

## A. MIKROKONTROLLER

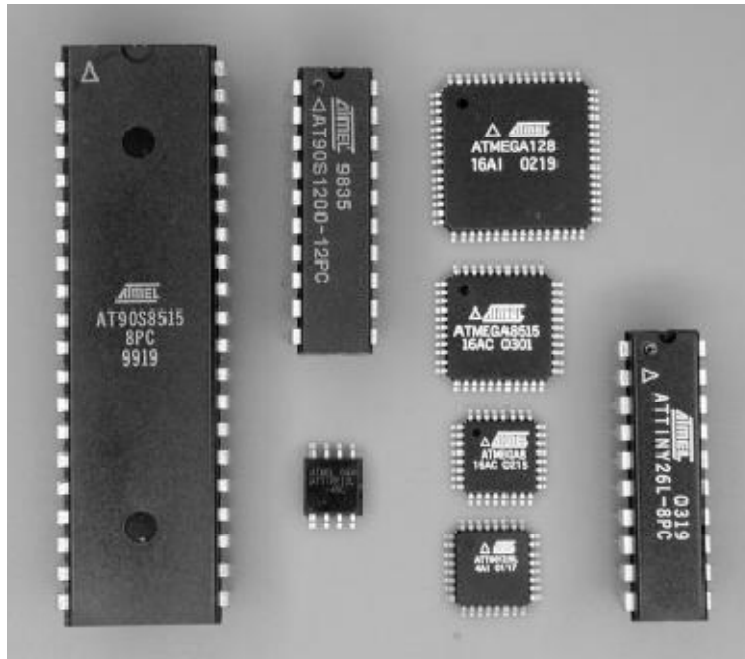
Sebenarnya apakah yang disebut dengan mikrokontroler? Sebuah kontroler digunakan untuk mengontrol suatu proses atau aspek-aspek dari lingkungan. Satu contoh aplikasi dari mikrokontroler adalah untuk memonitor rumah. Ketika suhu naik, kontroler membuka jendela dan sebaliknya. Pada masanya, kontroler dibangun dari komponen-komponen logika secara keseluruhan, sehingga menjadikannya besar dan berat. Setelah itu barulah dipergunakan mikroprosesor sehingga keseluruhan kontroler masuk kedalam PCB yang cukup kecil. Hingga saat ini masih sering kita lihat kontroler yang dikendalikan oleh mikroprosesor biasa (Zilog Z80, Intel 8088, Motorola 6809, dsb).

Proses pengecilan komponen terus berlangsung, semua komponen yang diperlukan guna membangun suatu kontroler dapat dikemas dalam satu keping. Maka lahirlah komputer keping tunggal (one chip microcomputer) atau disebut juga mikrokontroler. Mikrokontrolere adalah *suatu IC dengan kepadatan yang sangat tinggi, dimana semua bagian yang diperlukan untuk suatu kontroler sudah dikemas dalam satu keping*, biasanya terdiri dari:

- CPU (Central Processing Unit)
- RAM (Random Access Memory)
- EEPROM/EPROM/PROM/ROM
- I/O, Serial & Parallel
- Timer
- Interrupt Controller

Rata-rata mikrokontroler memiliki instruksi **manipulasi bit, akses ke I/O secara langsung dan mudah, dan proses interrupt yang cepat dan efisien**. Dengan kata lain mikrokontroler adalah " Solusi satu Chip" yang secara drastis mengurangi jumlah komponen dan biaya disain (harga relatif rendah).

Mikrokontroller adalah piranti elektronik berupa **IC (Integrated Circuit)** yang memiliki kemampuan manipulasi data (informasi) berdasarkan suatu urutan instruksi (program) yang dibuat oleh programmer. Mikrokontroller merupakan contoh suatu sistem komputer sederhana yang masuk dalam **kategori embedded komputer**. Dalam sebuah struktur mikrokontroller akan kita temukan juga komponen-komponen seperti: processor, memory, I/O, clock, dan lain-lain. Gambar dibawah ini merupakan beberapa bentuk mikrokontroller keluaran ATMEL.

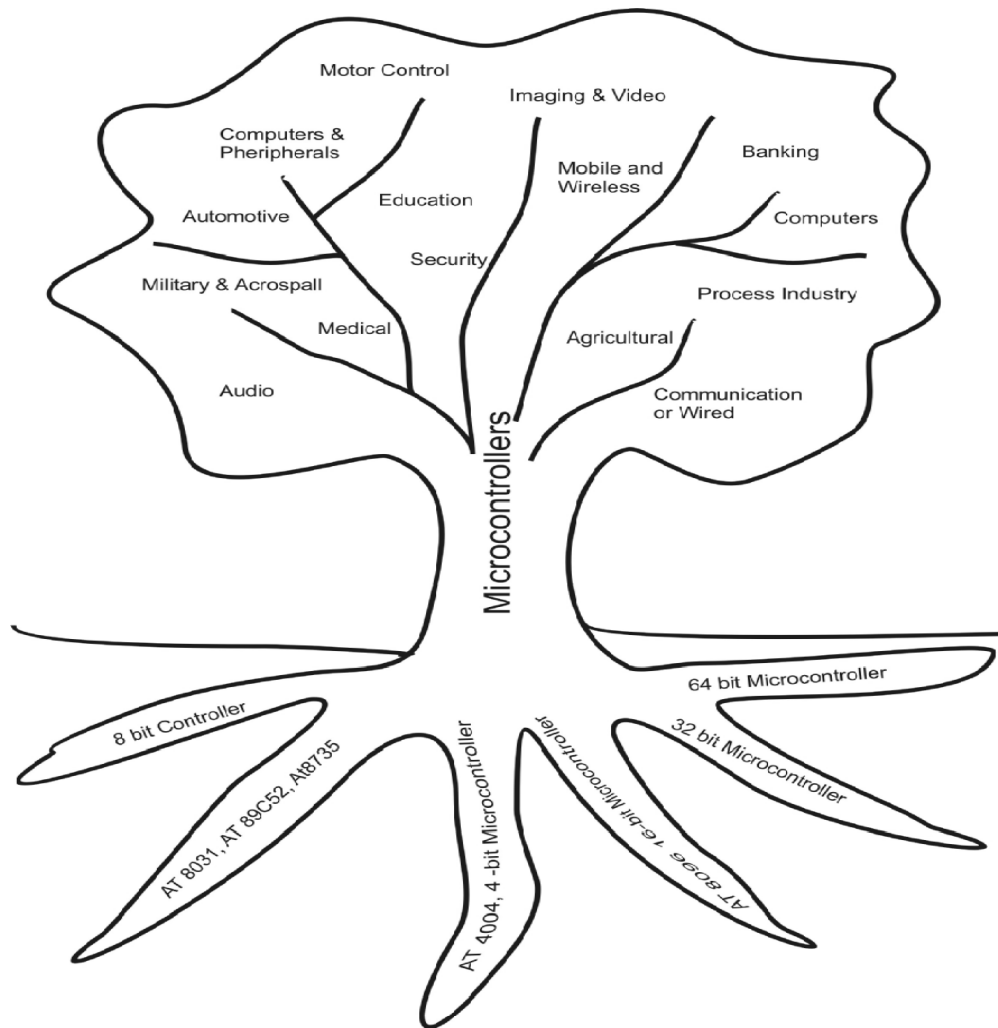


Gambar 1. contoh beberapa bentuk mikrokontroller ATMEL

## B. APLIKASI MIKROKONTROLLER

Selain sebagai sistem monitor rumah seperti diatas, mikrokontroler sering dijumpai pada peralatan rumah tangga (microwave oven, TV, stereo set dll), komputer dan perlengkapannya, mobil dan lain sebagainya. Pada beberapa penggunaan bisa ditemukan lebih dari satu prosesor didalamnya. Mikrokontroler biasanya digunakan untuk peralatan yang tidak terlalu membutuhkan kecepatan pemrosesan yang tinggi. Walaupun mungkin ada diantara kita yang membayangkan untuk mengontrol oven microwave dengan menggunakan sistem berbasis Unix, mengendalikan oven microwave dapat dengan mudah menggunakan mikrokontroler yang paling kecil.

Dilain pihak jika kita ingin mengendalikan rudal guna mengejar anjing tetangga yang selalu menyalak ditengah malam, kita akan memerlukan prosesor dengan kecepatan yang lebih tinggi. Sifat spesial dari mikrokontroler adalah *kecil dalam ukuran, hemat daya listrik serta fleksibilitasnya* menyebabkan mikrokontroler sangat cocok untuk dipakai sebagai pencatat/perekam data pada aplikasi yang tidak memerlukan kehadiran operator. Perhatikan gambar dibawah ini:



Gambar 2. Pohon Mikrokontroler

Mikrokontroler tersedia dalam beberapa pilihan, tergantung dari keperluan dan kemampuan yang diinginkan. Kita dapat memilih mikrokontroler 4, 8, 16 atau 32 bit. Disamping itu terdapat pula mikrokontroler dengan kemampuan komunikasi serial, penanganan keyboard, pemroses sinyal, pemroses video dll.

### C. MIKROKONTROLER ATMEL AVR

Pertimbangan Pemilihan Mikrokontroler Terdapat beberapa hal yang perlu dipertimbangkan dalam menentukan jenis mana yang akan dipergunakan dalam disain kita yaitu seperti berikut:

- Ketersediaan dan harga dari suatu development tools (Programmer, Emulator dan Simulator)

- Ketersediaan dokumentasi (Ref. Manual, Application notes, dan buku lainnya).
- Ketersediaan tempat bertanya.
- Ketersediaan komponen OTP, Mask, dan Programmable

Sebelum belajar lebih dalam tentang aplikasi mikrokontroler, ada baiknya membicarakan dulu tentang mikrokontroler yang digunakan. Pada workshop ini dipilih mikrokontroler jenis ATMEL AVR RISC dengan pertimbangan sebagai berikut:

- ATMEL AVR RISC memiliki fasilitas dan kefungisian yang lengkap dengan harga yang relatif murah.
- Kecepatan maksimum eksekusi instruksi mikrokontroler mencapai **16 MIPS (Million Instruction per Second)**, yang berarti hanya dibutuhkan 1 clock untuk 1 eksekusi instruksi.
- Konsumsi daya yang rendah jika dibandingkan dengan kecepatan eksekusi instruksi.
- Ketersediaan kompiler C (CV AVR) yang memudahkan user memprogram menggunakan bahasa C.

Berikut tabel perbandingan kecepatan processor dan efisiensi eksekusi beberapa mikrokontroler

Tabel 1. perbandingan kecepatan processor dan efisiensi

PROSESOR	COMPILED CODE SIZE	EXECUTION TIME (CYCLES)
AVR	46	335
8051	112	9.384
PIC16C74	87	2.492
68HC11	57	5.244

Dari tabel diatas dapat dilihat, ketika bekerja dengan kecepatan clock yang sama AVR 7 kali lebih cepat dibandingkan dengan PIC16C74, 15 kali lebih cepat daripada 68 HC11, dan 28 kali lebih cepat dibanding 8051. Dari kemampuan dan fasilitas yang dimiliki, AVR RISC cocok dipilih sebagai mikrokontroler untuk membangun bermacam-macam aplikasi embedded sistem.

#### D. PEMROGRAMAN ATMEL AVR

Ada 2 cara untuk memprogram mikrokontroler ini, menggunakan software AVR assembler yang berbasis pada bahasa assembly, dan menggunakan software bahasa tingkat tinggi seperti CV AVR (Code Vision AVR), WinAVR yang berbasis pada bahasa

C maupun BASCOM AVR yang menggunakan bahasa basic. Pada workshop ini digunakan CV AVR (Code Vision AVR) dengan pertimbangan kemudahan pembuatan program dari algoritma yang telah dibangun.

Workshop ini tidak menitikberatkan penggunaan bahasa C pada CV AVR, tapi lebih pada cara dan aplikasi dari mikrokontroller. Untuk itu peserta diharapkan membaca sendiri petunjuk pemakaian software ini. Berikut tampilan utama CVAVR,

## E. INPUT/OUTPUT AVR

Fasilitas input/output merupakan fungsi mikrokontroller untuk dapat menerima sinyal masukan (input) dan memberikan sinyal keluaran (output). Sinyal input maupun sinyal output adalah berupa data digital 1 (high, mewakili tegangan 5 volt) dan 0 (low, mewakili tegangan 0 volt). Mikrokontroller ATMEGA16 memiliki 4 buah PORT 8 bit bidirectional yang dapat difungsikan sebagai PORT input maupun PORT output yaitu **PORTA, PORTB, PORTC, dan PORT D**. Register digunakan untuk mengatur fungsi dari pin-pin pada tiap port. Register dapat dianalogikan sebagai kumpulan switch on/off yang digunakan untuk mengaktifkan fungsi apa yang akan dipakai dari port mikrokontroller. Pada setiap port pin terdapat 3 buah register 8 bit: **DDRxn, PORTxn, dan PINxn**.

Register DDRxn digunakan untuk menentukan arah dari pin yang bersangkutan. Jika DDRxn diberikan nilai 1 (high), maka pin digunakan sebagai output. Jika DDRxn diberikan nilai 0 (low), maka pin difungsikan sebagai input.

Register PORTxn digunakan untuk mengaktifkan pull-up resistor (pada saat pin difungsikan sebagai input), dan memberikan nilai keluaran pin high/low (pada saat difungsikan sebagai output). Konfigurasi PORTxn dan DDRxn dapat dilihat pada tabel dibawah,

Tabel 2. konfigurasi port pin

DDxn	PORTxn	PUD (in SFIOR)	I/O	Pull-up	Comment
0	0	X	Input	No	Tri-state (Hi-Z)
0	1	0	Input	Yes	Pxn will source current if ext. pulled low.
0	1	1	Input	No	Tri-state (Hi-Z)
1	0	X	Output	No	Output Low (Sink)
1	1	X	Output	No	Output High (Source)

Tri-state adalah kondisi diantara high dan low, atau biasa disebut dengan keadaan mengambang (floating). Kondisi tri-state sangat dihindari dalam dunia digital. Terlepas dari setting DDRxn, PINxn merupakan register yang berfungsi untuk mengetahui keadaan tiap-tiap pin pada mikrokontroller. Register ini sangat dibutuhkan untuk membaca keadaan pin pada saat difungsikan sebagai input.

## F. PEMROGRAMAN CODE VISION AVR

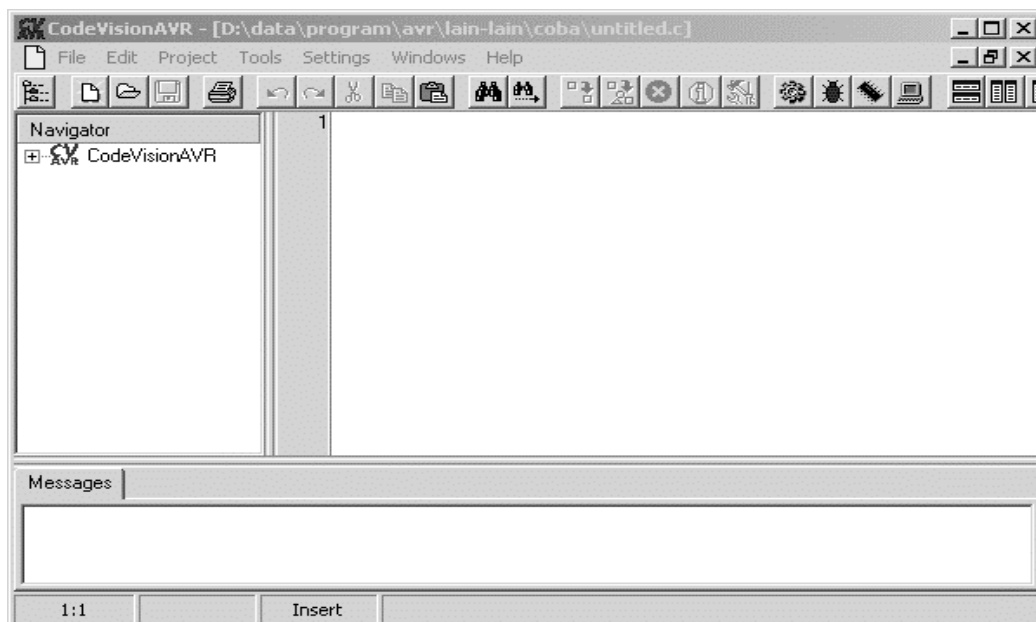
CodeVisionAVR pada dasarnya merupakan perangkat lunak pemrograman mikrokontroller keluarga AVR berbasis bahasa C. Ada tiga komponen penting yang telah diintegrasikan dalam perangkat lunak ini: Compiler C, IDE dan Program generator.

Berdasarkan spesifikasi yang dikeluarkan oleh perusahaan pengembangnya, Compiler C yang digunakan hampir mengimplementasikan semua komponen standar yang ada pada bahasa C standar ANSI (seperti struktur program, jenis tipe data, jenis operator, dan library fungsi standar-berikut penamaannya). Tetapi walaupun demikian, dibandingkan bahasa C untuk aplikasi komputer, compiler C untuk microcontroller ini memiliki sedikit perbedaan yang disesuaikan dengan arsitektur AVR tempat program C tersebut ditanamkan (*embedded*).

Khusus untuk library fungsi, disamping library standar (seperti fungsi-fungsi matematik, manipulasi String, pengaksesan memori dan sebagainya), CodeVisionAVR juga menyediakan fungsi-fungsi tambahan yang sangat bermanfaat dalam pemrograman antarmuka AVR dengan perangkat luar yang umum digunakan dalam aplikasi kontrol. Beberapa fungsi library yang penting diantaranya adalah fungsi-fungsi untuk pengaksesan LCD, komunikasi I<sup>2</sup>C, IC

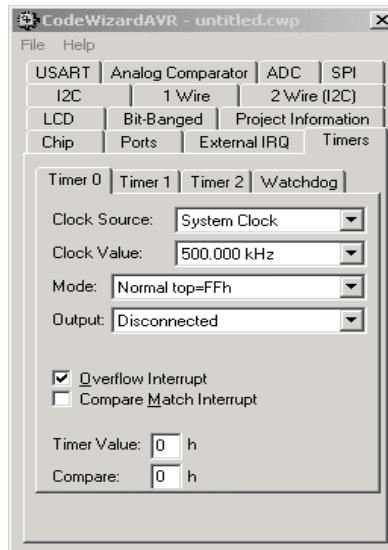
RTC (*Real time Clock*), sensor suhu LM75, SPI (*Serial Peripheral Interface*) dan lain sebagainya.

Untuk memudahkan pengembangan program aplikasi, CodeVisionAVR juga dilengkapi IDE yang sangat *user friendly* (gambar 3). Selain menu-menu pilihan yang umum dijumpai pada setiap perangkat lunak berbasis Windows, CodeVisionAVR ini telah mengintegrasikan perangkat lunak *downloader* (*In System Programmer*) yang dapat digunakan untuk mentransfer kode mesin hasil kompilasi kedalam sistem memori microcontroller AVR yang sedang diprogram.



Gambar 3. IDE perangkat lunak CodeVisionAVR

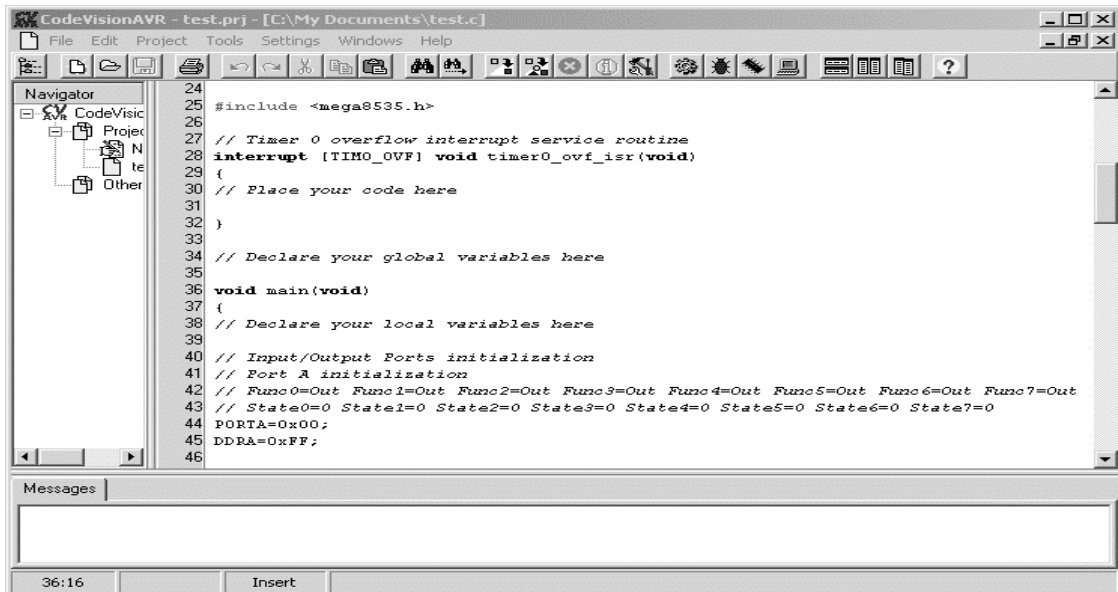
Selain itu, CodeVisionAVR juga menyediakan sebuah *tool* yang dinamakan dengan *Code Generator* atau *CodeWizardAVR*. Secara praktis, *tool* ini sangat bermanfaat membentuk sebuah kerangka program (*template*), dan juga memberi kemudahan bagi programmer dalam penginisialisasian register-register yang terdapat pada microcontroller AVR yang sedang diprogram. Dinamakan *Code Generator*, karena perangkat lunak CodeVision ini akan membangkitkan kode-kode program secara otomatis setelah fase inisialisasi pada jendela *CodeWizardAVR* selesai dilakukan.



Gambar 4. Code Generator CVAVR mikrokontroler AVR.

Gambar 5 dibawah memperlihatkan beberapa penggal baris kode program yang dibangkitkan secara otomatis oleh CodeWizardAVR. Secara teknis, penggunaan *tool* ini pada dasarnya hampir sama dengan *application wizard* pada bahasa-bahasa pemrograman Visual untuk komputer (seperti Visual C, Borland Delphi, dan sebagainya). Disamping versi yang komersil, Perusahaan Pavel Haiduc juga mengeluarkan CodeVisionAVR versi Demo yang dapat didownload dari internet secara gratis (lihat alamat URL: <http://www.hpinfotech.ro>) Dalam versi ini, memori flash yang dapat diprogram dibatasi maksimal 2K, selain itu tidak semua fungsi library yang tersedia dapat dipanggil secara bebas.

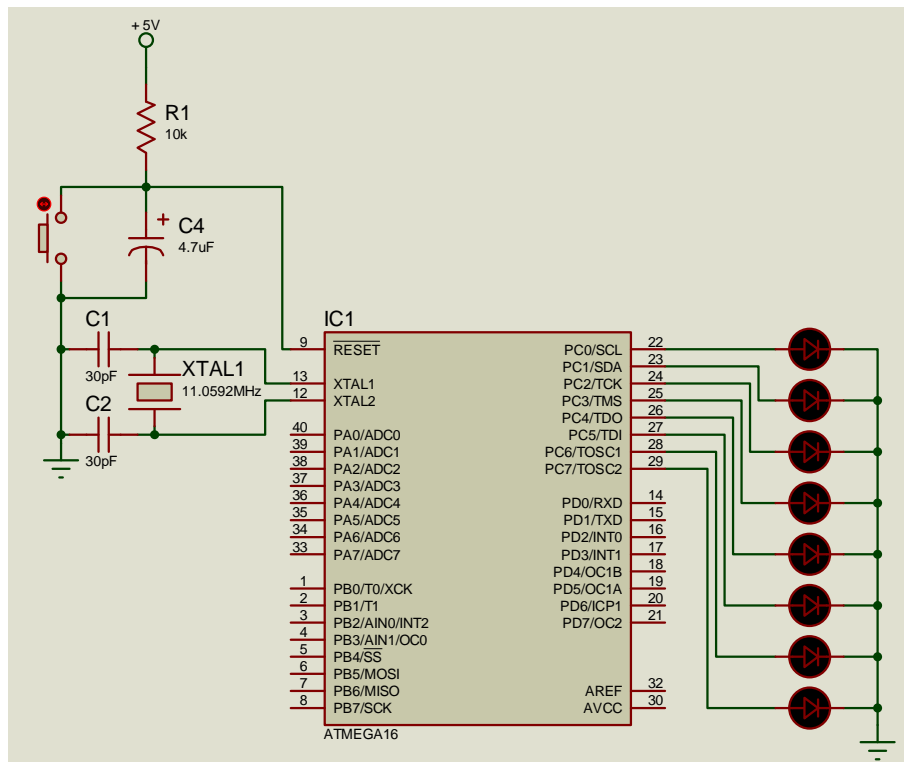




Gambar 5. Kode yang dibangkitkan oleh generator program

## G. CONTOH RANGKAIAN DAN PROGRAM

### Rangkaian 1



Gambar 6. Rangkaian Input/output

### Program 1

```
#include <mega16.h>
```

```

#include <delay.h>

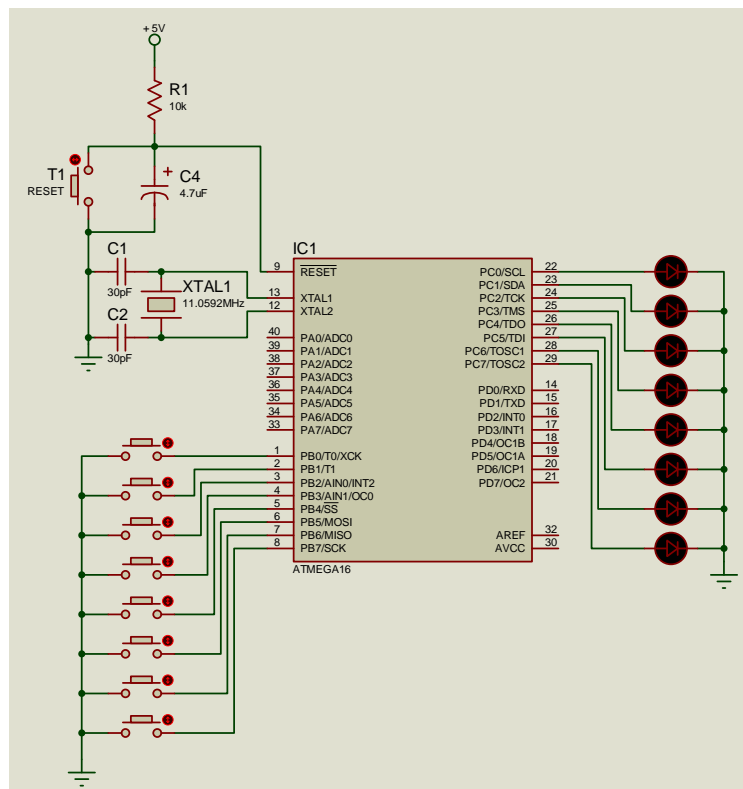
unsigned char out1=0x01;
void main(void)
{

int i;
DDRC=0xFF;
PORTC=0x00;

while (1)
{
for(i=1;i<=8;i++)
{
PORTC=out1;
delay_ms(200);
out1<<=1;
}
out1=0x01;
};
}

```

## Rangkaian 2



Gambar 7. Rangkaian Input/output

## Program 2

```

#include <mega16.h>
// Declare your global variables here
void main(void)

```

```
{
PORTB=0xFF;
DDRB=0xFF;

PORTC=0x00;
DDRC=0xFF;
while (1)
{
if (PINB.0==0) {PORTC=0x01;}
if (PINB.1==0) {PORTC=0x02;}
if (PINB.2==0) {PORTC=0x04;}
if (PINB.3==0) {PORTC=0x08;}
if (PINB.4==0) {PORTC=0x10;}
if (PINB.5==0) {PORTC=0x20;}
if (PINB.6==0) {PORTC=0x40;}
if (PINB.7==0) {PORTC=0x80;}
else {}
};
}
```