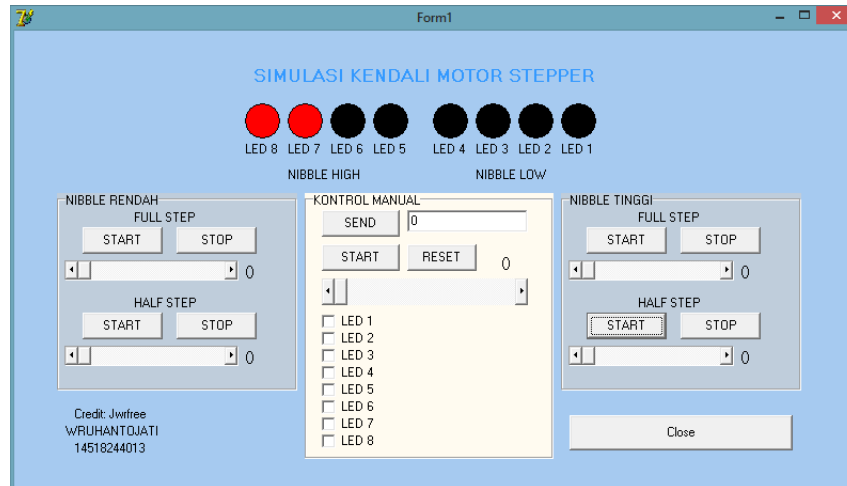


Tabel 3. data nibble tinggi full step

FULL STEP							
BERLAWANAN JARUM JAM				ARAH JARUM JAM			
LED8	LED7	LED6	LED5	LED8	LED7	LED6	LED5
1	0	0	0	0	0	0	1
0	1	0	0	0	0	1	0
0	0	1	0	0	1	0	0
0	0	0	1	1	0	0	0
1	0	0	0	0	0	0	1
0	1	0	0	0	0	1	0
0	0	1	0	0	1	0	0
0	0	0	1	1	0	0	0

### 4. Half Step Nibble Tinggi

Tampilan interface pada Delphi7 yaitu sebagai berikut:



Tabel 4. data nibble tinggi half step

HALF STEP							
BERLAWANAN JARUM JAM				SEARAH JARUM JAM			
LED8	LED7	LED6	LED5	LED8	LED7	LED6	LED5
1	0	0	0	0	0	0	1
1	1	0	0	0	0	1	1
0	1	0	0	0	0	1	0
0	1	1	0	0	1	1	0
0	0	1	0	0	1	0	0
0	0	1	1	1	1	0	0
0	0	0	1	1	0	0	0
1	0	0	1	1	0	0	1

## 5. Pembahasan

Pada pembahasan ini, modul yang digunakan yaitu motor stepper. Motor stepper adalah alat yang dapat mengubah data atau sinyal pulsa elektrik menjadi pergerakan mekanik. Motor stepper dapat berputar secara diskrit dengan derajat perubahan posisi yang tetap. Dimana setiap pulsa elektrik yang diterima, akan menghasilkan satu langkah putaran pada *Shaft* tersebut. Motor stepper dapat berotasi searah jarum jam (*clock wise*) atau berlawanan arah jarum jam (*counter clock wise*) tergantung dari sinyal yang diberikan.

Motor stepper dapat dikendalikan secara *full step* dan *half step*. Pengendalian secara *half step* lebih baik daripada pengendalian secara *full step* karena dengan pengendalian *half step* pergerakan dari motor lebih halus daripada menggunakan pengendalian dengan pengendalian *full step*.

a. Half Step

*Half Step* adalah cara mengendalikan motor stepper sehingga menghasilkan pergerakan motor yang lebih halus. Karena pergerakan rotor dalam motor stepper yang bergerak dengan sudut sebesar  $\frac{1}{2}$  derajat dari besar sudut antara 2 buah kutub (*coil*) yang berdekatan, sehingga pergerakan yang dihasilkan lebih halus.

b. Full step

*Full step* adalah cara mengendalikan motor stepper sehingga dihasilkan pergerakan motor namun tidak sehalus pergerakan *Half Step*. Hal ini disebabkan karena pergerakan rotor dalam motor stepper yang bergerak per 1 buah kutub (*coil*).

Berikut tabel perbandingan antara half step dan full step

Step ke	Full Step				Half Step			
1	1	0	0	0	1	0	0	0
2	0	1	0	0	1	1	0	0
3	0	0	1	0	0	1	0	0
4	0	0	0	1	0	0	1	0
5	Berulang ke step 1				0	0	1	0
6					0	0	1	1
7					0	0	0	1
8					1		0	1
					Berulang ke step 1			

Dari kedua tipe step tadi diujikan pada modul motor stepper dan diperoleh hasil sebagai berikut:

1. pada fasa full step baik nibble low maupun nibble high motor stepper diuji putaran selama 360 derajat dan diperoleh putaran selama 48 step.
2. pada fasa half step baik nibble low maupun nibble high motor stepper diuji putaran selama 360 derajat dan diperoleh putaran selama 48 step. Hal ini tidak menunjukkan adanya perbedaan dengan fasa full step.
3. pada fasa full step baik nibble low maupun nibble high motor stepper diuji putaran selama 90 derajat dan diperoleh putaran selama 12 step.
4. pada fasa half step baik nibble low maupun nibble high motor stepper diuji putaran selama 90 derajat dan diperoleh putaran selama 12 step.
5. pada fasa full step baik nibble low maupun nibble high motor stepper diuji putaran selama 60 derajat dan diperoleh putaran selama 8 step.

6. pada fasa half step baik nibble low maupun nibble high motor stepper diuji putaran selama 60 derajat dan diperoleh putaran selama 8 step.

## **VI. Kesimpulan**

Berdasarkan praktikum yang telah dilaksanakan, diperoleh kesimpulan sebagai berikut:

1. Half Step adalah cara mengendalikan motor stepper sehingga menghasilkan pergerakan motor yang lebih halus sedangkan full step adalah cara mengendalikan motor stepper sehingga dihasilkan pergerakan motor namun tidak sehalus pergerakan Half Step
2. Dalam uji coba rotasi 360 derajat, motor stepper yang digerakan dengan cara half step dan full step sama-sama bergerak selama 48 step.
3. Untuk input data nibble low menggunakan register data (\$378) sedangkan input data nibble high menggunakan register kontrol/kendali (\$37A)



## Lampiran.

unit delapanled;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls, ExtCtrls, Menus, ComCtrls;

type

```
TForm1 = class(TForm)
  Label1: TLabel;
  Shape1: TShape;
  Shape2: TShape;
  Shape3: TShape;
  Shape4: TShape;
  Shape5: TShape;
  Shape6: TShape;
  Shape7: TShape;
  Shape8: TShape;
  Edit1: TEdit;
  Button1: TButton;
  ScrollBar1: TScrollBar;
  Button2: TButton;
  Timer1: TTimer;
  Button3: TButton;
  Label3: TLabel;
  Label4: TLabel;
  Label5: TLabel;
  Button4: TButton;
  Button5: TButton;
  Timer2: TTimer;
  GroupBox1: TGroupBox;
  Label6: TLabel;
  GroupBox2: TGroupBox;
  Label2: TLabel;
  Label7: TLabel;
  Button6: TButton;
  Button7: TButton;
  Timer3: TTimer;
  ScrollBar2: TScrollBar;
  Label8: TLabel;
  ScrollBar3: TScrollBar;
  Label9: TLabel;
```

```
GroupBox3: TGroupBox;
Label10: TLabel;
Button8: TButton;
Button9: TButton;
ScrollBar4: TScrollBar;
Label11: TLabel;
Timer4: TTimer;
Label12: TLabel;
Button10: TButton;
Button11: TButton;
Timer5: TTimer;
CheckBox1: TCheckBox;
CheckBox2: TCheckBox;
CheckBox3: TCheckBox;
CheckBox5: TCheckBox;
CheckBox6: TCheckBox;
CheckBox7: TCheckBox;
CheckBox8: TCheckBox;
CheckBox9: TCheckBox;
Label13: TLabel;
Label14: TLabel;
Label15: TLabel;
Label16: TLabel;
Label17: TLabel;
Label18: TLabel;
Label19: TLabel;
Label20: TLabel;
Button12: TButton;
ScrollBar5: TScrollBar;
Label21: TLabel;
Label22: TLabel;
Label23: TLabel;
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure ScrollBar1Change(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure Timer2Timer(Sender: TObject);
procedure Button5Click(Sender: TObject);
procedure Timer3Timer(Sender: TObject);
procedure Button6Click(Sender: TObject);
procedure Button7Click(Sender: TObject);
procedure ScrollBar2Change(Sender: TObject);
procedure ScrollBar3Change(Sender: TObject);
```

```

procedure Timer4Timer(Sender: TObject);
procedure Button8Click(Sender: TObject);
procedure Button9Click(Sender: TObject);
procedure Timer5Timer(Sender: TObject);
procedure Button10Click(Sender: TObject);
procedure Button11Click(Sender: TObject);
procedure CheckBox1Click(Sender: TObject);
procedure CheckBox2Click(Sender: TObject);
procedure CheckBox3Click(Sender: TObject);
procedure CheckBox5Click(Sender: TObject);
procedure CheckBox6Click(Sender: TObject);
procedure CheckBox7Click(Sender: TObject);
procedure CheckBox8Click(Sender: TObject);
procedure CheckBox9Click(Sender: TObject);
procedure Button12Click(Sender: TObject);
procedure ScrollBar4Change(Sender: TObject);
procedure ScrollBar5Change(Sender: TObject);
private
  { Private declarations }
  procedure NYALAKAN8LED;
public
  { Public declarations }
end;

```

var

```

Form1: TForm1;
data : integer;
nilai : integer;
rendah : integer;
half : integer;
step: integer;
a : array [1..8] of integer=(16,48,32,96,64,192,128,144); //nibble high half step
b : array [1..4] of integer=(16,32,64,128);           //nibble high, full step
c : array [1..8] of integer=(1,3,2,6,4,12,8,9);       //nibble low, half step

```

implementation

```

{$R *.dfm}
procedure outportb(portid : integer; value : BYTE) ;
Begin
  asm
    mov edx, portid
    mov al, value
    out dx,al
  end;

```

```
end;
```

```
function inportb (portid : integer) : byte;
```

```
Var value : byte;
```

```
Begin
```

```
asm
```

```
mov edx, portid
```

```
in al, dx
```

```
mov value, al
```

```
end;
```

```
inportb := value;
```

```
end;
```

```
procedure TForm1.NYALAKAN8LED;
```

```
Begin
```

```
if (data AND 1)=1 then shape1.brush.color := clRED  
else Shape1.Brush.color := clBlack ;
```

```
if (data AND 2)=2 then shape2.brush.color := clRED  
else Shape2.Brush.color := clBlack ;
```

```
if (data AND 4)=4 then shape3.brush.color := clRED  
else Shape3.Brush.color := clBlack ;
```

```
if (data AND 8)=8 then shape4.brush.color := clRED  
else Shape4.Brush.color := clBlack ;
```

```
if (data AND 16)=16 then shape5.brush.color := clRED  
else Shape5.Brush.color := clBlack ;
```

```
if (data AND 32)=32 then shape6.brush.color := clRED  
else Shape6.Brush.color := clBlack ;
```

```
if (data AND 64)=64 then shape7.brush.color := clRED  
else Shape7.Brush.color := clBlack ;
```

```
if (data AND 128)=128 then shape8.brush.color := clRED  
else Shape8.Brush.color := clBlack ;
```

```
End;
```

```
procedure TForm1.Timer1Timer(Sender: TObject); //TIMER1
```

```
begin
```

```
data := data*2;
```

```
if (data>255) then
```

```
data := 1;
```

```
NYALAKAN8LED;
```

```
end;
```

```
procedure TForm1.Button1Click(Sender: TObject); //SEND
```

```
begin
```

```
data :=strtoint(edit1.Text);
```

```

    NYALAKAN8LED;
end;

procedure TForm1.Button3Click(Sender: TObject); //START
begin
    timer1.enabled := true;
    data := 1;
end;

procedure TForm1.Button2Click(Sender: TObject); //STOP
begin
    edit1.Text := '0';
    data := strtoint(edit1.Text);
    NYALAKAN8LED;
    timer1.enabled := false;
end;

procedure TForm1.ScrollBar1Change(Sender: TObject); //SCROLLBAR
begin
    timer1.Interval := scrollbar1.position;
    label6.Caption := inttostr(scrollbar1.Position);
    NYALAKAN8LED;
end;

procedure TForm1.Timer2Timer(Sender: TObject); //TIMER2 FULL STEP
begin
    data := data * 2;
    if(data>8) then
        data := 1;
    NYALAKAN8LED;
end;

procedure TForm1.Button4Click(Sender: TObject); //START2
begin
    data := 1;
    timer2.Enabled := true;
    NYALAKAN8LED;
end;

procedure TForm1.Button5Click(Sender: TObject); //STOP2
begin
    timer2.enabled := false;
    edit1.Text := '0';
    data := strtoint(edit1.Text);

```

```
    NYALAKAN8LED;  
end;
```

```
procedure TForm1.Timer3Timer(Sender: TObject); //TIMER3 HALF STEP  
begin  
    if(half>8) then half:=1;  
    data := c[half];  
    half := half+1;  
    NYALAKAN8LED;  
    //outporb($37A,data);  
end;
```

```
procedure TForm1.Button6Click(Sender: TObject); //START3  
begin  
    half := 1;  
    timer3.enabled := true;  
    NYALAKAN8LED;  
end;
```

```
procedure TForm1.Button7Click(Sender: TObject); //STOP3  
begin  
    nilai := 1;  
    timer3.enabled := false;  
    edit1.Text := '0';  
    data := strtoint(edit1.Text);  
    NYALAKAN8LED;  
end;
```

```
procedure TForm1.ScrollBar2Change(Sender: TObject);  
begin  
    timer2.Interval := scrollbar2.position;  
    label8.Caption := inttostr(scrollbar2.position);  
end;
```

```
procedure TForm1.ScrollBar3Change(Sender: TObject);  
begin  
    timer3.interval := scrollbar3.Position;  
    label9.Caption := inttostr(scrollbar3.position);  
end;
```

```
procedure TForm1.Timer4Timer(Sender: TObject); //FULL STEP  
begin  
    if rendah>4 then rendah := 1;
```

```

data := b[rendah];
rendah:= rendah+1;
NYALAKAN8LED;

end;

procedure TForm1.Button8Click(Sender: TObject);
begin
rendah := 1;
timer4.Enabled := true;
end;

procedure TForm1.Button9Click(Sender: TObject);
begin
timer4.enabled := false;

end;

procedure TForm1.Timer5Timer(Sender: TObject); //HALF STEP
begin
if step>8 then step := 1;
data := a[step];
step := step+1;
NYALAKAN8LED;
end;

procedure TForm1.Button10Click(Sender: TObject);
begin
timer5.enabled := true;
step := 1;
end;

procedure TForm1.Button11Click(Sender: TObject);
begin
timer5.enabled := false;
end;

//-----CHECKBOX-----
procedure TForm1.CheckBox1Click(Sender: TObject);
begin
data := data xor 1;
NYALAKAN8LED;
end;

procedure TForm1.CheckBox2Click(Sender: TObject);

```

```
begin
  data := data xor 2;
  NYALAKAN8LED;
end;

procedure TForm1.CheckBox3Click(Sender: TObject);
begin
  data := data xor 4;
  NYALAKAN8LED;
end;

procedure TForm1.CheckBox5Click(Sender: TObject);
begin
  data := data xor 8;
  NYALAKAN8LED;
end;

procedure TForm1.CheckBox6Click(Sender: TObject);
begin
  data := data xor 16;
  NYALAKAN8LED;
end;

procedure TForm1.CheckBox7Click(Sender: TObject);
begin
  data := data xor 32;
  NYALAKAN8LED;
end;

procedure TForm1.CheckBox8Click(Sender: TObject);
begin
  data := data xor 64;
  NYALAKAN8LED;
end;

procedure TForm1.CheckBox9Click(Sender: TObject);
begin
  data := data xor 128;
  NYALAKAN8LED;
end;

procedure TForm1.Button12Click(Sender: TObject);
begin
  close;
end;
```



```
procedure TForm1.ScrollBar4Change(Sender: TObject);  
begin  
timer4.interval := scrollbar4.Position;  
label11.Caption := inttostr(scrollbar4.position);  
end;
```

```
procedure TForm1.ScrollBar5Change(Sender: TObject);  
begin  
timer5.interval := scrollbar5.Position;  
label21.Caption := inttostr(scrollbar5.position);  
end;
```

```
end.
```