



**UNIVERSITAS INDONESIA**

**RANCANG BANGUN SISTEM KONTROL *SAMPLE CHANGER*  
BERBASIS PC PADA PERANGKAT RADIOIMMUNOASSAY  
(RIA) NUCLEUS MODEL 1600 GAMMA COUNTING  
MENGUNAKAN MODUL ANTAR MUKA USB  
DEVASYS I2C I/O**

**SKRIPSI**

**HARI NURCAHYADI**

**0706199376**

**FAKULTAS TEKNIK  
DEPARTEMEN TEKNIK ELEKTRO  
DEPOK  
JULI 2009**



**UNIVERSITAS INDONESIA**

**RANCANG BANGUN SISTEM KONTROL *SAMPLE CHANGER*  
BERBASIS PC PADA PERANGKAT RADIOIMMUNOASSAY  
(RIA) NUCLEUS MODEL 1600 GAMMA COUNTING  
MENGUNAKAN MODUL ANTAR MUKA USB  
DEVASYS I2C I/O**

**SKRIPSI**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik**

**HARI NURCAHYADI**

**0706199376**

**FAKULTAS TEKNIK  
DEPARTEMEN TEKNIK ELEKTRO  
DEPOK  
JULI 2009**

## HALAMAN PERNYATAAN ORISINALITAS

**Skripsi ini adalah hasil karya saya sendiri,  
dan semua sumber baik yang dikutip maupun dirujuk  
telah saya nyatakan dengan benar**

**Nama : Hari Nurcahyadi**

**NPM : 0706199376**

**Tanda Tangan :**



**Tanggal : 6 Juli 2009**

## HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : Hari Nurcahyadi  
NPM : 0706199376  
Program Studi : Teknik Elektro  
Judul Skripsi :

RANCANG BANGUN SISTEM KONTROL *SAMPLE CHANGER* BERBASIS PC PADA PERANGKAT RADIOIMMUNOASSAY (RIA) NUCLEUS MODEL 1600 GAMMA COUNTING MENGGUNAKAN MODUL ANTAR MUKA USB DEVASYS I2C I/O

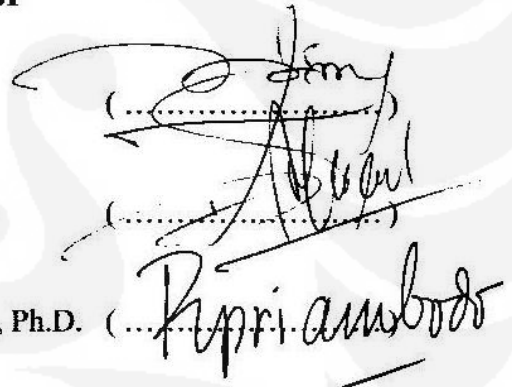
Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Elektro, Departemen Teknik Elektro, Fakultas Teknik, Universitas Indonesia

### DEWAN PENGUJI

Pembimbing : Arief Udhiarto, ST, MT

Penguji : Prof. Dr. Ir. Djoko Hartanto, M.Sc

Penguji : Ir. Purnomo Sidi Priambodo M.Sc., Ph.D.



(.....)

(.....)

(..... Priambodo)

Ditetapkan di : Depok

Tanggal : 6 Juli 2009

## KATA PENGANTAR

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka untuk melengkapi sebagian persyaratan menjadi Sarjana Teknik pada Program Studi Teknik Elektro, Departemen Teknik Elektro, Fakultas Teknik Universitas Indonesia. Saya menyadari bahwa, tanpa bantuan dan bimbingan dari pberbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi saya untuk menyelesaikan skripsi ini. Oleh karena itu, saya mengucapkan terimakasih kepada:

- (1) Bapak Arief Udhiarto, ST, MT, selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini.
- (2) Bapak Drs. Rukmono Pribadi, selaku pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini.
- (3) Orang tua, Istri tercinta, Ananda Galang tercinta, dan seluruh keluarga saya yang telah memberikan inspirasi, motivasi untuk saya.
- (4) Bapak Kepala BATAN, Bapak Kepala PUSDIKLAT – BATAN, dan Bapak Kepala PRPN – BATAN yang telah memberikan kesempatan kepada saya untuk dapat menempuh Tugas Belajar jenjang S1 di Program Studi Teknik Elektro, Departemen Teknik Elektro, Fakultas Teknik Universitas Indonesia.
- (5) Rekan-rekan di lingkungan unit kerja PRPN – BATAN, dan
- (6) Sahabat yang telah banyak membantu saya dalam menyelesaikan skripsi ini.

Akhir kata, saya berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Depok, 6 Juli 2009

Penulis

## HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

---

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Hari Nurcahyadi  
NPM : 0706199376  
Program Studi : Teknik Elektro  
Departemen : Teknik Elektro  
Fakultas : Teknik  
Jenis karya : Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty- Free Right*)** atas karya ilmiah saya yang berjudul :

” Rancang Bangun Sistem Kontrol *Sample Changer* Berbasis PC Pada Perangkat Radioimmunoassay (RIA) Nucleus Model 1600 Gamma Counting Menggunakan Modul Antar Muka USB Devasys I2C I/O ”

Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok  
Pada tanggal : 6 Juli 2009  
Yang menyatakan



( Hari Nurcahyadi )

## ABSTRAK

Nama : Hari Nurcahyadi  
Program Studi : Teknik Elektro  
Judul : Rancang Bangun Sistem Kontrol *Sample Changer* Berbasis PC  
Pada Perangkat Radioimmunoassay (RIA) Nucleus Model 1600  
Gamma Counting Menggunakan Modul Antar Muka USB  
Devasys I2C I/O

RIA Nucleus Model 1600 Gamma Counting merupakan salah satu perangkat Radioimmunoassay (RIA). Perangkat RIA Nucleus Model 1600 Gamma Counting yang terdapat di instalasi kedokteran nuklir Badan Tenaga Nuklir Nasional (BATAN) saat ini dalam kondisi sudah tidak dapat dioperasikan lagi. Untuk dapat dioperasikannya kembali perangkat RIA Nucleus Model 1600 Gamma Counting ini, diperlukan rancang bangun pada sistem kontrol pergerakan *sample changer*. Rancang bangun dilakukan pada bagian elektronik dan pada bagian perangkat lunak. Rancang bangun yang dilakukan pada bagian elektronik adalah pembuatan modul *driver* penggerak motor dan penggunaan modul antar muka *Universal Serial Bus* (USB) Devasys I2C I/O sebagai basis sistem kontrol dan sebagai media komunikasi antara perangkat sistem dengan *Personal Computer* (PC). Rancang bangun yang dilakukan pada bagian perangkat lunak adalah bagian pengendali dari sistem kontrol pergerakan *sample changer*, termasuk visualisasi kontrolnya, sehingga proses pergerakan *sample* bekerja dan berfungsi secara fleksibel dan optimal.

Kata kunci :  
Sistem kontrol, Radioimmunoassay (RIA), rancang bangun, *driver*, USB,  
*sample changer*

## ABSTRACT

Name : Hari Nurcahyadi  
Study Program : Electrical Engineering  
Judul : Design And Development Of Sample Changer Control System  
With PC Base On Radioimmunoassay (RIA) Peripheral Nucleus  
Model 1600 Gamma Counting using USB Interface Devasys  
I2C I/O Module

RIA Nucleus Model 1600 Gamma Counting is one of nuclear medicine peripheral Radioimmunoassay (RIA). Peripheral RIA Nucleus Model 1600 Gamma Counting that was available at nuclear medicine installation of National Nuclear Energy Of Indonesia (BATAN) this time in condition have no longer used. To get be run back this peripheral RIA Nucleus Model 1600 Gamma Counting, it is need to design and develop on system control of sample changer movement. Design and development system will done on electronics part and software part. On electronics part, design and development that done is to invent driver module of starting motor and using purpose Universal Serial Bus (USB) interface Devasys I2C I/O module as controls system base and as communication media among peripheral system with Personal Computer (PC). On software part, design and development that done is on controller sectioned of sample changer movement control systems, including it's control visualisation, so sample movement process can work and function optimally and flexible.

Key words:

Control system, Radioimmunoassay (RIA), design and development, driver, USB, sample changer



## DAFTAR ISI

HALAMAN JUDUL .....	i
HALAMAN PERNYATAAN ORISINALITAS.....	ii
HALAMAN PENGESAHAN.....	iii
KATA PENGANTAR .....	iv
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS.....	v
ABSTRAK .....	vi
DAFTAR ISI .....	viii
DAFTAR GAMBAR .....	x
DAFTAR TABEL .....	xii
DAFTAR LAMPIRAN .....	xiii
<b>1. PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Perumusan Masalah .....	2
1.3 Tujuan .....	2
1.4 Batasan Masalah .....	2
1.5 Metodologi .....	3
1.6 Sistematika Penulisan .....	4
<b>2. TEORI DASAR .....</b>	<b>6</b>
2.1 Perangkat Radioimmunoassay (RIA).....	6
2.2 Teknologi USB ( <i>Universal Serial Bus</i> ).....	8
2.3 Modul Antar Muka USB Devasys I2C I/O .....	10
2.4 PCF 8574 8-Bit I/O <i>Expander</i> .....	13
2.4.1 Konfigurasi PIN IC PCF 8574 .....	13
2.4.2 Pengalamatan IC PCF 8574 .....	14
2.4.3 Mode Tulis ( <i>Output</i> ) Pada IC PCF 8574 .....	15
2.4.4 Mode Baca ( <i>Input</i> ) Pada IC PCF 8574 .....	16
2.5 Motor Listrik .....	16
2.6 <i>Limit Switch</i> .....	19
2.7 Relay .....	21
2.8 Transistor Sebagai Saklar .....	22
<b>3. RANCANG BANGUN SISTEM.....</b>	<b>26</b>
3.1 Prinsip Kerja Sistem .....	26
3.2 Perangkat Keras .....	28
3.2.1 <i>Personal Computer</i> (PC) .....	28
3.2.2 <i>Interface</i> USB .....	28
3.2.2.1 <i>Interkoneksi</i> Port B .....	29
3.2.2.2 <i>Interkoneksi</i> Port C .....	29
3.2.2.3 <i>Interkoneksi</i> Port I2C I/O .....	29
3.2.3 <i>Driver</i> Motor .....	30
3.2.4 Perangkat Mekanik <i>Sample Changer</i> .....	31
3.2.5 <i>Power Supply</i> .....	33
3.3 Perangkat Lunak .....	34

3.3.1 Pemrograman Gerak Motor Horizontal .....	34
3.3.2 Pemrograman Gerak Motor Vertikal .....	35
3.3.3 Pemrograman Bagian Utama Sistem .....	37
<b>4. PENGUJIAN DAN ANALISA SISTEM.....</b>	<b>39</b>
4.1 Pengujian Modul Elektronik .....	39
4.1.1 Pengujian Modul <i>Interface</i> USB Devasys I2C I/O .....	39
4.1.2 Pengujian Modul <i>Driver</i> Motor .....	40
4.2 Pengujian Sistem Keseluruhan .....	43
4.2.1 Pengujian Gerak Posisi Kolimator <i>Detector</i> .....	45
4.2.2 Pengujian Proses <i>Counting Sample</i> Dengan perintah <i>Counting Current Sample</i> .....	46
4.2.3 Pengujian Proses <i>Counting Sample</i> Dengan perintah <i>Counting 25 Sample</i> .....	47
4.2.4 Pengujian Proses <i>Counting Sample</i> Dengan perintah <i>Counting All Sample (100 Sample)</i> .....	48
4.2.5 Pengujian Proses <i>Counting Sample</i> Pada Nomor <i>Sample</i> Tertentu .....	49
4.3 Analisa Sistem .....	50
<b>5. KESIMPULAN .....</b>	<b>53</b>
<b>DAFTAR ACUAN .....</b>	<b>54</b>
<b>DAFTAR REFERENSI .....</b>	<b>56</b>

## DAFTAR GAMBAR

Gambar 2.1	(a) Konektor tipe A ( <i>upstream</i> ).....	9
Gambar 2.1	(b) Konektor tipe B ( <i>downstream</i> ).....	9
Gambar 2.2	Konfigurasi PIN konektor tipe A ( <i>upstream</i> ) dan konektor tipe B ( <i>downstream</i> ).....	9
Gambar 2.3	Modul Antarmuka USB Devasys I2C/IO.....	11
Gambar 2.4	Blok Diagram EZ-USB AN2131QC.....	11
Gambar 2.6	Konfigurasi 80 PIN AN2131QC.....	12
Gambar 2.7	Blok Diagram PCF8574 8-bit I/O Expander.....	13
Gambar 2.8	Konfigurasi PIN IC PCF8574.....	13
Gambar 2.9	PCF8574A slave addresses.....	15
Gambar 2.10	Mode Tulis (Output) PCF8574.....	15
Gambar 2.11	Mode Baca (Input) PCF8574.....	16
Gambar 2.12	Mekanisme kerja untuk seluruh jenis motor secara umum.....	17
Gambar 2.13	Motor listrik AC sinkron hurst model T 6RPM 115V 60HZ P/N : 2602-013.....	18
Gambar 2.14	Komponen Utama Motor Sinkron.....	19
Gambar 2.15	(a) <i>Limit Switch Normaly Open</i> (NO).....	20
Gambar 2.15	(b) <i>Limit Switch Normaly Closen</i> (NC).....	21
Gambar 2.16	Aplikasi Limit Switch Pada Sistem Kontrol Posisi.....	21
Gambar 2.17	Relay Omron type MY2NJ.....	22
Gambar 2.18	Kurva karakteristik Transistor.....	23
Gambar 2.19	Konfigurasi Transistor sebagai saklar.....	24
Gambar 3.1	<i>Sample Tray</i> perangkat RIA Nucleus Model 1600 Gamma Counting.....	26
Gambar 3.2	Blok diagram sistem keseluruhan.....	27

Gambar 3.3	Rangkaian Dasar <i>Driver</i> Motor.....	30
Gambar 3.4	Rangkaian driver motor Perangkat RIA Nuclues Model 1600 Gamma Counting.....	32
Gambar 3.5	Perangkat Mekanik <i>Sample Changer</i> Perangkat RIA Nucleus Nodel 1600 Gamma Counting.....	33
Gambar 3.6	Rangkaian <i>Power Supply</i> +5 Vdc dan +12 Vdc.....	33
Gambar 3.7	Diagram alir kerja sistem.....	34
Gambar 3.8	Diagram alir program gerak motor horizontal.....	35
Gambar 3.9	Diagram alir program gerak motor vertikal.....	36
Gambar 3.10	Diagram alir program utama sistem.....	38
Gambar 4.1	Indikator yang menunjukkan berhasil atau tidaknya modul <i>interface</i> USB Devasys I2C I/O dikenali oleh PC.....	39
Gambar 4.2	Rangkaian pengujian modul <i>interface</i> USB Devasys I2C I/O dan modul <i>driver</i> motor.....	41
Gambar 4.3	Tampilan program pengujian modul <i>driver</i> motor.....	41
Gambar 4.4	Hasil pengujian modul <i>driver</i> motor.....	42
Gambar 4.5	Rangkaian pengujian sistem keseluruhan.....	44
Gambar 4.6	Tampilan program sistem kontrol <i>sample changer</i> perangkat RIA Nucleus Model 1600 Gamma Counting.....	45

## DAFTAR TABEL

Tabel 2.1	Fungsi-Fungsi PIN pada konektor USB.....	9
Tabel 2.3	Fungsi-Fungsi PIN PCF8574.....	14
Tabel 3.1	Fungsi Penggunaan Port B.....	29
Tabel 3.2	Fungsi Penggunaan Port C.....	29
Tabel 3.3	Fungsi Penggunaan Port I2C I/O.....	30
Tabel 4.1	Hasil pengujian modul <i>interface</i> USB Devasys I2C I/O . .....	40
Tabel 4.2	Hasil pengujian modul <i>driver</i> motor.....	42
Tabel 4.3	Hasil Pengujian gerak posisi kolimator <i>detector</i> . .....	45
Tabel 4.4	Hasil Pengujian proses <i>counting sample</i> dengan perintah <i>counting current sample</i> .....	46
Tabel 4.5	Hasil Pengujian proses <i>counting sample</i> dengan perintah <i>counting 25 sample</i> . .....	47
Tabel 4.6	Hasil Pengujian proses <i>counting sample</i> dengan perintah <i>counting All sample (100Sample)</i> .....	48
Tabel 4.7	Hasil Pengujian proses <i>counting sample</i> pada nomor <i>sample</i> tertentu. ....	50

## DAFTAR LAMPIRAN

Lampiran 1	Gambar 2.5 Rangkaian Modul Devasys I2C I/O.....	57
Lampiran 2	Tabel 2.2 Fungsi Konfigurasi 80 PIN AN2131QC.....	58
Lampiran 3	Listing Program.....	61

# BAB 1 PENDAHULUAN

## 1.1 LATAR BELAKANG

RIA Nucleus Model 1600 Gamma Counting merupakan salah satu perangkat Radioimmunoassay (RIA). Perangkat RIA banyak digunakan untuk menganalisis zat-zat yang ada didalam cairan tubuh diantaranya urin, *hormone*, dan lain-lain atau kultur media yang berkadar rendah dan memiliki matriks yang kompleks.

Perangkat RIA Nucleus Model 1600 Gamma Counting yang terdapat di instalasi kedokteran nuklir Badan Tenaga Nuklir Nasional (BATAN) saat ini dalam kondisi sudah tidak dapat dioperasikan lagi, beberapa hal yang melatarbelakanginya adalah umur pakai dari perangkat RIA Nucleus Model 1600 Gamma Counting dan sistem pengoperasiannya yang dinilai kurang efisien dan kurang fleksibel.

Dilihat dari prinsip kerja sistem kontrol pergerakan *sample changer*, perangkat RIA Nucleus Model 1600 Gamma Counting menggunakan sistem kontrol secara manual tanpa *Personal Computer (PC)* / (*stand alone*) dan sistem kontrol pergerakan dan proses *counting sample* dilakukan secara kontinuitas pada 100 *sample*, sehingga dalam pengoperasiannya dinilai kurang fleksibel. [1]

Sistem perangkat lunak perangkat RIA Nucleus Model 1600 Gamma Counting menggunakan tampilan sederhana berupa LCD (*Liquid Crystal Display*) dan input data dengan menggunakan *keypad* sederhana. Pada pengoperasiannya tampilan tersebut kurang komunikatif dan kurang informatif, karena tidak dapat melihat posisi *detector* dan posisi nomor *sample*.

Didasari dari hal-hal diatas maka pada skripsi ini akan dirancang sebuah sistem kontrol baru yang lebih efisien dan fleksibel (*user friendly*) dalam hal pengoperasian untuk menggerakkan *sample*. Perangkat RIA Nucleus Model 1600 Gamma Counting menggunakan sistem kontrol melalui PC sebagai pengendali pergerakan *sample* dan menerapkan komunikasi USB (*Universal Serial Bus*) sebagai jalur komunikasi data antara perangkat dengan PC.

## 1.2 PERUMUSAN MASALAH

Berdasarkan latar belakang maka dapat dirumuskan masalah sebagai berikut:

1. Bagaimana merancang dan membuat modul elektronik sebagai *driver* penggerak motor.
2. Bagaimana merancang dan membuat sistem baru dari kontrol pergerakan *sample* berbasis PC dengan modul Devasys I2C I/O yang mampu mengkoordinasikan dua buah motor penggerak, sistem transmisi, serta *limit switch* sebagai indikator posisi kolimator *detector* dan indikator posisi nomor *sample* dengan tujuan mendapatkan pergerakan *sample* yang optimal.
3. Bagaimana menggabungkan komponen pendukung antara sistem lama dengan sistem baru baik dari segi elektronik dan mekanik.
4. Bagaimana membuat perangkat lunak yang *user friendly* sebagai visualisasi kontrol dan instruksi dari pergerakan *sample*.

## 1.3 TUJUAN

Tujuan penulisan skripsi ini adalah untuk merancang, membuat dan menggabungkan sistem kontrol pergerakan *sample* pada perangkat RIA Nucleus Model 1600 Gamma Counting yang sudah tidak dapat dioperasikan dengan sistem kontrol pergerakan *sample changer* hasil rancangan yang baru, sebagai solusi dari beberapa latar belakang masalah sehingga didapatkan sistem kontrol pergerakan *sample* yang mampu bekerja optimal dan sesuai dengan kebutuhan.

## 1.4 BATASAN MASALAH

Dalam perancangan dan pembuatan sistem ini, terdapat beberapa pembatasan masalah, antara lain:

1. Sistem kontrol pergerakan *sample changer* adalah bagian dari sistem perangkat RIA Nucleus Model 1600 Gamma Counting. Sistem kontrol pergerakan *sample changer* adalah sistem yang hanya mengatur pergerakan *sample* sehingga proses pergerakan *sample* bekerja dan berfungsi secara fleksibel dan optimal.



2. Perangkat lunak yang dibuat hanya merupakan bagian dari sistem perangkat lunak RIA secara keseluruhan. Bagian yang dimaksud adalah bagian pengendali dari sistem kontrol pergerakan *sample changer*, termasuk visualisasi kontrolnya.

## 1.5 METODOLOGI

Perancangan sistem kontrol pergerakan *sample changer* RIA (*Radioimmunoassay*) dengan komunikasi USB menggunakan metodologi sebagai berikut:

1. Studi literatur

Studi literatur mengacu pada prinsip kerja alat secara fungsi, meliputi penggunaan dua buah motor penggerak, sistem transmisi, serta *limit switch* sebagai indikator posisi dan nomor sample

2. Perancangan sistem

Sistem yang dirancang dan dibuat secara prinsip harus mempunyai nilai lebih dari sistem sebelumnya, baik dari nilai fungsi dan nilai waktu juga dari peningkatan kemampuan dan unjuk kerja sistem.

Langkah-langkah dalam perencanaan sistem adalah:

- a. Pembuatan blok diagram rangkaian.
- b. Perancangan dan pengkajian *design* rangkaian untuk modul *driver* motor.
- c. Menguji rangkaian pada PCB Matriks
- d. Menyusun tiap-tiap blok menjadi satu sistem secara keseluruhan.
- e. Perancangan dan pembuatan perangkat lunak sistem.

3. Pembuatan modul *driver* motor dan perangkat lunak

Pembuatan dimulai dari pemasangan komponen dan penyambungan antar komponen dengan kabel, dan pembuatan perangkat lunak yang mendukung sistem secara keseluruhan.

4. Pengujian Alat

Pengujian dilakukan pada tiap blok rangkaian dan seluruh sistem. Dalam pengujian ini dilakukan pada blok-blok sebagai berikut:

- a. Pengujian perangkat keras (2 buah motor penggerak, sistem mekanik, serta *limit switch* sebagai indikator posisi kolimator *detector* dan indikator posisi nomor sample)
- b. Pengujian *driver* motor.
- c. Pengujian modul sistem komunikasi USB.
- d. Pengujian sistem secara keseluruhan.

## 1.6 SISTEMATIKA PENULISAN

Dalam Penulisan laporan skripsi ini, disusun berdasarkan bab-bab dan terdiri atas lima bab dan selanjutnya diperjelas dalam beberapa sub bab. Secara keseluruhan skripsi ini disusun dalam sistematika sebagai berikut:

### 1. Pendahuluan

Berisi tentang latar belakang masalah, perumusan masalah, tujuan penulisan, batasan masalah, metodologi dan sistematika penulisan.

### 2. Landasan Teori

Bagian pertama menjelaskan tentang perangkat RIA dan teknik RIA secara umum, bagian kedua menjelaskan tentang teknologi USB (*Universal Serial Bus*), bagian ketiga menjelaskan modul antar muka Devasys I2C IO dan mikrokontroler AN2131QC, bagian keempat menjelaskan PCF8574 8-bit sebagai *I/O expander*, bagian kelima menjelaskan tentang motor listrik, bagian keenam menjelaskan tentang *switch* dimana *limi tswitch* digunakan sebagai indikator posisi, bagian ketujuh menjelaskan tentang rangkaian Relay yang digunakan pada sistem kontrol pergerakan *sample changer*.

### 3. Rancang Bangun Sistem

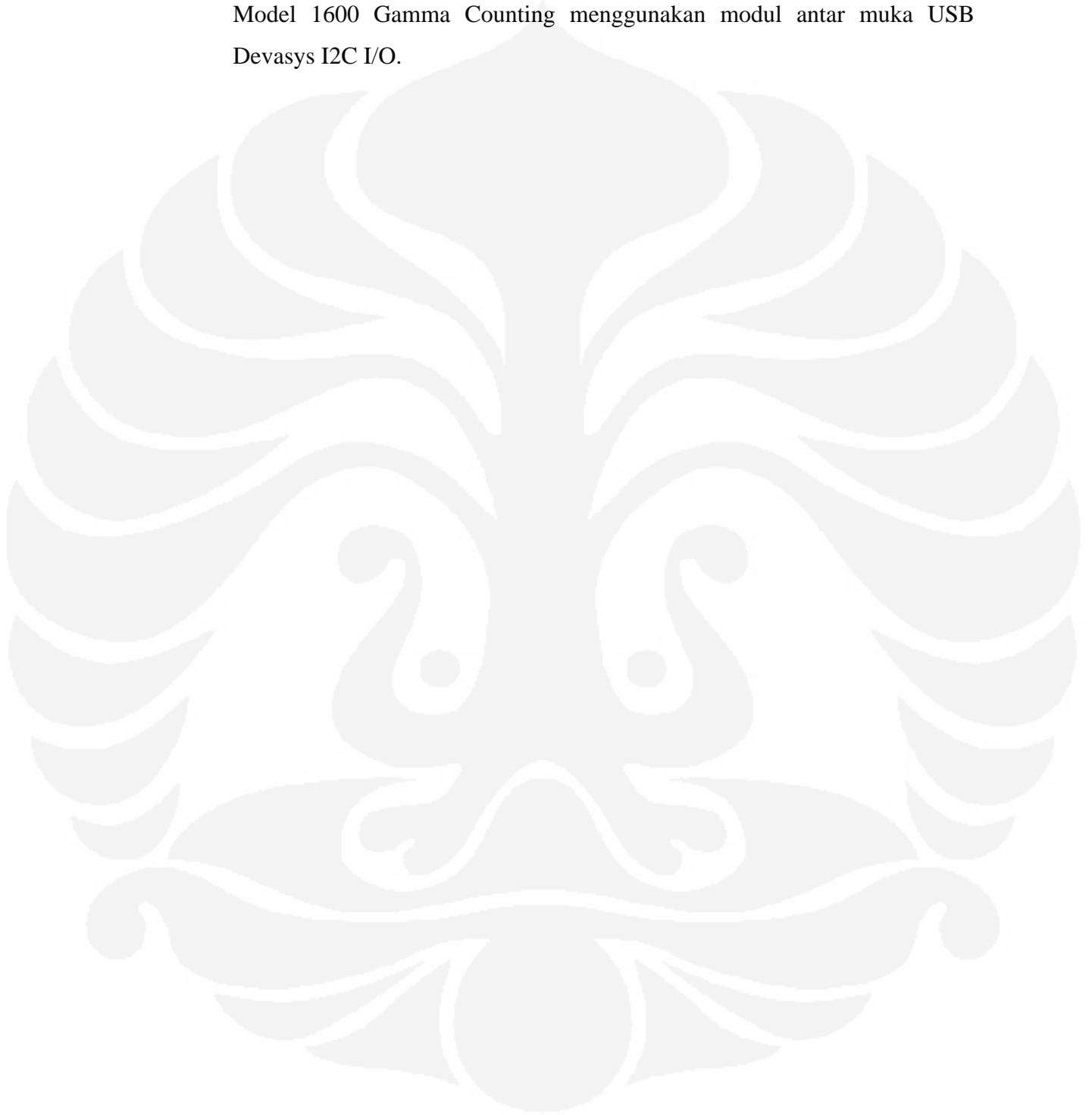
Bagian pertama menjelaskan prinsip kerja sistem, sedangkan bagian kedua menjelaskan perancangan perangkat keras. Bagian ketiga menjelaskan perancangan perangkat lunak.

### 4. Pengujian dan Analisa Sistem

Bagian pertama menjelaskan pengujian modul *interface* USB Devasys I2C I/Okemudian bagian kedua menjelaskan pengujian *driver* motor, bagian ketiga menjelaskan pengujian sistem keseluruhan.

## 5. Kesimpulan

Berisi tentang kesimpulan dari rancang bangun sistem kontrol *sample changer* berbasis PC pada perangkat Radioimmunoassay (RIA) Nucleus Model 1600 Gamma Counting menggunakan modul antar muka USB Devasys I2C I/O.



## BAB 2 LANDASAN TEORI

### 2.1 PERANGKAT RADIOIMMUNOASSAY (RIA)

Perangkat Radioimmunoasay (RIA) merupakan alat kedokteran nuklir yang digunakan untuk menganalisis zat-zat yang ada didalam cairan tubuh diantaranya urin, *hormone*, dan lain-lain atau kultur media yang berkadar rendah dan memiliki matriks yang kompleks. Teknik pengukuran RIA berdasarkan pada reaksi imunologi dengan menggunakan *radioisotope* sebagai perunutnya. Isotop yang paling umum digunakan I-125. Isotop lainnya yang dapat digunakan adalah I-131, P-32, S-35, dan C-14. Radioaktivitasnya kemudian diukur dengan pencacah gamma dan proporsional dengan jumlah antigen.

Pada tahun 1960 teknik analisis RIA pertama kali diperkenalkan oleh Rosaly Yalow dan Berson di Amerika yang hampir bersamaan dilakukan juga oleh Ekin di Inggris. Temuan mereka ini telah membawa kemajuan besar dalam pengukuran berbagai macam hal penting secara biologis. Kini setelah mengalami perkembangan, teknik RIA telah banyak digunakan oleh negara maju maupun berkembang. Teknik ini banyak dipilih karena keunggulan dalam beberapa hal, antara lain :

- a. Lebih sederhana dalam proses pencuplikan *sample*.
- b. Mempunyai ketepatan dan akurasi data.
- c. Mempunyai ketelitian yang tinggi.
- d. Dapat digunakan untuk berbagai analisa.

Hingga kini ketelitian penentuan *in vitro* secara radioimunologi, belum tertandingi oleh metode apapun, serta telah banyak membantu dalam *diagnosissecara dini* berbagai penyakit yang disebabkan oleh kelainan hormonal, kanker, terkena racun dan infeksi.

Mengingat fungsi dan kemampuannya yang handal, RIA banyak digunakan di rumah sakit - rumah sakit dan di instalasi kedokteran nuklir lainnya untuk berbagai keperluan penelitian. Perangkat RIA yang banyak dipakai dirumah sakit dan di instalasi kedokteran nuklir selama ini memiliki beberapa klasifikasi, diantaranya :

1. Perangkat RIA dengan media sample manual tanpa PC

Tipe perangkat RIA ini menggunakan banyak *detector*, seperti *multi well gamma counters* dan *multi detectors gamma counters*. Perangkat RIA seperti tipe diatas membutuhkan banyak *detector* sehingga harganya mahal. Sistem countingnya manual dan operator harus berada ditempat sampai mendapatkan hasil pencacahan. Secara elektronik perangkat tersebut masih banyak menggunakan rangkaian analog sehingga membutuhkan arus yang besar dan mengakibatkan sistem pengkabelan yang tidak sederhana. Akusisi datanya tidak bisa disimpan karena piranti *input-outputnya* tanpa PC, hanya menggunakan *keypad* dan printer.

2. Perangkat RIA media *sample* manual dengan PC

Perangkat RIA tipe ini rata-rata dimiliki oleh rumah sakit-rumah sakit. Contoh tipe seperti ini adalah Gamma Management System, yaitu perangkat RIA media *sample* manual multi *detector*. *Detector* yang digunakan jumlahnya bervariasi dari 6 sampai 10 *detector*. Perangkat ini *import* dan harganya mahal. Sistem perawatannya dibawah pengawasan perwakilan distributor di Indonesia sehingga membutuhkan biaya yang besar. Perangkat ini relatif sama dari tipe yang pertama. Perbedaannya, sistem akusisi datanya sudah memakai komputer, tetapi sistem interfacenya masih menggunakan *parallel port* sehingga teknologinya masih menggunakan teknologi lama.

3. Perangkat RIA media *sample changer* tanpa PC

Contoh dari perangkat ini adalah model 1600 automatic gamma counter, *single detector* (detektor tunggal). Kelebihan alat ini dari yang diatas, alat ini sudah menggunakan *sample changer*, sistem countingnya otomatis dan dapat ditinggal selama proses pencacahan. *Sample* yang akan dicacah sebanyak jumlah lubang pada *sample tray*, yaitu 100 *sample*. Kelemahan alat ini adalah pergerakan *sample changer* dinilai kurang fleksibel dan kurang efisien karena sistem pergerakan *sample* dilakukan secara kontinuitas pada 100 *sample*.

## 2.2 TEKNOLOGI USB (*Universal Serial Bus*)

USB – *Universal Serial Bus* merupakan teknik baru menghubungkan Komputer dengan peralatannya. Agar peralatan USB bisa “*Hot-plugable*” dan “*Plug & Play*”, komputer setiap saat akan melakukan “proses pengenalan” (*enumerated*) pada semua peralatan USB yang terpasang dalam saluran. Selama proses pengenalan tersebut, komputer akan menanyakan identitas kepada alat yang baru saja dihubungkan ke komputer sehingga belum dikenali komputer. Saat ini peralatan USB yang terpasang wajib melaporkan identitas dirinya serta informasi-informasi spesifik tentang dirinya.

Jika proses pengenalan ini berhasil, maka komputer akan mengambil program untuk mengendalikan alat tersebut (*driver*), dan berikutnya peralatan USB tersebut sudah langsung siap dipakai. Kalau hal ini terjadi pada Windows, selesai proses pengenalan suatu peralatan USB baru, maka pada Control Panel – System – Device Manager akan langsung terlihat ada peralatan USB baru yang siap dipakai.

Kecepatan transfer data USB dibagi menjadi tiga, antara lain:

- a. High speed data dengan frekuensi clock 480.00Mb/s dan toleransi pensinyalan data pada  $\pm 500$ ppm.
- b. Full speed data dengan frekuensi clock 12.000Mb/s dan toleransi pensinyalan data pada  $\pm 0.25\%$  atau 2,500ppm.
- c. Low speed data dengan frekuensi clock 1.50Mb/s dan toleransi pensinyalan data pada  $\pm 1.5\%$  atau 15,000ppm.

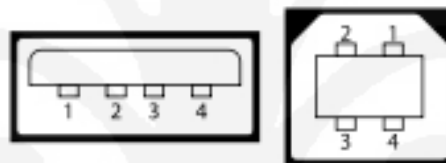
Untuk kecepatan tinggi (480.00 Mbps) dapat dilakukan dengan menggunakan USB versi 2.0. Dengan kecepatan ini, USB dapat menyaingi kecepatan yang diperoleh dengan menggunakan *firewire* dan dengan harga yang lebih murah.

Konektor pada USB mempunyai bentuk yang sama. Jenis konektor USB adalah konektor tipe A (*upstream*) dan konektor tipe B (*downstream*). Secara umum konektor USB ditunjukkan pada Gambar 2.1



**Gambar 2.1** (a) Konektor tipe A (*upstream*) [2]  
(b) Konektor tipe B (*downstream*). [2]

Adapun konfigurasi PIN konektor tipe A (*upstream*) dan konektor tipe B (*downstream*) ditunjukkan pada Gambar 2.2, dan fungsi konfigurasi PIN konektor tipe A (*upstream*) dan konektor tipe B (*downstream*) ditunjukkan pada Tabel 2.1.



**Gambar 2.2.** Konfigurasi PIN konektor tipe A (*upstream*) dan konektor tipe B (*downstream*). [3]

**Tabel 2.1** Fungsi-Fungsi PIN pada konektor USB

PIN	Fungsi
1	$V_{BUS}$ (4.75–5.25 V)
2	D <sup>-</sup>
3	D <sup>+</sup>
4	<u>GND</u>
Shell	<u>Shield</u>

Kabel USB terdiri dari 4 utas kabel ditambah konduktor pembungkus kabel, seperti pelindung yang biasanya dijumpai dalam kabel audio. Kabel nomor 1 dipakai untuk menyalurkan sumber daya dengan tegangan 5 Volt, jika diperlukan peralatan USB boleh mengambil daya dari saluran ini tidak lebih dari 100 mA. Komputer yang dilengkapi dengan kemampuan USB, wajib menyediakan daya sebesar 500 mA untuk keperluan ini. Peralatan USB yang memerlukan daya lebih dari ketentuan tersebut di atas, harus menyediakan sendiri sumber daya untuk keperluan kerja peralatan tersebut.

Kabel nomor 2 dan nomor 3 dipakai untuk pengiriman sinyal. Kabel nomor 2 bernama D- dan kabel nomor 3 bernama D+, tegangan pada dua saluran ini berubah antara 0 Volt dan 3,3 Volt. Sinyal digital yang dikirim melalui dua saluran ini dikatakan sebagai '*difference signal*', artinya sinyal digital '0' atau '1' tidak dinyatakan dengan besarnya tegangan pada saluran tersebut terhadap ground, seperti halnya sinyal digital yang dipakai dalam IC TTL (transistor Transistor Logic) atau dalam saluran RS232.

Sinyal digital dinyatakan dengan perbedaan tegangan antara dua kabel tersebut. Jika tegangan pada saluran D+ lebih tinggi dari tegangan pada saluran D-, maka informasi yang dikirimkan adalah sinyal digital '1', sebaliknya sinyal digital '0' dinyatakan dengan tegangan pada D+ < tegangan pada D-. Kabel nomor 4 adalah ground sebagai saluran balik sumber tegangan 5 Volt.

### 2.3 MODUL ANTAR MUKA USB DEVASYS I2C/IO

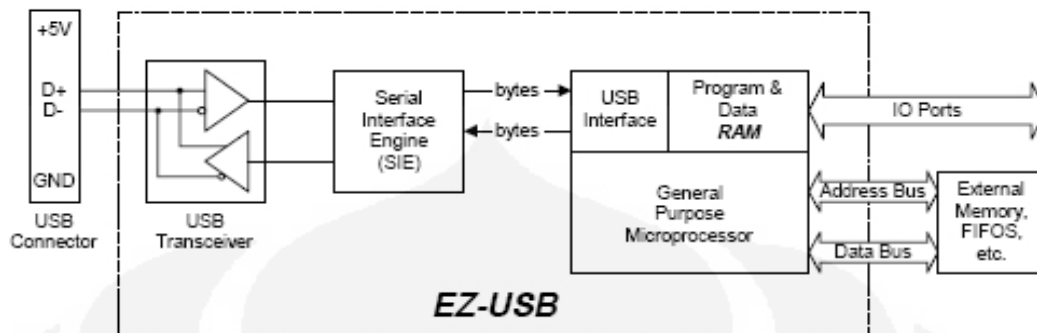
Devasys USB I2C/IO adalah sebuah modul antar muka yang dikembangkan oleh Devasys dengan mengadopsi mikrokontroler buatan Cypress AN2131QC yang merupakan salah satu dari keluarga EZ-USB. Gambar modul antar muka Devasys USB I2C I/O ditunjukkan pada Gambar 2.3.



**Gambar 2.3.** Modul Antarmuka USB Devasys I2C/IO. [4]

Gambar blok diagram dari AN2131QC 80 PIN yang merupakan keluarga EZ-USB ditunjukkan pada Gambar 2.4.





**Gambar 2.4.** Blok Diagram EZ-USB AN2131QC. [5]

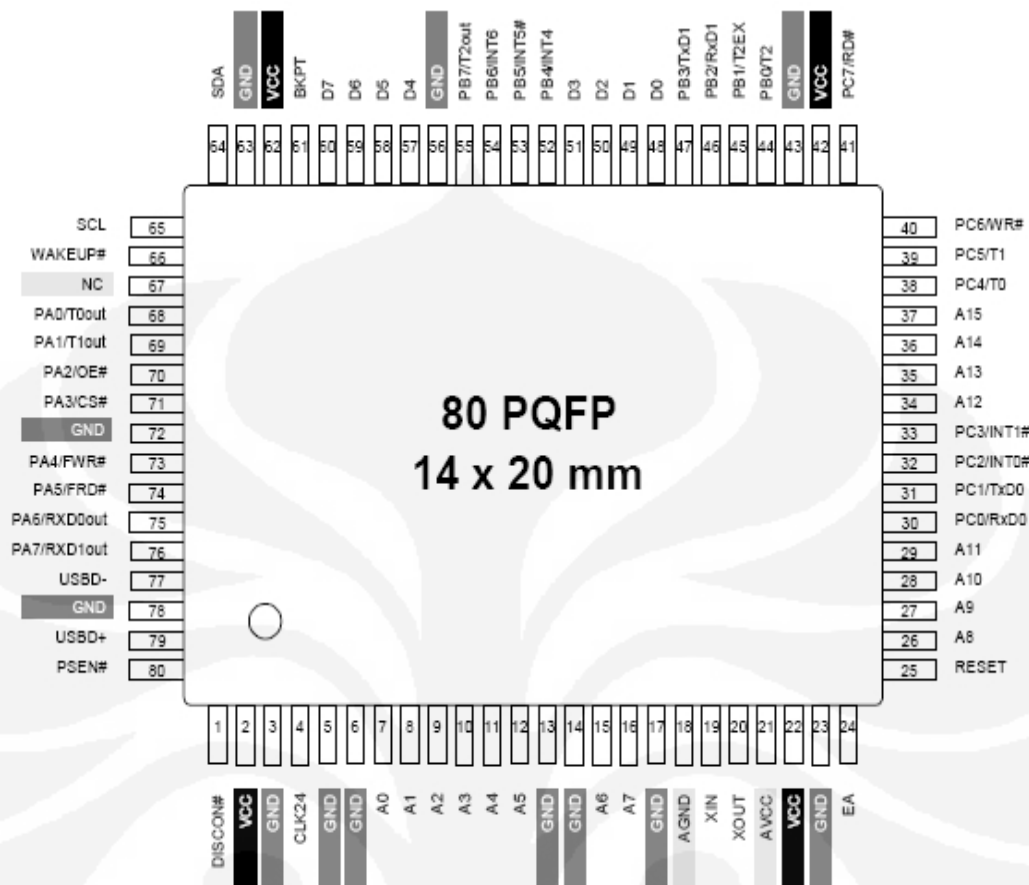
Modul antar muka Devasys I2C/IO menyediakan 24 bit konfigurasi digital I/O dengan 16 bit *address bus* (*Port A, Port B, Port C*) dan 8 bit *data bus* untuk penambahan eksternal memori. Gambar rangkaian modul Devasys I2C I/O ditunjukkan pada Gambar 2.5. (Lampiran 1).

Mikroprocessor AN2131QC yang terdapat pada modul antar muka Devasys USB I2C/IO, merupakan pengembangan dari mikroprocessor 8051 dengan penambahan fasilitas dan waktu eksekusi yang lebih cepat. Mikroprocessor AN2131QC ini menggunakan RAM internal untuk menyimpan data, sehingga memberikan kemudahan pada proses pemrograman.

Keluarga EZ-USB menggunakan pengembangan antar muka SIE/USB (yang disebut “*USB Core*”), dimana sebagai perangkat USB performannya akan lebih sempurna.

Mikrokontroler AN2131QC dikemas dalam tipe PQPF dan mempunyai 80 PIN, ukuran dimensinya 14 x 20 mm.

Konfigurasi PIN dari Mikrokontroler AN2131QC ditunjukkan pada Gambar 2.6.

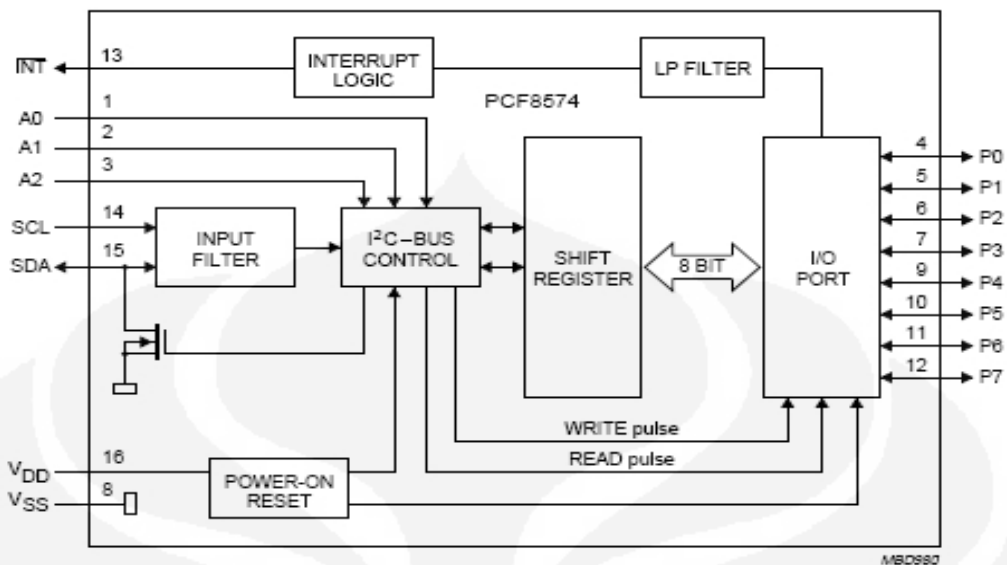


**Gambar 2.6.** Konfigurasi 80 PIN AN2131QC. [6]

Dari Gambar 2.6 dapat dijelaskan secara fungsional konfigurasi PIN AN2131QC pada Tabel 2.3. (lampiran 2).

#### 2.4 PCF 8574 8-BIT I/O EXPANDER

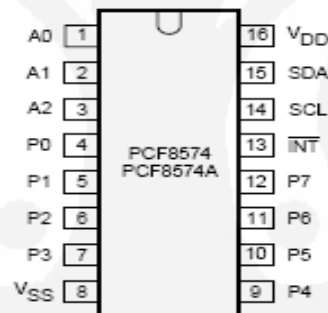
Fungsi utama dari PCF8574 ialah mengekspansi *port* I/O pada berbagai macam jenis mikrokontroler melalui 2 jalur *bidirectional* I<sup>2</sup>C Bus. Pada PCF8574 terdapat 8-Bit *quasi-bidirectional Port* dan I<sup>2</sup>C Bus *Interface*. PCF8574 mengkonsumsi daya yang rendah dan mempunyai *latched output* dengan kemampuan menggerakkan arus yang besar. PCF8574 juga memiliki jalur Interupsi (INT) yang dapat dihubungkan dengan *interrupt logic* dari mikrokontroler. Dengan mengirimkan sinyal interupsi, PCF8574 dapat menginformasikan mikrokontroler bahwa ada data yang datang tanpa menggunakan jalur I<sup>2</sup>C Bus. Blok diagram PCF8574 ditunjukkan pada gambar 2.7



Gambar 2.7. Blok Diagram PCF8574 8-bit I/O Expander. [7]

#### 2.4.1 Konfigurasi PIN IC PCF8574

Konfigurasi PIN IC PCF8574 ditunjukkan pada Gambar 2. 8



Gambar 2.8. Konfigurasi PIN IC PCF8574. [8]

Dari Gambar 2.8 dapat dijelaskan secara fungsional konfigurasi PIN IC PCF8574 pada Tabel 2.4.

Tabel 2.4. Fungsi-Fungsi PIN PCF8574

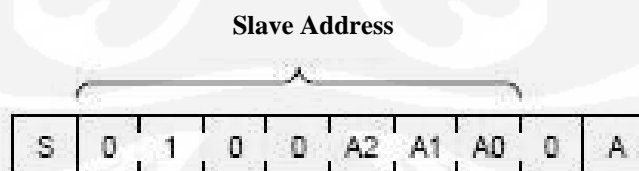
PIN	Nama	Keterangan
1	A0	address input 0
2	A1	address input 1
3	A2	address input 2
4	P0	quasi-bidirectional I/O 0
5	P1	quasi-bidirectional I/O 1
6	P2	quasi-bidirectional I/O 2
7	P3	quasi-bidirectional I/O 3

**Tabel 2.4.** Fungsi-Fungsi PIN PCF8574 (Lanjutan)

PIN	Nama	Keterangan
8	V <sub>SS</sub>	supply ground
9	P4	quasi-bidirectional I/O 4
10	P5	quasi-bidirectional I/O 5
11	P6	quasi-bidirectional I/O 6
12	P7	quasi-bidirectional I/O 7
13	INT	interrupt output (active LOW)
14	SCL	serial clock line
15	SDA	serial data line
16	V <sub>DD</sub>	supply voltage

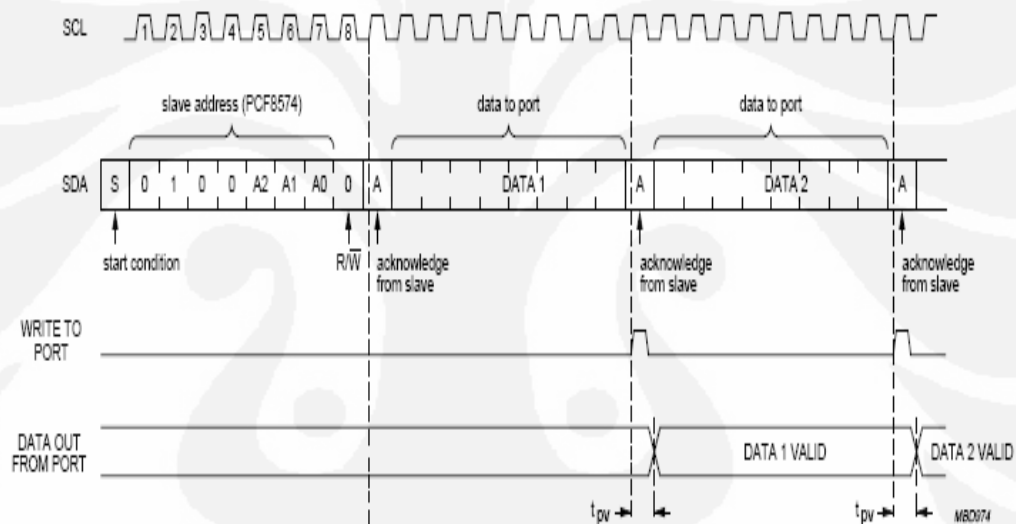
### 2.4.2 Pengalamatan PCF8574

Setelah *Master* mengirimkan sinyal *Start* sebagai tanda awal *transmisi*, *Master* kemudian mengirimkan alamat *Slave* ke piranti *Slave* yang ingin dituju, dalam hal ini IC I2C PCF8574. Empat bit pertama adalah nomor group, untuk PCF8574 adalah 0100. Pengalamatan ini telah ditetapkan oleh perusahaan Philips sebagai pencipta I2C. Tiga bit selanjutnya (A0, A1, A2) adalah nomor *chip* yang digunakan untuk menentukan IC mana yang akan diakses oleh *Master*. Bit terakhir digunakan untuk menentukan operasi baca atau operasi tulis yang akan dibentuk. Bit ini diisi 1 untuk membentuk operasi baca atau diisi 0 untuk membentuk operasi tulis. Setelah *Master* mengirimkan sinyal *Start* dan alamat *Slave* ke IC PCF8574, *Slave* merespon dengan mengirimkan sinyal *acknowledge* ke *Master*. Operasi pengalamatan pada *Slave* ditunjukkan pada Gambar 2.9.

**Gambar 2.9.** PCF8574A slave addresses. [9]

### 2.4.3 Mode Tulis (*Output*) pada IC PCF8574

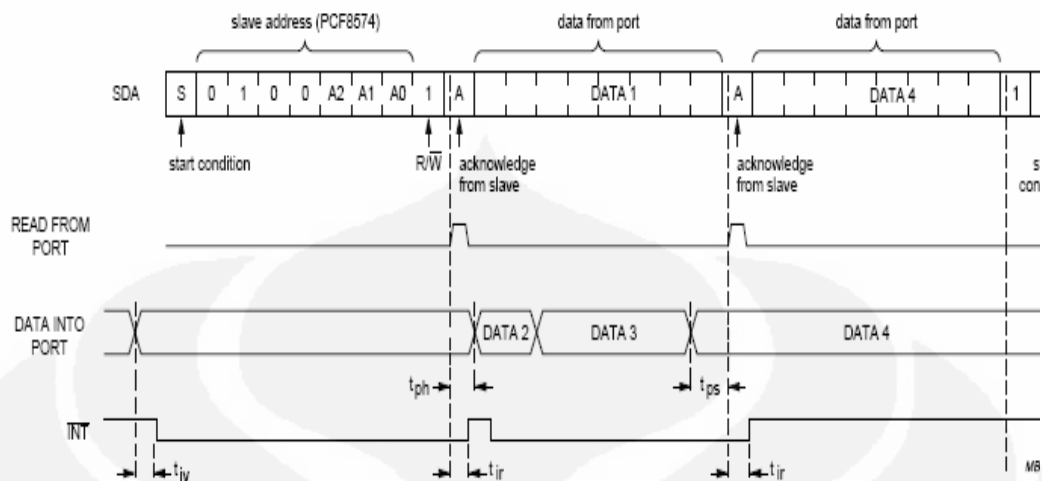
Setelah *Master* mengirimkan sinyal *Start* dan alamat *Slave* (dengan bit R/W diisi 0) pada IC PCF8574, IC PCF8574 mengirimkan sinyal *acknowledge* selanjutnya master mengirimkan *byte* data pertama. Apabila *byte* data pertama sudah dikeluarkan pada *port output* PCF8574, IC PCF8574 mengirimkan sinyal *acknowledge* lagi, kemudian master akan mengirimkan *byte* data ke-2 untuk dikeluarkan ke *port output* PCF8574. Setelah semua data sudah dikeluarkan, barulah *Master* mengirim sinyal *Stop* untuk mengakhiri transfer data. Berikut ini gambar mode tulis (*Output*) pada IC PCF8574 seperti yang ditunjukkan pada Gambar 2.10.



Gambar 2.10 Mode Tulis (Output) PCF8574. [10]

### 2.4.4 Mode Baca (*Input*) pada IC PCF8574

Proses membaca data dari *port I/O* PCF8574 sama dengan proses pada mode tulis, hanya bedanya pada bit R/W diisi 1. Berikut ini gambar mode baca (*Input*) pada IC PCF8574 seperti yang ditunjukkan pada Gambar 2.11.



**Gambar 2. 11** Mode Baca (Input) PCF8574. [11]

## 2.5 MOTOR LISTRIK

Motor listrik termasuk kedalam kategori mesin listrik dinamis dan merupakan sebuah perangkat elektromagnetik yang mengubah energi listrik menjadi energi mekanik. Energi mekanik di industri digunakan misalnya untuk memutar impeller pompa, fan atau blower, menggerakkan kompresor, mengangkat bahan, dan lain-lain dan digunakan juga pada peralatan listrik rumah tangga (seperti: mixer, bor listrik, kipas angin).

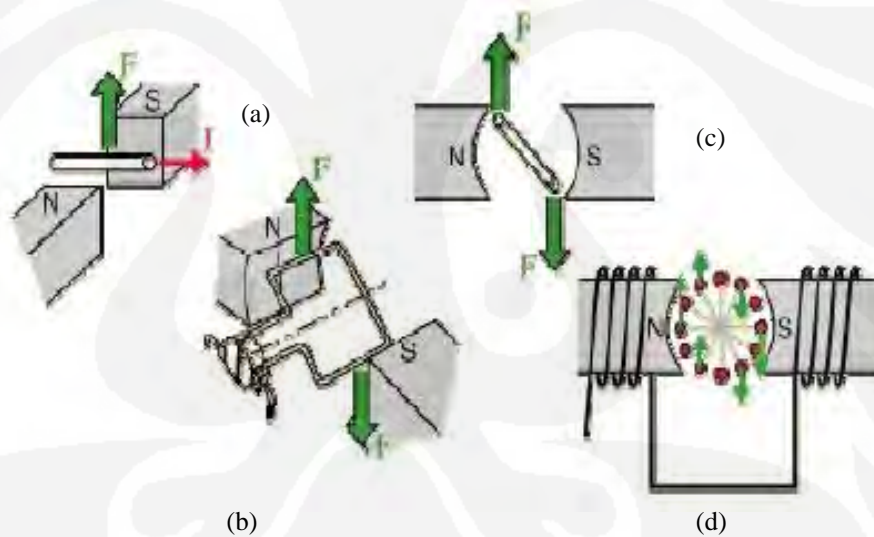
Dalam memahami sebuah motor, penting untuk mengerti apa yang dimaksud dengan beban motor. Beban mengacu kepada keluaran tenaga putar/*torque* sesuai dengan kecepatan yang diperlukan. Beban umumnya dapat dikategorikan kedalam tiga kelompok, yaitu :

- Beban torque konstan adalah beban dimana permintaan keluaran energinya bervariasi dengan kecepatan operasinya namun *torque* nya tidak bervariasi. Contoh beban dengan *torque* konstan adalah *conveyors*, *rotary kilns*, dan pompa *displacement* konstan.
- Beban dengan variabel torque adalah beban dengan *torque* yang bervariasi dengan kecepatan operasi. Contoh beban dengan variabel *torque* adalah pompa sentrifugal dan fan (*torque* bervariasi sebagai kwadrat kecepatan).

- c. Beban dengan energi konstan adalah beban dengan permintaan *torque* yang berubah dan berbanding terbalik dengan kecepatan. Contoh untuk beban dengan daya konstan adalah peralatan-peralatan mesin.

Mekanisme kerja untuk seluruh jenis motor secara umum adalah sama, seperti ditunjukkan pada Gambar 2.12.

- Arus listrik dalam medan magnet akan memberikan gaya
- Jika kawat yang membawa arus dibengkokkan menjadi sebuah lingkaran/*loop*, maka kedua sisi *loop*, yaitu pada sudut kanan medan magnet, akan mendapatkan gaya pada arah yang berlawanan.
- Pasangan gaya menghasilkan tenaga putar / *torque* untuk memutar kumparan.
- Motor-motor memiliki beberapa *loop* pada dinamanya untuk memberikan tenaga putaran yang lebih seragam dan medan magnetnya dihasilkan oleh susunan elektromagnetik yang disebut kumparan medan



**Gambar 2. 12** Mekanisme kerja untuk seluruh jenis motor secara umum. [12]

Jenis motor listrik yang digunakan pada perangkat RIA Nucleus Model 1600 Gamma Counting adalah motor listrik AC sinkron buatan hurst model T 6RPM 115V 60HZ, P/N : 2602-013. Bentuk fisik dari motor listrik AC sinkron hurst model T 6RPM 115V 60HZ ditunjukkan pada Gambar 2.13.



**Gambar 2.13** Motor listrik AC sinkron hurst model T 6RPM 115V 60HZ P/N : 2602-013. [13]

Motor sinkron adalah motor listrik arus bolak-balik (AC), bekerja pada kecepatan tetap pada sistem frekuensi tertentu. Motor ini memerlukan arus searah (DC) untuk pembangkitan daya dan memiliki *torque* awal yang rendah.

Komponen utama dari motor sinkron adalah :

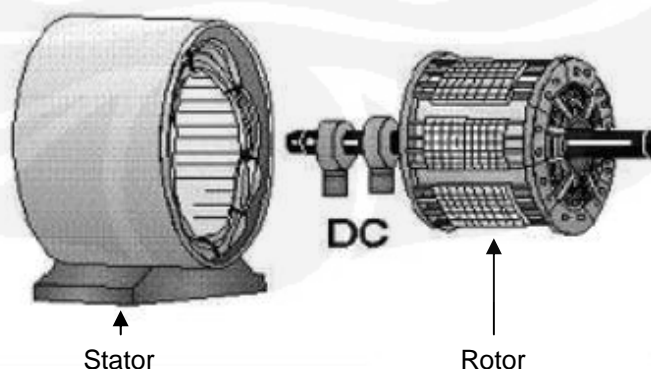
a. *Rotor*.

Perbedaan utama antara motor sinkron dengan motor induksi adalah bahwa rotor mesin sinkron berjalan pada kecepatan yang sama dengan perputaran medan magnet. Hal ini memungkinkan sebab medan magnet rotor tidak lagi terinduksi. Rotor memiliki magnet permanen atau arus *DC-excited*, yang dipaksa untuk mengunci pada posisi tertentu bila dihadapkan dengan medan magnet lainnya.

b. *Stator*.

Stator menghasilkan medan magnet berputar yang sebanding dengan frekuensi yang dipasok. Motor ini berputar pada kecepatan sinkron.

Secara umum komponen utama dari motor sinkron ditunjukkan pada Gambar 2.14



**Gambar 2.14** Komponen Utama Motor Sinkron. [14]



Sebuah motor sinkron dapat dinyalakan oleh sebuah motor dc pada satu sumbu. Ketika motor mencapai kecepatan sinkron, arus AC diberikan kepada belitan stator. Motor dc saat ini berfungsi sebagai generator dc dan memberikan eksitasi medan dc kepada rotor. Beban sekarang boleh diberikan kepada motor sinkron. Motor sinkron seringkali dinyalakan dengan menggunakan belitan sangkar tupai (*squirrel-cage*) yang dipasang di hadapan kutub rotor. Motor kemudian dinyalakan seperti halnya motor induksi hingga mencapai 95% kecepatan sinkron, saat mana arus searah diberikan, dan motor mencapai sinkronisasi. *Torque* yang diperlukan untuk menarik motor hingga mencapai sinkronisasi disebut *pull-in torque*.

Seperti diketahui, rotor motor sinkron terkunci dengan medan putar dan harus terus beroperasi pada kecepatan sinkron untuk semua keadaan beban. Selama kondisi tanpa beban (*no-load*), garis tengah kutub medan putar dan kutub medan dc berada dalam satu garis. Seiring dengan pembebanan, ada pergeseran kutub rotor ke belakang, relative terhadap kutub stator. Tidak ada perubahan kecepatan. Sudut antara kutub rotor dan stator disebut sudut *torque*.

## 2.6 LIMIT SWITCH

*Switch* adalah suatu alat untuk membuka dan menutup suatu rangkaian listrik tegangan yang masuk ke dalam suatu rangkaian listrik. Posisi membuka adalah *on* dan posisi menutup adalah *off*. Dalam suatu rangkaian listrik *switch* dihubungkan secara antara sumber dan beban. Dalam posisi menutup *switch* mempunyai resistansi yang sangat kecil oleh karena itu arus dapat mengalir secara maksimum ke beban dengan tegangan yang jatuh pada *switch* secara praktis hanya nol (0) volt. Sedangkan jika kita membuka *switch*, maka *switch* akan mempunyai resistansi yang sangat besar mengakibatkan tegangan dari sumber akan jatuh seluruhnya pada *switch*.

Jenis *switch* yang digunakan pada perangkat RIA Nucleus Model 1600 Gamma Counting adalah ***Limit Switch***.

*Limit Switch* adalah sensor peraba yang bersifat mekanis dan mendeteksi sesuatu setelah terjadi kontak fisik. Penggunaan sensor ini biasanya digunakan untuk membatasi gerakan maksimum sebuah mekanik.

*Limit switch* ini digunakan untuk mengoperasikan saklar hanya dengan bergeser dengan jarak perpindahan yang sangat kecil. Saklar jenis ini sangat sensitif. Sedikit tekanan saja pada tuas dapat mengakibatkan saklar berpindah dari satu posisi ke posisi lainnya. Kebanyakan *limit switch* memiliki kontak - kontak jenis SPDT (*Single Pole Double Trough*), sehingga saklar ini dapat digunakan untuk menyambungkan atau memutuskan sesuatu secara bersamaan. Kontak – kontak sebuah *limit switch* dilengkapi dengan pegas sehingga dalam keadaan normal, kontak jalur bersama tersambung ke kontak yang disebut *normally closed* (NC) dan kontak *normally open* (NO).

*Limit switch* dibuat dari spring pulpen dengan kawat dibagian tengahnya. Apabila kawat tersebut membentur sebuah benda, maka kawat tersebut akan bersentuhan dengan spring sehingga terjadi hubungan pendek. [15]

Berdasarkan pada kondisi normal, *limit switch* dibagi menjadi dua jenis yaitu *normally closed* (NC) dan *normally open* (NO) seperti ditunjukkan pada Gambar 2.15.a dan Gambar 2.15.b.



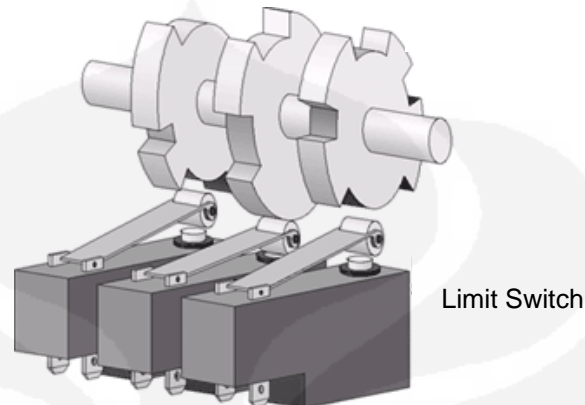
**Gambar 2. 15.a** *Limit Switch Normally Open* (NO).



**Gambar 2. 15.b** *Limit Switch Normally Close* (NC).

Penggunaan *limit switch* ini biasanya digunakan untuk membatasi gerakan maksimum sebuah mekanik, contohnya pada penggerak lengan di mana *limit switch* akan aktif dan memberikan masukan pada CPU untuk menghentikan gerak motor. *Limit switch* dapat juga digunakan untuk mendapatkan informasi logik

pada sebuah system. Sebagai informasi logik *limit switch* dapat digunakan untuk pembacaan suatu posisi pada suatu sistem penggerak. Contoh aplikasi *limit switch* yang digunakan pada sistem kontrol posisi ditunjukkan pada Gambar 2.16.



**Gambar 2. 16** Aplikasi Limit Switch Pada Sistem Kontrol Posisi. [16]

## 2.7 RELAY

Relay adalah saklar magnetis. Relay sering digunakan baik pada industri, otomotif, ataupun peralatan elektronika lainnya. Relay berfungsi untuk menghubungkan atau memutus aliran arus listrik yang dikontrol dengan memberikan tegangan dan arus tertentu pada koilnya. Relay biasanya mempunyai satu kumparan, tetapi relay dapat mempunyai beberapa kontak. Relay elektromekanis berisi kontak diam dan kontak bergerak. Kontak yang bergerak dipasangkan pada *plunger*. Kontak ditunjuk sebagai *Normally Open* ( NO ) dan *Normally Close* ( NC ).

Relay yang digunakan pada skripsi ini adalah Relay DC Omron tipe MY2NJ dengan tegangan koil 12V DC, arus yang diperlukan sekitar 20-30mA.. Bentuk fisik dari Relay Omron tipe MY2NJ ditunjukkan pada Gambar 2.17.



**Gambar 2. 17** Relay Omron type MY2NJ. [17]

Relay pada umumnya tidak bisa langsung menghubungkan *output* suatu IC logic (TTL/CMOS ) atau komponen lain seperti mikrokontroler dengan relay karena arusnya tidak cukup besar. Karena itu perlu digunakan *driver* untuk penguat arus yang biasanya berupa transistor dan dilengkapi dengan diode. Diode ini berfungsi untuk mencegah 'kickback' yaitu *transient* yang terjadi pada koil relay (beban induktif) saat relay dimatikan. Tegangan balik 'kickback' ini sangat besar, dan dapat mengakibatkan kerusakan pada transistor.

## 2.8 TRANSISTOR SEBAGAI SAKLAR

Transistor merupakan salah satu jenis komponen aktif yang banyak digunakan, baik dalam rangkaian analog maupun digital. Transistor yang banyak digunakan adalah transistor bipolar, yang terdiri dari dua jenis yaitu PNP dan NPN. Secara umum transistor digunakan untuk pensaklaran (*switching*) maupun penguatan. Transistor dapat bekerja bila diberi bias. Pembiasan pada transistor dapat dilakukan dengan cara sebagai berikut:

- a. Bias maju (*forward bias*) pada hubungan emitor dan basis, dimana bahan tipe P mendapat positif dan tipe N mendapat negatif.
- b. Bias mundur (*reverse bias*) pada hubungan kolektor dan basis, dimana bahan tipe P mendapat negatif dan tipe N mendapat positif.

Pada transistor terdapat tiga daerah kerja, yaitu:

### 1. Daerah mati (*Cut Off*)

Daerah mati merupakan daerah kerja saat transistor mendapat bias arus basis ( $I_b > 0$ ), maka arus kolektor dengan basis terbuka menjadi arus bocor dari basis ke emitor (ICEO). Hal yang sama dapat terjadi pada transistor hubungan kolektor-basis. Jika arus emitor sangat kecil ( $I_E = 0$ ), emitor dalam keadaan terbuka dan arus mengalir dari kolektor ke basis (ICBO).

### 2. Daerah aktif

Transistor dapat bekerja pada daerah aktif jika transistor mendapat arus basis ( $I_b > 0$ ). Tetapi jika lebih kecil dari arus basis maksimalnya, keluaran arus

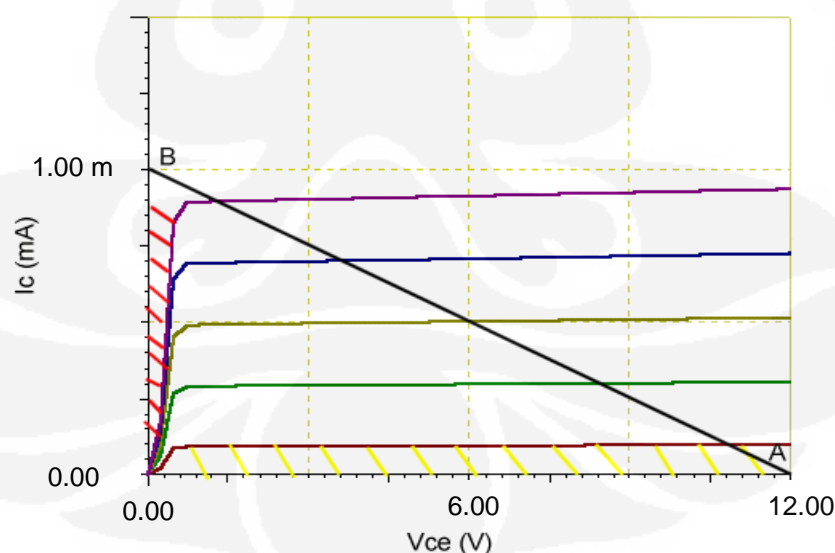
kolektor akan berubah-ubah sesuai dengan perubahan pemberian arus basisnya.

### 3. Daerah jenuh (Saturasi)

Transistor dapat bekerja pada daerah jenuh jika transistor mendapat arus basis ( $I_b$ ) lebih besar dari arus basis maksimalnya. Hal ini menimbulkan keluaran arus kolektor tidak dapat bertambah lagi.

Prinsip pengoperasian transistor sebagai saklar memiliki dua keadaan, yaitu keadaan tidak bekerja (*cut off*) dan keadaan jenuh. Dimana perubahan keadaannya dapat berupa perubahan tegangan ataupun arus.

Saat sebuah transistor digunakan pada suatu rangkaian, fungsi dari transistor tersebut ditentukan oleh kurva karakteristik-nya. Transistor memiliki kurva karakteristik input, output dan transfer, yang paling umum digunakan adalah kurva karakteristik output. Pada saat Transistor digunakan sebagai saklar, maka daerah yang digunakan pada kurva karakteristik ialah daerah "*cut-off*" dan daerah "*saturasi*". Gambar kurva karakteristik Transistor ditunjukkan pada Gambar 2.18.



**Gambar 2. 18** Kurva Karakteristik Transistor

Daerah yang diarsir kuning adalah daerah "*cut-off*". Pada saat "*cut-off*" kondisi dari transistor adalah arus basis sama dengan nol ( $I_B = 0$ ), Arus output pada kolektor sama dengan nol dan Tegangan pada kolektor maksimum atau sama dengan tegangan supply ( $V_{CE} = V_{CC}$ ).

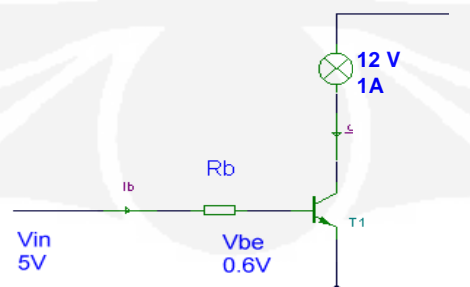
Daerah yang diarsir merah adalah daerah "*saturasi*". Pada saat "*saturasi*" kondisi dari transistor adalah arus basis maksimal ( $I_B = I_{B_{max}}$ ) sehingga menghasilkan arus kolektor maksimal ( $I_C = I_{C_{max}}$ ) dan tegangan Kolektor Emitor minimum ( $V_{CE} = 0$ ).

### Garis Beban.

Garis beban dapat dibangun apabila kita mengetahui arus beban pada rangkaian dan tegangan operasinya. Titik "A" pada Kurva Gambar 2.18 adalah kondisi saat Saat transistor OFF,  $I_C$  (arus kolektor) akan menjadi nol sedangkan  $V_{CE}$  (tegangan kolektor-emitor) akan menjadi hampir sama dengan tegangan supply. Pada Kurva Gambar 2.18 adalah kondisi saat transistor ON dimana  $I_C$  akan menjadi 1mA (sama dengan arus beban) dan  $V_{CE}$  nilainya sangat kecil hampir mendekati nol. Garis yang ditarik dari titik A ke titik B ini yang dinamakan garis beban.

### Rumus Perhitungan Transistor Sebagai Saklar.

Misalnya, sebuah transistor dengan tegangan supply 5V DC digunakan untuk mensaklar sebuah lampu 12V DC 1A. Transistor dipilih bervariasi dengan variasi  $h_{fe}$  dari 100 - 500. Rangkaian pada Gambar 2.19 menggunakan konfigurasi common-emitor. Tentukan nilai  $R_b$  (tahanan basis) agar transistor dapat bekerja pada kelompok penguatan yang sama.



**Gambar 2.19** Konfigurasi Transistor sebagai saklar

Saat  $V_{in} = 0$ , maka tidak ada arus yang mengalir pada  $R_b$  dan basis transistor sehingga transistor dalam kondisi tidak bekerja. Tidak ada arus yang mengalir kecuali arus bocor, sehingga kondisi ini identik dengan saklar terbuka (sambungan C-E terpisah) dan menyebabkan beban RL tidak bekerja. Saat  $V_{in}$  mendapat masukan yang cukup besar hingga dapat mengalirkan arus basis yang cukup untuk transistor, maka transistor akan jenuh. Pada kondisi ini arus kolektor akan mengalir (sambungan C-E) terhubung dan menyebabkan beban RL akan bekerja.

Pada saat transistor jenuh arus yang mengalir pada beban RL adalah:

$$I_{RL} = \frac{V_{CC} - V_{CE}}{R_L}$$

Besarnya arus basis dapat dicari dengan persamaan:

$$I_B = \frac{V_{IN} - V_{BE}}{R_B}$$

Karena transistor mempunyai hfe antara 100 - 500 maka kita pilih dulu menggunakan hfe minimum ( 100 ). Arus kolektor adalah 1A, maka Arus Basis yang dibutuhkan adalah:

$$hfe = I_c / I_b$$

$$I_b = I_c / hfe(\min) = 1/100 = 1\text{mA}$$

Nilai  $V_{in}$  adalah 5V DC, nilai  $V_{be}$  adalah 0,6V DC (konstanta) berarti tegangan yang melewati  $R_b$  adalah  $V_{in} - V_{be} = 4,4 \text{ V DC}$ . Sehingga Nilai  $R_b$  dapat kita hitung:

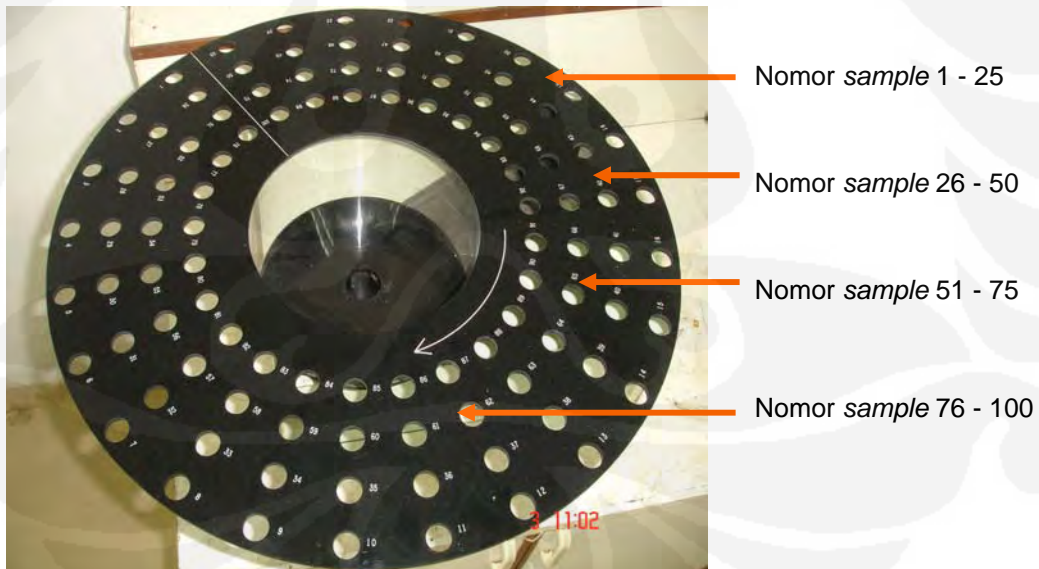
$$R_b = 4.4 / 1 = 4.4\text{K} = 4\text{K}4$$

## BAB 3 RANCANG BANGUN SISTEM

### 3.1 PRINSIP KERJA SISTEM

Sistem kontrol *sample changer* perangkat *Radioimmunoassay* (RIA) Nucleus Model 1600 Gamma Counting menggunakan satu modul *driver* motor, satu modul *interface* USB dan dua motor AC Sinkron sebagai penggerak kolimator *detector* secara horizontal (mendatar) dan penggerak tempat *sample* (*sample tray*) secara vertikal (naik turun) sekaligus rotasi (memutar).

Pada perangkat RIA Nucleus Model 1600 Gamma Counting memiliki *sample tray* yang bentuknya berupa lingkaran dengan lubang *sample* yang berjumlah 100, dimana dari 100 lubang *sample* dibagi menjadi empat bagian. Bagian pertama adalah nomor *sample* 1 – 25, bagian kedua adalah nomor *sample* 26 -50, bagian ketiga adalah nomor *sample* 51 – 75 dan bagian keempat adalah nomor *sample* 76 – 100. Adapun bentuk fisik dari *sample tray* perangkat RIA Nucleus Model 1600 Gamma Counting ditunjukkan pada Gambar 3.1.



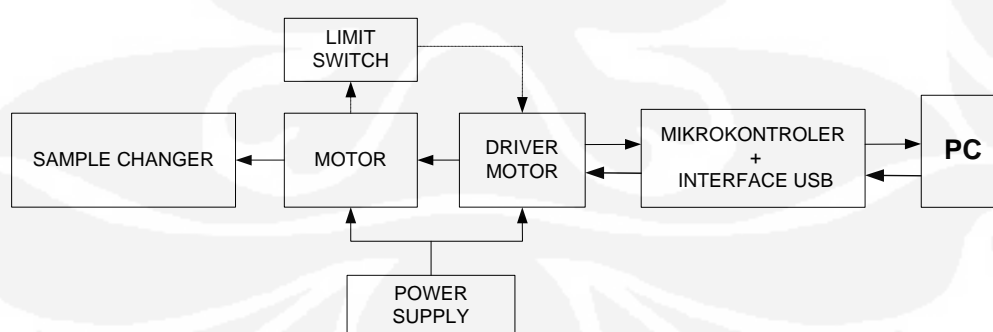
**Gambar 3.1.** *Sample Tray* perangkat RIA Nucleus Model 1600 Gamma Counting



Pergerakan kolimator *detector* terdiri dari empat posisi, dimana posisi pertama untuk melakukan pencacahan pada *range* nomor *sample* 1 – 25, posisi kedua untuk melakukan pencacahan pada *range* nomor *sample* 26 – 50, posisi ketiga untuk melakukan pencacahan pada *range* nomor *sample* 51 – 75, dan posisi keempat untuk melakukan pencacahan pada *range* nomor *sample* 76 – 100. Pada setiap posisi kolimator *detector* terdapat *limit switch* sebagai sensor posisi.

Pencacahan *sample* dilakukan saat *sample tray* turun dan tabung reaksi yang terdapat pada lobang *tray* masuk ke *detector*. Pada saat *sample tray* turun dan menyentuh *limit switch*, maka *limit switch* akan memberikan informasi ke *Personal Computer* (PC), sehingga PC akan membaca informasi dan memberikan perintah untuk mulai melakukan pencacahan (*start counting*). Lamanya waktu pencacahan diatur melalui program di PC dengan *Visual Basic*. Setelah selesai melakukan pencacahan, PC kembali memberikan perintah untuk menghentikan pencacahan (*stop counting*), kemudian *sample tray* bergerak naik. Jika tidak ada *sample* yang akan di cacah posisi *tray* berhenti di atas setelah menyentuh *limit switch* posisi atas dan memberikan informasi mengenai posisi nomor *sample* berikutnya. Apabila masih terdapat *sample* yang akan dicacah, maka *sample tray* akan bergerak ke nomor *sample* yang dituju.

Secara keseluruhan sistem terdiri atas beberapa bagian yang dapat digambarkan menjadi blok diagram pada Gambar 3.2.



**Gambar 3.2.** Blok diagram sistem keseluruhan

Secara garis besar, sistem dibagi menjadi lima bagian yaitu PC, *interface USB*, *driver motor*, perangkat mekanik *sample changer* dan *power supply*.

Fungsi dari masing – masing bagian adalah:

1. PC

PC digunakan sebagai pengolah data.

2. *Interface USB*

*Interface USB* adalah modul elektronik yang digunakan sebagai media komunikasi antara PC dan modul *driver*.

3. *Driver Motor*

*Driver motor* adalah modul elektronik hasil perancangan yang digunakan sebagai pengatur gerak mekanik *sample changer* .

4. Perangkat Mekanik *Sample Changer*

Perangkat mekanik *sample changer* adalah bagian mekanik dari system.

5. *Power Supply*

*Power supply* digunakan sebagai pencatu daya pada modul *driver*.

## 3.2 PERANGKAT KERAS

Perangkat keras yang digunakan pada sistem secara keseluruhan menggunakan komponen-komponen sesuai dengan fungsi dari blok-blok sistem yang digambarkan pada Gambar 3.2.

### 3.2.1 *Personal Computer (PC)*

PC merupakan bagian dari sistem kontrol *sample changer*. PC digunakan sebagai pengolah data untuk pengendali gerak *sample changer* dengan pemrograman *Visual Basic* versi 6.0.

### 3.2.2 *Interface USB*

Modul *interface USB* yang digunakan adalah modul Devasys I2C/IO. Modul Devasys USB I2C/IO adalah sebuah modul antar muka yang dikembangkan oleh Devasys dengan mengadopsi mikrokontroler buatan Cypress AN2131QC yang merupakan salah satu dari keluarga EZ-USB. Mikrokontroler ini merupakan mikrokontroler yang mempunyai USB *kernel* dan merupakan pengembangan dari mikroprocessor 8051 dengan penambahan fasilitas dan waktu

eksekusi yang lebih cepat. Modul Devasys memiliki satu buah port I2C dan tiga buah port I/O, masing-masing 8 bit Port A, 8 bit Port B dan 8 bit Port C.

Pertukaran data antara PC dengan modul Devasys dilakukan melalui Port B, Port C dan Port I2C. Port B digunakan untuk mengontrol status *limit switch* horizontal, Port C digunakan untuk mengontrol gerak motor dan Port I2C dengan menggunakan IC PCF8574 sebagai *expander I/O* digunakan untuk mengontrol status *limit switch* posisi nomor *sample*.

### 3.2.2.1 Interkoneksi Port B

Port B dari modul Devasys yang digunakan hanya 5 bit (PB0 – PB4). Secara fungsi penggunaan Port B dari modul Devasys ditunjukkan pada Tabel 3.1.

**Tabel 3.1** Fungsi Penggunaan Port B

Port B	Fungsi
PB0	Mengontrol status <i>limit switch</i> horizontal 1 (SH1).
PB1	Mengontrol status <i>limit switch</i> horizontal 2 (SH2).
PB2	Mengontrol status <i>limit switch</i> horizontal 3 (SH3).
PB3	Mengontrol status <i>limit switch</i> horizontal 4 (SH4).
PB4	Mengontrol status <i>limit switch</i> posisi atas

### 3.2.2.2 Interkoneksi Port C

Port C dari modul Devasys yang digunakan hanya 3 bit (PC4 – PC6). Secara fungsi penggunaan Port C dari modul Devasys ditunjukkan pada Tabel 3.2.

**Tabel 3.2** Fungsi Penggunaan Port C

Port C	Fungsi
PC4	Mengontrol gerak motor arah vertikal (naik/turun)
PC5	Mengontrol gerak motor arah horizontal (ON/OFF)
PC6	Mengontrol gerak motor arah horizontal (kiri/kanan)

### 3.2.2.3 Interkoneksi Port I2C

Port I2C dari modul Devasys menggunakan satu buah IC PCF8574 sebagai *expander I/O*. Secara fungsi penggunaan Port I2C dari modul Devasys ditunjukkan pada Tabel 3.3.

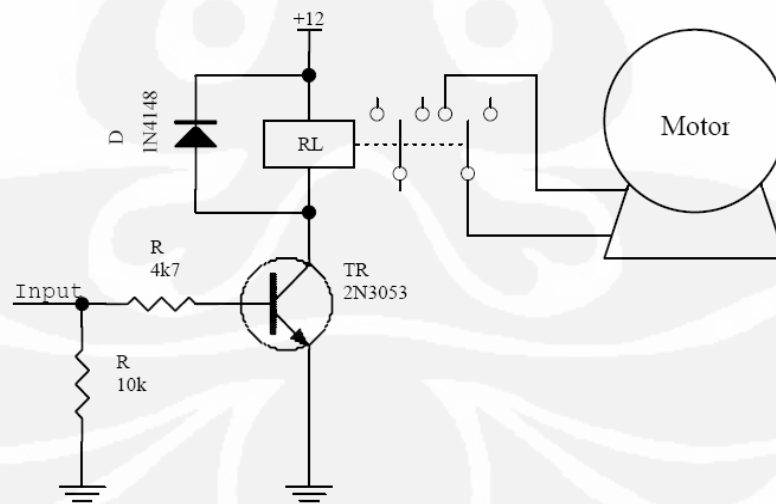
**Tabel 3.3** Fungsi Penggunaan Port I2C

Port I2C	Fungsi
D0	Mengontrol status <i>limit switch</i> vertical 1 (SV1).
D1	Mengontrol status <i>limit switch</i> vertical 2 (SV2).
D2	Mengontrol status <i>limit switch</i> vertical 3 (SV3).
D3	Mengontrol status <i>limit switch</i> vertical 4 (SV4).
D4	Mengontrol status <i>limit switch</i> vertical 5 (SV5).
D5	Mengontrol status <i>limit switch</i> posisi atas

### 3.2.3 Driver Motor

*Driver* motor adalah modul elektronik hasil perancangan yang digunakan sebagai pengatur gerak mekanik *sample changer*. Modul *driver* ini terdiri dari 2 bagian, yaitu bagian pengatur gerak kolimator *detector* arah horizontal dan bagian pengatur gerak *sample tray* arah memutar sekaligus naik/turun.

Komponen utama rancangan *driver* motor ini adalah transistor tipe NPN 2N3053 dan *relay* Omron tipe MY2NJ SPDT yang bekerja pada tegangan 12VDC. Transistor 2N3053 pada rangkaian ini berfungsi sebagai saklar pengatur jalan masukan untuk mengaktifkan *relay*, sedangkan *relay* berfungsi sebagai pengkondisi On/Off nya motor. Adapun rangkaian dasar rancangan *driver* motor ditunjukkan pada Gambar 3.3.

**Gambar 3.3.** Rangkaian Dasar *Driver* Motor

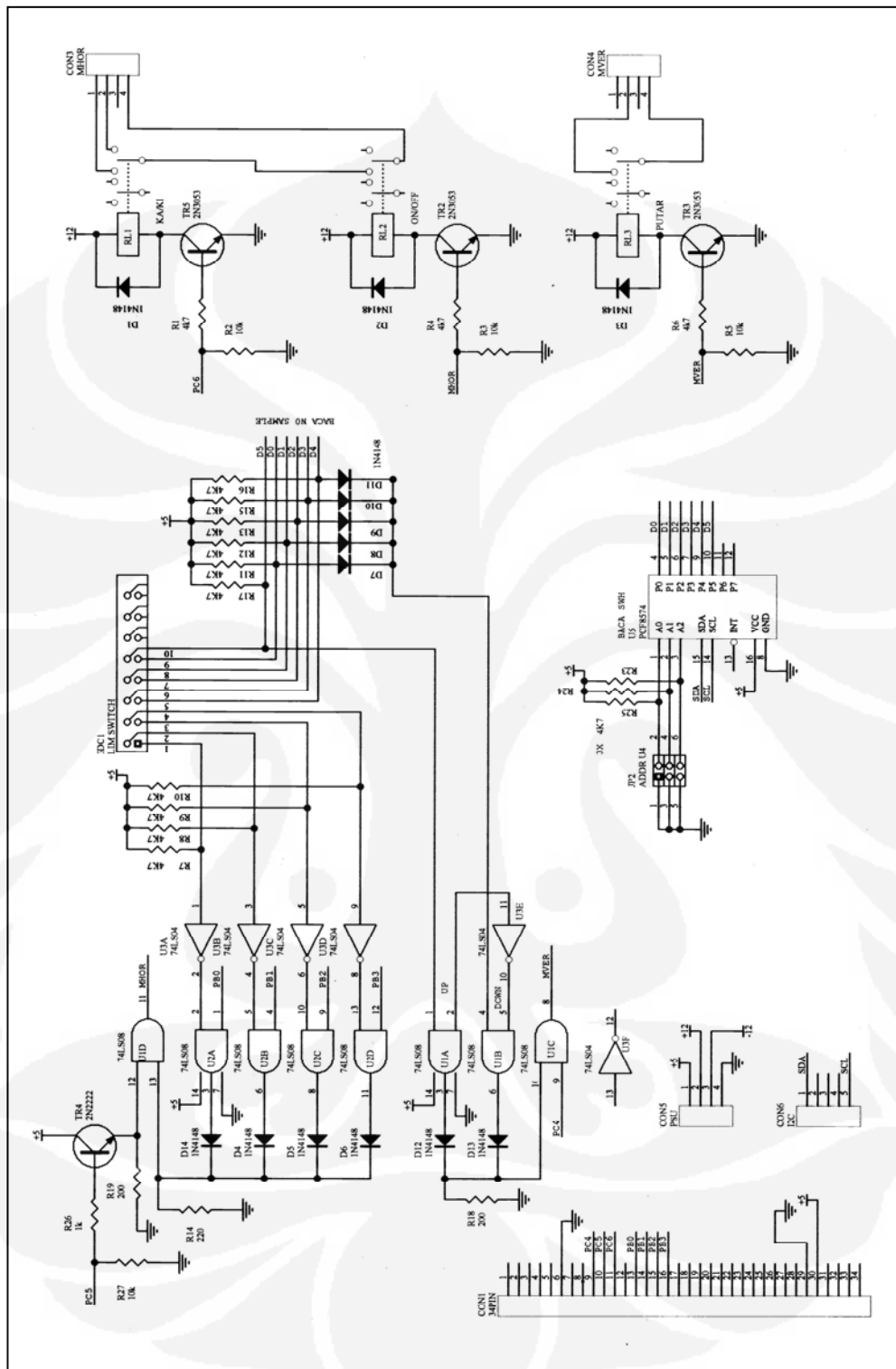
Dari rangkaian dasar rancangan *driver* motor pada Gambar 3.3, maka dirancang secara utuh rangkaian *driver* motor sebagai pengatur gerak mekanik *sample changer* perangkat RIA. Pada rangkaian *driver* motor ini digunakan beberapa tambahan komponen sebagai pendukung kerja sistem, diantaranya adalah IC 74LS08 (gerbang logika AND), IC 74LS04 (gerbang logika OR) dan IC PCF8574 sebagai *Expander I/O*. Secara utuh rangkaian *driver* motor ditunjukkan pada Gambar 3.4.

### 3.2.4 Perangkat Mekanik *Sample Changer*

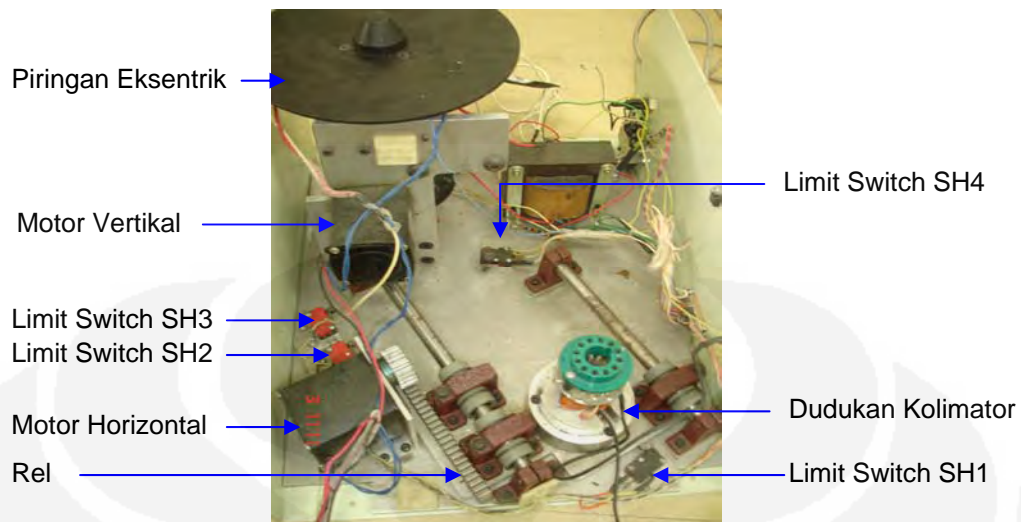
Perangkat mekanik *sample changer* pada perangkat RIA Nucleus model 1600 Gamma Counting terdiri atas beberapa komponen, yaitu :

- a. 2 buah motor AC
- b. 4 buah *limit switch Normally Close* (NC) sebagai sensor posisi kolimator *detector*.
- c. 5 buah *limit switch Normally Open* (NO) sebagai sensor posisi nomor *sample*.
- d. 1 buah *limit switch Normally Open* (NO) sebagai sensor posisi atas dari *sample tray*.
- e. Rel dan tuas penahan sebagai jalur pergerakan kolimator *detector*.
- f. Kolimator *Detector*
- g. *Sample Tray* dengan 100 lobang tempat *sample*.
- h. Piringan Eksentrik
- i. Mekanik pegas sebagai pemutar piringan eksentrik

Pada perangkat mekanik secara prinsip tidak dilakukan perubahan, hanya penambahan kapasitor pada kedua motor. Adapun gambar perangkat mekanik ditunjukkan pada Gambar 3.5.



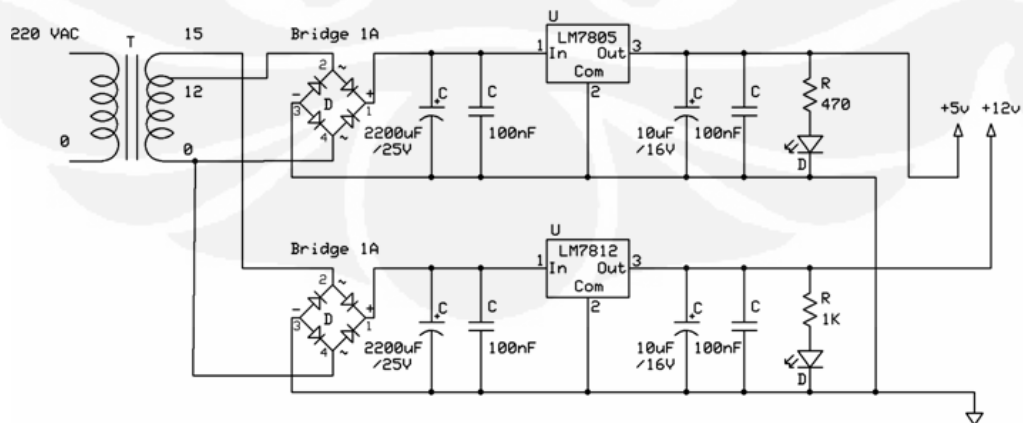
Gambar 3.4. Rangkaian driver motor Perangkat RIA Nuclues Model 1600 Gamma Counting



**Gambar 3.5.** Perangkat Mekanik *Sample Changer* Perangkat RIA Nuclues Model 1600

### 3.2.5 Power Supply

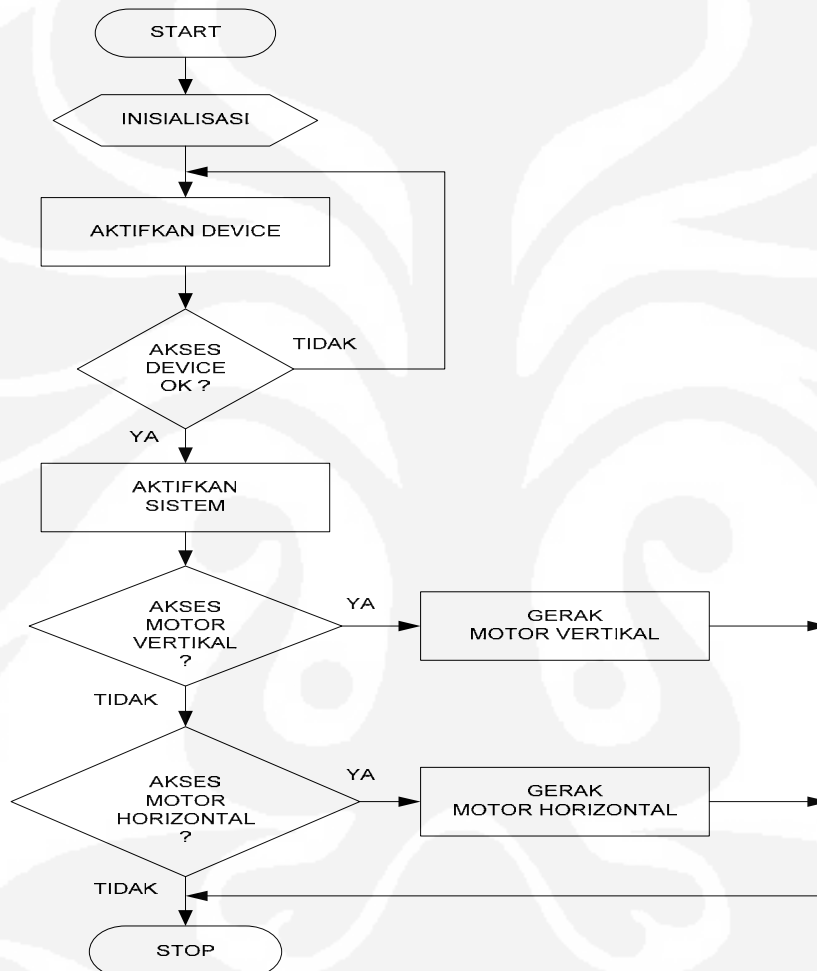
*Power Supply* merupakan bagian dari sistem yang tidak dapat dipisahkan. Pada sistem kontrol *sample changer* perangkat RIA Nucleus model 1600 Gamma Counting digunakan dua jenis *power supply*, yaitu *power supply* tegangan 220 VAC yang di turunkan menjadi tegangan 110 VAC dengan trafo *step down* untuk memberikan catu daya pada motor dan *power supply* DC dengan tegangan keluaran (+12, +5) VDC yang digunakan untuk memberikan catu daya pada modul elektronik. *Power supply* tegangan DC menggunakan *power supply* yang biasa digunakan pada PC. Dipilihnya *power supply* ini dengan alasan karena tegangan keluarannya stabil. Secara umum rangkaian *power supply* dengan *output* tegangan +5 dan +12VDC ditunjukkan pada Gambar 3.6.



**Gambar 3.6.** Rangkaian *Power Supply* +5 dan +12 VDC

### 3.3 PERANGKAT LUNAK

Pemrograman dibuat dengan menggunakan bahasa pemrograman *Visual Basic* versi 6.0. Program dibuat dan disesuaikan sehingga sistem dapat berjalan dengan baik. Secara umum diagram alir utama program sistem kontrol *sample changer* pada perangkat RIA Nucleus model 1600 Gamma Counting terlihat seperti pada Gambar 3.7.



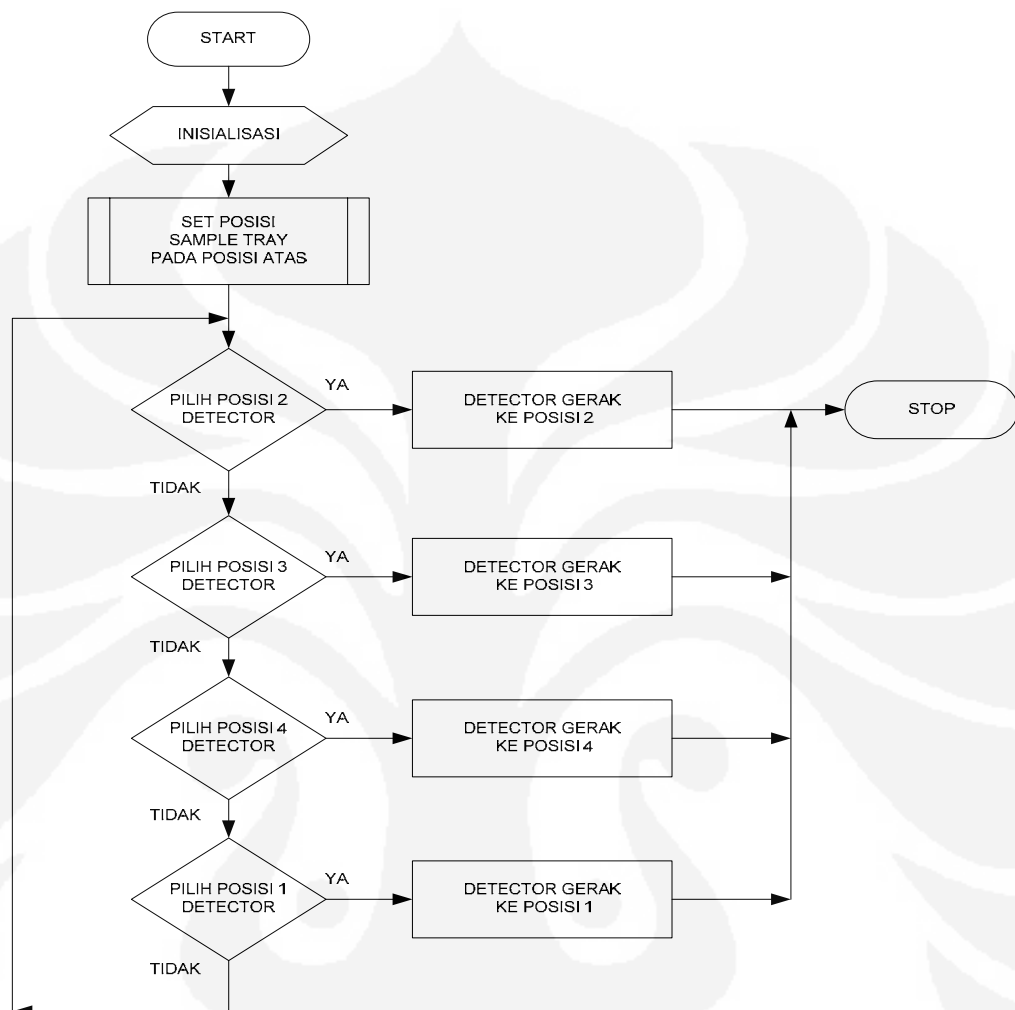
Gambar 3.7. Diagram alir kerja sistem

#### 3.3.1 Pemrograman Gerak Motor Horizontal

Dalam perancangan ini gerak motor ke arah kanan ataupun ke arah kiri dilakukan dengan membandingkan data yang diterima mikrocontroller dengan data hasil pembacaan *Limit switch* posisi kolimator *detector*. Untuk dapat menggerakkan motor ke arah kanan ataupun ke arah kiri, posisi *sample tray* harus



berada di posisi atas. Secara umum diagram alir dari program gerak motor horizontal adalah seperti pada Gambar 3.8.



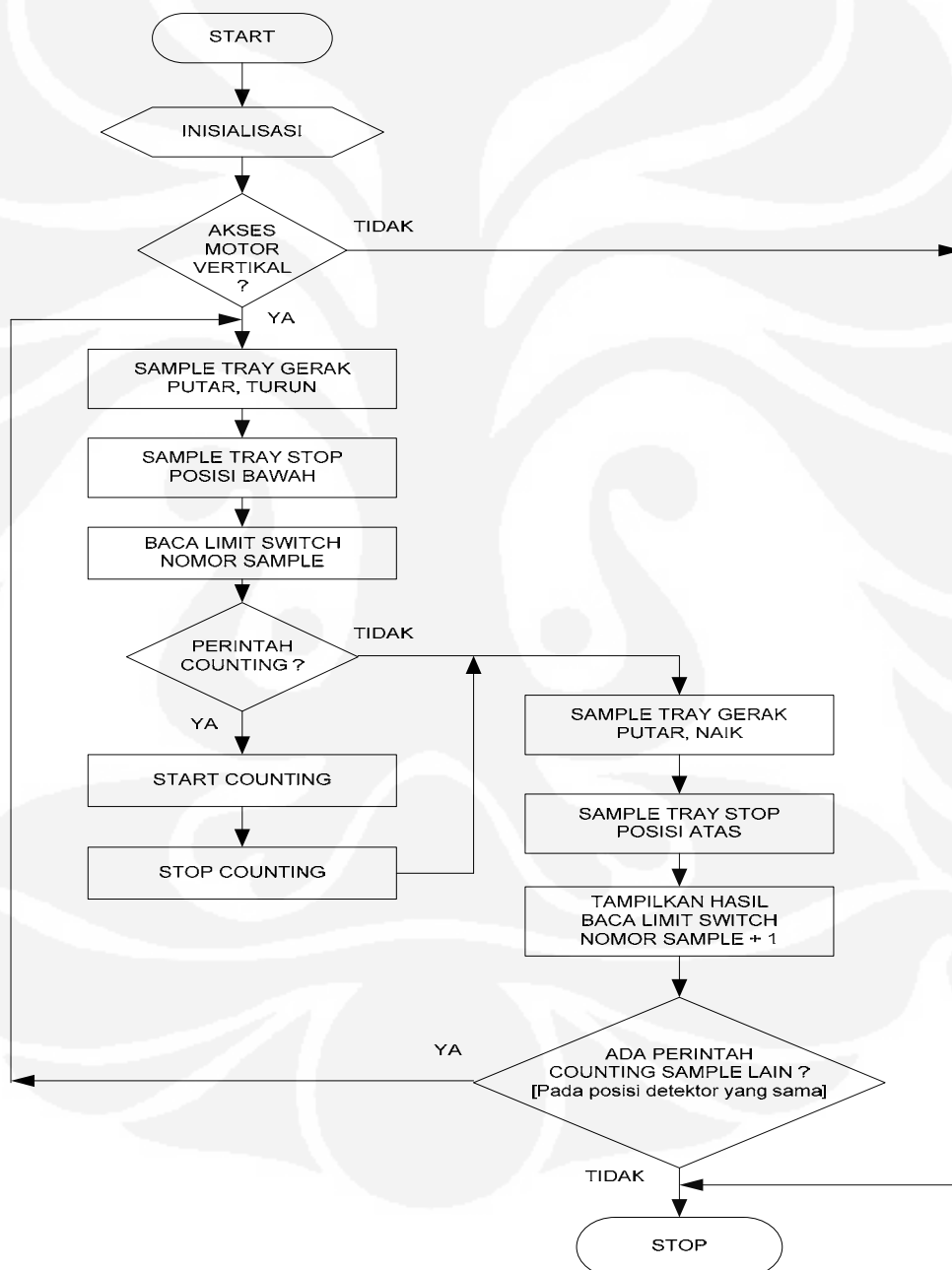
Gambar 3.8. Diagram alir program gerak motor horizontal

### 3.3.2 Pemrograman Gerak Motor Vertikal

Dalam perancangan ini gerakan motor vertikal dilakukan dengan memberikan *input* aktif ("1") pada PC4. Saat *sample tray* di posisi bawah dan menyentuh *limit switch* nomor *sample*, maka mikrokontroller akan membandingkan hasil pembacaan *limit switch* nomor *sample* dengan nomor *sample* yang dituju. Jika hasilnya cocok, maka *sample tray* akan berhenti bergerak dan proses *counting sample* akan berjalan selama waktu yang ditentukan.

Setelah proses *counting sample* selesai, motor akan kembali berputar menggerakkan *sample tray* ke atas, dan saat *sample tray* menyentuh *limit switch* posisi atas, maka akan ditampilkan hasil pembacaan *limit switch* posisi nomor *sample* ke layar monitor sebagai informasi posisi nomor *sample* berikutnya.

Jika tidak ada lagi proses *counting sample*, maka *sample tray* akan berhenti bergerak di posisi atas. Secara umum diagram alir dari program gerak motor vertikal adalah seperti pada Gambar 3.9.



**Gambar 3.9.** Diagram alir program gerak motor vertikal

### 3.3.3 Pemrograman Bagian Utama Sistem

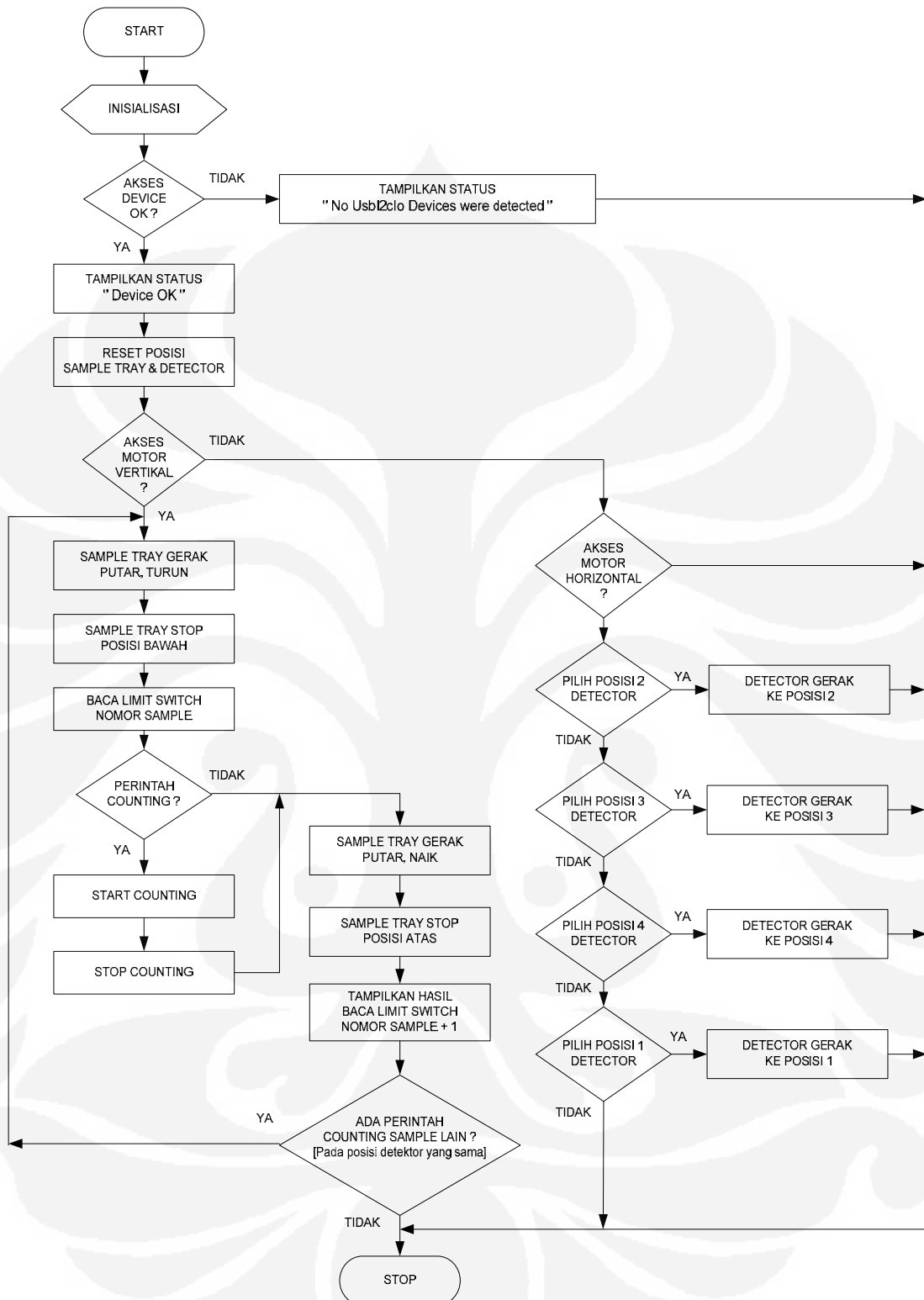
Pada pemrograman sistem kontrol *sample changer* pada perangkat RIA Nucleus model 1600 Gamma Counting terdapat satu buah *Form* yaitu *ControlForm* dan empat buah *modules* program, yaitu *modUsbI2cIoAPI*, *modWinAPI*, *Module1a* dan *MyModul*. *Modules* *modUsbI2cIoAPI* dan *modules* *modWinAPI* adalah *modules* aplikasi yang telah disediakan oleh Devasys, sedangkan *Modules* *Module1a* dan *modules* *MyModul* adalah *modules* aplikasi yang dibuat oleh penulis.

Secara fungsi *modules* *modUsbI2cIoAPI* ini mengandung definisi, jenis, dan deklarasi untuk mengakses fungsi yang disediakan oleh *UsbI2cIo API dll*, dan *modules* *modWinAPI* File mengandung definisi, jenis, dan deklarasi untuk mengakses fungsi yang disediakan oleh *Win32 API Windows*.

*Modules* *Module1a* adalah *modules* aplikasi yang dibuat dengan fungsi deklarasi untuk inialisasi data. Sedangkan *modules* *MyModul* adalah *modules* aplikasi yang dibuat dengan fungsi deklarasi untuk inialisasi *device* dan untuk mengakses fungsi *ReadIoPorts*, *WriteI2cIo*, dan *ReadI2cIo*.

*ControlForm* adalah *Form* yang berisi perintah - perintah untuk menjalankan menu pada system.

Beberapa saat ketika perangkat diaktifkan, pada layar monitor akan muncul tulisan "Device OK" yang menandakan bahwa sistem perangkat *Sample Changer* telah dikenali oleh PC dan perangkat *Sample Changer* dalam posisi *default*, yaitu posisi kolimator *detector* pada posisi 1 dan *sample tray* pada posisi nomor *sample* 1. Selanjutnya sistem akan berjalan sesuai dengan masukan yang diberikan. Adapun diagram alir dari program utama sistem adalah seperti ditunjukkan pada Gambar 3.10.



Gambar 3.10. Diagram alir program utama sistem

## BAB 4 PENGUJIAN DAN ANALISA SISTEM

### 4.1 PENGUJIAN MODUL ELEKTRONIK

#### 4.1.1 Pengujian Modul *Interface* USB Devasys I2C I/O

Pengujian modul *interface* USB Devasys I2C I/O adalah pengujian modul *interface* yang digunakan sebagai media komunikasi antara perangkat sistem dengan PC dan sebagai *base* sistem kontrol dalam sistem ini. Tujuan dari pengujian modul *interface* USB Devasys I2C I/O adalah untuk mengetahui apakah modul *interface* USB Devasys I2C I/O yang digunakan dapat bekerja dengan baik. Sebelum melakukan pengujian dilakukan penginstalan *driver* modul *interface* USB Devasys I2C I/O pada PC. Saat program dijalankan, PC akan mendeteksi keberadaan *device* (perangkat). Indikator yang menunjukkan berhasil atau tidaknya modul dikenali oleh PC adalah pada pesan yang ditampilkan di monitor. Jika modul *interface* USB Devasys I2C I/O berhasil dikenali oleh PC akan tampil pesan “ Device OK”, sedangkan bila tidak berhasil maka akan tampil pesan “ No UsbI2cIo Device were detected”. Indikator yang menunjukkan berhasil atau tidaknya modul dikenali oleh PC ditunjukkan pada Gambar 4.1. Rangkaian pengujian modul *interface* USB Devasys I2C I/O terlihat seperti Gambar 4.2. Pengujian dilakukan berulang-ulang sebanyak sepuluh kali dengan mengamati berhasil atau tidaknya modul *interface* USB Devasys I2C I/O dikenali oleh PC dan mencatat hasil kerja dari pengujian modul *interface* USB Devasys I2C I/O pada Tabel 4.1.



**Gambar 4.1.** Indikator yang menunjukkan berhasil atau tidaknya modul *interface* USB Devasys I2C I/O dikenali oleh PC

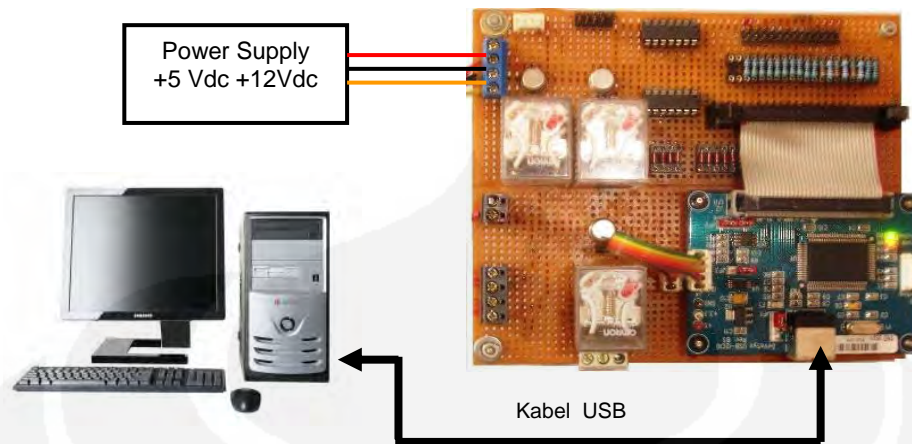
**Tabel 4.1.** Hasil pengujian modul *interface* USB Devasys I2C I/O

Pengujian Ke	Hasil pengujian	
	<i>Device</i> terkoneksi dengan PC	<i>Device</i> tidak terkoneksi dengan PC
1	Terdeteksi	Tidak terdeteksi
2	Terdeteksi	Tidak terdeteksi
3	Terdeteksi	Tidak terdeteksi
4	Terdeteksi	Tidak terdeteksi
5	Terdeteksi	Tidak terdeteksi
6	Terdeteksi	Tidak terdeteksi
7	Terdeteksi	Tidak terdeteksi
8	Terdeteksi	Tidak terdeteksi
9	Terdeteksi	Tidak terdeteksi
10	Terdeteksi	Tidak terdeteksi

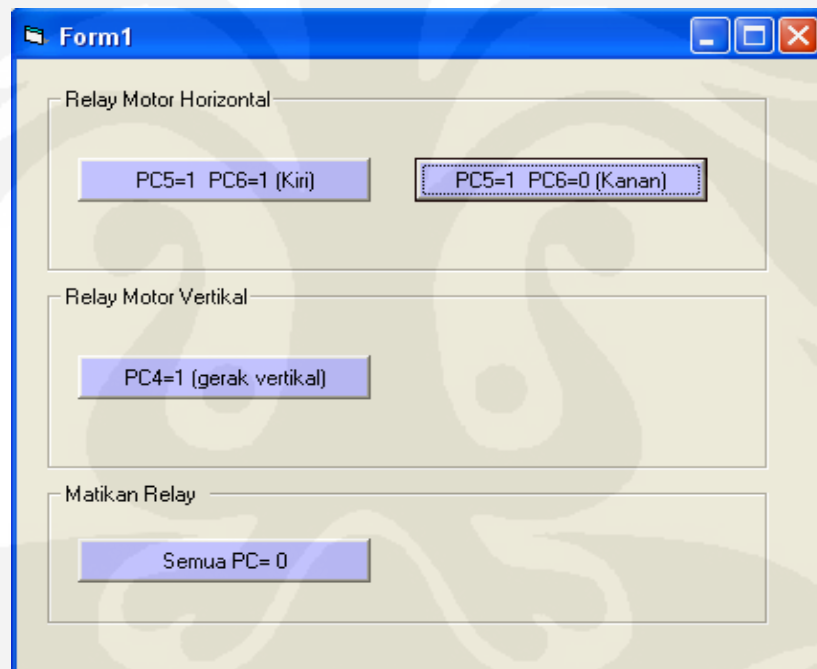
Hasil pengujian seperti ditunjukkan pada Tabel 4.1 menjelaskan bahwa setelah dilakukan pengujian sebanyak sepuluh kali modul *interface* USB Devasys I2C I/O dapat bekerja dan dapat dikenali oleh PC dengan baik.

#### 4.1.2 Pengujian Modul *Driver* Motor

Pengujian modul *driver* motor adalah pengujian modul *driver* motor hasil perancangan yang digunakan untuk menggerakkan motor dalam sistem ini. Tujuan dari pengujian modul *driver* motor adalah untuk mengetahui apakah modul *driver* motor hasil perancangan dapat bekerja dan dapat dikendalikan oleh sistem kontrol dengan baik. Rangkaian pengujian modul *driver* motor terlihat seperti Gambar 4.2, dengan sumber tegangan +5 Vdc dan +12 Vdc. Pengujian modul *driver* motor dilakukan dengan menggunakan program sederhana yang dibuat dengan pemrograman *visual basic 6.0*. Tampilan program pengujian modul *driver* motor ditunjukkan pada Gambar 4.3. Program tersebut dibuat untuk mengaktifkan ketiga Relay yaitu Relay motor vertikal, Relay *on/off* motor horizontal dan Relay ka/ki motor horizontal.

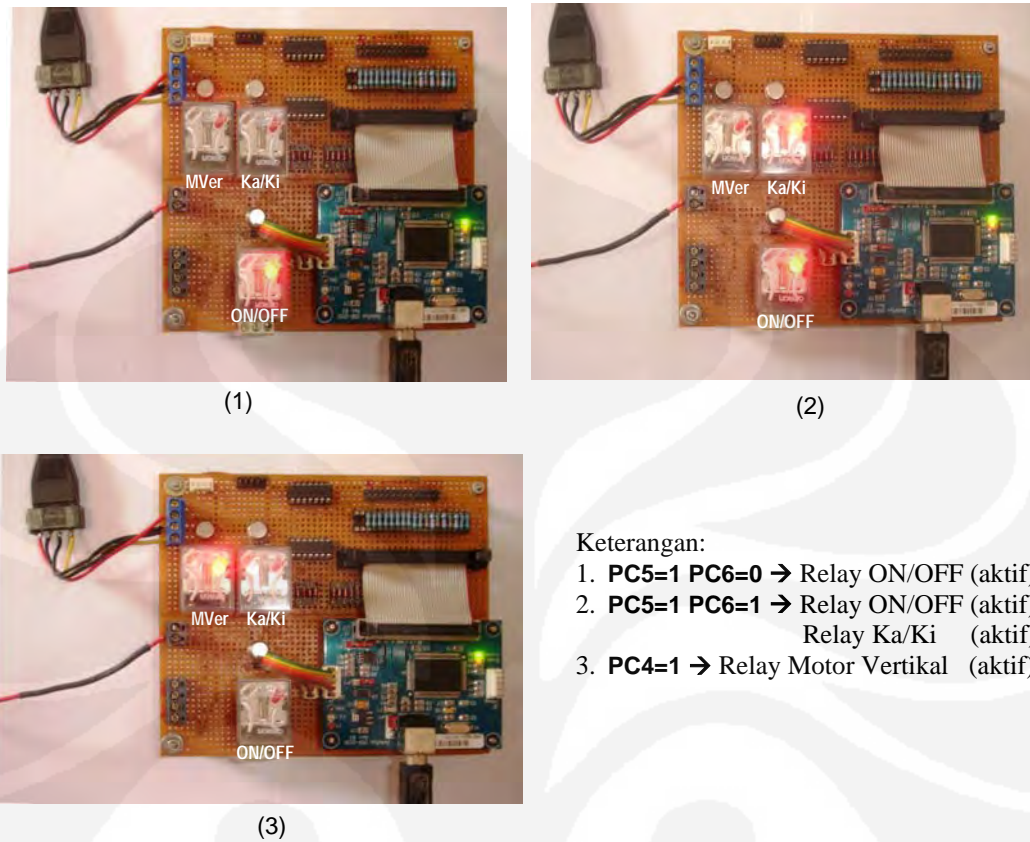


**Gambar 4.2.** Rangkaian pengujian modul *interface* USB Devasys I2C I/O dan modul *driver* motor



**Gambar 4.3.** Tampilan program pengujian modul *driver* motor

Hasil pengujian modul *driver* motor dapat dilihat pada Gambar 4.4



Keterangan:

1.  $PC5=1$   $PC6=0$  → Relay ON/OFF (aktif)
2.  $PC5=1$   $PC6=1$  → Relay ON/OFF (aktif)  
Relay Ka/Ki (aktif)
3.  $PC4=1$  → Relay Motor Vertikal (aktif)

**Gambar 4.4.** Hasil pengujian modul *driver* motor

Pengujian dilakukan berulang-ulang sebanyak sepuluh kali dengan mengamati kerja ketiga Relay dan mencatat hasil kerja dari pengujian modul *driver* motor pada Tabel 4.2.

**Tabel 4.2.** Hasil pengujian modul *driver* motor.

Pengujian Ke	Hasil			
	Relay ON/OFF $PC5=1$ $PC6=0$	Relay Ka/Ki $PC5=1$ $PC6=0$	Relay Motor Vertikal $PC4=1$	Matikan Relay Semua $PC=0$
1	ON	ON	ON	OFF
2	ON	ON	ON	OFF
3	ON	ON	ON	OFF
4	ON	ON	ON	OFF
5	ON	ON	ON	OFF
6	ON	ON	ON	OFF
7	ON	ON	ON	OFF
8	ON	ON	ON	OFF
9	ON	ON	ON	OFF
10	ON	ON	ON	OFF



Hasil pengujian seperti ditunjukkan pada Tabel 4.2 menjelaskan bahwa modul *driver* motor hasil perancangan dapat bekerja dan dapat dikendalikan oleh PC dengan baik.

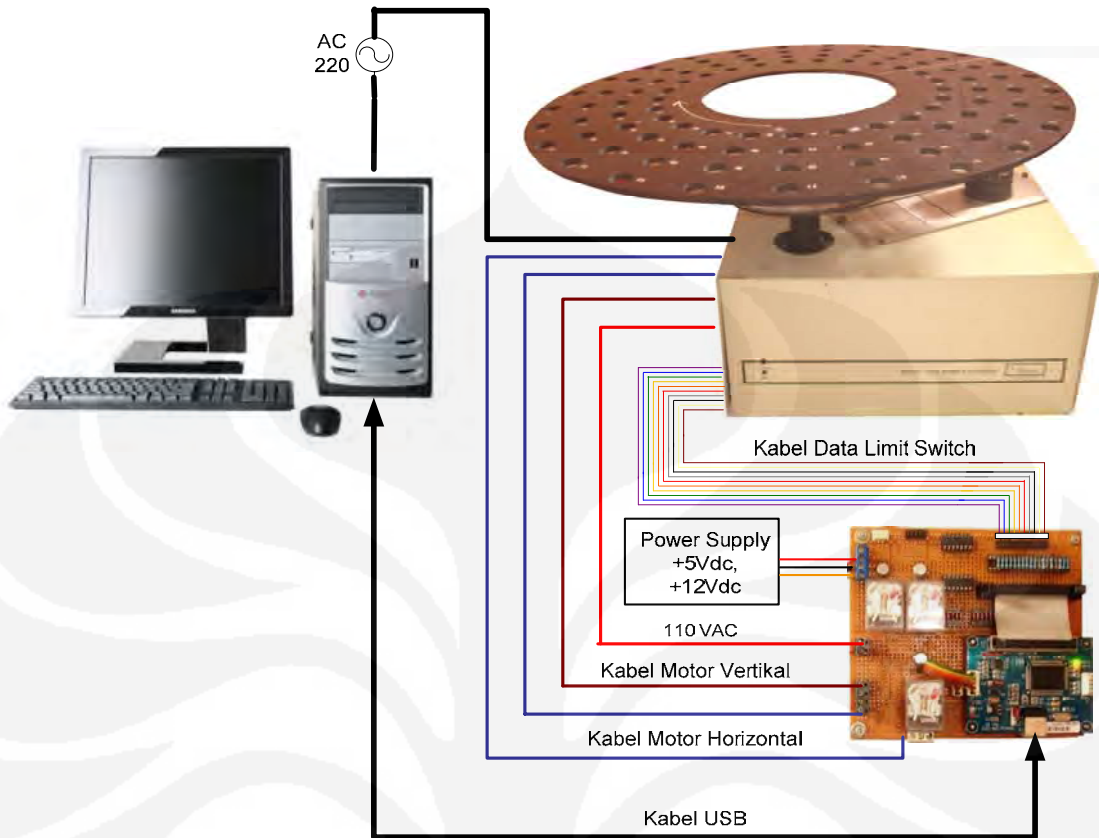
Relay pada rangkaian *driver* motor berfungsi untuk menggerakkan motor, dan untuk mengaktifkan Relay digunakan transistor yang difungsikan sebagai saklar. Saat kaki basis transistor menerima pulsa hingga dapat mengalirkan arus basis yang cukup untuk transistor, maka transistor akan berada pada kondisi jenuh (saturasi). Pada kondisi ini arus kolektor akan mengalir (sambungan C-E) terhubung dan menyebabkan coil pada Relay teraliri arus yang cukup besar dan menjadi magnet yang menarik saklar mekanik ke posisi sambung atau kontak. Sebaliknya, jika tidak ada arus yang mengalir pada Rb dan basis transistor, maka transistor dalam kondisi tidak bekerja (cutt off), kondisi ini identik dengan saklar yang terbuka (sambungan C-E terpisah) dan menyebabkan coil pada Relay hampir tidak dialiri arus dan saklar mekanik beralih ke posisi putus. Sambung dan putusnya saklar mekanik pada Relay ini menentukan gerak motor.

Resistor pada kaki basis (Rb) digunakan sebagai pembatas arus saat transistor pada kondisi saturasi (aktif). Dioda pada rangkaian Relay berfungsi untuk melindungi transistor dari tegangan balik (*kickback*) yang terlalu besar akibat perubahan arus pada coil

Input *driver* Relay dihubungkan dengan Port C pada mikrokontroler AN2131QC melalui modul *interface* USB Devasys I2C IO.

#### **4.2 PENGUJIAN SISTEM KESELURUHAN**

Pengujian sistem keseluruhan dilakukan setelah dilakukan pengujian pada modul *interface* USB Devasys I2C IO dan modul *driver* motor. Tujuan dari pengujian sistem keseluruhan adalah untuk mengetahui unjuk kerja dari sistem kontrol *sample changer* perangkat RIA Nucleus Model 1600 Gamma Counting, apakah sudah memenuhi tujuan yang diinginkan. Gambar rangkaian pengujian sistem keseluruhan ditunjukkan pada Gambar 4.5, selanjutnya melakukan beberapa proses pengujian dengan menggunakan program yang telah dibuat. Tampilan dari program sistem kontrol *sample changer* perangkat RIA Nucleus Model 1600 Gamma Counting ditunjukkan pada Gambar 4.6.



Gambar 4.5. Rangkaian pengujian sistem keseluruhan



Gambar 4.6. Tampilan program sistem kontrol *sample changer* perangkat RIA Nucleus Model 1600 Gamma Counting

Pengujian sistem secara keseluruhan dilakukan dengan beberapa proses pengujian. Proses pengujian yang dilakukan adalah pengujian gerak posisi kolimator *detector*, pengujian proses *counting sample* dengan perintah *counting current sample*, pengujian proses *counting sample* dengan perintah *counting 25 sample*, pengujian proses *counting sample* dengan perintah *counting All sample* (100 *sample*), pengujian proses *counting sample* pada nomor *sample tertentu*.

#### 4.2.1 Pengujian Gerak Posisi Kolimator *Detector*

Pengujian gerak posisi kolimator *detector* merupakan salah satu bagian dari pengujian system. Tujuan pengujian gerak posisi kolimator *detector* adalah untuk mengetahui unjuk kerja sistem pada pengontrolan pergerakan kolimator *detector*, apakah sudah sesuai dengan perintah yang diberikan melalui program. Terdapat empat pilihan perintah untuk menggerakkan posisi kolimator *detector*. Empat pilihan perintah tersebut adalah perintah menggerakkan posisi kolimator *detector* pada posisi *sample range* 1-25, perintah menggerakkan posisi kolimator *detector* pada posisi *sample range* 26-50, perintah menggerakkan posisi kolimator *detector* pada posisi *sample range* 51-75, perintah menggerakkan posisi kolimator *detector* pada posisi *sample range* 76-100. Pengujian gerak posisi kolimator *detector* dilakukan sebanyak sepuluh kali dan hasil pengujian seperti ditunjukkan pada Tabel 4.3.

**Tabel 4.3.** Hasil Pengujian gerak posisi kolimator *detector*

Pengujian Ke	Perintah Gerak		Hasil Pengujian		
	Dari Posisi <i>Sample Range</i>	Ke Posisi <i>Sample Range</i>	Arah Gerak Kolimator <i>Detector</i>	Berhenti Pada Posisi yang Dituju	Waktu Proses (detik)
1	1 - 25	26 - 50	Ke Kiri	✓	4.8
2	26 - 50	51 - 75	Ke Kiri	✓	5.1
3	51 - 75	76 - 100	Ke Kiri	✓	5.2
4	76 - 100	51 - 75	Ke Kanan	✓	4.7
5	51 - 75	26 - 50	Ke Kanan	✓	5.2
6	26 - 50	1 - 25	Ke Kanan	✓	5.4
7	1 - 25	51 - 75	Ke Kiri	✓	9.8
8	51 - 75	1 - 25	Ke Kanan	✓	9.7
9	1 - 25	76 - 100	Ke Kiri	✓	14.6
10	76 - 100	1 - 25	Ke Kanan	✓	14.7

Keterangan: ✓ = sesuai dengan perintah yang diberikan.

Hasil pengujian gerak posisi kolimator *detector* seperti ditunjukkan pada Tabel 4.3, menjelaskan bahwa *driver* motor dapat bekerja menggerakkan motor horizontal dan pergerakkan posisi kolimator *detector* dapat dikendalikan oleh sistem kontrol melalui PC dengan baik. Pada pengujian gerak posisi kolimator *detector*, waktu yang dibutuhkan untuk menggerakkan satu langkah gerak posisi kolimator *detector* adalah 5 detik.

#### 4.2.2 Pengujian Proses *Counting Sample* Dengan Perintah *Counting Current Sample*.

Pengujian proses *counting sample* dengan perintah *counting current sample* merupakan salah satu bagian dari pengujian sistem. Tujuan pengujian proses *counting sample* dengan perintah *counting current sample* adalah untuk mengetahui unjuk kerja sistem pada pengontrolan gerak motor vertical, apakah sudah sesuai dengan perintah yang diberikan melalui program. Pada pengujian ini *sample* yang akan di *counting* adalah *sample* yang posisinya saat itu tepat diatas *detector*. Sebelum dilakukan pengujian maka diperlukan pengesetan terhadap lamanya waktu proses *counting*, pada pengujian kali ini waktu yang diset adalah 3000 ms. Pengujian proses *counting sample* dengan perintah *counting current sample* dilakukan sebanyak sepuluh kali dan hasil pengujian seperti ditunjukkan pada Tabel 4.4.

**Tabel 4.4.** Hasil Pengujian proses *counting sample* dengan perintah *counting current sample*

Pengujian Ke	Sample Nomor	Hasil Pengujian		
		Proses Counting	Setelah <i>counting</i> nomor Sample yang ditampilkan	Waktu Proses (detik)
1	4	✓	5	15.3
2	5	✓	6	15.2
3	6	✓	7	15.2
4	7	✓	8	15.2
5	8	✓	9	15.2
6	9	✓	10	15.2
7	10	✓	11	15.2
8	11	✓	12	15.2
9	12	✓	13	15.2
10	13	✓	14	15.1

Keterangan: ✓ = sesuai dengan perintah yang diberikan.

Hasil pengujian proses *counting sample* dengan perintah *counting current sample* seperti ditunjukkan pada Tabel 4.4, menjelaskan bahwa Perintah *counting current sample* dapat dikendalikan oleh sistem kontrol melalui PC dengan baik., hasil pembacaan *limit switch* posisi nomor *sample* sesuai dengan posisi nomor *sample* yang sebenarnya dan setelah proses *counting* akan ditampilkan nomor *sample* berikutnya. Waktu proses *counting* Per-Sample adalah 15 detik.

#### 4.2.3 Pengujian Proses *Counting Sample* Dengan Perintah *Counting 25 Sample*.

Pengujian proses *counting sample* dengan perintah *counting 25 sample* merupakan salah satu bagian dari pengujian sistem. Tujuan pengujian proses *counting sample* dengan perintah *counting 25 sample* adalah untuk mengetahui unjuk kerja sistem pada pengontrolan gerak motor vertikal dan motor horizontal, apakah sudah sesuai dengan perintah yang diberikan melalui program. Pada pengujian ini *sample* yang akan di *counting* sebanyak 25 *sample* pada satu *sample range*. Sebelum dilakukan pengujian maka diperlukan pengesetan terhadap lamanya waktu proses *counting*, pada pengujian kali ini waktu yang diset adalah 3000 ms dan mengatur posisi kolimator *detector* pada *sample range* yang akan di lakukan proses *counting*. Pengujian proses *counting sample* dengan perintah *counting 25 sample* dilakukan sebanyak delapan kali dan hasil pengujian seperti ditunjukkan pada Tabel 4.5.

**Tabel 4.5.** Hasil Pengujian proses *counting sample* dengan perintah *counting 25 sample*.

Pengujian Ke	Counting Sample Nomor	Set posisi detector pada sample range	Hasil Pengujian		
			Proses Counting	Berhenti proses counting di Sample nomor	Waktu Proses ( menit )
1	1 – 25	1 – 25	✓	25	6'.17"
2	26 – 50	26 – 50	✓	50	6'.16"
3	51 – 75	51 – 75	✓	75	6'.17"
4	76 – 100	76 – 100	✓	100	6'.14"
5	1 – 25	1 – 25	✓	25	6'.17"
6	26 – 50	26 – 50	✓	50	6'.15"
7	51 – 75	51 – 75	✓	75	6'.17"
8	76 – 100	76 – 100	✓	100	6'.16"

Keterangan: ✓ = sesuai dengan perintah yang diberikan.

Hasil pengujian proses *counting sample* dengan perintah *counting 25 sample* seperti ditunjukkan pada Tabel 4.5, menjelaskan bahwa *driver* motor dapat bekerja menggerakkan motor horizontal dan motor vertikal, hasil pembacaan *limit switch* posisi nomor *sample* sesuai dengan posisi nomor *sample* yang sebenarnya dan setelah proses *counting* akan ditampilkan nomor *sample* berikutnya. Perintah *counting 25 sample* dapat dikendalikan oleh sistem kontrol melalui PC dengan baik.

#### 4.2.4 Pengujian Proses *Counting Sample* Dengan Perintah *Counting All Sample* (100 *sample*).

Pengujian proses *counting sample* dengan perintah *counting All sample* (100 *sample*) merupakan salah satu bagian dari pengujian sistem. Tujuan pengujian proses *counting sample* dengan perintah *counting all sample* adalah untuk mengetahui unjuk kerja sistem pada pengontrolan gerak motor vertikal, dan gerak motor horizontal, apakah sudah sesuai dengan perintah yang diberikan melalui program. Pada pengujian ini *sample* yang akan di *counting* sebanyak 100 *sample* sesuai dengan jumlah *sample* pada *sample tray*. Sebelum dilakukan pengujian maka diperlukan pengesetan terhadap lamanya waktu proses *counting*, pada pengujian kali ini waktu yang diset adalah 3000 ms. Pengujian proses *counting sample* dengan perintah *counting All sample* (100 *sample*) dilakukan sebanyak lima kali dan hasil pengujian seperti ditunjukkan pada Tabel 4.6.

**Tabel 4.6.** Hasil Pengujian proses *counting sample* dengan perintah *counting All sample*.

Pengujian Ke	Hasil Pengujian	
	Proses <i>Counting All Sample</i> (100 <i>Sample</i> )	Waktu Proses ( menit )
1	✓	25.19
2	✓	25.19
3	✓	25.18
4	✓	25.20
5	✓	25.19
6	✓	25.19
7	✓	25.17
8	✓	25.19

Keterangan: ✓ = sesuai dengan perintah yang diberikan.

Hasil pengujian proses *counting sample* dengan perintah *counting 100 sample* seperti ditunjukkan pada Tabel 4.6, menjelaskan bahwa *driver* motor dapat bekerja menggerakkan motor horizontal dan motor vertikal, hasil pembacaan *limit switch* posisi nomor *sample* sesuai dengan posisi nomor *sample* yang sebenarnya dan setelah proses *counting* akan ditampilkan nomor *sample* berikutnya. Setelah melakukan proses *counting 25 sample* pertama (1-25), *sample tray* akan berhenti sesaat di posisi atas dan posisi *detector* akan berpindah posisi pada *sample range* berikutnya (26–50), kemudian proses *counting 25 sample* dilakukan lagi sampai pada *sample* ke 100. Setelah melakukan proses *counting 100 sample*, maka posisi *detector* kembali ke posisi *sample range* 1-25. Perintah *counting 100 sample* dapat dikendalikan oleh sistem kontrol melalui PC dengan baik.

#### **4.2.5 Pengujian Proses *Counting Sample* Pada Nomor *Sample* Tertentu**

Pengujian proses *counting sample* pada nomor *sample* tertentu merupakan salah satu bagian dari pengujian sistem. Tujuan pengujian proses *counting sample* pada nomor *sample tertentu* adalah untuk mengetahui unjuk kerja sistem pada pengontrolan gerak motor vertikal, dan gerak motor horizontal, apakah sudah sesuai dengan perintah yang diberikan melalui program. Pada pengujian ini jumlah *sample* yang akan di *counting* ditentukan oleh operator. Penentuan nomor *sample* yang akan dilakukan proses *counting* adalah pada *sample range* yang sama dan berurutan jika jumlahnya lebih dari satu. Sebelum dilakukan pengujian maka diperlukan pengesetan terhadap lamanya waktu proses *counting*, pada pengujian kali ini waktu yang diset adalah 3000 ms. Pengujian proses *counting sample* pada nomor *sample* tertentu dilakukan sebanyak 10 kali dan hasil pengujian seperti ditunjukkan pada Tabel 4.7.

**Tabel 4.7.** Hasil Pengujian proses *counting sample* pada nomor *sample* tertentu.

Pengujian Ke	Proses <i>Counting Sample</i>		Hasil Pengujian
	Dari Nomor <i>Sample</i>	Ke Nomor <i>Sample</i>	
1	4	8	✓
2	12	16	✓
3	19	25	✓
4	27	28	✓
5	31	33	✓
6	34	38	✓
7	42	45	✓
8	49	50	✓
9	72	75	✓
10	77	85	✓

Keterangan: ✓ = sesuai dengan perintah yang diberikan.

Hasil pengujian proses *counting sample* pada nomor *sample* tertentu seperti ditunjukkan pada Tabel 4.7, menjelaskan bahwa *driver* motor dapat bekerja menggerakkan motor horizontal dan motor vertikal, hasil pembacaan *limit switch* posisi nomor *sample* sesuai dengan posisi nomor *sample* yang sebenarnya dan setelah proses *counting* akan ditampilkan nomor *sample* berikutnya. Proses *counting sample* pada nomor *sample* tertentu dapat dikendalikan oleh sistem kontrol melalui PC dengan baik.

### 4.3 ANALISA SISTEM

1. Modul *interface* USB Devasys I2C I/O yang digunakan pada sistem kontrol pergerakan *sample changer* pada perangkat RIA Nucleus Model 1600 Gamma Counting dapat bekerja dan dapat dikenali oleh PC dengan baik.
2. Modul *driver* motor hasil perancangan dapat bekerja menggerakkan motor horizontal dan motor vertikal dan pergerakan motor dapat dikontrol oleh PC dengan baik melalui modul *interface* USB Devasys I2C I/O.
3. Modul *driver* motor dapat menggerakkan motor saat kaki basis transistor menerima pulsa hingga dapat mengalirkan arus basis yang cukup untuk transistor, maka transistor akan berada pada kondisi jenuh (saturasi). Pada



kondisi ini arus kolektor akan mengalir (sambungan C-E) terhubung dan menyebabkan coil pada Relay teraliri arus yang cukup besar dan menjadi magnet yang menarik saklar mekanik ke posisi sambung atau kontak. Sebaliknya, jika tidak ada arus yang mengalir pada Rb dan basis transistor, maka transistor dalam kondisi tidak bekerja (cutt off), kondisi ini identik dengan saklar yang terbuka (sambungan C-E terpisah) dan menyebabkan coil pada Relay hampir tidak dialiri arus dan saklar mekanik beralih ke posisi putus. Sambung dan putusnya saklar mekanik pada Relay ini yang menentukan gerak motor.

4. Pada saat posisi *sample tray* menyentuh *limit sitch* posisi nomor *sample* akan dilakukan pembacaan *limit sitch* posisi nomor *sample* dan setelah melakukan proses *counting*, maka hasil pembacaan *limit switch* posisi nomor *sample* + 1 akan ditampilkan sebagai informasi dari nomor *sample* berikutnya.
5. Penentuan nomor *sample pada proses counting sample* tertentu tidak dapat dilakukan pada posisi *sample range* yang berbeda, jumlah *sample* yang akan di-*counting* tidak lebih dari dua puluh lima *sample*, posisinya pun harus berurutan. Jika aturan ini tidak dipenuhi maka akan terjadi kegagalan sistem.
6. Penentuan lamanya waktu untuk proses *counting* ditentukan oleh operator. Pada kegiatan rancang bangun sistem kontrol pergerakan *sample changer* pada perangkat RIA Nucleus Model 1600 Gamma Counting tidak ditentukan aturan baku terhadap lamanya waktu untuk proses *counting*.
7. Berdasarkan pengujian pada perintah menggerakkan posisi kolimator *detector*, rata – rata waktu yang dibutuhkan untuk menggerakkan satu langkah gerak posisi kolimator *detector* adalah 5 detik.
8. Berdasarkan pengujian, total waktu yang diperlukan untuk melakukan proses *counting current sample* dengan setting waktu per-*counting* 3000 ms adalah 15 detik.
9. Berdasarkan pengujian, total waktu yang diperlukan untuk melakukan proses *counting 25 sample* dengan setting waktu per-*counting* 3000 ms adalah 6 menit 17 detik.

10. Berdasarkan pengujian, total waktu yang diperlukan untuk melakukan proses *counting All sample* (100 *Sample*) dengan setting waktu *per-counting* 3000 ms adalah 25 menit 19 detik.
11. Berdasarkan hasil pengujian sistem keseluruhan, menunjukkan bahwa sistem kontrol pergerakan *sample changer* pada perangkat RIA Nucleus Model 1600 Gamma Counting hasil rancang bangun dapat bekerja dengan baik dan sesuai dengan yang diharapkan.



## **BAB 5 KESIMPULAN**

Dari hasil kegiatan rancang bangun dapat diambil kesimpulan sebagai berikut:

1. Telah berhasil dilakukan rancang bangun sebuah sistem kontrol *sample changer* berbasis PC pada perangkat RIA Nuclues Model 1600 Gamma Counting dengan menggunakan modul antar muka USB Devasys I2C I/O dan mikrokontroler AN2131QC yang bekerja sesuai dengan tujuan yang diinginkan.
2. Sistem kontrol *sample changer* perangkat RIA Nuclues Model 1600 Gamma Counting yang dirancang dan dibuat secara prinsip mempunyai nilai lebih dari sistem sebelumnya, baik dari nilai fungsi dan nilai waktu juga dari peningkatan kemampuan dan unjuk kerja sistem.
3. Dilihat dari nilai fungsi, sistem kontrol *sample changer* perangkat RIA Nuclues Model 1600 Gamma Counting hasil rancang bangun lebih fleksibel dan lebih *user friendly*, karena *operator* dapat melakukan proses *counting sample* sesuai dengan menu pilihan pada perangkat lunak.
4. Dilihat dari nilai waktu, sistem kontrol *sample changer* perangkat RIA Nuclues Model 1600 Gamma Counting hasil rancang bangun lebih hemat waktu dalam pengoperasiannya, karena *operator* dapat melakukan proses *counting sample* sesuai dengan jumlah *sample* tanpa harus melakukan proses *counting 100 sample*.
5. Dilihat dari aplikasi teknologi, sistem kontrol *sample changer* perangkat RIA Nuclues Model 1600 Gamma Counting hasil rancang bangun mengaplikasikan USB sebagai media komunikasi antara perangkat *sample changer* dengan PC sebagai basis sistem kontrol.

## DAFTAR ACUAN

- [1] Nazir, Fadil. (2009, Januari 15). Personal Interview. Pusat Acuan dan Kepakaran Kesehatan Masyarakat Berbasis Teknologi Nuklir. Jakarta.
- [2] Wikipedia (2009). *Universal Serial Bus*. Diakses 01 Mei 2009, dari Wikipedia bahasa Indonesia, ensiklopedia bebas.  
[http://id.wikipedia.org/wiki/Universal\\_Serial\\_Bus](http://id.wikipedia.org/wiki/Universal_Serial_Bus)
- [3] Wikipedia (2009). *Universal Serial Bus*. Diakses 01 Mei 2009, dari Wikipedia bahasa Indonesia, ensiklopedia bebas.  
[http://id.wikipedia.org/wiki/Universal\\_Serial\\_Bus](http://id.wikipedia.org/wiki/Universal_Serial_Bus)
- [4] Devasys (2009). *USB I2C/IO P.C.B. (Rev. B)*. Diakses 14 April 2009, dari devasys.com  
<http://www.devasys.com/usbi2ciob.htm>
- [5] Cypress Semiconductor Corporation (2000). *EZ-USB Technical Reference Manual Version 1.9*, hal. 1-3. Diakses 28 April 2009, dari Cypress  
<http://www.cypress.com/?rID=14667>
- [6] Cypress Semiconductor Corporation (2000). *EZ-USB Technical Reference Manual Version 1.9*, hal. 1-18. Diakses 28 April 2009, dari Cypress  
<http://www.cypress.com/?rID=14667>
- [7] Philips Semiconductors (1997). *PCF8574 Remote 8-bit I/O expander for I2C-bus Datasheet*. hal. 4. Diakses Diakses 20 April 2009, dari nxp.com.  
[www.nxp.com/acrobat\\_download/datasheets/PCF8574\\_4.pdf](http://www.nxp.com/acrobat_download/datasheets/PCF8574_4.pdf)
- [8] Philips Semiconductors (1997). *PCF8574 Remote 8-bit I/O expander for I2C-bus Datasheet*. hal. 5. Diakses Diakses 20 April 2009, dari nxp.com.  
[www.nxp.com/acrobat\\_download/datasheets/PCF8574\\_4.pdf](http://www.nxp.com/acrobat_download/datasheets/PCF8574_4.pdf)
- [9] Philips Semiconductors (1997). *PCF8574 Remote 8-bit I/O expander for I2C-bus Datasheet*. hal. 8. Diakses Diakses 20 April 2009, dari nxp.com.  
[www.nxp.com/acrobat\\_download/datasheets/PCF8574\\_4.pdf](http://www.nxp.com/acrobat_download/datasheets/PCF8574_4.pdf)
- [10] Philips Semiconductors (1997). *PCF8574 Remote 8-bit I/O expander for I2C-bus Datasheet*. hal. 9. Diakses Diakses 20 April 2009, dari nxp.com.  
[www.nxp.com/acrobat\\_download/datasheets/PCF8574\\_4.pdf](http://www.nxp.com/acrobat_download/datasheets/PCF8574_4.pdf)
- [11] Philips Semiconductors (1997). *PCF8574 Remote 8-bit I/O expander for I2C-bus Datasheet*. hal. 10. Diakses Diakses 20 April 2009, dari nxp.com.  
[www.nxp.com/acrobat\\_download/datasheets/PCF8574\\_4.pdf](http://www.nxp.com/acrobat_download/datasheets/PCF8574_4.pdf)
- [12] Dunia Listrik (2008). *Motor Listrik* . Diakses Diakses 31 Mei 2009, dari HaGe.  
<http://dunia-listrik.blogspot.com/2008/12/motor-listrik.html>

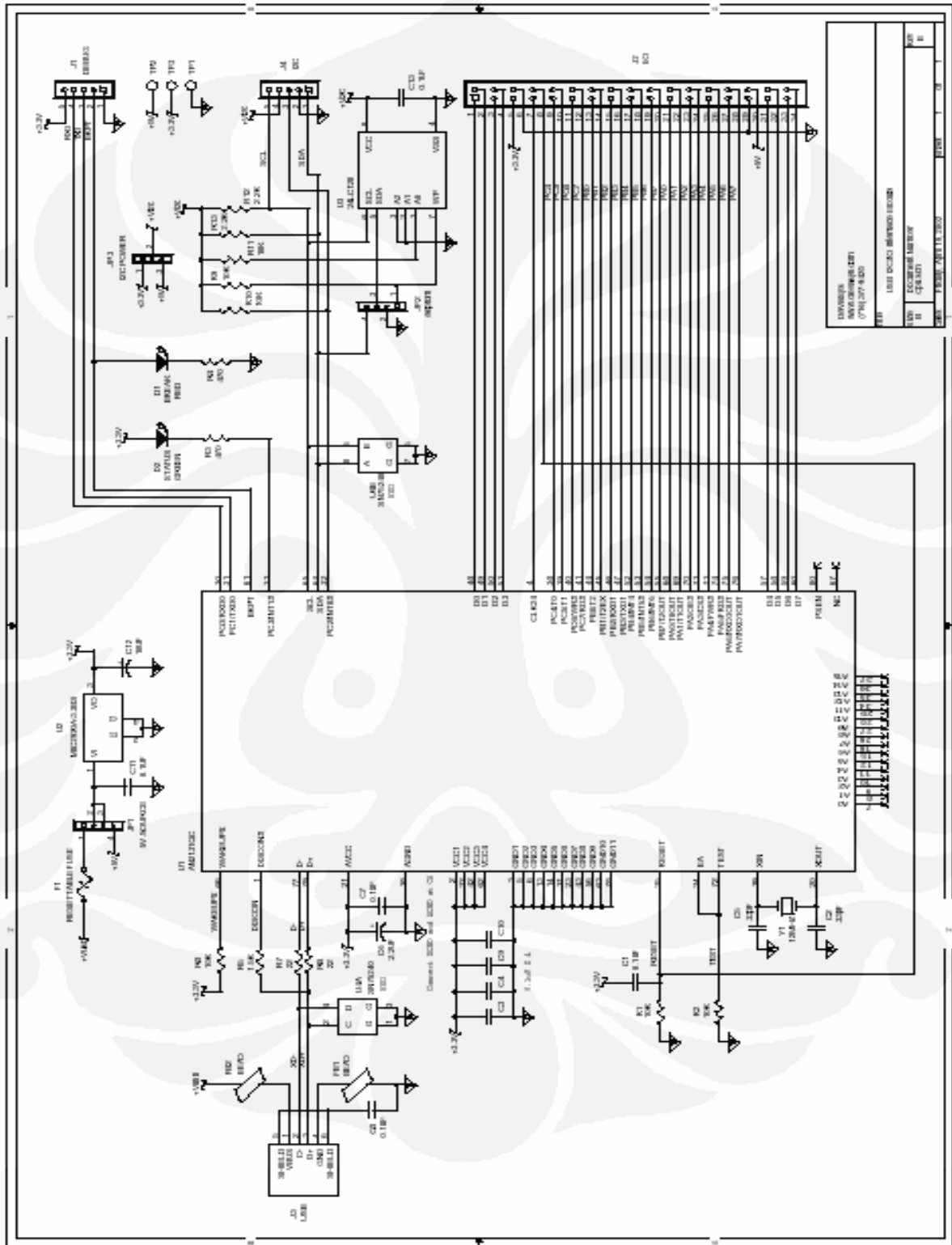
- [13] Hurst Manufacturing (2009). *Series T Geared Synchronous Motor*. Diakses Diakses 31 Mei 2009, dari Emerson Motor Technologies.  
<http://www.myhurst.com/>
- [14] Dunia Listrik (2008). *Motor Listrik* . Diakses Diakses 31 Mei 2009, dari HaGe.  
<http://dunia-listrik.blogspot.com/2008/12/motor-listrik.html>
- [15] Bishop, Owen (2004). *Dasar Dasar Elektronika*. Erlangga. Jakarta. hal 53
- [16] Advance Micro Control Inc (2009). *Mechanical Limit Switch* . Diakses Diakses 20 April 2009, dari google.  
<http://www.amci.com/tutorials/tutorials-what-is-programmable-limit-switch.asp>
- [17] Omron Electronics Pte Ltd (2009). *MY General-purpose Relay*. Diakses Diakses 2 Juni 2009, dari Omron Industrial Automation Singapore.  
[http://www.omron-ap.com/product\\_info/MY/index.asp](http://www.omron-ap.com/product_info/MY/index.asp)

## DAFTAR REFERENSI

- Bishop, Owen (2004). Dasar Dasar Elektronika. Erlangga. Jakarta.
- Cypress Semiconductor Corporation. (2000). EZ-USB Technical Reference Manual Version 1.9. Diakses 28 April 2009.  
<http://www.cypress.com/?rID=14667>
- Mavino, Paul Albert (1989). Prinsip Prinsip Dasar Elektronika. Erlangga. Jakarta
- Nasution H, Sofyan (1987). Analisis Dan Desain Rangkaian Terpadu Digital. Erlangga. Jakarta.
- Nazir, Fadil. (2009, Januari 15). Personal Interview. Pusat Acuan dan Kepakaran Kesehatan Masyarakat Berbasis Teknologi Nuklir. Jakarta.
- Philips Semiconductors (1997). PCF8574 Remote 8-bit I/O expander for I2C-bus Datasheet. . Diakses Diakses 20 April 2009, dari nxp.com.  
[www.nxp.com/acrobat\\_download/datasheets/PCF8574\\_4.pdf](http://www.nxp.com/acrobat_download/datasheets/PCF8574_4.pdf)
- Zuhal. (1982). Dasar Tenaga Listrik. Institut Teknologi Bandung. Bandung.

# LAMPIRAN

Lampiran 1.



Gambar 2.5 Rangkaian Modul Devasy's I2C I/O



## Lampiran 2.

Tabel 2.2 Fungsi Konfigurasi 80 PIN AN2131QC

PIN	Nama	Tipe	Keterangan
21	AVCC	Power	Analog Vcc
18	AGND	Power	Analog Ground
1	DISCON#	Output	Disconnect
77	USBD	I/O/Z	USB D- Signal
79	USBD	I/O/Z	USB D+ Signal
7-12, 15, 16, 26-29, 34-37	A0 – A5, A6, A7, A8 – A11, A12 – A15	Output	8051 Address bus
48-51, 57-60	D0 – D3, D4 – D7	I/O/Z	8051 Data bus
80	PSEN#	Output	<b>Program Store Enable</b> Merupakan pin yang membaca <i>strobe</i> pada memori eksternal.
61	BKPT	Output	<i>Breakpoint</i>
25	RESET	Input	<b>Active High Reset</b> Merupakan pin yang digunakan untuk me-reset mikrokontroler 8051 dan USB SIE
24	EA	Input	<i>External Access</i>
19	XIN	Input	<i>Crystal Input</i>
20	XOUT	Output	<i>Crystal Output</i>
68	PA0 / T0Out	I/O	<b>PA0</b> Merupakan pin I/O dua arah. Fungsi lainnya <b>T0Out</b> merupakan Output <i>timer</i> eksternal 0
69	PA1 / T1Out	I/O	<b>PA1</b> Merupakan pin I/O dua arah. Fungsi lainnya <b>T1Out</b> merupakan Output <i>timer</i> eksternal 1
70	PA2 / OE#	I/O	<b>PA2</b> Merupakan pin I/O dua arah. Fungsi lainnya <b>T1Out</b> merupakan Output <i>timer</i> eksternal 1
71	PA3 / CS#	I/O	<b>PA3</b> Merupakan pin I/O dua arah. Fungsi lainnya <b>CS</b> merupakan fungsi <i>Chip Select</i>

<b>PIN</b>	<b>Nama</b>	<b>Tipe</b>	<b>Keterangan</b>
73	PA4 / FWR#	I/O	<b>PA4</b> Merupakan pin I/O dua arah. Fungsi lainnya <b>FWR</b> merupakan pin <i>Fast Write</i>
74	PA5 / FRD#	I/O	<b>PA5</b> Merupakan pin I/O dua arah. Fungsi lainnya <b>FRD</b> merupakan pin <i>Fast Read</i>
75	PA6 / RXD0OUT	I/O	<b>PA6</b> Merupakan pin I/O dua arah. Fungsi lainnya <b>RXD0OUT</b> merupakan input port serial 0
76	PA7 / RXD1OUT	I/O	<b>PA7</b> Merupakan pin I/O dua arah. Fungsi lainnya <b>RXD1OUT</b> merupakan input port serial 1
44	PB0 / T2	I/O	<b>PB0</b> Merupakan pin I/O dua arah. Fungsi lainnya <b>T2</b> (Input <i>timer</i> eksternal 2)
45	PB1 / T2EX	I/O	<b>PB1</b> Merupakan pin I/O dua arah. Fungsi lainnya <b>T2</b> (Input <i>timer</i> eksternal 2)
46	PB2 / RXD1	I/O	<b>PB2</b> Merupakan pin I/O dua arah. Fungsi lainnya <b>RXD1</b> merupakan input port serial 1
47	PB3 / TXD1	I/O	<b>PB3</b> Merupakan pin I/O dua arah. Fungsi lainnya <b>TXD1</b> merupakan output port serial 1
52	PB4 / INT4	I/O	<b>PB3</b> Merupakan pin I/O dua arah. Fungsi lainnya <b>INT4</b> (Interupsi eksternal 4)
53	PB5 / INT5	I/O	<b>PB3</b> Merupakan pin I/O dua arah. Fungsi lainnya <b>INT5</b> (Interupsi eksternal 5)
54	PB6 / INT6	I/O	<b>PB3</b> Merupakan pin I/O dua arah. Fungsi lainnya <b>INT6</b> (Interupsi eksternal 6)
55	PB7 / T2OUT	I/O	<b>PB3</b> Merupakan pin I/O dua arah. Fungsi lainnya <b>T2OUT</b> (Input <i>timer</i> eksternal 2)
30	PC0 / RXD0	I/O	<b>PC0</b> Merupakan pin I/O dua arah. Fungsi lainnya <b>RXD0</b> merupakan input port serial 0
31	PC1 / TXD0	I/O	<b>PC1</b> Merupakan pin I/O dua arah. Fungsi lainnya <b>TXD0</b> merupakan output port serial 0

<b>PIN</b>	<b>Nama</b>	<b>Tipe</b>	<b>Keterangan</b>
32	PC2 / INT0	I/O	<b>PC2</b> Merupakan pin I/O dua arah. Fungsi lainnya <b>INT0</b> (Interupsi eksternal 0)
33	PC3 / INT1	I/O	<b>PC3</b> Merupakan pin I/O dua arah. Fungsi lainnya <b>INT1</b> (Interupsi eksternal 1)
38	PC4 / T0	I/O	<b>PC4</b> Merupakan pin I/O dua arah. Fungsi lainnya <b>T0</b> (Input <i>timer</i> eksternal 0)
39	PC5 / T1	I/O	<b>PC5</b> Merupakan pin I/O dua arah. Fungsi lainnya <b>T1</b> (Input <i>timer</i> eksternal 1)
40	PC6 / WR	I/O	<b>PC6</b> Merupakan pin I/O dua arah. Fungsi lainnya <b>WR</b> (Penulisan <i>strobe</i> pada memori data )
41	PC7 / RD	I/O	<b>PC7</b> Merupakan pin I/O dua arah. Fungsi lainnya <b>RD</b> (Pembacaan <i>strobe</i> pada memori data )
4	CLK24	Output	
66	WAKEUP#	Input	
65	SCL	OD	I <sup>2</sup> C Clock
64	SDA	OD	I <sup>2</sup> C Data
2, 22, 42, 62	Vcc	Power	Merupakan pin yang berfungsi sebagai pin masukan catu daya.
3, 5, 6, 13, 14, 17, 23, 43, 56, 63, 72, 78	GND	Power	Merupakan pin <i>ground</i> .
67	CPU12MHZ		

## Lampiran 3.

**Listing Program Utama**

```
Private kiri%, putar%, kanan%, PosisiDetektor%, TargetNumber%, StartNumber%, EndNumber%,
DelayValue&
```

```
Private Sub Form_Load()
BacaSwitch
ResetCmd_Click
BacaSwitch
NoSampleTxt.Text = Val(&H1)
End Sub
```

```
Private Sub BacaSwitch()
```

```
Select Case (&HF And ReadI2cIo(4))
```

```
Case 0:
```

```
Option2.Value = 0
Option3.Value = 0
Option4.Value = 0
Option5.Value = 0
Option2.Value = True
```

```
Case 1:
```

```
Option2.Value = 1
Option3.Value = 0
Option4.Value = 0
Option5.Value = 0
PosisiDetektor = 0
```

```
Case 2:
```

```
Option2.Value = 0
Option3.Value = 1
Option4.Value = 0
Option5.Value = 0
PosisiDetektor = 1
```

```
Case 4:
```

```
Option2.Value = 0
Option3.Value = 0
Option4.Value = 1
Option5.Value = 0
PosisiDetektor = 2
```

```
Case 8:
```

```
Option2.Value = 0
Option3.Value = 0
Option4.Value = 0
Option5.Value = 1
PosisiDetektor = 3
```

```
End Select
```

```
NomorPosisiSample = Val(ReadPort And &H1F) + 25 * PosisiDetektor
NoSampleTxt = NomorPosisiSample
NoSampleTxt.Refresh
```

```
End Sub
```

```

Private Sub BacaSwitchDown()
I2C4Txt.Text = Hex$(&HF And ReadI2cIo(4))

End Sub

Private Sub GerakKekanan()
Call DAPI_WriteIoPorts(hDevInstance, &H20000, &HF0000)

End Sub

Private Sub GerakKekiri()
Call DAPI_WriteIoPorts(hDevInstance, &H60000, &HF0000)

End Sub

Private Sub SetPosisi_Atas()
Call DAPI_WriteIoPorts(hDevInstance, &H10000, &H10000)
Call DAPI_WriteIoPorts(hDevInstance, &H1000, &H1000)
While (ReadPort And &H20) / &H20 <> 1
Wend
Call DAPI_WriteIoPorts(hDevInstance, &H0, &HFF000)

End Sub

Private Sub RightCmd_Click()
kanan = 1: kiri = 0

Call DAPI_WriteIoPorts(hDevInstance, &H20000, &HF0000)
Tdelay 1000
RightCmd.BackColor = &HFFC0C0

End Sub

Private Sub LeftCmd_Click()
kiri = 1: kanan = 0
GerakKekiri
Tdelay 1000
LeftCmd.BackColor = &HFFC0C0

End Sub

Sub SetKePosisiA()
SetPosisi_Atas 'rem posisikan keatas
If PosisiDetektor = 0 Then Exit Sub
RightCmd.BackColor = &HFF00&
Call DAPI_WriteIoPorts(hDevInstance, &H100, &HF00)
DoEvents
RightCmd_Click
While (&HF And ReadI2cIo(4)) <> 1
Tdelay 300
DoEvents
Wend

PosisiDetektor = 0
Call DAPI_WriteIoPorts(hDevInstance, &H0, &H20000) 'henti horizontal
End Sub

```

```

Sub SetKePosisiB()
If PosisiDetektor = 1 Then Exit Sub

SetPosisi_Atas          'rem posisikan keatas
While (ReadPort And &H20) / &H20 <> 1
Wend
Call DAPI_WriteIoPorts(hDevInstance, &H200, &HF00)

If PosisiDetektor > 1 Then
    RightCmd.BackColor = &HFF00&
    DoEvents
    RightCmd_Click
    While (&HF And ReadI2cIo(4)) <> 2
        Tdelay 300
        DoEvents
    Wend
End If

If PosisiDetektor < 1 Then
    LeftCmd.BackColor = &HFF00&

    DoEvents
    LeftCmd_Click
    While (&HF And ReadI2cIo(4)) <> 2
        Tdelay 300
        DoEvents
    Wend
End If

PosisiDetektor = 1
Call DAPI_WriteIoPorts(hDevInstance, &H0, &H20000) 'henti horisontal

End Sub

Sub SetKePosisiC()
If PosisiDetektor = 2 Then Exit Sub

SetPosisi_Atas          'rem posisikan keatas
While (ReadPort And &H20) / &H20 <> 1
Wend
Call DAPI_WriteIoPorts(hDevInstance, &H400, &HF00)

If PosisiDetektor > 2 Then
    RightCmd.BackColor = &HFF00&
    DoEvents
    RightCmd_Click
    While (&HF And ReadI2cIo(4)) <> 4
        Tdelay 300
        DoEvents
    Wend
End If

If PosisiDetektor < 2 Then
    LeftCmd.BackColor = &HFF00&
    DoEvents
    LeftCmd_Click
    While (&HF And ReadI2cIo(4)) <> 4
        Tdelay 300
        DoEvents
    Wend
End If

```

```

PosisiDetektor = 2
Call DAPI_WriteIoPorts(hDevInstance, &H0, &H20000) 'henti horizontal

End Sub

Sub SetKePosisiD()
If PosisiDetektor = 3 Then Exit Sub

SetPosisi_Atas      'rem posisikan keatas
While (ReadPort And &H20) / &H20 <> 1
Wend

Call DAPI_WriteIoPorts(hDevInstance, &H800, &HF00)

If PosisiDetektor > 3 Then
    RightCmd.BackColor = &HFF00&
    DoEvents
    RightCmd_Click
    While (&HF And ReadI2cIo(4)) <> 8
        Tdelay 300
        DoEvents
    Wend
End If

If PosisiDetektor < 3 Then
    LeftCmd.BackColor = &HFF00&
    DoEvents
    LeftCmd_Click
    While (&HF And ReadI2cIo(4)) <> 8
        Tdelay 300
        DoEvents
    Wend
End If

PosisiDetektor = 3
Call DAPI_WriteIoPorts(hDevInstance, &H0, &H20000) 'henti horizontal

End Sub

Private Sub Option2_Click()
    SetKePosisiA
End Sub

Private Sub Option3_Click()
    SetKePosisiB
End Sub

Private Sub Option4_Click()
    SetKePosisiC
End Sub

Private Sub Option5_Click()
    SetKePosisiD
End Sub

```

```

Sub CountCurrentSample()
Dim TargetNumber As Integer, naik As Integer
Call DAPI_WriteIoPorts(hDevInstance, &H1000, &H1000)
Call DAPI_WriteIoPorts(hDevInstance, &H0, &H1000)
DelayValue = Val(Text3.Text)

While Val(ReadPort And &H1F) = &H1F
Wend

Call DAPI_WriteIoPorts(hDevInstance, &H1000, &H1000) 'perintah gerak keatas
Tdelay 500 ' tunggu sampai switch benar-benar tertekan

Call DAPI_WriteIoPorts(hDevInstance, &H0, &H1000) 'perintah STOP
BacaSwitch
Option1.Value = True
BarIndikator DelayValue 'Tdelay DelayValue 'counting time

Option1.Value = False
ProgressBar1.Value = 0

SetPosisi_Atas
NoSampleTxt.Text = Val(NoSampleTxt.Text) + 1

If Val(NoSampleTxt.Text) = 26 Then
    NoSampleTxt.Text = 25
End If

If Val(NoSampleTxt.Text) = 51 Then
    NoSampleTxt.Text = 50
End If

If Val(NoSampleTxt.Text) = 76 Then
    NoSampleTxt.Text = 75
End If

If Val(NoSampleTxt.Text) = 101 Then
    NoSampleTxt.Text = 100
End If

DoEvents

End Sub

Private Sub Gerak25Sample()

Dim EndNumber As Integer, naik As Integer

CountCurrentSample
EndNumber = 25 + 25 * PosisiDetektor
NomorPosisiSample = Val(ReadPort And &H1F) + 25 * PosisiDetektor
NoSampleTxt = NomorPosisiSample

If NomorPosisiSample = EndNumber Then GoTo Done
Call DAPI_WriteIoPorts(hDevInstance, &H1000, &H1000)
Call DAPI_WriteIoPorts(hDevInstance, &H0, &H1000)
DelayValue = Val(Text3.Text)

While NomorPosisiSample <> EndNumber
    If Val(ReadPort And &H1F) = &H1F And naik = 1 Then
        Call DAPI_WriteIoPorts(hDevInstance, &H0, &H1000) 'perintah gerak kebawah
        naik = 0
    
```



```

End If

If Val(ReadPort And &H1F) <> &H1F And naik = 0 Then
    Call DAPI_WriteIoPorts(hDevInstance, &H1000, &H1000) 'perintah gerak keatas
    naik = 1
    Tdelay 500 ' tunggu sampai switch benar-benar tertekan

    Call DAPI_WriteIoPorts(hDevInstance, &H0, &H1000) 'perintah STOP
    BacaSwitch
    Option1.Value = True
    BarIndikator DelayValue

    Option1.Value = False
    ProgressBar1.Value = 0

    Call DAPI_WriteIoPorts(hDevInstance, &H1000, &H1000) 'perintah gerak keatas
    naik = 1
    Tdelay 500
    NoSampleTxt.Text = Val(NoSampleTxt.Text) + 1
    Tdelay 5000
    DoEvents
End If
Wend

Call DAPI_WriteIoPorts(hDevInstance, &H0, &H10000) ' perintah hentikan putaran motor
NoSampleTxt.Text = EndNumber

Done:

SetPosisi_Atas
NoSampleTxt.Text = Val(NoSampleTxt.Text) + 1

If Val(NoSampleTxt.Text) = 26 Then
    NoSampleTxt.Text = 25
End If

If Val(NoSampleTxt.Text) = 51 Then
    NoSampleTxt.Text = 50
End If

If Val(NoSampleTxt.Text) = 76 Then
    NoSampleTxt.Text = 75
End If

If Val(NoSampleTxt.Text) = 101 Then
    NoSampleTxt.Text = 100
End If

End Sub

Sub CountingAll()

Gerak25Sample
NoSampleTxt.Text = 26
SetKePosisiB
Gerak25Sample
NoSampleTxt.Text = 51
SetKePosisiC
Gerak25Sample
NoSampleTxt.Text = 76
SetKePosisiD

```

```

Gerak25Sample
NoSampleTxt.Text = 100
Tdelay 0
SetKePosisiA

NoSampleTxt.Text = Val(&H1)

End Sub

Sub PutarKeTargetNo(TargetNumber As Integer)
Call DAPI_WriteIoPorts(hDevInstance, &H1000, &H1000) ' gerak keatas
naik = 1
Call DAPI_WriteIoPorts(hDevInstance, &H10000, &H10000)
DelayValue = Val(Text3.Text)

While Val(ReadPort And &H1F) <> TargetNumber
  If Val(ReadPort And &H1F) = &H1F And naik = 1 Then
    Call DAPI_WriteIoPorts(hDevInstance, &H0, &H1000) ' perintah gerak kebawah
    naik = 0
  End If

  If Val(ReadPort And &H1F) <> &H1F And naik = 0 Then
    Call DAPI_WriteIoPorts(hDevInstance, &H1000, &H1000) 'perintah gerak keatas
    naik = 1
    Tdelay 500 ' tunggu sampai switch benar-benar tertekan
    Call DAPI_WriteIoPorts(hDevInstance, &H0, &H1000) 'perintah STOP
    BacaSwitch
    Option1.Value = True
    BarIndikator DelayValue
    Option1.Value = False
    ProgressBar1.Value = 0
    NoSampleTxt.Text = Val(NoSampleTxt.Text) + 1
    Call DAPI_WriteIoPorts(hDevInstance, &H1000, &H1000) 'perintah gerak keatas
    naik = 1
    DoEvents
  End If

Wend

Call DAPI_WriteIoPorts(hDevInstance, &H0, &H10000) ' perintah hentikan putaran motor
BacaSwitch

SetPosisi_Atas

End Sub

Private Sub Option8_Click()
StartTxt.Text = ""
EndTxt.Text = ""
StartTxt.Enabled = True
EndTxt.Enabled = True

End Sub

Private Sub StartTxt_Change()
StartNumber% = Val(StartTxt.Text) - 1
End Sub

```

```

Private Sub EndTxt_Change()
EndNumber% = Val(EndTxt.Text)
End Sub

Sub CariPosisiStart()
Dim StartNumber As Integer, naik As Integer

StartNumber = Val(StartTxt.Text) - 1
NomorPosisiSample = Val(ReadPort And &H1F) + 25 * PosisiDetektor
NoSampleTxt = NomorPosisiSample

If NomorPosisiSample = StartNumber Then GoTo Done
Call DAPI_WriteIoPorts(hDevInstance, &H1000, &H1000)
Call DAPI_WriteIoPorts(hDevInstance, &H0, &H1000)
Tdelay 2000

While NomorPosisiSample <> StartNumber
If Val(ReadPort And &H1F) = &H1F And naik = 1 Then
Call DAPI_WriteIoPorts(hDevInstance, &H0, &H1000) ' perintah gerak kebawah
naik = 0
End If

If Val(ReadPort And &H1F) <> &H1F And naik = 0 Then
Call DAPI_WriteIoPorts(hDevInstance, &H1000, &H1000) 'perintah gerak keatas
naik = 1
Tdelay 500 ' tunggu sampai switch benar-benar tertekan

Call DAPI_WriteIoPorts(hDevInstance, &H0, &H1000) 'perintah STOP
BacaSwitch
Call DAPI_WriteIoPorts(hDevInstance, &H1000, &H1000) 'perintah gerak keatas
naik = 1
Tdelay 500
NoSampleTxt.Text = Val(NoSampleTxt.Text) + 1
Tdelay 5000
DoEvents
End If
Wend

Call DAPI_WriteIoPorts(hDevInstance, &H0, &H1000) ' perintah hentikan putaran motor

NoSampleTxt.Text = Val(StartTxt.Text)
Done:

SetPosisi_Atas
NoSampleTxt.Text = Val(NoSampleTxt.Text)

End Sub

Sub CariPosisiTarget()
Dim EndNumber As Integer, naik As Integer

EndNumber = Val(EndTxt.Text)
NomorPosisiSample = Val(ReadPort And &H1F) + 25 * PosisiDetektor
NoSampleTxt = NomorPosisiSample
If NomorPosisiSample = EndNumber Then GoTo Done
Call DAPI_WriteIoPorts(hDevInstance, &H1000, &H1000)
Call DAPI_WriteIoPorts(hDevInstance, &H0, &H1000)
DelayValue = Val(Text3.Text)
Tdelay 2000

```

```

While NomorPosisiSample <> EndNumber
  If Val(ReadPort And &H1F) = &H1F And naik = 1 Then
    Call DAPI_WriteIoPorts(hDevInstance, &H0, &H1000) 'perintah gerak kebawah
    naik = 0
  End If

  If Val(ReadPort And &H1F) <> &H1F And naik = 0 Then
    Call DAPI_WriteIoPorts(hDevInstance, &H1000, &H1000) 'perintah gerak keatas
    naik = 1
    Tdelay 500 ' tunggu sampai switch benar-benar tertekan

    Call DAPI_WriteIoPorts(hDevInstance, &H0, &H1000) 'perintah STOP
    BacaSwitch
    Option1.Value = True
    BarIndikator DelayValue

    Option1.Value = False
    ProgressBar1.Value = 0

    Call DAPI_WriteIoPorts(hDevInstance, &H1000, &H1000) 'perintah gerak keatas
    naik = 1
    Tdelay 500
    NoSampleTxt.Text = Val(NoSampleTxt.Text) + 1
    Tdelay 5000
    DoEvents
  End If
Wend

Call DAPI_WriteIoPorts(hDevInstance, &H0, &H10000) 'perintah hentikan putaran motor
NoSampleTxt.Text = Val(EndTxt.Text)

Done:

SetPosisi_Atas
NoSampleTxt.Text = Val(NoSampleTxt.Text) + 1

If Val(NoSampleTxt.Text) = 26 Then
  NoSampleTxt.Text = 1
End If

If Val(NoSampleTxt.Text) = 51 Then
  NoSampleTxt.Text = 26
End If

If Val(NoSampleTxt.Text) = 76 Then
  NoSampleTxt.Text = 51
End If

If Val(NoSampleTxt.Text) = 101 Then
  NoSampleTxt.Text = 76
End If

End Sub

Private Sub ExitCmd_Click()
End
End Sub

```

```

Sub BarIndikator(DelayValue&)
ProgressBar1.Max = DelayValue
Dim A As Long, b As Long
  A = DelayValue + GetTickCount
  b = 0
  While b < A
    b = GetTickCount
    ProgressBar1.Value = DelayValue - (A - b)
    DoEvents
  Wend
End Sub

Private Sub StartCountCmd_Click()
If Option6.Value = True Then
  CountingAll
End If

If Option7.Value = True Then
  CountCurrentSample
End If

If Option8.Value = True Then
  CariPosisiStart
  CariPosisiTarget
End If

If Option9.Value = True Then
  Gerak25Sample
End If

End Sub

Private Sub ResetCmd_Click()
Dim TargetNumber As Integer, naik As Integer
EndNumber = 25 + 25 * PosisiDetektor
NomorPosisiSample = Val(ReadPort And &H1F) + 25 * PosisiDetektor
NoSampleTxt = NomorPosisiSample
If NomorPosisiSample = EndNumber Then GoTo Done
  Call DAPI_WriteIoPorts(hDevInstance, &H1000, &H1000)
  Call DAPI_WriteIoPorts(hDevInstance, &H0, &H1000)

  Tdelay 2000

  While NomorPosisiSample <> EndNumber
    If Val(ReadPort And &H1F) = &H1F And naik = 1 Then
      Call DAPI_WriteIoPorts(hDevInstance, &H0, &H1000) 'perintah gerak kebawah
      naik = 0
    End If

    If Val(ReadPort And &H1F) <> &H1F And naik = 0 Then
      Call DAPI_WriteIoPorts(hDevInstance, &H1000, &H1000) 'perintah gerak keatas
      naik = 1
      Tdelay 500 ' tunggu sampai switch benar-benar tertekan

      Call DAPI_WriteIoPorts(hDevInstance, &H0, &H1000) 'perintah STOP
      BacaSwitch

      Call DAPI_WriteIoPorts(hDevInstance, &H1000, &H1000) 'perintah gerak keatas

```

```

    naik = 1
    Tdelay 500
    NoSampleTxt.Text = Val(NoSampleTxt.Text) + 1
    Tdelay 5000
    DoEvents
    End If
Wend

Call DAPI_WriteIoPorts(hDevInstance, &H0, &H10000) ' perintah hentikan putaran motor
NoSampleTxt.Text = EndNumber

Done:

SetKePosisiA
NoSampleTxt.Text = Val(&H1)

End Sub

```

### My Module

```

Option Explicit
Dim sAppName As String ' Nama Aplikasi
Public sDevSymName As String ' Nama simbolik USB, Contoh: "UsbI2cIo"
Public byDevInstance As Byte ' Nomor device yang dipilih
Public hDevInstance As Long ' handle device yg dipilih
Public ReadI2c, IpsSerialId As Byte
Dim bDevicePresent As Boolean ' flag yg mengindikasikan keberadaan device
Dim MySerialID As Boolean

Sub InitDevice()
    sDevSymName = "UsbI2cIo" ' Nama simbolik USB
    sAppName = "ControlRIA" ' Nama Aplikasi
    'Counterfrm.Caption = sAppName & " - no device" ' Caption Aplikasi
    byDevInstance = 255 ' initial device(255 = no device)
    hDevInstance = INVALID_HANDLE_VALUE ' inialisasi file handle
End Sub

Sub OpenDevice()
    Dim I As Byte
    For I = 0 To 127 Step 1
        If (OpenDiHandle(I)) Then
            ' Suksessss!!!!!!!
            Exit For
        End If
    Next I
    ' Memberikan Pesan Suksessss atau Gagalllll
    If (I = 128) Then
        ' Tak ada device yg terpasang
        Simulation = 1
        Call MsgBox("No UsbI2cIo Devices were detected", vbOKOnly, "Device Open Error")
    Else
        ' Device ditemukan
        Simulation = 0

        byDevInstance = I
        'Counterfrm.Caption = sAppName & " - " & sDevSymName & Format(i) ' Application Title
    End If
End Sub

```

```

Function OpenDiHandle(byDevInstance As Byte) As Byte
    CloseDiHandle
    hDevInstance = DAPI_OpenDeviceInstance(sDevSymName, byDevInstance)
    If (hDevInstance <> INVALID_HANDLE_VALUE) Then
        OpenDiHandle = 1
    Else
        OpenDiHandle = 0
    End If

End Function

Sub SetConfig(PC, PB, PA) 'Output=0,Input=1
    Dim Data As Long
    Data = (PA * 255) + (PB * 65280) + (PC * 458752)
    If DAPI_ConfigIoPorts(hDevInstance, Data) Then
        ' Pemanggilan fungsi Suksesss!!!!
        Call MsgBox("Device OK", vbOKOnly, "Device Configuration OK")
    Else
        ' Gagallllll!!!!If Simulation = 1 Then Exit Sub
        Simulation = 1
        Call MsgBox("No UsbI2cIo Devices were detected", vbOKOnly, "Device Configuration Error")
    End If
End Sub

Sub CloseDiHandle()
    If hDevInstance <> INVALID_HANDLE_VALUE Then
        If DAPI_CloseDeviceInstance(hDevInstance) Then
            '
            Else
                '
            End If
            hDevInstance = INVALID_HANDLE_VALUE
        End If
    End Sub

Function CheckDevice() As Boolean
    If hDevInstance = INVALID_HANDLE_VALUE Then
        If OpenDiHandle(byDevInstance) Then
            CheckDevice = True
        Else
            CheckDevice = False
        End If
    ElseIf DAPI_DetectDevice(hDevInstance) Then
        CheckDevice = True
    Else
        CloseDiHandle
        CheckDevice = False
    End If
End Function

Sub Wait(d As Integer) 'd : detik
    Dim CurTime As Long
    Dim NextTime As Long
    Dim DifTime As Integer
    CurTime = Timer
    Do
        DoEvents
        NextTime = Timer
        DifTime = NextTime - CurTime
        If DifTime >= d Then
            Exit Do
        End If
    Loop
End Sub

Function ReadPort()

```

```

Dim Data As Integer
Dim DataPort As Long
Data = DAPI_ReadIoPorts(hDevInstance, DataPort)
ReadPort = DataPort
End Function

'=====
IWritten = DAPI_WriteI2c(hDevInstance, I2cTrans)
'=====

If (IWritten = 2) Then
    ' function call ok
Else
    ' function call failed
    Call MsgBox("Incorrect Return value", vbOKOnly, " Error calling DAPI_WriteI2C() function")
End If
End Sub

Sub WriteI2cIo(I2cData As Byte, I2cAddress As Byte)
    Dim I2cTrans As I2C_TRANS ' Dimension an I2C_TRANS structure
    Dim IWritten As Long      ' Dimension a long to hold the returned value
    Dim DevId As Integer
    If Simulation = 1 Then Exit Sub
    Select Case I2cAddress
        Case 0:
            DevId = &H70 'P0
        Case 1:
            DevId = &H72 'P1
        Case 2:
            DevId = &H74 'P2
        Case 3:
            DevId = &H76 'P3
        Case 4:
            DevId = &H78 'P4
        Case 5:
            DevId = &H7A 'P5
        Case 6:
            DevId = &H7C 'P6
        Case 7:
            DevId = &H7E 'P7
    End Select

    I2cTrans.byDevId = DevId      ' PCF8574 device
    I2cTrans.byType = I2C_TRANS_NOADR ' device does not use sub-address
    I2cTrans.wMemAddr.hi = 0      ' unused for I2C_TRANS_NOADDR
    I2cTrans.wMemAddr.lo = 0      ' unused for I2C_TRANS_NOADDR
    I2cTrans.wCount.hi = 0        ' only writing 1 byte, so set to 0
    I2cTrans.wCount.lo = 1        ' writing 1 byte
    I2cTrans.Data(0) = I2cData    ' the actual data that will be written to PCF8591

    IWritten = DAPI_WriteI2c(hDevInstance, I2cTrans)
    If (IWritten = 1) Then
        ' function call ok
    Else
        ' function call failed
        Call MsgBox("Incorrect Return value", vbOKOnly, " Error calling DAPI_WriteI2C() function")
    End If
End Sub

Function ReadI2cIo(I2cAddress As Byte)
    Dim I2cTrans As I2C_TRANS      ' Dimension an I2C_TRANS structure
    Dim IRead, IWritten As Long     ' Dimension a long to hold the returned value
    Dim DevId As Integer
    If Simulation = 1 Then Exit Function
    Select Case I2cAddress

```



```

Case 0:
    DevId = &H71
Case 1:
    DevId = &H73
Case 2:
    DevId = &H75
Case 3:
    DevId = &H77
Case 4:
    DevId = &H79
Case 5:
    DevId = &H7B
Case 6:
    DevId = &H7D
Case 7:
    DevId = &H7F
End Select

I2cTrans.byDevId = DevId          ' PCF8574 device ID
I2cTrans.byType = I2C_TRANS_NOADR ' device does not use sub-address
I2cTrans.wMemAddr.hi = 0          ' unused for I2C_TRANS_NOADDR
I2cTrans.wMemAddr.lo = 0          ' unused for I2C_TRANS_NOADDR
I2cTrans.wCount.hi = 0            ' only reading 1 byte, so set to 0
I2cTrans.wCount.lo = 1            ' reading 1 byte, so set to 1

lRead = DAPI_ReadI2c(hDevInstance, I2cTrans)
If (lRead = 1) Then
    ' function call ok
    ReadI2cIo = I2cTrans.Data(0)
Else
    ' function call failed
    ReadI2c = 0
    Call MsgBox("Incorrect Return value", vbOKOnly, " Error calling DAPI_ReadI2C() function")
End If
End Function

```

## **Module1a**

```

Option Explicit
Public listval(1 To 5)

Declare Function GetTickCount Lib "kernel32" () As Long
Public Lamda, TCount, HalfLife, CountingMode%, Simulation%
Public sFile$, TxtFile$, LFile$, CFile$, ChiLog$, DATAPATH$, CMDY$, LMDY, Lnum, CurrentDate$,
Cd$, Cm$, Cy$
Public Thyromode%, HospitalName$, User$
Public lym, LYrMn$, today$, mark$, pos1, pos2, pos3, CYrMn$, CYM
Public PauseChn, batch, LowerChn, UperChn, UperChnx, LowerChnx
Public GetSpect, Pitchy, ModPitchy, SpectM As Double, JisMxSave, GetSpectRun As Byte
Public LastQCpass As Integer, QCp%, EWindowOK As Integer, PeakSpect As Long, ChnKiSmx As
Long
Public Acquis, smth, PeakCount, USBModuleNotFound%
Public LLDx, ULdx, DDx, DDY, ChnKis(0 To 251) As Long
Public ChnKic(0 To 25) As Long, XScale%

Public TesDate$(0 To 10), TestTime$(0 To 10), Thy(0 To 10) As Long, BBGRnd(0 To 10), Std(0 To 10)
As Long, RBgrnd(0 To 10)

```

```

Public MaxData As Integer, MaxDat As Integer, m As Integer, OperateDate$, RichText$, RtfEdit$,
RtfCFile ' ChnKic(j)
Public maxpicx, maxpicy, Y0, X0, Ly, lh, Lx, Dx, Dy, ScalY, ScalX, skala As Integer
Public EChannel As Byte, EWindow As Byte
Public ChnkiCmx, JiCmx, ValidCopy As Boolean
Public ReadingFileError As Integer
Private reader As Integer, presetcont As Integer, maxcount As Integer
Public CurrDate$, CurrTime$, TNUMB$, TDate$, TDay$, TTIME$, PNAME$, AGE$, ADr1$, ADr2$,
SENDR$, DIAGDR$, ISOTPS$, DiagNote$, note$(1 To 15), countResult$(1 To 10) '11
Public ElapsedTime As Long, ElapsedMonth%, IpsSerialId As String, MyAuthorizedNumber As Double,
InitSerialNo As Double
Public tulis As Integer, NomorPosisiSample As Integer

```

```
Sub Main()
```

```
Rem Inisialisasi USB
```

```
InitDevice
```

```
OpenDevice
```

```
SetConfig 0, 0, 1
```

```
ControlFrm.Show
```

```
End Sub
```

```
Dim Data_To_I2c As Byte, I2c_Address As Byte
```

```
' dalam disain ini data yang akan ditulis ke bus AN2131QC dipersiapkan melalui PCF8547A (0) dengan
dibantu Three State Buffer 74244
```

```
' PCF 8257 (1) digunakan untuk mempersiapkan address AN2131QC (bit0 dan bit1) dan chip select
AN2131QC (bit2=0),serta enable untuk 74244 (bit3=0)
```

```
' perintah Read, Write, Open Gate menggunakan PortC DEVASYS bit 0, bit 1 dan bit 2
```

```
' prosedur yang dilakukan:
```

```
' 0: set PortC DEVASYS &H3 (Open Gate = low, Read = high , Write= high,
```

```
' 1: mempersiapkan address Inisialisasi (&H11) dari AN2131QC melalui PCF 8574(1)
&H03, maksudnya bit0=1 dan bit1=1 address Inisialisasi, chip select (bit2=0),enable
untuk 74244 (bit3=0)
```

```
' 2: mempersiapkan Inisialisasi data melalui PCF 8574 (0) &H38: Rem init AN2131QC
counter 0
```

```
' 3: kirim perintah Write menggunakan PortC DEVASYS &H2 (Open Gate = low, Read =
high , Write= low
```

```
' 4: Write selesai, kembalikan PortC DEVASYS ke &H3 (Open Gate = low, Read = high ,
Write= high,
```

```
' 5: mempersiapkan Inisialisasi data melalui PCF 8574 (0) &H78: Rem init AN2131QC
counter 1
```

```
' 6: kirim perintah Write menggunakan PortC DEVASYS &H2 (Open Gate = low, Read =
high , Write= low
```

```
' 7: Write selesai, kembalikan PortC DEVASYS ke &H3 (Open Gate = low, Read = high ,
Write= high,
```

```
' 8: mempersiapkan Inisialisasi data melalui PCF 8257(0) &HB8: Rem init AN2131QC
counter 2
```

```
' 9: kirim perintah Write menggunakan PortC DEVASYS &H2 (Open Gate = low, Read =
high , Write= low
```

```
' 10: Write selesai, kembalikan PortC DEVASYS ke &H3 (Open Gate = low, Read = high ,
Write= high,
```

```
' 11: kembalikan "enable" ke "disable" untuk 74244 (bit3=1) chip "unselect" (bit2=1),melalui
PCF PCF 8574 (1)
```

```
' &H07, maksudnya bit0=1 dan bit1=1 address Inisialisasi,
```

```

'mempersiapkan address Pra-Initialisasi 3, enabled=1, cs=1
  Data_To_I2c = &HF
  I2c_Address = &H1
  Call WriteI2cIo(Data_To_I2c, I2c_Address)

'set PortC DEVASYS &H3
  tulis = DAPI_WriteIoPorts(hDevInstance, &H30000, &HF0000)
'kita mulai
'mempersiapkan address Initalisasi 3, enabled=0, cs=0
  Data_To_I2c = &H3
  I2c_Address = &H1
  Call WriteI2cIo(Data_To_I2c, I2c_Address)

'mempersiapkan Initalisasi data
  Rem init AN2131QC counter 0
  Data_To_I2c = &H30
  I2c_Address = &H0
  Call WriteI2cIo(Data_To_I2c, I2c_Address)

'kirim perintah Write
tulis = DAPI_WriteIoPorts(hDevInstance, &H10000, &H70000)
'kirim Write selesai
tulis = DAPI_WriteIoPorts(hDevInstance, &H30000, &H70000)

'mempersiapkan Initalisasi data
  Rem init AN2131QC counter 0
  Data_To_I2c = &H70
  I2c_Address = &H0
  Call WriteI2cIo(Data_To_I2c, I2c_Address)

'kirim perintah Write
tulis = DAPI_WriteIoPorts(hDevInstance, &H10000, &H70000)
'kirim Write selesai
tulis = DAPI_WriteIoPorts(hDevInstance, &H30000, &H70000)

'mempersiapkan Initalisasi data
  Rem init AN2131QC counter 0
  Data_To_I2c = &HB0
  I2c_Address = &H0
  Call WriteI2cIo(Data_To_I2c, I2c_Address)

'kirim perintah Write
tulis = DAPI_WriteIoPorts(hDevInstance, &H10000, &H70000)
'kirim Write selesai
tulis = DAPI_WriteIoPorts(hDevInstance, &H30000, &H70000)

'kembalikan "enable" ke "disable" serta unselect chip AN2131QC
  Data_To_I2c = &HC
  I2c_Address = &H1
  Call WriteI2cIo(Data_To_I2c, I2c_Address)

End Sub

Sub Read_AN2131QC()

tulis = DAPI_WriteIoPorts(hDevInstance, &H80000, &H80000)
tulis = DAPI_WriteIoPorts(hDevInstance, &H0, &H80000)

End Sub

```

```
Sub Write_AN2131QC()
tulis = DAPI_WriteIoPorts(hDevInstance, &H40000, &H40000)
MsgBox "lho"
tulis = DAPI_WriteIoPorts(hDevInstance, &H0, &H40000)
MsgBox "lha"
End Sub
Sub Tdelay(thick)
Dim A As Long, b As Long
  A = thick + GetTickCount
  b = 0
  While b < A
    b = GetTickCount
  Wend
End Sub

Sub GetCurrDate()
CurrDate$ = (Format(Date, "mm/dd/yyyy"))
End Sub
Sub GetCurrTime()
CurrTime$ = Format(Time, "hh:mm:ss")
End Sub
```

