

PENGUNAAN *PORT* PARALEL KOMPUTER PRIBADI UNTUK MENGENDALIKAN 4 (EMPAT) BUAH *PROGRAMMABLE PERIPHERAL INTERFACE 8255* MENGGUNAKAN *GNU COMPILER COLLECTION* DAN *GAMBAS*

Imam Firmansyah¹, Muhammad Yusro², Hamiddilah Ajie³

^{1,2,3}Jurusan Teknik Elektro, Fakultas Teknik, Universitas Negeri Jakarta

Jl. Rawamangun Muka, Jakarta 13220

Email : imam_fisika@yahoo.co.id

Abstrak

Dalam teknik antarmuka PC (personal computer), port paralel sering digunakan sebagai port I/O (input/output) ke peralatan luar yang dikendalikan oleh PC, namun pada port paralel PC memiliki jumlah I/O yang terbatas, yaitu terdapat jalur I/O sebanyak 12 dan 5 jalur input saja. Sehingga jumlah peralatan yang dapat dikendalikan oleh PC juga terbatas, agar PC dapat menangani jumlah peralatan yang lebih, maka jumlah I/O yang dimiliki komputer perlu diekspansi atau diperluas salah satunya dengan menggunakan PPI (Programmable Peripheral Interface) 8255 yang biasanya menggunakan slot ISA pada computer yang sekarang susah ditemukan pada computer-komputer saat ini. Penelitian ini bertujuan untuk mengekspansi I/O computer dengan cara mengendalikan beberapa PPI 8255 (dalam penelitian ini 4 buah) menggunakan port paralel PC, perangkat lunak yang digunakan merupakan freeware dan berlisensi GPL (GNU Public License) yaitu bahasa pemrograman GUI Gambas di lingkungan linux Ubuntu 8.04 dengan bantuan compiler gcc. Teknik pemrograman yang digunakan yaitu dengan membuat library khusus untuk mengakses PPI 8255 dengan port paralel PC menggunakan bahasa C dengan bantuan compiler gcc yang terdapat pada Ubuntu 8.04, kemudian library tersebut digunakan dalam lingkungan bahasa pemrograman GUI Gambas.

Kata kunci : ekspansi I/O; GCC; Gambas; port paralel; PPI 8255

Pendahuluan

Port paralel pada PC banyak digunakan oleh mahasiswa teknik, peneliti dan lain sebagainya untuk mengendalikan peralatan luar sederhana misalnya pengendalian peralatan listrik rumah, pengendalian pada sistem pengamanan rumah dan lain sebagainya. kemudian komputer pun mulai digunakan untuk pengendalian peralatan-peralatan pada industri, badan penelitian juga perusahaan-perusahaan besar. Kemudian ketika komputer dituntut untuk mengendalikan banyak peralatan maka komputer harus dapat menyediakan banyak jalur I/O yang terhubung ke peralatan tersebut.

Untuk memperbanyak I/O komputer salah satunya dengan menggunakan IC *Programmable Peripheral Interface* (PPI) 8255 dengan menggunakan jalur alamat dan data yang terhubung ke slot ISA, namun slot ISA mulai menghilang dari *motherboard* mulai dari pentium III. Penelitian ini bertujuan untuk menggunakan *port* paralel sebagai pengganti dari slot ISA untuk mengendalikan beberapa PPI 8255. Konsepnya menggunakan konsep dari penggunaan jalur data dan alamat bersama seperti pada sistem mikrokomputer 8088, dimana menggunakan IC tambahan yaitu IC demultiplexer 74LS138 dan IC buffer tri-state 74LS244, untuk pengambilan masukan menggunakan metode *nibble* menggunakan IC 74LS157.

Pemrograman *port* paralel komputer pribadi dilakukan di lingkungan Linux menggunakan GCC (GNU Compiler Collection) merupakan compiler bahasa C dan dapat juga digunakan untuk membuat *file dynamic library* dari bahasa C yang telah dibuat. Dan bahasa pemrograman Gambas yang merupakan bahasa pemrograman visual mirip Visual Basic, digunakan untuk membuat pemrograman GUI dengan mudah seperti Visual Basic.

Teknik pemrograman dilakukan dengan membuat fungsi-fungsi khusus untuk mengirim data ke PPI 8255 berdasarkan *timing waveform* dari mode 0 PPI 8255 menggunakan bahasa C, fungsi-fungsi yang dibuat yaitu: fungsi keluar digunakan untuk mengeluarkan data dari komputer ke PPI 8255, fungsi masuk, digunakan untuk mengambil data dari PPI 8255 ke dalam komputer, dan fungsi *reset* digunakan untuk mereset PPI 8255 secara software.

Metode Penelitian

Penelitian ini dilaksanakan di Laboratorium Teknik Elektro Fakultas Teknik Universitas Negeri Jakarta. Waktu penelitian dilaksanakan mulai bulan Oktober 2007 sampai bulan Juni 2008

Metode yang digunakan pada penelitian ini adalah eksperimen laboratorium. dimulai dengan perancangan rangkaian, pembuatan rangkaian, perancangan program, pembuatan program, pengujian rangkaian dan program serta pengujian dengan menggerakkan led dot matriks dan menerima masukan *pushbutton*.

Perancangan Rangkaian dan Program

1. Perancangan Rangkaian Sistem Pengendalian 4 PPI 8255 dengan Port Paralel PC

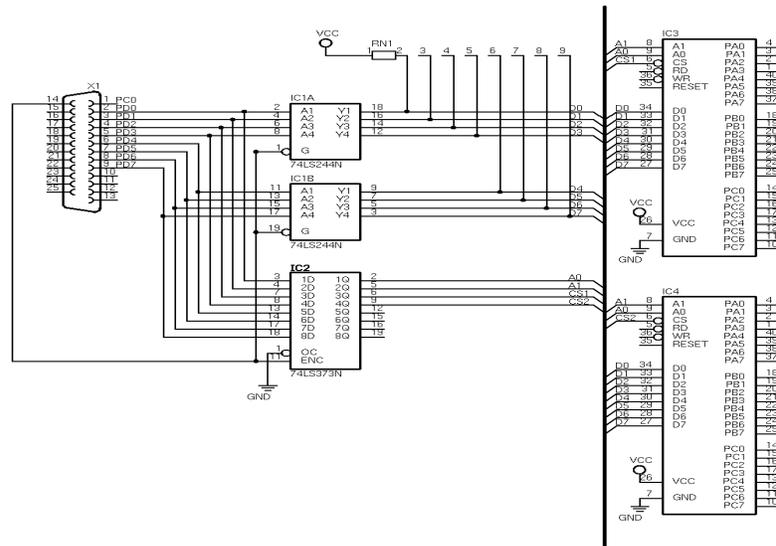
a. Perancangan Penggunaan Port Data sebagai Jalur Data dan Alamat PPI 8255

Sistem Pengendalian 4 PPI 8255 dengan port paralel PC merupakan pengendalian 4 PPI 8255 dengan port paralel PC dengan menggunakan konsep penggunaan jalur data dan alamat bersama.

Port data pada port paralel digunakan sebagai jalur data dan alamat ke 4 PPI 8255. Agar port data port paralel dapat digunakan untuk jalur data dan alamat maka dibutuhkan komponen IC tambahan yaitu:

- 1) 74LS244 merupakan IC *Buffer Tristate* untuk meneruskan atau tidak meneruskan data dari port data ke jalur data PPI 8255.
- 2) IC 74LS373 merupakan IC *Latch* (Pengunci) digunakan untuk meneruskan data dari port paralel ke jalur alamat pada PPI dan mengunci data yang telah dikirimkan.
- 3) *Resistor Pack* digunakan sebagai Resistor *Pull-up* pada jalur data PPI 8255 yang fungsinya untuk memberikan logika tinggi pada saat jalur data PPI 8255 *High Impedance* (kondisi ambang).

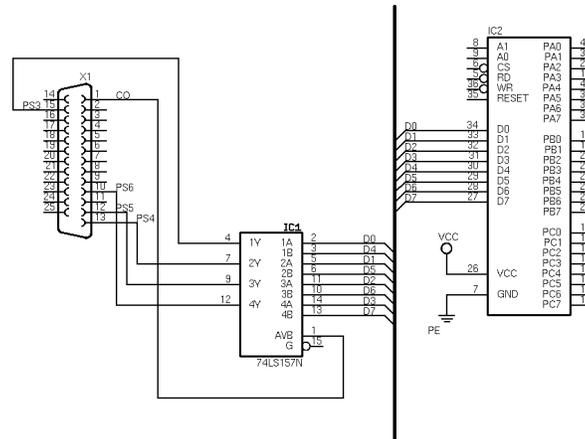
Untuk mengendalikan IC 74LS244 dan IC 74LS373 yaitu untuk mengendali apakah data ingin diteruskan jalur data atau jalur alamat PPI 8255, digunakan port kontrol 1 (C1). Gambar 1 berikut merupakan skematik penggunaan port data pada port paralel sebagai jalur data dan alamat pada PPI 8255



Gambar 1. Skematik Penggunaan Port Data pada Port Paralel sebagai Jalur Data dan Alamat pada PPI 8255

b. Perancangan Penggunaan Port Status sebagai Penerima Masukan dari port PPI 8255

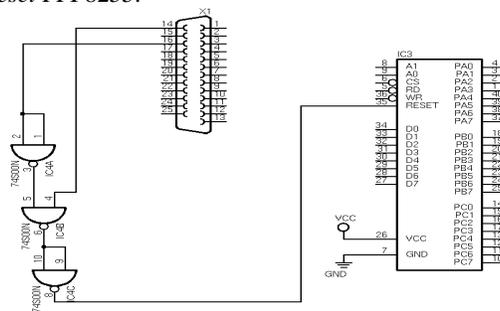
Port status pada port paralel hanya memiliki 4 pin, maka untuk mengambil 8 masukan dari port PPI 8255 dapat digunakan mode nibel yaitu dengan menggunakan IC 74LS157. IC ini merupakan IC *Quad 2 to 1 multiplexer* untuk membaca data 4 bit (nibel) dari data 8 bit, kemudian membaca lagi 4 bit berikutnya. Melalui perangkat lunak kedua data nibel tersebut disatukan menjadi data 8 bit. Untuk mengendalikan IC ini digunakan port kontrol 1 (C1) dan port kontrol 0 (C0). Berikut merupakan Gambar 20 skematik port status sebagai penerima masukan dari port PPI 8255.



Gambar 2. Skematik Penggunaan *Port Status* sebagai Penerima Masukan dari *Port PPI 8255* Menggunakan Mode Nibel

c. Perancangan Kendali *Reset PPI 8255* melalui *Port Kontrol* pada *Port Paralel*

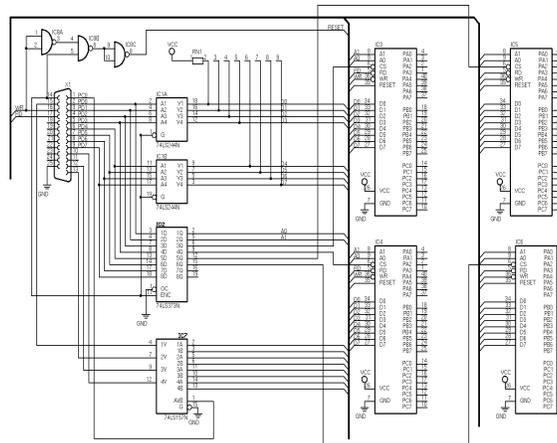
Karena keterbatasan pada *port kontrol port* paralel maka untuk mengendalikan reset PPI 8255 menggunakan *port kontrol 2 (C2)* dan *port kontrol 1 (C1)* yang berfungsi ganda. C2 juga digunakan sebagai sinyal WR sedangkan C1 digunakan untuk mengendalikan IC 74LS244. Untuk mengendalikan reset PPI 8255 dipilih kondisi C2 dan C1 yang tidak gunakan pada operasi apapun. Gambar 3 berikut merupakan skematik pengendalian *reset PPI 8255*.



Gambar 3. Skematik Pengendalian reset PPI 8255

d. Perancangan Sistem Pengendalian PPI 8255 dengan *Port Paralel PC*

Berikut ini merupakan sistem pengendalian PPI 8255 dengan *Port Paralel komputer pribadi*, yang menggunakan penggunaan jalur data dan alamat bersama dan mode nibel untuk mengambil masukkan, yang beroperasi pada mode 0. Gambar 4 berikut merupakan skematik pengendalian 4 PPI 8255 dengan *port* paralel komputer pribadi.



Gambar 4. Skematik pengendalian 4 PPI dengan *port* paralel komputer pribadi

2. Perancangan Program

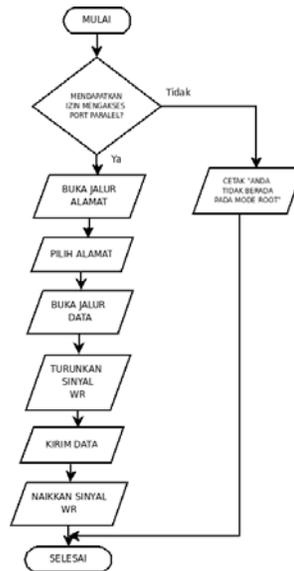
Pada perancangan program ini ada beberapa tahap pelaksanaan yaitu:

- Perancangan fungsi-fungsi operasi dasar PPI 8255 pada mode nol yaitu keluaran, masukan, dan *reset* menggunakan bahasa C dan dikompilasi dengan GCC.
- Pembuatan *shared library* yang merupakan kumpulan-kumpulan fungsi-fungsi operasi dasar PPI 8255 pada mode 0 yang telah dibuat sebelumnya.
- Pembuatan program dengan bahasa program GAMBAS dengan menggunakan fungsi – fungsi yang telah dibuat yang terdapat pada *shared library* yang dideklarasikan pada program.

a. Perancangan Fungsi Keluaran ke Port PPI, Masukkan dari Port PPI dan Reset PPI

1) Fungsi Keluaran ke Port PPI 8255

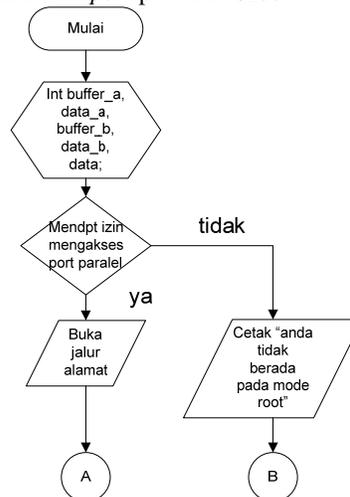
Flowchart fungsi program keluaran (*output*) yang dibuat dengan bahasa C ini didasarkan pada *timing waveform* proses mengeluarkan data ke *port* pada PPI 8255 mode 0. Gambar 5 berikut ini merupakan *flowchart* fungsi keluaran ke *port* pada PPI 8255.



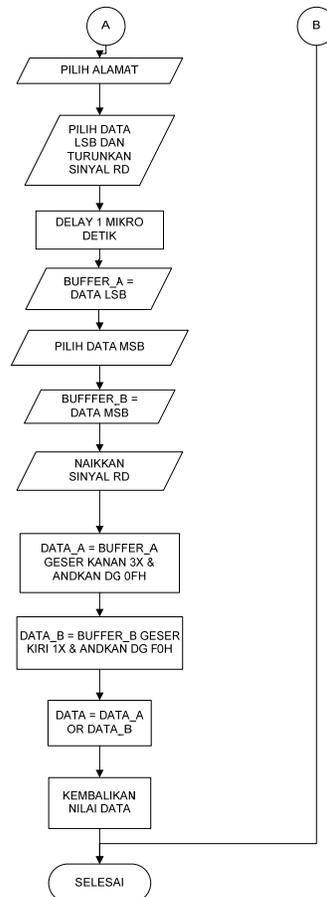
Gambar 5. Flowchart Program Fungsi Keluar Port PPI 8255

2) Perancangan Fungsi Masukan dari Port PPI 8255

Flowchart fungsi program masukan (*input*) yang dibuat dengan bahasa C ini didasarkan pada *timing waveform* proses mengambil data dari *port* pada PPI 8255 mode 0. Gambar 6 dan 7 berikut ini adalah *flowchart* fungsi masukan dari *port* pada PPI 8255.



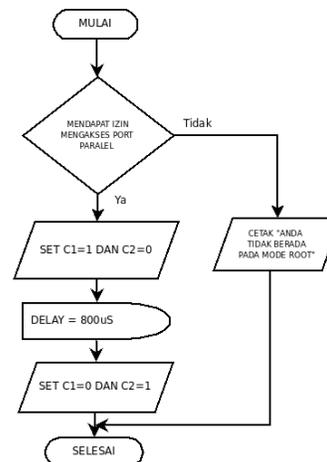
Gambar 6. Flowchart – I Fungsi Masukan dari Port PPI 8255.



Gambar 7. Flowchart – II Fungsi Masukan dari Port PPI 8255.

3) Fungsi Program *Reset PPI 8255*

Fungsi program reset merupakan kondisi saat C1 dan C2 yaitu 1 dan 0, *Flowchart* programnya dapat dilihat pada Gambar 8.



Gambar 8. Flowchart Fungsi Reset PPI 8255

b. Pembuatan *Shared Library*

Agar fungsi-fungsi yang dapat dibuat dengan bahasa C diatas dapat digunakan oleh bahasa pemrograman lain seperti misalnya gambas maka fungsi-fungsi tersebut disatukan menjadi sebuah *shared library*. Tujuannya adalah agar fungsi-fungsi tersebut dapat dipanggil oleh bahasa pemrograman lain, misalnya gambas dimana gambas tidak mempunyai fungsi-fungsi tersebut secara *original*.

Untuk membuat *shared library* maka digunakan *tools* berupa *compiler* GCC (GNU *Collection Compiler*). Untuk membuat *shared library* langkah-langkahnya yaitu sebagai berikut:

- 1) Program bahasa C yang sudah dibuat di compile untuk dijadikan *file* objek, *file* objek ini nantinya akan dikumpulkan dalam *shared library*. Perintahnya yaitu:
\$ gcc -c -O -fpic keluar_ppi.c masuk_ppi.c reset.c
- 2) Dari langkah no .1 dihasilkan *file* objek; keluar_ppi.o, masuk_ppi.o, dan reset.o. *File* objek ini akan disatukan menjadi *file shared library* dengan perintah dibawah ini:
\$ gcc -shared keluar_ppi.o masuk_ppi.o reset.o -o libppi.so
- 3) Kemudian dari langkah no. 2 akan dihasilkan *file shared library* lib_ppi.so. Untuk menggunakannya, *copy*-kan *file* .so ini ke sistem libaray yaitu di path /lib
- 4) dan dideklarasikan pada bahasa pemrograman gambah kemudian fungsi-fungsi tersebut dapat panggil dan digunakan.

c. Pembuatan Program dengan Bahasa Pemrograman Gambah

Untuk menggunakan fungsi-fungsi yang terdapat dalam *file shared library* maka *shared library* tersebut terlebih dahulu dideklarasikan dalam bahasa pemrograman Gambah. Berikut merupakan potongan program deklarasi *shared library* di Gambah

```
EXTERN keluar_ppi(data AS Integer, alamat AS Integer) AS Integer IN "libppi"
EXTERN reset() AS Integer IN "libppi"
EXTERN masuk_ppi(alamat AS Integer) AS Integer IN "libppi"
```

kemudian fungsi-fungsi yang telah dideklarasikan siap untuk digunakan. Berikut ini adalah pemanggilan fungsi-fungsi dalam *shared library* di Gambah

```
PUBLIC SUB Button2_Click()
    keluar_ppi(&H0F, &Hfa)
END
PUBLIC SUB Button7_Click()
    text_input.Text = masuk_ppi(&Hf9)
END
PUBLIC SUB Button4_Click()
    reset
END
```

d. Menentukan Alamat PPI 8255 dalam Sistem Pengendalian PPI 8255 dengan Port Paralel

Pada penelitian ini, tiap IC PPI 8255 yang terdapat pada sistem pengendalian PPI 8255 dengan *port* paralel PC memiliki alamat tertentu, dimana tiap IC PPI 8255 memiliki 4 alamat yaitu: Alamat *Port A*, *Port B*, *Port C* dan *Control Word*. Alamat tiap IC PPI 8255 ditentukan oleh hubungan CS (*Chip Select*) ke jalur alamat yaitu keluaran IC 74LS373.

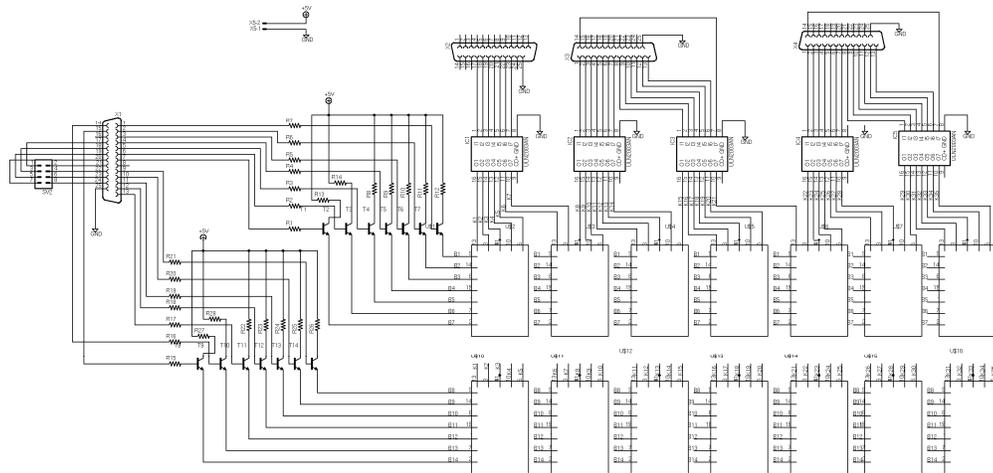
Dengan berdasarkan rancangan rangkaian sistem pengendalian PPI 8255 dengan *port* paralel, maka alamat-alamat PPI 8255 dapat diberikan pada Tabel 1 berikut ini.

Tabel 1. Daftar Alamat PPI 8255

Identitas	A0	A1	CS	Alamat Dasar
PPI 8255 I	D0	D1	D3	F4h
PPI 8255 II	D0	D1	D4	ECh
PPI 8255 III	D0	D1	D2	F8h
PPI 8255 IV	D0	D1	D5	DCh

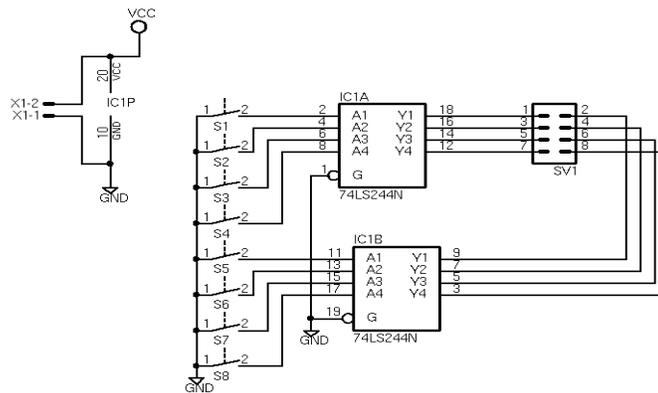
e. Perancangan Aplikasi LED Dot Matriks dan Masukan Push button untuk Pengujian I/O Sistem Pengendalian 4 PPI 8255 dengan Port Paralel

Untuk menguji keluaran dan masukan sistem pengendalian 4 PPI 8255 dengan *port* paralel PC, yaitu dengan mengendalikan led dot matriks dan menerima masukan dari *push button* yang memberikan logika terhadap *port* PPI 8255. Gambar 9 berikut adalah skematik dari led matriks yang akan dikendalikan oleh PPI 8255.



Gambar 9. Skematik penggerak Led dot Matriks terhubung ke 4 PPI 8255

Untuk menguji masukan digunakan *push button* yang dihubungkan dengan IC 74LS244. Gambar 10 berikut adalah skematik *push button* yang dihubungkan ke IC 74LS244.



Gambar 10. Skematik rangkaian masukan push button

Hasil Penelitian

1. Hasil Pengujian Pengiriman Data Keluaran ke Port PPI 8255 I

Berdasarkan data hasil pengujian pada Tabel 2 dapat disimpulkan bahwa pengiriman data keluaran ke PPI 8255 I berkerja dengan baik karena sesuai dengan kriteria pengujiannya.

Tabel 2. Hasil Pengujian Pengiriman Data ke Port PPI 8255 I

Port	Data yang dikirim dalam Desimal	Data pada Port dalam Biner	Data pada Port dalam Desimal	Tegangan Logika
Port A	0	00000000	0	0,1 V
	255	11111111	255	4,8 V
Port B	0	00000000	0	0,1V
	255	11111111	255	4,8 V
Port C	0	00000000	0	0,1
	255	11111111	255	4,8 V

2. Hasil Pengujian Rangkaian Reset

Berdasarkan data hasil pengujian pada Tabel 3 dapat disimpulkan bahwa rangkaian *reset* dapat berfungsi dengan baik karena sesuai dengan kriteria pengujiannya.

Tabel 3. Hasil Pengujian Rangkaian *Reset*

Logika Port kontrol 1	Logika Port Kontrol 2	Logika keluaran rangkaian reset
Rendah	Rendah	Tinggi
Tinggi	Rendah	Rendah
Rendah	Tinggi	Rendah
Tinggi	Tinggi	Rendah

3. Hasil Pengujian Implementasi pada Rangkaian Led Dot Matriks

Berdasarkan data hasil pengujian pada Tabel 4 dapat disimpulkan bahwa Implementasi sistem Pengendalian 4 buah PPI 8255 pada led dot matriks dan masukan push button dapat berkerja dengan baik karena sesuai dengan kriteria pengujiannya.

Tabel 4. Hasil pengujian rangkaian led matriks

Logika <i>port</i> PPI 8255 pada Baris rangkaian	Logika <i>port</i> PPI 8255 pada Kolom rangkaian	Kondisi Led (Mennyal / Padam)	Arus pada LED (mA)
Rendah	Rendah	Padam	0
Rendah	Tinggi	Padam	0
Tinggi	Rendah	Padam	0
Tinggi	Tinggi	Menyal	22

4. Hasil Pengujian Implementasi pada Rangkaian Push Button

Berdasarkan data hasil pengujian pada Tabel 5 dapat disimpulkan bahwa Implementasi sistem Pengendalian 4 buah PPI 8255 masukan *push button* dapat berkerja dengan baik karena sesuai dengan kriteria pengujiannya.

Tabel 5. Hasil Pengujian Masukan Push Button

Kondisi push button	Tegangan Keluaran IC 74LS244	Nilai Pembacaan Oleh Program
Ditekan	0,2 Volt	0
Tidak di tekan	3,6 Volt	1

Kesimpulan

Port paralel dapat digunakan untuk mengendalikan 4 buah PPI 8255, untuk digunakan sebagai *peripheral* antarmuka serba guna. Metode penggunaan jalur alamat dan data bersama pada sistem pengendalian 4 buah PPI 8255, dapat digunakan untuk memperbanyak jumlah I/O komputer sehingga dapat mengendalikan banyak peralatan. Dan juga dapat digunakan untuk mengendalikan PPI 8255 yang selama ini menggunakan slot ISA PC yang mulai menghilang pada *motherboard* sekarang.

Lingkungan Linux dapat dijadikan alternatif lingkungan sistem operasi yang *Open Source* untuk melakukan pemrograman antarmuka *port* paralel. Dengan GCC (*GNU Compiler Collection*) yang merupakan kompilator yang dapat digunakan untuk mengkompilasi program bahasa C dan membuat *shared library* dari program fungsi bahasa C, yang kemudian *file library* tersebut dapat digunakan oleh bahasa pemrograman lain misalkan gamba.

Gamba merupakan bahasa pemrograman visual yang berjalan di lingkungan Linux dapat juga digunakan sebagai bahasa pemrograman untuk mengakses *port* paralel, yang umumnya sekarang ini menggunakan Microsoft Visual Basic. Penggunaan Gamba dapat dijadikan alternatif pemrograman antarmuka *port* paralel PC di lingkungan *freeware* dan *Open Source*.

Daftar Pustaka

- Axelson, Jan, (1996), "*Parallel Port Complete*", Lakeview Research, Winnebago St. Madison
- Eko Putra, Agfianto, (2002), "*Teknik Antarmuka Komputer: Konsep dan Aplikasi*", Graha Ilmu, Yogyakarta
- Griffith, Arthur, (2002), "*GCC: The Complete and Reference*", McGraw-Hill, United States of America
- Kuo, Benjamin C, (1998), "*Teknik Kontrol Automatik*", Prehallindo, Jakarta
- Rusmanto & Yunianto, (2005), "*Menguasai Gamba Pemrograman "Visual Basic" di Linux*", Dian Rakyat, Jakarta
- Saikkonen, Riku, (1995), "*Linux How To Document*"
- Schildt, Herbert, (2000), "*C The Complete Reference*", McGraw- Hill, United States of America