

## **BAB III**

### **LANDASAN TEORI**

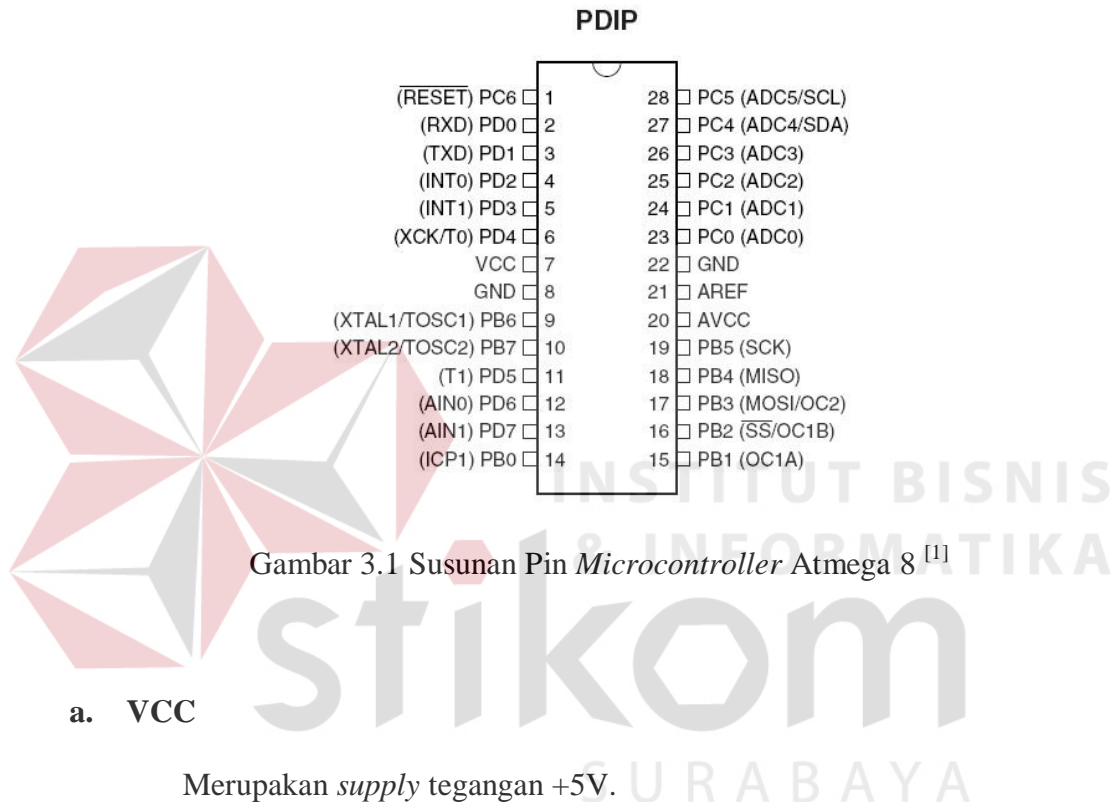
#### **3.1     *Microcontroller Atmega 8***

AVR merupakan salah satu jenis mikrokontroler yang di dalamnya terdapat berbagai macam fungsi. Perbedaan pada mikro yang pada umumnya digunakan seperti MCS51 adalah pada AVR tidak perlu menggunakan oscillator eksternal karena di dalamnya sudah terdapat internal oscillator. Selain itu kelebihan dari AVR adalah memiliki Power-On Reset, yaitu tidak perlu ada tombol reset dari luar karena cukup hanya dengan mematikan supply, maka secara otomatis AVR akan melakukan reset. Untuk beberapa jenis AVR terdapat beberapa fungsi khusus seperti ADC, EEPROM sekitar 128 byte sampai dengan 512 byte.

AVR Atmega 8 adalah mikrokontroler CMOS 8-*Bit* berarsitektur AVR RISC yang memiliki 8K byte in-System Programmable Flash. Mikrokontroler dengan konsumsi daya rendah ini mampu mengeksekusi instruksi dengan kecepatan maksimum 16MIPS pada frekuensi 16MHz. Jika dibandingkan dengan Atmega 8L perbedaannya hanya terletak pada besarnya tegangan yang diperlukan untuk bekerja. Untuk Atmega 8 tipe L, mikrokontroler ini dapat bekerja dengan tegangan antara 2,7 - 5,5 V sedangkan untuk Atmega 8 hanya dapat bekerja pada tegangan antara 4,5 – 5,5 V.

### 3.1.1 Konfigurasi Pin Atmega 8

Atmega 8 memiliki 28 Pin seperti pada Gambar 3.1, yang masing-masing pin nya memiliki fungsi yang berbeda-beda baik sebagai *Port* maupun fungsi yang lainnya. Berikut akan dijelaskan fungsi dari masing-masing kaki Atmega 8.



Gambar 3.1 Susunan Pin *Microcontroller* Atmega 8 <sup>[1]</sup>

#### a. VCC

Merupakan *supply* tegangan +5V.

#### b. GND

Merupakan *supply* tegangan 0V.

#### c. Port B (PB7...PB0)

Didalam *Port* B terdapat XTAL1, XTAL2, TOSC1, TOSC2. Jumlah *Port* B adalah 8 buah pin, mulai dari pin B.0 sampai dengan B.7. Tiap pin dapat digunakan sebagai input maupun output. *Port* B merupakan sebuah 8-*Bit bi-directional I/O* dengan internal *pull-up* resistor. Sebagai input, pin-pin

yang terdapat pada *Port B* yang secara eksternal diturunkan, maka akan mengeluarkan tegangan jika *pull-up* resistor diaktifkan. Khusus PB6 dapat digunakan sebagai input Kristal (*inverting oscillator amplifier*) dan input ke rangkaian *clock internal*, bergantung pada pengaturan *Fuses Bit* yang digunakan untuk memilih sumber *clock*. Sedangkan untuk PB7 dapat digunakan sebagai output Kristal (*output oscillator amplifier*) bergantung pada pengaturan *Fuses Bit* yang digunakan untuk memilih sumber *clock*. Jika sumber *clock* yang dipilih dari *oscillator internal* (*Synchronous*), PB7 dan PB6 dapat digunakan sebagai I/O atau jika menggunakan *oscillator external* (*Asynchronous*) *Timer/Counter2* maka PB6 dan PB7 (TOSC2 dan TOSC1) digunakan untuk saluran *input timer*.

#### d. **Port C (PC5...PC0)**

*Port C* merupakan sebuah 7-Bit bi-directional I/O *Port* yang di dalam masing-masing pin terdapat *pull-up* resistor. Jumlah pinnya hanya 7 buah mulai dari pin C.0 sampai dengan pin C.6. Sebagai keluaran/output, *Port C* memiliki karakteristik yang sama dalam hal menyerap arus (*sink*) ataupun mengeluarkan arus (*source*).

#### e. **RESET/PC6**

Jika RSTDISBL *Fuses* diprogram, maka PC6 akan berfungsi sebagai pin I/O. Pin ini memiliki karakteristik yang berbeda dengan pin-pin yang terdapat pada *Port C* lainnya. Namun jika RSTDISBL *Fuses* tidak diprogram, maka pin ini akan berfungsi sebagai *input reset*. Dan jika *level* tegangan yang masuk ke pin

ini rendah (0 V) dan pulsa yang ada lebih pendek dari pulsa minimum (1.5 us), maka akan menghasilkan suatu kondisi *reset* meskipun *clock*-nya tidak bekerja.

**f. Port D (PD7...PD0)**

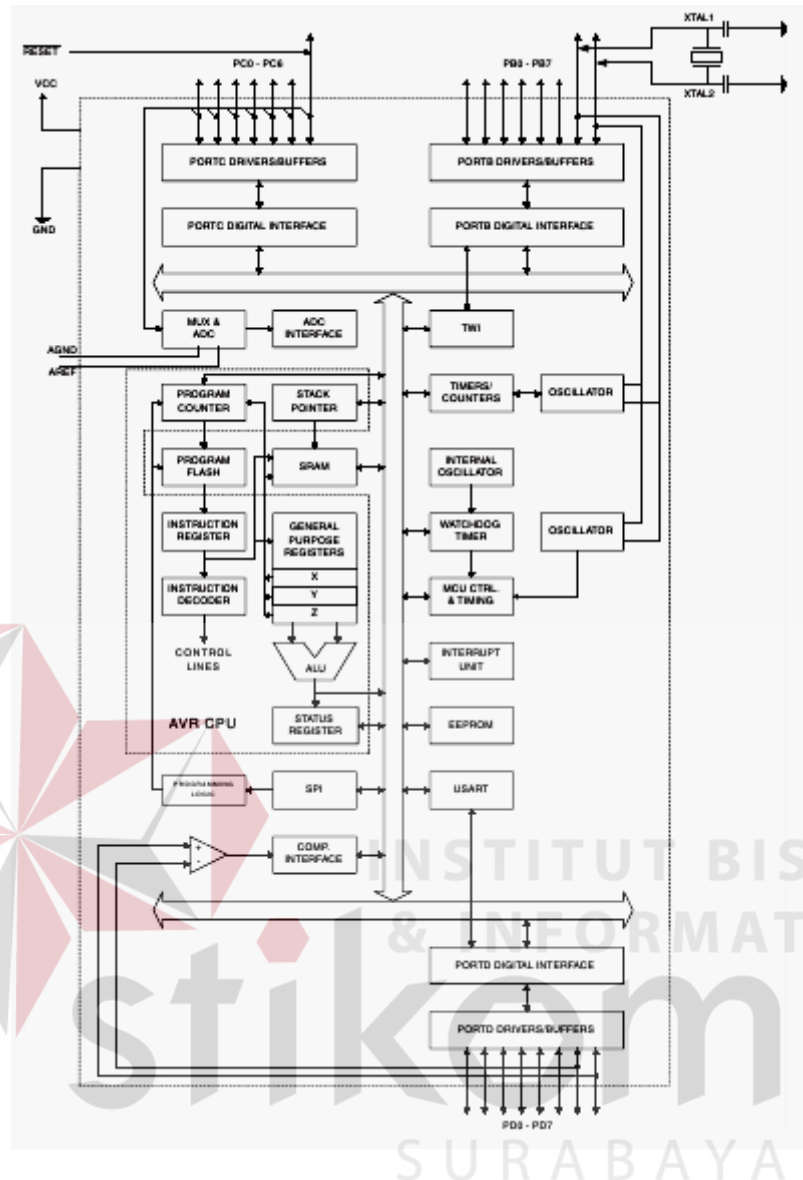
*Port D* merupakan 8-Bit bi-directional I/O dengan internal *pull-up* resistor. Fungsi dari *Port* ini sama dengan *Port-Port* yang lain. Hanya saja pada *Port* ini tidak terdapat kegunaan-kegunaan yang lain. Pada *Port* ini hanya berfungsi sebagai masukan dan keluaran saja atau biasa disebut dengan I/O.

**g. AVcc**

Pin ini berfungsi sebagai *supply* tegangan untuk ADC. Pin ini harus dihubungkan secara terpisah dengan VCC karena pin ini digunakan untuk analog saja. Bahkan jika ADC pada AVR tidak digunakan tetap saja disarankan untuk menghubungkannya secara terpisah dengan VCC. Jika ADC digunakan, maka AVcc harus dihubungkan ke VCC melalui *low pass filter*.

**h. AREF**

Merupakan pin referensi jika menggunakan ADC.



Gambar 3.2 Blok Diagram *Microcontroller Atmega 8* <sup>[1]</sup>

Status *Register* berisi informasi tentang hasil dari instruksi aritmatika yang terakhir dieksekusi. Informasi ini dapat digunakan untuk mengubah aliran program untuk melakukan operasi kondisional. *Register* ini di-update setelah semua operasi ALU (*Arithmetic Logic Unit*) dilakukan, seperti ditentukan dalam *Instruction Set Reference*. Hal ini mengurangi penggunaan instruksi perbandingan, sehingga prosesnya lebih cepat dan lebih terstruktur. Status

*Register* ini tidak secara otomatis tersimpan ketika menjalankan interupsi dan juga ketika menjalankan sebuah perintah setelah kembali dari interupsi. Sehingga hal tersebut harus dilakukan melalui *software*. Status *register* ada pada Gambar 3.3 berikut ini.

Bit	7	6	5	4	3	2	1	0	
	I	T	H	S	V	N	Z	C	SREG
Read/write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 3.3 Status *Register* Atmega 8 <sup>[1]</sup>

**i. Bit 7(I)**

Merupakan *bit Global Interrupt Enable*. *Bit* ini harus di-*set* agar semua perintah interupsi dapat dijalankan. Jika *bit* ini di-*reset*, maka semua perintah interupsi baik yang individual maupun yang secara umum akan diabaikan. *Bit* ini akan di-*reset* oleh *software* setelah sebuah interupsi dijalankan dan akan di-*set* kembali oleh perintah RETI (*Return form Interrupt*). *Bit* ini juga dapat di-*set* dan di-*reset* melalui aplikasi dan intruksi SEI (*Set Enable Interrupt*) dan CLI (*Clear Interrupt*).

**j. Bit 6(T)**

Merupakan *bit Copy Storage*. Instruksi *Bit Copy* BLD (*Bit Load*) dan BST (*Bit Storage*) menggunakan T-*bit* sebagai sumber atau tujuan untuk *bit* yang telah dioperasikan. Sebuah *bit* dari sebuah *register* dalam *Register File* dapat

disalin ke dalam T-bit dengan menggunakan instruksi BST, dan sebuah *bit* di dalam T-bit ini dapat disalin ke dalam *bit* di dalam *register* pada *Register File* dengan menggunakan perintah BLD.

**k. Bit 5(H)**

Merupakan *bit Half Carry Flag*. *Bit* ini menandakan sebuah *Half Carry* dalam beberapa operasi aritmatika. *Bit* ini berfungsi dalam aritmatika BCD.

**l. Bit 4(S)**

Merupakan *Sign Bit*. *Bit* ini merupakan *bit* eksklusif atau di antara *Negative Flag* (N) dan *two's Complement Overflow Flag* (V).

**m. Bit 3(V)**

Merupakan *Bit Two's Complement Overflow Flag*. *Bit* ini menyediakan fungsi aritmatika dua komplemen.

**n. Bit 2(N)**

Merupakan *Bit Negative Flag*. *Bit* ini mengindikasikan sebuah hasil *negative* di dalam sebuah fungsi logika atau aritmatika.

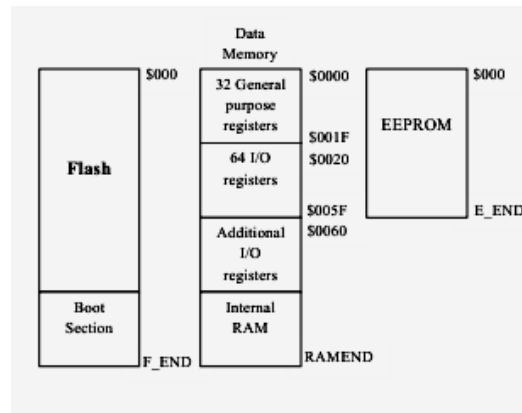
**o. Bit 1(Z)**

Merupakan *Bit Zero Flag*. *Bit* ini mengindikasikan sebuah hasil nol "0" dalam sebuah fungsi aritmatika atau logika.

**p. Bit 0(C)**

Merupakan *Bit Carry Flag*. *Bit* ini mengindikasikan sebuah *Carry* atau sisa dalam sebuah aritmatika atau logika.

### 3.1.2 Memori AVR Atmega



Gambar 3.4 Peta Memori Atmega<sup>[1]</sup>

Memori atmega terbagi menjadi tiga yaitu seperti pada Gambar 3.4:

#### 1. Memori Flash

Memori *flash* adalah memori ROM tempat kode-kode program berada.

Kata *flash* menunjukkan jenis ROM yang dapat ditulis dan dihapus secara elektrik. Memori *flash* terbagi menjadi dua bagian yaitu bagian aplikasi dan bagian *boot*. Bagian aplikasi adalah bagian kode-kode program aplikasi berada. Bagian *boot* adalah bagian yang digunakan khusus untuk *booting* awal yang dapat diprogram untuk menulis bagian aplikasi tanpa melalui *programmer/downloader*, misalnya melalui USART (*Universal Serial Asynchronous Receiver Transmitter*)..

#### 1. Memori Data

Memori data adalah memori RAM yang digunakan untuk keperluan program. Memori data terbagi menjadi empat bagian yaitu, 32 GPR (*General*



*Purpose Register*), 64 I/O *register*, *Additional I/O*, dan 1024 internal RAM. GPR adalah *register* khusus yang bertugas untuk membantu eksekusi program oleh ALU (*Arithmatic Logic Unit*), dalam instruksi *assembler* setiap instruksi harus melibatkan GPR. Dalam bahasa C biasanya digunakan untuk *variabel* global atau nilai balik fungsi dan nilai-nilai yang dapat memperingan kerja ALU. Dalam istilah *processor* komputer umumnya GPR dikenal sebagai “*cache memory*”. I/O *register* dan *Additional I/O register* adalah *register* yang difungsikan khusus untuk mengendalikan berbagai *pheripheral* dalam *microcontroller* seperti pin *Port*, *timer/counter*, USART dan lain-lain. *Register* ini dalam keluarga *microcontroller* MCS51 dikenal sebagi SFR (*Special Function Register*).

## 2. EEPROM

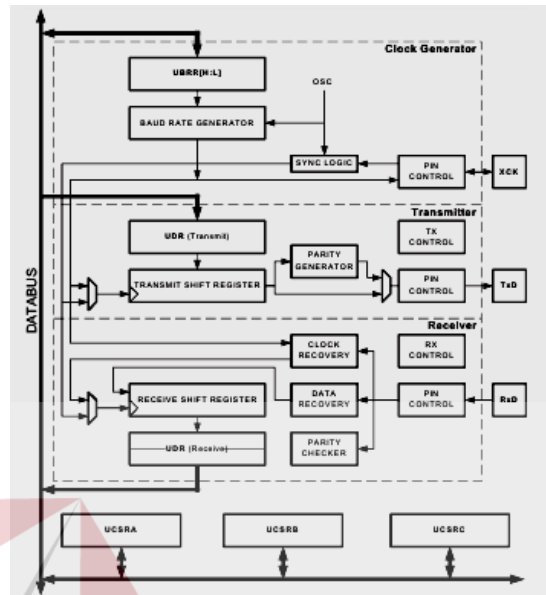
EEPROM adalah memori data yang masih dapat menyimpan data walaupun ketika *chip microcontroller* dalam keadaan mati (*off*), digunakan untuk keperluan penyimpanan data yang tahan terhadap gangguan catu daya.

### 3.1.2 Komunikasi Serial Pada Atmega 8

*Microcontroller* AVR Atmega 8 memiliki *Port* USART pada pin 2 dan pin 3 untuk melakukan komunikasi data antara *microcontroller* dengan *microcontroller* ataupun *microcontroller* dengan komputer. USART dapat difungsikan sebagai transmisi data sinkron dan asinkron. Sinkron berarti *clock* yang digunakan antara *transmitter* dan *receiver* satu sumber *clock*. Sedangkan

asinkron berarti *transmitter* dan *receiver* mempunyai sumber *clock* sendiri-sendiri.

USART terdiri dalam tiga blok yaitu *clock generator*, *transmitter*, dan *receiver*.



Gambar 3.5 Blok USART <sup>[1]</sup>

### 3.1.3 Clock Generator

*Clock generator* berhubungan dengan kecepatan transfer data (baud rate), *register* yang bertugas menentukan baud rate adalah *register* UBRR (USART Baud Rate Register).

Tabel 3.1 Baud Rate Atmega 8 <sup>[1]</sup>

Operating Mode	Equation for Calculating Baud Rate	Equation for Calculating UBRR Value
Asynchronous Normal Mode (U2X = 0)	$\text{Baud} = \frac{f_{osc}}{16(\text{UBRR}+1)}$	$\text{UBRR} = \frac{f_{osc}}{16(\text{BAUD}+1)} - 1$
Asynchronous Double Speed Mode (U2X = 1)	$\text{Baud} = \frac{f_{osc}}{8(\text{UBRR}+1)}$	$\text{UBRR} = \frac{f_{osc}}{8(\text{BAUD}+1)} - 1$

Synchrhronous Master Mode	$\text{Baud} = \frac{f_{osc}}{2(\text{UBRR}+1)}$	$\text{UBRR} = \frac{f_{osc}}{2(\text{BAUD}+1)} - 1$
---------------------------	--	--

Dimana:

1.  $f_{osc}$  adalah frekuensi *oscillator* yang digunakan.
2. BAUD adalah transfer *Bit* per detik.

### 3.1.4 USART Transmitter

USART *transmitter* berhubungan dengan data pada Pin TX. Perangkat yang sering digunakan adalah *register* UDR sebagai tempat penampungan data yang akan ditransmisikan. *Flag* TXC sebagai indikator bahwa data yang ditransmisikan telah sukses (*complete*), dan *flag* UDRE sebagai indikator jika UDR kosong dan siap untuk diisi data yang akan ditransmisikan lagi <sup>[1]</sup>.

### 3.1.5 USART Receiver

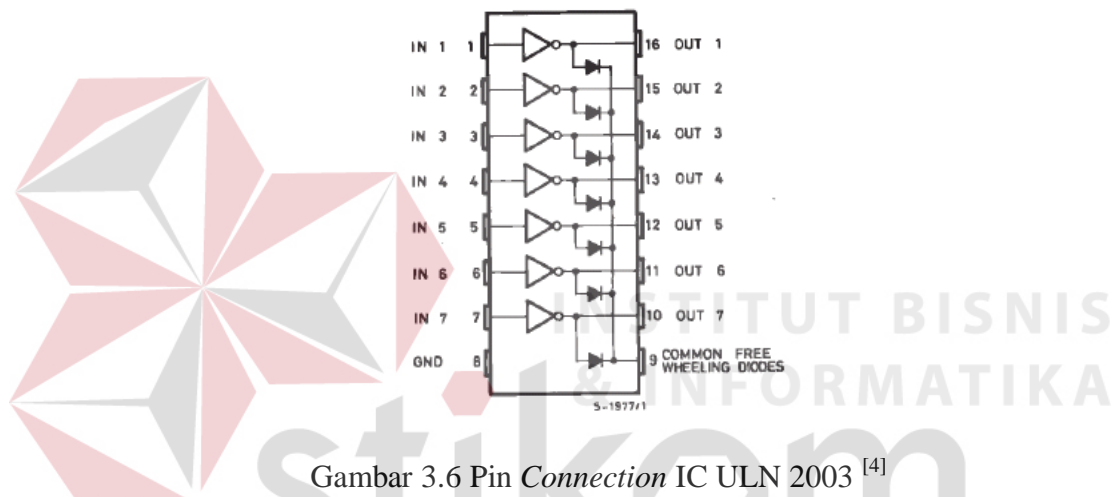
USART *receiver* berhubungan dengan penerimaan data dari Pin RX. Perangkat yang sering digunakan adalah *register* UDR sebagai tempat penampung data yang telah diterima, dan *flag* RXC sebagai indikator bahwa data telah sukses (*complete*) diterima <sup>[1]</sup>.

## 3.2 IC ULN2003

*Integrated Circuit* (IC) ULN2003 adalah IC bertegangan tinggi IC ULN 2003 memiliki 7 pasang kaki-kaki yang berfungsi sebagai masukan dan keluaran

sinyal, satu kaki yang berfungsi sebagai *Ground*, dan satu kaki *Common* Seperti Gambar 3.6. IC ini dapat meningkatkan arus yang dikirimkan melalui *parallel Port* yang hanya beberapa mA menjadi 500mA, sedangkan arus puncak yang mampu ditingkatkan oleh IC ULN2003 adalah 600A.

IC ULN 2003 sangat ideal untuk digunakan sebagai driver *seven segment* untuk mengontrol arus agar pada setiap *segment* didapatkan kecerahan yang dibutuhkan.



Gambar 3.6 Pin Connection IC ULN 2003 <sup>[4]</sup>

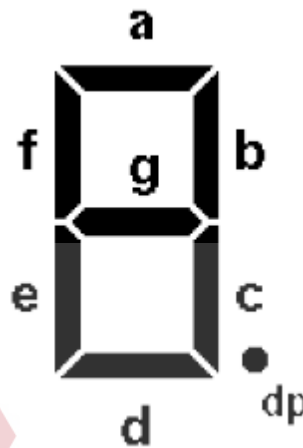
### 3.3 Seven Segment

*Seven segment display* adalah indikator penunjuk angka, terdiri dari tujuh buah LED (*Light emitting Diode*) yang disusun sehingga menjadi satu komponen. Dibawah ini Gambar 3.7 merupakan gambar *output* hasil tampilan *seven segment*:



Gambar 3.7 Tampilan *Seven Segment*

Setiap LED pada penampil *seven segment* diberi kode huruf untuk menyatakan LED mana yang nyala. Kode tersebut adalah a, b, c, d, e, f, g. sebagai contoh apabila yang menyala segmen a, b, g, e, dan d maka yang tampil adalah desimal 2. Berikut ini Gambar 3.7 adalah gambar posisi kode huruf pada penampil *seven segment*:



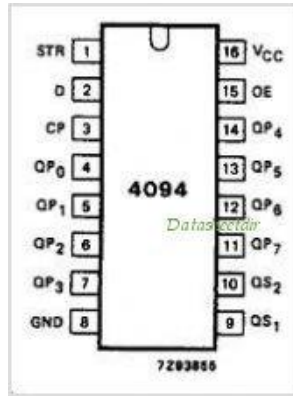
Gambar 3.8 Posisi Kode *Seven Segment Display*

Salah satu fungsi dari *seven segment* adalah untuk menampilkan sistem bilangan, penampil *seven segment* terdiri dari dua jenis yaitu *common anode* dan *common katode*. Pada *common anode* kaki-kaki anodanya terhubung ke *ground*, sebaliknya pada *common katode* kaki-kaki katodanya terhubung ke VCC.

### 3.4 IC *Shift Register* 4094

IC 4094 adalah IC *shift register* 8 bit yang memiliki *register latch* untuk setiap *bit* yang berguna untuk memindahkan data dari saluran *serial* ke saluran paralel dengan pergeseran *bit* Q0 sampai *bit* Q7 menuju *output*. *Output* paralel dapat dihubungkan langsung dengan jalur data umum. Data digeser ketika terjadi

perubahan sinyal *clock* dari *Low* ke *High*, selanjutnya data digeser dari *register* geser ke *register* penyimpanan, kemudian dengan memberikan logika *high* pada pin OE akan menggeser data dari register penyimpanan menuju register *output*.



Gambar 3.9 IC *Shift Register* 4094 <sup>[3]</sup>

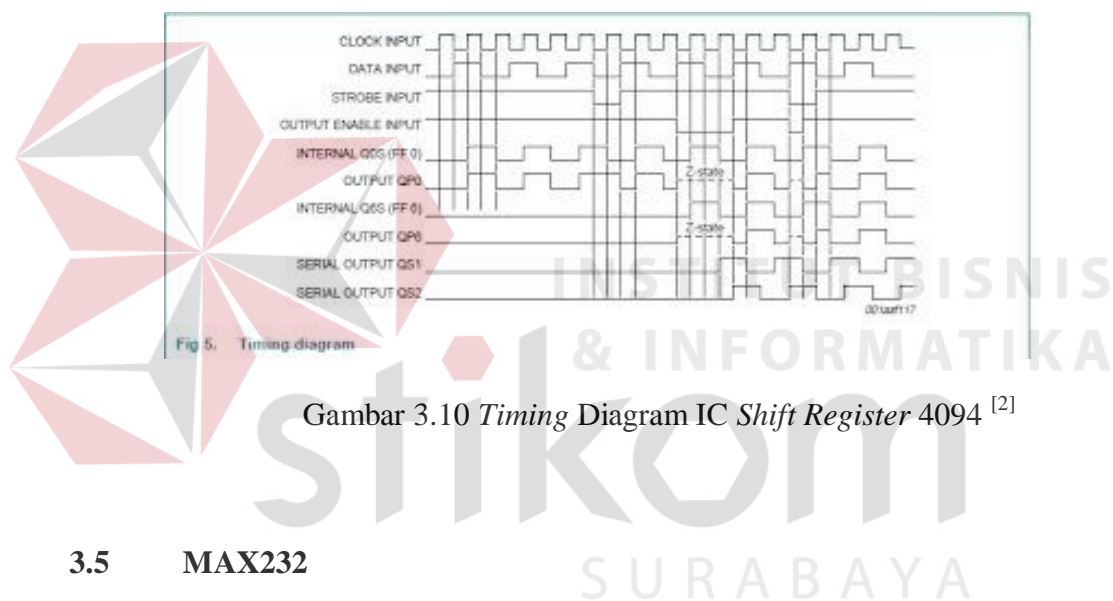
Tabel 3.2 Keterangan Pin IC 4094 <sup>[2]</sup>

No.	Nama Pin	Keterangan
1.	OE	<i>Output enable</i>
2.	QP0-QP7	<i>Output Paralel 0 – Output Paralel 7</i>
3.	D	<i>Input Data Serial</i>
4.	CP	<i>Clock Input</i>
5.	QS1-QS2	<i>Output Serial1 - Output Serial2</i>
6.	STR	<i>StrobeInput</i>
7.	VCC	V+
8.	GND	GND

### 3.4.2 Cara Kerja *Shift register*

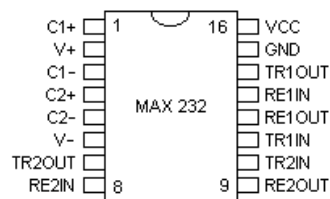
Data masuk secara serial melalui pin D (1). Pada IC *shift register* ini data masuk baru disimpan setelah terjadi *clock* jadi cara memasukkan data pada *shift register* ini adalah data masuk- *clock*- data masuk-*clock*-data masuk-*clock* begitu

seterusnya. Pin OE atau *Output Enable* digunakan untuk mengaktifkan *output* serial maupun *output* paralel. Logika 1 untuk *enable* dan logika 0 untuk *disable*. QP0 - QP7 adalah *output* paralel dari *shift register* ini sedangkan QS1 - QS2 adalah *output* serial dari *shift register* ini. Jika menggunakan lebih dari satu IC *shift register* maka pin data dari ic *shift register* selanjutnya dihubungkan ke *output* serial dari ic *shift register* sebelumnya. Untuk lebih jelasnya bisa dilihat pada Gambar 3.10 dibawah ini:



Gambar 3.10 Timing Diagram IC Shift Register 4094 <sup>[2]</sup>

### 3.5 MAX232



Gambar 3.11 Pin Max232 <sup>[3]</sup>

Kegunaan IC MAX232 adalah sebagai *driver*, yang akan mengkonversinilai tegangan atau kondisi logika *TTL* dari mikrokontroler agar

sesuai dengan level tegangan pada modem komunikasi yang digunakan. Max232 yang dipakai pada sistem ini memiliki 16 pin seperti pada Gambar 3.11 dengan tegangan sebesar 5 Volt.

Pada dasarnya max232 memerlukan komponen tambahan berupa kapasitor *eksternal* yang dipasang pada pin-pin tertentu. Kapasitor ini merupakan rangkaian baku yang berfungsi sebagai *charge pump* untuk menyuplai muatan sebagian pengubah tegangan, dimana nilai setiap kapasitor yang dipakai bernilai 1uF<sup>[3]</sup>.



Gambar 3.12 Altium *Designer* 6

Altium *designer* 6 adalah perusahaan perangkat lunak publik milik Australia yang menyediakan perangkat lunak desain elektronik berbasis PC untuk para *Engginer*. Didirikan di Tasmania, Australia tahun 1985, Altium sekarang memiliki kantor regional di Australia, China, Amerika Serikat, Eropa, dan Jepang,



dengan *reseller* di semua pasar utama lainnya. Perusahaan ini dikenal sebagai “Protel” sampai tahun 2001.

Altium *designer* merupakan salah satu *software* yang populer di dunia elektronika. Altium *designer* merupakan perkembangan dari protel versi sebelumnya Protel 99SE, yang sudah dilengkapi *database* yang terintegrasi. Altium *designer* 6 ini memiliki tampilan luar yang menarik, fitur-fitur yang baru dibandingkan dengan versi sebelumnya.

