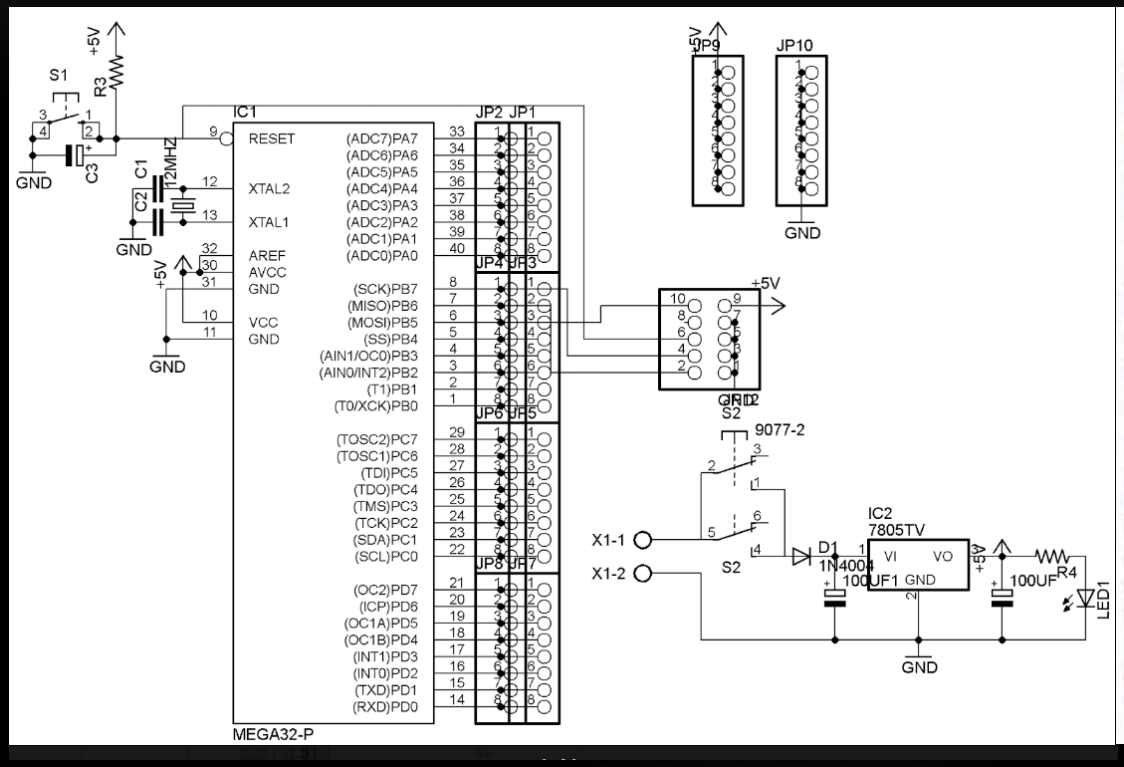
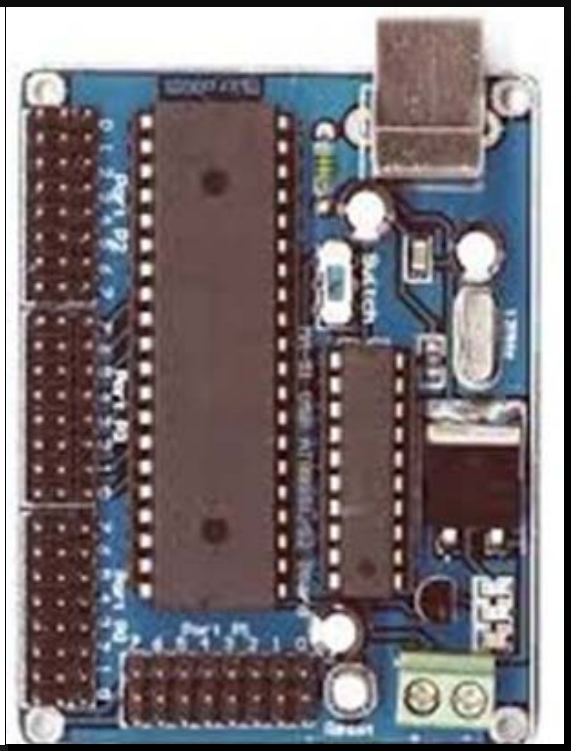
SISTEM MIKROPROSESOR DAN MIKROKONTROLER

4 SKS (3 TEORI & 1 PRAKTEK)
RUANGAN B2.2

MINIMUM SISTEM MIKROKONTROLLER





PENGENALAN CVAVR

- Pengenalan CVAVR
- Pengenal (identifier)
- Tipe Data
- Variabel
- Konstanta
- Komentar
- Preprosesor
- Pernyataan
- Operator
- – Operator Aritmatika
- – Operator Relasional
- – Operator Logika
- – Operator Bit
- Fungsi Pustaka
- Pernyataan If
- Pernyataan If ... Else
- Pernyataan If Bersarang
- Pernyataan Switch
- Pernyataan While
- Pernyataan Do ... While
- Pernyataan For
- Fungsi
- Bentuk Dasar Bahasa C
- Pemangilan Bahasa Assembler

TIPE DATA

Type	Size (Bits)	Range
bit	1	0,1
char	8	-128 to 127
unsigned char	8	0 to 255
signed char	8	-128 to 127
int	16	-32768 to 32767
short int	16	-23768 to32767
unsigned int	16	0 to 65535
signed int	16	-32768 to 32767
long int	32	-2147483648 to 2147483647
unsigned long int	32	0 to 4294967295
signed long int	32	-2147483648 to 2147483647
float	32	±1.175e-38 to ±3.402e38
double	32	±1.175e-38 to ±3.402e38

VARIABEL

- Variabel merupakan tempat menyimpan data di dalam memori yang isinya dapat diubah.
- Pendeklarasian konstanta :

  Tipe_data nama_variabel = nilai ,

  Contoh : char PINA = 0x20;

- Variabel dibagi menjadi dua bagian, yaitu:

– Variabel Global variabel yang dapat dikenali oleh semua fungsi yang ada di dalam program dan selalu dideklarasikan di luar fungsi.

Contoh : char DDA;

– Variabel Lokal Variabel yang hanya dapat dikenali oleh fungsi tertentu. dideklarasikan hanya di dalam fungsi tertentu saja,

contoh : void main(void) { Char PORTB = 0xFF; }

KONSTANTA

- Konstanta merupakan sebuah tempat untuk menyimpan data di dalam memori dimana data dari konstanta tidak dapat diubah. Pendeklarasi konstanta :

  Const tipe_data nama_konstanta =Nilai

  Contoh : Const char PINB = 0xFF;

KOMENTAR > Tulisan yang tidak dianggap sebagai bagian dari program sehingga tidak akan dieksekusi oleh compiler

• Digunakan sebagai penjelas, informasi ataupun keterangan dari program yang dibuat untuk mempermudah memahami program

• Untuk membuat komentar satu baris digunakan tanda “//” atau atau untuk beberapa baris digunakan tanda “/*” dan diakhiri dengan “*/”.

- PREPROSESOR

Untuk mendeklarasikan atau mendefinisikan prosesor yang digunakan di dalam program

Dengan preprosesor ini maka pendeklarasian register-register dan penamaannya dilakukan oleh file lain.

Cara penulisan: #include <nama_preprosesor>

Contoh : #include <mega16.h> // otomatis saat setting awal

- PERNYATAAN

satu buah atau satu blok intruksi lengkap yang berdiri sendiri.

Contoh pernyataan: PORTA = 0xF0;

Contoh blok pernyataan: { PORTB =0x00; PORTC = 0xFF; }

- OPERATOR

simbol khusus yang merepresentasikan perhitungan sederhana seperti penambahan dan perkalian.

Nilai yang digunakan oleh operator disebut operand. • Ekspresi merupakan kombinasi dari operand dan operator

OPERATOR

- simbol khusus yang merepresentasikan perhitungan sederhana seperti penambahan dan perkalian.
- Nilai yang digunakan oleh operator disebut operand.
- Ekspresi merupakan kombinasi dari operand dan operator

Operator Aritmatika

- Operator aritmatika adalah operator yang digunakan untuk melakukan perhitungan aritmatika.

Operator	Keterangan
+	Operator untuk operasi penjumlahan
-	Operator untuk operasi pengurangan
*	Operator untuk operasi perkalian
/	Operator untuk operasi pembagian
%	Operator untuk operasi sisa pembagian

- Operator Relasional

disebut juga operator pembanding adalah operator yang digunakan untuk membandingkan 2 buah data

Hasil perbandingan dari 2 buah data berupa pernyataan benar (‘1’) atau salah (‘0’) tidak berupa hasil yang lain.

Operator	Contoh	Keterangan
==	x == y	Bernilai benar jika kedua data sama dan bernilai salah jika keduanya bernilai beda
!=	x != y	Bernilai benar jika kedua data tidak sama dan bernilai salah jika kedua data sama
>	x > y	Bernilai benar jika nilai x lebih besar dari pada y dan bernilai salah jika y lebih besar dari x
<	x < y	Bernilai benar jika x lebih kecil daripada y dan bernilai salah jika x bernilai lebih besar
>=	x >= y	dari y Bernilai benar jika x lebih besar sama dengan y dan bernilai salah jika sebaliknya
<=	x <= y	Bernilai benar jika x lebih kecil sama dengan y dan bernilai salah jika sebaliknya

- Operator Logika

digunakan untuk membentuk suatu logika atau dua buah kondisi atau lebih.

Operator	Keterangan
&&	Operator untuk logika AND
||	Operator untuk logika OR
!	Operator untuk logika NOT

- Operator Bit

Operator logika yang bekrja pada level bit.

Dalam operator logika data yang dihasilkan adalah data yang berupa bilangan biner.

Operator	Keterangan
&	Operator untuk operasi AND level bit (biner)
!	Operator untuk operasi OR level bit (biner)
^	Operator untuk operasi XOR level bit (biner)
~	Operator untuk operasi NOT level bit (biner)
<<	Operator untuk operasi geser kiri pada data biner
>>	Operator untuk operasi geser kanan pada data biner

- FUNGSI PUSTAKA

Bahasa CVAVR memiliki sejumlah fungsi pustaka yang berada pada file-file tertentu Disediakan untuk menangani berbagai hal dengan cara pemangilan fungsi-fungsi yang telah dideklarasikan di dalam file tersebut.

Sintaks untuk menggunakan fungsi pustaka adalah sebagai berikut: #include <nama_fungsi_pustaka.h>

Dalam CodeVisionAVR telah disertakan fungsi pustaka yang mendukung pemrograman mikrokontroler antara lain:

· Fungsi tipe karakter (ctype.h)
· Fungsi standar I/O (stdio.h)
· Fungsi matematika (math.h)
· Fungsi string (string.h)
· Fungsi konversi BCD (bcd.h)
· Fungsi konversi akses memori (mem.h)
· Fungsi waktu tunda (delay.h)
· Fungsi LCD (lcd.h)
· Fungsi I2C (i2c.h)
· Fungsi SPI (spi.h)
· Fungsi RTC (ds1302.h, ds1307.h)
· Fungsi Sensor suhu LM75, DS1621, dll (lm75.h, ds162.h)

PERNYATAAN IF

- Pernyataan if digunakan untuk melakukan pengambilan keputusan terhadap dua buah kemungkinan yaitu mengerjakan suatu blok pernyataan atau tidak, jika dan hanya jika persyaratannya terpenuhi.

if (kondisi) { // pernyataan }; // Artinya adalah pernyataan akan dijalankan jika kondisi terpenuhi.

- Perhatikan contoh :

```
if (a<0x50) { PORTC=0x55; };
```

contoh ini PORTC akan dikirim data 0x55 jika nilai a lebih kecil dari 0x50.

PERNYATAAN IF ... ELSE

- Pernyataan If ... else digunakan untuk meakukan pengambilan keputusan terhadap dua buah kemungkinan, kedua kemungkinan tersebt adalah mengerjakan pernyataan satu atau mengerjakan pernyataan yang lain.

```
if (kondisi) { // pernyataan a }
        else { // pernyataan b };
```

Artinya adalah pernyataan a akan dijalankan jika kondisi terpenuhi dan pernyataan b akan dijalankan jika kondisi tidak terpenuhi

PERNYATAAN IF BERSARANG

- Pernyataan if bersarang (nested If) adalah pernyataan if maupun if ... else dimana di dalam blok pernyataan yang akan dikerjakan terdapat pernyataan if atau if ... else lagi.

```
If (kondisi_1)
        { If (kondisi_2)
         { //blok pernyataan } }
         else
{ If (kondisi_3) { // pernyataan } }
```

PERNYATAAN SWITCH

- Pernyataan switch digunakan untuk melakukan pengambilan keputusan terhadap banyak kemungkinan
- Pernyataan switch – case digunakan jika terjadi banyak percabangan.

```
.....
switch (ekspresi)
{ case konstanta1: Pernyataan1; break;
   case konstanta2: pernyataan2 break;
......
case konstantaN: pernyataanN break; }
```

- PERNYATAAN SWITCH

```
switch (a)
{ case 1 : PORTC=0x01 ; break;
case 2 : PORTC=0x02; break;
case 3 : PORTC=0x04; break; }
```

PORTC akan dikirim data 0x01 jika nilai a=1,

PORTC akan dikirim data 0x02 jika nilai a=2 dan

PORTC akan dikirim data 0x04 jika nilai a=3. Perhatikan contoh dibawah ini:

- PERNYATAAN WHILE

Pernyataan while digunakan untuk pengulangan sebuah pernyataan atau blok pernyataan secara terus menerus selama kondisi tertentu masih terpenuhi.

Bentuk perulangan while adalah sebagai berikut:

```
while (kondisi) { pernyataan-pernyataan; }
```

Jika kondisi memenuhi (bernilai true) maka pernyataan-pernyataan dibawahnya akan dijalankan hingga selesai, kemudian akan menguji kembali kondisi diatas.

PERNYATAAN WHILE

Perhatikan contoh dibawah ini:

```
i=1;
a=1;
while (i<50) {
        a=a*2;
        PORTC=a;
        i++ ; };
```

PERNYATAAN DO ... WHILE

- Pernyataan Do ... While digunakan sama seperti pengunaan pernyataan dari while.
- Bentuk perulangan ini kebalikan dari while – do, yaitu pernyataan dilakukan terlebih dahulu kemudian diuji kondisinya

```
do
{
pernyataan-pernyataan;
}
while (kondisi);
```

Perhatikan contoh berikut ini:

```
 i=1;
a=1;
do
{
        a=a*2;
        PORTC=a;;
        i++ ;
 } while (i<50);
```

PERNYATAAN FOR

- Pernyataan for digunakan juga untuk melakukan pngulangan sebuah pernyataan atau blok pernyataan, tetapi beberapa kali jumlah pengulangannya dapat ditentukan secara lebih spesifik.
- Pernyataan for akan melakukan perulangan berapa kali sesuai yang diinginkan.
- Struktur penulisan perulangan for

```
For (mulai ; kondisi ; penambahan/pengurangan)
        {
            Pernyataan-pernyataan;
        };
```

- Perhatikan contoh dibawah:

```
a=1;
for (i=1; i<50; i++) {
a=a*2; PORTC=a;
};
```

Contoh program di samping akan melakukan perulangan 49 kali, yaitu dari 1 hingga 50 dengan penambahan 1 (i++).
Hasilnya PORTC akan dikirim data 1, kemudian data 2,4,8, …..
Sesuai dengan persamaan a=a*2

FUNGSI

- Fungsi adalah kumpulan pernyataan-pernyataan yang dikemas dalam satu wadah kemudian diberi nama dan selanjutnya dapat dipangil beberapa kali dalam sebuah program.
- Fungsi dapat digunakan untuk memecah logika program menjadi lebih kecil sehingga akan lebih memudahkan untuk mengelola dan memahami alur logika program pada saat menulis ataupun mengoreksi program.
- Fungsi dapat dibedakan berdasarkan hasil fungsi tersebut menjadi dua, yaitu:
  - Fungsi dengan nilai balik
  - Fungsi tanpa nilai balik

Fungsi dengan nilai balik

- Fungsi ini akan menghasilkan data keluaran baru setelah fungsi ini dipanggil/ dieksekusi oleh program.
- Bentuk umum dari fungsi ini adalah sebagai berikut:

```
tipe_data nama_fungsi (tipe_data_1 parameter_1, ...)
{
Pernyaaan_1;
Pernyataan_2;
..... }
```

Contoh :

```
int jarak (int data_1, int data_2)
{
hasil = data_1 / data_2;
return hasil;
}
```

Fungsi tanpa nilai balik

- Fungsi dapat di sebut dengan fungsi tanpa nilai balik jika fungsi tersebut apabila dipangil tidak menghasilkan nilai.
- Bentuk umum fungsi ini adalah sebagai berikut:

```
tipe_data nama_fungsi (tipe_data_1 parameter_1, ....)
 {
 Pernyaaan_1;
Pernyataan_2;
..... }
```

Contoh :

```
void port (char A, char B, char C, char D)
 {
DDRA = A;
DDRB = B;
DDRC = C;
DDRD = D;
}
```

BENTUK DASAR BAHASA C

- Sebuah program dalam bahasa C harus memiliki sebuah fungsi utama.
- Fungsi utama atau sering disebut dengan fungsi main memiliki kerangka program sebagai berikut:

```
void main(void)
{
// pernyataan-pernyataan
}
```

- Fungsi utama merupakan fungsi yang pertama kali akan dieksekusi oleh program, walaupun di dalam program tersebut terdapat beberapa fungsi.
- Kedudukan fungsi-fungsi yang lain lebih rendah dari fungsi utama, sehingga fungsi selain fungsi utama dapat dipanggil/ digunakan di dalam fungsi utama.

TERIMA KASIH