

Terminal Port Komputer sebagai Perantara Pemrograman Bahasa Tingkat Tinggi

Eddy Nurraharjo

Program Studi Teknik Informatika, Universitas Stikubank

email : eddynurraharjo@gmail.com

Abstrak

Perkembangan teknologi dalam teknologi terminal perantara serta pengembangan teknik pendeteksian besaran elektronis atau fisis dengan pemrograman bahasa tingkat untuk piranti lunak telah banyak dilakukan dan dalam hal ini hampir segenap implementasinya senantiasa mengacu pada perantara sistem pengamatan akuisisi data, khususnya pada sistem akuisisi data dalam operasi kendali terpadu terapan digital mikro dengan menggunakan perantara mikrokontroler sebagai pengubah bentuk data secara visual dari sebuah sensor atau elemen transducer sebagai mata rantai paling ujung bagi sistem kendali. Untuk implementasi dalam sistem yang melibatkan bahasa tingkat tinggi, dalam hal ini menggunakan Visual Basic maka diperlukan adanya metode perantaraannya untuk digunakan menjadi penentu sekaligus mediator bagi pemrogramannya, dalam hal ini menggunakan VB. Untuk itu pada kesempatan penulisan kali ini kami penulis akan melakukan analisa kinerja terminal port komputer sebagai perantara pemrograman bahasa tingkat tinggi dalam hal ini visual basic. Konsep rancang bangun ini yang menjadi dasar pemikiran karena keunikannya dalam memberikan keluaran 8 bit data olahan dan kemampuannya dalam mode *free running or controlled running*, sehingga diharapkan mampu mendeteksi lebih jauh besaran elektronis maupun fisis yang akan diamati.

Kata kunci : port, data, VB, komputasi

PENDAHULUAN

Dunia perkembangan dan pemanfaatan teknologi elektronika khususnya dalam pengembangan teknologi kendali mikro dalam hal ini penggunaan mikrokontroler, yang dikemas dalam sebuah keping rangkaian terpadu, menambah peranan mikrokontroler sebagai pemberi warna unik dalam sistem otomasi dan memiliki keunggulan tersendiri dibandingkan dengan teknologi sekelasnya. Perkembangan ini didukung pula oleh tingkat kebutuhan bagi industri untuk menggantikan sistem analog yang dimilikinya dengan sistem yang dianggapnya baru guna memberikan perbaikan atas kekurangan sistem analog. Produsen mikrokontroler pun telah bermunculan serta berupaya meningkatkan produksi.

Bagian dari sebuah sistem digital tak dapat sepenuhnya lepas dari sistem analog, hal ini dibuktikan dengan penggunaan aplikasi elektronika dengan perbedaan konsumsi daya

yang besar dan bersifat periodik masih menggunakan komponen-komponen klasik seperti transistor (FET/MOSFET/JFET). Dan berbagai kekurangan atas sistem elektronika analog ini mengakibatkan peranan sistem digital menjadi sangat relevan dan memiliki kedudukan yang tidak dapat diabaikan dalam sistem analog. Untuk sebuah sistem digital, sistem ini mampu menangani kinerja dari ribuan bahkan hingga jutaan transistor yang memiliki sistem dengan prinsip yang lebih baik dan stabil.

Transducer sebagai mata rantai paling ujung dalam suatu sistem kendali memiliki karakteristik yang unik mulai dari proses pembuatan hingga menjadikannya sebagai komponen jadi atas suatu perangkat elektronis terpadu yang modern. Teknik dalam pendeteksian besaran elektronis maupun fisis mutlak diperlukan dan diperhatikan sebagai susunan data terpadu yang diharapkan mampu memberikan data awal olahan dalam suatu proses deteksi akurat dan presisi. Sebuah metode

untuk pemodelan yang terpadu dan sistematis dengan dasar matematis mampu memberikan data siap olah untuk selanjutnya diproses oleh komputer sebagai data baku maupun sebagai data tambahan untuk pengolahannya lebih lanjut. Perolehan data dalam bentuk digital ini memerlukan suatu "mata" yang sanggup melihat dan mengamati perubahan yang sebenarnya terjadi.

Pada kesempatan ini diperlukan sistem akuisisi data yang "akurat" dalam suatu sistem kendali terpadu sebagai pra-proses dapat memberikan validitas data yang baik, keunggulan atas sistem terpadu menjadi lebih "cerdas" diawali dengan akuisisi data analog dari sisi pengaturan sistemnya. *Keyword* utama dalam pemrosesan ini tidak lepas dari karakteristik unik dari unit pengubah atau pengkonversi besaran analog menjadi besaran digital (ADC).

Implementasi sistem digital memiliki permasalahan yang mungkin timbul adalah dengan sifat "tidak semua" sistem analog klasik mampu ditangani dengan sistem digital modern, atau dengan kata lain sistem digital hanya terbatas untuk pengganti sistem analog tertentu saja dalam berbagai faktor non teknis pun masih memerlukan pertimbangan tersendiri bagi pengguna produk digital tersebut karena perbedaan kedua sifat dan perilaku terhadap sistem utamanya.

Main unit dari sistem kendali adalah rangkaian sensor, dimana rangkaian ini menjadi ujung tombak penentu awal terhadap suatu eksekusi program atau perintah yang diharapkan bagi perancang sistem kendali. Namun peranan adanya komponen sensor itu sendiri tidak dengan serta merta dapat langsung berkomunikasi dengan mikrokontroler melainkan dengan bantuan seperangkat antarmuka yang akan menjembatani terhadap terminal masukan sistem kendali mikro (*signal conditioning*). Penyesuaian untuk beberapa mode ini diperlukan beberapa trik dan tips tersendiri sehingga linieritas sistem dapat terwujud diiringi dengan pendekatan idealnya terhadap perubahan pada terminasi masukan/perantaranya.

Laju perkembangan teknologi elektronika dan terbukanya peluang untuk rekayasa sistem aplikatif mikrokontroler, maka penulis memberikan bobot pandangan tersendiri terhadap beberapa permasalahan yang menyangkut hal pengaturan antarmuka dalam sebuah diantaranya adalah diperlukannya sebuah elemen sistem yaitu serangkaian komponen elektronika dalam wujud antarmuka universal. Sebagai perbandingan saja, beberapa permasalahan yang tidak kalah pentingnya dalam rancang bangun sistem kendali mikrokontroler adalah pada rangkaian penyesuaian sinyal yang dihasilkan dari sensor atau biasa disebut rangkaian pengkondisi sinyal (*signal conditioning circuit*), disamping teknik konversi terhadap besaran analog menjadi besaran digital. Untuk itu diperlukan pengamatan dan pengembangan konsep, guna memberikan pendekatan terbaik terhadap perubahan data sesuai dengan variabel yang diperlukan oleh sistem kendali dalam memberikan keputusan akhir atau eksekusi program, dalam hal ini pemrograman bahasa tingkat tinggi.

LANDASAN TEORI

Pengenalan Dasar Pemrograman Antarmuka berbasis VB

Beberapa contoh file mediasi sebagai elemen utama konsep interfacing ini adalah mempelajari beberapa model untuk mengaktifkan file pelengkap utama. Untuk bahasa pemrograman delphi yang diperlukan adalah hwinterface.ocx serta inpout32.dll, sedangkan untuk bahasa pemrograman Visual Basic yang diperlukan adalah file io.dll.

Untuk menjadi sistem operasi yang sangat aman, sejak dulu Windows NT sudah memberikan beberapa hak dan pembatasan untuk berbagai jenis program yang berjalan pada platform Windows NT tersebut. Hal ini mengklasifikasikan semua program dalam dua kategori, yaitu User mode dan modus Kernel, berjalan di ring3 dan ring0 mode.

Program modus pengguna yang berjalan dalam modus ring3 dan program modus Kernel berjalan dalam mode ring0. Program biasanya Anda menulis jatuh dalam kategori mode pengguna. Program modus user dibatasi untuk

menggunakan instruksi tertentu seperti IN, OUT dll. Setiap kali sistem operasi menemukan bahwa program modus pengguna mencoba untuk menjalankan instruksi tersebut, maka sistem operasi berhenti pelaksanaan program-program dan akan menampilkan pesan kesalahan.

Akhirnya program perlu interfacing untuk bisa berhenti di mana mereka melaksanakan IN atau OUT sebagai instruksi untuk membaca atau menulis data ke port paralel. Tapi dalam program yang sama modus Kernel waktu yang sama sekali tidak dibatasi dalam melaksanakan instruksi tersebut.

Device driver mampu berjalan dalam mode kernel. Jadi bekerja dengan menggunakan file interfacing ini, berhadapan dengan beberapa permasalahan konektifitas. Point utama untuk masalah tersebut di atas adalah untuk menulis sebuah driver kernel mode yang mampu membaca dan menulis data ke port paralel dan membiarkan program mode pengguna untuk berkomunikasi dengannya.

Proses tulis yang akan menuliskan kode driver merupakan pekerjaan yang tidak mudah bagi programmer bahkan mengalami beberapa permasalahan. Tapi proses tulis driver sederhana untuk berkomunikasi dengan port paralel adalah penugasan yang sederhana ketika driver seperti USB, kartu suara dll. Meskipun Anda mendapatkan driver untuk dapat bekerja pada port terminal lain, instalasi dan konfigurasinya akan menjadi pekerjaan yang sangat rumit.

Solusi

Mengenalkan `inpout32.dll` untuk WIN 98/NT/2000/XP. Dll (library) ini memiliki fitur berikut :

- Dapat diperasikan pada semua versi windows (WIN 98, NT, 200 dan XP).
- Menggunakan sebuah *kernel mode driver* yang tertanam dalam dll (library-nya).
- Tidak perlu software (piranti lunak) atau instalasi driver khusus yang dibutuhkan
- Driver akan otomatis di-*install* dan dikonfigurasi secara otomatis ketika dll di-*load*.

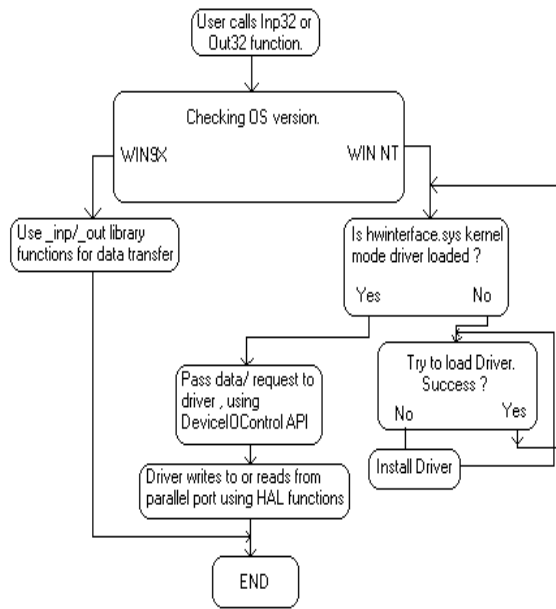
- Tidak memiliki API khusus yang dibutuhkan karena hanya memiliki dua fungsi yaitu `Inp32` dan `Out32`
- Dapat dengan mudah digunakan dengan Visual C++ dan Visual Basic

Memiliki fungsi-fungsi yang kompatibel dengan Jan Axelons `inpout32.dll` (yang tersedia di www.lvr.com). Jadi dll (library) ini dapat digunakan dengan contoh program yang tersedia dengan buku Port Paralel Lengkap, tanpa perlu beberapa modifikasi apapun.

Cara kerjanya

Fitur luar biasa dari library `inpout32.dll` ini adalah, dapat bekerja dengan semua versi Windows tanpa modifikasi dalam kode pengguna atau DLL sendiri. Artikel ini selanjutnya akan menjelaskan bagaimana harapan itu bisa tercapai, apa metode pemrograman yang digunakan, apa API yang digunakan, dan lain sebagainya ...

Dll (library) akan memeriksa versi sistem operasi ketika suatu fungsi dipanggil, dan jika sistem operasi Win9x, DLL akan menggunakan `_inp()` dan fungsi `_outp` untuk melakukan proses membaca / menulis port paralel. Di sisi lain, jika sistem operasi adalah WIN NT, 2000 atau XP, maka akan menginstal *kernel mode driver* dan memberitahukan kepada port paralel melalui driver tersebut. Kode dari pengguna secara tidak langsung dari versi OS akan dijalankan. DLL (library) ini dapat digunakan dalam WIN NT maupun sistem operasi yang serupa itu adalah Win9x. Alur dari program ini diberikan di bawah ini.



Gambar 1 Alur program

Sumber : logic4u.net

Dua blok bangunan penting dari program ini adalah

- a. Sebuah perangkat driver kernel mode tertanam dalam DLL dalam bentuk kode bilangan biner
- b. DLL (library) itu sendiri

Kernel mode driver Hwinterface.sys

Kode *source* dari Hwinterface.sys driver kernel mode terletak di direktori "kernel_mode_driver_source". Di mana "hwinterfacedrv.c" adalah aplikasi file source utama.

Tiga fungsi yang diimplementasikan pada driver adalah sebagai berikut :

- a. DriverEntry(), dipanggil ketika *driver* dimuat (*loaded*). Membuat objek perangkat dan link simbolik.
- b. hwinterfaceUnload (), dipanggil ketika *driver* dilepaskan, dan akan melakukan membersihkannya.
- c. hwinterfaceDeviceControl (), menangani panggilan yang dilakukan melalui DeviceIoControl API. Melakukan baca tulis

ke port paralel sesuai dengan kode kontrol yang melaluinya.

DLL Inpout32

Fungsi dalam DLL diimplementasikan dalam dua file sumber, "inpout32drv.cpp" dan "osversion.cpp". osversion.cpp akan senantiasa memeriksa versi sistem operasi. "Inpout32drv.cpp" akan menginstal driver kernel mode, memuatnya, menulis / membaca port paralel dan lain sebagainya.

Dua fungsi yang dikeluarkan dari inpout32.dll adalah :

1. Inp32(), membaca data dari register paralel port tertentu.
2. Out32(), menuliskan data ke port paralel yang telah ditentukan oleh register.

Adapun fungsi lain yang dapat diimplementasikan dalam inpout32.dll adalah

- a.DllMain(), dipanggil ketika dll (library) dimuat atau dilepaskan. Ketika dll (library) dimuat, maka ia akan mengecek versi OS dan memuat hwinterface.sys, seandainya diperlukan.
- b. Closedriver(), menangani penutupan *driver* yang sdang digunakan/dibuka, dan dipanggil sebelum melepas *driver*.
- c. Opendriver(), menangani pembukaan untuk driver hwinterface.
- d. inst(), akan melakukan proses ekstraksi 'hwinterface.sys' dari sumber binernya ke direktori 'systemroot \ driver' dan menciptakan layanan baru. Fungsi ini dipanggil saat fungsi 'Opendriver' gagal untuk membuka sebuah penanganan yang valid kepada layanan 'hwinterface'.
- e. start(), memulai layanan hwinterface menggunakan Service Control Manager API.
- f. SystemVersion(), mengecek versi OS dan mengembalikannya dengan kode yang sesuai.

Penjelasan kontrol hwinterface.ocx ActiveX

Ini adalah kontrol ActiveX dengan fitur yang sama dengan inpout32.dll. Hal ini dapat digunakan baik dengan Visual C++ atau Visual Basic. Hal itu akan memberikan kenyamanan besar bila digunakan dengan VB. Data dapat dituliskan ("read") ke port paralel menggunakan metode Outport dan dapat dibaca dengan menggunakan metode Inport.

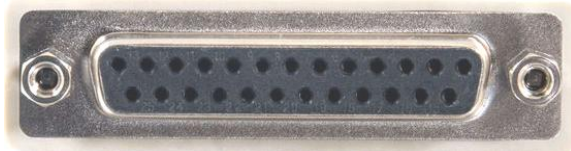
Terminasi/Port Komputer

Tabel 1 Koneksi pin DB25

Pin	Description	Direction
1	Strobe	In/Out
1 to 9	Data	In/Out
10	Busy	In
11	Out of Paper	In
12	Select	In
13	Auto feed	In
14	Error	In/Out
15	Dados	In
16	Select In	In/Out
17	Init	In/Out
18 to 25	Ground	-

Tabel 2 Register dan Alamat

Register	LPT1 Address
Data Register	Base Address (378H) + 0
Status Register	Base Address (378H) + 1
Control Register	Base Address (378H) + 2

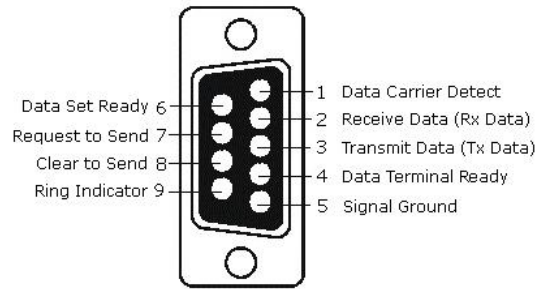


Gambar 2 Fisik DB25

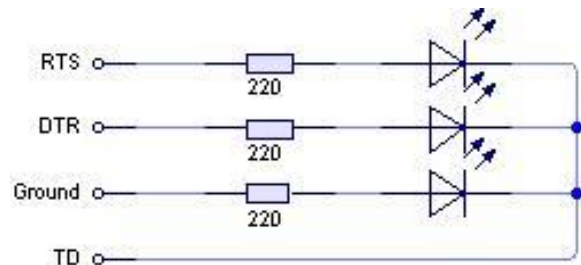
Untuk komputer sendiri memiliki beberapa port / terminal penghubung dengan user maupun komputer lain dalam upaya untuk saling berinteraksi dan komunikasi data diantara mereka, yang dapat dilakukan dengan melalui

port serial, paralel, I/O maupun port USB, yang lebih Universal seperti sifatnya Universal Serial Bus.

Pada prinsipnya pengamatan port paralel/seriap pada komputer dapat dilakukan sehubungan dengan ketersediaan register-register yang dimiliki oleh mikroprosesor. Karakteristik port paralel maupun serial dibandingkan dengan port USB memang berbeda baik dilihat dari kegunaanya, maupun bentuk dan kecepatan transernya.



Gambar 3 Port serial DB-9



Gambar 4 Koneksi beban port serial

Banyak cara untuk mengakses port serial ini, antara lain dengan menggunakan visual basic 6.0. Salah satu cara mengakses port serial menggunakan Visual Basic adalah melalui komponen MSCOMM.

Cara mengaktifkan MSComm yaitu dari VB6

- a. pilih Select Project
- b. Components
- c. Pada bagian Controls, Centang Microsoft Comm Control 6.0
- d. Tekan OK.

Pada bagian Toolbox akan ada gambar / simbol Telpon.

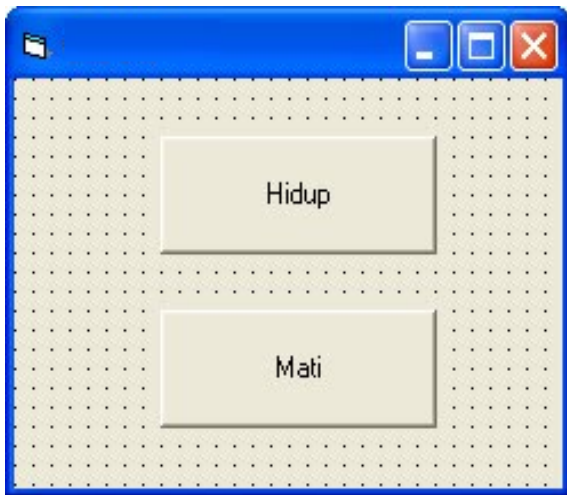
PEMBAHASAN

Beberapa Prinsip dalam Proses Perantaraan Program

Program Visual Basic dengan hardware melalui Paralel Port dapat dilakukan dengan beberapa cara diantaranya adalah :

- a. Interfacing via Printer Port
- b. Interfacing melalui card I/O IC8255

Untuk mengaktifkan file ini kita terlebih dahulu harus mendeklarasikan pada sebuah form atau dapat juga dilakukan pada media jendela kodingnya, berupa dua buah tombol yaitu untuk menyalakan dan mematikan port.



Gambar 5 Form

Form tes untuk pengujian dilakukan dengan sederhana saja yang terdiri dari 2 tombol

Tabel 3 Properties

Kontrol	Properti	Setting
Command1	Name	On
	Caption	Hidup
Command2	Name	Off
	Caption	Mati

Modul yang diperlukan :

- 1. Mengaktifkan Program Visual Basic (dengan catatan program ini sudah di install di Komputer) dengan cara klik

start>all program>microsoft visual studio 6.0>Microsoft visual basic 6.0.

- 2. Pilih standart exe kemudian klik ok.
- 3. Ada baiknya terlebih dahulu untuk membuat sebuah module dengan cara klik project>add module.
- 4. Kemudian berikan masukan deklarasi seperti ini:

```
Public Declare Function Inp Lib "inpout32.dll" _ Alias "Inp32" (ByVal PortAddress As Integer) As Integer
```

```
Public Declare Sub Out Lib "inpout32.dll" _ Alias "Out32" (ByVal PortAddress As Integer, ByVal Value As Integer)
```

- 5. Setelah itu dilanjutkan dengan merancang program yang diinginkan.
- 6. Kemudian tambahkan dua buah *command* klik pada form yang sudah ada.
- 7. Mengatur propertis dari masing-masing *command* (sesuai keinginan/kebutuhan) seperti tabel yang telah dibuat.
- 8. Memberikan koding masukan untuk *user interface* dengan meng-klik dua kali tombol "on" maka akan muncul jendela *source* kode kemudian mengetikkan listing berikut.

```
Out 888, 255
```

- 9. Memberikan koding masukan untuk *user interface* dengan meng-klik dua kali tombol "off" dan juga harus mengetikkan

```
Out 888,0
```

```
Private Sub on_Click()
Out 888, 255
End Sub
```

```
Private Sub off_Click()
Out 888, 0
End Sub
```

Adapun model pemrograman lanjutan adalah penggunaan listing deklarasi sebagai berikut, kemudian menentukan fungsi dari tiap deklarasi yang dilakukan untuk memperoleh alur sistem/algorithm dan pemrograman yang diperlukannya :

```
Public Declare Sub PortOut Lib
"io.dll" _
    (ByVal Port As Integer, ByVal
Data As Byte)
```

- **PortOut** akan mengirimkan data dalam format byte (8-bit).

```
Public Declare Sub PortWordOut
Lib "io.dll" _
    (ByVal Port As Integer, ByVal
Data As Integer)
```

- **PortWordOut** akan mengirimkan data dalam format word (16-bit).

```
Public Declare Sub PortDWordOut
Lib "io.dll" _
    (ByVal Port As Integer, ByVal
Data As Long)
```

- **PortDWordOut** akan mengirimkan data dalam format double word (32-bit).

```
Public Declare Function PortIn
Lib "io.dll" _
    (ByVal Port As Integer) As Byte
```

- **PortIn** akan membaca data dalam format byte (8-bit).

```
Public Declare Function
PortWordIn Lib "io.dll" _
    (ByVal Port As Integer) As
Integer
```

- **PortWordIn** akan membaca data dalam format word (16-bit).

```
Public Declare Function
PortDWordIn Lib "io.dll" _
    (ByVal Port As Integer) As Long
```

- **PortDWordIn** akan membaca data dalam format double word (32-bit).

```
Public Declare Sub SetPortBit
Lib "io.dll" _
    (ByVal Port As Integer, ByVal
Bit As Byte)
```

- **SetPortBit** akan men-Set bit (=1)

```
Public Declare Sub ClrPortBit
Lib "io.dll" _
    (ByVal Port As Integer, ByVal
Bit As Byte)
```

- **ClrPortBit** akan me-Reset bit (=0).

```
Public Declare Sub NotPortBit
Lib "io.dll" _
    (ByVal Port As Integer, ByVal
Bit As Byte)
```

- **NotPortBit** akan melakukan proses inversi (NOT) bit

```
Public Declare Function
GetPortBit Lib "io.dll" _
    (ByVal Port As Integer, ByVal
Bit As Byte) As Boolean
```

- **GetPortBit** akan membacakan status dari bit.

```
Public Declare Function
RightPortShift Lib "io.dll" _
    (ByVal Port As Integer, ByVal
Val As Boolean) As Boolean
```

- **RightPortShift** akan menggeserkan bit ke kanan, LSB -> MSB.

```
Public Declare Function
LeftPortShift Lib "io.dll" _
    (ByVal Port As Integer, ByVal
Val As Boolean) As Boolean
```

- **LeftPortShift** akan menggeser bit ke kiri, MSB -> LSB.

```
Public Declare Function
IsDriverInstalled Lib "io.dll"
    () As Boolean
```

- **IsDriverInstalled** : fungsi utama dari fungsi ini adalah untuk memastikan bahwa *driver* penggerak mode kernel pada NT/2000/XP telah disiapkan dan dapat diakses/digunakan.

Jika sudah memahami konsep deklarasinya maka dapat dilakukan koneksi pada alamat-alamat port berikut :

COM1 - 3F8 (hex) → 1016

COM2 - 2F8 (hex) → 760

Selanjutnya contoh program uji adalah sebagai berikut :

```
Sub test()
Call PortOut(1016, 63)
Call PortOut(1016, 13)
```

```

For i = 1 To 100
Debug.Print PortIn(1016)
Next
End Sub

```

KESIMPULAN DAN SARAN

Hasil pengamatan yang telah dilakukan memberikan beberapa kesimpulan sebagai dasar teori dan keilmuan sebagai berikut :

1. Pemrograman perantara dengan menggunakan port parallel/serial memerlukan sebuah ADC yang dapat difungsikan untuk melakukan konversi besaran fisis menjadi besaran digital, dan dengan penambahan IC RS232 maka dapat diimplementasikan pada port parallel komputer, baik dengan menggunakan bahasa tingkat tinggi Delphi maupun visual basic.
2. Sistem akuisisi data yang baik akan dapat menentukan desain model perantara melalui beberapa port komunikasi serial maupun parallel dengan menggunakan pemrograman bahasa tingkat tinggi sebagai pengolahan visualnya seperti Visual Basic, Borland Delphi, Java dan lain sebagainya.
3. Untuk melakuka koneksi via USB diperlukan trik khusus selain dari trik pengujian diatas, dimana posisi port akan memiliki perbedaan, dan terkadang penguji melakukan proses konversi port terlebih dahulu sebelum melakukan pengamatan untuk proses akuisisi datanya.

DAFTAR PUSTAKA

- Budiarso Z., Raharjo EN., Lusiana V., (2006), Laporan Penelitian berjudul Sistem Kendali Terpadu dengan Menggunakan Metode Octal Bus Transceiver with Non Inverting 3 State Output, Semarang
- Pranowo L., (2006), Laporan Tugas Akhir berjudul Simulasi Sistem Kendali Pengaturan Kecepatan Motor Berdasarkan Perubahan Warna, Semarang
- Raharjo EN., Hanum SY., Lusiana V., Kurniawati WH, (2002), Laporan Penelitian berjudul Pendeteksi dan Penghitung Detak Jantung, Semarang
- Tocci, R. J., & Widmer, N. S. (1995). *Digital systems: Principles and applications*. Englewood Cliffs, N.J: Prentice Hall.
- Utomo, P. (2005). Laporan Tugas Akhir berjudul Pendeteksi Suhu dengan Menggunakan Mikrokontroler dan Termokopel. Semarang
- Putra, A.E. (2009). Akses Port Paralel (Menggunakan Visual BASIC 6.0). <http://agfi.staff.ugm.ac.id/blog/index.php/2009/01/akses-port-paralel-menggunakan-visual-basic-60/>
- Andiero. (n.d.). Interfacing Serial Port (LED) dengan Visual Basic(VB) 6.0. Retrieved July 27, 2012, from <http://bocah-cakil.blogspot.com/2010/01/interfacing-serial-port-led-dengan.html>
- Input32.dll & VB6 Problem in Windows XP. (n.d.). *Control.com - Forum for Automation Professionals*. Retrieved July 27, 2012, from <http://www.control.com/thread/1026190131>
- Thiagoblmeira. (n.d.). Parallel Port Control with Delphi: Parallel Port, Delphi, LPTPort, Device Driver. *Experts Exchange - Your Technology Questions Answered*. Retrieved July 27, 2012, from http://www.experts-exchange.com/Programming/Languages/Pascal/Delphi/A_2161-Parallel-Port-Control-with-Delphi.html
- Kurniawan, D. (n.d.). Komunikasi Serial dengan Delphi. Retrieved July 27, 2012, from <http://ilmukomputer.org/2008/10/01/komunikasi-serial-dengan-delphi/>
- Input32.dll for Windows 98/2000/NT/XP. (n.d.). *Logix4u*. Retrieved July 27, 2012, from http://logix4u.net/index.php?option=com_content&task=view&id=16&Itemid=1
- Serial port trend: HW Virtual Serial Port, Serial Port Terminal, Serial Port Mapper. (n.d.). *Podnova Windows Library*. Retrieved July 27, 2012, from

http://windows.podnova.com/trends/serial_port.html

Bengkelprogram. (n.d.). *bengkelprogram.com*. Retrieved July 27, 2012, from <http://bengkelprogram.com/#sharing14291ast>