

KUNCI ELEKTRONIS BERBASIS DIGITAL

SKRIPSI

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Teknik Elektro



Oleh :

Almido Haryanto Ginting

NIM : 955114024

NIRM : 950051123107120023

**PROGRAM STUDI TEKNIK ELEKTRO
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS SANATA DHARMA
YOGYAKARTA
2001**

**KUNCI ELEKTRONIS
BERBASIS DIGITAL**

SKRIPSI

**Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Teknik Elektro**

Disusun Oleh :

Nama : Almido Haryanto Ginting

NIM : 955114024

NIRM : 950051123107120023

**PROGRAM STUDI TEKNIK ELEKTRO
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS SANATA DHARMA
YOGYAKARTA
2001**

KUNCI ELEKTRONIS BERBASIS DIGITAL

SKRIPSI

Disusun oleh :

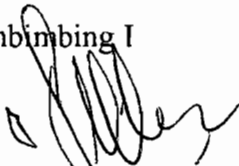
Nama : Almido Haryanto Ginting

NIM : 955114024

NIRM : 950051123107120023

Telah disetujui oleh :

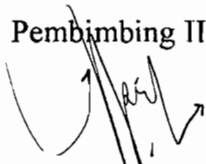
Pembimbing I



Ir. Bambang Sutopo, M. Phil

tanggal :

Pembimbing II



Ir. Th. Prima. A

tanggal :


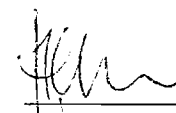
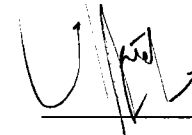
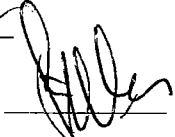

KUNCI ELEKTRONIS BERBASIS DIGITAL

SKRIPSI

Disusun oleh :
Nama : Almido Haryanto Ginting
NIM : 955114024
NIRM : 950051123107120023

Telah dipertahankan di depan panitia penguji
pada tanggal : 21 Juli 2001
dan dinyatakan memenuhi syarat

Susunan Panitia Penguji

Nama Lengkap	tanda tangan
Ketua : Ir. M. Linggo Sumarno, M.T	
Sekretaris : Petrus Setyo Prabowo, S.T	
Anggota : Ir. Bambang Sutopo, M. Phil	
Anggota : Ir. Th. Prima. A	
Anggota : Djoko Untoro S, S.Si, M.T	

Yogyakarta,.....
Fakultas Teknik
Universitas Sanata Dharma
Yogyakarta

Dekan


Ir. G. Harjanto

Halaman Persembahan

Skripsi ini kupersembahkan kepada:

Bapak U. Ginting
Mamak A. Br. Purba
Yanti Ginting
Lita Ginting
Bulang dan Karo
Rina Juita Sembiring
Kila dan Bibik

Serta Almamaterku tercinta

KATA PENGANTAR

Syukur kepada Tuhan Yang Maha Esa, karena kasihNya sehingga penulis dapat menyelesaikan skripsi ini.

Skripsi dengan judul “Kunci Elektronik Berbasis Digital”. Disusun guna memenuhi salah satu syarat untuk memperoleh gelar Sarjana Teknik Jurusan Teknik Elektro, Universitas Sanata Dharma Yogyakarta.

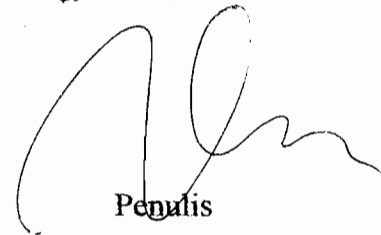
Karena keterbatasan kemampuan sehingga penulis banyak mengalami kesulitan. Tetapi berkat bantuan dan bimbingan dari berbagai pihak akhirnya penulis dapat menyelesaikan skripsi ini. Oleh karena itu penulis mengucapkan terima kasih kepada :

1. Bapak Jr. Bambang Sutopo, M. Phil selaku pembimbing I yang memberikan arahan dan masukan dalam penyusunan skripsi ini.
2. Ibu Ir. Th. Prima A selaku pembimbing II yang juga telah banyak meluangkan waktu, memberikan arahan bimbingan, masukan-masukan dan petunjuk dalam penyusunan skripsi ini.
3. Segenap bapak dan ibu dosen Fakultas Teknik Jurusan Teknik Elektro, Universitas Sanata Dharma Yogyakarta, selaku pribadi yang telah membimbing penulis semasa perkuliahan.
4. Bapak dan mamak yang dengan susah payah membiayai, mendorong, memberi semangat dan mendoakan, serta kakakku dan adikku atas dorongan dan semangatnya.
5. Teman dekatku Rina Juita Sembiring atas dorongan, perhatian, doa, semangat dan kasihnya.

6. Saudara Albert, DM. Naibaho, SE, Banehata Silalahi, Sin Adestin Berutu, Vinsentius Joko Winarno, Despriman, ST dan teman-teman angkatan 95 Fakultas Teknik Jurusan Teknik Elektro Universitas Sanata Dharma atas perhatian dan bantuannya selama penelitian.

Penulis menyadari bahwa tidak ada satupun yang sempurna di dunia ini, demikian juga dengan skripsi yang penulis susun. Oleh karena itu penulis menghargai kritik dan saran yang bersifat membangun. Akhir kata, semoga skripsi ini berguna bagi semua pihak.

Yogyakarta, Juli 2001



Penulis



DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PERSEMBAHAN	iv
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	x
DAFTAR TABEL	xi
BAB I PENDAHULUAN	
1.1. Latar Belakang.....	1
1.2. Perumusan Masalah	2
1.3. Pembatasan Masalah	2
1.4. Tujuan Penelitian.....	2
1.5. Manfaat Penelitian	2
1.6. Sistematika Penulisan.....	2
BAB II DASAR TEORI	
2.1. Keypad.....	5
2.1.1. Rangkaian Penghilang Getaran	5
2.1.2. Rangkaian Encoder Desimal ke BCD.....	6

2.2. Memori.....	8
2.2.1. Register Geser.....	8
2.2.2. Penggerendel-D (D-Latch)	9
2.3. Unit Kendali / Pencacah	10
2.4. Pembanding	11
2.5. Penampil	12
2.5.1. Penampil Tujuh Ruas Katoda Bersama	12
2.5.2. Antarmuka LED.....	12

BAB III PERANCANGAN PERANGKAT KERAS

3.1. Keypad.....	16
3.2. Unit Kendali (control Unit).....	19
3.3. Pembanding 4 bit.....	24
3.4. Penggerendel-D (D-Latch).....	26
3.5. Register Geser	27
3.6. Penampil Tujuh Ruas.....	29

BAB IV ANALISIS DATA

4.1. Analisis Rangkaian Keypad.....	32
4.2. Analisis Rangkaian Pembentuk Pulsa	33
4.3. Analisis Register Geser.....	34
4.4. Analisis Pembanding 4 Bit	35
4.5. Analisis Rangkaian Pengganti Isi Memori dan Pengaktifan Kunci.....	37
4.6. Analisis Rangkaian Kunci Elektronis.....	38

BAB V KESIMPULAN DAN SARAN

5.1. Kesimpulan	40
5.2. Saran.....	41

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Skema kunci elektronik berbasis digital	4
Gambar 2.2 Rangkaian penghilang getaran dan bentuk gelombang	5
Gambar 2.3 Rangkaian Encoder	7
Gambar 2.4 Register Geser	9
Gambar 2.5 Penyimpanan 4 buah data	10
Gambar 2.6 Pembandingan 4 bit menggunakan gerbang XOR	12
Gambar 2.7 Penampil tujuh ruas katoda bersama	13
Gambar 2.8 Ilustrasi bentuk angka 0 sampai 9	13
Gambar 2.9 Antarmuka LED	14
Gambar 3.1 Diagram kotak perancangan perangkat keras	15
Gambar 3.2 Rangkaian keypad	18
Gambar 3.3 Rangkaian pembentuk pulsa	20
Gambar 3.4 Bentuk pulsa yang dihasilkan	22
Gambar 3.5 Pengaktifan pencacah	22
Gambar 3.6 Pembandingan data masukan dengan memori	25
Gambar 3.7 Pengisian data pada memori	26
Gambar 3.8 Penyimpanan 4 data masukan	28
Gambar 3.9 Pembatas arus pada LED	31

DAFTAR TABEL

	Halaman
Tabel 2.1 Tabel keadaan rangkaian enkoder desimal ke BCD.....	7
Tabel 4.1 Tabel kebenaran rangkaian keypad	32
Tabel 4.2 Tabel kebenaran rangkaian pembentuk pulsa	33
Tabel 4.3 Tabel kebenaran register geser.....	35
Tabel 4.4 Contoh kondisi komparator perbandingan 4 data biner 4 bit	36
Tabel 4.5 Tabel pengamatan rangkaian pengganti isi memori dan pengaktifan kunci.....	37
Tabel 4.6 , Tabel kebenaran rangkaian kunci elektronik berbasis digital	38

BAB I

PENDAHULUAN

1.1. Latar belakang

Pada masa sekarang ini, keamanan merupakan sesuatu hal yang sangat penting. Saat ini telah banyak kunci pengaman manual yang beredar dipasaran dengan berbagai macam bentuk dan cara kerjanya. Tetapi kunci pengaman yang manual itu mempunyai kekurangan yang dapat mengakibatkan keamanan menjadi kurang terjamin. Kekurangan pada kunci manual antara lain kunci menjadi mudah hilang, mudah dicuri, dan juga mudah diduplikat.

Dengan adanya perkembangan teknologi yang sangat pesat, terutama dalam bidang elektronis, banyak peralatan yang dibuat berdasarkan pada sistem elektronis sehingga alat tersebut menjadi semakin simpel. Hal ini disebabkan karena rangkaian yang besar atau rumit dapat dibuat menjadi kepingan yang kecil yang kita kenal dengan sebutan IC (*Integrated Circuit*).

Perkembangan teknologi elektronis ini juga dapat mengatasi kekurangan kunci manual tersebut dengan di buatnya kunci yang berdasarkan pada sifat elektronis. Kunci ini tidak mudah hilang, dicuri, atau juga diduplikat. Karena itu penulis mengambil topik kunci elektronis untuk mendapatkan sebuah kunci pengaman yang lebih terjamin keamanannya dibanding kunci manual, agar dapat membantu manusia dalam bidang keamanan.

1.2. Perumusan Masalah

Permasalahannya adalah bagaimana cara membuat sebuah kunci elektronik berbasis digital dengan masukan berupa kombinasi 4 angka desimal.

1.3. Pembatasan Masalah

Pembuatan kunci elektronik berbasis digital ini mempunyai batasan permasalahan, yaitu:

- a. Kunci elektronik berbasis digital ini menggunakan dasar-dasar teknik digital
- b. Masukan dari kunci elektronik ini menggunakan *keypad* dengan keluaran bilangan BCD (*Binnary Code/Decimal*).
- c. Kombinasi angka masukan dibatasi sebanyak empat buah.

1.4. Tujuan Penelitian

Merancang dan membuat kunci elektronik berbasis digital dengan masukan kombinasi 4 angka desimal.

1.5. Manfaat Penelitian

Kunci elektronik berbasis digital ini, diharapkan dapat dipergunakan sebagai kunci pribadi yang lebih aman, khusus untuk sesuatu yang bersifat pribadi baik itu ruangan ataupun yang lainnya.

1.6. Sistematika Penulisan

Sistematika penulisan pada Tugas Akhir ini adalah sebagai berikut:

a. Bab I Pendahuluan

Pendahuluan ini berisi latar belakang, perumusan masalah, pembatasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan

b. Bab II Dasar Teori

Bab ini berisi penjelasan tentang peralatan-peralatan yang digunakan untuk membuat kunci elektronik berbasis digital yang terdiri dari rangkaian penghilang getaran, rangkaian encoder, pencacah, register geser, penggerendel-D, pembanding, penampil tujuh ruas katoda bersama, dan antarmuka LED.

c. Bab III Perancangan Perangkat Keras

Bab ini berisi penjelasan tentang perancangan perangkat keras yang dibuat pada tugas akhir ini, yang terdiri dari rangkaian *keypad*, unit kendali, register geser, penggerendel D, pembanding empat bit dan penampil.

d. Bab IV Analisis dan Pembahasan

Pada bab ini berisi mengenai analisis dan pembahasan hasil perancangan perangkat keras

e. Bab V Kesimpulan dan Saran

Bab ini berisi ringkasan hasil penelitian yang telah dilakukan dan usulan yang berisi ide-ide untuk perbaikan atau pengembangan terhadap penelitian yang telah dilakukan.

f. Daftar pustaka

Pada bagian ini akan ditunjukkan literatur yang digunakan dalam perancangan perangkat keras

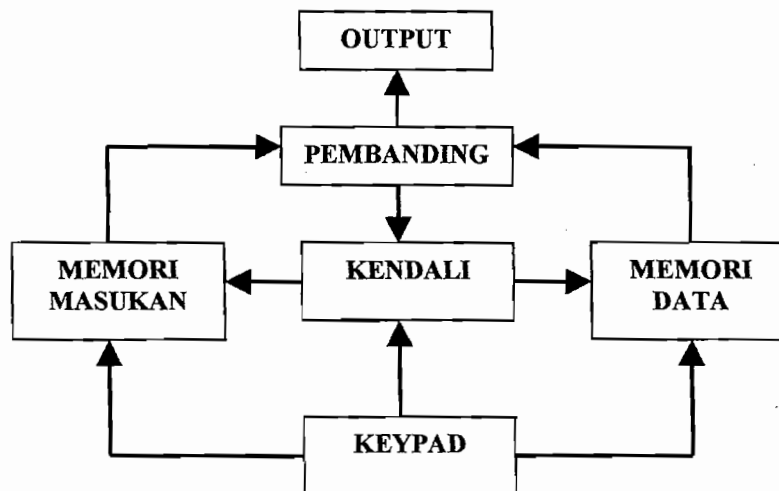
BAB II

DASAR TEORI

Kunci Elektronik adalah suatu alat yang digunakan untuk mengunci secara elektronik. Kunci elektronik ini bekerja berdasarkan pada prinsip kerja pembandingan (*Comparator*) yaitu dengan membandingkan data masukan dengan data pada memori, apakah data masukan dan data pada memori sama atau tidak. Apabila kedua data sama, maka kunci dapat diaktifkan. Tetapi apabila kedua data tidak sama, kunci tidak dapat diaktifkan.

Ada beberapa macam kunci elektronik. Salah satunya adalah kunci elektronik yang dibuat pada tugas akhir ini, yaitu kunci elektronik dengan masukan berupa kombinasi empat buah angka.

Skema kunci elektronik berbasis digital ini dapat dilihat pada gambar 2.1.



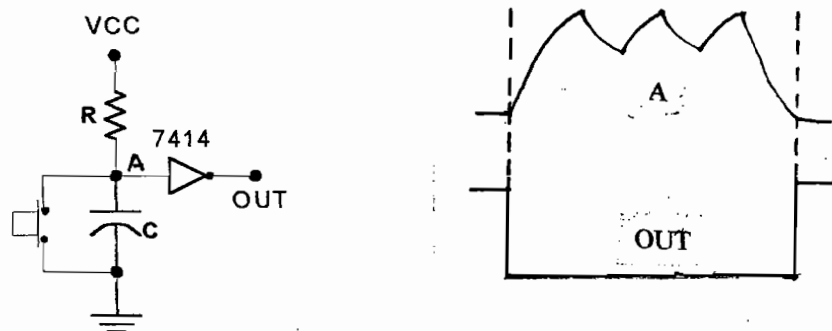
Gambar 2.1
Skema kunci elektronik berbasis digital

2.1. Keypad

Keypad merupakan suatu piranti yang digunakan untuk memasukkan sinyal masukan. *Keypad* merupakan piranti masukan yang terdiri dari tombol-tombol.

2.1.1. Rangkaian Penghilang Getaran

Tombol-tombol saklar mekanik pada *keypad* memiliki kekurangan yaitu adanya getaran dalam kontak saklar yang disebabkan oleh gerakan pegas dari saklar yang mengakibatkan terjadinya beberapa kontak sebelum kembali normal. Untuk mengatasi hal ini, maka dibutuhkan rangkaian penghilang getaran yang dipasang pada setiap saklar. Ada beberapa rangkaian penghilang getaran, salah satunya dapat dilihat pada gambar 2.2.



Gambar 2.2
Rangkaian penghilang getaran
dan bentuk gelombang

Rangkaian penghilang getaran yang ditunjukkan oleh gambar 2.2 di atas menggunakan prinsip pengisian kapasitor hingga penuh. Lama pengisian kapasitor hingga penuh dapat dihitung dengan rumus

$$\tau = R \times C$$

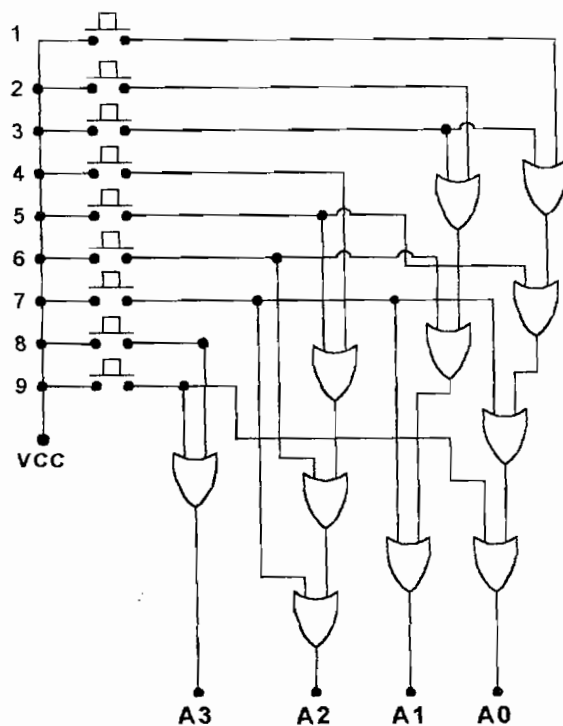
di mana τ =konstanta waktu, R=resistor dan C=capasitor.

Pada saat saklar terbuka, kapasitor C akan terisi oleh arus yang mengalir melalui Resistor hingga penuh. Pada saat ini masukan gerbang NOT adalah 1 keluarannya akan bernilai 0. Sebaliknya pada saat saklar terhubung, akan terjadi pembuangan isi kapasitor ke *ground* secara cepat. Selama saklar masih terhubung, pengosongan kapasitor akan terus berjalan. Hal ini mengakibatkan masukan gerbang NOT akan menjadi 0 dan menghasilkan keluaran bernilai 1. Gambar bentuk masukan dan keluaran dari gerbang NOT di atas dapat dilihat pada lampiran

2.1.2. Rangkaian Encoder Desimal ke BCD

Rangkaian *encoder* ini terdiri dari sebelas buah gerbang OR dengan dua masukan. Rangkaian logika ini dapat dilihat pada gambar 2.3. Keluaran dari tombol saklar akan dihubungkan dengan gerbang OR. Keluaran dari gerbang yang satu dihubungkan dengan masukan pada gerbang OR yang lain dengan metode tertentu, yang pada akhirnya keluarannya akan menghasilkan bilangan BCD (*Binary Coded Decimal*).

Pada saat ada saklar yang ditekan, gerbang OR yang terhubung dengan saklar tersebut akan aktif karena adanya sinyal berkeadaan tinggi atau bernilai 1. Dengan adanya kombinasi beberapa gerbang OR sedemikian rupa, rangkaian *encoder* ini akan mengubah sebuah bilangan desimal menjadi sebuah bilangan BCD 4 bit. Hasil dari rangkaian ini dapat dilihat pada tabel 2.1.



Gambar 2.3
Rangkaian encoder

Tabel 2.1

Tabel Keadaan Rangkaian Encoder Desimal ke BCD

Saklar yang ditekan	KELUARAN			
	A3	A2	A1	A0
-----	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Persamaan yang didapat dari tabel di atas adalah :

$$A_3 = S_8 + S_9$$

$$A_1 = S_2 + S_3 + S_6 + S_7$$

$$A_2 = S_4 + S_5 + S_6 + S_7$$

$$A_0 = S_1 + S_3 + S_5 + S_7 + S_9$$

2.2. Memori

Untuk melakukan fungsinya, sistem digital memerlukan fasilitas memori untuk penyimpanan data secara permanen atau sementara. Ada dua tipe memori: *volatile* dan *non-volatile*. Memori *volatile* akan kehilangan data yang tersimpan di dalamnya pada saat catu daya dihilangkan. Pada memori *non-volatile*, data akan tetap tersimpan di dalam memori tidak bergantung ada tidaknya catu daya. Yang dibuat di sini adalah memori yang dibuat dari register geser dan penggerendel-D (*D-Latch*).

2.2.1. Register Geser

Register merupakan sekelompok flip-flop yang dapat digunakan untuk menyimpan sebuah bilangan biner. Flip-flop yang dipakai dalam register geser ini adalah D flip-flop yang terpicu pinggiran positif.

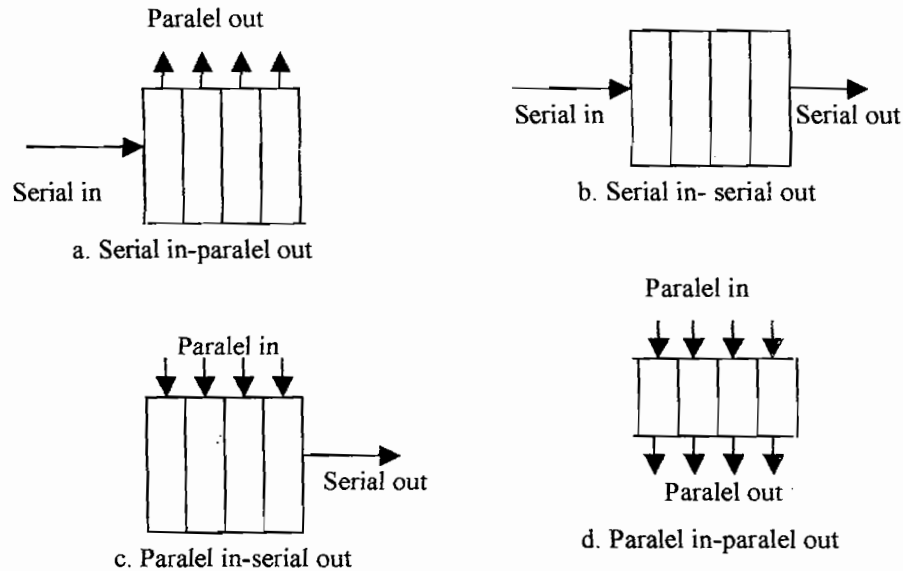
Ada dua metoda untuk menggeser informasi biner ke dalam suatu register. Yang pertama adalah yang berhubungan dengan penggeseran informasi ke dalam register bit demi bit secara seri atau berderet dan dikembangkan menjadi register geser seri (*Serial Shift Register*). Metoda yang kedua berkenaan dengan penggeseran semua bit ke dalam register pada saat yang bersamaan dan dikembangkan menjadi register geser paralel (*Parallel Shift Register*).

Ada empat cara agar register geser dapat digunakan untuk menyimpan dan memindahkan data dari satu bagian ke bagian sistem yang lain :

1. Masukan seri ke keluaran paralel (*Serial In Parallel Out*)
2. Masukan seri ke keluaran seri (*Serial In Serial Out*)
3. Masukan paralel ke keluaran seri (*Parallel In Serial Out*)

4. Masukan paralel ke keluaran paralel (*Paralel In Paralel Out*).

Ilustrasi keempat cara ini dapat dilihat pada gambar 2.4.



Gambar 2.4
Register geser

2.2.2. Penggerendel-D (D-Latch)

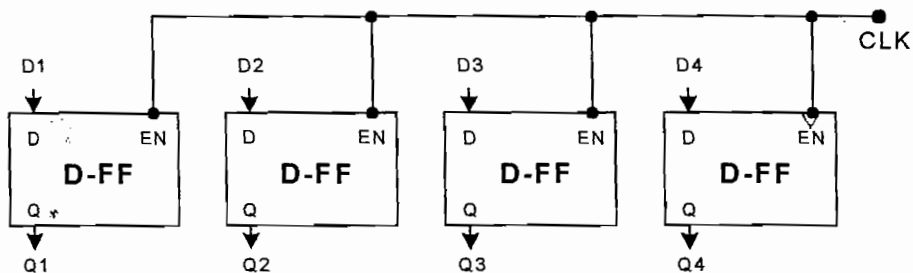
D-Latch merupakan jenis flip-flop D yang nilai Q-nya dapat mengikuti nilai D pada saat *clock* berada pada keadaan tinggi. Dengan perkataan lain, jika bit data berubah pada saat *clock* berada pada keadaan tinggi, nilai terakhir D sebelum *clock* turun kembali merupakan nilai D yang tersimpan

Untaian ini terdiri dari sebuah AND Set- Clear flip-flop dan pengatur pulsa gerbang AND satu dan dua. Proses kerjanya digambarkan sebagai berikut :

Ketika *clock* berada dalam keadaan rendah, masukan D tidak akan mempunyai efek, sebab keadaan rendah ini menjaga masukan AND keduanya adalah tinggi. Sebaliknya ketika *clock* berada dalam keadaan tinggi, masukan D akan menghasilkan keadaan rendah pada *Set-Clear* yang mengakibatkan keluaran

Q akan berada pada keadaan yang sama dengan masukan D. Jika masukan D berubah ketika *clock* berada dalam keadaan tinggi, keluaran Q akan mengikuti perubahan ini.

Gambar 2.5 menggambarkan gagasan tentang penyimpanan sementara. empat buah penggerendel-D di *drive* oleh pulsa *clock* yang sama. Pada saat pulsa *clock* naik, data masukan dimuatkan ke dalam flip-flop dan muncul pada keluaran. Selanjutnya pada saat *clock* turun, keluaran akan mempertahankan data yang bersangkutan.



Gambar 2.5
Penyimpanan 4 buah data

Nilai keluaran Q akan terus dipertahankan selama *clock* berada dalam keadaan rendah dan tidak kehilangan catu daya.

2.3. Unit Kendali / Pencacah

Unit kendali ini terdiri dari beberapa fungsi. Salah satunya adalah berfungsi sebagai pencacah.

Pencacah adalah sekelompok flip-flop yang disusun sedemikian rupa sehingga menunjukkan cacah pulsa total yang diumpankan pada masukan.

Pencacah ini menggunakan pulsa sebagai masukan dan digit biner sebagai keluarannya.

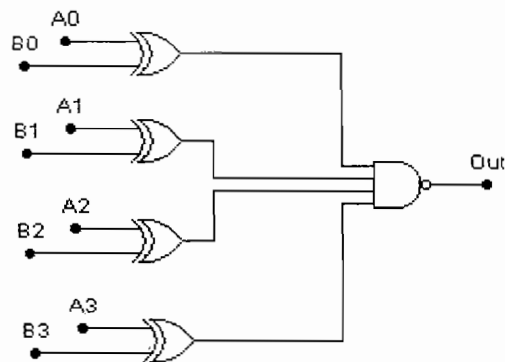
Pencacah secara garis besar dapat dibagi menjadi dua bagian, yaitu pencacah tak sinkron atau tak langsung dan pencacah sinkron atau langsung. Perbedaan ini terletak pada terminal *clock* flip-flop. Pada pencacah sinkron, terminal *clock* semua flip-flop mendapat pulsa *clock* yang sama. Sedangkan pada pencacah tak sinkron, terminal *clock* pada flip-flop mendapat pulsa *clock* yang berbeda sehingga perubahan yang terjadi pada setiap flip-flop tidak seragam.

2.4. Perbandingan

Membandingkan dua angka adalah sebuah operasi yang menghasilkan 3 keadaan apakah angka yang satu lebih besar dari, lebih kecil dari atau sama dengan angka yang lain.

Perbandingan yang paling sederhana membandingkan bit demi bit dari dua bilangan biner, dan outputnya akan bernilai 1 jika keduanya sama. Sebuah gerbang XOR adalah cara paling mudah untuk membandingkan kesamaan bit. Jika kedua bit adalah sama (0-0 atau 1-1), keluaran dari gerbang XOR adalah satu (1).

Pada perbandingan yang digunakan untuk membandingkan dua data yang masing-masing terdiri dari 4 bit, dibutuhkan empat buah gerbang XOR. Untuk mendapatkan hasil akhir perbandingan, hubungkan ke-empat keluaran gerbang XOR pada sebuah gerbang AND (lihat gambar 2.6). Dengan jalan itu, bila ke-empat keluaran bernilai 1 maka keluaran gerbang AND juga akan tinggi atau bernilai 1.



Gambar 2.6
Pembanding 4 bit menggunakan gerbang XOR

2.5. Penampil

Penampil ini dapat bermacam-macam jenisnya, diantaranya adalah penampil tujuh ruas katoda bersama yang didukung dengan adanya rangkaian antarmuka LED

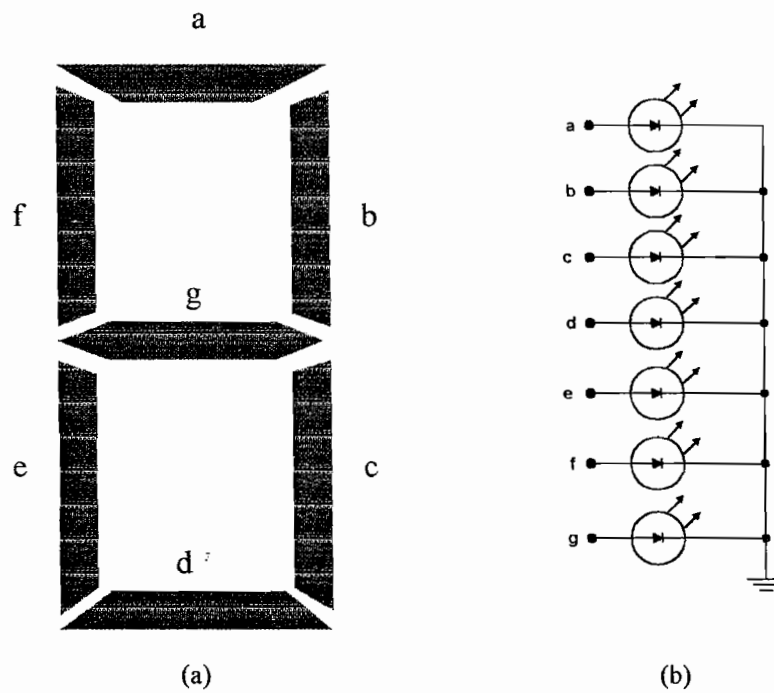
2.5.1. Penampil Tujuh Ruas Katoda Bersama

Penampil tujuh ruas katoda bersama terdiri dari tujuh ruas terpisah yang diberi label S0 sampai S6 seperti ditunjukkan pada gambar 2.7.

Tujuh ruas merupakan cacah ruas minimum yang diperlukan untuk menampilkan angka 0 sampai 9 seperti diilustrasikan pada gambar 2.8.

2.5.2. Antarmuka LED

Pada dasarnya penampil tujuh ruas katoda bersama terdiri dari 7 buah LED (*Light Emitting Diode*). LED digunakan untuk mengubah arus listrik menjadi cahaya sehingga untuk menyalakan salah satu ruas dari tampilan, arus listrik harus diarahkan ke dioda dari ruas yang dimaksud.



Gambar 2.7
Penampil 7 ruas katoda bersama

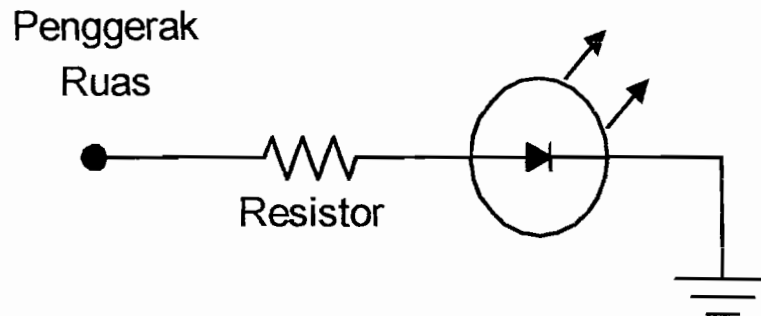
1234567890

Gambar 2.8
Ilustrasi bentuk angka 0 sampai 9

Sebelum ruas digerakkan, keluaran sistem digital harus diubah kedalam bentuk yang sesuai untuk menggerakkan tampilan yaitu ke dalam bentuk BCD (*Binary Coded Decimal*) yang akan diubah menjadi isyarat tujuh jalur oleh dekoder untuk menggerakkan masing-masing ruas.

LED pada dasarnya adalah sebuah dioda yang memancarkan cahaya jika ada arus yang mengalir ke arah maju. Tegangan maju dioda LED yaitu sebesar 2V. Karena rangkaian TTL bekerja dengan tegangan yang lebih tinggi, harus

terdapat resistor pembatas arus yang dihubungkan antara setiap keluaran dekoder dengan masukan peraga yang bersesuaian.



Gambar 2.9
Antarmuka LED

Nilai R yang dibutuhkan dapat dicari dengan cara sebagai berikut :

$$R = (5 - V_d) / I$$

Dimana R = resistor yang dibutuhkan

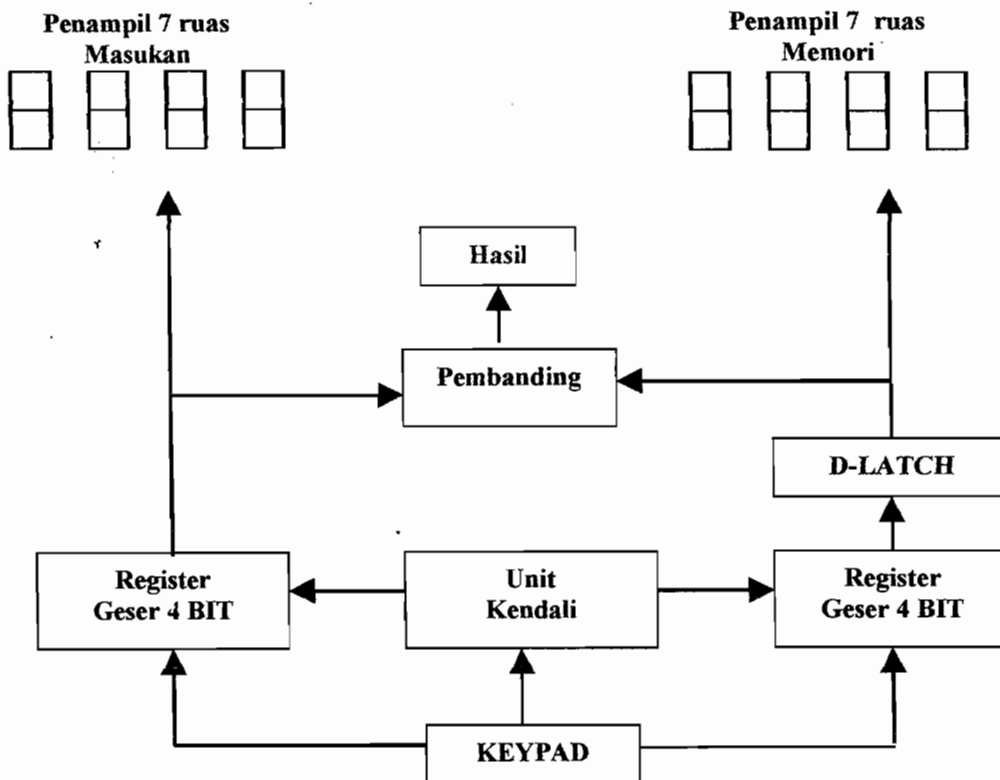
V_d = tegangan maju dioda LED (+ ½V sd 2V)

I = besar arus yang dibutuhkan oleh tiap ruas (≈10 mA sd 20 mA).

BAB III

PERANCANGAN PERANGKAT KERAS

Dalam bab ini akan dibahas mengenai pembuatan perangkat keras pada kunci elektronik berbasis digital. Diagram kotak dari perancangan perangkat keras ini dapat dilihat pada gambar 3.1.



Gambar 3.1
Diagram kotak perancangan perangkat keras

3.1. Keypad

Keypad ini menggunakan 12 buah saklar *push-on* yang merupakan saklar mekanik. Saklar-saklar ini memiliki 2 buah kutub yang berbeda. Karena saklar yang digunakan adalah saklar mekanik, maka keluaran dari saklar tersebut harus dihubungkan dengan suatu untai penghilang getaran. Hal ini disebabkan karena saklar mekanik mempunyai kekurangan karena adanya *contact Bounce* yang disebabkan gerakan pegas dari saklar yang mengakibatkan terjadinya beberapa sinyal. Dengan adanya rangkaian penghilang getaran ini, sinyal yang dihasilkan adalah kontak pertama yang terjadi pada saklar, sedangkan kontak berikutnya akan diabaikan.

Rangkaian penghilang getaran pada *keypad* ini menggunakan resistor sebesar 56 K Ω dan kapasitor sebesar 1 μ F. Dengan nilai R dan C tersebut, konstanta waktu pengisian kapasitor dapat diketahui, yaitu :

$$\begin{aligned}\tau &= R \times C \\ &= 56 \text{ K}\Omega \times 1 \mu\text{F} \\ &= 56.10^3 \times 1.10^{-6} \\ &= 0,056 \text{ detik}\end{aligned}$$

Dengan konstanta waktu sebesar 0,056 detik ini, waktu untuk penekanan tombol dan pulsa yang dibentuk akan stabil. Bila konstanta waktu makin kecil, maka pulsa yang dihasilkan pada penekanan tombol bisa lebih dari satu pulsa. Bila konstanta waktu semakin besar, jarak waktu penekanan tombol pertama dan kedua menjadi semakin besar atau lama.

Tombol bersimbol “0”, “*” dan “#” tidak dihubungkan dengan untai *encoder*, tetapi dihubungkan langsung sesuai dengan fungsinya masing-masing. Oleh karena itu, pada saat penekanan tombol-tombol ini tidak akan ada data yang dihasilkan oleh *keypad*.

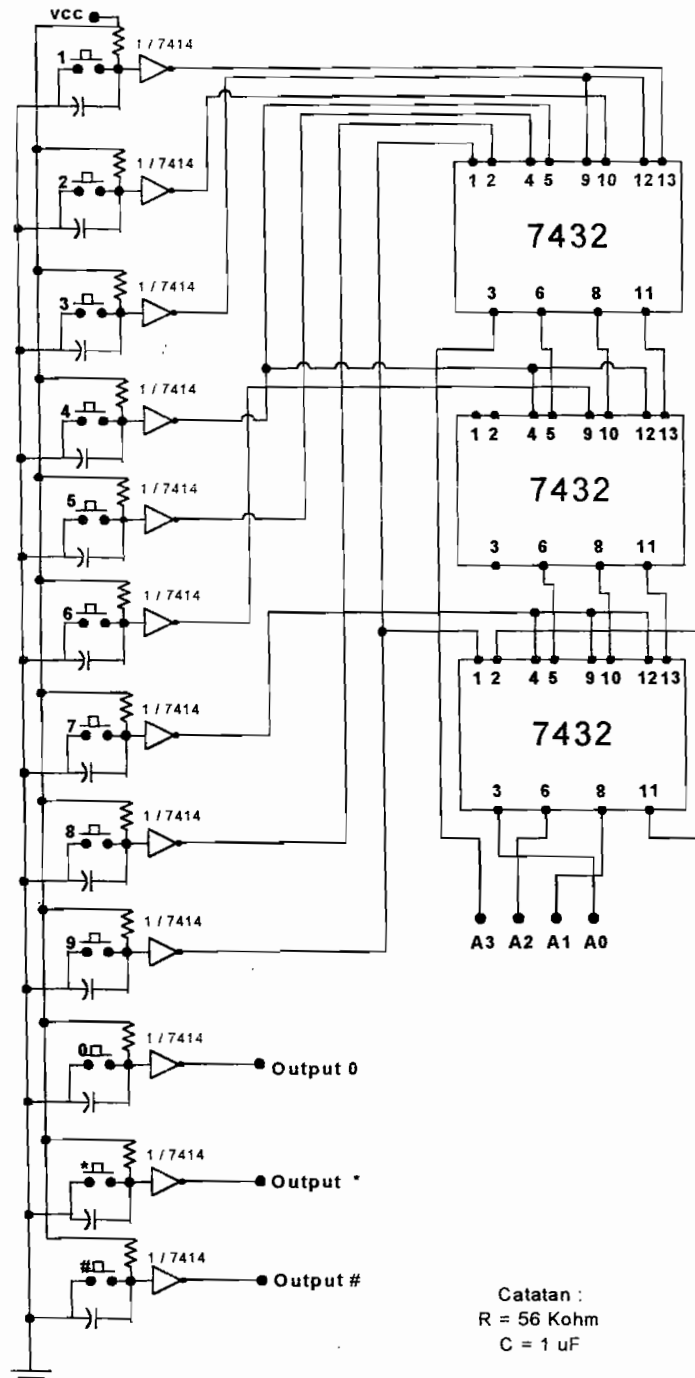
Keluaran yang dihasilkan oleh *keypad* ini adalah bilangan BCD (*Binary Coded Decimal*) yang terdiri dari bilangan biner 4 bit. Keadaan 0 atau 0000 dihasilkan pada saat tidak ada penekanan tombol pada *keypad*, sedangkan keadaan 1 sampai 9 akan dihasilkan sesuai dengan tombol berangka yang ditekan. Gambar rangkaian *keypad* ini dapat dilihat pada gambar 3.2.

Cara kerja dari *keypad* ini adalah sebagai berikut : Pada saat tombol bersimbol 1 sampai 9 pada *keypad* ini ditekan, akan ada sinyal yang masuk pada rangkaian *encoder*. Rangkaian *encoder* yang terdiri dari kombinasi beberapa gerbang OR ini akan mengubah sinyal yang masuk menjadi bilangan BCD 4 bit sesuai dengan tombol yang ditekan.

Misalnya tombol yang ditekan adalah tombol 3, rangkaian *encoder* akan mengubah sinyal yang masuk menjadi bilangan BCD 0011 yang berarti bernilai 3. Bila tombol 6 pada *keypad* ditekan, maka sinyal yang masuk pada *encoder* akan diubah menjadi bilangan BCD 0110 yang berarti bernilai 6.

Bila tombol bersimbol angka 0 ditekan, rangkaian *encoder* tidak akan aktif karena tidak ada sinyal masukan berupa keadaan tinggi yang masuk pada rangkaian *encoder* ini. Hal ini sama dengan keadaan pada saat tidak ada tombol yang ditekan yaitu keluaran 0000 yang dihasilkan oleh *keypad*. Tetapi dengan penekanan tombol bersimbol 0 ini, dihasilkan pulsa yang digunakan untuk

memasukkan data pada memori. Sehingga dengan ditekannya tombol bersimbol “0” ini, data yang masuk adalah data 0 yang dalam bentuk BCD ditulis dengan simbol 0000.



Gambar 3.2
Rangkaian keypad

Begitu juga dengan penekanan tombol bersimbol * dan #. Pada kasus ini rangkaian *encoder* tidak akan aktif dan keluaran *keypad* berada pada keadaan 0000. Tetapi penekanan tombol ini tidak akan memasukkan data 0000 pada memori seperti yang terjadi pada penekanan tombol 0. Hal ini disebabkan karena penekanan tombol bersimbol * dan # tidak akan menghasilkan pulsa.

Rangkaian penghilang getaran pada keypad ini menggunakan inverter dengan *trigger schmitt* yang dikemas dalam bentuk IC 7414. Prinsip kerja pemacu *Schmitt* ini adalah sebagai berikut ; Pada saat tegangan berubah dari keadaan rendah ke keadaan tinggi, setelah tegangan naik 1,7 V dari keadaan rendah, pemacu *schmitt* akan langsung menghasilkan keluaran pada keadaan tinggi, sebaliknya saat tegangan berubah dari keadaan tinggi ke keadaan rendah, setelah tegangan turun sebesar 0,9 V dari keadaan tinggi, pemacu *schmitt* akan langsung menghasilkan keluaran pada keadaan rendah. Karena itu dengan menggunakan pemacu *Schmitt* ini, dapat diperoleh waktu naik dan turun yang cepat, tidak peduli bagaimana lambatnya perubahan tegangan masukan.

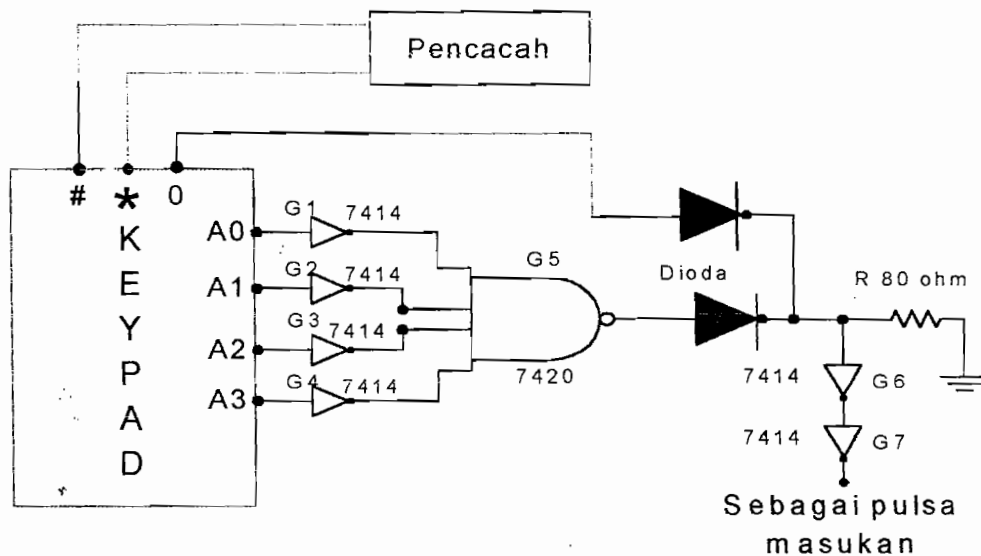
3.2. Unit Kendali (Control Unit)

Kunci elektronik berbasis digital ini memerlukan suatu unit kendali (*control unit*) agar dapat berjalan dengan baik.

Pada tugas akhir ini, *control unit* ini mempunyai peranan yang beraneka ragam. Dan peranan yang paling penting adalah untuk menghasilkan pulsa yang sangat dibutuhkan oleh sistem. Tanpa adanya pulsa ini, suatu sistem digital tidak dapat bekerja sama sekali.

Pulsa ini tidak dihasilkan oleh suatu pembangkit pulsa (osilator) melainkan diperoleh dari pensaklaran pada *keypad* yang diolah oleh unit kendali ini. Hal ini dilakukan agar pergeseran data pada sistem terjadi pada saat yang bersamaan dengan penekanan saklar.

Proses pembangkitan pulsa ini adalah sebagai berikut (lihat gambar 3.3):



Gambar 3.3
Rangkaian pembentuk pulsa

Keluaran (*Output*) dari *keypad* pada saat tidak ada tombol yang ditekan adalah bernilai 0 yang dalam bentuk bilangan BCD bertanda 0000. Keluaran tersebut akan di inverterkan oleh gerbang G1 sampai G4 sehingga keluaran *Keypad* ini akan berubah menjadi 1111 dan menjadi masukan bagi gerbang G5. Keluaran gerbang G5 akan menjadi rendah (0). Akibatnya keluaran pada *output* juga akan rendah (0) setelah diinverterkan sebanyak dua (2) kali. Hal ini dibuat untuk mendapatkan keluaran yang bernilai tinggi (1) pada saat tombol ditekan. Inverter disini menggunakan inverter dengan *trigger Schmitt* (dikemas dalam bentuk IC 7414) agar pulsa yang dihasilkan akan lebih tajam.

Pada saat ada tombol yang ditekan, keluaran *keypad* tidak semua dalam keadaan rendah (0) tetapi ada sebagian yang bernilai tinggi (1) tergantung tombol yang mana yang ditekan. Akibatnya keluaran gerbang NAND G5 adalah tinggi (1) sehingga keluaran berubah dari rendah menjadi tinggi (0 menjadi 1). Pada saat tombol dilepas, keluaran akan berubah dari keadaan 1 menjadi keadaan 0 atau kembali kepada posisi semula (normal) pada saat tombol tidak ditekan.

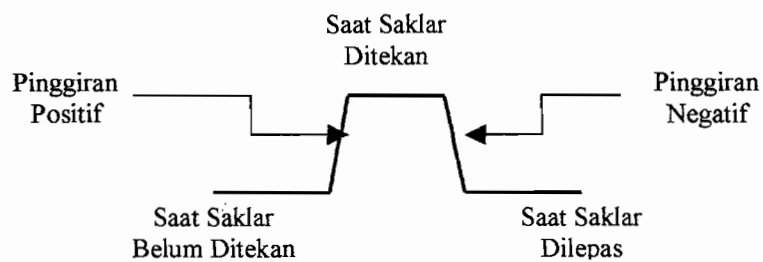
Dioda digunakan untuk mengatur arah arus agar arus tidak masuk ke terminal keluaran. Dioda tersebut tidak dapat dihubungkan langsung dengan gerbang inverter, karena keluaran dioda tersebut akan mengambang disebabkan karena terjadinya suatu rangkaian terbuka. Karena itu keluaran tersebut harus dihubungkan dengan *ground* dengan melewati sebuah tahanan. Dengan adanya tahanan tersebut, masukan inverter G6 tidak akan mengambang.

Keadaan mengambang tersebut mengakibatkan adanya tegangan sebesar 1,2 V dengan arus yang melewati dioda sebesar ± 15 mA. Nilai arus sebesar 15 mA ini merupakan besar arus keluaran rangkaian TTL pada keadaan rendah. Untuk menghilangkan tegangan ambang sebesar 1,2 V tersebut, besar tahanan yang dibutuhkan adalah sebesar

$$\begin{aligned} R &= V / I \\ &= 1,2 \text{ V} / 15 \text{ mA} \\ &= 80 \Omega \end{aligned}$$

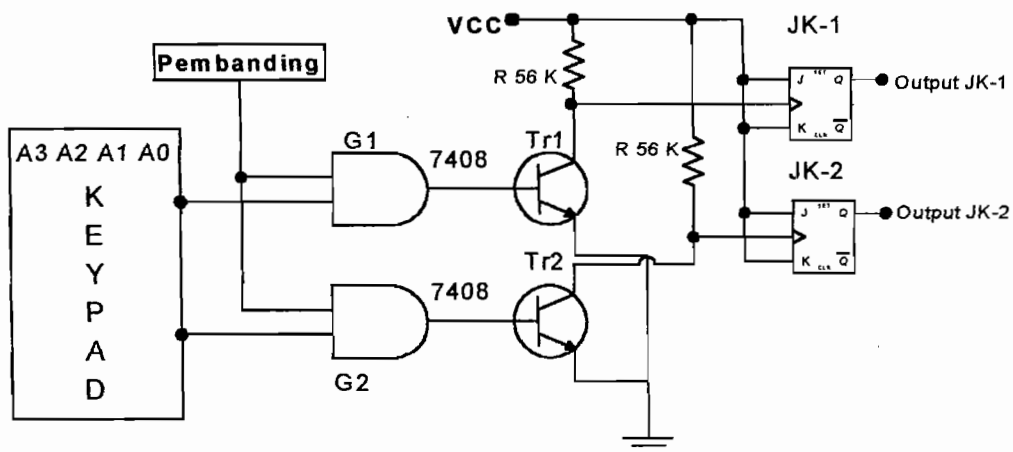
Dengan digunakan tahanan sebesar 80Ω ini, semua arus dapat dilewatkan ke *ground* sehingga pada saat masukan dioda sebesar 0 V , masukan inverter G6 juga akan sebesar 0 V .

Apabila ketiga keadaan tersebut digabungkan, maka akan dihasilkan bentuk sebuah pulsa yang dapat dilihat pada gambar 3.4.



Gambar 3.4
Bentuk pulsa yang dihasilkan

Selain menghasilkan pulsa, unit kendali ini juga melakukan fungsi logika untuk mengaktifkan penggantian isi memori dan juga mengaktifkan kunci. Rangkaian ini dapat dilihat pada gambar 3.5.



Gambar 3.5
Pengaktifan pencacah

Pada rangkaian ini, keadaan pembanding $A=B$ sangat berpengaruh penting. Sebab jika nilai $A=B$ berada dalam keadaan rendah (0), rangkaian ini tidak akan berfungsi sebab gerbang AND tidak akan terbuka pada keadaan ini walaupun berapapun nilai masukan yang terjadi, tidak akan menghasilkan perubahan pada keluaran. Hal ini dikenal dengan keadaan *don't care* (tidak peduli).

Apabila $A=B$ berada dalam keadaan tinggi (1) gerbang AND akan aktif bila masukan 1 dan 2 bernilai 1. Pada kondisi ini, apabila saklar bertanda * ditekan, gerbang AND G1 akan menghasilkan keluaran bernilai tinggi. Ketika saklar dilepas, pulsa yang terbentuk dari pensaklaran akan menggerakkan pencacah yang akan mengakibatkan keluaran flip-flop JK-1 akan berubah. Keluaran flip-flop JK-1 ini digunakan untuk mengaktifkan penggantian isi memori.

Ketika saklar bertanda # ditekan, gerbang AND G2 akan menghasilkan keluaran bernilai tinggi. Hal ini juga akan menggerakkan pencacah seperti halnya di atas yang akan mengakibatkan perubahan keluaran pada flip-flop JK-2. Keluaran flip-flop ini akan digunakan untuk mengaktifkan kunci.

Pada rangkaian penggantian isi memori dan mengaktifkan kunci, transistor digunakan untuk menguatkan sinyal, sedangkan resistor digunakan untuk mengatur agar perubahan yang terjadi pada kolektor transistor bernilai 0 pada keadaan rendah dan 1 pada keadaan tinggi.

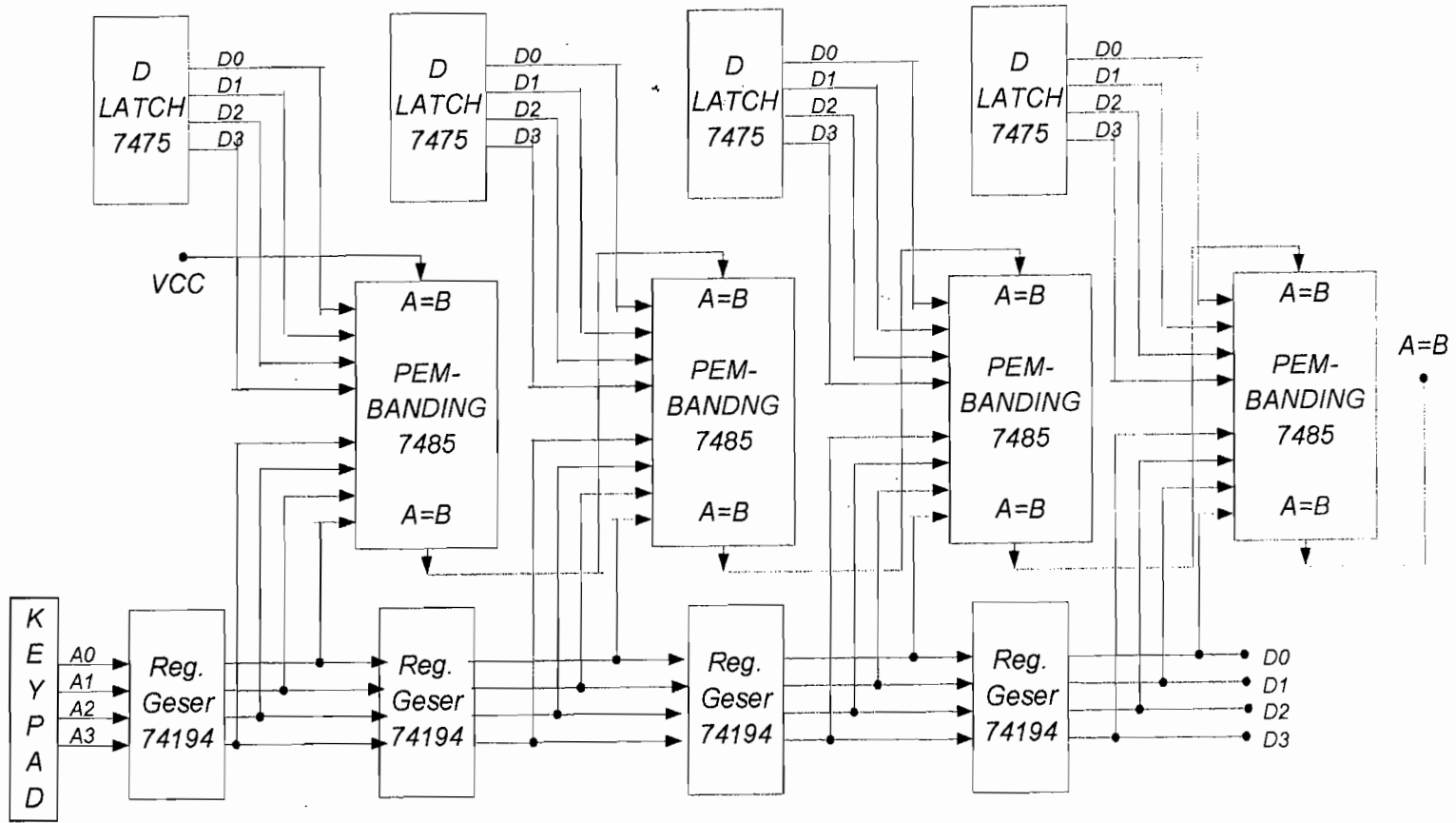
3.3. Pembanding 4 Bit

Pembanding menggunakan pembanding 4 bit yang dikemas dalam bentuk IC 7485. Untuk membandingkan 4 buah data, dibutuhkan 4 buah IC 7485 yang dihubungkan dengan cara terminal A=B pada pembanding I diberi keadaan tinggi sehingga pembanding ini akan menghasilkan keluaran bernilai 1 pada terminal keluaran A=B pada pembanding I apabila data yang dibandingkan adalah sama. Terminal keluaran ini di hubungkan dengan terminal masukan A=B pada pembanding II, begitu seterusnya (lihat gambar 3.6).

Pembanding ini akan membandingkan data pada memori dengan data pada pada register geser atau memori masukan. Pembanding I akan membandingkan data pada register geser I dengan data pada *D-Latch* I. Pembanding II akan membandingkan data pada register geser II dengan data pada *D-Latch* II. Begitu seterusnya.

Data dibandingkan dengan cara membandingkan bit per bit sesuai dengan bobotnya masing-masing. Misalnya *MSB* (*Most Significant Bit*) akan dibandingkan dengan *MSB* dan *LSB* (*Least Significant Bit*) akan dibandingkan dengan *LSB*. Sehingga apabila terjadi perbedaan walaupun hanya satu bit, maka keluaran terminal A=B pada pembanding tidak akan bernilai 1.

Apabila salah satu dari ke empat pembanding keluaran terminal A=B berada pada keadaan rendah (0) maka keluaran A=B pada pembanding selanjutnya juga akan bernilai 0 walaupun data yang dibandingkan oleh pembanding ini sama. Misalnya, setelah dibandingkan oleh pembanding I ternyata ada bit yang tidak sama. Akibatnya keluaran pada terminal A=B adalah rendah.



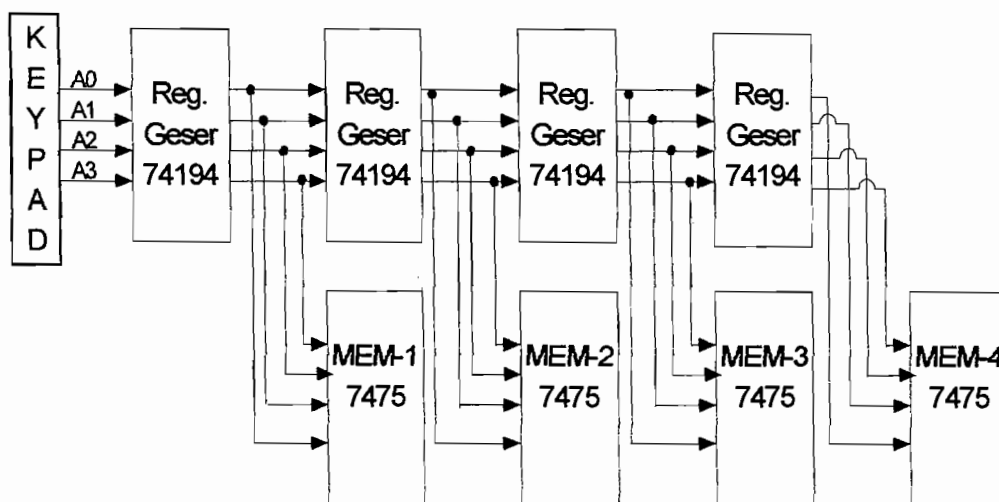
Gambar 3.6
Pembandingan Data Masukan
Dengan Isi Memori



Keluaran rendah ini akan dimasukkan pada terminal masukan A=B pada pembanding II. Akibat masukan yang rendah ini, secara otomatis keluaran pembanding II akan berada pada keadaan rendah walaupun hasil pembandingannya adalah sama. Akibatnya keluaran pada pembanding yang terakhir juga akan berada dalam keadaan rendah. Hasil pembandingan akhir ditunjukkan oleh hasil pembandingan comparator ke-4.

3.4. Penggerendel-D (*D-Latch*)

Sebagai memori penyimpan data, digunakan penggerendel-D yang dikemas dalam bentuk IC 7475. Data pada memori ini dimasukkan dari data pada memori masukan yang dihubungkan seperti terlihat pada gambar 3.7. Untuk menyimpan empat buah data dibutuhkan empat buah IC 7475 dimana terminal *enable*-nya dijadikan satu dan dihubungkan dengan unit kendali.



Gambar 3.7
Pengisian data pada memori

Untuk mengisi data pada memori, terminal *enable* pada memori ini harus berada dalam keadaan tinggi.

Pada pengisian data pada memori, isi memori akan berubah-ubah sesuai dengan data keluaran pada masing-masing memori masukan. Sebagai contoh, akan diperlihatkan pengisian data pada memori dengan pemasukan data 0011, 0101, 0010 dan 1001.

Pada penekanan tombol I, isi *D-Latch* I adalah 0011 sesuai dengan isi register geser I. Sedangkan isi *D-Latch* II, III dan IV adalah 0000.

Pada penekanan tombol II, isi *D-Latch* I adalah 0101 dan isi *D-Latch* II adalah 0011. Sedangkan isi *D-Latch* III dan IV adalah 0000.

Pada penekanan tombol III, isi *D-Latch* I adalah 0010, isi *D-Latch* II adalah 0010, dan isi *D-Latch* III adalah 0011. Sedangkan isi *D-Latch* IV adalah 0000.

Pada penekanan tombol IV, isi *D-Latch* I adalah 1001, *D-Latch* II adalah 0010, *D-Latch* III adalah 0101 dan *D-Latch* IV adalah 0011.

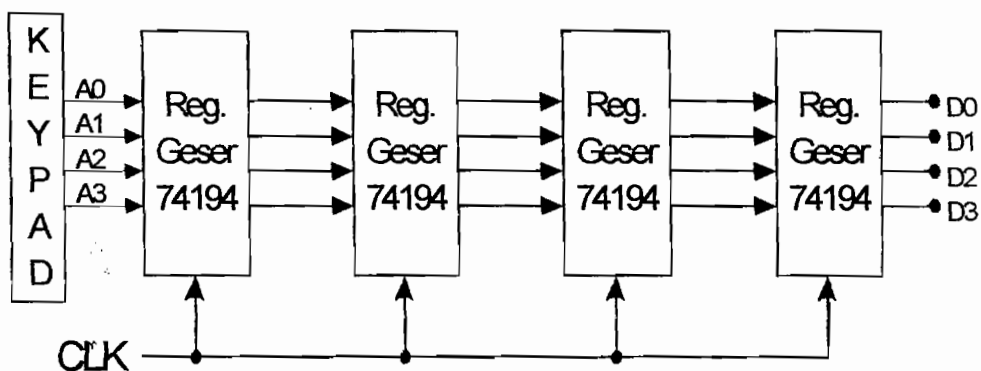
Dengan melihat contoh di atas, isi pada memori ini akan berubah sesuai dengan isi pada register geser sebagai memori masukan apabila terminal *enable* pada *D-Latch* berada pada keadaan tinggi. Apabila terminal *enable* berada pada keadaan rendah, isi pada memori ini tidak akan berubah walaupun data pada register geser berubah.

3.5. Register Geser

Sebagai memori masukan menggunakan register geser *parallel in parallel out* yang didapatkan dari IC 74194. Untuk mendapatkan penyimpanan data sebanyak 4 buah angka, register geser ini harus dihubungkan secara seri sebanyak 4 buah

(lihat gambar 3.8). Terminal *clock* dari keempat register geser ini dijadikan satu dan dihubungkan dengan untai penghasil pulsa *clock* pada unit kendali.

Register geser ini pada dasarnya terbentuk oleh flip-flop D. Karena itu perubahan data atau pergeseran data akan terjadi pada saat pulsa *clock* berada pada keadaan tinggi. Register geser ini akan menggeser data ke arah register di sebelahnya setiap ada pulsa *clock*.



Gambar 3.8
Penyimpanan 4 data masukan

Misalnya data pada semua register adalah 0000. Pada saat tidak ada penekanan tombol pada *keypad*, tidak ada pulsa *clock* yang akan mengaktifkan register geser sehingga data pada semua register akan tetap berada pada keadaan semula yaitu 0000. Pada saat ada penekanan tombol pada *keypad*, muncul pulsa *clock* yang mengaktifkan register sehingga akan ada data yang dimasukkan pada register pertama sedangkan isi pada register II digeserkan ke register III, dan begitu seterusnya.

Sebagai contoh akan ditunjukkan pada saat penekanan tombol pada *Keypad* sebanyak empat kali atau pemasukan 4 buah data (3 5 2 9).

Penekanan pertama adalah tombol dengan simbol angka 3. Setelah tombol ini ditekan, data 0011 akan dimasukkan pada register geser I sehingga keluaran register ini akan menjadi 0011, dan isi pada register II, III dan IV adalah 0000.

Penekanan kedua adalah tombol dengan simbol angka 5. Setelah tombol ini ditekan, data 0101 akan dimasukkan pada register geser I dan isi register I sebelumnya digeserkan atau dipindahkan ke register II. Akibatnya keluaran register I adalah 0101, register II adalah 0011, register III dan IV adalah 0000.

Penekanan ketiga adalah tombol dengan simbol angka 2. Setelah tombol ini ditekan data 0010 akan dimasukkan pada register I sehingga keluaran register ini menjadi 0010. Sedangkan isi register I sebelumnya digeserkan ke register II dan seterusnya sehingga keluaran register II adalah 0101, register III adalah 0011 dan register IV adalah 0000.

Penekanan ke empat adalah tombol dengan simbol angka 9. Akibatnya keluaran register I adalah 1001, register II adalah 0010, register III adalah 0101 dan register IV adalah 0011.

Dengan melihat contoh di atas, terlihat bahwa data akan digeserkan satu kali atau digeserkan pada register selanjutnya setiap ada penekanan tombol pada *Keypad* yang akan menghasilkan sebuah pulsa *clock* yang dapat mengaktifkan register geser.

3.6. Penampil Tujuh Ruas

Penampil menggunakan penampil tujuh ruas katoda bersama. Penampil ini dibagi menjadi 2 bagian yaitu penampil pertama digunakan untuk menampilkan

data pada register geser atau memori masukan, dan penampil II digunakan untuk menampilkan data pada *D-Latch* atau memori data.

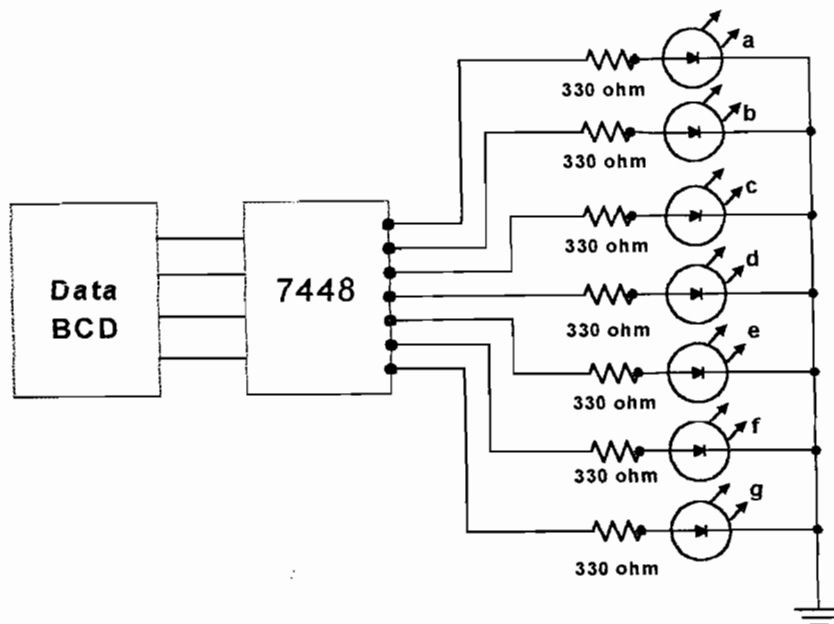
Data-data ini dapat langsung dihubungkan dengan *decoder BCD to Seven segment converter*, karena data tersebut sudah dalam bentuk bilangan BCD. Karena penampil yang dipakai adalah penampil tujuh ruas katoda bersama, maka dekoder yang digunakan adalah dekoder BCD to *Seven Segment converter* yang dikemas dalam bentuk IC 7448.

Decoder ini akan mengubah data masukan berupa bilangan BCD menjadi isyarat 7 jalur untuk menggerakkan masing-masing ruas. Isyarat 7 jalur ini tidak dapat langsung dihubungkan dengan LED karena dapat merusak LED. Karena itu isyarat 7 jalur ini terlebih dahulu harus dihubungkan dengan untai antarmuka LED yaitu berupa sebuah pembatas arus (lihat gambar 3.9). Pembatas arus dipasang di sini untuk menjaga agar dekoder tidak dipaksa untuk menghasilkan arus yang besar yang dapat merusak dekoder tersebut.

Suatu tampilan perlu mendapatkan arus sebesar 10 mA per ruas dan beroperasi dari rangkaian TTL 5 V. Dengan demikian, tegangan untuk logika 1 adalah 5 V dengan kehilangan tegangan maju pada ruas sebesar 2 V. Karena itu arus 30 mA ini akan dihasilkan dari tegangan 3V sehingga resistor yang diperlukan menurut hukum Ohm adalah :

$$\begin{aligned} R &= V / I \\ &= 3 \text{ V} / 0,01 \text{ A} \\ &= 300 \Omega \end{aligned}$$

Untuk lebih baiknya, digunakan tahanan sebesar 330 Ω .



Gambar 3.9
Pembatas arus pada LED

BAB IV

ANALISIS DAN PEMBAHASAN

Pada bab ini akan dipaparkan mengenai analisis dan pembahasan hasil perancangan perangkat keras, antara lain mengenai *keypad*, rangkaian penghasil pulsa *clock*, register geser, pembanding, rangkaian pengganti isi memori dan pengaktifan kunci serta rangkaian keseluruhan.

4.1. Analisis Rangkaian Keypad

Untuk mengetahui tabel kebenaran dari rangkaian *keypad* ini, terlebih dahulu lihat gambar rangkaian keypad pada lampiran.

Tabel 4.1

Tabel Kebenaran Rangkaian Keypad

Tombol	Keluaran Gerbang											Data Biner
	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	
-	0	0	0	0	0	0	0	0	0	0	0	0000
1	1	0	1	0	1	0	0	0	1	0	0	0001
2	0	1	0	0	0	1	0	1	0	0	0	0010
3	1	1	1	0	1	1	0	1	1	0	0	0011
4	0	0	0	1	0	0	1	0	0	1	0	0100
5	0	0	1	1	1	0	1	0	1	1	0	0101
6	0	0	0	0	0	1	1	1	0	1	0	0110
7	0	0	0	0	1	0	0	1	1	1	0	0111
8	0	0	0	0	0	0	0	0	0	0	1	1000
9	0	0	0	0	0	0	0	0	1	0	1	1001
0	0	0	0	0	0	0	0	0	0	0	0	0000
*	0	0	0	0	0	0	0	0	0	0	0	0000
#	0	0	0	0	0	0	0	0	0	0	0	0000

Dengan melihat hasil yang ditunjukkan pada tabel 4.1, diketahui bahwa data yang didapat pada perancangan adalah sesuai dengan teori yaitu keluaran gerbang G1 sampai G11 yang merupakan suatu gerbang OR akan menghasilkan keluaran bernilai tinggi (1) apabila salah satu atau semua masukan dari gerbang OR ini dalam keadaan tinggi, tetapi akan menghasilkan keluaran bernilai rendah (0) apabila semua masukan bernilai rendah. Pada perancangan *keypad* ini, tegangan masukan pada gerbang adalah sebesar 4 V pada keadaan tinggi dan 0 V pada keadaan rendah. Dan keluaran yang dihasilkan oleh gerbang pada keadaan rendah adalah 0 V dan 4 V pada keadaan tinggi. Keadaan-keadaan ini sudah melewati keadaan terburuk yang dapat mengakibatkan kesalahan.

4.2. Analisis Rangkaian Pembentuk Pulsa

Untuk mengetahui tabel kebenaran dari rangkaian penghasil pulsa, terlebih dahulu lihat gambar 3.3.

Tabel 4.2

Tabel Kebenaran Rangkaian Pembentuk Pulsa

Tombol										Pulsa Clock		
	D	C	B	A	G1	G2	G3	G4	G5		G6	G7
-	0	0	0	0	1	1	1	1	0	1	0	-----
1	0	0	0	1	1	1	1	0	1	0	1	Ada
2	0	0	1	0	1	1	0	1	1	0	1	Ada
3	0	0	1	1	1	1	0	0	1	0	1	Ada
4	0	1	0	0	1	0	1	1	1	0	1	Ada
5	0	1	0	1	1	0	1	0	1	0	1	Ada
6	0	1	1	0	1	0	0	1	1	0	1	Ada
7	0	1	1	1	1	0	0	0	1	0	1	Ada
8	1	0	0	0	0	1	1	1	1	0	1	Ada
9	1	0	0	1	0	1	1	0	1	0	1	Ada
0	0	0	0	0	1	1	1	1	0	0	1	Ada
*	0	0	0	0	1	1	1	1	0	0	0	-----
#	0	0	0	0	1	1	1	1	0	0	0	-----

Dari data pengamatan di atas dapat dilihat bahwa keluaran dari gerbang G1, G2, G3, G4, G6 dan G7 yang merupakan gerbang NOT adalah invers atau kebalikan dari masukan dan keluaran gerbang G5 yang merupakan sebuah gerbang NAND 4 masukan adalah tinggi bila keempat masukan dalam keadaan rendah. Hal ini sesuai dengan perhitungan secara teori.

Dengan melihat data pengamatan pada tabel 4.2, terlihat bahwa pulsa *clock* akan dihasilkan apabila keluaran gerbang G5 bernilai 1. Tetapi hal ini ada pengecualian pada saat penekanan tombol "0". Walaupun keluaran gerbang G5 adalah 0, tetapi pulsa *clock* juga dihasilkan. Hal ini disebabkan karena tombol "0" pada *keypad* dihubungkan secara langsung pada gerbang NOT G6 pada rangkaian penghasil pulsa.

Rangkaian penghasil pulsa menggunakan tambahan dioda sebagai pembatas arah arus agar tidak ada arus yang masuk menuju terminal masukan. Katoda dari dioda ini dihubungkan dengan *ground* melalui sebuah resistor untuk menghilangkan tegangan sebesar 1,2 V pada keadaan rendah akibat keluaran dioda yang menjadi mengambang.

4.3. Analisis Register Geser

Pada pengamatan register geser ini, akan dimasukkan 4 buah data yang dihasilkan oleh penekanan tombol 1 sampai 4 (lihat gambar 3.8) dengan data yang dihasilkan adalah 0000, 0001, 0010 dan 0100.

Dari hasil pengamatan pada perancangan yang ditunjukkan pada tabel 4.3, terlihat bahwa untuk memasukkan data hingga keseluruhan register geser terisi adalah dibutuhkan 4 buah pulsa *clock*. Data akan digeserkan satu kali setiap ada

satu pulsa *clock* yang masuk pada terminal *clock*. Pergeseran data pada keempat register geser ini bersifat langsung (*sinkron*), karena semua terminal *clock* pada keempat register geser ini dihubungkan menjadi satu dan mendapat pulsa *clock* yang sama. Hal ini sesuai dengan teori register geser yaitu data akan digeserkan satu kali setiap ada pulsa *clock* yang masuk pada terminal *clock*. Pergeseran data ini akan terjadi pada saat pulsa *clock* berada pada keadaan transisi perubahan tegangan dari kondisi rendah menuju kondisi tinggi atau lebih dikenal dengan sebutan aktif pinggiran positif.

Tabel 4.3

Tabel Kebenaran Register Geser

Clock Ke-	Data	Reg. Ges.I	Reg. Ges II	Reg. Ges. III	Reg. Ges.IV
	DCBA	DCBA	DCBA	DCBA	DCBA
0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
1	0 0 0 1	0 0 0 1	0 0 0 0	0 0 0 0	0 0 0 0
2	0 0 1 0	0 0 1 0	0 0 0 1	0 0 0 0	0 0 0 0
3	0 0 1 1	0 0 1 1	0 0 1 0	0 0 0 1	0 0 0 0
4	0 1 0 0	0 1 0 0	0 0 1 1	0 0 1 0	0 0 0 1

4.4. Analisis Pembandingan 4 Bit

Pada pengamatan pembandingan 4 bit ini, dimisalkan data pada memori adalah 0101, 1001, 0000 dan 0010. Sedangkan data masukan pada register geser adalah beberapa macam kombinasi 4 buah data (lihat gambar 3.6).

Pembandingan akan membandingkan data biner 4 bit dengan cara membandingkan data bit per bit, sehingga bila salah satu bit berbeda dengan bit lain yang dibandingkan, maka data yang dibandingkan tersebut dikatakan tidak sama. Yang menentukan hasil akhir pembandingan 4 buah data tersebut adalah

keadaan pembanding ke-4. Apabila keadaan pembanding ke-4 ini bernilai 1, data yang dibandingkan adalah sama, sebaliknya bila keadaan pembanding ke-4 adalah bernilai 0, maka data yang dibandingkan adalah tidak sama.

Tabel 4.4

Contoh kondisi pembanding

Saat membandingkan 4 data biner 4 bit

Register Geser Ke-				Memori Ke-				Pembanding Ke-			
1	2	3	4	1	2	3	4	1	2	3	4
DCBA	DCBA	DCBA	DCBA	DCBA	DCBA	DCBA	DCBA				
0100	0101	0010	0010	0101	1001	0000	0010	0	0	0	0
1000	0001	0000	1001	0101	1001	0000	0010	0	0	1	0
1001	1001	0011	0010	0101	1001	0000	0010	0	1	0	0
0111	1001	0000	0011	0101	1001	0000	0010	0	1	1	0
0101	0011	0101	0000	0101	1001	0000	0010	1	0	0	0
0101	0000	0000	1001	0101	1001	0000	0010	1	0	1	0
0101	1001	1001	0010	0101	1001	0000	0010	1	1	0	0
0101	1001	0000	0010	0101	1001	0000	0010	1	1	1	1

Dengan melihat hasil pengamatan pada tabel 4.4, terlihat bahwa bila hasil pembandingan salah satu pembanding menyatakan bahwa data yang dibandingkan tidak sama, maka pada pembanding berikutnya walaupun data yang dibandingkan sama, hasil pembandingan tetap dianggap tidak sama. Hal ini berlaku juga pada pembanding ke-4, walaupun pembanding ke-4 ini yang menentukan hasil pembandingan 4 buah data biner 4 bit. Hal ini disebabkan karena terminal masukan A=B pada pembanding, sebenarnya dihubungkan dengan sebuah gerbang AND pada terminal keluaran A=B (lihat gambar pada lampiran). Jika masukan A=B berada pada keadaan rendah (0), maka gerbang AND pada terminal

keluaran $A=B$ akan menghasilkan keadaan rendah sehingga keluaran $A=B$ juga berada dalam keadaan rendah.

4.5. Analisis Rangkaian Pengganti Isi Memori dan Pengaktifan Kunci

Tabel 4.5

Tabel Pengamatan

Rangkaian pengganti isi memori dan pengaktifan kunci

Pembanding Ke-4	Tombol		Gerbang		Keluaran Pencacah Ke-	
	*	#	G1	G2	1	2
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	0	0	0
1	0	0	0	0	0	0
1	0	1	0	1	0	1
1	1	0	1	0	1	0
1	1	1	1	1	1	1

Dari hasil pengamatan yang ditunjukkan oleh tabel 4.5, terlihat bahwa yang berperan penting untuk penggantian isi memori yang ditunjukkan oleh pencacah ke-1 dan pengaktifan kunci yang ditunjukkan oleh pencacah ke-2 adalah keadaan pembanding ke-4 atau keadaan $A=B$ (lihat gambar 3.5), sebab jika keadaan Pembanding ke-4 ini dalam keadaan rendah, rangkaian ini tidak berfungsi karena gerbang AND G1 dan G2 tidak akan terbuka.

Keluaran dari gerbang AND G1 dan G2 sangat rendah pada kondisi tinggi yaitu hanya sebesar 1,8 volt. Tegangan sebesar ini kurang dari keadaan terburuk sebagai keluaran keadaan tinggi yaitu 3 Volt, sehingga tidak dapat mengaktifkan

pencacah ke-1 dan ke-2. Karena itu keluaran tersebut harus dihubungkan dengan sebuah penguat.

4.6. Analisis Rangkaian Kunci Elektronis

Pada pengamatan ini, isi memori mula-mula adalah 0001, 0010, 0011 dan 0100. Setelah itu isi memori akan diganti dengan data 1000, 0111, 0110 dan 0101 (lihat lampiran).

Dengan melihat data pengamatan pada tabel 4.6, terlihat bahwa perbandingan 4 buah data biner 4 bit dikatakan sama apabila hasil perbandingan pada pembanding ke-1 sampai dengan pembanding ke-4 adalah bernilai 1. Dengan hasil perbandingan yang sama, penggantian isi memori dan pengaktifan kunci dapat dilakukan. Bila keluaran pencacah JK-1 dalam keadaan tinggi, data pada memori akan berubah-ubah sesuai dengan data masukan pada register geser, sebaliknya bila keluaran pencacah JK-1 adalah 0, maka isi memori tidak akan berubah walaupun data pada register geser berubah-ubah.

Pada saat perbandingan 4 buah data adalah sama, maka apabila terjadi penekanan tombol * ataupun # sebanyak 2 kali atau kelipatan 2, maka keluaran pencacah JK-1 dan JK-2 tidak akan aktif atau kembali ke keadaan 0. Hal ini disebabkan karena bagian ini menggunakan pencacah bermodulo 2.

Tabel 4.6
Tabel Kebenaran Rangkaian Kunci Elektronik
Berbasis Digital

Tombol	Data	Clock	Register Geser Ke-				Memori Ke-				Komparator Ke-				Tombol		Ganti PIN	Aktif Kunci
			1	2	3	4	1	2	3	4	1	2	3	4	*	#		
			DCBA	DCBA	DCBA	DCBA	DCBA	DCBA	DCBA	DCBA	DCBA	DCBA	DCBA	DCBA	DCBA	DCBA		
----	0000	----	0000	0000	0000	0000	0001	0010	0011	0100	0	0	0	0	0	0	0	0
1	0001	Ada	0001	0000	0000	0000	0001	0010	0011	0100	0	0	0	0	1	1	0	0
2	0010	Ada	0010	0001	0000	0000	0001	0010	0011	0100	1	0	0	0	0	0	0	0
3	0011	Ada	0011	0010	0001	0000	0001	0010	0011	0100	0	0	0	0	0	0	0	0
4	0100	Ada	0100	0011	0010	0001	0001	0010	0011	0100	0	0	0	0	0	0	0	0
3	0011	Ada	0011	0100	0011	0010	0001	0010	0011	0100	0	0	0	0	0	0	0	0
2	0010	Ada	0010	0011	0100	0011	0001	0010	0011	0100	0	0	0	0	0	0	0	0
1	0001	Ada	0001	0010	0011	0100	0001	0010	0011	0100	0	0	0	0	0	0	0	0
----	----	----	0001	0010	0011	0100	0001	0010	0011	0100	1	1	1	1	0	0	0	0
----	----	----	0001	0010	0011	0100	0001	0010	0011	0100	1	1	1	1	1	1	1	1
5	0101	Ada	0101	0001	0010	0011	0101	0001	0010	0011	1	1	1	1	0	1	1	0
6	0110	Ada	0110	0101	0001	0010	0110	0101	0001	0010	1	1	1	1	0	0	1	0
7	0111	Ada	0111	0110	0101	0001	0111	0110	0101	0001	1	1	1	1	0	0	1	0
8	1000	Ada	1000	0111	0110	0101	1000	0111	0110	0101	1	1	1	1	1	0	0	0
9	1001	Ada	1001	1000	0111	0110	1000	0111	0110	0101	0	0	0	0	1	1	0	0

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan analisis dan pembahasan yang telah dilakukan, maka dapat disimpulkan sebagai berikut :

- a. Kunci elektronik berbasis digital ini bekerja dengan masukan kombinasi 4 buah angka desimal
- b. Penggerendel-D (*D-Latch*) dapat digunakan sebagai memori selama masih ada catu daya. Untuk mencegah agar memori tidak hilang, D-Latch ini dihubungkan dengan catu daya tersendiri seperti batu baterai atau accu dengan tegangan 5 V. Data pada memori akan berubah sesuai dengan isi pada memori masukan selama terminal *enable* dalam keadaan tinggi
- c. Untuk mendapatkan perbandingan A=B dari IC 7485, terminal masukan A<B dan A>B dihubungkan dengan *ground*, sedangkan terminal masukan A=B dihubungkan dengan tegangan VCC. Apabila beberapa IC 7485 dihubungkan secara kaskade, hasil perbandingan akan dinyatakan sama bila keseluruhan data yang dibandingkan adalah sama
- d. Dalam memasukkan data, ada kemungkinan terjadinya pulsa *clock* yang lebih, dikarenakan mutu sakelar mekanik yang kurang baik

5.2. Saran

Saran yang dapat diberikan dalam tugas akhir ini adalah :

- a. Untuk memberi masukan dengan menggunakan sakelar mekanik sebaiknya dipilih sakelar mekanik yang baik mutunya dan menambahkan suatu rangkaian penghilang getaran pada masing-masing sakelar
- b. Agar penggantian bateray tidak terlalu sering, disarankan agar baterai yang digunakan adalah baterai yang tahan lama dan baik mutunya
- c. Bila ingin dibuat tanpa menggunakan baterai, dapat dipakai SRAM sebagai memori.

DAFTAR PUSTAKA

- Ibrahim, K. F., 1998, "Teknik Digital," Andi Offset, Yogyakarta
- Jacson, H. G, dan Hodges, D. A., 1987, "Analisis dan Desain Rangkaian Terpadu Digital," Erlangga, Jakarta
- Kleit, W., 1996, "Digital Elektronik," A Practical Approach, Prentice-Hall International Inc, Englewood Cliffs, New Jersey
- Malvino, L., 1992, "Prinsip-Prinsip dan Penerapan Digital," Erlangga, Jakarta
- Millman, J., 1986, "Mikro Elektronika," Erlangga, Jakarta
- Mano, M. M., 1979, "Digital Logic and Computer Design," Prentice-Hall Inc, Englewood Cliffs, New Jersey
- Sinclair, I. R., 1996, "Panduan Belajar Elektronika Digital," Elex Media Komputindo Jakarta

LAMPIRAN

IC TTL

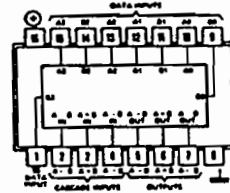
54/74 (LS)83 ... 85

Penjelasan

Penanding-penanding besar empat-bit ini melaksanakan penandingan sandi-sandi biner murni dan sandi BCD (8-4-2-1) murni. Dibentuklah tiga keputusan yang tersandi penuh dua kata @ 4-bit (A,B) dan dapat diperoleh secara ekstern di tiga jalankeluar. Peranti-peranti ini dapat sepenuhnya diperluas (*expandable*) sampai sebarang

85 Penanding besar 4-bit

	85	LS85	
supply current	55	10	mA
typ. delay (4 bit words)	23	24	ns



jumlah bit tanpa gerbang-gerbang ekstern. Kadakan penanding-penanding. Kata-kata dengan panjang lebih besar akan dapat ditan-tingkat yang menangani bit paling ringan dikoneksikan dengan mengkas-

kepada jalanmasuk-jalanmasuk $A>B$, $A<B$, dan $A=B$ yang berpadanan di tingkat berikutnya yang menangani bit berbobot lebih tinggi. Tingkat yang menangani bit paling ringan perlu dikenai tegangan bertaraf tinggi pada jalanmasuk $A=B$. Jalan-ari-jalan kaskade dari '85 dan LS85 dilaksanakan dengan tundaan hanya sepanjang dua-teraf gerbang guna mengecilkan waktu penan-dingan menyeluruh bagi kata yang panjang-panjang.

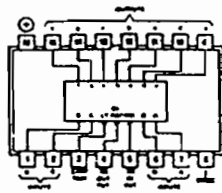
Tabel fungsi

Comparing Input				Cascading Input			Output		
A3, B3	A2, B2	A1, B1	A0, B0	A>B	A<B	A=B	A>B	A<B	A=B
A3 > B3	X	X	X	X	X	X	H	L	L
A3 < B3	X	X	X	X	X	X	L	H	L
A3 = B3	A2 > B2	X	X	X	X	X	H	L	L
A3 = B3	A2 < B2	X	X	X	X	X	L	H	L
A3 = B3	A2 = B2	A1 > B1	X	X	X	X	H	L	L
A3 = B3	A2 = B2	A1 < B1	X	X	X	X	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 > B0	X	X	X	H	L	L
A3 = B3	A2 = B2	A1 = B1	A0 < B0	X	X	X	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	H	L	L	H	L	L
A3 = B2	A2 = B2	A1 = B1	A0 = B0	L	H	L	L	H	L
A3 = B2	A2 = B2	A1 = B1	A0 = B0	L	L	H	L	L	H
A3 = B2	A2 = B2	A1 = B1	A0 = 3C	X	X	H	L	L	H
A3 = B2	A2 = B2	A1 = B1	A0 = B0	H	H	L	L	L	L
A3 = B2	A2 = B2	A1 = B1	A0 = B0	L	L	L	H	H	L

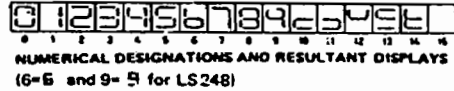
Jalanmasuk A, B, dan A=B memiliki isi-masukan 3.

IC TTL

54/74(LS)48



48, 248 Penggerak/dekoder BCD-7-segmen dengan resistor pengangkat (pull-up) 2 kΩ intern



Decimal or Function	Inputs				BI/RBO†	Outputs							note
	LT	RBI	D	C		B	A	a	b	c	d	e	
0	H	X	L	L	L	L	H	H	H	H	H	H	L
1	H	X	L	L	L	H	H	L	H	L	L	L	L
2	H	X	L	L	H	L	H	H	L	H	L	L	L
3	H	X	L	L	H	H	H	H	L	H	L	L	L
4	H	X	L	H	L	L	H	L	H	H	L	L	H
5	H	X	L	H	L	H	H	L	H	L	H	L	H
6	H	X	L	H	H	L	H	L	A/HOL	H	H	H	H
7	H	X	L	H	H	H	H	H	H	H	L	L	L
8	H	X	H	L	L	L	H	H	H	H	H	H	H
9	H	X	H	L	L	H	H	H	H	L	A/HOL	H	H
10	H	X	H	L	H	L	H	L	L	L	H	L	H
11	H	X	H	L	H	H	H	L	L	H	L	L	H
12	H	X	H	H	L	L	H	L	H	L	L	L	H
13	H	X	H	H	L	H	H	L	L	L	H	L	H
14	H	X	H	H	H	L	H	L	L	L	H	H	H
15	H	X	H	H	H	H	H	L	L	L	L	L	L
BI	X	X	X	X	X	X	L	L	L	L	L	L	L
RBI	H	L	L	L	L	L	L	L	L	L	L	L	L
LT	L	X	X	X	X	X	H	H	H	H	H	H	H

Function table
 * 48, LS48
 o LS248

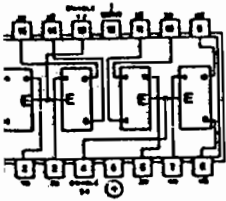
CATATAN:

- Jalanmasuk polosan (BI) harus terbuka atau berada dalam taraf logika tinggi kalau diinginkan fungsi-fungsi keluaran 0 sd 15. Jalanmasuk polosan kerut (RBI) harus terbuka atau tinggi kalau tidak diinginkan polosan nol dasan.
 - Kalau taraf logika rendah dikenakan secara langsung kepada jalanmasuk polosan (BI), maka semua jalankeluar segmen adalah rendah, tidak peduli akan taraf yang ada pada sesuatu jalanmasuk yang lain.
 - Kalau jalanmasuk polosan kerut (RBI) dan jalanmasuk A, B, C, dan D berada dalam taraf rendah dan jalanmasuk lamp test tinggi, maka semua jalankeluar segmen menjadi rendah dan jalankeluar polosan kerut (RBO) menuju ke taraf rendah (kondisi tanggap).
 - Kalau jalanmasuk/jalankeluar polosan kerut (BI/RBO) terbuka atau dipertahankan tinggi, dan kepada jalanmasuk lamp-test diberikan rendah maka semua jalankeluar segmen adalah tinggi.
- † BI/RBO adalah sebuah logika AND kawat (wire AND) berguna sebagai jalanmasuk polosan (BI) dan/atau jalankeluar polosan kerut (RBO).

ET: lamp test

standing input

		Condition	
		Fan-in	Fan-out
48	BI/RBO inp.	L	2,5
		H	1
248	other inp.	L/H	1
	BI/RBO outp.	L/H	5
	other outp.	L	4
		H	10
LS48	BI/RBO inp.	L	3
	H	1	
LS248	other inp.	L/H	1
	BI/RBO outp.	L	8
		H	2,5
	other outp.	L	15
H		5	



75 Gerendel (*latch*) dwimantap 4-bit

Tabel fungsi

(Each Latch)

INPUTS		OUTPUTS	
D	E	Q	\bar{Q}
L	H	L	H
H	H	H	L
x	L	Q_0	\bar{Q}_0

Q_0 = taraf Q sebelum E bertransisi tinggi-ke-rendah

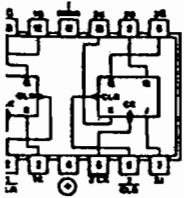
	75	LS75	
supply curr.	32	6,4	mA
t_E width min.	20	20	ns
tinp. set up min.	20	20	ns
tinp. hold min.	5	5	ns
t_{PLH}	16	15	ns
t_{PHL} E to Q	7	14	ns
t_{PLH}	16	16	ns
t_{PHL} E to \bar{Q}	7	7	ns

75 Jalanmasuk D mempunyai isi-muatan 2

75, LS75 Jalanmasuk memiliki isi-masukan 4

TTL

74(LS)73-74



73 Gulang-guling J-K berdua dengan clear

73
Tabel Fungsi

INPUTS				OUTPUTS	
CLEAR	CLOCK	J	K	Q	\bar{Q}
L	X	X	X	L	H
H	\nearrow	L	L	Q_0	\bar{Q}_0
H	\nearrow	H	L	H	L
H	\nearrow	L	H	L	H
H	\nearrow	H	H	TOGGLE	TOGGLE

LS73A
Tabel Fungsi

INPUTS				OUTPUTS	
CLEAR	CLOCK	J	K	Q	\bar{Q}
L	X	X	X	L	H
H	\downarrow	L	L	Q_0	\bar{Q}_0
H	\downarrow	H	L	H	L
H	\downarrow	L	H	L	H
H	\downarrow	H	H	TOGGLE	TOGGLE
H	H	X	X	Q_0	\bar{Q}_0

	73	LS73A	
supply curr.	20	4	mA
f_{ck} , max.	15	30	MHz
t_{ck} , width min.	20	20	ns
t_{input} set up min.	0 \uparrow	20 \downarrow	ns
t_{input} hold min.	0 \downarrow	0 \downarrow	ns
t_{PLH} CK to Q	16	15	ns
t_{PHL} or \bar{Q}	25	15	ns

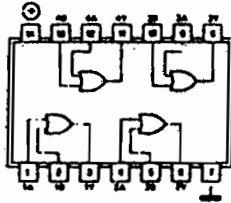
73 Jalanmasuk CLR dan CK memiliki isi-masukan 2
LS73A Jalanmasuk CLR dan CK, masing-masing memiliki isi-masukan 3, dan 4

IC TTL

54/74(LS)32...45

32 Gerbang OR 2-jalanmasuk berempat

$$Y = A + B$$

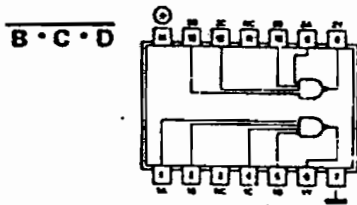


- 33:** lihat 02
- 37:** lihat 00
- 38:** lihat 00
- 40:** lihat 20

	supply curr. (mA)	t _{PLH} (ns)	t _{PHL} (ns)
32	19	10	14
LS32	4	14	14

IC TTL 54/74(LS)20...30

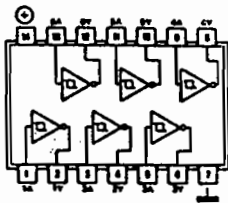
rangkai NAND 4-jalan masuk
 dengan jalankeluar kolektor
 terbuka
 $\text{fan-out} = 3 \times \text{fan-out standar}$



	supply curr. (mA)	t_{pLH} (ns)	t_{pHL} (ns)
20	4	12	8
LS 20	0,8	9	10
22	4	35	8
LS 22	0,8	17	15
40	10	13	8
LS 40	1,7	12	12

14 Penjungkir penyulut Schmitt berenam

$Y = \bar{A}$



	supply curr. (mA)	tpLH (ns)	tpHL (ns)
14	31	15	15
LS 14	10,3	15	15

	positive VT+ going tresh voltage (V)	negative VT- going tresh voltage (V)
14	1,7	0,9
LS 14	1,6	0,8

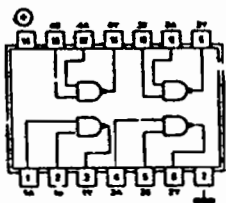
- 15:** lihat 11
- 16:** lihat 04
- 17:** lihat 07

TTL

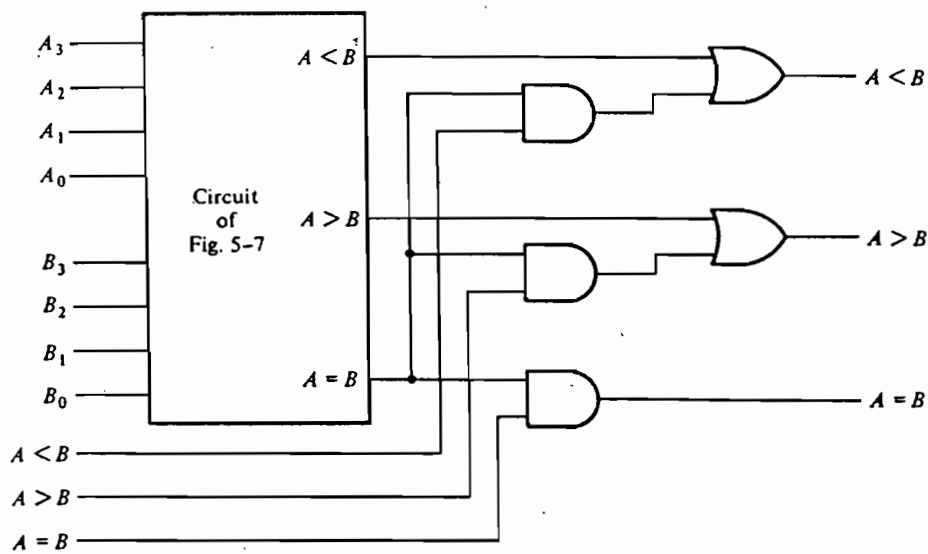
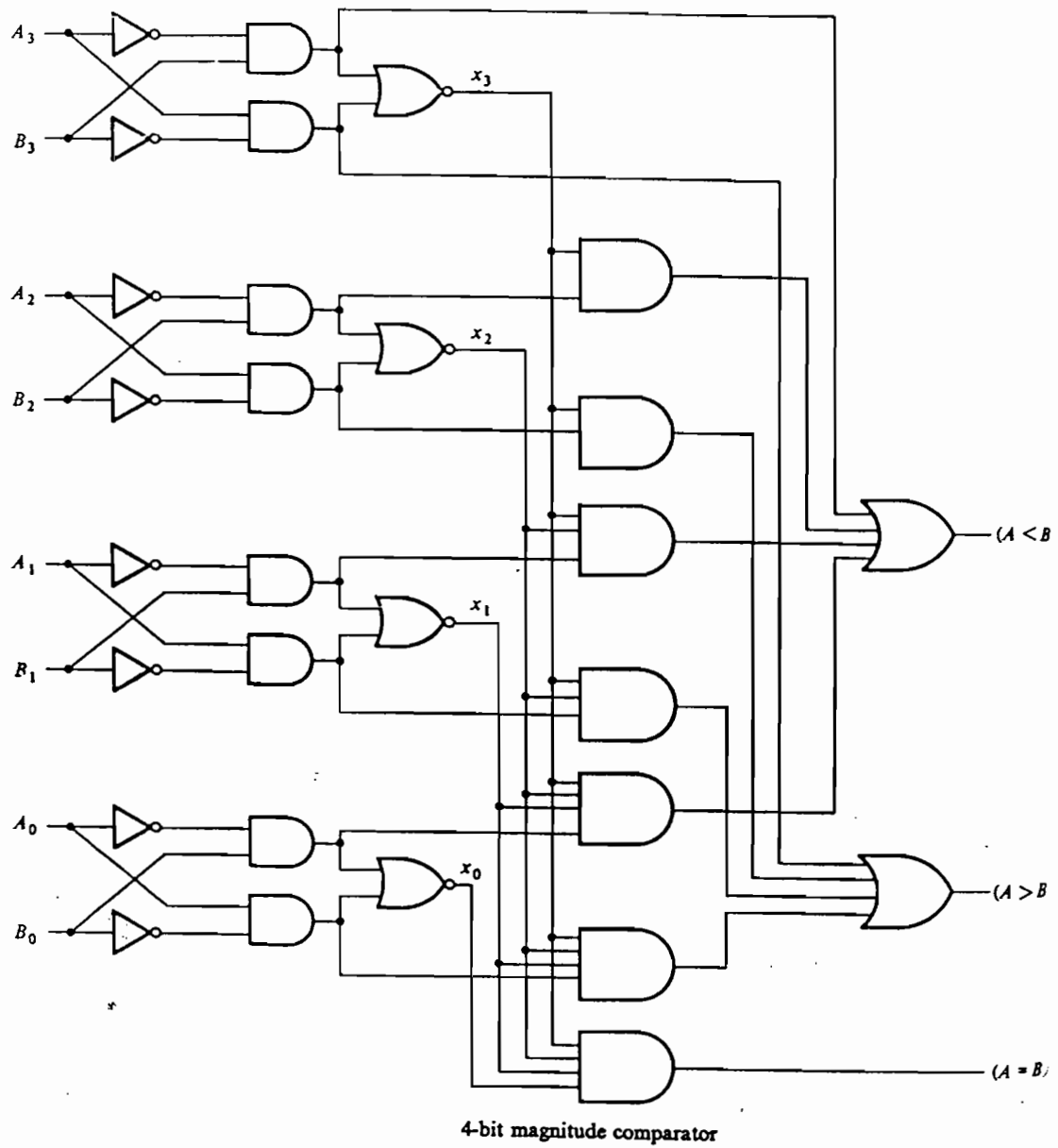
/74 (LS)00...03

Gerbang NAND 2-jalanmasuk berempat
 Dengan jalankeluar kolektor terbuka
 Dengan jalankeluar kolektor terbuka (15 V)
Fan-out = 3 × *fan-out* standar
 Dengan jalankeluar kolektor terbuka
fan-out = 3 × *fan-out* standar

A - B



	supply curr. (mA)	t _{PLH} (ns)	t _{PHL} (ns)
00	8	11	7
LS 00	1,6	9	10
03	8	35	8
LS 03	1,6	17	15
26	8	16	11
LS 26	1,6	17	15
37	21	13	8
LS 37	3,4	12	12
38	20	14	11
LS 38	3,4	20	18



Jalankeluar keempat gulang-guling disuiit oleh transisi rendah-ke-tinggi salah satu masukan cacah (lonceng). Arah cacahan ditentukan oleh jainmasuk manakah yang dimasuki denyut-denyut cacah, sementara jalanmasuk cacah yang lain adalah tinggi. Semua pencacah dapat diacarai sepenuhnya; artinya: setiap jalankeluar dapat distel-awal (preset) pada sebarang taraf dengan jalan memasukkan data yang diinginkan di jalanmasuk data sementara jalanmasuk load adalah rendah. Keluaran akan berubah berpadanan dengan masukan-masukan data, tidak bergantung pada denyut-denyut cacah. Sifat ini memungkinkan pencacah untuk

dipakai sebagai pembagi modul-N dengan jalan hanya mengubah panjang cacah dengan menggunakan jalanmasuk yang distel-awal. Jalanmasuk clear yang diberi taraf tinggi memaksakan semua jalankeluar ke taraf rendah. Fungsi clear adalah takbergantung pada jalanmasuk count dan jalanmasuk load. Jalanmasuk jalanmasuk clear, count, dan load, semuanya tersangguna merendahkan persyaratan penggerak. Hal ini menurunkan banyaknya penggerak lonceng dan lain sebagainya, yang diperlukan untuk kata-kata panjang. Pencacah-pencacah ini telah dirancang untuk dikaskadarkan tanpa memerlukan rang-

kaian ekstern. Jalankeluar-jalankeluar borrow dan carry dapat dipakai untuk mengkaskadkan fungsi-fungsi cacah mundur maupun maju. Jalankeluar borrow mengeluarkan denyut yang lebarnya sama dengan masukan cacahan mundur bila pencacah melimpah-turun. Demikian pula, jalankeluar carry mengeluarkan denyut yang lebarnya sama dengan masukan cacah maju bila ada kejadian limpahan-naik. Maka pencacah-pencacah pun akan dengan mudah dikaskadkan dengan jalan mengumpangkan keluaran-keluaran borrow dan carry ke jalanmasuk count-down atau count-up di pencacah berikutnya.

Lonceng terlarang (do-nothing)
Pengisian jajar sinkron dilakukan dengan mengenakan keempat bit data dan membikin tinggi kedua-dua kemudi ragam, S0 dan S1. Data pun dimasukkan ke gulang-guling yang bersesuaian dan muncul di jalankeluar sesudah transisi positif masukan lonceng. Selama pengisian, aliran data tercegah.
Geser kanan dilakukan secara sinkron dengan tepian-naik denyut lonceng bila S0 tinggi dan S1 rendah. Data deret untuk ragam ini dimasukkan di jalanmasuk data

H = taraf tinggi (status stasioner)
L = taraf rendah (status stasioner)
X = sebarang (sebarang masukan, termasuk transisi)
↑ = transisi dari taraf rendah ke tinggi
a, b, c, d = taraf status stasioner di jalanmasuk A, B, C, atau D
Q_{AO}, Q_{BO}, Q_{CO}, Q_{DO} = taraf Q_A, Q_B, Q_C, atau Q_D, sebelum terjadinya kondisi jalanmasuk status stasioner yang ditunjukkan
Q_{An}, Q_{Bn}, Q_{Cn}, Q_{Dn} = taraf Q_A, Q_B, Q_C atau Q_D sebelum transisi ↑ lonceng paling akhir.
geser kanan (shift-right input). Kalau S0 rendah dan S1 tinggi, maka data bergeser ke kiri secara sinkron, dan data baru dimasukkan di jalanmasuk shift-left serial. Peloncengan gulang-guling

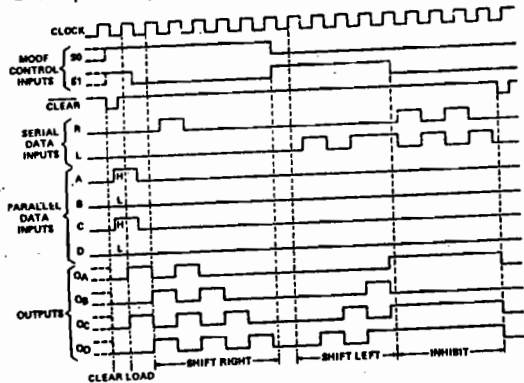
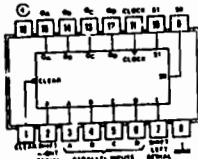
CLEAR	MODE		CLOCK	INPUTS SERIAL				PARALLEL				OUTPUTS			
	S1	S0		LEFT		RIGHT		A	B	C	D	Q _A	Q _B	Q _C	Q _D
				LEFT	RIGHT	LEFT	RIGHT								
L	X	X	X	X	X	X	X	X	X	X	L	L	L	L	
H	X	X	L	X	X	X	X	X	X	X	Q _{AO}	Q _{BO}	Q _{CO}	Q _{DO}	
H	H	H	↑	X	X	X	X	X	X	X	a	b	c	d	
H	L	H	↑	X	H	X	X	X	X	X	H	Q _{An}	Q _{Bn}	Q _{Cn}	
H	H	L	↑	X	L	X	X	X	X	X	L	Q _{An}	Q _{Bn}	Q _{Cn}	
H	H	L	↑	H	X	X	X	X	X	X	Q _{Bn}	Q _{Cn}	Q _{Dn}	H	
H	H	L	↑	L	X	X	X	X	X	X	Q _{Bn}	Q _{Cn}	Q _{Dn}	L	
H	L	L	X	X	X	X	X	X	X	X	Q _{AO}	Q _{BO}	Q _{CO}	Q _{DO}	

Penjelasan

Register-register geser dua arah ini telah dirancang untuk memenuhi semua sifat yang diharapkan dari register geser oleh seseorang perancang sistem. Rangkaiannya terdiri atas 46 gerbang yang setara dan memiliki jalanmasuk jajar, jalankeluar jajar, geser-kiri dan geser-kanan masuk deret, jalanmasuk kemudi ragam operasi, dan saluran clear langsung. Register memiliki empat ragam pengoperasian:
Masuk jajar
Geser kanan (arah Q_A menuju Q_D)
Geser kiri (arah Q_D menuju Q_A)

194 Register geser universal dua arah 4-bit

	supply curr. (mA)	f _{clock} max (MHz)
194	39	25
LS 194	15	25

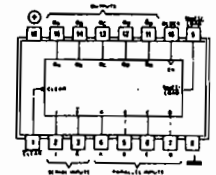


Penjelasan

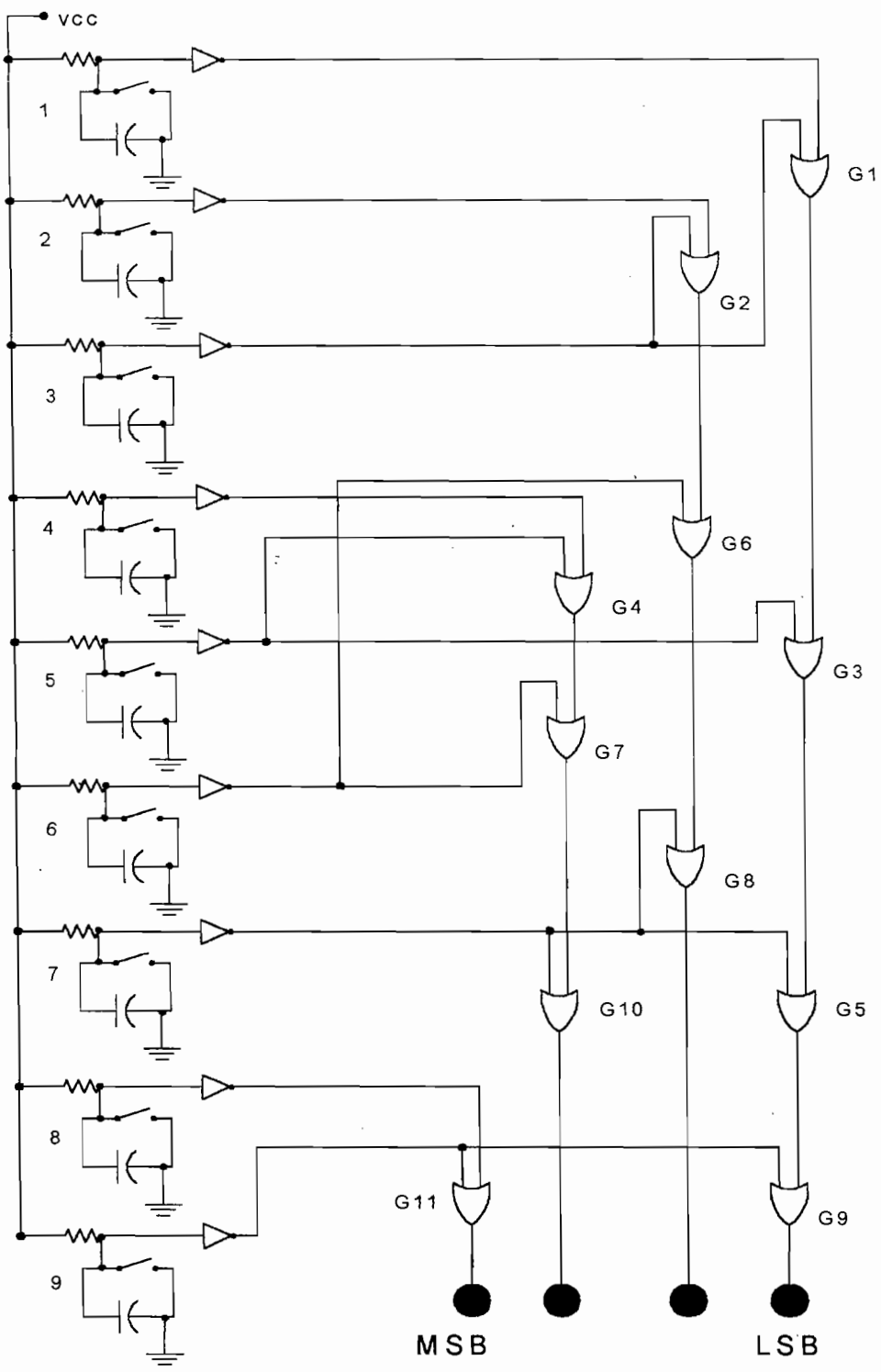
Register-register 4-bit ini memiliki masukan jajar, keluaran jajar, masukan deret J-K, masukan kemudi shift/load dan sebuah clear langsung. Semua jalanmasuk tersangga guna memperkecil persyaratan penggerak masukan. Register memiliki dua ragam operasi:
Masukan jajar
Geser (arah Q_A ke Q_D)
Pengisian jajar dilakukan dengan memasukkan keempat bit data dan membikin rendah jalanmasuk kemudi

195 Register geser tergapai-jajar, 4-bit
196: lihat 176
197: lihat 177

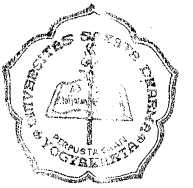
	supply curr. (mA)	f _{clock} max (MHz)
195	39	30
LS 195A	14	30



CLEAR	SHIFT/LOAD	CLOCK	SERIAL		PARALLEL				OUTPUTS			
			J	K	A	B	C	D	Q _A	Q _B	Q _C	Q _D
L	X	X	X	X	X	X	X	X	L	L	L	L
H	L	↑	X	X	a	b	c	d	a	b	c	d
H	H	↑	L	X	X	X	X	X	Q _{AO}	Q _{BO}	Q _{CO}	Q _{DO}
H	H	↑	L	H	X	X	X	X	H	Q _{An}	Q _{Bn}	Q _{Cn}
H	H	↑	L	L	X	X	X	X	L	Q _{An}	Q _{Bn}	Q _{Cn}
H	H	↑	H	H	X	X	X	X	H	Q _{An}	Q _{Bn}	Q _{Cn}
H	H	↑	H	L	X	X	X	X	L	Q _{An}	Q _{Bn}	Q _{Cn}



Rangkaian Keypad



95.5114024

