

SISTEM MIKROPROSESOR

Mochamad Fajar Wicaksono, S.Kom., M.Kom.

ATURAN PERKULIAHAN

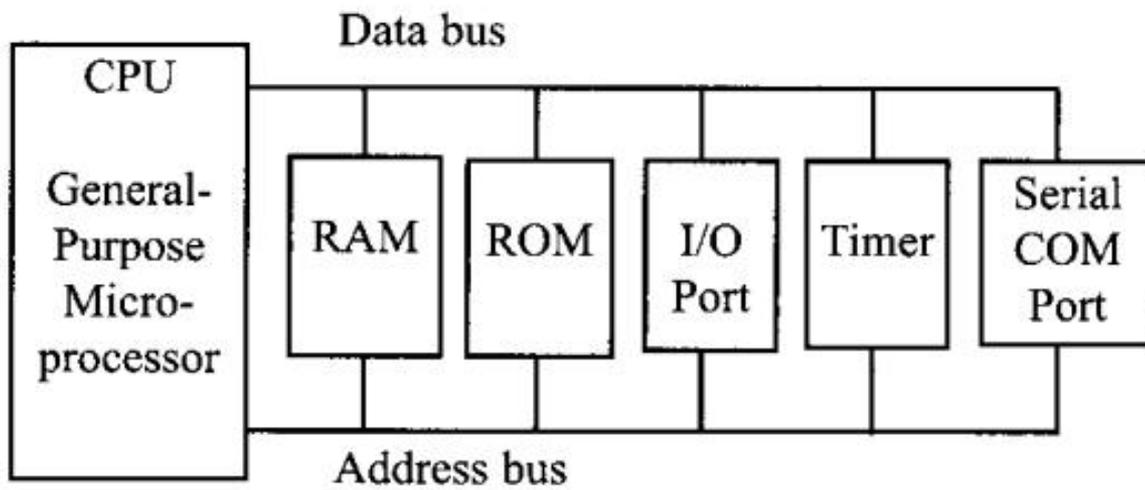
- Toleransi keterlambatan **15 menit**.
- 3 kali tidak masuk = **E**.
- Tugas, kuis, uts dan uas tidak boleh plagiat. **Plagiat = E**.

PENILAIAN

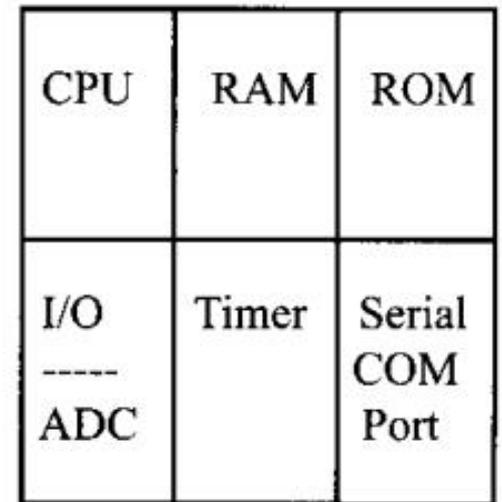
Komponen Penilaian	Porsi terhadap NA
QUIZ	30%
UTS	30%
UAS	40%

BUKU ACUAN

- Syahrul, 2012. Mikrokontroler AVR ATmega85325, Informatika, Bandung.
- Muhammad Ali Mazidi, Sarmad Naimi, Sepehr Naimi, 2011. The AVR Microcontroller and Embedded System. Prentice Hall



(a) General-Purpose Microprocessor System



(b) Microcontroller

Mikroprosesor dan Mikrokontroler

- **Mikroprosesor** adalah suatu unit pemroses yang berdiri sendiri, di dalamnya belum terdapat memori, IO, dan peripheral lainnya.
- **Mikrokontroler** adalah mikroprosesor yang telah dilengkapi dengan memori, IO, dan peripheral dalam **satu chip**

ALASAN UTAMA

- **Kebutuhan pasar (market need)** : kebutuhan yang luas dari produk-produk elektronik akan perangkat pintar sebagai pengontrol dan pemroses data
- **Perkembangan teknologi baru**: perkembangan teknologi semikonduktor yang memungkinkan pembuatan chip dengan kemampuan komputasi yang sangat cepat, bentuk yang semakin mungil, dan harga yang semakin murah.

APLIKASI MIKROKONTROLER

- Mainan anak-anak
- Perangkat elektronik rumah tangga
- Perangkat pendukung otomotif
- Peralatan industri
- Peralatan telekomunikasi
- Peralatan medis dan kedokteran, sampai dengan pengendali robot serta persenjataan militer.

KEUNGGULAN

Terdapat beberapa keunggulan yang diharapkan dari alat-alat yang berbasis mikrokontroler (microcontroller-based solutions) :

- **Kehandalan tinggi (high reliability)** dan **kemudahan integrasi** dengan komponen lain (high degree of integration)
- **Ukuran** yang semakin dapat diperkecil (reduced in size)
- **Penggunaan komponen sedikit** (reduced component count) yang juga akan menyebabkan biaya produksi dapat semakin ditekan (lower manufacturing cost)
- **Waktu pembuatan lebih singkat** (shorter development time) sehingga lebih cepat pula dijual ke pasar sesuai kebutuhan (shorter time to market)
- **Konsumsi daya yang rendah** (lower power consumption)

KRITERIA

Efisien penanganannya dan murah

Kecepatan, kemasan, konsumsi daya, berapa banyak RAM dan ROM didalam Chip

Jumlah pin I/O dan timer

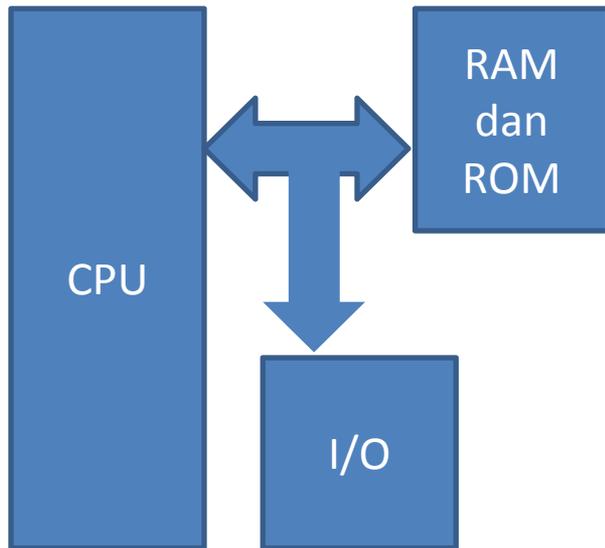
Kemudahan meningkatkan ke kinerja yang lebih tinggi atau versi konsumsi daya yang lebih rendah

Harga per unit

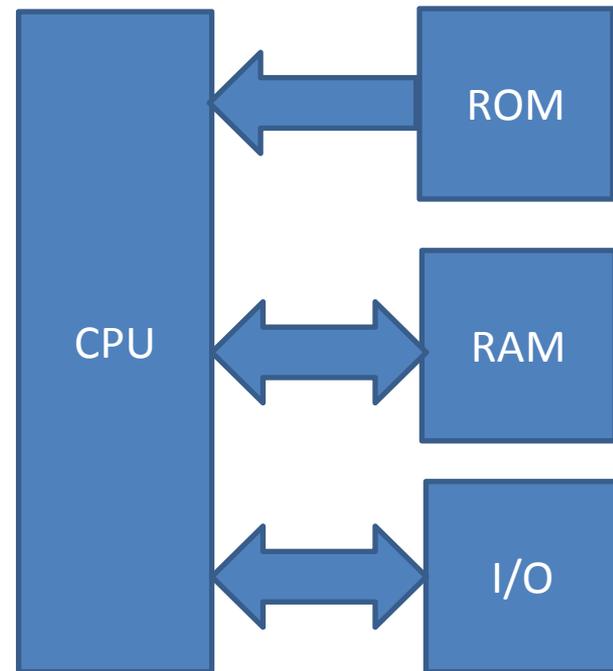
KRITERIA

- Seberapa mudah mengembangkan produk-produk seputar itu. Ketersediaan sebuah assembler, debugger, compiler, emulator serta dukungan teknis.
- Ketersediaan yang siap dalam kuantitas kebutuhan untuk saat ini dan untuk masa yang akan datang.

Arsitektur Von Neumann VS Architektur Harvard



Arsitektur Von Neumann



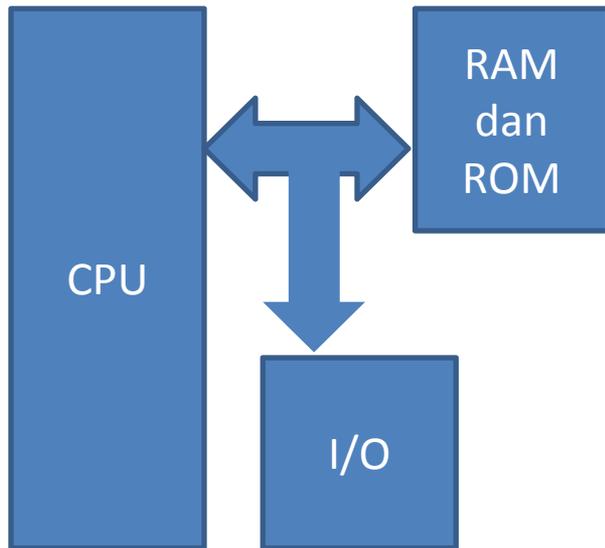
Arsitektur Harvard

Arsitektur Von Neumann VS Arsitektur Harvard

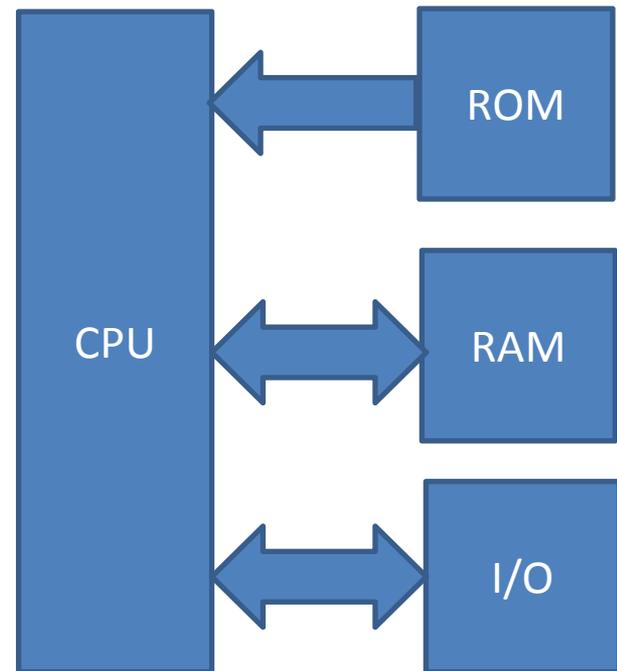
Arsitektur Von Neumann

- Menggunakan bus tunggal untuk mengakses memori (RAM dan ROM). Dengan kata lain menggunakan bus secara bergantian ketika akan mengakses instruksi dan data.
- Bandwidth program harus sama dengan bandwidth data (jika memori data 8 bit maka program juga 8 bit).

Arsitektur Von Neumann VS Architektur Harvard



Arsitektur Von Neumann



Arsitektur Harvard

Arsitektur Von Neumann VS Arsitektur Harvard

Arsitektur Harvard

- Menerapkan pemisahan secara fisik antara memori program dengan memori data.
- User dapat meletakkan instruksi pada alamat yang dikehendaki, sehingga **user dapat mengatur sendiri peta memori dan efektifitasnya.**
- Pengalamatan RAM dan ROM terpisah.
- Mikrokontroler AVR merupakan salah satu sistem komputer yang menggunakan arsitektur harvard.

BERBAGAI SERI MIKROKONTROLER AVR

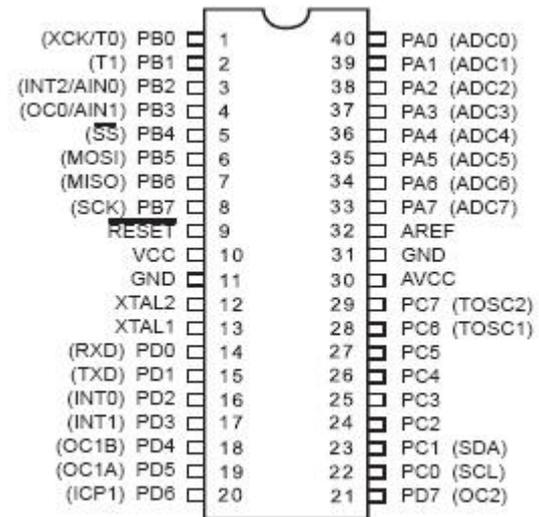
Seri	Flash (KBytes)	RAM (Bytes)	EEPROM (KBytes)	Pin I/O	Timer 16-bit	Timer 8-bit	UART	PWM	ADC 10-bit	SPI	ISP
ATmega8	8	1024	0.5	23	1	1	1	3	6/8	1	Ya
ATmega8535	8	512	0.5	32	2	2	1	4	8	1	Ya
ATmega16	16	1024	0.5	32	1	2	1	4	8	1	Ya
ATmega162	16	1024	0.5	35	2	2	2	6	8	1	Ya
ATmega32	32	2048	1	32	1	2	1	4	8	1	Ya
ATmega128	128	4096	4	53	2	2	2	8	8	1	Ya
ATtiny12	1	-	0.0625	6	-	1	-	-	-	-	Ya
ATtiny2313	2	128	0.125	18	1	1	1	4	-	1	Ya
ATtiny44	4	256	0.25	12	1	1	-	4	8	1	Ya
ATtiny84	8	512	0.5	12	1	1	-	4	8	1	Ya

Mikrokontroler AVR

- Merupakan mikrokontroler 8 bit.
- Program dan data ditempatkan di memory yang berbeda. Program ditempatkan di memori program, sedangkan data di memori data
- Memori program merupakan flash memory yg bisa dihapus-tulis berkali-kali, bersifat non-volatile (isi tidak hilang kalau catu daya dicabut)
- Memori data berupa SRAM (static ram)
- Jenisnya bermacam-macam:
 - AT90Sxxxx AVR klasik
 - ATtiny AVR dengan kemasan kecil, peripheral terbatas
 - ATmega AVR dengan peripheral lebih lengkap

Fitur ATmega8535

- Mikrokontroler AVR 8 bit
- Program memori : 8 KB flash memory
- 512 byte SRAM
- 512 byte EEPROM sebagai penyimpan data non-volatile
- 2 buah timer 8 bit, 1 buah timer 16bit
- 4 port I/O, masing-masing 8 bit
- ADC 10 bit, 8 channel
- PWM, 4 channel
- Interrupt external
- USART
- Watchdog timer
- In system programming dengan SPI

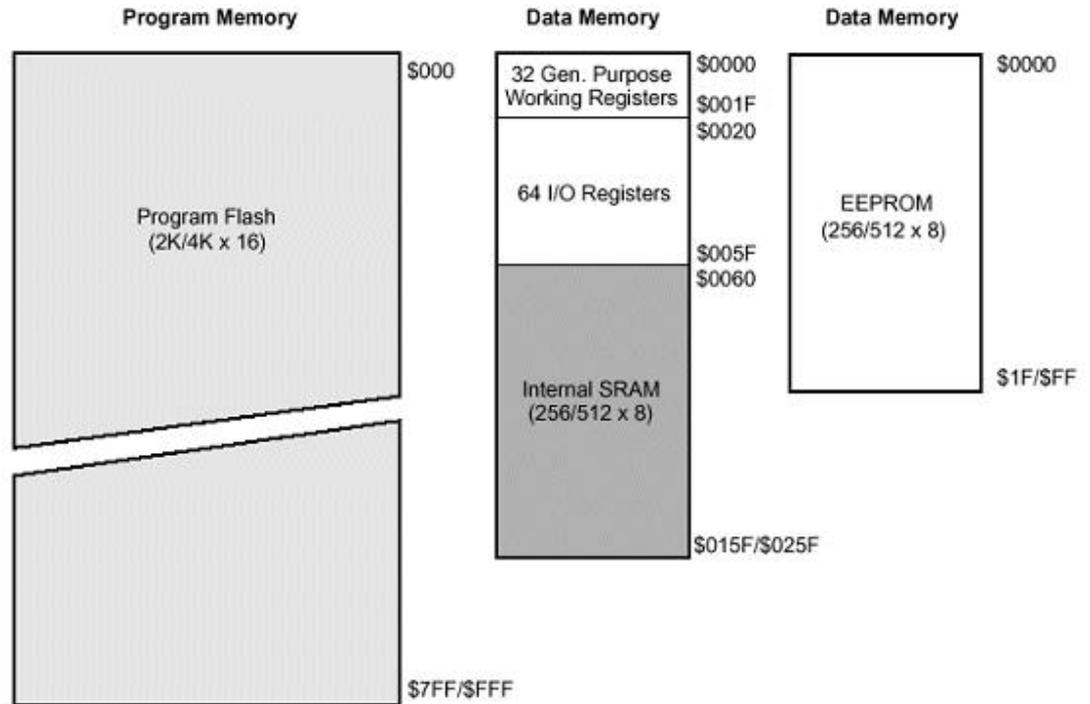


MEMORY MAP

- Mikrokontroler AVR memiliki peta memori yang terpisah antara memori data dan program.

Memory data

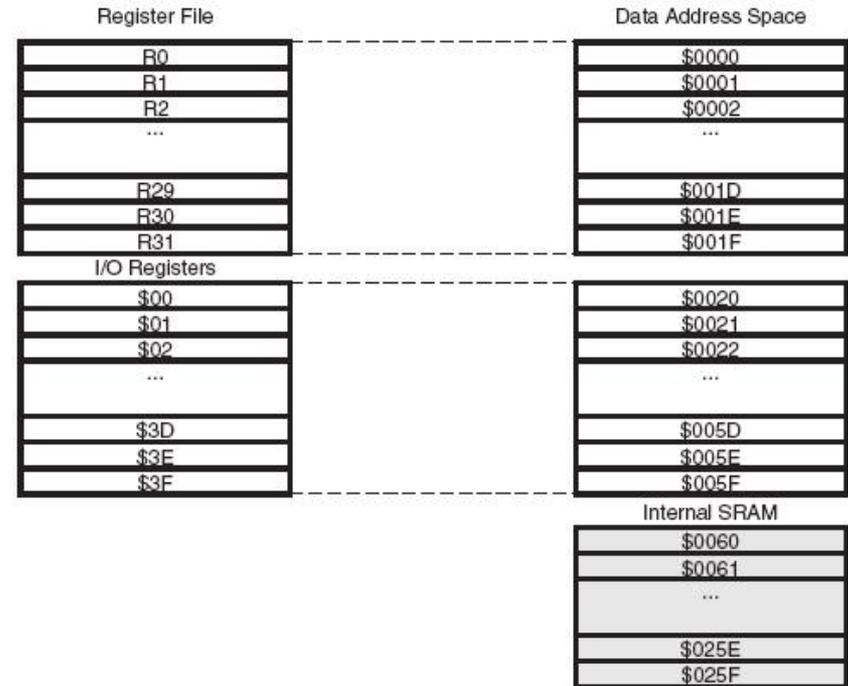
Memori program



Memory data

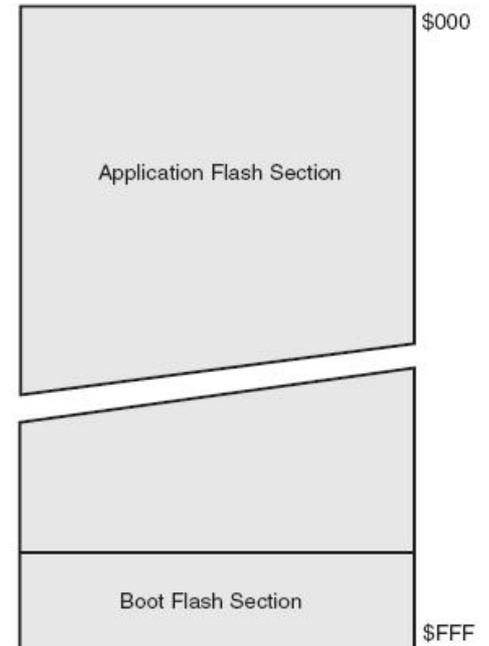
Memori data: berupa memory volatile (RAM) yang terbagi atas :

- 32 register General purpose (alamat 0x00 sampai 0x1F).
- 64 register I/O (alamat 0x20 sampai 0x5F) register yang digunakan untuk mengatur fungsi beberapa peripheral mikrokontroler.
- SRAM internal (alamat 0x60 sampai 0x25F)



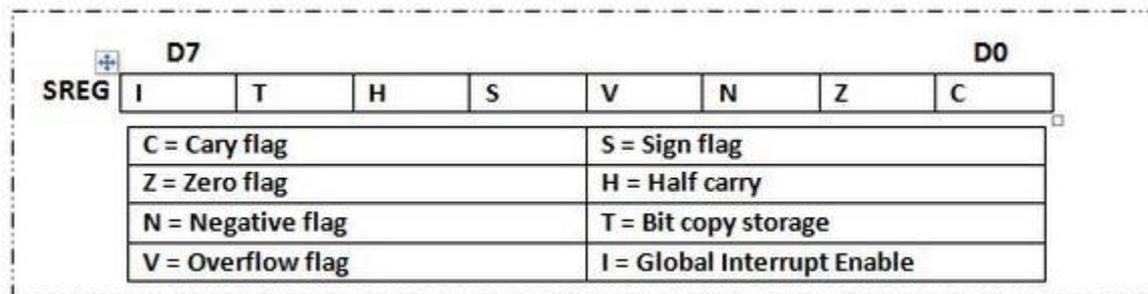
Memori program

- Berupa flash memory (non volatile) yang bisa dihapus tulis.
- Memory program tersusun atas word (2 byte) karena setiap instruksi memiliki lebar 16 bit atau 32 bit.
-
- ATmega8535 memiliki 4KWord flash dengan alamat 0x000 sampai 0xFFF. Flash tersebut dialamati oleh program counter (PC).



AVR status register

- Status register merupakan register 8-bit. disebut sebagai flag register.
- Bit C, Z, N, V, S dan H dinamakan conditional flags, berarti bahwa mereka menandakan beberapa kondisi yang terjadi setelah instruksi dijalankan. Setiap conditional flags bisa digunakan untuk melakukan percabangan kondisional (jump).



Gambar 1. Bit dari Status Register (SREG)

AVR status register

- **C, the carry flag**

Flag ini akan diset ketika ada carry yang keluar dari bit D7. Flag bit ini terpengaruh setelah ada penjumlahan atau pengurangan 8-bit.

- **Z, the zero flag**

Zero flag mencerminkan hasil dari sebuah operasi logis atau aritmetik. Jika hasilnya nol, maka $Z = 1$. Begitu pula dengan, $Z = 0$ jika hasilnya bukan nol. Flag ini biasa digunakan pada operasi perulangan (loop).

AVR status register

- **N, the negative flag**

Representasi biner dari signed numbers (bilangan bulat) menggunakan D7 sebagai bit penanda. Negative flag mencerminkan hasil dari sebuah operasi aritmetik. Jika bit D7 dari sebuah hasil ada nol, maka $N = 0$ dan hasilnya adalah positif. Jika bit D7 adalah satu, maka $N = 1$ dan hasilnya adalah negatif. Bit negatif dan V flag digunakan untuk operasi aritmetika bilangan bulat.

- **V, the overflow flag**

Flag ini diset ketika hasil dari sebuah operasi bilangan bulat terlalu besar, mengakibatkan bit yang paling tinggi melampaui bit penanda. Pada umumnya, carry flag digunakan untuk mendeteksi kesalahan pada operasi aritmetik bilangan bulat non negatif (unsigned number) sementara itu overflow flag digunakan untuk mendeteksi kesalahan pada operasi bilangan bulat (signed number).

AVR status register

- **S, sign bit**

Flag ini merupakan hasil dari logika XOR antara flag N dan V.

- **H, Half carry flag**

Jika ada carry dari D3 ke D4 selama operasi ADD atau SUB, maka bit ini akan diset; selain itu akan dikosongkan. Flag bit ini digunakan oleh instruksi yang melakukan aritmetik BCD (binary code decimal). Pada beberapa mikroprosesor ini dinamakan AC Flag (Auxiliary Carry flag).

CONTOH

Contoh 1

Perlihatkan status dari Z flag selama eksekusi dari program berikut :

```
LDI  R16, 4           ;R20 = 4
DEC  R16              ;R20 = R16 -1
```

Solusi :

Z Flag bernilai satu ketika hasil operasinya adalah nol. Selain itu, tidak diset (nol). Sehingga :

Setelah	Nilai Dari R20	Z Flag
LDI R16, 4	4	0
DEC R16	3	0
DEC R16	2	0
DEC R16	1	0
DEC R16	0	1

CONTOH

Contoh 2

Perlihatkan status dari C, H, dan Z flag setelah penjumlahan antara 0x38 dan 0x2F pada intruksi berikut :

```
LDI  R20, 0x38
LDI  R21, 0x2F
ADD  R20, R21 ;tambahkan R21 ke R20
```

Solusi :

```
      $38   0011 1000
+     $2F   0010 1111
-----
     $67   0110 0111   R20 = 0x67
```

C = 0 karena tidak terdapat carry melebihi bit D7.

H = 1 karena terdapat carry dari bit D3 ke D4.

Z = 0 karena R20 (hasilnya) mempunyai nilai selain 0 setelah dilakukannya penjumlahan.

Show how the flag register is affected by the following instructions.

```
LDI    R21,0xF5    ;R21 = F5H
LDI    R22,0x0B    ;R22 = 0x0BH
ADD    R21,R22     ;R21 = R21+R22 = F5+0B = 00 and C = 1
```

Solution:

```
    F5H           1111 0101
+  0BH           + 0000 1011
-----
   100H          0000 0000
```

After the addition, register R21 contains 00 and the flags are as follows:

C = 1 because there is a carry out from D7.

Z = 1 because the result in destination register (R21) is zero.

H = 1 because there is a carry from D3 to D4.

Show the status of the C, H, and Z flags after the addition of 0x9C and 0x64 in the following instructions:

```
LDI R20, 0x9C
LDI R21, 0x64
ADD R20, R21 ;add R21 to R20
```

Solution:

\$9C	1001 1100	
+ \$64	<u>0110 0100</u>	
\$100	0000 0000	R20 = 00

C = 1 because there is a carry beyond the D7 bit.

H = 1 because there is a carry from the D3 to the D4 bit.

Z = 1 because the R20 (the result) has value 0 in it after the addition.

Show the status of the C, H, and Z flags after the addition of 0x88 and 0x93 in the following instructions:

```
LDI R20, 0x88
LDI R21, 0x93
ADD R20, R21 ;add R21 to R20
```

Solution:

\$ 88	1000 1000	
+ \$ 93	<u>1001 0011</u>	
\$11B	0001 1011	R20 = 0x1B

C = 1 because there is a carry beyond the D7 bit.

H = 0 because there is no carry from the D3 to the D4 bit.

Z = 0 because the R20 has a value other than 0 after the addition.

Example 5-16

Examine the following code, noting the role of the V and N flags:

```
LDI    R20,7           ;R20 = 0000 0111
LDI    R20,18          ;R20 = 0001 0010
ADD    R20,R21          ;R20 = (+7) + (+18)
                               ;R20 = 00000111 + 00010010 = 0001 1001
                               ;R20 = (+7) + (+18) = +25, N = 0, positive
                               ;and correct, V = 0
```

Solution:

```
  + 7 0000 0111
+ +18 0001 0010
-----
+25 0001 1001  N = 0 (positive 25) and V = 0
```

According to the CPU, this is +25, which is correct, and V = 0 indicates that.

Example 5-13

Examine the following code and analyze the result, including the N and V flags.

```
LDI    R20,0X60    ;R20 = 0110 0000 (+70)
LDI    R21,0x46    ;R21 = 0100 0110 (+96)
ADD    R20,R21     ;R20 = (+96) + (+70) = 1010 0110
                          ;R20 = A6H = -90 decimal, INVALID!!
```

Solution:

```
   +96    0110 0000
+  +70    0100 0110
+ 166    1010 0110  N = 1 (negative) and V = 1 Sum = -90
```

According to the CPU, the result is negative ($N = 1$), which is wrong. The CPU sets $V = 1$ to indicate the overflow error. Remember that the N flag is the D7 bit. If $N = 0$, the sum is positive, but if $N = 1$, the sum is negative.

Example 5-14

Examine the following code, noting the role of the V and N flags:

```
LDI    R20,0x80    ;R20 = 1000 0000 (80H = -128)
LDI    R21,0xFE    ;R21 = 1111 1110 (FEH = -2)
ADD    R20,R21     ;R20 = (-128) + (-2)
                          ;R20 = 1000000 + 11111110 = 0111 1110,
                          ;N = 0, R0 = 7EH = +126, invalid
```

Solution:

-128	1000 0000	
<u>+ - 2</u>	<u>1111 1110</u>	
- 130	0111 1110	N = 0 (positive) and V = 1

According to the CPU, the result is +126, which is wrong, and $V = 1$ indicates that. Notice that the N flag indicates the sign of the corrupted result, not the sign that the real result should have.

Examine the following code, noting the role of the V and N flags:

```
LDI    R20,-2           ;R20 = 1111 1110 (R20 = FEH)
LDI    R21,-5           ;R21 = 1111 1110 (R21 = FBH)
ADD    R20,R21          ;R20 = (-2) + (-5) = -7 or F9H
                          ;correct, since V = 0
```

Solution:

```
  -2    1111 1110
+ -5    1111 1011
-----
  -7    1111 1001    and V = 0 and N = 1. Sum is negative
```

According to the CPU, the result is -7, which is correct, and the V indicates that (V = 0).

Examine the following code, noting the role of the V and N flags:

```
LDI    R20,7           ;R20 = 0000 0111
LDI    R20,18          ;R20 = 0001 0010
ADD    R20,R21          ;R20 = (+7) + (+18)
                          ;R20 = 00000111 + 00010010 = 0001 1001
                          ;R20 = (+7) + (+18) = +25, N = 0, positive
                          ;and correct, V = 0
```

Solution:

```
    + 7 0000 0111
+ +18 0001 0010
+25 0001 1001   N = 0 (positive 25) and V = 0
```

According to the CPU, this is +25, which is correct, and V = 0 indicates that.

SISTEM MINIMUM

