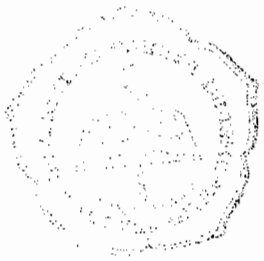
COUNTER UP ENAM DIGIT
BERBASIS MIKROKONTROLER AT89S51

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Teknik
Pada Program Studi Teknik Elektro



Disusun Oleh :

NAMA : BONEFASIUS OLDAM
NIM : 985114040

PROGRAM STUDI TEKNIK ELEKTRO
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS SANATA DHARMA
YOGYAKARTA
2006

SIX DIGITS COUNTER UP
BASED ON ATMEL AT89S51 MICROCONTROLLER

FINAL PROJECT

Presented as Partial Fulfillment of te Requirements
To Obtain The *Sarjana Teknik* Degree
in Electrical Engineering Study Program

By :

Name : BONEFASIUS OLDAM
Student Number: 985114040

ELECTRICAL ENGINEERING STUDY PROGRAM
ELECTRICAL ENGINEERING DEPARTMENT
ENGINEERING FACULTY
SANATA DHARMA UNIVERSITY
YOGYAKARTA
2006

TUGAS AKHIR

COUNTER UP ENAM DIGIT BERBASIS MIKROKONTROLER AT89S51

Disusun oleh :

BONEFASIUS OLDAM

NIM : 985114040

Telah disetujui oleh :

Dosen Pembimbing I

Ir. Iswanjono, M T Tanggal : 29-7-06

TUGAS AKHIR

COUNTER UP ENAM DIGIT BERBASIS MIKROKONTROLER AT89S51

Disusun oleh :

BONEFASIUS OLDAM

NIM : 985114040

Telah dipertahankan di depan Panitia Penguji

Pada tanggal : 20 Juli 2006

dan dinyatakan memenuhi syarat.

Susunan Panitia Penguji :

	Nama Lengkap	Tanda Tangan
Ketua :	Martanto, ST, MT	.....
Sekretaris :	Ir. Iswanjono, M.T.	.....
Anggota :	Wiwien Widyastuti,ST, MT	.....

Yogyakarta, 29 Juli 2006

Fakultas Teknik

Universitas Sanata Dharma

Dekan Fakultas Teknik,

Ir.Greg. Heliarko SJ.,SS.,BST.,MA.,MSc

PERNYATAAN KEASLIAN KARYA

"Saya menyatakan dengan sesungguhnya bahwa tugas akhir yang saya tulis ini tidak memuat karya atau bagian karya orang lain, kecuali yang telah disebutkan dalam kutipan dan daftar pustaka, sebagaimana layaknya karya ilmiah"

Yogyakarta, 29 juli 2006

BONEFASIUS OLDAM

HALAMAN MOTTO DAN PERSEMBAHAN

"Kau mungkin saja kecewa jika percobaan mu gagal
Tetapi kau pasti takan berhasil jika tidak mencoba"

"Jika aku dapat meminta agar hidupku sempurna
Itu merupakan godaan mengiurkan
Namun aku akan terpaksa menolak,
Karena dengan begitu
Aku tidak dapat lagi menarik pelajaran dari kehidupan"

"kelemahan terbesar dari kebanyakan
manusia adalah keseganan untuk
menyatakan pada orang lain betapa mereka
menyayangi orang-orang itu
sewaktu mereka masih hidup"

KUPERSEMBAHKAN UNTUK :

- ***Yesus Kristus dan Bunda Maria,*** *Engkaulah andalanku*
- ***Papa Michael & Mama Khatrine*** *Tersayang, trima kasih atas dorongan dan doa kalian yang selalu mengiringi setiap langkahku hingga studiku dapat selesai*
- ***Ka' tince,ka' alex,ka' merry, ka' jose, ka' rocky,*** *dorongan dan bantuan dari kalian adalah semangatku*
- ***Dan semuanya yang spesial,*** *all of you are the best*
- ***Jeane Vianey***

KATA PENGANTAR

Puji syukur penulis haturkan kepada Bapa Yang Maha Kasih dan Bunda Maria atas penyertaannya serta limpahan kasihNya serta banyak pihak sehingga penulis dapat menyelesaikan tugas akhir ini. Karena itu dengan segenap kerendahan hati, penulis menyampaikan terima kasih yang setulusnya kepada :

1. Bapak Augustinus Bayu Primawan, S.T., M.Eng, selaku Ketua Jurusan Teknik Elektro Fakultas Teknik Universitas Sanata Dharma Yogyakarta.
2. Bapak Ir. Iswanjono, MT, selaku Dosen Pembimbing I yang telah berkenan meluangkan waktu guna memberikan bimbingan dan pengarahan.
3. Pak Djito yang penuh keramahan dan senyum, dosen-dosen dan karyawan-karyawan laboratorium TE yang sangat membantu penulis.
4. Yang terkasih saudara&keluargaku, ka' Albert, ka' Bella, ka' Doyok, ka' Indah. Keponakan-keponakan tersayang, Berry, Jerry, Grey, Endik, Fernando, Claudio, Dilivio, Titin.
5. Yang terbaik: Tatok&Lady, Dhany&Monic, Inok&Yane, Ndenk&Iponk, Pablo&Nane, Any&Dony, Tochan&Rolyn, Jelly&Yuck, Edward&Yanti, Christ, Yonas, Eman, Nong, Papa Lory, Om Tomy, Yolan, Tuti, Vivi, tanta Adel, Rio, Veny, Mandala Crew, Kimpur, Mbeak, Induk, Sagan Cs, Gaba, Kevin, Petung Cs, Janti Cs, Spesial 'tuk **Cole**, buat **'LABOTH'** dan semua yang terbaik yang tak bisa disebut satu persatu.
6. Dan semua pihak yang tidak bisa disebutkan satu persatu, terima kasih.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari sempurna, maka dengan segenap kerendahan hati penulis mengharapkan saran atau kritik yang bermanfaat. Semoga Tugas Akhir ini dapat berguna dan memberikan manfaat bagi semua pihak yang membutuhkan.

Yogyakarta, 29 Juli 2006

Penulis

COUNTER UP ENAM DIGIT
BERBASIS MIKROKONTROLER AT89S51

NAMA : BONEFASIUS OLDAM

NIM : 985114040

INTISARI

Penelitian ini dibuat untuk memberikan tampilan enam digit keluaran dari proses penghitungan kenaikan data *counter*.

Pada alat ini memiliki 3 piranti utama yaitu, piranti masukan yang terdiri dari matrix keypad 5x3, sebuah piranti pemroses mikrokontroler AT89S51 dan piranti keluarannya berupa tampilan *seven segment* enam digit. Pada awal dijalankan, akan muncul tampilan L O sebagai tanda meminta agar nilai awal dimasukkan, setelah enter ditekan muncul tampilan H I meminta nilai akhir dimasukkan. Setelah enter ditekan muncul J . untuk interval yang diinginkan, bila enter ditekan program berjalan otomatis dan bila manual ditekan program naik secara manual dengan kenaikan data setiap kali tombol manual ditekan.

S*even segment* enam digit digunakan sebagai penampil keluaran yang menampilkan waktu dalam cacahan detik dengan *error* 0,1%.

Kata Kunci : *Counter Up* Enam Digit, Matrix *keypad,* Aplikasi Mikrokontroler MCS 51

SIX DIGITS COUNTER UP
BASED ON ATMEL AT89S51 MICROCONTROLLER

NAME : BONEFASIUS OLDAM

NIM : 985114040

ABSTRACT

This research is made in order to give an appearance that is an output in the six digits form from an enumerating process of increasing *counter* data.

This instrument has three main apparantuses namely, an input apparatus which consists of *keypad matrix* 5x3, a microcontroller processing apparatus AT89S51, and an output apparatus in the form of *seven segments* appearance of six digits. When running the apparantus, L O appearance will emerge as a sign to ask the initial value to be entered. After pressing *enter* button, H I appearance emerges to ask final value. After pressing *enter* button, J emerges. For the wanted interval, if the *enter* button is pressed, then the program runs automatically. On the contrary, if the manual button is pressed, then the program increases manually where the data increases every time manual button pressed.

The seven segments of six digit used as the output appearance which presents time in the form of second digital where the error is 0,1 %.

Keywords: *Counter Up* six digit, *keypad matrix*, Application of Microcontroller

DAFTAR ISI

halaman

Halaman judul i
Halaman Persetujuan iii
Halaman Pengesahan iv
Halaman Keaslian Karya v
Halaman Motto dan persembahan vi
Kata Pengantar vii
Intisari viii
Abstract ix
Daftar Isi x
Daftar Gambar xiii
Daftar Tabel xiv
Daftar Lampiran xv

BAB I PENDAHULUAN

1.1. Judul 1
1.2. Latar Belakang Masalah 1
1.3. Tujuan dan Manfaat Penelitian 2
1.4. Batasan Masalah 2
1.5. Metodologi Penelitian 2

BAB II DASAR TEORI

2.1. *Counter Up* Enam Digit 3
2.2. Masukan atau Pirantri Input 3
2.3. Mikrokontroler AT89S51 6
2.3.1. *Timer* dan *Counter* dalam Mikrokontroler AT89S51 6
2.3.1.a. *Timer* Mode Register (**TMOD**) 7
2.3.1.b. *Timer Control Register Timer 0 dan 1* 8

2.4. Piranti Output 9
2.4.1. LED 9
2.4.2. Transistor 10
2.4.3. Penampil *Seven Segment* 10

BAB III PERANCANGAN ALAT

3.1. Perancangan Perangkat Keras 12
3.1.1. Rangkaian *keypad* dan Mikrokontroler AT89S51 13
3.1.2. Mikrokontroler AT89S51 16
3.1.2.a. Rangkaian Reset 16
3.1.2.b. Rangkaian Osilator 17
3.1.3. Unit Penampil 17
3.1.3.a. Dioda Pemancar cahaya 17
3.1.3.b. Rangkaian penampil dengan *seven segment* 18
3.2. Perancangan Perangkat Lunak 22
3.2.1. Diagram Alir Program Utama 22
3.2.2. Diagram Alir Subrutin *keypad* 24
3.2.3. Diagram Alir memasukkan data 26

BAB IV HASIL DAN PEMBAHASAN

4.1 Hasil Akhir Perancangan 28
4.2 Listing Program 28
4.2.1. Listing Program Utama 28
4.2.2. Listing Program Keypad 31
4.3. Pembahasan 34
4.3.1. Cara Kerja Alat 34
4.3.2. Data Penggunaan Alat 35

BAB V KESIMPULAN DAN SARAN

5.1 kesimpulan 38
5.2 Saran 39

DAFTAR PUSTAKA 38

LAMPIRAN

DAFTAR GAMBAR

Halaman

Gambar 2.1 Matrix *keypad* 5x3 4
Gambar 2.2. Rangkaian tombol *push-on* dengan *Rpull-up* 5
Gambar 2.3. Efek *bouncing* 5
Gambar 2.4. Register TMOD 7
Gambar 2.5. Register TCON 8
Gambar 2.6. Rangkaian LED 10
Gambar 2.7. Bentuk Tampilan *Seven Segment* 11
Gambar 3.1. Diagram blok *Counter Up* Enam Digit 12
Gambar 3.2. Rangkaian *interface matrix keypad 5x3* 13
Gambar 3.3. Rangkaian reset 16
Gambar 3.4. Rangkaian osilator 17
Gambar 3.5. Rangkaian indikator LED 18
Gambar 3.6. Rangkaian Penampil *seven segment* 19
Gambar 3.7. Skema Dasar Konfigurasi Saklar Menggunakan Transistor 20
Gambar 3.8. Diagram alir program utama 23
Gambar 3.9. Diagram alir dari subroutine *keypad* 25
Gambar 3.10. Diagram alir memasukkan data 26

DAFTAR TABEL

Halaman

Tabel 2.1. Mode operasi pemilih *Timer/counter* AT89S51 8

Tabel 2.2. Tabel Kebenaran Seven Segment..... 11

Tabel 3.1. Tombol *keypad* dan fungsinya..... 11

Tabel 4.1. Tabel data penggunaan alat..... 29

DAFTAR LAMPIRAN

Lampiran 1	Gambar Rangkaian
Lampiran 2	*Listing* Program
Lampiran 3	*Data Sheet* Penguat Operasi Daya Rendah Berempat
Lampiran 4	*Data Sheet* Mikrokontroler AT89S51
Lampiran 5	*Data Sheet* Transistor A733
Lampiran 6	*Data Sheet* 7 *Segment* LED *Display*

BAB I

PENDAHULUAN

1.1. Judul

Counter Up Enam Digit Berbasis Mikrokontroler AT89S51

1.2. Latar Belakang Masalah

Mikrokontroler banyak digunakan dalam program-program tertentu, terutama dalam mengontrol suatu sistem kerja secara otomatis.

Begitupun dalam pengerjaan sistem *counter up* enam digit. Mikrokontroler digunakan sebagai pengontrol utama proses. *Counter up* banyak dipakai dalam berbagai hal misalnya, sebagai pencatat angka dalam pertandingan olah raga.

Secara khusus menyangkut penggunaan interval kenaikan, *counter up* enam digit biasa digunakan sebagai penghitung penambahan jumlah barang yang berjumlah banyak sampai dengan hitungan ratusan ribu.

Dengan masukan berupa *keypad*, sangatlah mudah mengetahui penambahan jumlah barang yang masuk tanpa harus dihitung dengan jari ataupun dengan bantuan kalkulator atau alat penghitung terbatas lainnya.

1.3. Tujuan Dan Manfaat Penelitian

- Merancang suatu sistem *counter up* dengan input terprogram berbasis mikrokontroler AT89S51.
- Menanggulangi kesalahan dalam perhitungan secara manual.

1.4. Batasan Masalah

Dalam pengerjaannya alat ini memiliki batasan interval sebagai berikut:

1. Interval minimum : 0 Interval maksimum : 9
2. Nilai awal minimum :0 Nilai awal maksimum : 900.000
3. Nilai ahlir minimum : 0 Nilai ahkir maksimum : 999999

1.5. Metodologi Penelitian

Dalam proses pembuatan alat ini, penelitian yang digunakan dengan studi literature, pengujian laboratorium, simulasi program pada komputer dengan *software*.

BAB II

DASAR TEORI

2.1. *Counter Up* Enam Digit

Aplikasi ini digunakan sebagai proses perhitungan untuk penambahan-penambahan dalam jumlah yang banyak. Pencacahan dimulai dari 0 sampai batas maksimal 999999. Selain itu alat ini dapat diset agar pencacahan dimulai dari nilai tertentu sebagai nilai awal, sampai dengan nilai akhir yang ditentukan melalui *keypad.*

Alat ini dapat diset agar mengulang proses perhitungan sampai mencapai batas maksimal yang ditentukan.

2.2. Masukan atau Piranti Input

Masukkan yang dimaksud adalah proses awal pengoperasian dari pengerjaan pencacah naik. Dimana dalam hal ini masukkan melalui *matrix keypad* 5×3 yang di dalamnya terdapat tombol angka 0 sampai angka 9, *ok, clear, enter.*

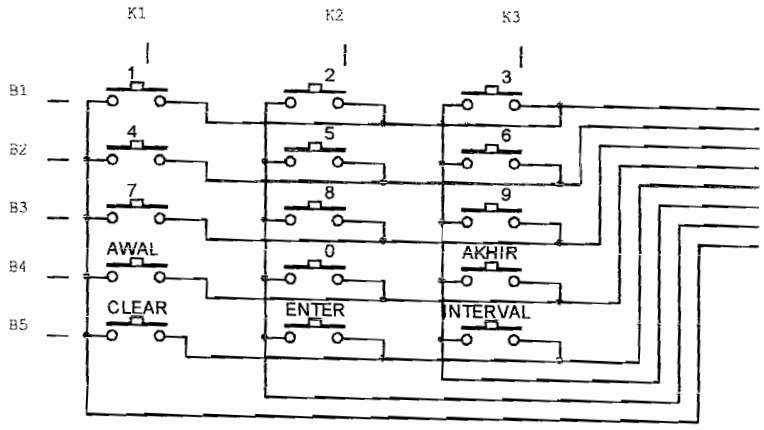
Matrix keypad

Matrix keypad adalah *keypad* yang saklarnya disusun sebagai sarana masukkan ke mikrokontroler, terdiri dari 5 baris dan 3 kolom.

Meskipun jumlah tombol ada 15 tetapi hanya memerlukan 8 jalur, dimana 5 jalur untuk baris dan 3 jalur untuk kolom.

Pembacaan *matrix keypad* dilakukan dengan cara *scanning proses*, yaitu apabila tombol ditekan maka piranti akan menghubungkan suatu baris dengan suatu kolom.

Pada gambar 2.1. terdapat 15 tombol yang digunakan sebagai tombol masukan untuk pengisian data yang dibutuhkan. Masing-masing tombol mempunyai fungsi tersendiri sesuai dengan kebutuhan yang diinginkan. Misalnya untuk memasukan angka 2 maka tombol 2 pada baris 1 kolom 2 yang ditekan,begitukan untuk penggunaan tombol yang lainnya sesuai dengan yang tertera pada gambar 2.1.



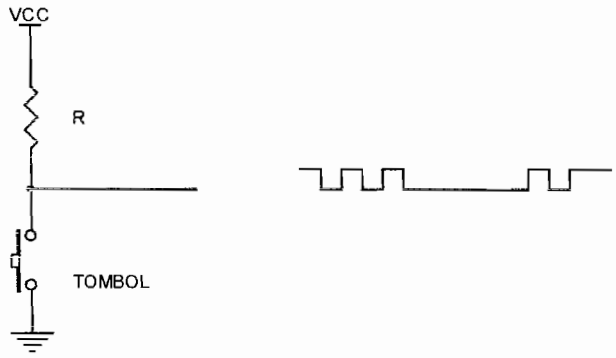
Gambar 2.1. Matrix keypad 5×3

Pada gambar 2.2. rangkaian tombol *push-on* dengan *Rpull-up* akan menghasilkan efek *bounching* karena tombol ditekan akan menghasilkan getaran,

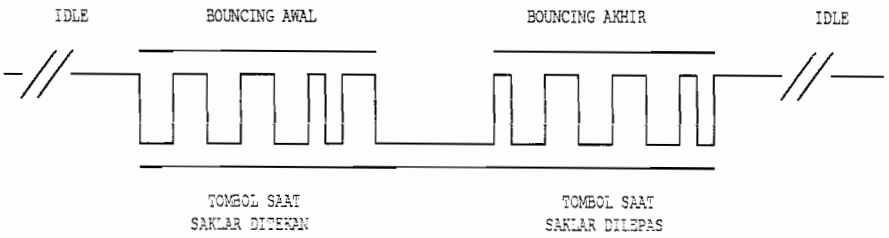
atau sebelum mencapai keadaan stabil tombol tersebut akan ON atau OFF. Sifat ini mengakibatkan pembacaan tombol merasakan adanya penekanan tombol yang berulang-ulang, meskipun pada dasarnya tombol hanya ditekan sekali saja.

Diagram waktu guncangan (bouncing) pada tombol sewaktu ditekan maupun pada saat dilepas ditunjukkan pada gambar 2.3.

Gambar 2.2 memperlihatkan rangkaian tombol *push-on*



Gambar 2.2. Rangkaian tombol *push-on* dengan *Rpull-up*



Gambar 2.3. Efek *Bouncing* pada saat penekanan tombol

Resistor pembatas arus berfungsi agar pada saat sebuah tombol ditekan dua pin *port* 0 yang terhalang pada tombol tersebut tidak sepenuhnya terhubung singkat. Dengan demikian saat *port* 0 berfungsi sebagai *output* untuk mengirim data ke unit penampil, data dapat dikirim tanpa cacat walaupun terdapat tombol yang ditekan. Karena arus maksimal yang keluar dari *port 0* sebesar 26mA, maka Rmin sebesar:

$$\text{Rmin} = \frac{Vcc}{IoL} = \frac{5Volt}{26mA} = 192{,}3\Omega$$

Di sini digunakan Rpack sebesar 10K (*data sheet)*, sehingga akan menghasilkan arus sebesar 0,5mA. Dimana arus ini lebih kecil dari arus maksimal.

2.3. Mikrokontroler AT89S51

Mikrokontroler AT89S51 merupakan mikrokontroler buatan *Atmel* yang menguasai teknologi pembuatan FPEROM (*Flash Programmable and Erasable Read Only Memory*). FPEROM merupakan ROM (*Read Only Memory*) yang dapat dihapus dan ditulis kembali Kelebihan *flash* ini adalah mikrokontroler dapat menyimpan program secara internal, tidak membutuhkan ROM *eksternal*.

AT89S51 memiliki 4 *kBytes* FPEROM, 256 *Bytes* RAM, 32 jalur I/O *(Input/Output*), dan dua 16-bit *timers/counters.*

2.3.1. *Timer* dan *Counter* dalam Mikrokontroler AT89S51

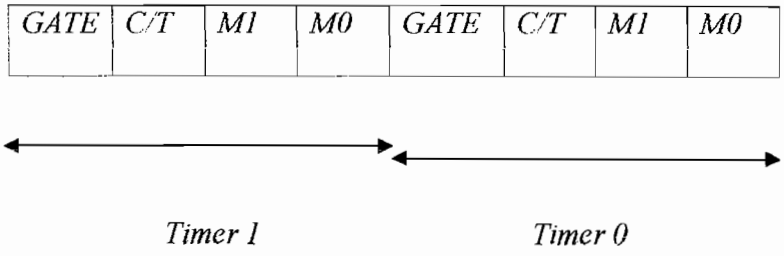
Mikrokontroler AT89S51 dilengkapi dengan dua buah *timer/counter*, yaitu *timer* 0 dan *timer* 1. Pencacah *timer/counter* AT89S51 merupakan pencacah biner

naik (*count-up binary counter*) yang mencacah dari 0000h sampai FFFFh. Saat kondisi pencacah berubah dari FFFFh kembali ke 0000h, akan timbul sinyal limpahan (*overflow*).

Masing-masing *timer* juga dapat berfungsi sebagai *counter*. Pada saat sebagai *timer*, register naik satu (*increment*) setiap satu siklus mesin,yaitu saat register kembali pada keadaan semula dimana pencacah berubah kembali dari 0000h kembali ke FFFFh.

2.3.1. a. *Timer Mode Register* (TMOD)

Penggunaan register TMOD yang ditunjukkan pada Gambar 2.4. adalah untuk mengatur kerja *Timer 0* dan *Timer 1*. Register TMOD terbagi atas dua yaitu, bit 0 sampai 3 (TMOD.0 sampai TMOD.3) untuk mengatur kerja *Timer* 0, sedangkan bit 4 sampai 7 (TMOD.4 sampai TMOD.7) untuk mengatur kerja *Timer* 1. Register TMOD merupakan register dengan bit *addressable*,maksudnya isi register ini tidak dapat diatur per bit tetapi pengaturannya langsung *byte*.



Gambar 2.4. Register *TMOD*

GATE : Jika *GATE* = 1, maka *timer/counter* "x" aktif bila pin INTx *high* dan pin TRx juga *high*.

Jika *GATE* = 0, maka *timer/counter* "x" aktif jika hanya pin TRx *high*.

C/T : *low* untuk fungsi *timer* dan *high* untuk fungsi *counter*.

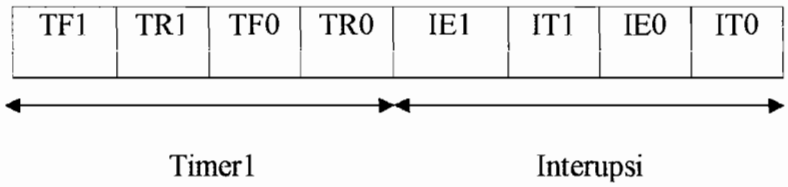
M1 dan M0 : pemilih mode *timer/counter* (Mode 0 sampai Mode3) yang konfigurasinya dapat dilihat pada tabel II.1.

Tabel II.1. Mode operasi pemilih *Timer/counter* AT89S51

M1	M0	Mode	Operasi
0	0	0	*Timer/counter* 13-bit
0	1	1	*Timer/counter* 16-bit
1	0	2	*Timer/counter* 8-bit isi ulang (*auto reload*)
1	1	3	Gabungan *timer/counter* 16-bit dan 8-bit

2.3.1.b. Timer control Register Timer 0 dan 1

TCON merupakan *bit addressable* sehingga bisa diatur per-bitnya (dengan instruksi SETB atau CLR). Register TCON seperti yang ditunjukkan pada gambar 2.5., berisi pengaturan *timer* dan *interupsi eksternal* sekaligus dalam 1 *byte*.



Gambar 2.5. Register TCON

Jika dalam pemrograman tidak memakai *interupsi eksternal*, maka IE dan IT dapat diabaikan (diset '0').

TR1 dan TR0: pengatur aktif dan nonaktif *timer / counter.*

TF1 dan TF0 : penampung bit limpahan (*overflow*) *timer/counter*

IE1 dan IE0 : tanda (*flag*) interupsi eksternal

IT1 dan IT0 : menentukan pen-*trigger*-an interupsi eksternal.

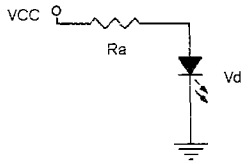
2.4. Piranti Output

Dalam *counter up* enam digit digunakan *seven segment* sebagai penampil keluarannya.

2.4.1. LED

Cara kerja LED adalah dengan memancarkan cahaya. Terjadinya pancaran cahaya adalah sebagai berikut, saat dioda menghantarkan arus, elektron lepas dari ikatannya dengan menggunakan tanaga yang diambil dari catu daya listrik.

Setelah elekton lepas, elektro-elektron tersebut banyak yang bergabung dengan *hole* di sekitarnya. Saat masuk *hole* lain, elektron mengeluarkan tenaga yang terkumpul lalu diradiasikan dalam bentuk cahaya sehingga dioda akan memancarkan cahaya.



Gambar 2.6. Rangkaian indikator LED

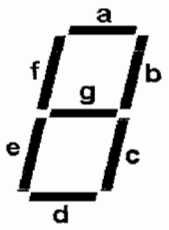
Penggunaan resistor rangkaian indikator LED adalah sebagai pembatas arus,maksudnya resistor membatasi arus yang masuk pada LED dari sumber (Vcc) sesuai dengan yang dibutuhkan,seperti yang terlihat pada gambar 2.6.

2.4.2. Transistor

Transistor PNP A733 yang digunakan berfungsi sebagai saklar untuk menghubungkan antara *seven segment* dengan tegangan Vcc. Transistor saat berada dalam kondisi saturasi seperti sebuah saklar yang tertutup dari terminal emiter ke kolektor dan apabila transistor dalam kondisi *cut off* maka transistor seperti sebuah saklar yang terbuka dari terminal emiter ke kolektor. Resistor R_B dan R_E digunakan sebagai pembatas arus yang masuk ke dalam transistor. Arus kolektor transistor adalah $I_C = I_E - I_B$, karena nilai I_B sangat kecil maka nilai $I_C \cong I_E$.

2.4.3. Penampil *Seven Segment*

Penampil merupakan suatu alat peraga dari suatu besaran yang diukur. Ada berbagai jenis penampil yang digunakan untuk mengetahui suatu hasil proses, salah satu yang biasa digunakan adalah *seven segme*nt yang ditunjukkan pada gambar 2.7.



Gambar 2.7. Bentuk Tampilan *Seven Segment*

Pada *seven segment*, untuk menampilkan suatu lambang harus dinyalakan tiap *segment* yang berkaitan dengan lambang tersebut yang digerakkan oleh saklar. Sebagai contoh, bila diinginkan desimal 6 menyala, saklar a, f, e, d, c, g ditutup, maka *segment* LED a, f, e, d, c, g menyala sehingga akan ditampilkan desimal 6.

Untuk contoh yang lainnya dapat dilihat pada tabel II.2.

Tabel II.2. Tabel Kebenaran *Seven Segment*

Cacahan	Segmen Yang Menyala
0	a, b, c, d, e, f
1	b, c
2	a, b, g, e, d
3	a, b, g, c, d
4	f, g, b, c
5	a, f, g, c, d
6	a, f, g, c, d
7	a, b, c
8	a, b, c, d, e, f, g
9	a, b, g, f, c

BAB III

PERANCANGAN ALAT

3.1. Perancangan Perangkat Keras

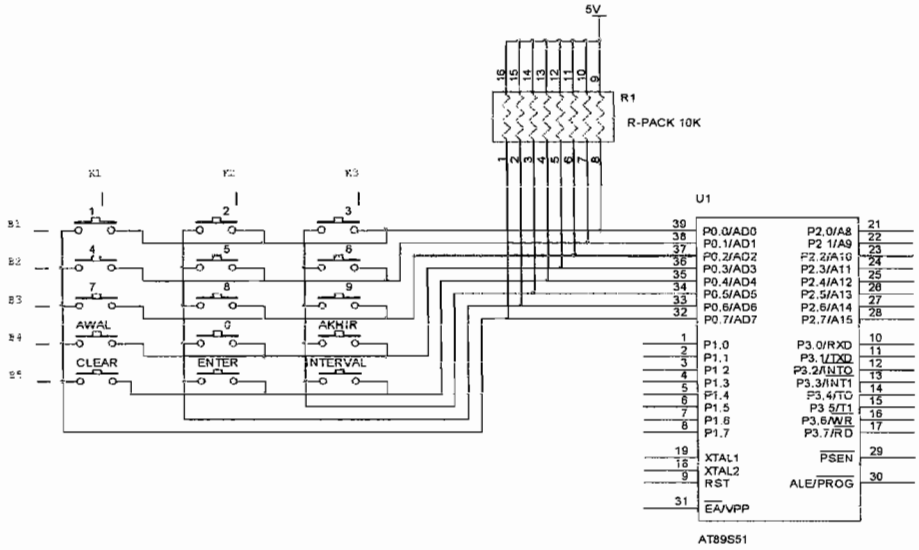
Counter up enam digit memiliki tiga bagian utama yaitu masukan, pengontrol dan keluaran. Piranti masukannya adalah *matrix keypad* sebagai masukan yang tersedia beberapa tombol yang digunakan sebagai masukan data yang diperlukan. Piranti kontrolnya adalah mikrokontroler AT89S51, sedangkan piranti keluarannya digunakan penampil *seven segment* enam digit.

Gambar 3.1. Diagram Blok *Counter Up* Enam Digit

Melalui *matrix keypad* data yang diperlukan dimasukan, kemudian diantar menuju proses pengolahan data yaitu pada mikrokontroler. Mikrokontroler sebagai pengolah, menterjemahkan masukan sesuai dengan yang diinginkan. Setelah proses pengolahan selesai maka data yang ada tersebut ditampilkan pada unit penampil, seperti yang tertera pada gambar 3.1

3.1.1 . Rangkaian *keypad* dengan Mikrokontroler AT89S51

Keypad yang digunakan adalah *matrix keypad 5×3. Matrix keypad* ini dihubungkan dengan *port* 3 dari mikrokontroler, dimana kelima barisnya (B1-B5) dihubungkan dengan jalur P3.3....P3.7, sedangkan ketiga kolomnya (K1-K3) dihubungkan dengan jalur P3.0....P3.2. Adapun *interface* rangkaian *matrix keypad* dengan mikrokontroler dapat dilihat pada gambar 3.2.



Gambar 3.2. Rangkaian *interface matrix keypad 5 x3*

Baris 1 sampai dengan baris 5 (P3.3....P3.7) berfungsi sebagai masukkan mikrikontroler , sedangkan kolom 1 sampai dengan kolom 3 (P3.0....P3.2) berfungsi sebagai keluaran pada mikrokontroler.

Adapun cara kerja dari *matrix keypad* ini adalah dengan menggunakan sistem *scanning* tombol satu per satu. Keluaran dari mikrokontroler akan memberikan kondisi '0' kepada setiap *port* keluaran secara bergantian, sehingga apabila ada tombol yang ditekan maka baris pada tombol tersebut akan memiliki kondisi '0' pula dan memberikan masukan pada mikrokontroler. *Matrix keypad* ini digunakan untuk memasukkan instruksi pilihan bagi alat, memasukkan nilai *modulus* pada menu *counter* dan juga memasukkan nilai *first count* pada menu *counter.*

Saat proses pengambilan data dari *keypad,* pertama-tama jalur kolom (P0.0....P0.2) berfungsi sebagai output dengan logika keluaran 0 sedangkan jalur baris (P0.3....P0.7) berfungsi sebagai input dengan *pull up* resistor. Jika sebuah tombol ditekan salah satu pin dari jalur baris akan terhubung dengan jalur kolom yang berlogika 0 dan menyebabkan logika dari pin baris tersebut menjadi 0. Empat pin baris lainnya tetap berlogika 1 karena *pull up* resistor. Selanjutnya program mengecek pin mana yang berlogika 0.

Kemudian dengan cara yang sama namun menukar fungsi baris menjadi *output* dan kolom menjadi *input,* program akan menentukan tombol mana yang sedang ditekan.

Tabel III.1. merupakan petunjuk penggunaan tombol-tobol pada kotak masukan yang mana digunakan sebagai tombol masukan bagi data-data yang diperlukan sebagai proses awal dari keseluruhan cara kerja alat.

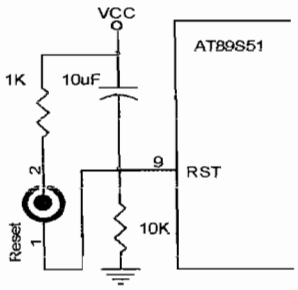
Tabel III.1 Tombol *keypad* dan fungsinya

Tombol Keypad	Fungsi
1	Angka 1
2	Angka 2
3	Angka 3
4	Angka 4
5	Angka 5
6	Angka 6
7	Angka 7
8	Angka 8
9	Angka 9
0	Angka 0
NILAI AWAL	Untuk Memasukkan Nilai Awal
NILAI AKHIR	Untuk Memasukkan Nilai Akhir
ENTER	Untuk Menjalankan
CLEAR	Untuk Mengclearkan Data
INTERVAL	Untuk Memasukkan Nilai Interval

3.1.2 Mikrokontroler AT89S51

3.1.2.a. Rangkaian Reset

Rangkaian reset digunakan untuk mereset mikrokontroler pada saat catu daya dihidupkan seperti pada gambar 3.3.



Gambar 3.3. Rangkaian reset

Keadaan reset pada mikrokontroler diperoleh apabila pin reset diberi logika tinggi (biasanya dalam waktu beberapa milidetik). Waktu reset tersebut dapat dihitung dengan rumus T = RC. Pada perancangan ini waktu reset 100 ms dengan menggunakan kapasitor C = 10 μF, maka nilai resistansi dapat dihitung :

$$100 \text{ ms} = 10 \mu\text{F} \times R$$

$$R = 100.10^{-3} / 10.10^{-6}$$

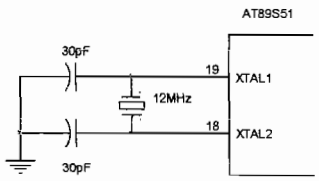
$$R = 10 \text{ K}\Omega.$$

Cara kerja rangkaian reset adalah sebagai berikut, bila tegangan catu dihidupkan arus akan mengalir melewati kapasitor sehingga akan menimbulkan

beda tegangan pada resistor. Tegangan pada pin reset merupakan beda tegangan antara Vcc dengan kapasitor.

3.1.2.b. Rangkaian Osilator

Mikrokontroler mempunyai rangkaian osilator internal (*on-chip oscillator*) yang dapat digunakan sebagai sumber *clock* bagi CPU. Untuk dapat menggunakan rangkaian osilator dalam *chip* tersebut, harus ditambahkan sebuah kristal dan dua buah kapasitor pada pin XTAL1 dan pin XTAL2 (pin 19 dan pin 18) seperti pada gambar 3.4.



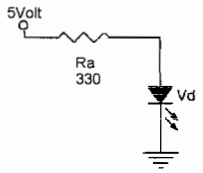
Gambar 3.4. Rangkaian osilator

Rangkaian osilator ini menggunakan kristal 12 MHz dan dua buah kapasitor 30 pF sehingga frekuensi detak pada CPU adalah 12 MHz.

3.1.3. Unit Penampil

3.1.3.a. Dioda Pemancar cahaya

LED yang diberi pra-tegangan maju akan memancarkan cahaya, untuk rangkaian seperti ditunjukkan pada gambar 3.5.



Gambar 3.5. Rangkaian indikator LED

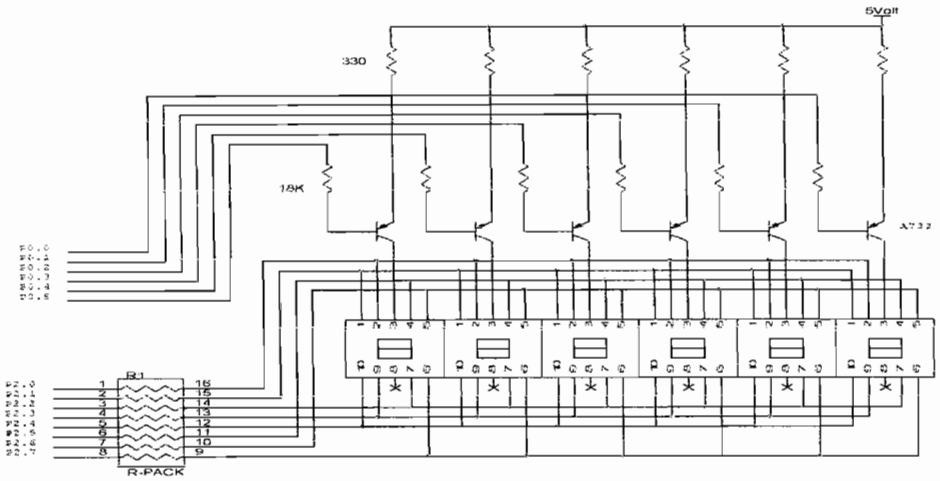
Pada perancangan rangkaian ini, berdasarkan *data sheet* V_d = 1,7 Volt dan I_d = 10mA serta Vcc yang digunakan adalah 5 Volt, maka nilai R_a adalah

$$R_a = \frac{5V - 1{,}7V}{10mA} = 330\Omega$$

Jadi resistor yang digunakan adalah 330 Ω sebagai hambatan penahan arus yang melewati LED.

3.1.3.b. Rangkaian penampil dengan *seven segment*

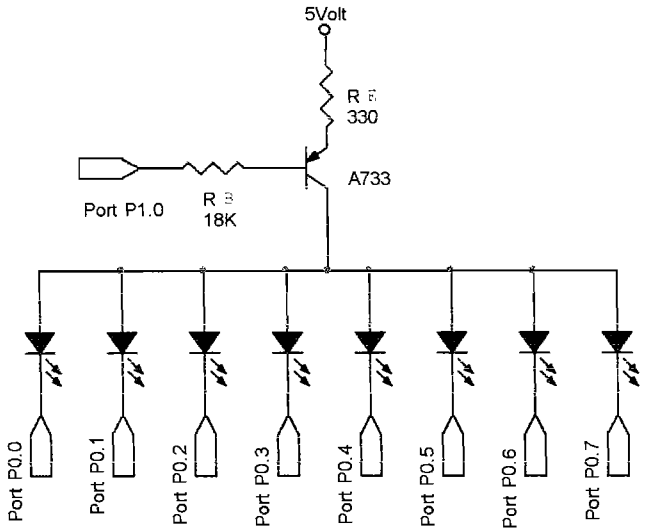
Rangkaian penampil yang digunakan adalah *seven segment* tiga digit yang langsung dihubung dengan mikrokontroler AT89S51 pada *port* P0 sebagai saluran data. Dan *port* P2 (P2.0....P2.5) digunakan sebagai saklar yang menghubungkan antara Vcc dan CA (*Common Anoda*) pada *seven segment*, seperti ditunjukkan pada gambar 3.6.



Gambar 3.6. Rangkaian Penampil *seven segment*

Untuk menampilkan datanya dengan metode *scanning*, yaitu pengiriman data keluaran dari mikrokontroler ke *seven segment* secara bergantian dan dengan cepat sehingga terlihat seakan-akan hidup secara bersamaan.

Transistor PNP A733 yang digunakan berfungsi sebagai saklar untuk menghubungkan antara *seven segment* dengan tegangan Vcc seperti terlihat pada gambar 3.7.



Gambar 3.7. Skema Dasar Konfigurasi Saklar Menggunakan Transistor

Agar transistor menjadi ON (kondisi jenuh) maka pada keluaran port P1.0 harus diberi logika 0 sehingga terdapat arus yang mengalir dari Vcc ke CA.

Dari *data sheet* transistor A733 dapat diperoleh besar arus penguatan dc (*hfe*) adalah 60, arus kolektor (I_C) maksimal 100mA dan besarnya V_{EC} adalah 0,18 Volt. Diketahui pula arus LED 10mA sampai 20mA, maka ditentukan nilai I_C saturasi 10mA yang disesuaikan dengan arus minimal LED.

Saat keluaran Port P1.0 bernilai 0 maka transistor berada dalam keadaan aktif, maka diperoleh persamaan sebagai berikut.

Dari *data sheet* A733, $V_{EB} = 0{,}7V$, $I_C = 10mA$, $V_{EC} = 0{,}18V$, $V_{LED} = 1{,}7V$.

Nilai R_E di dapat dengan persamaan :

$$V_{CC} - V_E - V_{EC} - V_{LED} = 0$$

$$V_E = V_{CC} - V_{EC} - V_{LED}$$

$$V_E = 5 - 0{,}18 - 1{,}7$$

$$V_E = 3{,}12V$$

$$V_E = I_E R_E \quad ; \; R_E = \frac{V_E}{I_E}$$

Besarnya nilai R_E berdasarkan persamaan di atas adalah :

$$R_E = \frac{3{,}12}{10.10^{-3}} = 312\Omega$$

Besarnya nilai R_E yang di dapat dari perhitungan adalah 312Ω, maka digunakan resistor yang mendekati harga tersebut yaitu resistor sebesar 330Ω.

Saat transistor saturasi I_B mencapai nilai jenuh sebesar :

$$I_{B\,jenuh} = \frac{I_C}{Hfe\min} = \frac{10.10^{-3}}{60} = 0{,}167mA$$

Maka besarnya nilai R_B yang digunakan sebagai penahan muka arus yang masuk ke transistor melalui kaki basis adalah :

$$R_B = \frac{V_B}{I_B} = \frac{3{,}12}{0{,}167.10^{-3}} = 18682{,}6\Omega$$

Dalam perhitungan di dapat nilai R_B sebesar 18682,6Ω maka digunakan resistor yang mudah di dapat di pasaran, yaitu resistor sebesar 18KΩ.

3.2. PERANCANGAN PERANGKAT LUNAK

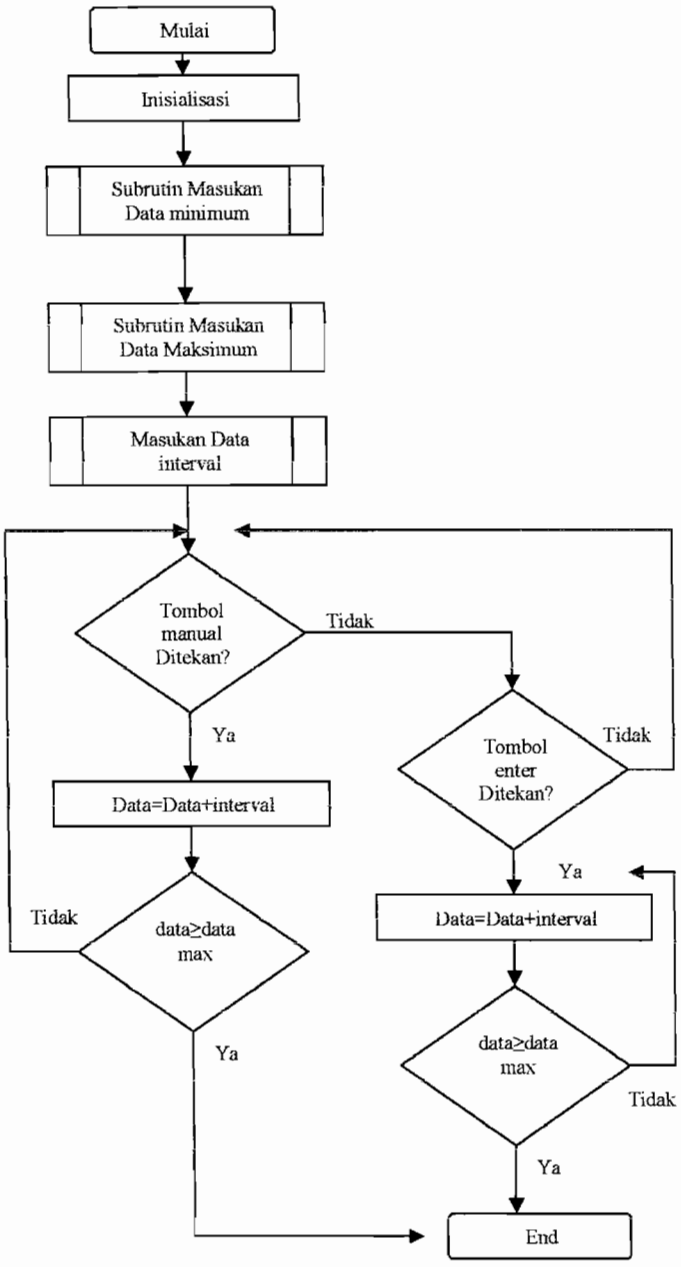
3.2.1. Diagram Alir program utama

Diagram alir program utama dimulai dengan proses inisialisasi alamat penyimpan data. Kemudian dilakukan pemanggilan masukan data minimum untuk memasukkan data minimum. Kemudian dilakukan pemanggilan subrutin data maksimum yang berguna menginput data maksimum, dan program akan lanjut ke masukkan data interval.

Kemudian dilakukan pengecekan tombol MANUAL. Bila tidak ditekan maka akan dicek apakah tombol *ENTER* ditekan. Bila tombol *ENTER* ditekan maka akan ditambahkan data dengan interval secara otomatis dari angka minimum hingga mencapai angka maksimum, tetapi bila *ENTER* tidak ditekan maka akan kembali ke pengecekan tombol MANUAL.. Dan bila tombol MANUAL ditekan maka data akan ditambah dengan interval secara manual satu per satu dari nilai minimum hingga mencapai nilai maksimum.

Pada penambahan dilakukan perbandingan data dengan data maksimum. Bila data belum mencapai data maksimum maka akan kembali ke pengecekan penekanan tombol MANUAL, dan bila data sudah sama atau lebih besar maka program akan ke bagian akhir.

Diagram alir program utama digambarkan pada gambar 3.8



Gambar 3.8. Diagram Alir program utama

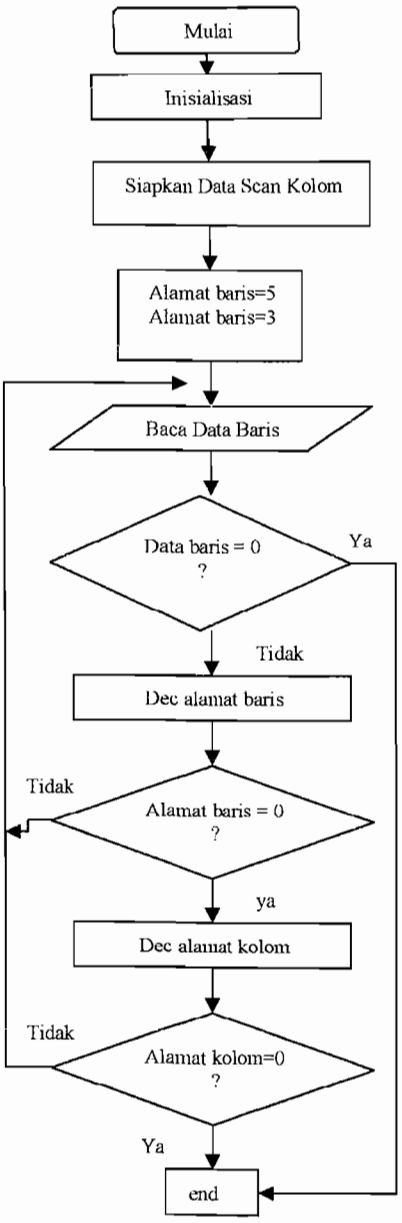
3.2.2 . Diagram Alir Subrutin Keypad

Pada subrutin keypad kolom (P0.0....P0.2) berfungsi sebagai output dengan logika keluaran 0 sedangkan jalur baris (P0.3....P0.7) berfungsi sebagai input.

Awalnya dilakukan inisialisasi dan data kolom 1 dibuat sama dengan nol. Kemudian dilakukan pembacaan baris 1 dan dibandingkan dengan angka nol. Bila sam dengan nol maka akan ke bagian akhir program dan keluar dari subrutin, tetapi bila tidak sam dilanjutkan dengan pembacaan data baris 2.

Hal ini sama untuk pembacaan baris 3, baris 4 dan baris 5. Kemudian dilakukan pengesetan kolom 3 bernilai nol, dilanjutkan pembacaan baris dari baris 1 hingga baris 5. Proses untuk kolom 3 juga sama seperti kolom 2 sebelumnya. Dan bila tidak ada tombol yang ditekan maka akan diulangi lagi dari awal proses *scanning keypad.*

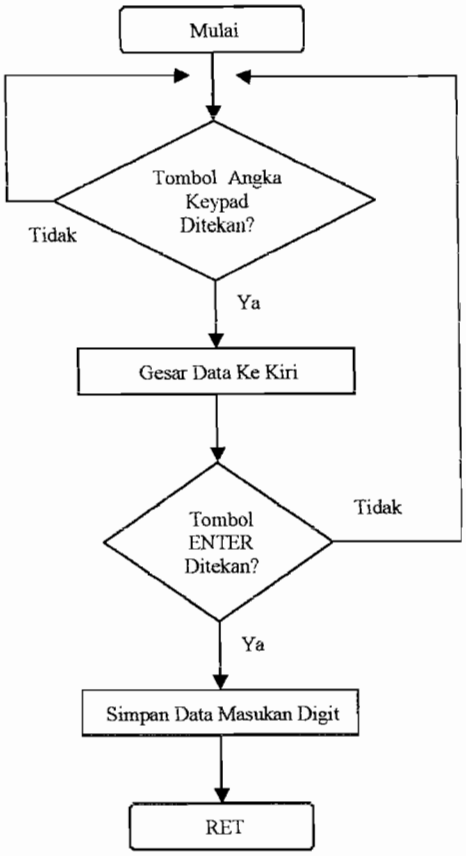
Diagram Alir Subrutin Keypad dapat dilihat pada Gambar 3.9.



Gambar 3.9. Diagram Alir Subrutin Keypad

3.2.3. **Diagram Alir Memasukkan Datan**

Diagram Alir Subrutin Memasukkan Data dapat dilihat pada Gambar 3.10.



Gambar 3.10. Diagram Alir Memasukkan Data

Pada subrutin diagram alir ini, dimulai dari pengecekan data tombol yang ditekan. Bila tidak ada penekanan tombol angka maka akan berulang proses ini. Dan bila ada tombol angka yang ditekan maka dilanjutkan dengan menggeser data ke kiri. Lalu dilakukan pengecekan tombol enter. Bila tombol enter tidak ditekan maka program akan kembali ke pengecekan penekanan tombol keypad. Dan bila

enter ditekan berlanjut ke simpan data maksimum enam digit dan keluar dari subrutin ini.

BAB IV

HASIL DAN PEMBAHASAN

4.1. Hasil Akhir Perancangan

Sistem counter *up* enam digit berbasis mikrokontroler AT89S51 ini dirancang untuk menghitung dan menampilkan kenaikan data sesuai dengan masukan yang diberikan. Data yang ada dimulai dari data awal yaitu data minimum dengan penambahan interval hingga mencapai data maksimum.

Dalam proses pengerjaannya dapat diberikan data interval sesuai keinginan dimulai dari angka 0 sebagai interval minimum sampai dengan angka 9 sebagai interval maksimum.

4.2. Listing Program

Sistem kerja *Counter Up* Enam Digit Berbasis Mikrokontroler AT89S51 ini disusun menjadi suatu program berdasarkan cara kerja perangkat keras diagram alir yang telah dibuat.

Listing program ini diuraikan menjadi Program Utama dan subrutin-subrutin, seperti : listing program data minimum dan listing program data maksimum, dan listing program keypad.

4.2.1. Listing Program Utama

```
UTAMA:
        LCALL   AWAL_TAMPIL                 ;TAMPILAN AWAL, 3 DETIK, DISPLAY ------
        MOV     KEYDATA_LAMA,#0FFH
        MOV     SET_LOW_SATUAN,#0EH
        MOV     SET_LOW_PULUHAN,#0EH
```

```
        MOV     SET_LOW_RATUSAN,#0EH            ;UNTUK MEMBENTUK DISPLAY L0....
        MOV     SET_LOW_RIBUAN,#0EH
        MOV     SET_LOW_PULUH_RIBU,#0H
        MOV     SET_LOW_RATUS_RIBU,#0CH
        MOV     TEMPORER,#0FFH
        LCALL   TAMPIL_LOW                      ;PANGGIL RUTIN DISPLAY LO


        MOV     SET_HIGH_SATUAN,#0EH            ;UNTUK MEMBENTUK DISPLAY H1....
        MOV     SET_HIGH_PULUHAN,#0EH
        MOV     SET_HIGH_RATUSAN,#0EH
        MOV     SET_HIGH_RIBUAN,#0EH
        MOV     SET_HIGH_PULUH_RIBU,#1H
        MOV     SET_HIGH_RATUS_RIBU,#0BH
        MOV     TEMPORER,#0FFH
        LCALL   TAMPIL_HIGH                     ;PANGGIL RUTIN DISPLAY H1

        MOV     SET_INTERVAL,#0EH               ;UNTUK MEMBENTUK TAMPILAN J.....
        MOV     ANGKA_PULUHAN,#0EH
        MOV     ANGKA_RATUSAN,#0EH
        MOV     ANGKA_RIBUAN,#0EH
        MOV     ANGKA_PULUH_RIBU,#0EH
        MOV     ANGKA_RATUS_RIBU,#0DH
        MOV     TEMPORER,#0FFH

        MOV     BANYAK_TAMPIL,#09FH             ;BISA UNTUK SETTING JEDA WAKTUNYA
ANTAR KENAIKAN
        LCALL   TAMPIL_INTERVAL

        LCALL   UBAH_DESIMAL                    ;UBAH SEMUA SET DARI INPUT YG DIISI 0EH
(BLANK0

;CEK MANUAL ATAU AUTO COUNTER
;SETELAH MASUKAN J, JIKA DITEKAN ENTER = AUTO, MAN = MANUAL
        MOV     A,TEMPORER_3
        CJNE    A,#0FFH,LAGI_COUNTER
        MOV     TEMPORER,#0FFH

COUNTER_MANUAL:

        MOV     BANYAK_TAMPIL,#0AH              ;BISA UNTUK SETTING JEDA WAKTUNYA
        MOV     ANGKA_SATUAN,SET_LOW_SATUAN
        MOV     ANGKA_PULUHAN,SET_LOW_PULUHAN
        MOV     ANGKA_RATUSAN,SET_LOW_RATUSAN
        MOV     ANGKA_RIBUAN,SET_LOW_RIBUAN
        MOV     ANGKA_PULUH_RIBU,SET_LOW_PULUH_RIBU
```

```
        MOV     ANGKA_RATUS_RIBU,SET_LOW_RATUS_RIBU

        LCALL   TAMPILKAN

;TUNGGU TOMBOL MAN DITEKAN
;===================================
TUNGGU_TOMBOL_MAN:
        LCALL   INPUT_KEYPAD
        MOV     A,KEYDATA
        CJNE    A,#0FFH,TUNGGU_TOMBOL_MAN_2   ;DITEKAN LOMPAT
        SJMP    COUNTER_MANUAL
TUNGGU_TOMBOL_MAN_2:
        MOV     A,KEYDATA
        CJNE    A,#0FH,COUNTER_MANUAL ;DITEKAN LOMPAT

;===================================

TERUS_MANUAL:

        LCALL   CEK_DATA
        MOV     A,TEMPORER_2
        CJNE    A,#0H,SELESAI

        MOV     TEMP_LOW_SATUAN,SET_LOW_SATUAN
        MOV     TEMP_LOW_PULUHAN,SET_LOW_PULUHAN
        MOV     TEMP_LOW_RATUSAN,SET_LOW_RATUSAN
        MOV     TEMP_LOW_RIBUAN,SET_LOW_RIBUAN
        MOV     TEMP_LOW_PULUH_RIBU,SET_LOW_PULUH_RIBU
        MOV     TEMP_LOW_RATUS_RIBU,SET_LOW_RATUS_RIBU

        LCALL   COUNTER_UP
        LCALL   CEK_DATA
        MOV     A,TEMPORER_2
        CJNE    A,#0H,SELESAI

        LCALL   TAMPILKAN
        SJMP    COUNTER_MANUAL

LAGI_COUNTER:
        MOV     BANYAK_TAMPIL,#0AFH
        MOV     ANGKA_SATUAN,SET_LOW_SATUAN
        MOV     ANGKA_PULUHAN,SET_LOW_PULUHAN
        MOV     ANGKA_RATUSAN,SET_LOW_RATUSAN
        MOV     ANGKA_RIBUAN,SET_LOW_RIBUAN
        MOV     ANGKA_PULUH_RIBU,SET_LOW_PULUH_RIBU
        MOV     ANGKA_RATUS_RIBU,SET_LOW_RATUS_RIBU
```

```
        LCALL   CEK_DATA
        MOV     A,TEMPORER_2
        CJNE    A,#0H,SELESAI

        MOV     TEMP_LOW_SATUAN,SET_LOW_SATUAN
        MOV     TEMP_LOW_PULUHAN,SET_LOW_PULUHAN
        MOV     TEMP_LOW_RATUSAN,SET_LOW_RATUSAN
        MOV     TEMP_LOW_RIBUAN,SET_LOW_RIBUAN
        MOV     TEMP_LOW_PULUH_RIBU,SET_LOW_PULUH_RIBU
        MOV     TEMP_LOW_RATUS_RIBU,SET_LOW_RATUS_RIBU

        LCALL   COUNTER_UP
        LCALL   TAMPILKAN
        SJMP    LAGI_COUNTER

SELESAI:
        MOV     ANGKA_SATUAN,TEMP_LOW_SATUAN
        MOV     ANGKA_PULUHAN,TEMP_LOW_PULUHAN
        MOV     ANGKA_RATUSAN,TEMP_LOW_RATUSAN
        MOV     ANGKA_RIBUAN,TEMP_LOW_RIBUAN
        MOV     ANGKA_PULUH_RIBU,TEMP_LOW_PULUH_RIBU
        MOV     ANGKA_RATUS_RIBU,TEMP_LOW_RATUS_RIBU
        LCALL   TAMPILKAN
        SJMP    SELESAI
```

4.2.2. Listing Program Keypad

```
INPUT_KEYPAD:
        ACALL   KEYPAD3X4
        MOV     A,KEYDATA
        CJNE    A,KEYDATA_LAMA,SCAN_KEY_LAGI
        MOV     KEYDATA,#0FFH
        RET

SCAN_KEY_LAGI:
        MOV     KEYDATA_LAMA,KEYDATA
        RET

KEYPAD3X4:
        MOV     KEYBOUNC,#50
        MOV     KEYPORT,#0FFh

        CLR     KOLOM1
UL1:    JB      BARIS1,KEY1
        DJNZ    KEYBOUNC,UL1
```

```
        MOV     KEYDATA,#1H             ;1
        RET
KEY1:   JB      BARIS2,KEY2
        DJNZ    KEYBOUNC,KEY1
        MOV     KEYDATA,#4H             ;4
        RET
KEY2:   JB      BARIS3,KEY3
        DJNZ    KEYBOUNC,KEY2
        MOV     KEYDATA,#7H             ;7
        RET
KEY3:   JB      BARIS4,KEY4
        DJNZ    KEYBOUNC,KEY3
        MOV     KEYDATA,#0EH            ;ENTER
        RET

KEY4:   SETB    KOLOM1
        CLR     KOLOM2
        JB      BARIS1,KEY5
        DJNZ    KEYBOUNC,KEY4
        MOV     KEYDATA,#2H             ;2
        RET
KEY5:   JB      BARIS2,KEY6
        DJNZ    KEYBOUNC,KEY5
        MOV     KEYDATA,#5H             ;5
        RET
KEY6:   JB      BARIS3,KEY7
        DJNZ    KEYBOUNC,KEY6
        MOV     KEYDATA,#8H             ;8
        RET
KEY7:   JB      BARIS4,KEY8
        DJNZ    KEYBOUNC,KEY7
        MOV     KEYDATA,#0H             ;0
        RET


KEY8:   SETB    KOLOM2
        CLR     KOLOM3
        JB      BARIS1,KEY9
        DJNZ    KEYBOUNC,KEY8
        MOV     KEYDATA,#3H             ;3
        RET
KEY9:   JB      BARIS2,KEY10
        DJNZ    KEYBOUNC,KEY9
        MOV     KEYDATA,#6H             ;6
        RET
KEY10:  JB      BARIS3,KEY11
```

```
        DJNZ    KEYBOUNC,KEY10
        MOV     KEYDATA,#9H             ;9
        RET
KEY11:  JB      BARIS4,KEY12
        DJNZ    KEYBOUNC,KEY11
        MOV     KEYDATA,#0FH            ;MAN
        RET
KEY12:  MOV     KEYDATA,#0FFH           ;TDK DITEKAN
        RET
```

4.3. Pembahasan

4.3.1. Cara Kerja Alat

Cara kerja dari alat *Counter Up* Enam Digit Berbasis Mikrokontroler AT89S51 adalah sebagai berikut :

- Saat dihidupkan pertama kali, akan muncul tampilan berupa garis datar untuk tiap digit dari *seven segmen*. Setelah beberapa detik muncul tulisan L O.
- L O berarti kita diminta memasukan data awal yang diinginkan, kemudian bila sudah memasukkan data maka diakhiri dengan menekan tombol enter.
- Setelah tombol enter ditekan maka tampilan berikut adalah H I. Saat dalam kondisi ini kita memasukan nilai maksimum yang kita inginkan kemudian tekan enter.
- J. adalah lambang, sebagai tanda agar kita memasukan nilai interval. Dalam kondisi ini kita bisa memilih proses pengerjaan alat secara otomattis atau manual. Dimana apabila setelah memasukan nilai interval kita langsung menekan tombol enter maka proses penghitungan akan dimulai secara otomatis. Namun bila ingin secara manual kita tinggal menekan tombol manual bukan tombol enter. Secara manual angka yang ditampilkan akan menghitung naik atau bertambah setiap kali tombol manual ditekan.
- Kemudian program akan berjalan sesuai perintah yang diinginkan dengan penambahan data sebanyak interval.

4.3.2. Data Penggunaan Alat

Tabel Data Penggunaan Alat

no	L O	H I	J .	Angka kenaikan
1	1	9	1	1, 2, 3, 4, 5, 6, 7, 8
2	10	40	3	10, 13, 16, 19, 22, 25, 28, 31, 34, 37
3	120	200	6	120, 126, 132, 138, 144, 150, 156, 162, 168, 174, 180, 186, 192, 198
4	3000	3080	8	3000, 3008, 3016, 3024, 3032, 3040, 3048, 3056, 3064, 3072, 3078
5	900080	900150	5	900080, 900085, 900090, 900095, 900100, 900105, 900110, 900115, 900120, 900125, 900130, 900135, 900140, 900145
6	100	10	9	100
7	1000	800	7	1000
8	90000	50000	2	90000
9	999999	888888	4	999999

Dari tabel dapat dijelaskan bahwa setelah ada tampilan L O, angka yang diinginkan dimasukan kemudian tekan enter. Setelah itu akan ada tampilan H I yang maksudnya meminta angka tertinggi dimasukan,

setelah menekan enter muncul tampilan J . yang menginginkan angka interval dimasukan.

Setelah angka interval ditekan maka ada pilihan sesuai dengan keinginan pengguna alat apakah mau menghitung proses kenaikan secara otomatis atau manual, apabila memilih secara otomatis maka tinggal menekan tombol enter namun bila menginginkan secara manual maka tekan tombol manual untuk setiap kenaikannya.

Data nomor 1 sampai 5 merupakan data kenaikan yang sebenarnya, maksudnya sesuai dengan sistem kerja alat. Angka kenaikan yang ditampilkan sudah sesuai dengan intervalnya masing-masing, hanya jumlah akhir dari penambahan untuk penambahan akhir tidak mencapai nilai maksimal yang diinginkan, walaupun dari penjumlahan data yang sesungguhnya jumlahnya bisa sampai pada data maksimum (data $\leq$ nilai maks). Hal ini terjadi karena dalam program penghitungannya menggunakan masukan data lebih kecil dari nilai maksimum (data < data maks), dengan alasan untuk penghitungan penambahan angka puluhan sampai ratusan ribu dengan interval ganjil jumlahnya tidak lebih kecil sama dengan nilai maksimum.

Data nomor 6 sampai dengan nomor 9 menunjukan contoh kesalahan penggunaan alat, artinya tidak ada proses kenaikan data apabila data awal atau nilai awal (L O) lebih besar dari data akhir atau

nilai akhir (H I). Sehingga pada proses ini nilai yang ditunjukan pada alat adalah nilai awal saja.

BAB V

KESIMPULAN DAN SARAN

Berdasarkan hasil pengamatan dan pembahasan dapat diambil beberapa kesimpulan dan saran guna penyempurnaan dan pengembangan lebih lanjut dari alat ini.

5.1. Kesimpulan

Beberapa hal penting yang dapat dijadikan acuan sebagai kesimpulan dari proses perancangan serta dalam menganalisis data dari pekerjaan alat ini adalah :

1. Alat ini berfungsi untuk menghitung dan menampilkan *counter up* dengan tampilan enam *seven segment*
2. Proses kenaikan data tidak tepat pada batas maksimal yang diinginkan untuk semua masukan, walaupun ada kemungkinan untuk mencapai angka tersebut. Hal ini terjadi karena dalam program penghitungannya jumlah data (data + interval) selalu lebih kecil dari nilai akhir (H I)
3. Bila data masukan untuk data maksimum lebih kecil dari data minimum maka yang akan ditampilkan adalah data minimum
4. Sistem kerjanya dapat dioperasikan secara manual dan otomatis

5.2. Saran

Sehubungan dengan keadaan alat yang belum terlalu sempurna maka penulis memberikan saran-saran bagi pengembangan yang lebih lanjut yaitu :

1. Tampilan akan lebih baik jika dibuat lebih besar misalnya dengan menggunakan *seven segment* yang berukuran besar sehingga dapat lebih jelas terlihat.
2. Untuk menghindari terputusnya catu daya secara tiba-tiba (listrik padam) maka akan lebih baik apabila dilengkapi dengan catu daya cadangan di dalamnya.

DAFTAR PUSTAKA

Albert Paul Malvino, PH.D.,E.E, 1987, *Prinsip-prinsip Elektronika*, Penerbit Salemba Teknika, Jakarta

Atmel, 1997; *"AT89C51 : 8-bit Microcontroller and 4K Bytes Flash"*, Atmel Inc. (http://www.Atmel.com),USA

Eko Putro, 2002 : *Belajar Mikrokontroler AT89S51/52/55 Teori dan Aplikasi*,Penerbit Gava Media,Yogyakarta

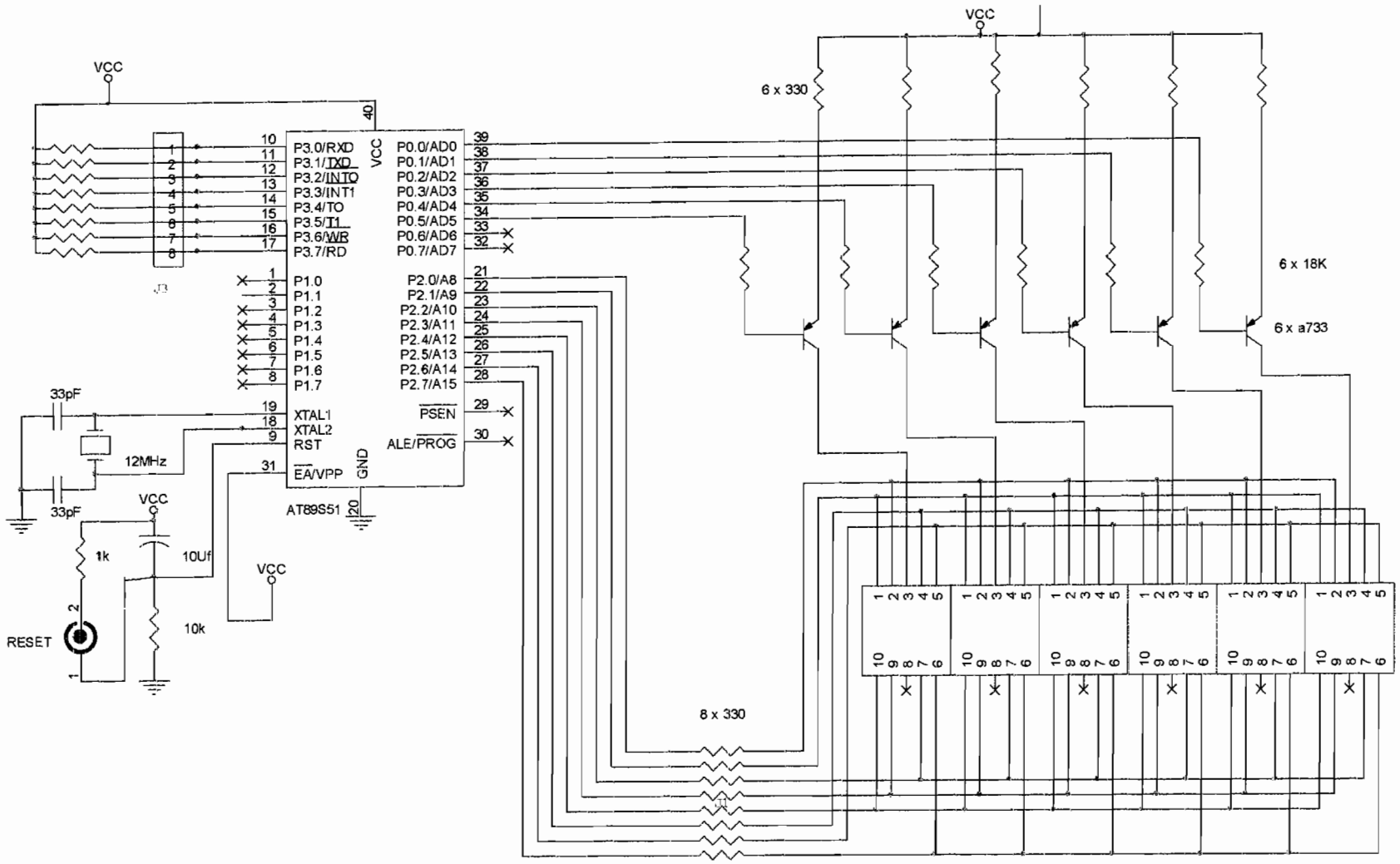
Riyanto, Bambang,1998, *Display 8 Digit 7 segment*, Data Sheet Down Loder Haline

Sutrisno,1987, *Elektronika: Teori dasar dan penerapannya*, Penerbit ITB, Bandung

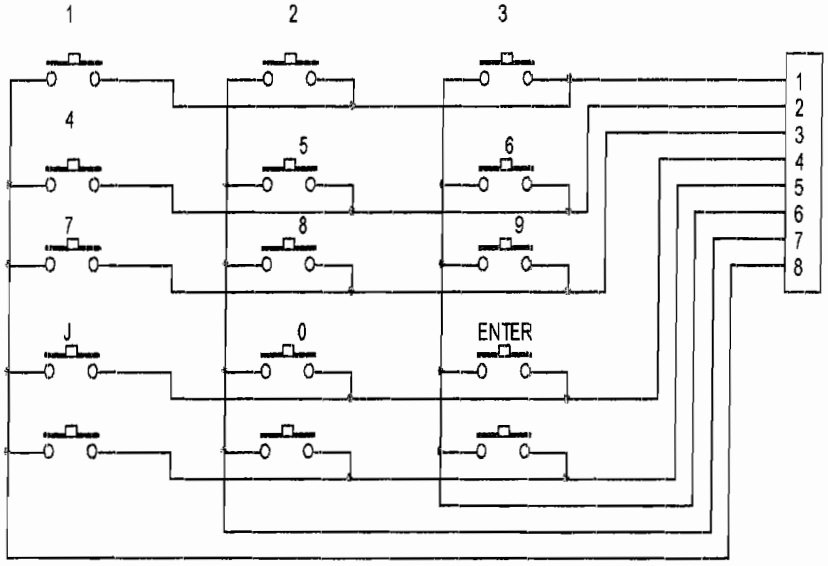
Thomas Sri Widodo,DEA,Dipl.Ing, 2002, *Elektronika Dasar*, Penerbit Salemba Teknika, Jakarta

LAMPIRAN

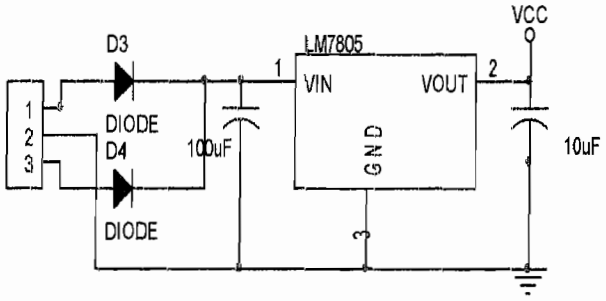
Rangkaian *Counter Up* Enam Digit Berbasis Mikrokontroler AT89S51



Rangkaian Keypad dan Power Supply



RANGKAIAN KEYPAD



RANGKAIAN POWER SUPPLY

```
;=============================================================================

; COUNTER UP ENAM DIGIT BERBASIS MIKROKONTROLER AT89S51

; Nama:Bonefasius Oldam

; Program Studi Teknik Elektro

;=============================================================================
        ORG     0H

;INI STANDAR BANYAKNYA TAMPILAN DIPANGGIL

KOLOM1 BIT      P3.7    ; kiri (1,4,7,ENT)
KOLOM2 BIT      P3.6
KOLOM3 BIT      P3.5
BARIS1  BIT     P3.0    ; atas (1,2,3)
BARIS2  BIT     P3.1
BARIS3  BIT     P3.3
BARIS4  BIT     P3.2
KEYPORT                 EQU     P3
KEYDATA                 EQU     43h
KEYBOUNC        EQU     42h
KEYDATA_LAMA            EQU     52H

;MASUKAN UNTUK DI DISPLAY
ANGKA_SATUAN            EQU     35H     ;PALING KANAN
ANGKA_PULUHAN           EQU     36H
ANGKA_RATUSAN           EQU     37H
ANGKA_RIBUAN            EQU     38H
ANGKA_PULUH_RIBU        EQU     39H
ANGKA_RATUS_RIBU        EQU     40H     ;PALING KIRI

;UNTUK SETTING ANGKA
SET_LOW_SATUAN                  EQU     45H     ;BATAS BAWAH
SET_LOW_PULUHAN                 EQU     46H
SET_LOW_RATUSAN                 EQU     47H
SET_LOW_RIBUAN          EQU     48H
SET_LOW_PULUH_RIBU      EQU     49H
SET_LOW_RATUS_RIBU      EQU     4AH

SET_HIGH_SATUAN                 EQU     4BH     ;BATAS ATAS
SET_HIGH_PULUHAN        EQU     4CH
SET_HIGH_RATUSAN        EQU     4DH
SET_HIGH_RIBUAN                 EQU     4EH
SET_HIGH_PULUH_RIBU     EQU     4FH
SET_HIGH_RATUS_RIBU     EQU     50H

SET_INTERVAL            EQU     51H     ;INTERVAL-NYA

TEMP_LOW_SATUAN                 EQU     59H
TEMP_LOW_PULUHAN        EQU     5AH
TEMP_LOW_RATUSAN        EQU     5BH
TEMP_LOW_RIBUAN                 EQU     5CH
TEMP_LOW_PULUH_RIBU     EQU     5DH
TEMP_LOW_RATUS_RIBU     EQU     5EH

TEMPORER_2              EQU     66H
TEMPORER_3              EQU     66H

;=======================================
UTAMA:
        LCALL   AWAL_TAMPIL                     ;TAMPILAN AWAL, 3 DETIK, DISPLAY ------
        MOV     KEYDATA_LAMA,#0FFH
        MOV     SET_LOW_SATUAN,#0EH
        MOV     SET_LOW_PULUHAN,#0EH
        MOV     SET_LOW_RATUSAN,#0EH            ;UNTUK MEMBENTUK DISPLAY L0....
        MOV     SET_LOW_RIBUAN,#0EH
```

```
        MOV     SET_LOW_PULUH_RIBU,#0H
        MOV     SET_LOW_RATUS_RIBU,#0CH
        MOV     TEMPORER,#0FFH
        LCALL   TAMPIL_LOW                      ;PANGGIL RUTIN DISPLAY LO

        MOV     SET_HIGH_SATUAN,#0EH            ;UNTUK MEMBENTUK DISPLAY H1....
        MOV     SET_HIGH_PULUHAN,#0EH
        MOV     SET_HIGH_RATUSAN,#0EH
        MOV     SET_HIGH_RIBUAN,#0EH
        MOV     SET_HIGH_PULUH_RIBU,#1H
        MOV     SET_HIGH_RATUS_RIBU,#0BH
        MOV     TEMPORER,#0FFH
        LCALL   TAMPIL_HIGH                     ;PANGGIL RUTIN DISPLAY H1

        MOV     SET_INTERVAL,#0EH               ;UNTUK MEMBENTUK TAMPILAN J.....
        MOV     ANGKA_PULUHAN,#0EH
        MOV     ANGKA_RATUSAN,#0EH
        MOV     ANGKA_RIBUAN,#0EH
        MOV     ANGKA_PULUH_RIBU,#0EH
        MOV     ANGKA_RATUS_RIBU,#0DH
        MOV     TEMPORER,#0FFH
        MOV     BANYAK_TAMPIL,#09FH             ;BISA UNTUK SETTING JEDA WAKTUNYA
ANTAR KENAIKAN
        LCALL   TAMPIL_INTERVAL

        LCALL   UBAH_DESIMAL            ;UBAH SEMUA SET DARI INPUT YG DIISI 0EH (BLANK0

;CEK MANUAL ATAU AUTO COUNTER
;SETELAH MASUKAN J, JIKA DITEKAN ENTER = AUTO, MAN = MANUAL
        MOV     A,TEMPORER_3
        CJNE    A,#0FFH,LAGI_COUNTER
        MOV     TEMPORER,#0FFH

COUNTER_MANUAL:
        MOV     BANYAK_TAMPIL,#0AH              ;BISA UNTUK SETTING JEDA WAKTUNYA
        MOV     ANGKA_SATUAN,SET_LOW_SATUAN
        MOV     ANGKA_PULUHAN,SET_LOW_PULUHAN
        MOV     ANGKA_RATUSAN,SET_LOW_RATUSAN
        MOV     ANGKA_RIBUAN,SET_LOW_RIBUAN
        MOV     ANGKA_PULUH_RIBU,SET_LOW_PULUH_RIBU
        MOV     ANGKA_RATUS_RIBU,SET_LOW_RATUS_RIBU
        LCALL   TAMPILKAN

;TUNGGU TOMBOL MAN DITEKAN
;=================================
TUNGGU_TOMBOL_MAN:
        LCALL   INPUT_KEYPAD
        MOV     A,KEYDATA
        CJNE    A,#0FFH,TUNGGU_TOMBOL_MAN_2     ;DITEKAN LOMPAT
        SJMP    COUNTER_MANUAL
TUNGGU_TOMBOL_MAN_2:
        MOV     A,KEYDATA
        CJNE    A,#0FH,COUNTER_MANUAL ;DITEKAN LOMPAT

;=================================
TERUS_MANUAL:
        LCALL   CEK_DATA
        MOV     A,TEMPORER_2
        CJNE    A,#0H,SELESAI
        MOV     TEMP_LOW_SATUAN,SET_LOW_SATUAN
        MOV     TEMP_LOW_PULUHAN,SET_LOW_PULUHAN
        MOV     TEMP_LOW_RATUSAN,SET_LOW_RATUSAN
        MOV     TEMP_LOW_RIBUAN,SET_LOW_RIBUAN
        MOV     TEMP_LOW_PULUH_RIBU,SET_LOW_PULUH_RIBU
        MOV     TEMP_LOW_RATUS_RIBU,SET_LOW_RATUS_RIBU
        LCALL   COUNTER_UP
        LCALL   CEK_DATA
        MOV     A,TEMPORER_2
        CJNE    A,#0H,SELESAI
```

```
        LCALL   TAMPILKAN
        SJMP    COUNTER_MANUAL
LAGI_COUNTER:
        MOV     BANYAK_TAMPIL,#0AFH
        MOV     ANGKA_SATUAN,SET_LOW_SATUAN
        MOV     ANGKA_PULUHAN,SET_LOW_PULUHAN
        MOV     ANGKA_RATUSAN,SET_LOW_RATUSAN
        MOV     ANGKA_RIBUAN,SET_LOW_RIBUAN
        MOV     ANGKA_PULUH_RIBU,SET_LOW_PULUH_RIBU
        MOV     ANGKA_RATUS_RIBU,SET_LOW_RATUS_RIBU
        LCALL   CEK_DATA
        MOV     A,TEMPORER_2
        CJNE    A,#0H,SELESAI

        MOV     TEMP_LOW_SATUAN,SET_LOW_SATUAN
        MOV     TEMP_LOW_PULUHAN,SET_LOW_PULUHAN
        MOV     TEMP_LOW_RATUSAN,SET_LOW_RATUSAN
        MOV     TEMP_LOW_RIBUAN,SET_LOW_RIBUAN
        MOV     TEMP_LOW_PULUH_RIBU,SET_LOW_PULUH_RIBU
        MOV     TEMP_LOW_RATUS_RIBU,SET_LOW_RATUS_RIBU

        LCALL   COUNTER_UP
        LCALL   TAMPILKAN
        SJMP    LAGI_COUNTER

SELESAI:
        MOV     ANGKA_SATUAN,TEMP_LOW_SATUAN
        MOV     ANGKA_PULUHAN,TEMP_LOW_PULUHAN
        MOV     ANGKA_RATUSAN,TEMP_LOW_RATUSAN
        MOV     ANGKA_RIBUAN,TEMP_LOW_RIBUAN
        MOV     ANGKA_PULUH_RIBU,TEMP_LOW_PULUH_RIBU
        MOV     ANGKA_RATUS_RIBU,TEMP_LOW_RATUS_RIBU
        LCALL   TAMPILKAN
        SJMP    SELESAI


;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
; RUTIN UNTUK COUNTER UP
;=========================================
COUNTER_UP:
        MOV     A,SET_LOW_SATUAN
        ADD     A,SET_INTERVAL
        MOV     R3,A            ;R3 HASIL 1
        SUBB    A,#0AH          ;A, SKRG ADL HASIL 2, LIHAT CARRY
        JNC     NEXT_DATA_1
        MOV     SET_LOW_SATUAN,R3
        RET

NEXT_DATA_1:
        MOV     SET_LOW_SATUAN,A
        MOV     A,SET_LOW_PULUHAN
        ADD     A,#1H
        MOV     R3,A            ;R3 HASIL 1
        SUBB    A,#0AH          ;A, SKRG ADL HASIL 2, LIHAT CARRY
        JNC     NEXT_DATA_2
        MOV     SET_LOW_PULUHAN,R3
        RET

NEXT_DATA_2:
        MOV     SET_LOW_PULUHAN,A
        MOV     A,SET_LOW_RATUSAN
        ADD     A,#1H
        MOV     R3,A            ;R3 HASIL 1
        SUBB    A,#0AH          ;A, SKRG ADL HASIL 2, LIHAT CARRY
        JNC     NEXT_DATA_3
        MOV     SET_LOW_RATUSAN,R3
        RET
```

```
NEXT_DATA_3:
        MOV     SET_LOW_RATUSAN,A
        MOV     A,SET_LOW_RIBUAN
        ADD     A,#1H
        MOV     R3,A                ;R3 HASIL 1
        SUBB    A,#0AH              ;A, SKRG ADL HASIL 2, LIHAT CARRY
        JNC     NEXT_DATA_4
        MOV     SET_LOW_RIBUAN,R3
        RET
NEXT_DATA_4:
        MOV     SET_LOW_RIBUAN,A
        MOV     A,SET_LOW_PULUH_RIBU
        ADD     A,#1H
        MOV     R3,A                ;R3 HASIL 1
        SUBB    A,#0AH              ;A, SKRG ADL HASIL 2, LIHAT CARRY
        JNC     NEXT_DATA_5
        MOV     SET_LOW_PULUH_RIBU,R3
        RET
NEXT_DATA_5:
        MOV     SET_LOW_PULUH_RIBU,A
        MOV     A,SET_LOW_RATUS_RIBU
        ADD     A,#1H
        MOV     R3,A                ;R3 HASIL 1
        SUBB    A,#0AH              ;A, SKRG ADL HASIL 2, LIHAT CARRY
        JNC     KELUAR_COUNTER_UP
        MOV     SET_LOW_RATUS_RIBU,R3

KELUAR_COUNTER_UP:
        RET

;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
; RUTIN UNTUK CEK NILAI SET_LOW APAKAH
; LEBIH BESAR DARI SET HIGH
;=================================================================
CEK_DATA:
        MOV     A,SET_LOW_RATUS_RIBU
        MOV     R4,A
        MOV     R5,SET_HIGH_RATUS_RIBU
        CJNE    A,SET_HIGH_RATUS_RIBU,CEK_DATA_TIDAK_SAMA

            ;JIKA SAMA MAKA CEK YANG BAWAHNYA
        MOV     A,SET_LOW_PULUH_RIBU
        MOV     R4,A
        MOV     R5,SET_HIGH_PULUH_RIBU
        CJNE    A,SET_HIGH_PULUH_RIBU,CEK_DATA_TIDAK_SAMA

            ;JIKA SAMA MAKA CEK YANG BAWAHNYA
        MOV     A,SET_LOW_RIBUAN
        MOV     R4,A
        MOV     R5,SET_HIGH_RIBUAN
        CJNE    A,SET_HIGH_RIBUAN,CEK_DATA_TIDAK_SAMA

            ;JIKA SAMA MAKA CEK YANG BAWAHNYA
        MOV     A,SET_LOW_RATUSAN
        MOV     R4,A
        MOV     R5,SET_HIGH_RATUSAN
        CJNE    A,SET_HIGH_RATUSAN,CEK_DATA_TIDAK_SAMA

            ;JIKA SAMA MAKA CEK YANG BAWAHNYA
        MOV     A,SET_LOW_PULUHAN
        MOV     R4,A
        MOV     R5,SET_HIGH_PULUHAN
        CJNE    A,SET_HIGH_PULUHAN,CEK_DATA_TIDAK_SAMA

            ;JIKA SAMA MAKA CEK YANG BAWAHNYA
        MOV     A,SET_LOW_SATUAN
        MOV     R4,A
        MOV     R5,SET_HIGH_SATUAN
```

```
CEK_DATA_TIDAK_SAMA:
        MOV     A,R4
        SUBB    A,R5
        JNC     CEK_DATA_BESAR                  ;LOW >HIGH
        MOV     TEMPORER_2,#01I         ;LOW <HIGH
        JMP     CEK_DATA_KELUAR

CEK_DATA_BESAR:
        MOV     TEMPORER_2,#0FFH

CEK_DATA_KELUAR:
        RET
;=============================
;UBAH KE DESIMAL
;=============================
UBAH_DESIMAL:
        MOV     A,SET_LOW_SATUAN
        CJNE    A,#0EH,UBAH_DESIMAL_1
        MOV     SET_LOW_SATUAN,#0H

UBAH_DESIMAL_1:
        MOV     A,SET_LOW_PULUHAN
        CJNE    A,#0EH,UBAH_DESIMAL_2
        MOV     SET_LOW_PULUHAN,#0H

UBAH_DESIMAL_2:
        MOV     A,SET_LOW_RATUSAN
        CJNE    A,#0EH,UBAH_DESIMAL_3
        MOV     SET_LOW_RATUSAN,#0H

UBAH_DESIMAL_3:
        MOV     A,SET_LOW_RIBUAN
        CJNE    A,#0EH,UBAH_DESIMAL_4
        MOV     SET_LOW_RIBUAN,#0H

UBAH_DESIMAL_4:
        MOV     A,SET_LOW_PULUH_RIBU
        CJNE    A,#0EH,UBAH_DESIMAL_5
        MOV     SET_LOW_PULUH_RIBU,#0H

UBAH_DESIMAL_5:
        MOV     A,SET_LOW_RATUS_RIBU
        CJNE    A,#0EH,UBAH_DESIMAL_6
        MOV     SET_LOW_RATUS_RIBU,#0H

UBAH_DESIMAL_6:
        MOV     A,SET_HIGH_SATUAN
        CJNE    A,#0EH,UBAH_DESIMAL_7
        MOV     SET_HIGH_SATUAN,#0H

UBAH_DESIMAL_7:
        MOV     A,SET_HIGH_PULUHAN
        CJNE    A,#0EH,UBAH_DESIMAL_8
        MOV     SET_HIGH_PULUHAN,#0H

UBAH_DESIMAL_8:
        MOV     A,SET_HIGH_RATUSAN
        CJNE    A,#0EH,UBAH_DESIMAL_9
        MOV     SET_HIGH_RATUSAN,#0H

UBAH_DESIMAL_9:
        MOV     A,SET_HIGH_RIBUAN
        CJNE    A,#0EH,UBAH_DESIMAL_10
        MOV     SET_HIGH_RIBUAN,#0H

UBAH_DESIMAL_10:
        MOV     A,SET_HIGH_PULUH_RIBU
        CJNE    A,#0EH,UBAH_DESIMAL_11
        MOV     SET_HIGH_PULUH_RIBU,#0H
```

```
UBAH_DESIMAL_11:
        MOV     A,SET_HIGH_RATUS_RIBU
        CJNE    A,#0EH,UBAH_DESIMAL_12
        MOV     SET_HIGH_RATUS_RIBU,#0H

UBAH_DESIMAL_12:
        MOV     A,SET_INTERVAL
        CJNE    A,#0FH,UBAH_DESIMAL_KELUAR
        MOV     SET_INTERVAL,#0H

UBAH_DESIMAL_KELUAR:
        RET

;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
; RUTIN AWAL TAMPIL
;SET UNTUK TAMPILAN YANG PERTAMA KALI
;TAMPILKAN TULISAN "------"
;SELAMA 3 DETIK
;========================================================================
AWAL_TAMPIL:
        MOV     ANGKA_SATUAN,#0AH
        MOV     ANGKA_PULUHAN,#0AH
        MOV     ANGKA_RATUSAN,#0AH
        MOV     ANGKA_RIBUAN,#0AH
        MOV     ANGKA_PULUH_RIBU,#0AH
        MOV     ANGKA_RATUS_RIBU,#0AH
        MOV     BANYAK_TAMPIL,#0FFH
        LCALL   TAMPILKAN
        LCALL   TAMPILKAN
        RET

;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
; RUTIN TAMPIL UNTUK MINTA INPUT BATAS LOW
;TAMPILKAN TULISAN "L0    "
;========================================================================
TAMPIL_LOW:
        MOV     ANGKA_SATUAN,SET_LOW_SATUAN
        MOV     ANGKA_PULUHAN,SET_LOW_PULUHAN
        MOV     ANGKA_RATUSAN,SET_LOW_RATUSAN
        MOV     ANGKA_RIBUAN,SET_LOW_RIBUAN
        MOV     ANGKA_PULUH_RIBU,SET_LOW_PULUH_RIBU
        MOV     ANGKA_RATUS_RIBU,SET_LOW_RATUS_RIBU
        MOV     BANYAK_TAMPIL,#08H
        LCALL   TAMPILKAN
        ACALL   INPUT_KEYPAD
        MOV     A,KEYDATA
        CJNE    A,#0FFH,TAMPIL_LOW_CEK_1        ;DITEKAN LOMPAT
        JMP     TAMPIL_LOW

TAMPIL_LOW_CEK_1:
        CJNE    A,#0FH,TAMPIL_LOW_CEK_2         ;DITEKAN MAN
        JMP     TAMPIL_LOW

TAMPIL_LOW_CEK_2:
        CJNE    A,#0EH,TAMPIL_LOW_MASUK_DATA            ;DITEKAN ENTER
        RET

TAMPIL_LOW_MASUK_DATA:
        MOV     A,TEMPORER
        CJNE    A,#0FFH,TAMPIL_LOW_MASUK_DATA_2
        MOV     SET_LOW_PULUH_RIBU,#0EH
        MOV     SET_LOW_RATUS_RIBU,#0EH
        MOV     TEMPORER,#0H

TAMPIL_LOW_MASUK_DATA_2:
        LCALL   GESER_DATA_LOW
        MOV     SET_LOW_SATUAN,KEYDATA

        JMP     TAMPIL_LOW
```

```
GESER_DATA_LOW:
        CLR     A
        XCH     A,SET_LOW_SATUAN                ;DATA DI SATUAN GESER KE PULUHAN
        XCH     A,SET_LOW_PULUHAN               ;PULUHAN GESER DI RATUSAN
        XCH     A,SET_LOW_RATUSAN               ;DST
        XCH     A,SET_LOW_RIBUAN
        XCH     A,SET_LOW_PULUH_RIBU
        XCH     A,SET_LOW_RATUS_RIBU
        RET

;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
; RUTIN TAMPIL UNTUK MINTA INPUT BATAS HIGH
;TAMPILKAN TULISAN "H1   "
;SELAMA 1 DETIK
;=========================================================================
TAMPIL_HIGH:
        MOV     ANGKA_SATUAN,SET_HIGH_SATUAN
        MOV     ANGKA_PULUHAN,SET_HIGH_PULUHAN
        MOV     ANGKA_RATUSAN,SET_HIGH_RATUSAN
        MOV     ANGKA_RIBUAN,SET_HIGH_RIBUAN
        MOV     ANGKA_PULUH_RIBU,SET_HIGH_PULUH_RIBU
        MOV     ANGKA_RATUS_RIBU,SET_HIGH_RATUS_RIBU
        MOV     BANYAK_TAMPIL,#08H
        LCALL   TAMPILKAN
        ACALL   INPUT_KEYPAD
        MOV     A,KEYDATA
        CJNE    A,#0FFH,TAMPIL_HIGH_CEK_1       ;DITEKAN LOMPAT
        JMP     TAMPIL_HIGH

TAMPIL_HIGH_CEK_1:
        CJNE    A,#0FH,TAMPIL_HIGH_CEK_2                ;DITEKAN MAN
        JMP     TAMPIL_HIGH

TAMPIL_HIGH_CEK_2:
        CJNE    A,#0EH,TAMPIL_HIGH_MASUK_DATA           ;DITEKAN ENTER
        RET

TAMPIL_HIGH_MASUK_DATA:
        MOV     A,TEMPORER
        CJNE    A,#0FFH,TAMPIL_HIGH_MASUK_DATA_2
        MOV     SET_HIGH_PULUH_RIBU,#0EH
        MOV     SET_HIGH_RATUS_RIBU,#0EH
        MOV     TEMPORER,#0H

TAMPIL_HIGH_MASUK_DATA_2:
        LCALL   GESER_DATA_HIGH
        MOV     SET_HIGH_SATUAN,KEYDATA
        JMP     TAMPIL_HIGH

GESER_DATA_HIGH:
        CLR     A
        XCH     A,SET_HIGH_SATUAN               ;DATA DI SATUAN GESER KE PULUHAN
        XCH     A,SET_HIGH_PULUHAN              ;PULUHAN GESER DI RATUSAN
        XCH     A,SET_HIGH_RATUSAN              ;DST
        XCH     A,SET_HIGH_RIBUAN
        XCH     A,SET_HIGH_PULUH_RIBU
        XCH     A,SET_HIGH_RATUS_RIBU
        RET

;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
; RUTIN TAMPIL UNTUK MINTA INPUT  INTERVAL
;TAMPILKAN TULISAN "J    "
;SELAMA 1 DETIK
;=========================================================================
TAMPIL_INTERVAL:
        MOV     ANGKA_SATUAN,SET_INTERVAL
        MOV     BANYAK_TAMPIL,#08H
        LCALL   TAMPILKAN
        ACALL   INPUT_KEYPAD
```

```
        MOV     A,KEYDATA
        CJNE    A,#0FFH,TAMPIL_INTERVAL_CEK_1   ;DITEKAN LOMPAT
        JMP     TAMPIL_INTERVAL

TAMPIL_INTERVAL_CEK_1:
        CJNE    A,#0FH,TAMPIL_INTERVAL_CEK_2            ;DITEKAN MAN
        MOV     TEMPORER_3,#0FFH
        RET

TAMPIL_INTERVAL_CEK_2:
        CJNE    A,#0EH,TAMPIL_INTERVAL_MASUK_DATA               ;DITEKAN ENTER
        MOV     TEMPORER_3,#0H
        RET

TAMPIL_INTERVAL_MASUK_DATA:
        MOV     A,TEMPORER
        CJNE    A,#0FFH,TAMPIL_INTERVAL_MASUK_DATA_2
        MOV     ANGKA_RATUS_RIBU,#0EH
        MOV     TEMPORER,#0H

TAMPIL_INTERVAL_MASUK_DATA_2:
        MOV     SET_INTERVAL,KEYDATA
        JMP     TAMPIL_INTERVAL

;================================================
; RUTIN UNTUK TOMBOL MATRIK 3X4 pada P3
;OUTPUT ADA PADA ALAMAT => KEYDATA  70H
;MEMORI
;       KEYDATA         EQU     43h
;       KEYBOUNC        EQU     42h
;================================================
INPUT_KEYPAD:
        ACALL   KEYPAD3X4
        MOV     A,KEYDATA
        CJNE    A,KEYDATA_LAMA,SCAN_KEY_LAGI
        MOV     KEYDATA,#0FFH
        RET

SCAN_KEY_LAGI:
        MOV     KEYDATA_LAMA,KEYDATA
        RET

KEYPAD3X4:
        MOV     KEYBOUNC,#50
        MOV     KEYPORT,#0FFh

        CLR     KOLOM1
UL1:    JB      BARIS1,KEY1
        DJNZ    KEYBOUNC,UL1
        MOV     KEYDATA,#1H             ;1
        RET
KEY1:   JB      BARIS2,KEY2
        DJNZ    KEYBOUNC,KEY1
        MOV     KEYDATA,#4H             ;4
        RET
KEY2:   JB      BARIS3,KEY3
        DJNZ    KEYBOUNC,KEY2
        MOV     KEYDATA,#7H             ;7
        RET
KEY3:   JB      BARIS4,KEY4
        DJNZ    KEYBOUNC,KEY3
        MOV     KEYDATA,#0EH            ;ENTER
        RET
KEY4:   SETB    KOLOM1
        CLR     KOLOM2
        JB      BARIS1,KEY5
        DJNZ    KEYBOUNC,KEY4
        MOV     KEYDATA,#2H             ;2
        RET
```

```
KEY5:   JB      BARIS2,KEY6
        DJNZ    KEYBOUNC,KEY5
        MOV     KEYDATA,#5H             ;5
        RET
KEY6:   JB      BARIS3,KEY7
        DJNZ    KEYBOUNC,KEY6
        MOV     KEYDATA,#8H             ;8
        RET
KEY7:   JB      BARIS4,KEY8
        DJNZ    KEYBOUNC,KEY7
        MOV     KEYDATA,#0H             ;0
        RET
KEY8:   SETB    KOLOM2
        CLR     KOLOM3
        JB      BARIS1,KEY9
        DJNZ    KEYBOUNC,KEY8
        MOV     KEYDATA,#3H             ;3
        RET
KEY9:   JB      BARIS2,KEY10
        DJNZ    KEYBOUNC,KEY9
        MOV     KEYDATA,#6H             ;6
        RET
KEY10:  JB      BARIS3,KEY11
        DJNZ    KEYBOUNC,KEY10
        MOV     KEYDATA,#9H             ;9
        RET
KEY11:  JB      BARIS4,KEY12
        DJNZ    KEYBOUNC,KEY11
        MOV     KEYDATA,#0FH            ;MAN
        RET
KEY12:  MOV     KEYDATA,#0FFH           ;TDK DITEKAN
        RET

;=============================================================
; RUTIN UNTUK MENAMPILKAN DATA
;PEMAKAIAN REGISTER
;       R1 => ACC
;       R0 => DELAY_DISPLAY
;       R2 => BANYAKNYA TAMPIL

;MASUKANNYA LEWAT ALAMAT MEMORI
;ANGKA_SATUAN           EQU     35H     ;PALING KANAN
;ANGKA_PULUHAN                  EQU     36H
;ANGKA_RATUSAN                  EQU     37H
;ANGKA_RIBUAN           EQU     38H
;ANGKA_PULUH_RIBU       EQU     39H
;ANGKA_RATUS_RIBU       EQU     40H     ;PALING KIRI
;=============================================================

TAMPILKAN:
        MOV     R2,BANYAK_TAMPIL

DISPLAY:
        Mov     DPTR,#DATA_HEX          ;Isi data pointer dg alamat DATA_HEX
        Mov     R1,#0DFH        ;R1 = Common

        MOV     A,ANGKA_RATUS_RIBU      ;ANGKA PALING KIRI
        Movc    A,@A+DPTR               ;Pindahkan data ke n ke A
        ACALL   DISPLAY_2

        MOV     A,ANGKA_PULUH_RIBU
        Movc    A,@A+DPTR               ;Pindahkan data ke n ke A
        ACALL   DISPLAY_2

        MOV     A,ANGKA_RIBUAN
        Movc    A,@A+DPTR               ;Pindahkan data ke n ke A
        ACALL   DISPLAY_2

        MOV     A,ANGKA_RATUSAN
```

```
        Movc    A,@A+DPTR               ;Pindahkan data ke n ke A
        ACALL   DISPLAY_2

        MOV     A,ANGKA_PULUHAN
        Movc    A,@A+DPTR               ;Pindahkan data ke n ke A
        ACALL   DISPLAY_2

        MOV     A,ANGKA_SATUAN                  ;ANGKA PALING KANAN
        Movc    A,@A+DPTR               ;Pindahkan data ke n ke A
        ACALL   DISPLAY_2
        DJNZ    R2,DISPLAY
        RET             ;KELUAR DARI RUTIN "TAMPILKAN"

;=============================================================================
DISPLAY_2:
        Mov     P2,A                    ;Keluarkan data A ke port 2
        Mov     A,R1                    ;Isi Accumulator dengan Register 1.
                                        ;BALIK NILAI A
        Mov     P0,A                    ;Keluarkan isi A (Common) ke port 1

        RR      A                       ;Rotate Accumulator Right
                                        ;BALIKKAN LAGI NILAI A
        Mov     R1,A                    ;Kembalikan isi Accumulator ke R1
        ACALL   DELAY_DISPLAY
        mov     P2,#0FFh        ;Buat P2 berlogika 1 (semua LED padam)
        mov     P0,#0FFh
        RET

;=============================================================================
=
;=====================
;DELAY UNTUK DISPLAY
;=====================
DELAY_DISPLAY:
        MOV     R0,#0FFH
DELAY2:
        DJNZ    R0,DELAY2
        RET

;==================
; DATA AREA
;==================
DATA_HEX:

;URUTAN          0  1  2  3  4  5  6  7  8  9
;-----------------------------------------------
;HASILNYA        0  1  2  3  4  5  6  7  8  9

        DB      50H,5FH,68H,4AH,47H,0C2H,0C0H,5BH,40H,42H


;URUTAN          10/0AH  11/0BH  12/0CH  13/0DH  14/0EH
;-----------------------------------------------
;HASILNYA        -       H       L       J       BLANK

        DB      0EFH,45H,0F4H,1CH,0FFH

;=============================================================================


        END
```

Features

- Compatible with MCS®-51 Products
- 4K Bytes of In-System Programmable (ISP) Flash Memory
  - Endurance: 1000 Write/Erase Cycles
- 4.0V to 5.5V Operating Range
- Fully Static Operation: 0 Hz to 33 MHz
- Three-level Program Memory Lock
- 128 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-bit Timer/Counters
- Six Interrupt Sources
- Full Duplex UART Serial Channel
- Low-power Idle and Power-down Modes
- Interrupt Recovery from Power-down Mode
- Watchdog Timer
- Dual Data Pointer
- Power-off Flag
- Fast Programming Time
- Flexible ISP Programming (Byte and Page Mode)

Description

The AT89S51 is a low-power, high-performance CMOS 8-bit microcontroller with 4K bytes of In-System Programmable Flash memory. The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard 80C51 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with In-System Programmable Flash on a monolithic chip, the Atmel AT89S51 is a powerful microcontroller which provides a highly-flexible and cost-effective solution to many embedded control applications.

The AT89S51 provides the following standard features: 4K bytes of Flash, 128 bytes of RAM, 32 I/O lines, Watchdog timer, two data pointers, two 16-bit timer/counters, a five-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S51 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power-down mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next external interrupt or hardware reset.

8-bit Microcontroller with 4K Bytes In-System Programmable Flash

AT89S51

3437E–MICRO–12/03

Pin Description

VCC

Supply voltage (all packages except 42-PDIP)

GND

Ground (all packages except 42-PDIP, for 42-PDIP GND connects only the logic core and the embedded program memory)

VDD

Supply voltage for the 42-PDIP which connects only the logic core and the embedded program memory

PWRVDD

Supply voltage for the 42-PDIP which connects only the I/O Pad Drivers. The application board **MUST** connect both VDD and PWRVDD to the board supply voltage.

PWRGND

Ground for the 42-PDIP which connects only the I/O Pad Drivers. PWRGND and GND are weakly connected through the common silicon substrate, but not through any metal link. The application board **MUST** connect both GND and PWRGND to the board ground

Port 0

Port 0 is an 8-bit open drain bi-directional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs.

Port 0 can also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode, P0 has internal pull-ups.

Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. **External pull-ups are required during program verification.**

Port 1

Port 1 is an 8-bit bi-directional I/O port with internal pull-ups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}) because of the internal pull-ups.

Port 1 also receives the low-order address bytes during Flash programming and verification.

Port Pin	Alternate Functions
P1.5	MOSI (used for In-System Programming)
P1.6	MISO (used for In-System Programming)
P1.7	SCK (used for In-System Programming)

Port 2

Port 2 is an 8-bit bi-directional I/O port with internal pull-ups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}) because of the internal pull-ups.

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, Port 2 uses strong internal pull-ups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ Ri), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

Port 3

Port 3 is an 8-bit bi-directional I/O port with internal pull-ups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pull-ups.

Port 3 receives some control signals for Flash programming and verification.

Port 3 also serves the functions of various special features of the AT89S51, as shown in the following table.

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{WR}$ (external data memory write strobe)
P3.7	$\overline{RD}$ (external data memory read strobe)

RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device. This pin drives High for 98 oscillator periods after the Watchdog times out. The DISRTO bit in SFR AUXR (address 8EH) can be used to disable this feature. In the default state of bit DISRTO, the RESET HIGH out feature is enabled.

ALE/$\overline{PROG}$

Address Latch Enable (ALE) is an output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input ($\overline{PROG}$) during Flash programming.

In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external data memory.

If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

$\overline{PSEN}$

Program Store Enable ($\overline{PSEN}$) is the read strobe to external program memory.

When the AT89S51 is executing code from external program memory, $\overline{PSEN}$ is activated twice each machine cycle, except that two $\overline{PSEN}$ activations are skipped during each access to external data memory.

$\overline{EA}$/VPP

External Access Enable. $\overline{EA}$ must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, $\overline{EA}$ will be internally latched on reset.

$\overline{EA}$ should be strapped to V_{CC} for internal program executions.

This pin also receives the 12-volt programming enable voltage (V_{PP}) during Flash programming.

XTAL1

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

XTAL2

Output from the inverting oscillator amplifier.

Timer 0 and 1

Timer 0 and Timer 1 in the AT89S51 operate the same way as Timer 0 and Timer 1 in the AT89C51. For further information on the timers' operation, refer to the Atmel Web site (http://www.atmel.com). From the home page, select "Products", then "Microcontrollers" then "8051-Architecture", then "Documentation", and "Other Documents". Open the Adobe Acrobat file "AT89 Series Hardware Description".

Interrupts

The AT89S51 has a total of five interrupt vectors: two external interrupts ($\overline{INT0}$ and $\overline{INT1}$), two timer interrupts (Timers 0 and 1), and the serial port interrupt. These interrupts are all shown in Figure 1.

Each of these interrupt sources can be individually enabled or disabled by setting or clearing a bit in Special Function Register IE. IE also contains a global disable bit, EA, which disables all interrupts at once.

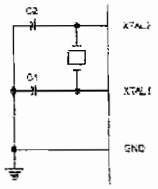
Note that Table 4 shows that bit positions IE.6 and IE.5 are unimplemented. User software should not write 1s to these bit positions, since they may be used in future AT89 products.

The Timer 0 and Timer 1 flags, TF0 and TF1, are set at S5P2 of the cycle in which the timers overflow. The values are then polled by the circuitry in the next cycle.

Oscillator Characteristics

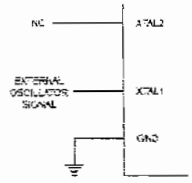
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier that can be configured for use as an on-chip oscillator, as shown in Figure 2. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven, as shown in Figure 3. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Figure 2. Oscillator Connections



Note: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

Figure 3. External Clock Drive Configuration



Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on Any Pin with Respect to Ground	-1.0V to +7.0V
Maximum Operating Voltage	6.6V
DC Output Current	15.0 mA

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC Characteristics

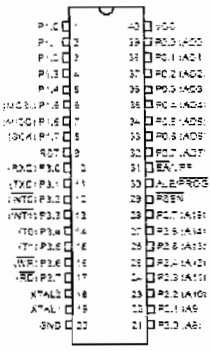
The values shown in this table are valid for T_A = -40°C to 85°C and V_{CC} = 4.0V to 5.5V, unless otherwise noted.

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low Voltage	(Except $\overline{EA}$)	-0.5	0.2 V_{CC}-0.1	V
V_{IL1}	Input Low Voltage ($\overline{EA}$)		-0.5	0.2 V_{CC}-0.3	V
V_{IH}	Input High Voltage	(Except XTAL1, RST)	0.2 V_{CC}+0.9	V_{CC}+0.5	V
V_{IH1}	Input High Voltage	(XTAL1, RST)	0.7 V_{CC}	V_{CC}+0.5	V
V_{OL}	Output Low Voltage[1] (Ports 1,2,3)	I_{OL} = 1.6 mA		0.45	V
V_{OL1}	Output Low Voltage[1] (Port 0, ALE, $\overline{PSEN}$)	I_{OL} = 3.2 mA		0.45	V
V_{OH}	Output High Voltage (Ports 1,2,3, ALE, $\overline{PSEN}$)	I_{OH} = -60 µA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -25 µA	0.75 V_{CC}		V
		I_{OH} = -10 µA	0.9 V_{CC}		V
V_{OH1}	Output High Voltage (Port 0 in External Bus Mode)	I_{OH} = -800 µA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -300 µA	0.75 V_{CC}		V
		I_{OH} = -80 µA	0.9 V_{CC}		V
I_{IL}	Logical 0 Input Current (Ports 1,2,3)	V_{IN} = 0.45V		-50	µA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1,2,3)	V_{IN} = 2V, V_{CC} = 5V ± 10%		-650	µA
I_{LI}	Input Leakage Current (Port 0, $\overline{EA}$)	0.45 < V_{IN} < V_{CC}		±10	µA
RRST	Reset Pulldown Resistor		50	300	KΩ
C_{IO}	Pin Capacitance	Test Freq. = 1 MHz, T_A = 25°C		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz		26	mA
		Idle Mode, 12 MHz		6.5	mA
	Power-down Mode[2]	V_{CC} = 5.5V		50	µA

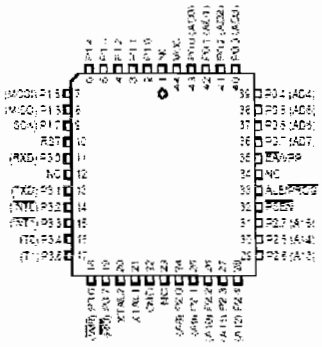
Notes: 1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
Maximum I_{OL} per port pin: 10 mA
Maximum I_{OL} per 8-bit port:
Port 0: 26 mA Ports 1, 2, 3: 15 mA
Maximum total I_{OL} for all output pins: 71 mA
If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
2. Minimum V_{CC} for Power-down is 2V.

Pin Configurations

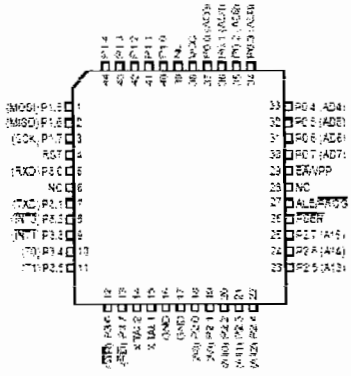
PDIP



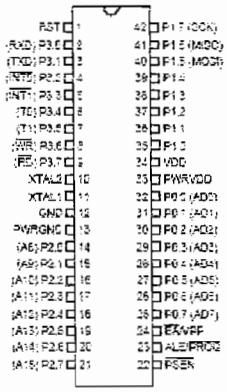
PLCC



TQFP



PDIP



Data Sheet Transistor A733

POL	: PNP
BAHAN	: SILIKON
V_{CB} MAX	: 50 V
V_{CE} MAX	: 40 V
V_{EB} MAX	: 5 V
Ic MAX	: 100 mA
T_0 MAX	: 125°C
Hfe	: 60

7 Segment LED Display – Single 0.56"

Absolute Maximum Rating (Ta = 25°C)

PARAMETER	RED	AMBER	GREEN	BLUE	WHITE	UNITS
DC Forward Current Per Segment	30	30	25	30	20	mA
Peak Current Per Segment (1)	70	50	50	25	25	mA
Avg. Forward Current (Pulse Operation) Per Segment	30	30	25	25	25	mA
Derating Linear From 25°C Per Segment	0.3					mA/°C
Reverse Voltage (2)	3					V
Operating Temperature	-25 to +85					°C
Storage Temperature	-30 to +85					°C

(1) Pulse conditions of 1/10 duty and 1 msec width for long operating life, max. of 20mA recommended
(2) Reverse biasing of the dot matrix is not recommended, will cause damage to the leds

Electro-optical Characteristics (Ta = 25°C)

PART NUMBER	NICE MATERIAL (COLOR)	PEAK WAVELENGTH nano	MAX. REVERSE CURRENT SEGMENT (μA)	VF (V) TYP	VF (V) MAX.	LUMINOUS INTENSITY / SEGMENT AVERAGE (IF = 10mA)
LCS-5612AUR11	AlGaAs Red	660	10	1.8	2.3	10,000 ucd
LCS-5612TB11	InGaN Blue	468	10	3.3	4.0	28,000 ucd
LCS-5612YG11	GaP Green	568	10	1.9	2.3	12,000 ucd
LCS-5612AY11	AlInGaP Amber	592	10	1.8	2.3	12,000 ucd
LCS-5612TW11	InGaN White	5,500K	10	3.3	4.0	28,000 ucd

DEVICE DIAGRAM