

LAPORAN INTERFACE PERIPHERAL INTERFACE PORT PARALLEL (DB-25) DENGAN MATLAB



Disusun oleh:

- | | |
|---------------------------|----------------|
| 1. Mareta Syafitri | (061230701282) |
| 2. Muhammad Jefri | (061230701283) |
| 3. Muhammad Reza Fahlevi | (061230701284) |
| 4. Muhammad Rizky Wahyudi | (061230701295) |
| 5. Prinda Prima | (061230701286) |
| 6. Rahma Aulia | (061230701287) |

Kelas: 2 TCA

Instruktur:

**Politeknik Negeri Sriwijaya
Tahun Akademik 2013**

LAPORAN INTERFACE PERIPHERAL

INTERFACE PORT PARALLEL (DB-25) DENGAN MATLAB

A. TUJUAN

- Mahasiswa mampu memahami jenis dan fungsi-fungsi setiap pin pada port parallel.
- Mahasiswa dapat memahami fungsi-fungsi dan syntax bahasa program matlab.
- Mahasiswa mampu membuat program menggunakan matlab untuk komunikasi menggunakan port parallel.
- Mahasiswa dapat mengirimkan data pada port parallel melalui port data (pin 2-9) yang ditunjukkan dengan indicator menggunakan LED.
- Mahasiswa dapat menerima data pada port parallel melalui port status (pin 10, 11, 12, 13 dan 15).

B. TEORI DASAR

1.1. PARALLEL PORT

Parallel port atau biasa disebut dalam bahasa Indonesia adalah port paralel merupakan sebuah port pada personal computer yang berfungsi sebagai alat komunikasi komputer (motherboard) dengan perangkat luar yang bersifat paralel. Pada port paralel, pemindahan informasi dapat dilakukan secara bersamaan sehingga informasi yang terkirim lebih banyak daripada port seri. Port paralel biasa dikenal dengan Printer Port atau Centronics Port. Fungsi port paralel adalah sebagai penghubung motherboard dengan Printer jenis lama, Zip drive, beberapa Scanner, Sound Cards, Web Cams, Gamepads, Joystick, pemrograman EPROM, peralatan SCSI melalui adapter paralel ke SCSI, percobaan dengan TTL 12 driver, dan External CD-R atau CD-RW. Selain itu, port paralel juga digunakan sebagai uji coba sederhana dalam perancangan peralatan elektronika.



Port Parallel 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 Parallel ini terdiri dari :

- 4 Jalur Kontrol.
- 5 Jalur Status.
- 8 Jalur Data.

Port paralel atau port printer sebenarnya terdiri dari tiga bagian yang masing-masing diberi nama sesuai dengan tugasnya dalam melaksanakan pencetakan pada printer. Tiga bagian tersebut adalah Data Port (DP), Printer Control (PC), dan Printer Status (PS). DP digunakan untuk mengirimkan kode-kode kontrol dari komputer ke printer, misalkan kode kontrol untuk menggulung kertas, dan PS digunakan untuk mengirimkan kode-kode status printer ke komputer, misalnya untuk menginformasikan bahwa kertas telah habis. DP, PC, dan PS sebenarnya adalah port-port 8 bit, namun hanya DP yang benar-benar 8 bit. Untuk PC dan PS, hanya beberapa bit saja yang dipakai yang berarti hanya beberapa bit saja dari port-port ini yang dapat kita manfaatkan untuk keperluan interfacing. Port PC adalah port baca atau tulis (read atau write), PS adalah port baca saja (read only), sedangkan port DP adalah port baca atau tulis juga. Akan tetapi, kemampuan ini hanya dimiliki oleh Enhanced Parallel Port (EPP), sedangkan port paralel standar hanya memiliki kemampuan tulis saja. Pada EPP, pengaturan arah jalur data DP dilakukan lewat bit 5 PC. Jika bit 5 PC bernilai 0, maka jalur data dwi-arah DP menjadi output dari port paralel, sebaliknya jika bit 5 PC bernilai 1 maka jalur data dwi-arah DP menjadi input dari port paralel.

Fungsi Pin Parallel Port pada Printer

1. Pin 1 membawa sinyal **strobe**. Sinyal ini menjaga levelnya tetap diantara 2.8 sampai 5 volt, tetapi voltasenya akan turun di bawah 0.5 volt ketika komputer mengirimkan data. Turunnya tegangan ini memberitahu printer bahwa data sedang dikirim.
2. Pin 2 sampai ke 9 digunakan untuk membawa 8 bit data, **carry data**. Untuk menyatakan nilai bit tersebut bernilai 1 adalah dengan pemberian tegangan 5 volt ke salah satu pin. Tidak adanya pemberian tegangan pada pin menandakan bit bernilai 0. Cara ini sangat efektif dalam mentransmisikan informasi digital melalui kabel analog.
3. Pin 10 mengirim sinyal **acknowledge** dari printer ke komputer. Seperti halnya pin 1, pin ini tetap menjaga tegangannya dan menurunkan tegangan hingga di bawah 0.5 volt agar komputer mengetahui bahwa data telah diterima.
4. Jika printer sibuk, **busy**, printer akan memberi tegangan ke pin 11. Kemudian tegangan akan turun di bawah 0.5 volt untuk memberi tahu komputer bahwa printer siap menerima data lagi.
5. Printer akan memberi informasi ketika sedang kehabisan kertas dengan memberi tegangan ke pin 12.
6. Selama komputer masih menerima tegangan dari pin 13, printer dinyatakan masih aktif, **online**.
7. Komputer mengirimkan sinyal **auto feed** ke printer melalui pin 14 menggunakan tegangan 5 volt.
8. Jika printer mempunyai masalah lainnya, printer menurunkan tegangan menjadi kurang dari 0.5 volt pada pin 15 dimaksudkan agar komputer mengetahui bahwa terdapat kesalahan.

9. Kapanpun print job yang baru sudah siap, komputer akan menurunkan tegangan pada pin 16 untuk inisialisasi printer.
10. Pin 17 digunakan oleh komputer untuk mengubah status printer menjadi offline. Selama pin ini masih diberi tegangan oleh komputer, printer akan tetap dalam status offline.
11. Pin 18-25 merupakan **ground** dan digunakan sebagai referensi sinyal tegangan rendah, dibawah 0.5 volt.

No Pin (DB-25)	No Pin (36-pin)	Nama Sinyal	Arah	Register-bit	Inverted
1	1	*strobe	In / out	Control-0	Yes
2	2	Data0	Out	Data-0	No
3	3	Data1	Out	Data-1	No
4	4	Data2	Out	Data-2	No
5	5	Data3	Out	Data-3	No
6	6	Data4	Out	Data-4	No
7	7	Data5	Out	Data-5	No
8	8	Data6	Out	Data-6	No
9	9	Data7	Out	Data-7	No
10	10	*ack	In	Status-6	No
11	11	Busy	In	Status-7	Yes
12	12	Paper out	In	Status-5	No
13	13	Select	In	Status-4	No
14	14	Auto feed	In / out	Control-1	Yes
15	15,32	*Error	In	Status-3	No
16	16,31	*Init	In / out	Control-2	No
17	17,36	*Select printer	In / out	Control-3	Yes
18-25	18-30,33	GND	-	-	-

1.2.MATLAB

MATLAB adalah sebuah bahasa dengan kemampuan tinggi untuk komputasi teknis. Ia menggabungkan **komputasi, visualisasi, dan pemrograman** dalam satu kesatuan yang mudah digunakan di mana masalah dan penyelesaiannya diekspresikan dalam notasi matematik yang sudah dikenal. Pemakaian MATLAB meliputi :

- Matematika dan komputasi
- Pengembangan algoritma
- Akuisisi data
- Pemodelan, simulasi dan prototype
- Grafik saintifik dan engineering
- Perluasan pemakaian, seperti graphical user interface (GUI).

MATLAB adalah system interaktif yang mempunyai basis data array yang tidak membutuhkan dimensi. Ini memungkinkan kita dapat menyelesaikan banyak masalah komputasi teknis, khususnya yang berkaitan dengan formulasi matrik dan vector.

Nama MATLAB merupakan singaktn dari *matrix labolatory* . MATLAB awalnya dibuat untuk memudahkan dalam mengakses software matriks yang telah dikembangkan oleh LINPACK dan EISPACK. Dalam perkembangannya, MATLAB mampu mengintegrasikan beberapa software matriks sebelumnya dalam satu software untuk komputasi matriks. Tidak hanya itu, MATLAB juga mampu melakukan komputasi simbolik yang biasa dilakukan oleh MAPLE.

Sistem MATLAB terdiri atas lima bagian utama :

- **Development Environment.** Ini adalah kumpulan semua alat-alat dan fasilitas untuk membantu kita dalam menggunakan fungsi dan file MATLAB. Bagian ini memuat **desktop, Command window, command history, editor and debugger,** dan **browser** untuk **melihat help, workspace, files.**
- **The MATLAB Mathematical Function Library.** Bagian ini adalah koleksi semua algoritma komputasi, mulai dari fungsi sederhana seperti **sum, sine, cosine** sampai fungsi lebih rumit seperti, **invers matriks, nilai eigen, fungsi Bessel** dan **fast Fourier transform.**
- **The MATLAB language.** Ini adalah bahasa matriks/array level tinggi dengan control flow, fungsi, struktur data, input/output, dan fitur objek programming lainnya.
- **Graphics.** MATLAB mempunyai fasilitas untuk menampilkan vector dan matriks sebagai grafik. Fasilitas ini mencakup visualisasi data dua / tiga dimensi, pemrosesan citra (image), animasi, dan grafik animasi.
- **The MATLAB Application Program Interface (API).** Paket ini memungkinkan kita menulis bahasa C dan Fortran yang berinteraksi dengan MATLAB. Ia memuat fasilitas untuk pemanggilan kode-kode dari MATLAB (dynamic linking), yang disebut MATLAB sebagai mesin penghitung, dan untuk membaca dan menulis MAT-files.

1.3.KOMUNIKASI MENGGUNAKAN PARALLEL PORT

Untuk berhubungan dengan alat lain, computer diberi tambahan berupa port parallel dan port serial. Pada port parallel data dikirimkan dalam format parallel (8-bit sekaligus) sehingga dapat dikirim dengan cepat. Namun untuk itu diperlukan banyak kabel dan tidak dapat dipergunakan untuk jarak jauh. Pada saat ini port parallel computer sudah jarang digunakan, akan tetapi beberapa peralatan elektronika masih banyak menggunakan port parallel ini untuk saling berkomunikasi.

Port parallel mempunyai beberapa nama yaitu LPT0, LPT1, dan LPT2 yang memiliki alamat sendiri-sendiri untuk nama-nama tersebut, yaitu:

LPT-0	LPT-1	LPT-2	NAMA
3BCH	378H	278H	PORT DATA
3BDH	379H	279H	PORT STATUS
3BEH	37AH	27AH	PORT CONTROL

Port parallel dapat digunakan dalam beberapa mode dan normalnya dapat dikonfigurasi melalui perangkat lunak. Mode-mode tersebut adalah:

- Mode SPP (Standard Parallel Port)
- Mode EPP (Enhanced Parallel Port)
- Mode ECP (Extended Capabilities Port)

Mode SSP merupakan suatu port parallel standar yang satu arah saja, sehingga bit-5 pada kanal control tidak akan respon. Pada mode ini port parallel hanya dapat digunakan sebagai output saja dan tidak dapat digunakan untuk masukan.

Mode EPP merupakan mode dua arah (bidirectional). Pada mode ini port parallel digunakan sebagai saluran input dan output. Dengan mode EPP ini dapat dilakukan akses ke register-register SPP (data, status dan control) serta mengakses register-register EPP. Pada mode ini juga, dapat membalikkan arah port menggunakan bit-5 dari register control. Mode EPP memiliki 2 tipe yaitu EPP1.7 dan EPP1.9 yang dapat mengakses bit time out pada register EPP. Selain itu mode EPP mempunyai kecepatan pengiriman data (transfer rate) antara 500kb/s sampai 2mbps.

Mode ECP akan memberikan port parallel dengan kemampuan tambahan (Extended Capabilities Port). Mode ini dapat diset melalui register control tambahan (Extended control register) dari ECP.

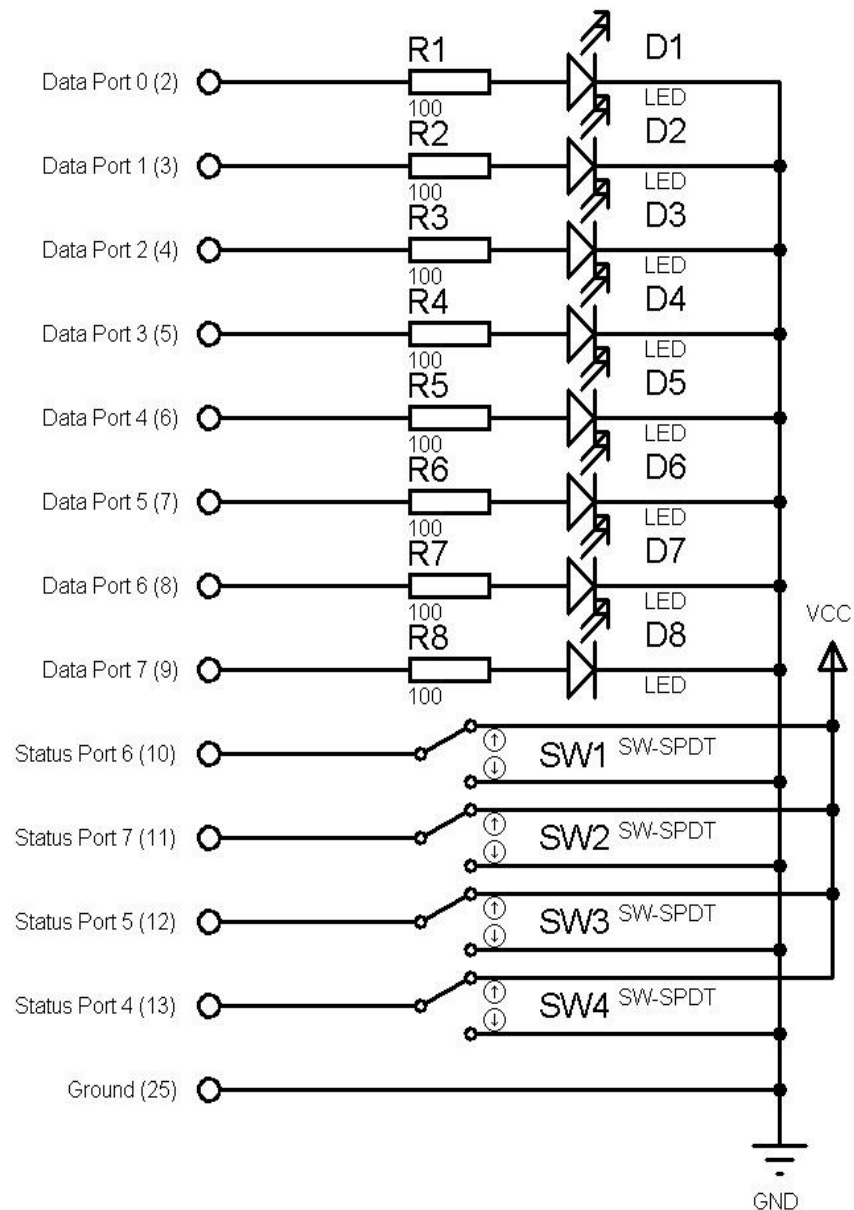
C. Alat dan Bahan

- Untuk rangkaian:

1. Port parallel (DB-25) male 1 buah
2. Papan PCB berpola 1 buah

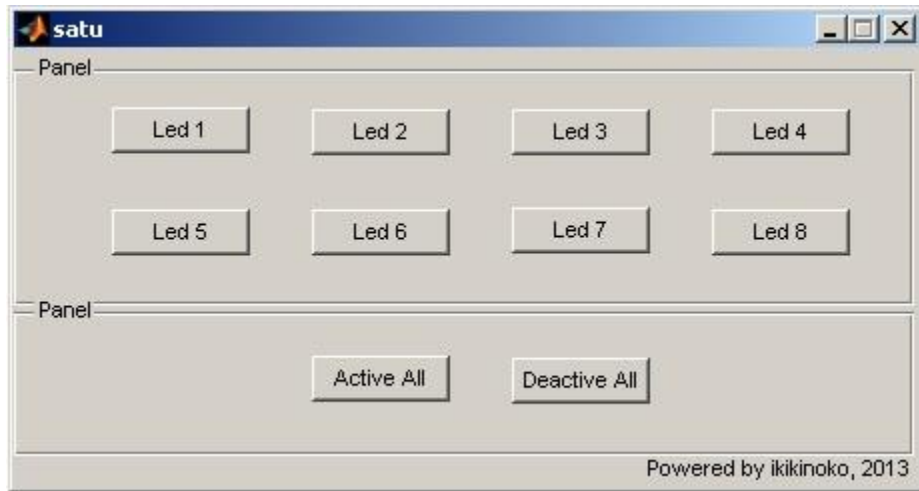
- | | |
|---------------------|------------|
| 3. LED | 8 buah |
| 4. Resistor 100 Ohm | 8 buah |
| 5. Saklar tiga kaki | 4 buah |
| 6. Kabel jumper | secukupnya |
- Untuk program:
 1. MATLAB R2007B dengan system operasi Microsoft Windows XP Professional

D. Gambar Rangkaian

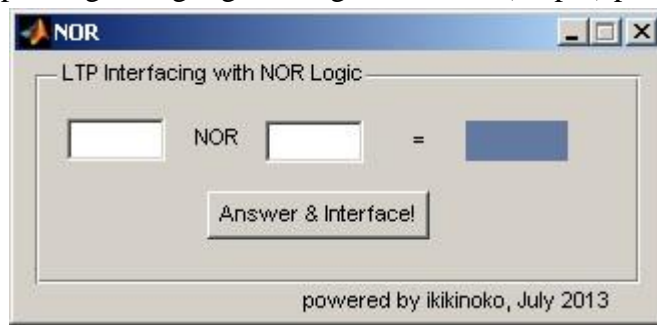


E. Gambar Program

1. Program mengaktifkan LED melalui port data parallel:



2. Program operasi gerbang logika dengan keluaran (output) port data parallel:



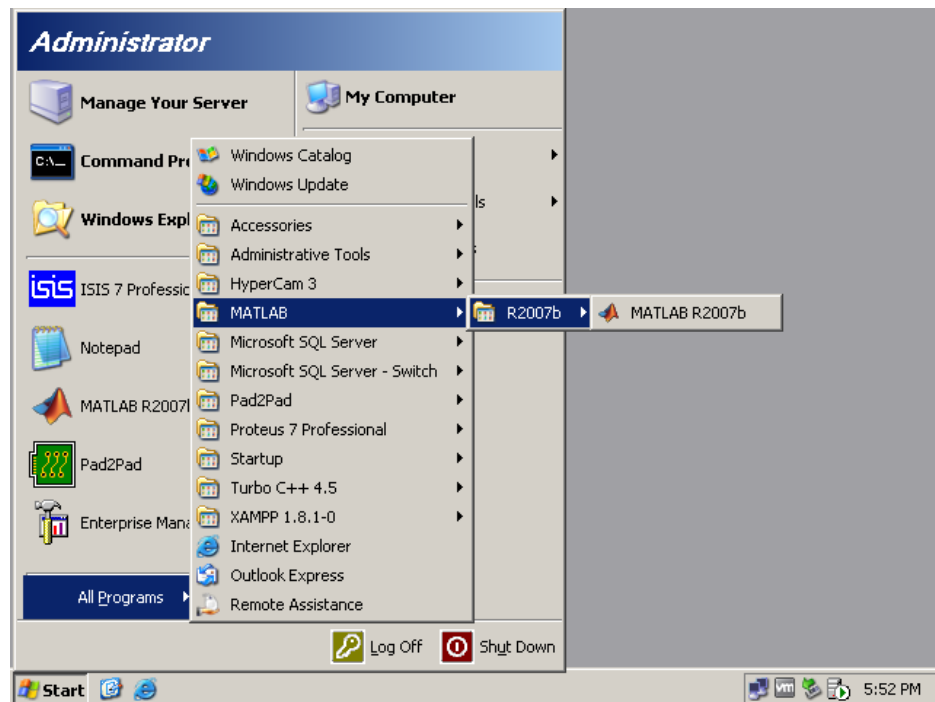
3. Program mengecek kondisi port status dengan keluaran (output) port data parallel:



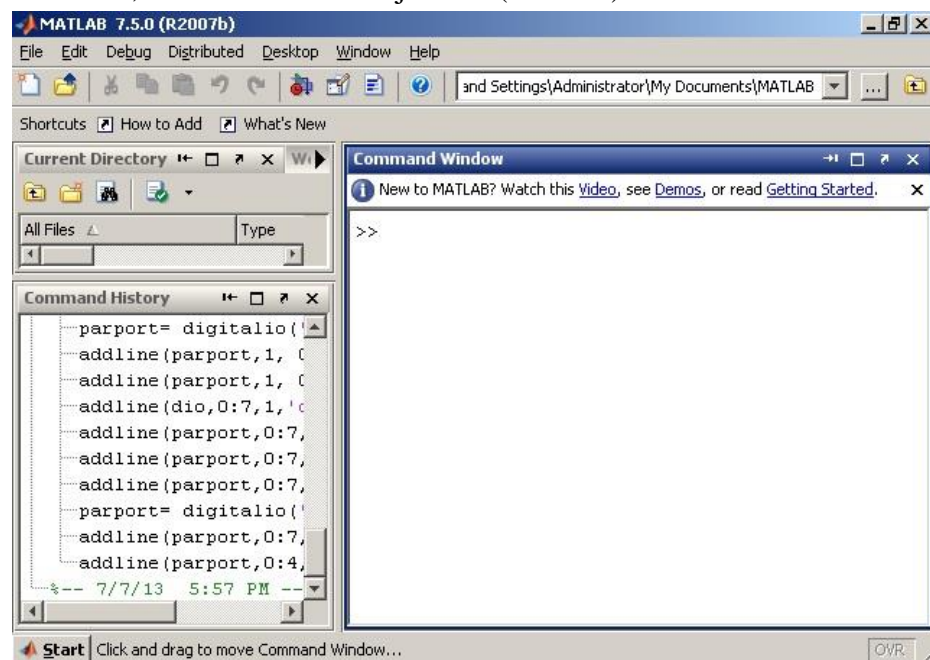
F. Langkah Kerja

- Untuk rangkaian:
 1. Buatlah rangkaian sesuai pada Gambar Rangkaian diatas papan PCB berpola.
 2. Perhatikan, bahwa VCC dan GND akan dihubungkan kedalam CPU yang akan digunakan. Bisa digunakan potongan kabel USB dengan kabel VCC dan GND yang sesuai.
 3. Pastikan bahwa rangkaian yang sudah dibuat benar, dan tidak ada beberapa kesalahan yang akan mengganggu kelancaran program yang nanti akan dibuat dan dijalankan.

4. Jika sudah dipastikan rangkaian benar, hubungkan rangkaian (DB-25 male) ke port parallel CPU (DB-25 female) atau tunggu sampai program telah selesai dibuat kemudian hubungkan kedua port tersebut.
- Untuk program:
1. Buka program MATLAB, untuk versi R2007B klik Start->All Programs->MATLAB->R2007B ->MATLAB R2007B.



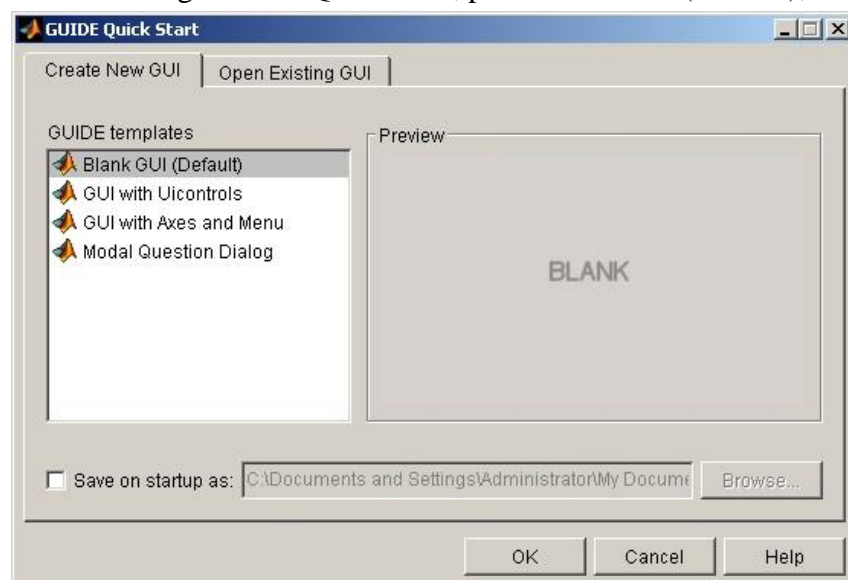
2. Setelah dibuka, maka akan muncul jendela (window) MATLAB R2007B.



3. Kemudian, bukalah MATLAB GUI dengan mengklik Start (MATLAB, bukan Windows)->MATLAB->GUIDE (GUI Builder).

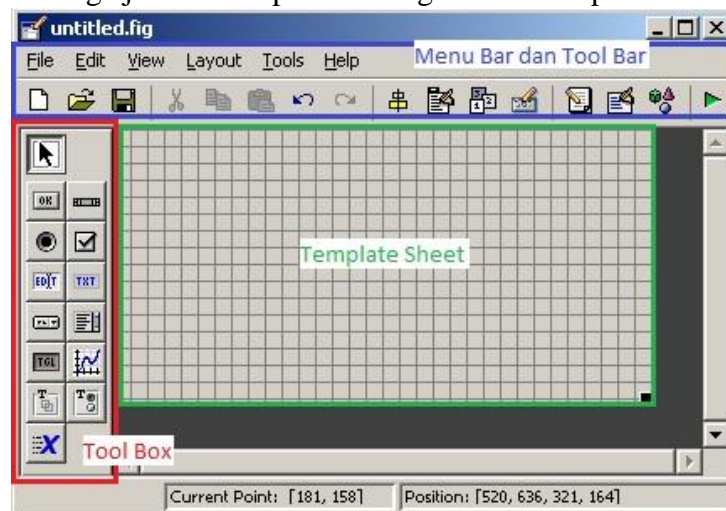



4. Lalu muncullah dialog GUIDE Quick Start, pilih Blank GUI (Default), dan klik

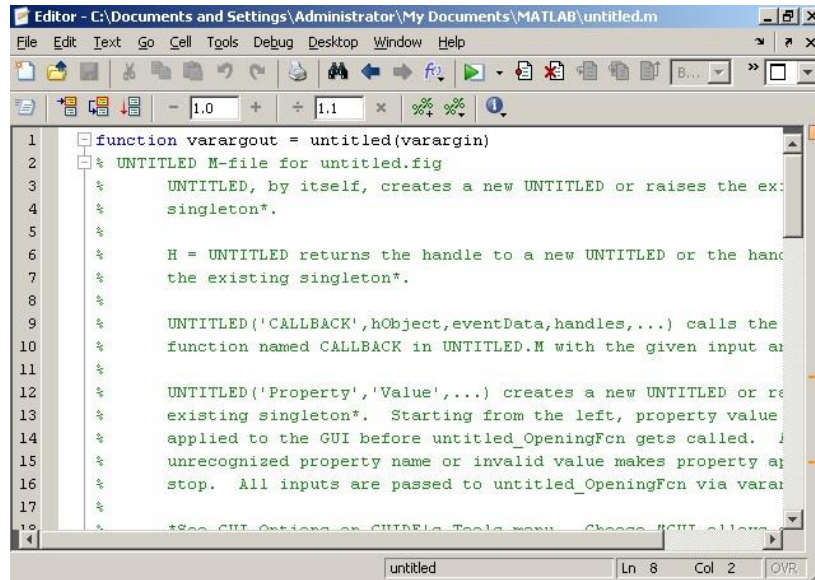


OK.

5. Setelah itu, muncullah jendela template.
6. Lalu buatlah tampilan GUI seperti pada tiga Gambar Program dengan menggunakan beberapa tool box yang sudah tersedia. Pembuatan GUI bisa dilakukan dalam tiga jendela template sekaligus secara terpisah.



7. Jika tampilan GUI sudah dibuat, klik M-File  Editor() dan kemudian akan mendapat konfirmasi untuk menyimpan GUI. Jadi, simpanlah GUI yang sudah dibuat dalam file .fig. Sebagai contoh:
 - Program pertama: satu.fig dengan satu.m
 - Program kedua: NOR.fig dengan NOR.m
 - Program ketiga: stat.fig dengan stat.m
8. Setelah disimpan, akan muncul jendela M-File Editor.



9. Perlu diketahui bahwa .m file merupakan inti dari program GUI yang akan kita pakai dalam melakukan antarmuka dengan port parallel. Dan setiap user interface yang digunakan seperti Push Button, Text Box, dan yang lainnya juga memiliki perintah atau syntax yang diperlukan sesuai kondisi yang dimana memiliki tanda atau tag yang berbeda-beda. Berikut ini merupakan perintah yang dipakai pada ketiga GUI:

- Program pertama:
 - a. Tombol Led 1 (tag: led1)


```
% --- Executes on button press in led1.
function led1_Callback(hObject, eventdata, handles)
% hObject      handle to led1 (see GCBO)
% eventdata    reserved - to be defined in a future version
of MATLAB
% handles      structure with handles and user data (see
GUIDATA)
parport= digitalio('parallel','LPT1');
get(parport,'PortAddress');
addline(parport, 0:7, 'out');
dataout = logical([1 0 0 0 0 0 0 0]);
putvalue(parport,dataout);
```
 - b. Tombol Led 2(tag: led2)


```
% --- Executes on button press in led2.
function led2_Callback(hObject, eventdata, handles)
% hObject      handle to led2 (see GCBO)
% eventdata    reserved - to be defined in a future version
of MATLAB
% handles      structure with handles and user data (see
GUIDATA)
parport= digitalio('parallel','LPT1');
get(parport,'PortAddress');
```

```

        addline(parport, 0:7, 'out');
        dataout = logical([0 1 0 0 0 0 0 0]);
        putvalue(parport,dataout);
c. Tombol Led 3(tag: led3)
% --- Executes on button press in led3.
function led3_Callback(hObject, eventdata, handles)
% hObject      handle to led3 (see GCBO)
% eventdata    reserved - to be defined in a future version
of MATLAB
% handles      structure with handles and user data (see
GUIDATA)
parport= digitalio('parallel','LPT1');
get(parport, 'PortAddress');
addline(parport, 0:7, 'out');
dataout = logical([0 0 1 0 0 0 0 0]);
putvalue(parport,dataout);
d. Tombol Led 4(tag: led4)
% --- Executes on button press in led4.
function led4_Callback(hObject, eventdata, handles)
% hObject      handle to led4 (see GCBO)
% eventdata    reserved - to be defined in a future version
of MATLAB
% handles      structure with handles and user data (see
GUIDATA)
parport= digitalio('parallel','LPT1');
get(parport, 'PortAddress');
addline(parport, 0:7, 'out');
dataout = logical([0 0 0 1 0 0 0 0]);
putvalue(parport,dataout);
e. Tombol Led 5(tag: led5)
% --- Executes on button press in led5.
function led5_Callback(hObject, eventdata, handles)
% hObject      handle to led5 (see GCBO)
% eventdata    reserved - to be defined in a future version
of MATLAB
% handles      structure with handles and user data (see
GUIDATA)
parport= digitalio('parallel','LPT1');
get(parport, 'PortAddress');
addline(parport, 0:7, 'out');
dataout = logical([0 0 0 0 1 0 0 0]);
putvalue(parport,dataout);
f. Tombol Led 6(tag: led6)
% --- Executes on button press in led6.
function led6_Callback(hObject, eventdata, handles)
% hObject      handle to led6 (see GCBO)
% eventdata    reserved - to be defined in a future version
of MATLAB
% handles      structure with handles and user data (see
GUIDATA)
parport= digitalio('parallel','LPT1');
get(parport, 'PortAddress');
addline(parport, 0:7, 'out');
dataout = logical([0 0 0 0 0 1 0 0]);
putvalue(parport,dataout);
g. Tombol Led 7(tag: led7)
% --- Executes on button press in led7.
function led7_Callback(hObject, eventdata, handles)
% hObject      handle to led7 (see GCBO)

```

```

% eventdata reserved - to be defined in a future version
of MATLAB
% handles structure with handles and user data (see
GUIDATA)
parport= digitalio('parallel','LPT1');
get(parport,'PortAddress');
addline(parport, 0:7, 'out');
dataout = logical([0 0 0 0 0 0 1 0]);
putvalue(parport,dataout);

```

h. Tombol Led 8(tag: led8)

```

% --- Executes on button press in led8.
function led8_Callback(hObject, eventdata, handles)
% hObject handle to led2 (see GCBO)
% eventdata reserved - to be defined in a future version
of MATLAB
% handles structure with handles and user data (see
GUIDATA)
parport= digitalio('parallel','LPT1');
get(parport,'PortAddress');
addline(parport, 0:7, 'out');
dataout = logical([0 0 0 0 0 0 0 1]);
putvalue(parport,dataout);

```

i. Tombol Active All (tag: ledall)

```

% --- Executes on button press in ledall.
function ledall_Callback(hObject, eventdata, handles)
% hObject handle to ledall (see GCBO)
% eventdata reserved - to be defined in a future version
of MATLAB
% handles structure with handles and user data (see
GUIDATA)
parport= digitalio('parallel','LPT1');
get(parport,'PortAddress');
addline(parport, 0:7, 'out');
dataout = logical([1 1 1 1 1 1 1 1]);
putvalue(parport,dataout);

```

j. Tombol Deactive All (tag: lednone)

```

% --- Executes on button press in lednone.
function lednone_Callback(hObject, eventdata, handles)
% hObject handle to lednone (see GCBO)
% eventdata reserved - to be defined in a future version
of MATLAB
% handles structure with handles and user data (see
GUIDATA)
parport= digitalio('parallel','LPT1');
get(parport,'PortAddress');
addline(parport, 0:7, 'out');
dataout = logical([0 0 0 0 0 0 0 0]);
putvalue(parport,dataout);

```

- Program kedua:

- a. Untuk masukan pertama (tag: edit1)
Perintah untuk tag ini tidak perlu diisi atau default.
- b. Untuk masukan kedua (tag: edit2)
Sama dengan edit1.
- c. Untuk keluaran (tag: text1)
Sama dengan a. dan b.
- d. Tombol Answer and Interface! (tag: pushbutton1)

```

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
box1str = get(handles.edit1, 'string');
box1num = str2num(box1str);
box2str = get(handles.edit2, 'string');
box2num = str2num(box2str);

box3num = ~(box1num | box2num);
box3str = num2str(box3num);

set(handles.text2, 'string', box3str);

if box3num == 1
    parport= digitalio('parallel', 'LPT1');
    get(parport, 'PortAddress');
    addline(parport, 0:7, 'out');
    dataout = logical([1 0 0 0 0 0 0 0]);
    putvalue(parport, dataout);
else
    parport= digitalio('parallel', 'LPT1');
    get(parport, 'PortAddress');
    addline(parport, 0:7, 'out');
    dataout = logical([0 0 0 0 0 0 0 0]);
    putvalue(parport, dataout);
end

```

- Program ketiga:

- a. Untuk keluaran “Your Output” (tag: edit1)
Tidak ada modifikasi.

- b. Tombol Update! (tag: pushbutton1)

```

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
parport= digitalio('parallel', 'LPT1');
inreg = addline (parport, 1:4, 1, 'in');
ram = getvalue(parport.line(1:4));

if ram == [0 0 0 0]
    parport= digitalio('parallel', 'LPT1');
    get(parport, 'PortAddress');
    addline(parport, 0:3, 'out');
    dataout = logical([0 0 0 0]);
    putvalue(parport, dataout);
    ans = 0
end

if ram == [1 0 0 0]
    parport= digitalio('parallel', 'LPT1');
    get(parport, 'PortAddress');

```

```

addline(parport, 0:3, 'out');
dataout = logical([1 0 0 0]);
putvalue(parport,dataout);
ans = 1
end

if ram == [0 1 0 0]
parport= digitalio('parallel','LPT1');
get(parport,'PortAddress');
addline(parport, 0:3, 'out');
dataout = logical([0 1 0 0]);
putvalue(parport,dataout);
ans = 2
end

if ram == [1 1 0 0]
parport= digitalio('parallel','LPT1');
get(parport,'PortAddress');
addline(parport, 0:3, 'out');
dataout = logical([1 1 0 0]);
putvalue(parport,dataout);
ans = 3
end

if ram == [0 0 1 0]
parport= digitalio('parallel','LPT1');
get(parport,'PortAddress');
addline(parport, 0:3, 'out');
dataout = logical([0 0 1 0]);
putvalue(parport,dataout);
ans = 4
end

if ram == [1 0 1 0]
parport= digitalio('parallel','LPT1');
get(parport,'PortAddress');
addline(parport, 0:3, 'out');
dataout = logical([1 0 1 0]);
putvalue(parport,dataout);
ans = 5
end

if ram == [0 1 1 0]
parport= digitalio('parallel','LPT1');
get(parport,'PortAddress');
addline(parport, 0:3, 'out');
dataout = logical([0 1 1 0]);
putvalue(parport,dataout);
ans = 6
end

if ram == [1 1 1 0]
parport= digitalio('parallel','LPT1');
get(parport,'PortAddress');
addline(parport, 0:3, 'out');
dataout = logical([1 1 1 0]);
putvalue(parport,dataout);
ans = 7
end

if ram == [0 0 0 1]

```

```

parport= digitalio('parallel','LPT1');
get(parport,'PortAddress');
addline(parport, 0:3, 'out');
dataout = logical([0 0 0 1]);
putvalue(parport,dataout);
ans = 8
end

if ram == [1 0 0 1]
parport= digitalio('parallel','LPT1');
get(parport,'PortAddress');
addline(parport, 0:3, 'out');
dataout = logical([1 0 0 1]);
putvalue(parport,dataout);
ans = 9
end

if ram == [0 1 0 1]
parport= digitalio('parallel','LPT1');
get(parport,'PortAddress');
addline(parport, 0:3, 'out');
dataout = logical([0 1 0 1]);
putvalue(parport,dataout);
ans = 10
end

if ram == [1 1 0 1]
parport= digitalio('parallel','LPT1');
get(parport,'PortAddress');
addline(parport, 0:3, 'out');
dataout = logical([1 1 0 1]);
putvalue(parport,dataout);
ans = 11
end

if ram == [0 0 1 1]
parport= digitalio('parallel','LPT1');
get(parport,'PortAddress');
addline(parport, 0:3, 'out');
dataout = logical([0 0 1 1]);
putvalue(parport,dataout);
ans = 12
end

if ram == [1 0 1 1]
parport= digitalio('parallel','LPT1');
get(parport,'PortAddress');
addline(parport, 0:3, 'out');
dataout = logical([1 0 1 1]);
putvalue(parport,dataout);
ans = 13
end

if ram == [0 1 1 1]
parport= digitalio('parallel','LPT1');
get(parport,'PortAddress');
addline(parport, 0:3, 'out');
dataout = logical([1 1 1 1]);
putvalue(parport,dataout);
ans = 14
end

```



```

if ram == [1 1 1 1]
parport= digitalio('parallel','LPT1');
get(parport,'PortAddress');
addline(parport, 0:3, 'out');
dataout = logical([1 1 1 1]);
putvalue(parport,dataout);
ans = 15
end

```

```

set(handles.text1,'string', ans);

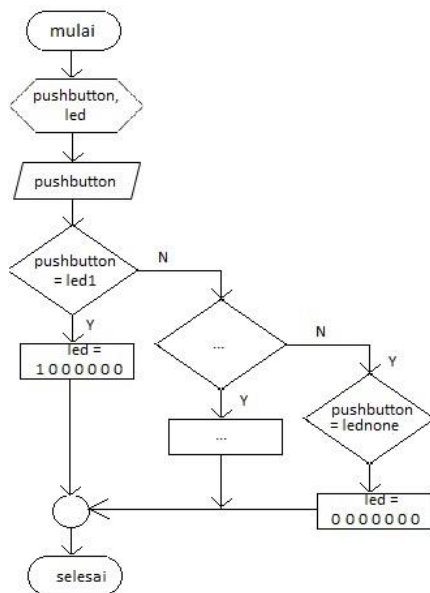
```

10. Jika semua perintah sudah diisi sesuai dengan tag, simpan M-File Editor.
11. Kemudian, jalankan program dengan mengklik Run “nama program” (). Lalu akan muncul GUI yang sudah dibuat sebelumnya.
12. Sebagai analisa, coba hubungkan rangkaian yang sudah dirangkai ke port parallel CPU dan jalankan ketiga program yang sudah dibuat. Lakukan perintah sesuai program yang dijalankan serta amati apa yang akan terjadi pada rangkaiannya.

G. Hasil dan Analisa Percobaan

Berikut hasil dan analisa dari ketiga program.

1. Program Pertama



Program ini pada dasarnya untuk menguji semua port data dari 0 hingga 7 dengan led sebagai indikator aliran tegangan. Kita tahu bahwa semua port data tidak ada yang bersifat inverter yang artinya semua akan ‘true’ jika aktif tinggi (tegangan 5 volt).

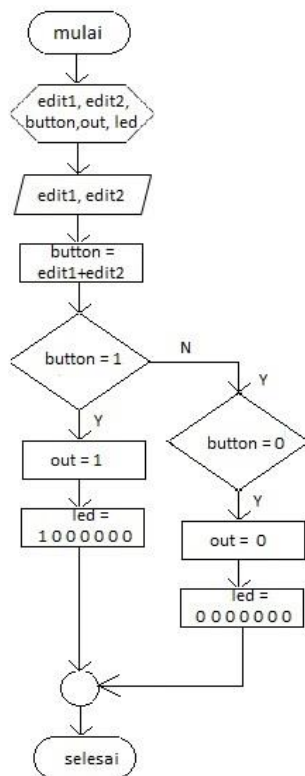
Pada program MATLAB sendiri, proses mengaktifkan led tidak hanya memanggil satu port data saja, melainkan semua port data dengan menggunakan operasi matriks. Misalnya, untuk mengaktifkan port data ke-4:

```
dataout = logical([0 0 0 1 0 0 0 0]);
```

Kita bisa lihat, bahwa [0 0 0 1 0 0 0 0] merupakan operasi matriks yang mewakili semua port data yang dimana port keempat akan bersifat aktif tinggi.

Dan seterusnya dengan memasukkan yang sama, hingga saat kita mengaktifkan semua led atau mematikan semua led pada port data.

2. Program Kedua



Program ini tidak jauh berbeda dengan program pertama, yang juga memanfaatkan port data sebagai keluaran. Namun keluaran ini berasal dari masukan operasi gerbang logika NOR.

Kondisi keluaran akan satu jika semua masukan adalah nol, dan kemudian akan mengaktifkan port data 1 untuk menhidupkan led. Serta, sebaliknya akan mematikan seluruh led jika keluarannya nol.

Pada MATLAB, operasi gerbang logika yang dikenal hanya AND (&), OR (|), NOT (~), dan XOR (xor). Jadi saat ingin meminta operasi NOR, maka prosesnya:

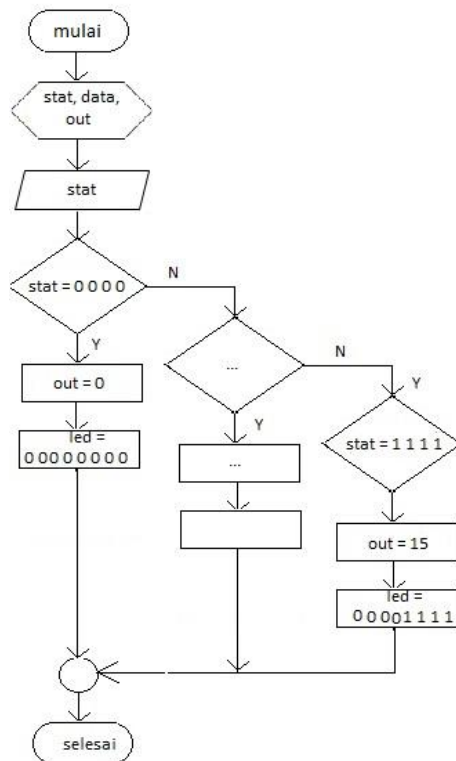
```
box3num = ~(box1num | box2num);
```

Yang dimana box1num (edit1) di-OR-kan dengan box2num (edit2) dan kemudian semuanya di-NOT-kan (NOR artinya NOT OR). Disamping itu, untuk statemen pada keluarannya digunakan fungsi if...else...end:

```

if box3num == 1
    parport= digitalio('parallel', 'LPT1');
...
else
    parport= digitalio('parallel', 'LPT1');
...
End
  
```

3. Program Ketiga



Program ini agak sedikit lebih sulit dibandingkan kedua program sebelumnya, yang dimana tidak hanya menggunakan port data saja sebagai keluaran tapi juga menggunakan port status sebagai masukan.

Dasarnya program ini hanya menggunakan port status 10 hingga 13 saja, sehingga saat menambahkan line pada program menjadi:

```
inreg=addline (parport, 1:4,1,'in');
```

Salah satu port status yang kita pakai ada yang bersifat inverter yakni port ke-11. Jadi kita harus mengkondisikan aktif rendah agar dapat mendapat 'true' untuk port ini.

Dalam mengaktifkan kondisi nol dan satu untuk port status ini, digunakanlah saklar tiga kaki pada rangkaian yang dimana satu kaki dihubungkan ke port status dan sisanya dihubungkan ke VCC maupun GND sebagai keluaran.

Dengan saklar, kita mampu memberikan kondisi port status ke program dengan memanfaatkan port data sebagai indikator keluaran dari program berupa fungsi IF. Dan perlu diketahui, bahwa MATLAB tidak mendukung system pengulangan (if...else if...end) berantai yang dimana terdapat fungsi IF lebih dari dua dalam satu

pengulangan sehingga pada program hanya dipakai fungsi if...end sebanyak 15 kali (dari 0000 hingga 1111).

```
if ram == [0 0 0 1]
...
end

if ram == [0 0 0 1]
...
End
...
if ram == [1 1 1 1]
...
End
```

4. Kesamaan dan Perbedaan

Kedamaan dari ketiga program ini adalah cara memanggil port parallel pada MATLAB. Pada dasarnya, MATLAB memerlukan interface terhadap perangkat keras terlebih dahulu melalui perintah digitalio dan disertai nama perangkat dan alias (bisa salah satu):

```
parport= digitalio('parallel','LPT1');
```

Biasanya MATLAB akan memberikan respon, jika kita tidak menyertakan “;” diakhir perintah:

```
>> parport= digitalio('parallel','LPT1')
```

```
Display Summary of DigitalIO (DIO) Object Using 'PC Parallel Port Hardware'.
```

```
Port Parameters:  Port 0 is port configurable for reading and writing.
```

```
Port 1 is port configurable for reading.
```

```
Port 2 is port configurable for reading and writing.
```

```
Engine status:  Engine not required.
```

```
DIO object contains no lines.
```

```
>>
```

Untuk melihat alamat port:

```
>> get(parport,'PortAddress')
```

```
ans =
```

```
0x378
```

```
>>
```

Setelah selesai memanggil port parallel, kita perlu menambahkan port data atau port status. Untuk port data:

```
>> addline(parport, 0:7, 0, 'out')
```

Index:	LineName:	HwLine:	Port:	Direction:
1	'Pin2'	0	0	'Out'
2	'Pin3'	1	0	'Out'
3	'Pin4'	2	0	'Out'
4	'Pin5'	3	0	'Out'
5	'Pin6'	4	0	'Out'

6	'Pin7'	5	0	'Out'
7	'Pin8'	6	0	'Out'
8	'Pin9'	7	0	'Out'

>>

Dari hasil perintah menunjukkan bahwa kita menambahkan 8 port (0:7) yang merupakan port 0 (port data adalah 0) yang mengarah sebagai keluaran ('Out'). Untuk memanggil salah satu port data, bisa dipakai:

```
>>putvalue(parport.line(1),1);
```

Perintah tersebut memanggil port data 1 (pin 3, line(1)) dalam kondisi aktif tinggi (1). Atau bisa menggabungkan operasi gerbang logika untuk memanggil lebih dari satu port data:

```
>>dataout = logical([0 0 0 0 1 0 0 0]);
>>putvalue(parport,dataout);
```

Sedangkan untuk menambahkan port status:

```
>> addline(parport, 0:4, 1, 'in')
```

Index:	LineName:	HwLine:	Port:	Direction:
9	'Pin15'	0	1	'In'
10	'Pin13'	1	1	'In'
11	'Pin12'	2	1	'In'
12	'Pin10'	3	1	'In'
13	'Pin11'	4	1	'In'

>>

Dari hasil perintah menunjukkan bahwa kita menambahkan 5 port (0:4) yang merupakan port 1 (port status adalah 1) yang mengarah sebagai masukan ('In'). Sedangkan untuk mengecek kondisi port status dimana port yang akan dicek dari port status 10, 11, 12, 13, dan 15:

```
>>ram = getvalue(parport.line(0:4));
```

Perbedaan dari ketiga program sebenarnya tidak ada yang mencolok, melainkan tujuan dari program yang akan dilaksanakan. Program pertama hanya untuk mengaktifkan led dari port data, program kedua juga serupa dengan program pertama tapi hanya saja menggunakan masukan berupa operasi gerbang logika, dan program ketiga menggunakan port status sebagai masukan dengan keluaran berupa angka dan port data.

H. Kesimpulan

Singkat cerita, bahwa antarmuka (interface) dengan port parallel (DB-25) juga dapat dioperasikan melalui program MATLAB yang dimana program ini lebih mencolok dalam operasi matematika meskipun MATLAB merupakan bahasa pemrograman komputer. Disamping itu, perintah yang digunakan tidak serumit dan kompleks ketimbang beberapa program lain seperti C++, Visual Basic, dan lainnya. Dan yang terakhir, MATLAB didukung graphic user interface (GUI) yang membuat penggunaan program lebih praktis dan *user friendly*.

---***---

Palembang, 8 Juli 2013
Instruktur,