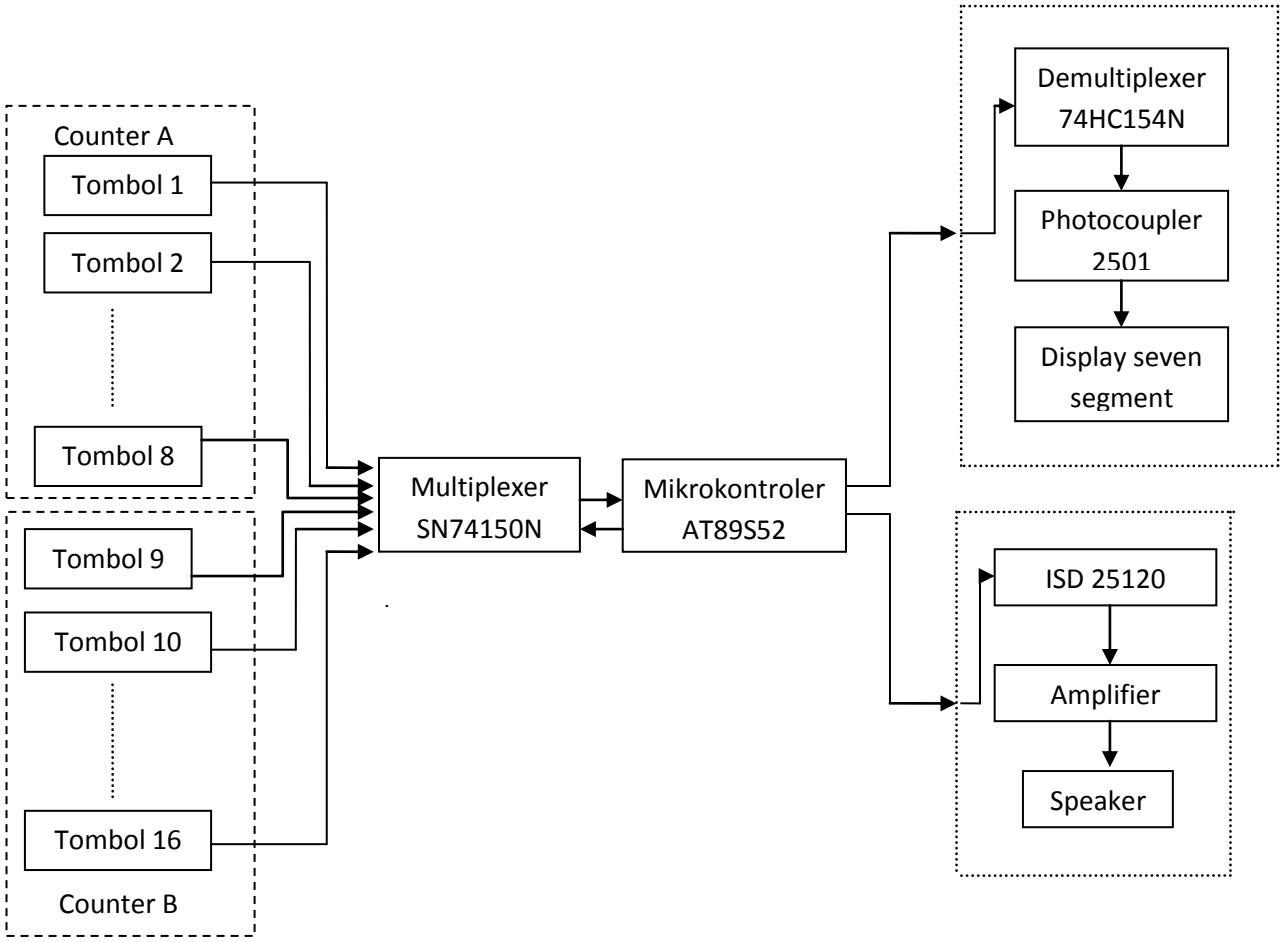
BAB III
PERANCANGAN SISTEM

Pada bab perancangan sistem ini akan membahas tentang prinsip kerja sistem, perancangan perangkat keras, algoritma program mikrokontroler.

3.1 Prinsip Kerja Sistem

Dalam sistem ini, para nasabah mengambil kartu yang berisi nomor urut, kemudian menunggu sambil duduk santai di tempat yang telah tersedia sampai terdengar suara panggilan nomor urut. Selain itu, di *teller* terdapat display yang menunjukkan nomor urut yang akan dilayani. Keadaan ini terus berlangsung sampai nomor antrian terakhir.

Teller akan memberikan input dengan cara menekan tombol, maka input tersebut akan diseleksi oleh IC Multiplexer SN74150N, kemudian input hasil seleksi dari multiplexer akan digunakan untuk menentukan *teller* yang telah menekan tombol. Hasil dari serangkaian proses tersebut adalah output suara dan display pada seven segment. Untuk menghasilkan output suara, digunakan IC ISD25120 untuk merekam dan memainkan suara. Untuk menghasilkan output suara yang tepat, mikrokontroler AT89S52 akan mengirimkan output data yang berupa alamat memory untuk memanggil suara-suara yang telah direkam pada IC ISD25120. Tampilan display sevent segment yang didapat merupakan hasil dari input yang diberikan dari mikrokontroler ke IC Demultiplexer 74HC154N. Output hasil seleksi dari demultiplexer akan menjadi input bagi photocoupler 2501. Photocoupler 2501 pada rangkaian ini digunakan sebagai penyesuai tegangan untuk mengkontrol transistor PNP C9012 yang akan memberi tegangan 12V yang dibutuhkan oleh seven segment. Adapun sistem yang dirancang dapat dilihat pada gambar 3.1.



Gambar 3.1. Perancangan Sistem *Teller* Berbasis Mikrokontroler

3.2 Perancangan Perangkat Keras

3.2.1. Mikrokontroler AT89S52

Jenis mikrokontroler yang digunakan adalah AT89C52 keluarga dari MCS-51 keluaran Atmel. Pada perancangan alat *teller* berbasis mikrokontroler, mikrokontroler digunakan untuk mengimplementasikan pengaturan data yang dikirim untuk ditampilkan pada *seven segment* dan pengaturan pengiriman data untuk memanggil alamat-alamat pada IC ISD25120 agar dapat menghasilkan output yang sesuai dengan input yang diberikan. Berikut ini pin-pin pada mikrokontroler AT89S52 yang digunakan:

a. Port P0

Port P0.0 sampai dengan P0.7 (pin 39 sampai dengan pin 32) dan digunakan untuk mengirimkan input ke IC ISD25120 yang hasilnya akan mengalokasikan suara yang akan direkam ke alamat memori yang dikehendaki.

b. Port P1

Port P1.0 sampai dengan P1.3 (pin 1 sampai dengan pin 4) digunakan untuk mengirimkan output ke demultiplexer IC 74HC154N yang hasilnya memberikan output ABCD sebagai selektor.

c. Port P2

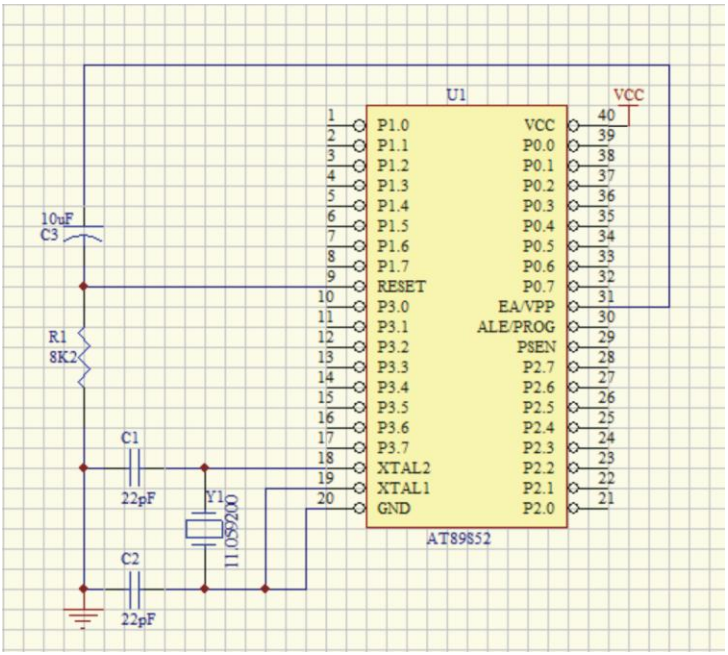
Port P2.0 sampai dengan P2.7 (pin 21 sampai dengan pin 28) digunakan untuk menghubungkan ke display seven segment.

d. Port P3

Port P3.0 sampai dengan P3.3 (pin 10 sampai dengan pin 13) digunakan untuk mengirimkan output ke multiplexer IC SN74150N yang hasilnya memberikan output ABCD sebagai selektor.

Port P3.5 sampai dengan P3.7 (pin 15 sampai dengan pin 17) digunakan untuk mengirimkan input ke pin CE, PD, EOM pada IC ISD25120 untuk proses perekaman.

Pin-pin di atas digunakan sebagai pin utama dalam perancangan, selain itu masih terdapat beberapa pin lainnya yang digunakan untuk keperluan khusus, misalnya pin untuk osilator, EA/VP, VCC, GND dan untuk *reset* seperti terlihat pada gambar 3.2.



Gambar 3.2. Rangkaian Minimum AT89S52

Pin EA/VP dihubungkan dengan VCC berfungsi untuk menjalankan program yang ada di memori internal dari AT89S52. Keterangan mengenai bagian osilator adalah sebagai berikut :

Osilator merupakan subsistem dari mikrokontroler yang berfungsi untuk membangkitkan *clock* pada mikrokontroler. *Clock* diperlukan oleh mikrokontroler untuk mensinkronkan proses yang sedang berlangsung dalam mikrokontroler. Rangkaian ini tersusun atas komponen kristal dan dua buah kapasitor. Frekuensi kristal yang biasa digunakan antara 6 sampai 24 MHz. Pada pembuatan alat ini digunakan kristal 11.0592 Mhz agar mempermudah perhitungan *baudrate* komunikasi serial. Adapun kapasitor yang dapat dipergunakan berdasarkan *data sheet* antara 20pF sampai 40pF untuk osilator yang menggunakan kristal. Kapasitor yang digunakan adalah jenis *monocap* bernilai 33 pF. Kapasitor jenis ini dipilih karena cocok untuk aplikasi frekuensi tinggi. Fungsi kapasitor adalah untuk menstabilkan osilasi yang dihasilkan oleh kristal. Penempatan

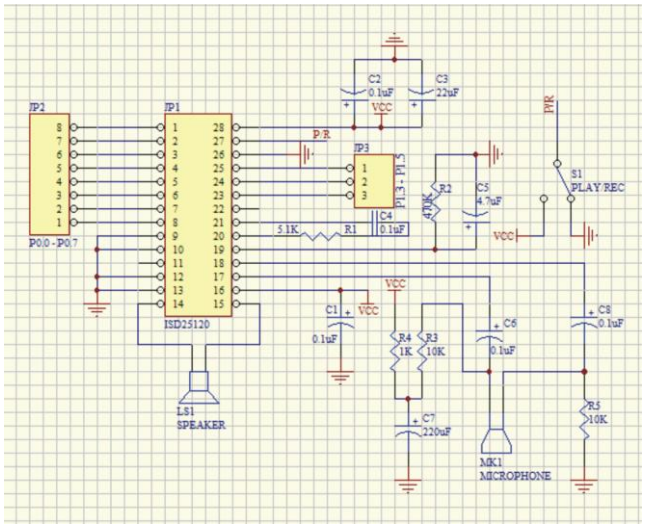
kristal dan kapasitor diusahakan sedekat mungkin dengan bagian mikrokontroler untuk menghindari kemungkinan terjadinya *noise*.

3.2.2. ISD 25120

Rangkaian penghasil suara ini menggunakan IC tipe ISD 25120 yang mempunyai kapasitas penyimpanan suara maksimum selama 120 detik, dimana A0 sampai A7 digunakan untuk mengisi dan memanggil alamat suara yang disimpannya.

Pada saat mengisi suara, pin P/R diberi *logic low* dan pin A0 – A7 untuk mengalokasikan suara yang akan direkam ke alamat memori yang dikehendaki. Kemudian pin CE diberi *logic low*, untuk *start* atau memulai perekaman suara melalui mikrophone jenis kondensor. Setelah selesai pengisian suara, pin PD diberi *logic high* untuk stop.

Untuk memanggil suara, pin P/R harus diberi *logic high*, kemudian alamat memori diberikan melalui pin A0 sampai A7. Selanjutnya pin CE diberi *logic low*, maka suara yang sudah direkam sebelumnya akan keluar melalui speaker. Setelah selesai pemutaran suara, pin PD diberi *logic high* untuk stop. Rangkaian suara ISD 25120 dapat dilihat pada gambar 3.3.



Gambar 3.3. Rangkaian Suara ISD 25120

A. Pengalamatan Suara ISD 25120

Untuk mengetahui suara yang sudah direkam ada dialamat mana, dibuatlah alat pengalamatan suara. Dimana dialat tersebut dapat diatur alamat mana yang akan diisi suara yang dikehendaki. Supaya tidak terjadi penumpukan suara di suatu alamat tertentu, maka alamat suara dikasih jarak 13H antara suara satu ke suara lainnya. Setelah pengisian suara dialamat tertentu telah selesai, maka memudahkan mikrokontroler untuk memanggil suara yang di kehendaki.

Rangkaian Pengalamatan suara ini terdiri 3 bagian yaitu:

1. Rangkaian Mikrokontroler dengan menggunakan IC AT89S52 yang akan mengatur alamat yang akan di setting ke ISD Suara.

2. Rangkaian display seven segment dan tombol tekan yang terdiri dari 3 buah display seven segment sebagai penunjuk alamat yang akan di alamati ke ISD suara dan 4 buah tombol tekan terdiri dari tombol *play/record*, tombol *stop*, tombol alamat naik dan tombol alamat turun.

3. Rangkaian ISD Suara yang terdiri dari port pengalamatan, *Mic input*, dan tombol *on/off* untuk mode *Record* atau *Play*.

Cara kerja alat pengalamatan suara pada ISD secara keseluruhan pengendali utama dilakukan oleh mikrokontroller AT89S52. IC Mikrokonroler ini mengecek (*scanning*) tiap tombol satu persatu. Apabila ada tombol yang di tekan, Mikrokontroler akan melakukan tugasnya sesuai yang telah di program. Cara kerja alat pengalamatan suara pada ISD ini ada dua mode yaitu :

1. Mode Playback

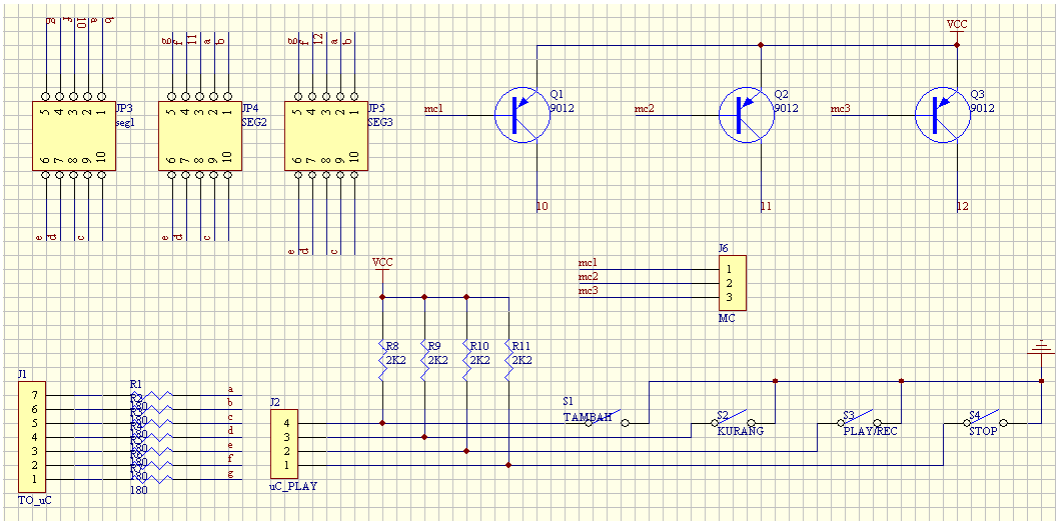
   Tombol ISD dilepas (*off*) akan di set ke mode *PlayBack*. Kemudian setting alamat dengan menekan tombol alamat *Up/Down* untuk data alamat yang akan dikirim ke port alamat ISD Suara. Apabila Tombol *Play/Record* ditekan, Mikrokontrol P3.5 (CE) akan di set logic *high* dan P3.6 (PD) akan diset logic *low* dan kedua kontrol logic CE dan PD diset ke ISD kendali maka ISD akan mengeluarkan suara. Apabila tombol stop ditekan, Maka

kontrol logic CE pada ISD akan di set logic *high* dan ISD akan berhenti memutar suara (*play*).

2. Mode Record

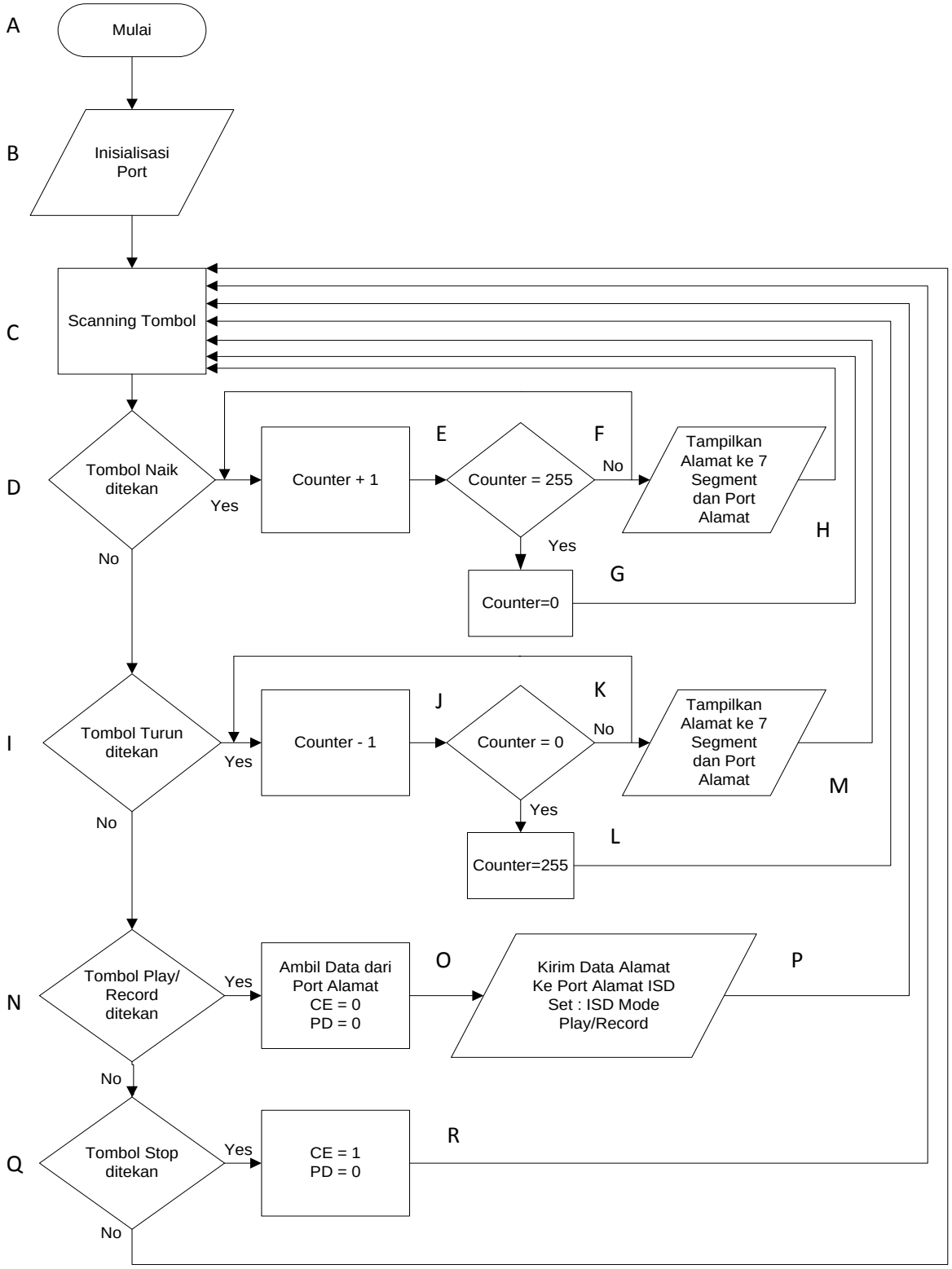
Tombol ISD ditekan (*on*) akan di set ke mode *Record*. Apabila akan melakukan rekam suara, Tombol alamat *Up/Dowm* di setting dari alamat desimal 0 – 255 dan data alamat ini akan dikirim ke ISD port alamat. Kemudian tombol *Play/Record* ditekan terus selama panjang suara yang akan di rekam.

Untuk rangkaian pengalamatan suara pada ISD25120 dapat dilihat pada gambar 3.4.



Gambar 3.4 Rangkaian Pengalamatan Suara pada ISD25120

Untuk flowchart dapat dilihat pada gambar 3.5



Gambar 3.5 Flowchart Pengalamatan Suara pada ISD25120

Penjelasan *flowchart* pada gambar 3.5 sebagai berikut :

A. Proses program mulai.
B. Menginisialisasikan port mana saja yang dipakai.
C. *Scanning* pada semua tombol.
D. Apakah tombol naik ditekan :
   Jika ya, proses dilanjutkan
   Jika tidak, *scanning* pada tombol lain.
E. Counter di incrementkan sebanyak satu.
F. Apakah counter sudah sampai dengan 255 :
   Jika ya, counter di reset kembali ke 0.
   Jika tidak, proses di lanjutkan.
G. Counter direset kembali ke 0 dan kembali ke *scanning* tombol.
H. Menampilkan alamat suara pada seven segment dan pada port alamat dan kembali ke *scanning* tombol.
I. Apakah tombol turun ditekan :
   Jika ya, proses dilanjutkan
   Jika tidak, *scanning* pada tombol lain.
J. Counter dikurangi sebanyak satu.
K. Apakah counter sudah sampai 0 :
   Jika ya, counter diisi dengan 255
   Jika tidak, proses dilanjutkan.
L. Counter diisi dengan 255 dan kembali ke *scanning* tombol.
M. Menampilkan alamat suara pada seven segment dan pada port alamat dan kembali ke *scanning* tombol.
N. Apakah tombol *play/record* ditekan :
   Jika ya, proses dilanjutkan.
   Jika tidak, *scanning* pada tombol lain.
O. Ambil data dari port alamat, pin CE diberi logika 0 dan pin PD diberi logika 0.
P. Kirim data alamat ke port alamat ISD dan setting ke mode *play/record* dan kembali ke *scanning* tombol.

Q. Apakah tombol *stop* ditekan :

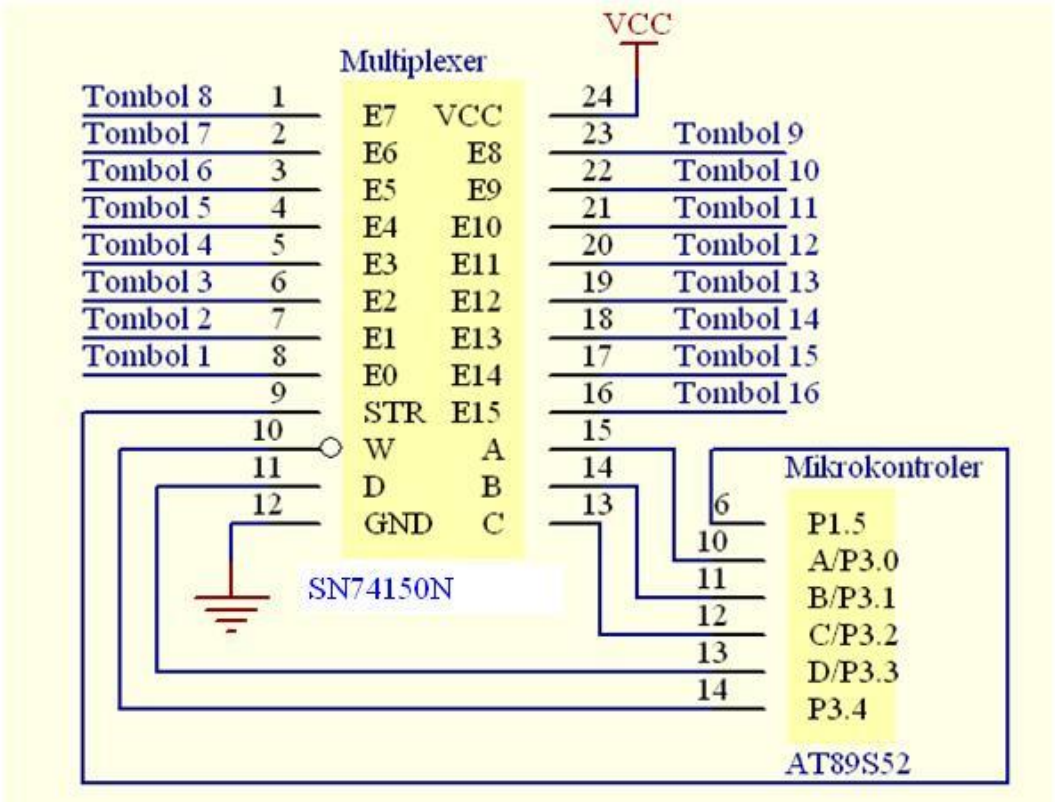
Jika ya, proses dilanjutkan.

Jika tidak, kembali ke *scanning* tombol.

R. Pin CE diberi logika 1 dan pin PD diberi logika0 dan kembali ke *scanning* tombol.

3.2.3. Multiplexer SN74150N

Pada rangkaian *Multiplexer* SN74150N ini input ABCD digunakan sebagai selektor yang dihubungkan ke P3.0, P3.1, P3.2 dan P3.3 pada mikrokontroler. Dengan selektor ini, jika input ABCD = 0000 maka E_0 terpilih dan output W akan berada dalam kondisi status *active low*. Dengan cara yang sama, apabila diberikan selektor input ABCD = 1000 maka input E_1 terpilih dan output W akan berada dalam kondisi status *active low* dan seterusnya. Untuk rangkaian multiplexer dapat dilihat pada gambar 3.6.

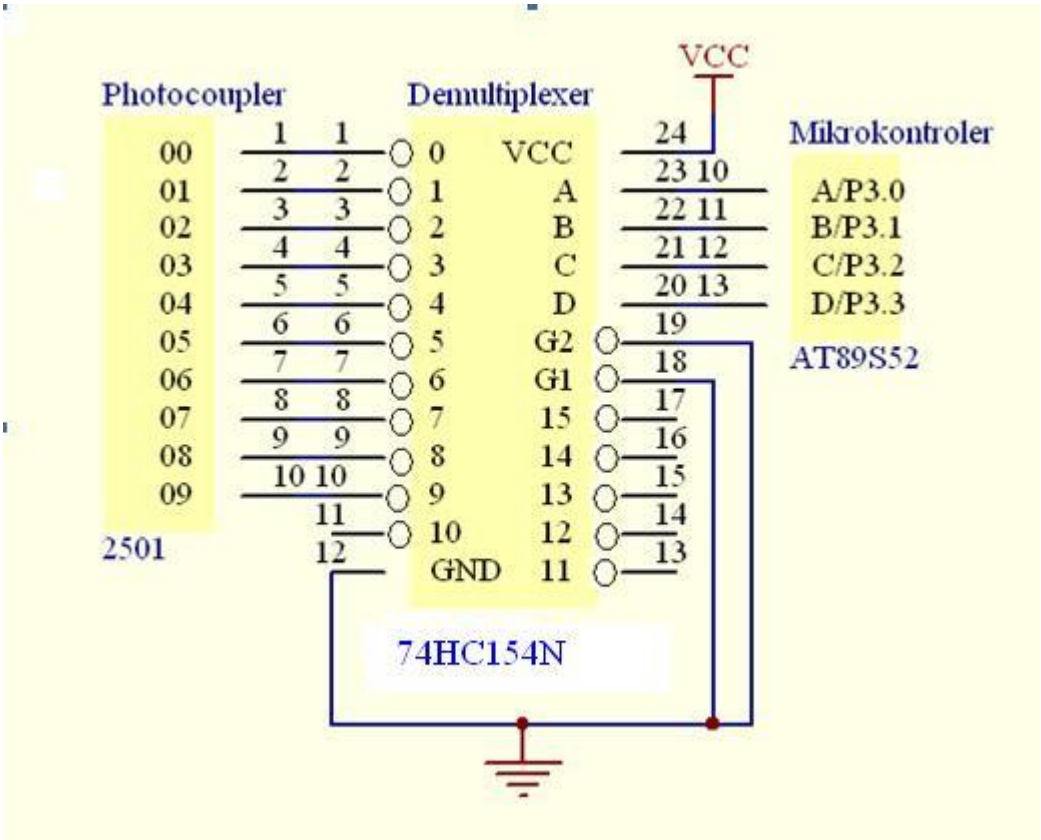


Gambar 3.6. Rangkaian Multiplexer SN74150N

16 buah input tombol akan terhubung ke multiplexer IC SN74150N, ke 16 input ini akan diseleksi oleh multiplexer dan menghasilkan output data biner sebanyak 4 bit.

3.2.4. Demultiplexer 74HC154N

Pada rangkaian *Demultiplexer* 74HC154N ini input ABCD digunakan sebagai selektor yang dihubungkan ke ke P1.0, P1.1, P1.2 dan P1.3 pada mikrokontroler. Dengan selektor ini, jika input ABCD mendapat 0000 maka output O_0 terpilih dan semua output lainnya akan berada dalam kondisi status *active high.* Dengan cara yang sama, apabila diberikan selektor input ABCD = 1000 maka output O_1 akan terpilih dan seterusnya. Untuk rangkaian demultiplexer dapat dilihat pada gambar 3.7.



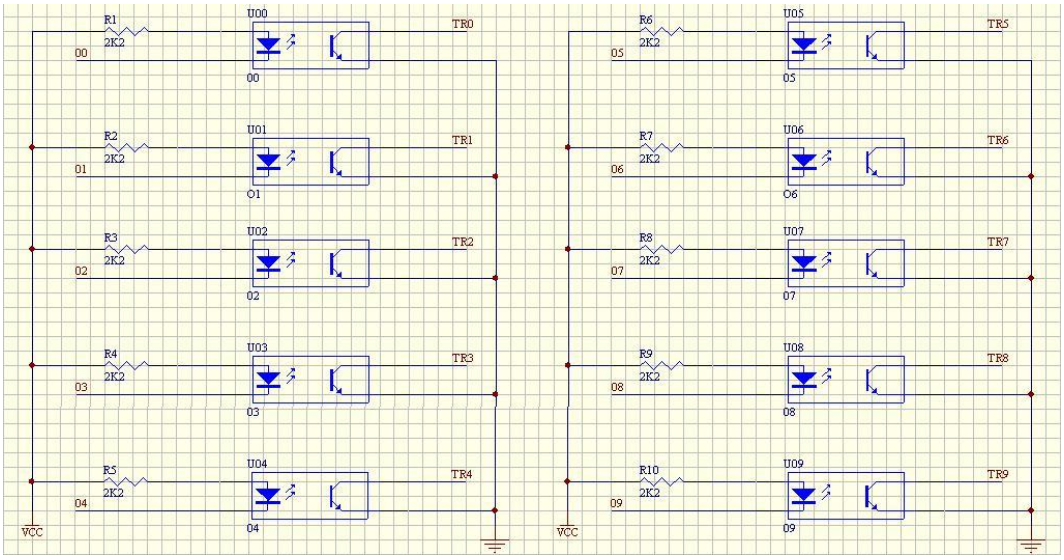
Gambar 3.7. Rangkaian Demultiplexer 74HC154N

4 buah input mikrokontroler akan terhubung ke demultiplexer 74HC154N, ke 4 input ini akan diseleksi oleh demultiplexer dan menghasilkan 10 output untuk dikirimkan ke Photocoupler 2501.

3.2.5. Photocoupler 2501

Ditinjau dari kegunaan fisik photocoupler dapat berbentuk bermacam-macam. Bila hanya digunakan untuk mengisolasi level tegangan atau data pada sisi transmitter dan sisi receiver, maka photocoupler ini biasanya dibuat dalam bentuk solid (tidak ada ruang antara LED dan Photodiode). Sehingga sinyal listrik yang ada pada input dan output akan terisolasi. Dengan kata lain Photocoupler ini digunakan sebagai optosilator jenis IC.

Untuk pemakaian rangkaian photocoupler jenis IC ini disini difungsikan sebagai penyesuaian tegangan untuk mengkontrol Transistor tipe PNP (C9012) yang akan mengsupply tegangan 12 volt yang akan dibutuhkan oleh display seven segment. Untuk rangkaian photocoupler dapat dilihat pada gambar 3.8.



Gambar 3.8. Rangkaian Photocoupler 2501

Ket : 00 sampai 09 : dari Demultiplexer 74HC154N

TR0 samai TR9 : ke Transistor PNP C9012

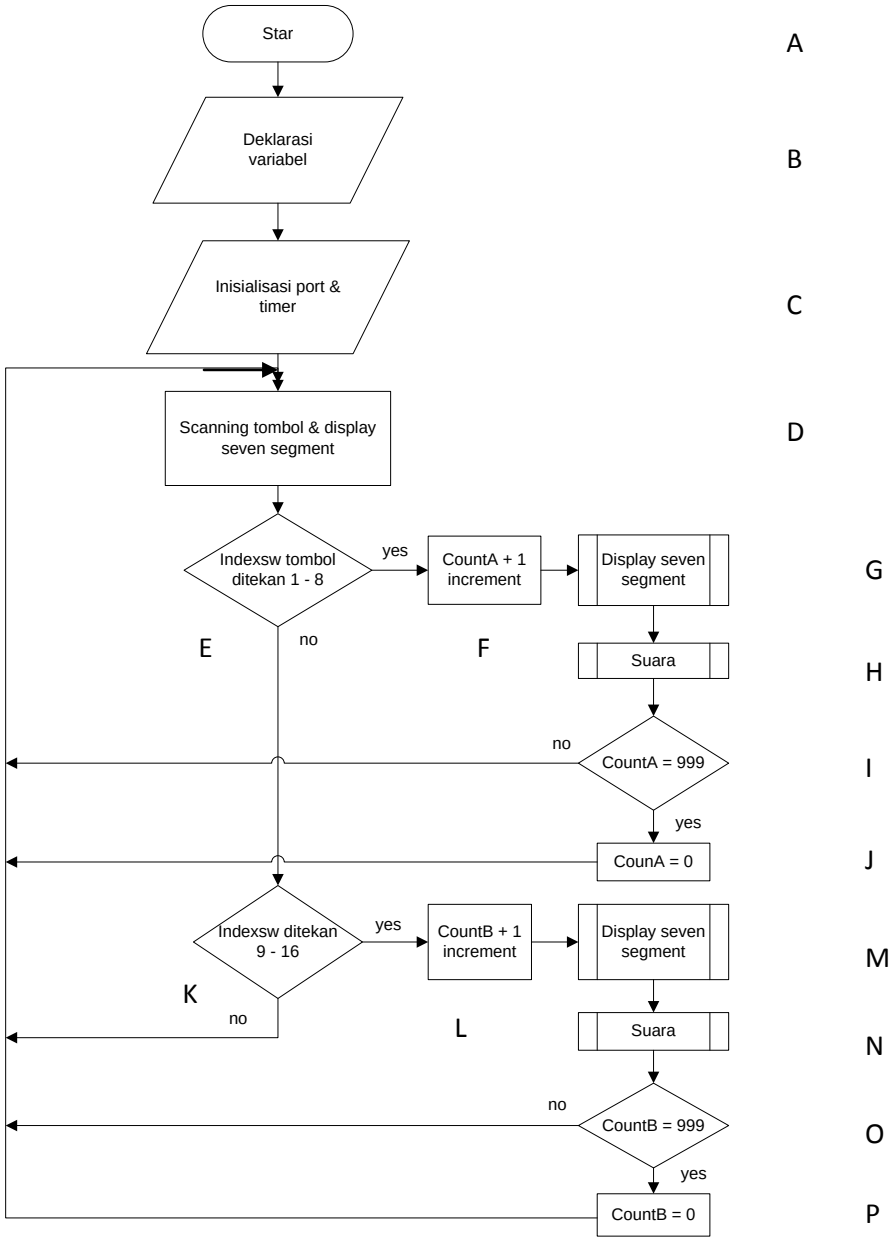
3.3 Perangkat Lunak

Perancangan perangkat lunak dilakukan bersamaan dengan pembuatan bagian perangkat keras (*hard ware*), sehingga pada saat pembuatan listing program selesai dapat langsung dicoba pada perangkat keras (*hard ware*).

Untuk mempermudah pembuatan listing program, dibuatlah diagram alir (*flowchart*) dari sistem yang akan dibangun, karena dengan *flowchart* dapat

ditelusuri program satu persatu seandainya program tidak berjalan sesuai keinginan.

Diagram alir dari software sistem ini ditunjukan pada gambar 3.9.



Gambar 3.9. *Flowchart* Perangkat Lunak

Penjelasan *flowchart* pada gambar 3.9 sebagai berikut :

A. Proses program mulai.
B. Mendeklarasikan variabel-variabel yang akan dipakai pada program.
C. Port 3.0 – 3.3 digunakan sebagai input dari multiplexer.

Port 3.4 digunakan sebagai signal masukan ke Mikrokontroler untuk deteksi penekanan tomb

Inisialisasi timer menggunakan mode 1.

D. Pada saat alat mulai dinyalakan, display sevent segment semuanya menyala lalu mikrokontroler AT89S52 akan menunggu input tombol yang ditekan oleh *user*.
E. Apakah tombol 1 sampai dengan 8 ditekan.

   Jika ya, maka counter A.

   Jika tidak, kembali penyeleksian tombol.
F. Incrementkan sebanyak satu dan hasilnya disimpan di variabel_num.
G. Menyalakan display seven segment.
H. Menyalakan Suara.
I. Apakah countA sudah sampai nilai 999,

   Jika ya, maka proses di lanjutkan.

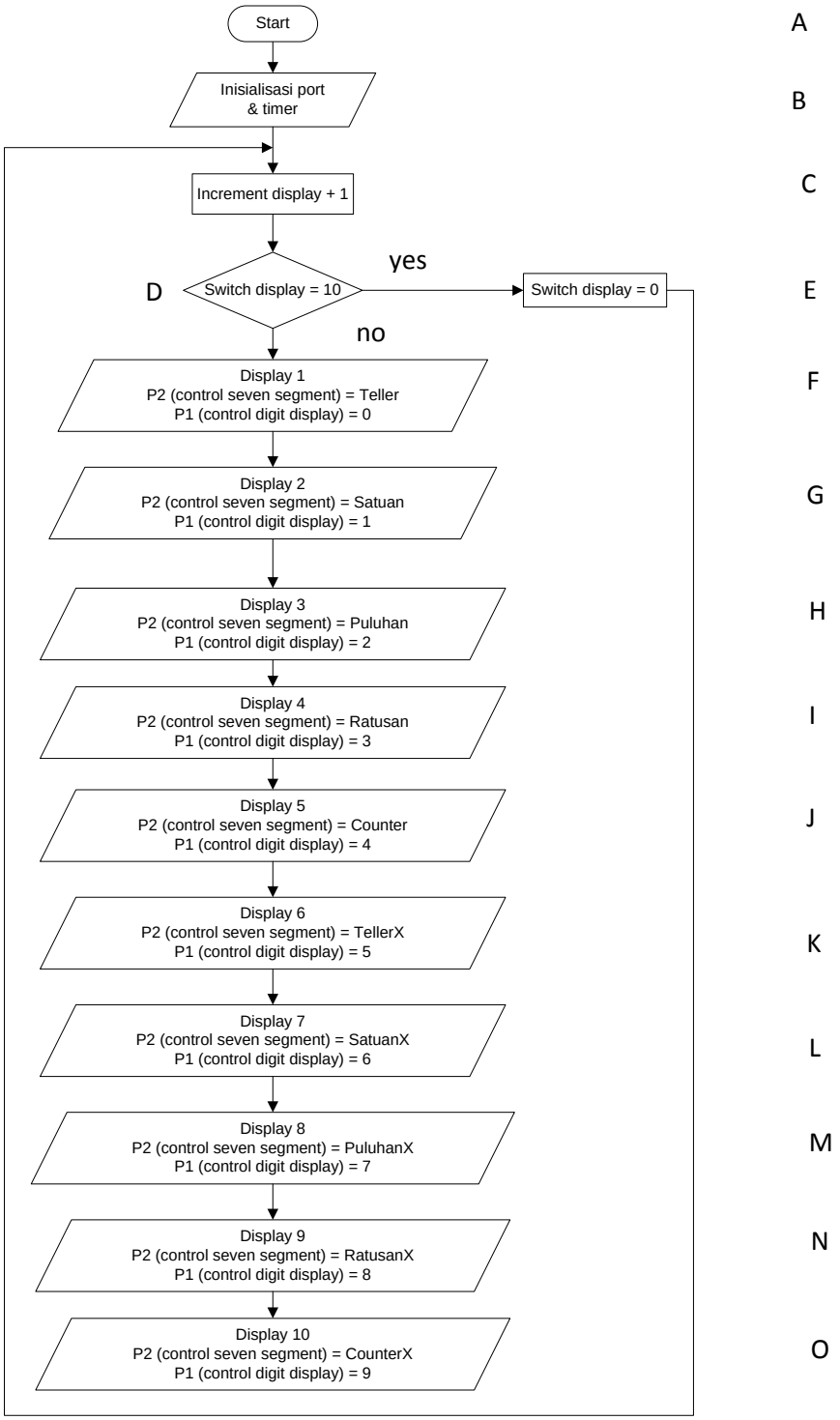
   Jika tidak, kembali scanning tombol.
J. Nilai countA direset kembali ke 0 dan kembali ke scanning tombol.
K. Apakah tombol 9 sampai dengan 16 ditekan.

   Jika ya, maka countB.

   Jika tidak, kembali scanning tombol.
L. Increment sebanyak satu dan hasilnya disimpan di variabel_num.
M. Menyalakan display seven segment.
N. Menyalakan Suara.
O. Apakah countB sudah sampai nilai 999.

   Jika ya, maka proses dilanjutkan.

   Jika tidak, kembali ke scanning tombol.
P. Nilai countB direset kembali ke 0 dan kembali ke scanning tombol.



Gambar 3.10 *Flowchart* Pengaktif Display

Penjelasan *flowchart* pada gambar 3.10 sebagai berikut :

A. Proses menyalakan display seven segment.

B. Inisialisasi timer menggunakan mode 1.

C. Display diincrementkan sebanyak 1.

D. Apakah display sudah sampai 10.
   Jika tidak, proses dilanjutkan.
   Jika ya, display direset menjadi 0.

E. Selector display direset menjadi 0 dan kembali keincrement display.

F. Display 1 : jika P1 bernilai 0, maka P2 akan mengaktifkan display seven segment *Teller* baris pertama.

G. Display 2 : jika P1 bernilai 1, maka P2 akan mengaktifkan display seven segment Satuan baris pertama.

H. Display 3 : jika P1 bernilai 2, maka P2 akan mengaktifkan display seven segment Puluhan baris pertama.

I. Display 4 : jika P1 bernilai 3, maka P2 akan mengaktifkan display seven segment Ratusan baris pertama.

J. Display 5 : jika P1 bernilai 4, maka P2 akan mengaktifkan display seven segment Counter baris pertama.

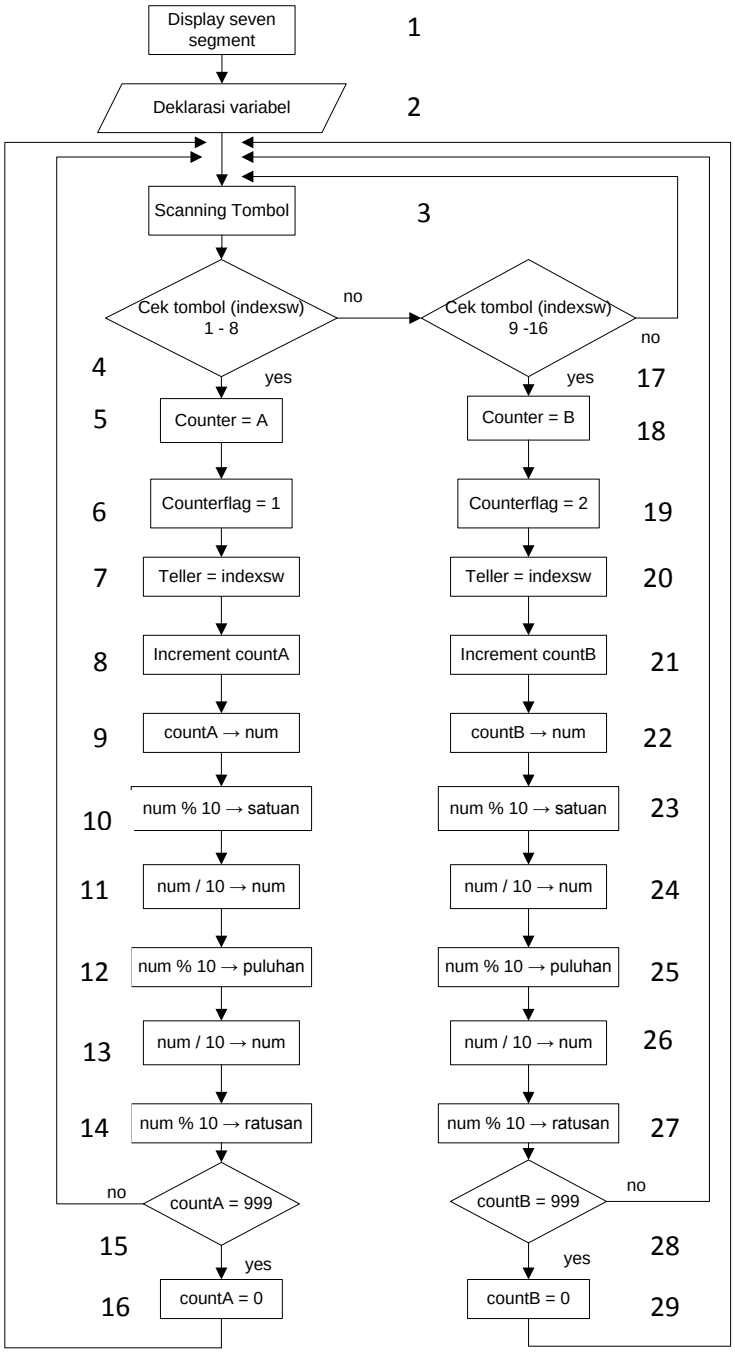
K. Display 6 : jika P1 bernilai 5, maka P2 akan mengaktifkan display seven segment *Teller* baris kedua.

L. Display 7 : jika P1 bernilai 6, maka P2 akan mengaktifkan display seven segment Satuan baris kedua.

M. Display 8 : jika P1 bernilai 7, maka P2 akan mengaktifkan display seven segment Puluhan baris kedua.

N. Display 9 : jika P1 bernilai 8, maka P2 akan mengaktifkan display seven segment Ratusan baris kedua.

O. Display 10 : jika P1 bernilai 9, maka P2 akan mengaktifkan display seven segment Counter baris kedua.



Gambar 3.11 *Flowchart* Counter A dan Counter B

Penjelasan *flowchart* pada gambar 3.11 sebagai berikut :

1. Display seven segment menyala.
2. Mendeklarasikan variabel-variabel yang akan dipakai pada program.
3. Scan pada tombol.
4. Apakah tombol 1 sampai 8 ditekan :

Jika ya, maka counter menunjukkan counter A.

Jika tidak, penyeleksian tombol 9 sampai 16.

5. Counter menunjukkan counter A.
6. Karena yang terpilih tombol 1 sampai 8, maka counterflag bernilai 1.
7. *Teller* diisi dengan indexsw, indexsw didapat dari penekanan tombol.
8. CountA diincremenkan dengan 1.
9. Nilai countA disalin ke variabel_num.
10. Variabel_num di modulus 10, hasilnya disimpan di variabel_satuan.
11. Variabel_num di bagi 10, hasilnya disimpan kembali ke variabel_num.
12. Variabel_num di modulus 10, hasilnya disimpan di variabel_puluhan.
13. Variabel_num di bagi 10, hasilnya disimpan kembali ke variabel_num.
14. Variabel_num di modulus 10, hasilnya disimpan di variabel_ratusan.
15. Apakah nilai countA sudah mencapai 999 :

    Jika ya, proses dilanjutkan.

    Jika tidak, kembali ke scanning tombol.
16. CountA direset menjadi 0 dan kembali ke scanning tombol.
17. Apakah tombol 9 sampai 16 ditekan :

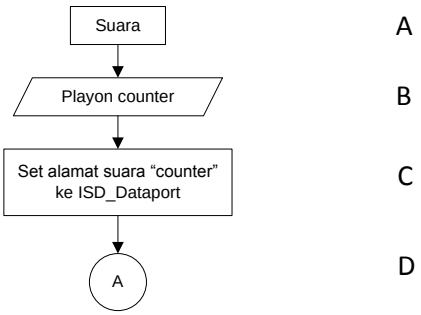
    Jika ya, maka counter menunjukkan counter B.

    Jika tidak, kembali ke scanning tombol.
18. Counter menunjukkan counter B.
19. Karena yang terpilih tombol 9 sampai 16, maka counterflag bernilai 2.
20. *Teller* diisi dengan indexsw, indexsw didapat dari penekanan tombol.
21. CountB diincremenkan dengan 1.
22. Nilai countB disalin ke variabel_num.
23. Variabel_num di modulus 10, hasilnya disimpan di variabel_satuan.
24. Variabel_num di bagi 10, hasilnya disimpan kembali ke variabel_num.
25. Variabel_num di modulus 10, hasilnya disimpan di variabel_puluhan.
26. Variabel_num di bagi 10, hasilnya disimpan kembali ke variabel_num.
27. Variabel_num di modulus 10, hasilnya disimpan di variabel_ratusan.
28. Apakah nilai countB sudah mencapai 999 :

    Jika ya, proses dilanjutkan.

    Jika tidak, kembali ke scanning tombol.

29. countB direset menjadi 0 dan kembali ke scanning tombol.



Gambar 3.12 *Flowchart* Suara Counter

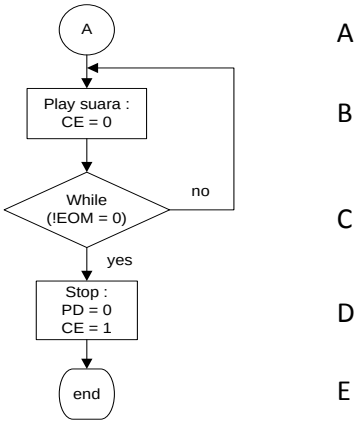
Penjelasan *flowchart* pada gambar 3.12 sebagai berikut :

A. Menyalakan suara.

B. Putar suara counter dengan memanggil alamatnya pada ISD.

C. Mengatur alamat suara "counter" ke alamat data ISD.

D. Lanjutkan ke A.



Gambar 3.13 *Flowchart* Playon

Penjelasan *flowchart* pada gambar 3.13 sebagai berikut :

A. Lanjutan A.

B. Putar suara, CE diberi nilai 0 agar ISD menyala.

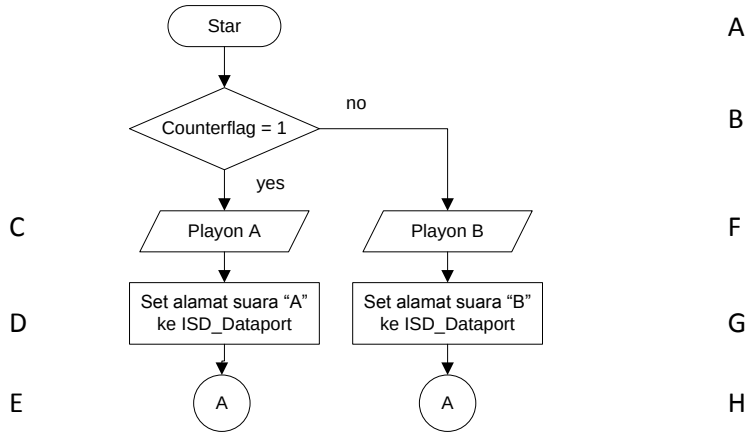
C. Apakah EOM bernilai 0.

   Jika ya, proses dilanjutkan.

   Jika tidak, kembali ke Play Suara..

D. Menghentikan suara, PD diberi nilai 0 dan CE diberikan nilai 1 agar proses pemutaran suara berhenti.

E. Proses berakhir.



Gambar 3.14 *Flowchart* Suara A dan B

Penjelasan *flowchart* pada gambar 3.14 sebagai berikut :

A. Mulai proses pemutaran suara.

B. Apakah counterflag bernilai 1.

   Jika ya, counter A.

   Jika tidak, counter B.

C. Mainkan suara counter A.

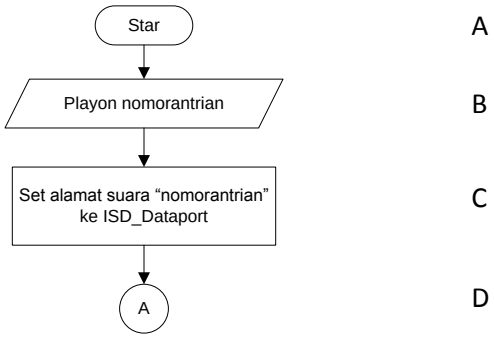
D. Mengatur alamat suara "A" ke alamat data ISD.

E. Lanjutkan ke A.

F. Mainkan suara counter B.

G. Mengatur alamat suara “B” ke alamat data ISD.

H. Lanjutkan ke A.



Gambar 3.15 *Flowchart* Suara Nomor Antrian

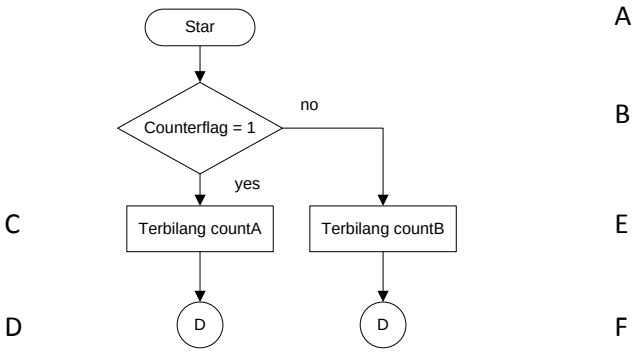
Penjelasan *flowchart* pada gambar 3.15 sebagai berikut :

A. Mulai proses pemutara suara.

B. Putar suara nomorantrian dengan memanggil alamatnya pada ISD.

C. Mengatur alamat suara “nomorantrian” ke alamat data ISD.

D. Lanjutkan ke A.



Gambar 3.16 *Flowchart* Terbilang Nomor Antrian

Penjelasan *flowchart* pada gambar 3.16 sebagai berikut :

A. Mulai proses pemutaran suara.

B. Apakah counterflag bernilai 1.

   Jika ya, counter A.

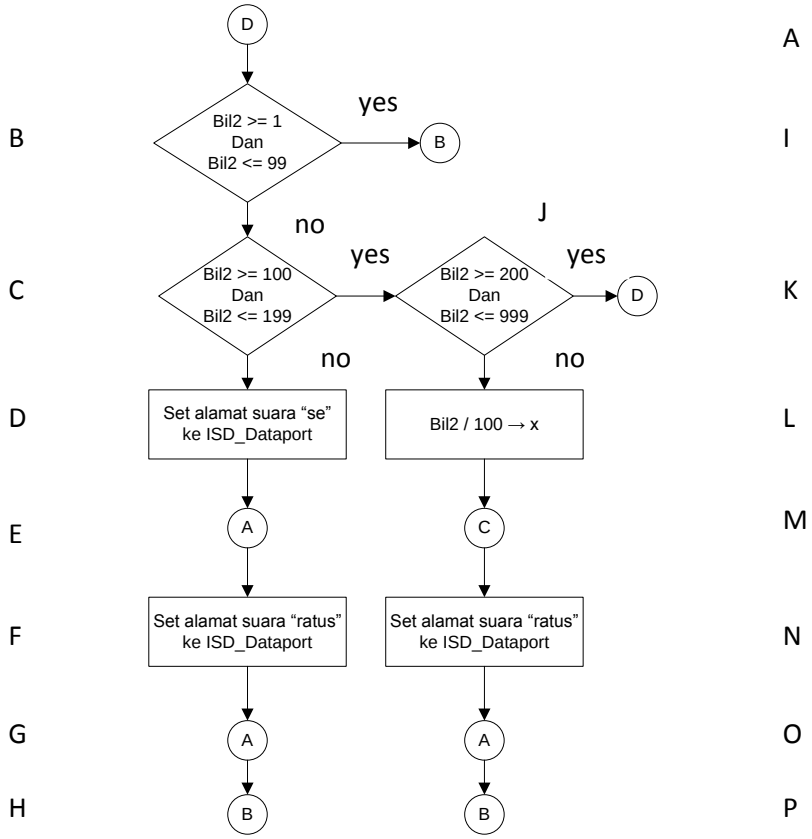
   Jika tidak, counter B.

C. Putar suara bilangan untuk Counter A.

D. Lanjutkan ke D.

E. Putar suara bilangan untuk Counter B.

F. Lanjutkan ke D.



Gambar 3.17 *Flowchart* Play Ratusan

Penjelasan *flowchart* pada gambar 3.17 sebagai berikut :

A. Lanjutan D.

B. Apakah bil2 lebih besar sama dengan 1 dan bil2 lebih kecil sama dengan 99.

   Jika ya, lanjutkan ke B.

   Jika tidak, proses dilanjutkan.

C. Apakah bil2 lebih besar sama dengan 100 dan bil2 lebih kecil sama dengan 199.

   Jika ya, pemilihan akan dilakukan lagi dengan menyeleksi bil2.

   Jika tidak, proses dilanjutkan.

D. Mengatur alamat suara “se” ke alamat data ISD.

E. Lanjutkan ke A.

F. Mengatur alamat suara “ratus” ke alamat data ISD.

G. Lanjutkan ke A.

H. Lanjutkan ke B.

I. Lanjutkan ke B.

J. Apakah bil2 lebih besar sama dengan 200 dan bil2 lebih kecil sama dengan 999.

   Jika ya, lanjutkan ke D.

   Jika tidak, proses dilanjutkan.

K. Lanjutkan ke D.

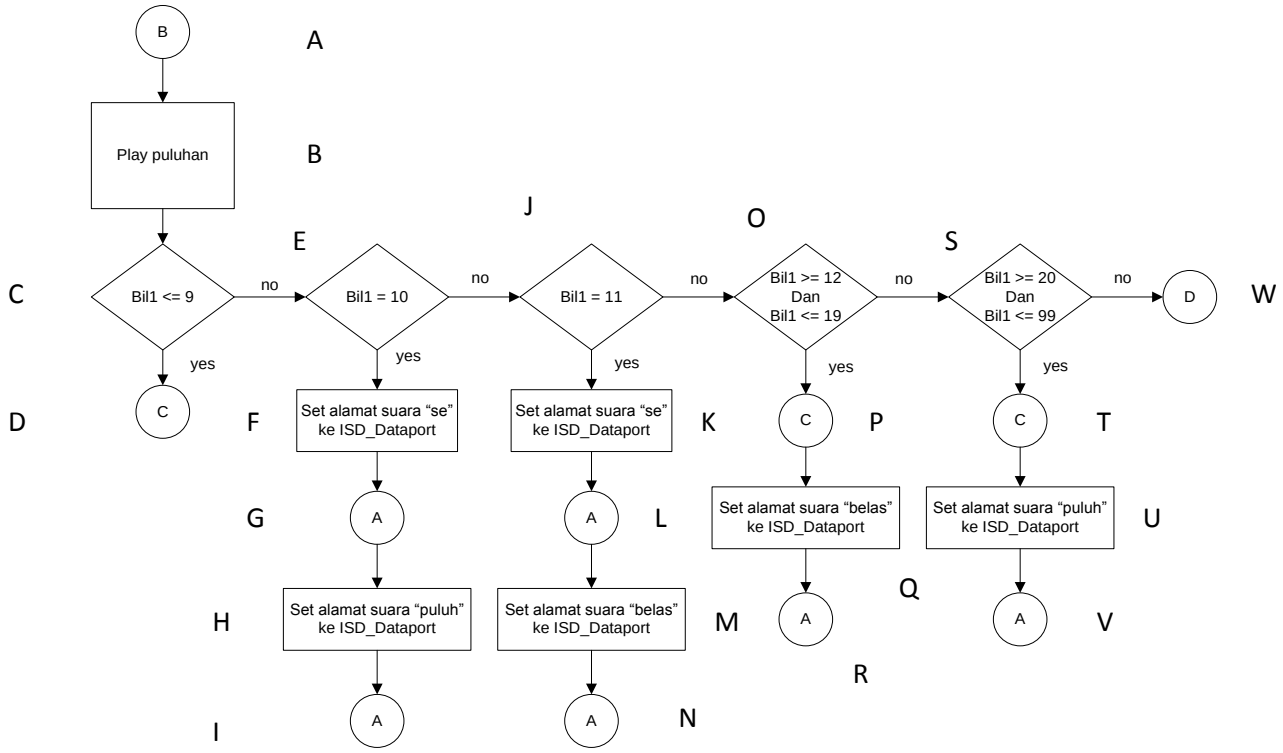
L. Bil2 dibagi 100, hasilnya disimpan pada variabel_x.

M. Lanjutkan ke C.

N. Mengatur alamat suara “ratus” ke alamat data ISD.

O. Lanjutkan ke A.

P. Lanjutkan ke B.



Gambar 3.18 *Flowchart* Play Puluhan

Penjelasan *flowchart* pada gambar 3.18 sebagai berikut :

A. Lanjutan B.

B. Putar suara puluhan dengan memanggil alamatnya pada ISD.

C. Apakah bil1 lebih kecil sama dengan 9.

   Jika ya, lanjutkan ke C.

   Jika tidak, pemilihan akan dilakukan lagi dengan menyeleksi bil1.

D. Lanjutkan ke C.

E. Apakah bil1 sama dengan 10.

   Jika ya, proses dilanjutkan.

   Jika tidak, pemilihan akan dilakukan lagi dengan menyeleksi bil1.

F. Mengatur alamat suara “se” ke alamat data ISD.

G. Lanjutkan ke A.

H. Mengatur alamat suara “puluh” ke alamat data ISD.

I. Lanjutkan ke A.

J. Apakah bil1 sama dengan 11.

   Jika ya, proses dilanjutkan.

   Jika tidak, pemilihan akan dilakukan lagi dengan menyeleksi bil1.

K. Mengatur alamat suara “se” ke alamat data ISD.

L. Lanjutkan ke A.

M. Mengatur alamat suara “belas” ke alamat data ISD.

N. Lanjutkan ke A.

O. Apakah bil1 lebih besar sama dengan 12 dan lebih kecil sama dengan 19.

   Jika ya, lanjutkan ke C.

   Jika tidak, pemilihan akan dilakukan lagi dengan menyeleksi bil1.

P. Lanjutkan ke C.

Q. Mengatur alamat suara “belas” ke alamat data ISD.

R. Lanjutkan ke A.

S. Apakah bil1 lebih besar sama dengan 20 dan lebih kecil sama dengan 99.

   Jika ya, lanjutkan ke C.

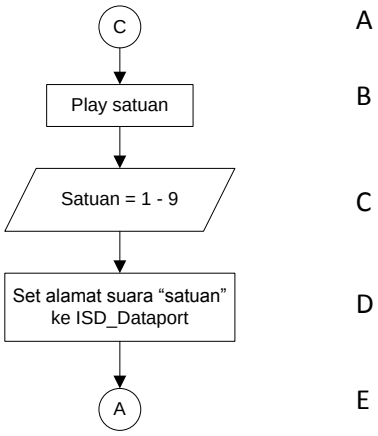
   Jika tidak, lanjutkan ke D.

T. Lanjutkan ke C.

U. Mengatur alamat suara “puluh” ke alamat data ISD.

V. Lanjutkan ke A.

W. Lanjutkan ke D.



Gambar 3.19 *Flowchart* Play Satuan

Penjelasan *Flowchart* pada gambar 3.19 sebagai berikut :

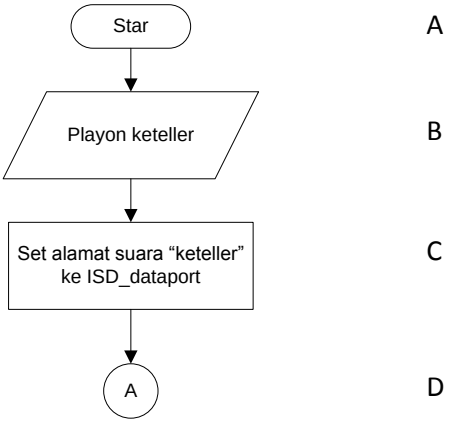
A. Lanjutan C.

B. Putar suara satuan dengan memanggil alamatnya pada ISD.

C. 1 sampai 9 disalin ke satuan.

D. Mengatur alamat suara “satuan” ke alamat data ISD.

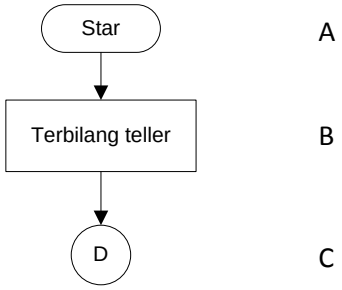
E. Lanjutkan ke A.



Gambar 3.20 *Flowchart* Play Suara Keteller

Penjelasan *flowchart* pada gambar 3.20 sebagai berikut :

A. Mulai proses pemutaran suara.
B. Putar suara keteller dengan memanggil alamatnya pada ISD.
C. Mengatur alamat suara “keteller” ke alamat data ISD.
D. Lanjutkan ke A.



Gambar 3.21 *Flowchart* Suara Terbilang *Teller*

Penjelasan *flowchart* pada gambar 3.21 sebagai berikut :

A. Mulai proses pemutaran suara.
B. Putar suara nomor *teller* dengan memanggil alamatnya pada ISD.
C. Lanjutkan ke D.

Langkah awal dari program adalah menginisialisasi mikrokontroler, yaitu mendefinisikan port mana sebagai input, output atau lain sebagainya. Setelah melakukan inisialisasikan mikrokontroler, dilanjutkan dengan menginisialisasikan display *seven segment*, dan unit suara.

Pada gambar 3.9 setelah melakukan inisialisasi, program selanjutnya adalah melakukan pendeteksi pada tombol *teller* apakah tombol tersebut ditekan atau tidak. Jika tidak ada penekanan pada tombol *teller*, pendeteksi/pembacaan dilanjutkan ke tombol *teller* lainnya apakah ada penekanan atau tidak. Jika tidak juga ada penekanan pada tombol *teller* maka program berada dalam kondisi *standby* dan display seven segment menyala.

Apabila ada penekanan tombol *teller*, maka *Master Teller Unit* akan menghasilkan suara sesuai dengan nomor antrian yang harus dilayani. Selain itu, juga data nomor antrian dikirim ke display *seven segment* baris ke satu untuk menampilkan dan kemudian untuk display *seven segment* baris ke dua akan ditampilkan setelah penekanan tombol selesai dan display yang baris ke satu akan pindah ke display baris ke dua.