BAB III

PERANCANGAN SISTEM

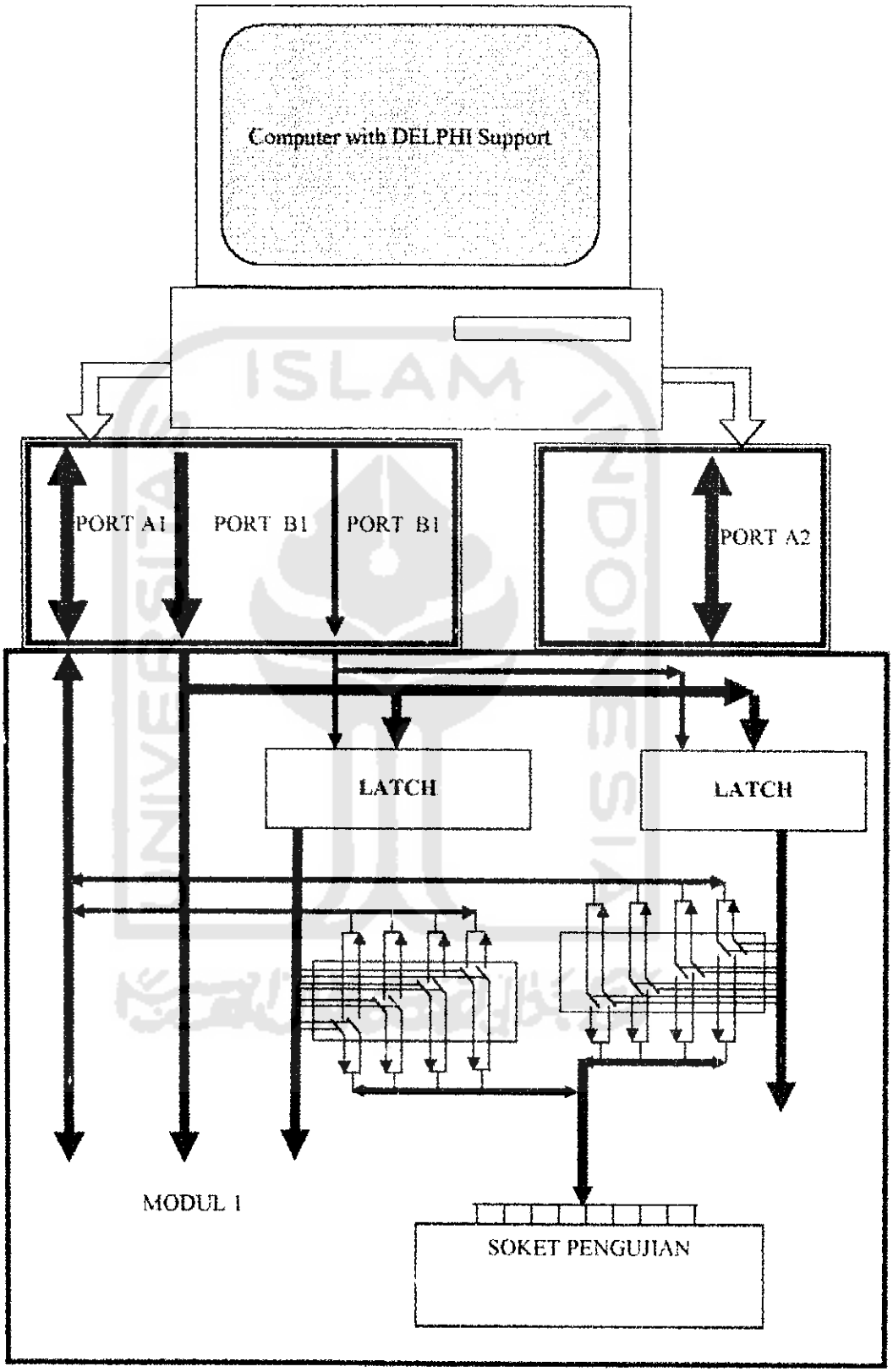
3.1 Perancangan Sistem

Alat penguji IC ini memanfaatkan *port* paralel komputer dan PPI (*Programmable Peripheral Interface*) 8255 serta beberapa IC pendukung yang digunakan sebagai kendali sinyal. Diagram kotak sistem yang dirancang seperti gambar 3.1.

Berdasarkan gambar 3.1 tampak bahwa modul terdiri atas komputer sebagai sumber sinyal kendali dan sinyal data. Untuk menghubungkan antara modul IC dan komputer digunakan dua buah PPI yang terhubung komputer dengan paralel *port* dan slot ISA.

Port PPI yang terdiri atas tiga buah *port* 8 bit yang dapat diprogram dihubungkan dengan perangkat pengendali di modul penguji IC. Dalam perancangan, satu *port* 8 bit digunakan sebagai jalur lalu lintas data dan dua *port* 8 bit lainnya digunakan sebagai jalur kendali.

Penggunaan *latch* dimaksudkan untuk menahan sinyal data kendali yang dikirimkan komputer yang digunakan sebagai penahan keadaan (*state*) untuk kendali saklar digital. *Latch* dirancang agar dapat diaktifkan sesaat untuk meneruskan data pada input sehingga tampak pada output. Untuk pemilihan pengaktifan *latch* dikendalikan perangkat lunak.



Gambar 3.1 Diagram Blok Sistem

Saklar digital menghubungkan bagian masukan dan keluaran IC yang diuji untuk dikirim atau dibaca sinyalnya. Kondisi saklar digital dikendalikan oleh perangkat lunak berdasarkan tipe IC yang akan diuji. Arah sinyal data yang akan ditulis atau dibaca dilayani oleh sepasang saklar digital yang arahnya dapat diatur dan dijaga dengan IC *latch* yang telah dijelaskan sebelumnya.

Berdasarkan diagram blok di atas komputer berfungsi untuk mengolah dan membaca data serta mengeluarkan sinyal kendali untuk menjalankan modul tes IC.

3.2 Perancangan Perangkat Keras

Perancangan perangkat keras dititikberatkan pada penjelasan mengenai komponen yang digunakan serta teknik antarmuka yang digunakan antara modul/perangkat pembentuk penguji IC dengan PPI yang digunakan.

Untai yang membentuk penguji IC ini terdiri atas beberapa rangkaian terintegrasi yang memiliki operasi tertentu dan dikendalikan oleh sinyal digital yang berasal dari komputer. Komponen terintegrasi tersebut dijelaskan di bawah ini.

3.2.1 Saklar Digital

Saklar digital digunakan untuk memutuskan atau menghubungkan antara kaki IC yang dideteksi dengan bus data yang menghubungkan komputer dengan kaki-kaki IC yang akan diuji (dalam hal ini soket IC). Karena IC yang akan dideteksi merupakan keluarga TTL maka saklar digital yang

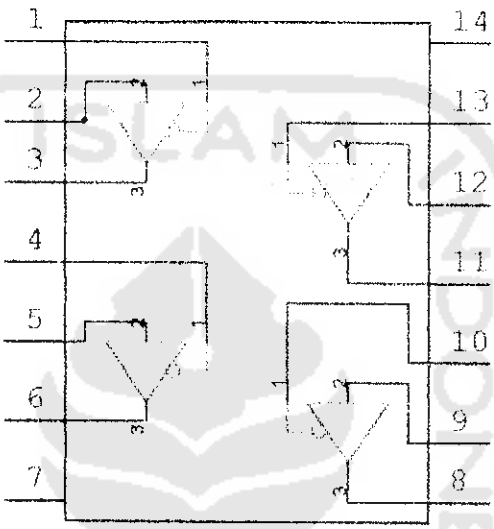
melayani/mengendalikan sinyal digital pun adalah IC yang mempunyai *output* keluaran TTL. Alasannya karena kemudahan antarmuka dan *power supply* yang digunakan menjadi lebih sederhana. Tabel 3.1 menjelaskan tentang karakteristik IC TTL yang menjadi pertimbangan dalam perancangan.

Tabel 3.1 Karakteristik IC TTL

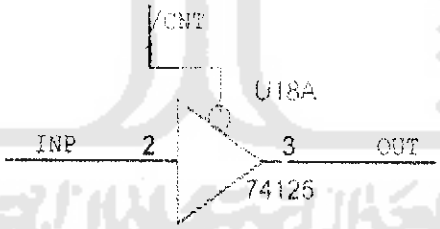
Deskripsi	Simbol	Keluarga IC	
		74HC	74LS
Konsumsi daya per gerbang (mW)		10	2
Speed Power Product (pJ)		90	18
LOW-level input voltage (V)	V_{IImax}	0,8	0,8
LOW-level output Voltage (V)	V_{OLmax}	0,4	0,5
HIGH-level input voltage (V)	V_{IHmin}	2,0	2,0
HIGH-level output voltage (V)	V_{OHmin}	2,4	2,4
LOW-level input Current (mA)	I_{IImax}	-1,6	-2,0
LOW-level output current (mA)	I_{OLmax}	16	8
HIGH-level input current (mA)	I_{IHmax}	40	20
HIGH-level output current (uA)	I_{OHmax}	-400	-400

Seperti yang dijelaskan pada bab II untuk saklar digital digunakan IC yang mempunyai karakteristik tiga keadaan yaitu "0", "1"" dan *High Impedance* atau "hi-Z". IC dengan keadaan tiga kondisi demikian disebut *tristate buffer* dimana level "0" dan "1" *tristate buffer* harus memiliki karakteristik TTL.

Salah satu IC *tristate buffer* yang memiliki karakteristik TTL adalah keluarga 74 LS atau 74 HC. Dalam perancangan ini IC yang digunakan adalah 74HC125 dan 74LS125. Kode HC memiliki kecepatan yang lebih tinggi dibanding LS. Gambar IC *tristate buffer* seperti pada gambar 3.2.



Gambar 3.2 Gambar Internal IC 74HC125



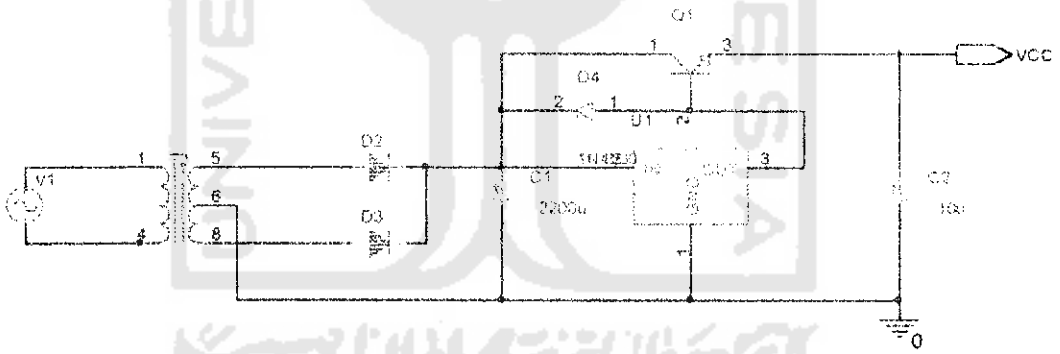
Gambar 3.3 Fungsi Kaki-kaki Pada IC *Tristate buffer*

Setiap satu gerbang IC *tristate buffer* terdiri atas tiga buah kaki sinyal dan dua buah kaki catu daya. Masing-masing kaki *tristate buffer* mempunyai fungsi tersendiri. Kaki /CNT berfungsi untuk menentukan hubungan antara kaki INP dan kaki OUT. Sinyal digital yang diterapkan pada kaki /CNT adalah, Bila diberi logika '0' kaki maka data yang terdapat pada kaki INP dapat dibaca pada kaki

OUT. Sedangkan bila diberi logika '1' maka antara kaki INP dan kaki OUT seolah-oleh terisolasi. Kondisi demikian disebut *high impedance*. Dalam kondisi high impedance maka sinyal digital yang terdapat pada kaki INP tidak dapat dibaca pada kaki OUT.

3.2.2 Rangkaian Catu Daya

Bagian catu daya menyediakan tegangan untuk rangkaian penguji IC dan juga IC yang diuji. Rangkaian ini dibentuk dengan transformator, dioda penyearah (D1 dan D2), IC 7805 dan komponen filter yang berupa kapasitor (C1 dan C2). Trafo berhubungan langsung dengan sumber tegangan jala-jala listrik 220 volt. Untai catu daya tampak seperti gambar 3.4



Gambar 3.4 Rangkaian Catudaya

IC 7805 untuk mempertahankan tegangan keluaran agar teregulasi pada tegangan 5 volt. Transistor Q1 untuk memberikan arus yang besar pada rangkaian sehingga kemampuan arus yang diserap oleh rangkaian penguji IC dan rangkaian PPI dapat tersedia cukup. Filter kapasitor digunakan untuk memperkecil riak

akibat ketidakhalusan keluaran dari dioda penyearah serta untuk menghilangkan pengaruh frekuensi tinggi dari jala-jala PLN yang dapat muncul.

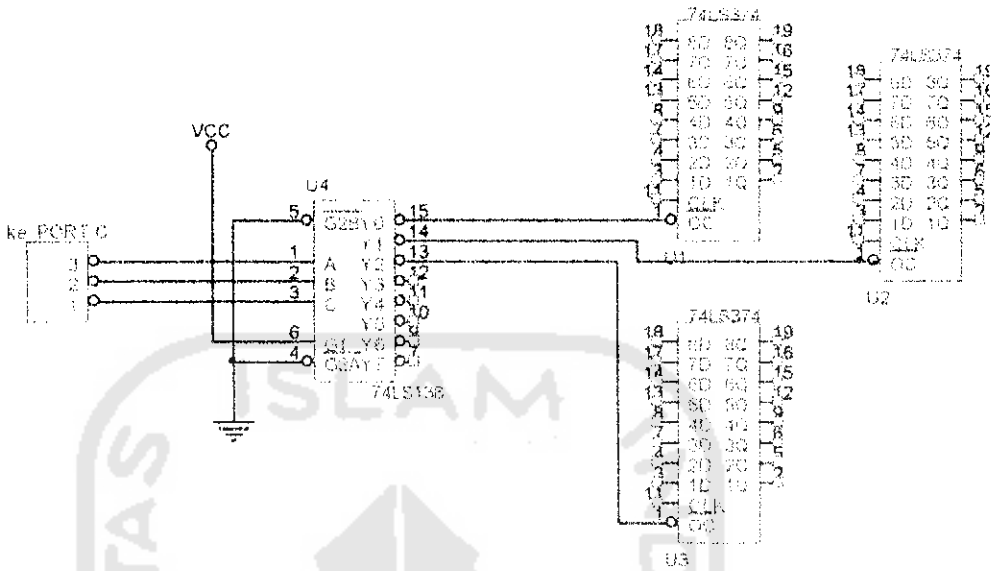
3.2.3 *Latch* (IC 74HC374)

Latch atau penahan digunakan untuk memegang sinyal kendali yang dikirimkan oleh komputer untuk mengendalikan arah data baik menuju *(input)* maupun keluar *(output)* IC yang akan diuji.

Rangkaian penahan tampak seperti gambar 3.4, dimana setiap kaki *output latch* terhubung dengan kaki kontrol gerbang *tristate buffer*. Untuk menentukan kondisi gerbang tinggal memberikan logika "0" atau "1" pada *output* IC 74LS374. Bila keluaran *latch* bernilai satu, maka *tristate buffer* berada dalam keadaan Hi-Z, hubungan masukan *tristate buffer* dalam keadaan mengambang, tidak berada dalam salah satu keadaan.

Sebagai penahan digunakan IC 74HC374 yang merupakan register 8 bit. 74HC374 terdiri atas 8 buah D *flip-flop edge-trigger* dengan sifat logika masukan yang akan muncul pada keluaran pada saat terjadi *edge-trigger* pada kaki *CLK*. Setiap keluaran *flip-flop* tergandeng dengan *tristate buffer* yang dikendalikan oleh kaki kendali dengan aktif "*high*".

Kaki $\overline{OC}$ setiap IC 74HC374 dihubungkan dengan kaki keluaran *decoder* , seperti gambar 3.5 pengaktifan setiap IC penahan *(latch)* dilakukan dengan mengirimkan kode biner tiga bit yang dimiliki oleh IC *decoder* 74HC138.



Gambar 3.5 Hubungan antara *decoder* 74HC138 dengan IC Penahan

Sehingga dengan menghubungkan kaki $\overline{OC}$ dengan keluaran *decoder* menghemat satu kaki yang terdapat pada PPI 8255. Tiga buah kaki *decoder* 3 ke 8 terhubung dengan 3 kaki PORT C *upper* yaitu kaki PC4, PC5 dan PC6, sedangkan kaki PC7 dihubungkan dengan semua kaki CLK IC *latch*.

Untuk menghubungkan salah satu kaki PORTC yang akan men-*drive* tiga buah kaki CLK perlu dipertimbangkan mengenai *fan-out* kaki PORT C.

3.2.4 Rangkaian PPI 8255

Untuk menghubungkan komputer dengan dunia luar digunakan sebuah IC antarmuka yaitu 8255. IC 8255 memiliki 3 buah *port* dengan masing-masing *port* berjumlah 8 jalur. IC ini dapat digunakan sebagai masukan ataupun keluaran dengan mengatur *control word*.

Terdapat dua mode operasi yang dapat dipilih melalui perangkat lunak. Yaitu:

- Mode 0 (fungsi dasar *input output*) konfigurasi fengsional ini menyediakan operasi *input* dan *output* yang sederhana untuk tiga port yang dimiliki. Tidak terjadi proses *handshaking*, data dibaca dan dituliskan dari dan ke port PPI 8255.
- Mode 1 Konfigurasi fungsional mode ini menyediakan cara mentransfer data I/O dari dan ke *port* dengan menggunakan sinyal *strobes* atau *handshaking*.

Pada tugas akhir ini untuk mengirimkan data dari dan ke *port* PPI menggunakan mode dasar yaitu mode 0. Pada mode 0 untuk menentukan fungsi *port*, adalah dengan memberikan nilai tertentu pada register *control word* maka IC 8255 dapat mempunyai fungsi keluaran atau masukan pada *port* A dan *port* B dan *port* C. Nilai yang diberikan untuk register *control word* yang menentukan operasi PPI 8255 pada mode 0 dapat dilihat pada tabel 3.2.

Dengan memberikan nilai tertentu pada register *control word* maka IC 8255 dapat mempunyai fungsi keluaran atau masukan pada *port* A dan *port* B dan *port* C. Nilai yang diberikan untuk register *control word* yang menentukan operasi PPI 8255 dapat dipelajari pada lampiran.

Tabel 3. 2 Operasi Mode 0

No	Control Word	Grup A		Group B	
		Port A	Port C upper	Port B	Port C lower
0	$80	Output	Output	output	output
1	$81	Output	Output	Output	input
2	$82	Output	Output	Input	Output
3	$83	Output	Output	Input	input
4	$88	Output	Input	Output	output
5	$89	Output	Input	Output	input
6	$8A	Output	Input	output	input
7	$8B	output	Input	Input	input
8	$90	Input	Output	Output	output
9	$91	Input	Output	Output	input
10	$92	Input	Output	Input	output
11	$93	Input	output	Output	output
12	$98	Input	Input	Output	output
13	$99	Input	Input	Output	input
14	$9A	Input	Input	Input	output
15	$9B	Input	Input	Input	Input

3.3 Perancangan Perangkat Lunak *(Software)*

Membuat program berarti merencanakan serangkaian instruksi yang dapat dimengerti oleh komputer dan disusun menurut urutan yang logis. Pekerjaan

tentang penulisan program tersebut tergantung pada pengertian tentang persoalan yang dihadapi dan pada struktur atau rencana penyelesaian-penyelesainnya. Pekerjaan membuat program ini dipecahkan dalam beberapa tahapan, yaitu :

1. Menyatakan persoalan yang dihadapi sejelas mungkin dan secara terperinci
2. Menyusun Algoritma, yaitu prosedur penyelesaian persoalan secara bertahap
3. Menyusun *Flow Chart* atau peta prosedur penyelesaian (diagram alir) yang menguraikan algoritma secara terperinci
4. Menerjemahkan peta prosedur penyelesaian dalam bahasa yang dapat diproses oleh komputer

Menyusun algoritma merupakan tahapan yang penting setelah persoalan didefinisikan. Pada dasarnya algoritma ini dari langkah-langkah sederhana yang dapat diartikan sebagai penjabaran proses dari keadaan awal ke keadaan akhir. *Flow chart* sangat membantu dalam pembuatan program yang terstruktur dengan baik. *Flow chart* adalah gambaran dari penyelesaian suatu masalah langkah demi langkah dengan menggunakan simbol-simbol tertentu.

Dalam perancangan terlebih dahulu akan dirancang perangkat lunak yang digunakan untuk menguji sistem *board* perangkat keras. Kehandalan alat penguji ini ditentukan terutama oleh kemampuan sistem perangkat keras yang diuji.

Perancangan perangkat keras meliputi perancangan sistem penguji perangkat keras (*main board*) dan perancangan sistem perangkat lunak secara keseluruhan (*main control software*).

Pengujian sistem *board* terutama ditujukan untuk mengetahui kondisi perangkat keras agar bekerja sesuai dengan apa yang kita inginkan. Perangkat lunak yang dibuat berupa program yang ditulis dengan bahasa pemrograman delphi untuk menguji sistem I/O perangkat keras sehingga kerjanya sesuai dengan yang kita inginkan. Tahapan perancangan program pengujian perangkat keras ini adalah sebagai berikut :

1. Mengetahui sistem input dan output perangkat keras
2. Merancang algoritma perangkat lunak (program)
3. Membuat kode program
4. Pengujian langsung ke perangkat keras
5. Evaluasi hasil pengujian

Untuk langkah a sampai c akan di jelaskan pada bab III ini sementara untuk langkah pengujian akan dijelaskan pada BAB IV.

Sistem I/O perangkat keras telah di jelaskan pada perancangan perangkat keras. Seperti yang telah dijelaskan diatas bahwa masukan bagi papan pengujian IC ini berasal dari pararel *port* komputer sedangkan untuk keluaran dari papan penguji menuju pararel *port* yang terpasang pada bus ISA.

Masukan berupa LPT1 dengan register sebagia berikut:

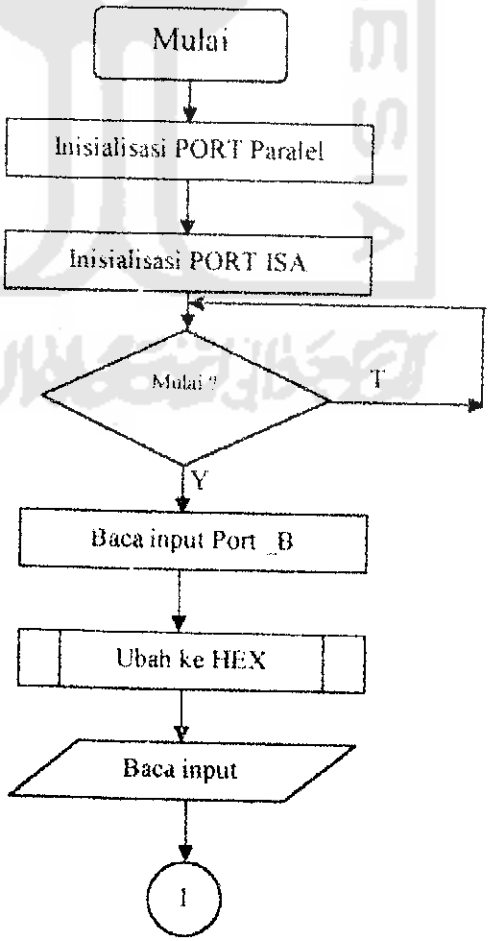
1. *Register Data Port* merupakan jalur data yang digunakan untuk lalu lintas data, memiliki panjang 8 bit dan arahnya *bidirectional* serta mempunyai alamat $378 dan mempunyai level TTL .
2. *Register Printer Control* merupakan jalur kendali untuk sinyal kontrol printer terdiri dari 8 bit dengan bit 5,6 dan 7 tidak

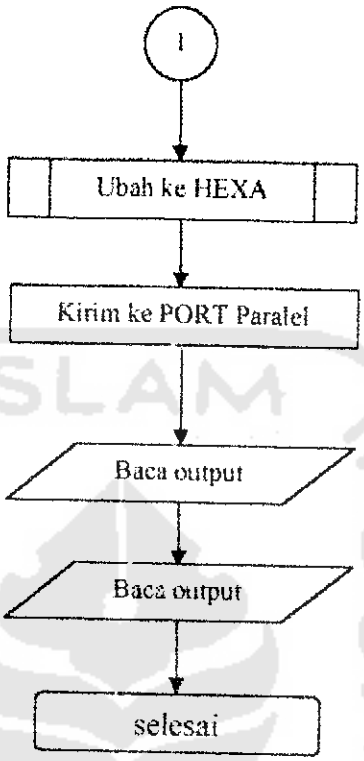
digunakan pada sistem PPI, paralel yang digunakan terhubung dengan pengendali untuk menentukan *port* yang dipakai.

3. *Register Printer Status* merupakan jalur status yang digunakan untuk mengetahui status selama proses pencetakan.

3.3.1 Merancang Algoritma Perangkat Lunak Penguji Perangkat keras

Setelah menentukan masukan dan keluaran untuk sinyal data maupun sinyal kendali langkah selanjutnya adalah membuat algoritma program. Metode diagram alir digunakan untuk merancang algoritma yang akan digunakan. Adapun algoritma tersebut adalah seperti gambar 3.6





Gambar 3.6 Diagram Alir Program Pengujian Perangkat Keras

Blok inisialisasi merupakan kode yang digunakan untuk menginisialisasi PPI *card* yang terhubung dengan *port* paralel. PPI *card* yang terhubung dengan *parallel port* mempunyai IC 8255 yang harus diinisialisasi terlebih dahulu.

Inisialisasi IC PPI 8255 adalah dengan membuat seluruh *port* yang dimiliki oleh PPI menjadi keluaran. Dari sub bab 3.2.4 diperoleh bahwa agar *port* PPI menjadi keluaran register kendali (*control register*) harus dituliskan dengan data 80H.

Berdasarkan data perangkat keras modul PPI diperoleh hubungan antara konfigurasi LPT dengan modul PPI. Seperti tabel 3.3.

Tabel 3.3 Hubungan Modul PPI dengan LPT

PORT	Urutan Data								
PORT DATA	D7	D6	D5	D4	D3	D2	D1	D0	
MODUL PPI	X	X	X	RD	CS	WR	A1	A0	
PORT CONTROL	D7	D6	D5	D4	$\overline{D3}$	D2	$\overline{D1}$	$\overline{D0}$	
MODUL PPI	X	X	X	X	PPI CNTR	157 SLC	574 EN	PPI DT	

Keterangan:

X adalah logika tidak peduli

$\overline{\ }$ Adalah kondisi inverting (membalik)

PPI CNTR adalah *PPI control pin*

157 SLC adalah *157 select pin*

574 EN adalah *74HC574 enable pin*

PPI DT adalah pin pemilih data PPI

Berdasarkan tabel 3.3 dapat dibuat keadaan yang akan digunakan sebagai algoritma bagi penanganan program penguji perangkat keras tersebut.

Pertama-tama adalah menginisialisasi IC PPI 8255 yang terdapat pada modul FPI dengan cara mengisi *port register control*nya dengan data untuk menentukan mode operasi dari IC PPI 8255 yang terdapat pada modul PPI. Kode sumber yang digunakan ditempatkan pada prosedur inisialisasi sebagai berikut:

```
    procedure TForm1.inisialisasi(dt:byte);
begin
 port_masuk(LPTIO,$FF);
 port_masuk(LPTIO,dt);      {kirim $80 hexa ke $378}
```

```
  delay(2);
  port_masuk(LPTIO+2,SFF);   {0000}   pastikan terkirm ke output
74hc574
  port_masuk(LPTIO+2,SFE);   {1000}
  port_masuk(LPTIO+2,SFC);   11111100 biar aman
  { memasukkan control word ke PPI : }
  { port data    :  D7 D6 D5 D4  D3 D2 D1 D0    }
{                 ..... - - -  RD  CS WR A1 A0    }

  port_masuk(LPTIO,$17);      WR high  00010111
  port_masuk(LPTIO+2,SF4);    aktifkan pemilih kontrol PPI
  port_masuk(LPTIO+2,SFC);    matikan kontrol pemilih PPI
  port_masuk(LPTIO,$13);      WR low   00010011 = $13
  port_masuk(LPTIO+2,SF4);
  port_masuk(LPTIO+2,SFC);
  port_masuk(LPTIO,$17);      WR high
  port_masuk(LPTIO+2,SF4);    aktifkan pemilih kontrol PPI
  port_masuk(LPTIO+2,SFC);    matikan kontrol pemilih PPI
  port_masuk(CW_PPI,$91);     inisialisasi PPI card ISA

end;
```

Untuk menginisialisasi seperti kode sumber diatas adalah pertama-tama siapkan data mode operasi (dalam hal ini $80) pada *port* data LPT. Kemudian mengaktifkan IC penahan 74HC574 untuk menahan data pada keluaran IC 74HC374 (dalam modul keluaran 74HC374 terhubung dengan *port* data IC 8255, lihat lampiran). Aktifkan alamat kendali register IC 8255 dengan memberikan instruksi pin *WR* diberi logika '1' kemudian '0' dan kembali '1'. Sehingga sekarang mode operasi IC 8255 adalah semua *port* PPI 8255 menjadi *port* keluaran (karena *port control register* di beri nilai $80, lihat sub bab 3.2.4).

Untuk mengirimkan data dari LPT dilakukan dengan instruksi berikut ini.

```
  procedure TForm1.outA(dt:byte);
begin
  port_masuk(LPTIO+2,SFF);   {1000}
  port_masuk(LPTIO,dt);      {data yg dikirim }
```

```
 port_masuk(LPTIO+2,SFE);   {0000}   pastikan terkirm ke output
74hc574
 port_masuk(LPTIO+2,SFF);
 port_masuk(LPTIO+2,SFC);
 { port data   :  D7 D6 D5 D4  D3 D2 D1 D0   }
{              ..... - - -  RD  CS WR A1 A0   }

 port_masuk(LPTIO,S14);      WR high   00010100 = S14
 {                                    }
{ port control :  D3     D2      D1      D0    }
{          ....  PPIctrl 157stc   574en   PPIdt  }

 port_masuk(LPTIO+2,SF4);          aktifkan pemilih konrol PPI
11110100
 port_masuk(LPTIO+2,SFC);          matikan kontrol pemilih PPI
11111100
 port_masuk(LPTIO,S10);      WR low    00010000 = S10
 port_masuk(LPTIO+2,SF4);
 port_masuk(LPTIO+2,SFC);
 port_masuk(LPTIO,S14);     WR high
 port_masuk(LPTIO+2,SF4);    aktifkan pemilih kontrol PPI
 port_masuk(LPTIO+2,SFC);    matikan kontrol pemilih PPI
end;
```

Untuk mengakses *port* sebenarnya sama saja dengan instruksi untuk menginisialisasi IC 8255. Perbedaannya adalah terletak pada data yang digunakan untuk pemilih alamat *register port*, dimana kombinasi antara pin RD, CS, WR, A1 dan A0 berbeda tergantung pada alamat *register* yang ditentukan oleh A1 dan A0.

3.3.2 Perancangan Perangkat Lunak Penguji IC

Program penguji IC berfungsi untuk mengendalikan perangkat keras agar bekerja sesuai dengan prosedur dan langkah-langkah pengujian. Program perangkat perangkat lunak penguji IC terdiri atas program aplikasi dan program *database* berbasis *database access* yang dimiliki oleh *Microsoft Office*.

3.3.3 Perancangan *Database*

Database menyimpan dan mengorganisasi informasi serta data-data penting yang akan digunakan untuk mempermudah pengaksesan (penyimpanan dan pengambilan) data. Salah satu model *database* yang digunakan adalah *database* relasional . Dalam *database* seperti ini, sebuah *database* tersusun atas sejumlah tabel.

Setiap tabel tersusun atas sejumlah *record*. Setiap *record* mempunyai sejumlah *field* yang sama.

Implementasi *database* pada prinsipnya terbagi dua , yaitu:

1. Model pertama mengemas seluruh data yang terkait dalam sebuah *database* kedalam sebuah berkas. Model seperti ini dijumpai pada *access*, *Interbase* dan kebanyakan *server* SQL lainnya.
2. Model kedua menggunakan sejumlah berkas untuk menyimpan data, indeks dan hal-hal lain yang terkait dengan *database*. Biasanya keseluruhan file disimpan pada direktori yang sama. Dbase, Foxpro dan paradox termasuk dalam kategori ini.

Dalam tugas akhir ini database yang digunakan adalah *database Microsoft Access*. *Database access* mempunyai kemampuan dimana *access* mendukung.

Tabel 3.4 Gambar Rancangan *Database*

ID_IC	NO_SERI	NM_IC	UJI_1	UJI_2	HASIL_1	HASIL_2	GAMBAR
IC001	7400	NAND	(0000)(0000)	(1111)(1111)	(1111)	(0000)	C:\My Documents\My Pictures\crayon.BMP
IC002	74002	NOR	(0000)(0000)	(1111)(1111)	(1111)	(0000)	C:\My Documents\My Pictures\compteur.BMP

Pemrograman SQL *(Structured Query Language)* yang memungkinkan pengaksesan setiap kolom ataupun *record* dengan perintah-perintah SQL.

Tabel yang dirancang terdiri atas *field-field* sebagai berikut:

- ID_IC , ID_IC merupakan nomor urut IC yang dibuat, pada *field* ini setiap nomor IC harus unik artinya berbeda antara dalam satu *field* harus berbeda. ID_IC akan digunakan sebagai kunci *(key)*. Kunci menjadi penting bila akan mengakses *database* dari program aplikasi .
- NO_SERI, NO_SERI adalah *field* yang menyimpan data seri IC berdasarkan seri IC yang dikeluarkan pabrik.
- NM_IC, merupakan *field* yang digunakan untuk menyimpan jenis gerbang yang terdapat pada IC misalkan : NAND, NOR, OR dan lain-lain.
- UJI_1, merupakan *field* yang berisi data untuk pengujian masukan bagi IC yang digunakan untuk pengujian pertama.
- UJI_2, merupakan *field* yang berisi data untuk pengujian masukan bagi IC yang digunakan untuk pengujian kedua.
- HASIL_1, merupakan *field* yang berisi data hasil pengujian berdasarkan tabel kebenaran dari *field* pengujian pertama.
- HASIL_2, merupakan *field* yang berisi data hasil pengujian berdasarkan tabel kebenaran dari *field* pengujian kedua.
- GAMBAR, merupakan gambar IC yang akan dipanggil sesuai dengan jenis IC, NM_IC yang bersesuaian.

Setiap *field* yang digunakan menggunakan tipe data string hal ini digunakan untuk mempermudah pengaksesan semata. Dalam pengaksesan data pada *database* dengan program aplikasi dimana tipe data string pada setiap *field*, akan dikonversi menjadi tipe data lain yang bersesuaian yang digunakan untuk komputasi pada program penguji IC.

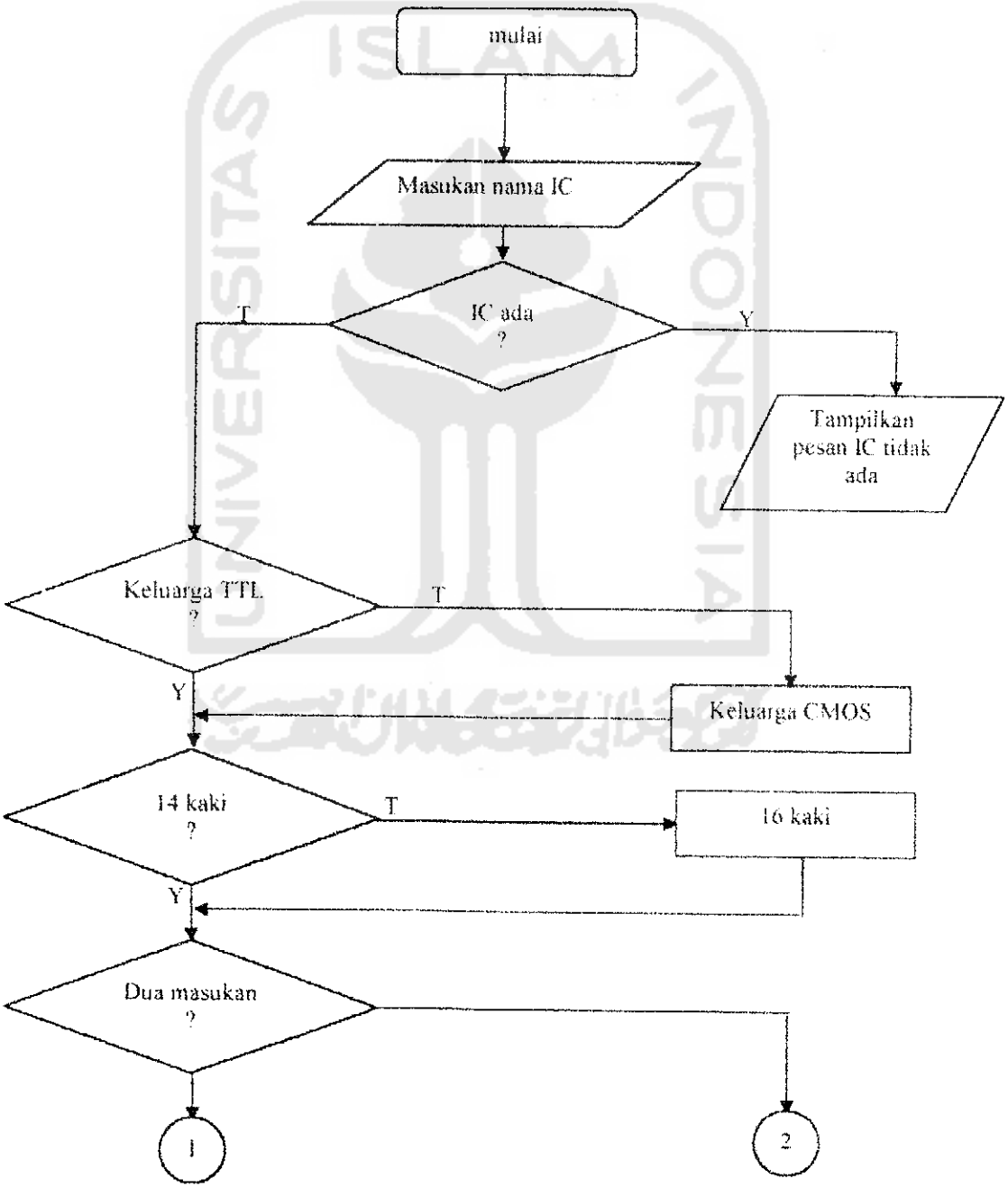
3.3.4 Perancangan Program Aplikasi Penguji IC

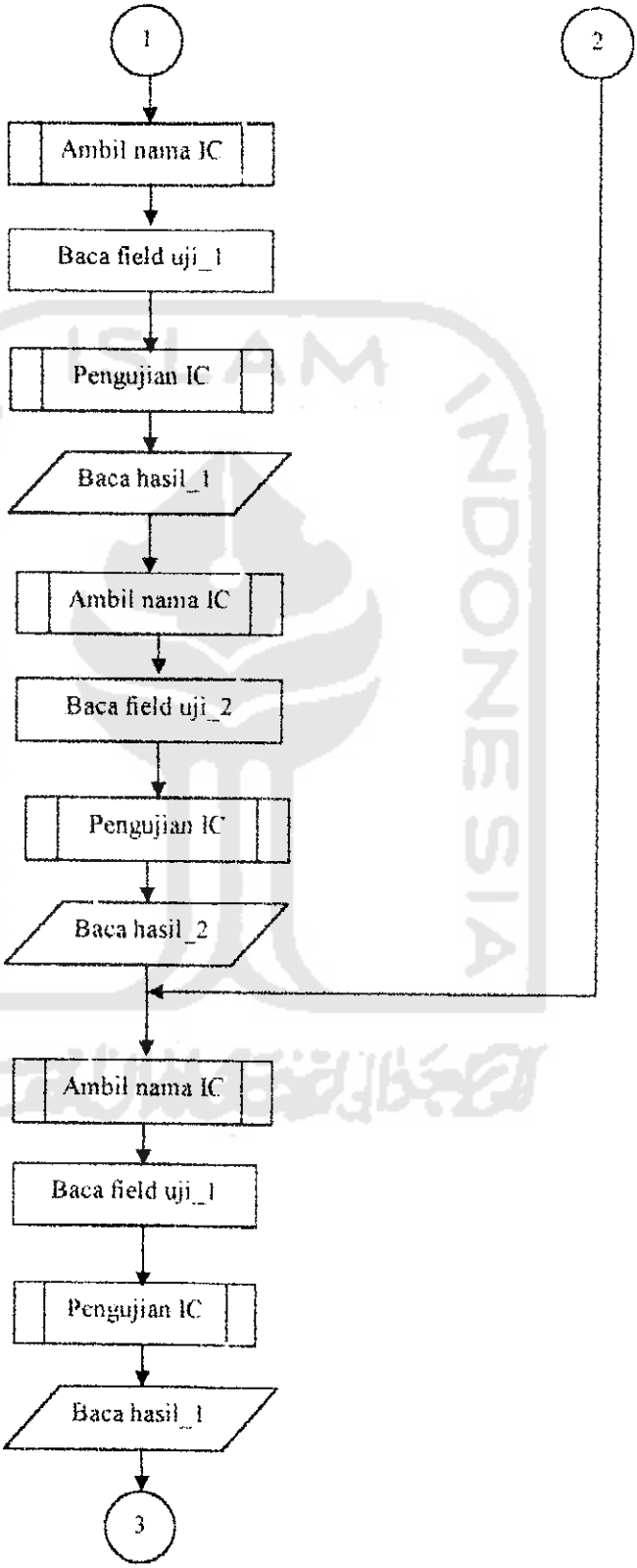
Setelah merancang perangkat lunak untuk menguji papan perangkat keras yang digunakan untuk mengetahui berfungsi atau tidaknya komponen yang mendukung sistem/rangkaian penguji IC tersebut. Perancangan berikutnya adalah membuat algoritma program yang digunakan untuk menguji IC yang mengintegrasikan antara database dan program aplikasi pengujian IC tersebut. Program Penguji IC ini menggunakan bahasa pemrograman visual Delphi 6.

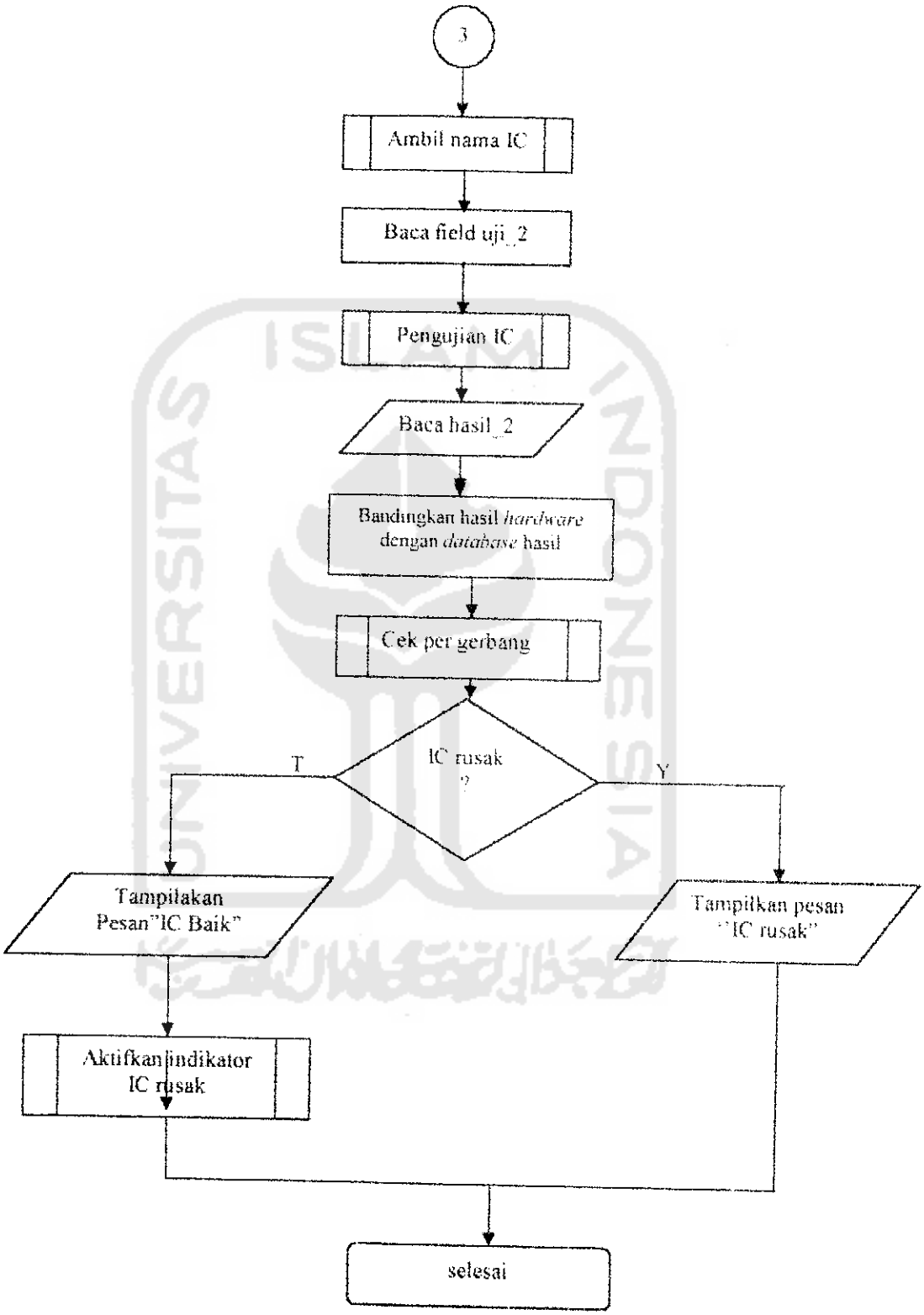
Algoritma program dibuat dengan metode *flowchart* seperti pada gambar 3... dalam diagram alir, mula-mula di*check* nama IC yang terdapat pada *database* dengan perintah sebagai berikut

```
frmUtama.Q1.SQL.Text:='SELECT*FROM ic WHERE
M_IC='''+DT_IC.Text+''''
frmUtama.Q1.Open;
Data_slh:=frmUtama.Q1.FieldByName('NM_IC').AsString;
if (Data_slh='') then
begin
MessageDlg('nama IC tidak ada',mtInformation,[mbOk],1);
exit;
end;
FileGambar:=frmUtama.Q1.FieldByName('gambar').AsString;
Image1.Picture.LoadFromFile(FileGambar);NOSR.Text:=frmUtam
a.Q1.FieldByName('no_seri').AsString;
```

Kemudian ditentukan apakah nama IC tersebut ada atau tidak ada yang di masukan ke variable *Data slh* dan diuji . Bila nama IC tidak ada maka akan ditampilkan sebuah *message dialog* yang menjelaskan bahwa nama IC tersebut tidak terdapat pada *database*.




1
2
Ambil nama IC
Baca field uji_1
Pengujian IC
Baca hasil_1
Ambil nama IC
Baca field uji_2
Pengujian IC
Baca hasil_2
Ambil nama IC
Baca field uji_1
Pengujian IC
Baca hasil_1
3



Gambar 3.7 Diagram Alir Program Pengujian IC

Pengujian IC terdiri atas dua buah pengujian yaitu UJI_1 dan UJI_2. untuk mendapatkan hasil pengujian yang akurat setiap kombinasi logika masukan harus diterapkan pada masukan IC yang akan diuji. Perintah untuk melakukan pengujian terdapat pada komponen *radio button* yang dimiliki oleh Delphi. Setiap *even* /kejadian berupa melakukan *check* pada komponen *radio button*, maka akan dieksekusi perintah/*statement* yang terdapat pada *even radio button click.*

```
procedure TForm1.RadioButton3Click(Sender: TObject);
begin
    RadioButton2.Checked:=False;
    RadioButton3.Checked:=true;
    if RadioButton3.tag <> 1 then
    exit;
    frmUtama.Q1.SQL.Text:='SELECT * FROM ic WHERE
NM_IC='''+DT_IC.Text+''';
    frmUtama.Q1.Open;
    Data_uji2:=frmUtama.Q1.FieldByName('UJI_2').AsString;
    uji2.text:=data_uji2;
    //StrtoURUTBIN(Data_uji2);
end;
```

Prosedur diatas digunakan untuk memilih pengujian yang kedua dengan ditandai oleh kejadian bila kita mengklik pada *radiobutton.* Dalam prosedur/kejadian diatas juga pengujian yang pertama di *disable* dengan perintah *RadioButton2.Checked:=False* dan *RadioButton3.Checked:=true.*

Kemudian nama IC diambil oleh perintah *frmUtama.Q1.SQL.Text:='SELECT*FROM ic WHERE M_IC='''+DT_IC.Text+''''* Maka nama IC akan di ambil dari *database.*

Data pengujian pertama yang terdapat pada *field* UJI_1 diambil dengan perintah

Data_uji2:=frmUtama.Q1.FieldByName('UJI_2').AsString;

Data pada *field* uji diberikan pada variable *Data_uji2* yang masih berupa data dalam tipe string. Selanjutnya data tersebut akan digunakan pada prosedur*/even radiobutton click* akan memanggil fungsi yang berfungsi untuk mengkonversi data string menjadi tipe data integer dan menentukan jenis pengujian yang akan dilakukan oleh program.

Selanjutnya program akan mengirimkan data hasil pembacaan pada uji_2 untuk dikirimkan ke perangkat keras penguji IC. Adapun program yang digunakan seperti perintah-perintah pada sub bab 3.3.1.

Hasil keluaran IC dibaca oleh PPI modul yang terhubung dengan slot ISA. Pembacaan data oleh modul PPI yang terhubung dengan slot ISA dilakukan oleh perintah berikut.

```
Function TForm1.in_port(alamat:word):byte;
var   data : byte;
begin
  asm
     mov    dx,alamat
     in     al,dx
     mov    data,al
  end;
  in_port := data;
end;
```

Parameter format alamat yang bertipe *word* digunakan untuk menentukan alamat slot ISA yang digunakan oleh modul PPI ISA. Dalam hal ini IC 8255 yang dipasang pada PPI slot ISA , untuk *register Port* A menggunakan alamat $300, untuk *register Port* B menggunakan alamat $301, alamat register *Port* C menggunakan alamat $302 dan untuk alamat *register control* menggunakan alamat $303.

Hasil pembacaan pada PPI yang kedua kemudian dibandingkan hasilnya dengan *field* HASIL_UJI 1 atau 2 yang telah dibuat pada *database*. Bila hasil pengujian berbeda dengan hasil uji yang terdapat pada *field* hasil uji, maka program akan menampilkan pesan bahwa IC yang diuji telah rusak. Sementara bila hasil pengujian menunjukan bahwa keluaran PPI yang kedua sesuai dengan PPI yang pertama, maka IC yang di uji masih dalam kondisi baik. Setiap pengujian harus dilakukan sebanyak dua kali untuk mengganti logika pengujian pada *input* masing-masing IC. Misalnya pada pengujian pertama *input* yang sama mendapatkan giliran pengujian dengan kondisi logika yang berbeda yaitu '0' dan '1'.

Untuk membandingkan hasil uji antara keluaran pada IC yang diuji dengan *database* menggunakan persamaan *XOR*. Pada tabel kebenaran XOR bila dua *input* mempunyai logika yang sama, maka keluarannya adalah '1', sedangkan bila kedua *input* memiliki logika yang berbeda maka keluarannya adalah '0'.

Dengan tabel kebenaran XOR inilah diperoleh hasil pengujian untuk menentukan kondisi IC.