INDOBOT ACADEMY
THE FASTEST WAY TO IOT MASTERY

MINGGU KE-2 : BAB 7

Pengenalan Microcontroller dan Microprocessor

Kelas Memulai Jadi IoT Engineer Hebat

A. Microcontroller

1. Pengenalan *Microcontroller*

Microcontroller adalah sebuah komputer kecil yang dikemas dalam bentuk chip IC (*Integrated Circuit*) dan dirancang untuk melakukan tugas atau operasi tertentu. Pada dasarnya, sebuah IC *Microcontroller* terdiri dari satu atau lebih Inti Prosesor (CPU), Memori (RAM dan ROM) serta perangkat *INPUT* dan *OUTPUT* yang dapat di program.

Ciri Khas *Microcontroller* :

- **Kemampuan CPU Yang Tidak Terlalu Tinggi**

  Berbeda dengan CPU, umumnya *Microcontroller* sederhana hanya dapat melakukan atau memproses beberapa perintah saja, meskipun saat ini telah banyak dibuat *Microcontroller* dengan spesifikasi yang lebih canggih tapi tentunya belum dapat menyamai kemampuan CPU dalam memproses data dari perangkat lunak.

- ***Microcontroller* Memiliki Memori Internal Yang Kecil**

  Tentu bagi anda yang sering melihat *Microcontroller*, maka dapat melihat jumlah memori internal dari *Microcontroller* itu terbilang kecil. Umumnya sebuah *Microcontroller* hanya berisikan ukuran *Bit*, *Byte*, atau *Kilobyte*.

- ***Microcontroller* dibekali *Non-Volatile Memory***

  Dengan adanya memori *non-volatile* pada *Microcontroller*, maka perintah yang telah dibuat dapat dihapus ataupun dibuat ulang, selain itu dengan penggunaan memori *non-volatile*, maka memungkinkan data yang telah disimpan dalam *Microcontroller* tidak akan hilang meskipun tidak disuplai oleh *power supply* (Catu daya).

- **Perintah Relatif Sederhana**

  Dengan kemampuan CPU yang tidak terlalu tinggi, maka berimbas pada kemampuan dalam melakukan pemrosesan data yang tidak tinggi pula. Meskipun begitu, *Microcontroller* terus dikembangkan menjadi canggih, contohnya *Microcontroller* yang dipakai untuk melakukan pengolahan sinyal dan sebagainya.

- **Program atau Perintah Berhubungan Langsung Dengan *Port I/O***

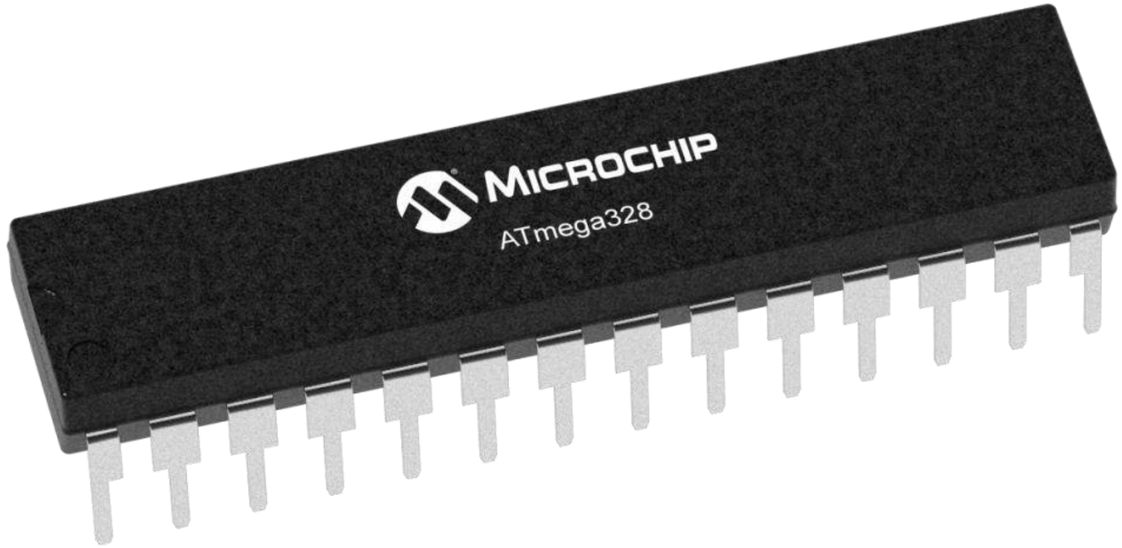
  Salah satu komponen utama *Microcontroller* adalah *Port I/O*, *Port input* maupun *output (I/O)* memiliki fungsi utama sebagai jalan komunikasi. Sederhananya *Port I/O* membangun komunikasi antara piranti masukan dan piranti keluaran. Fungsi *Microcontroller* :

  1. *Microcontroller* Sebagai *Timer* / Pewaktu.
  2. *Microcontroller* Sebagai Pembangkit Osilasi.
  3. *Microcontroller* Sebagai *Flip - Flop*.
  4. *Microcontroller* Sebagai ADC (*Analog Digital Converter*).
  5. *Microcontroller* Sebagai *Counter*.
  6. *Microcontroller* Sebagai *Decoder* dan *Encoder*.

2. Arsitektur *Microcontroller* ATMega 328

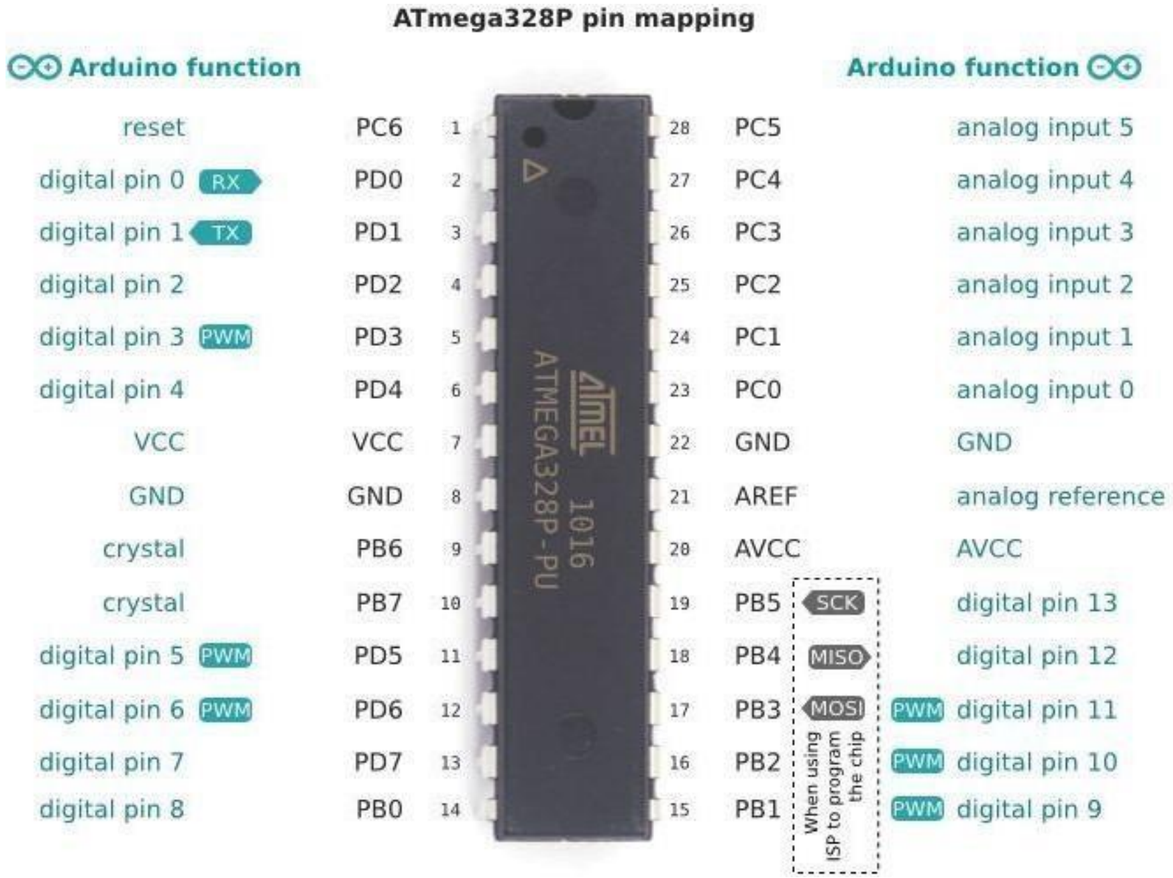
Microcontroller ATmega328 memiliki arsitektur *Harvard*, yaitu memisahkan memori untuk kode program dan memori untuk data sehingga dapat memaksimalkan kerja dan *parallelism*. Instruksi-instruksi dalam memori program dieksekusi dalam satu alur tunggal, dimana pada saat satu instruksi dikerjakan lalu instruksi berikutnya sudah diambil dari memori program.

Konsep inilah yang memungkinkan instruksi-instruksi dapat dieksekusi dalam setiap satu siklus *clock*. 32 x 8-*bit register* serbaguna yang digunakan untuk mendukung operasi pada ALU (*Arithmatic Logic Unit*) dapat dilakukan dalam satu siklus. 6 dari *register* serbaguna ini dapat digunakan sebagai 3 buah *register pointer* 16-*bit* pada mode pengalamatan tidak langsung untuk mengambil data pada ruang memori data. Ketiga *register pointer* 16-*bit* ini disebut dengan *register X* (gabungan R26 dan R27), *register Y* (gabungan R28 dan R29), dan *register Z* (gabungan R30 dan R31).

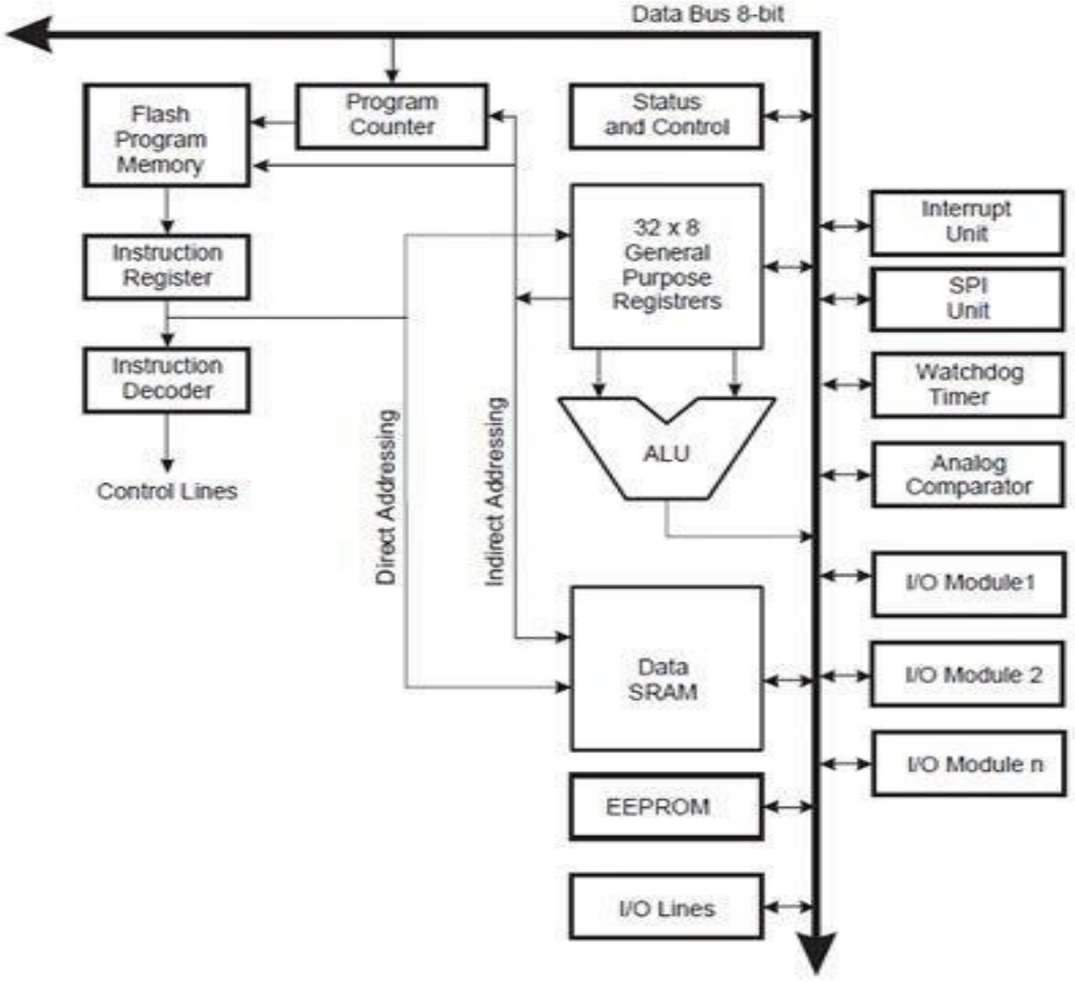


INDOBOT ACADEMY
THE FASTEST WAY TO IOT MASTERY

3. Fungsi tiap pin ATMega328



4. Arsitektur ATMega328



INDOBOT ACADEMY
THE FASTEST WAY TO IOT MASTERY

5. Fungsi Pin Port B

Port	Alternate Functions
PB7	XTAL2 (Chip Clock Oscilator pin 2) TOSC2 (Timer Oscilator pin 2) PCINT7 (Pin Chane Interrupt 7)
PB6	XTAL1 (Chip Clock Oscilator pin 1 or External Clock Input) TOSC1 (Timer Oscilator pin 1) PCINT7 (Pin Chane Interrupt 6)
PB5	SCK (SPI Bus Master Clock Input) PCINT5 (Pin Change Interrupt 5)
PB4	MISO (SPI Bus Master Input/Slave Output) PCINT4 (Pin Change Interrupt 4)
PB3	MOSI (SPI Bus Master Input/Slave Output) OC2A (Timer/Counter2 Output Compare Match A Output) PCINT3 (Pin Change Interrupt 3)
PB2	SS (SPI Bus Master Slave select) OC1B (Timer/Counter1 Output Compare Match B Output) PCINT2 (Pin Change Interrupt 2)
PB1	OC1A (Timer/Counter2 Output Compare Match A Output) PCINT1 (Pin Change Interrupt 1)
PB0	ICP1 (Timer/Counter1 Input Capture Input) CLKO(Divided System Clock Output) PCINT1 (Pin Change Interrupt 0)

6. Fungsi Pin Port C

Fungsi Pin Port C	
Port	Alternate Functions
PC6	RESET (Reset Point) PCINT14 (Pin Change Interrupt 14)
PC5	ADC5 (ADC Input Channel 5) SCL (2-Wire Serial Bus Clock Line) PCINT13(Pin Change Interrupt13)
PC4	ADC4 (ADC Input Channel 4) SCL (2-Wire Serial Bus Data Input/Output Line) PCINT13(Pin Change Interrupt12)
PC3	ADC3 (ADC Input Channel 3) PCINT11 (Pin Change Interrupt11)
PC2	ADC2 (ADC Input Channel 2) PCINT10 (Pin Change Interrupt10)
PC1	ADC1 (ADC Input Channel 1) PCINT10 (Pin Change Interrupt9)
PC0	ADC2 (ADC Input Channel 0) PCINT10 (Pin Change Interrupt8)

7. Fungsi Pin Port D

Port	Alternate Functions
PD7	AIN1 (Analog Comparator Negative Input) PCINT23 (Pin Change Interrupt23)
PD6	AIN0 (Analog Comparator Negative Input) OC0A (Timer/Counter0 Output Compare Match A Output) PCINT22 (Pin Change Interrupt22)
PD5	T1 (Timer/Counter 1 External Counter Input) OC0B (Timer/Counter0 Output Compare Match B Output) PCINT21 (Pin Change Interrupt21)
PD4	XCK (USART External Clock Input/Output) T0(Timer/Counter 0 External Counter Input)
PD3	INT1 (External Interrupt 1 Input) OC2B(Timer/Counter2 Output Compare Match B Output) PCINT19 (Pin Change Interrupt 19)
PD2	INT0 (External Interrupt 0 Input) PCINT18 (Pin Change Interrupt 18)
PD1	TXD (USART Output Pin) PCINT17 (Pin Change Interrupt 17)
PD0	RXD (USART Input Pin) PCINT16 (Pin Change Interrupt 16)

8. Sejarah Bahasa Pemrograman C

Bahasa pemrograman C dibuat pertama kali oleh seorang yang bernama Dennis M. Ritchie pada tahun 1972. Saat itu Ritchie bekerja di *Bell Labs*, sebuah pusat penelitian yang berlokasi di Murray Hill, New Jersey, Amerika Serikat.

Ritchie membuat bahasa pemrograman C untuk mengembangkan sistem operasi UNIX. Dahulunya, sistem operasi UNIX dibuat dengan menggunakan bahasa rakitan (*assembly language*). Namun bahasa tersebut sendiri sangat rumit dan sulit untuk dikembangkan. Dengan niat untuk mengganti bahasa *assembly*, peneliti di *Bell Labs* membuat bahasa pemrograman baru yaitu B. Namun bahasa pemrograman B sama-sama memiliki beberapa kekurangan, yang pada akhirnya dapat dilengkapi oleh bahasa pemrograman C.

Dengan bahasa C itulah sistem operasi UNIX ditulis ulang. Pada akhirnya, UNIX menjadi dasar dari banyak sistem operasi modern sampai saat ini, termasuk Linux, Mac OS (iOS), sampai sistem operasi Android.

9. Keunggulan Bahasa Pemrograman C

- **C sebagai bahasa pemrograman prosedural**

  Pada dasarnya konsep pemrograman prosedural yaitu sebuah metode pemrograman yang setiap baris perintahnya diproses secara berurutan dari baris paling atas hingga ke baris paling bawah. Selain itu juga terdapat fungsi tambahan (*function*) yang digunakan untuk menyelesaikan berbagai jenis tugas. Bahasa pemrograman C termasuk ke dalam kelompok ini. Selain konsep prosedural, terdapat juga konsep pemrograman objek (*Object-Oriented Programming*). Dalam bahasa pemrograman objek, dimana setiap tugas akan dijalankan dengan menggunakan *class* dan *object*. Contoh bahasa pemrograman objek yaitu JAVA.

- **Bahasa C sangat cepat dan efisien**

  Aplikasi yang dibuat menggunakan bahasa C ini maka bisa dieksekusi dengan sangat cepat dan juga berukuran kecil. Karena C dapat langsung berkomunikasi dengan *hardware*, hal tersebut memiliki fitur yang jarang tersedia di bahasa pemrograman modern seperti JAVA, PHP, dan juga Python. Namun hal ini juga memiliki daya kelemahan. Bahasa C relatif sederhana dan tidak memiliki fitur-fitur modern seperti *garbage collection* dan juga *dynamic typing*.

- **C adalah *portable language***

  Artinya, bahasa pemrograman C bisa di-*compile* berulang supaya berjalan di berbagai sistem operasi tanpa perlu mengubah kode-kode yang ada. Aplikasi yang dibuat di *Windows* dengan bahasa C bisa dipindahkan ke Linux dengan sedikit atau tanpa adanya modifikasi.

- **C merupakan “inti" dari bahasa pemrograman modern**

Bahasa pemrograman C sudah banyak menginspirasi terciptanya bahasa pemrograman lain, seperti contoh C++, C#, Objective C, PHP, JAVA, JavaScript dan masih banyak lagi lainnya. Dengan mempelajari bahasa C, maka anda akan merasa familiar dan lebih mudah saat pindah ke bahasa pemrograman lain yang merupakan turunan dari bahasa C itu sendiri.

Contoh program Arduino dengan bahasa C :

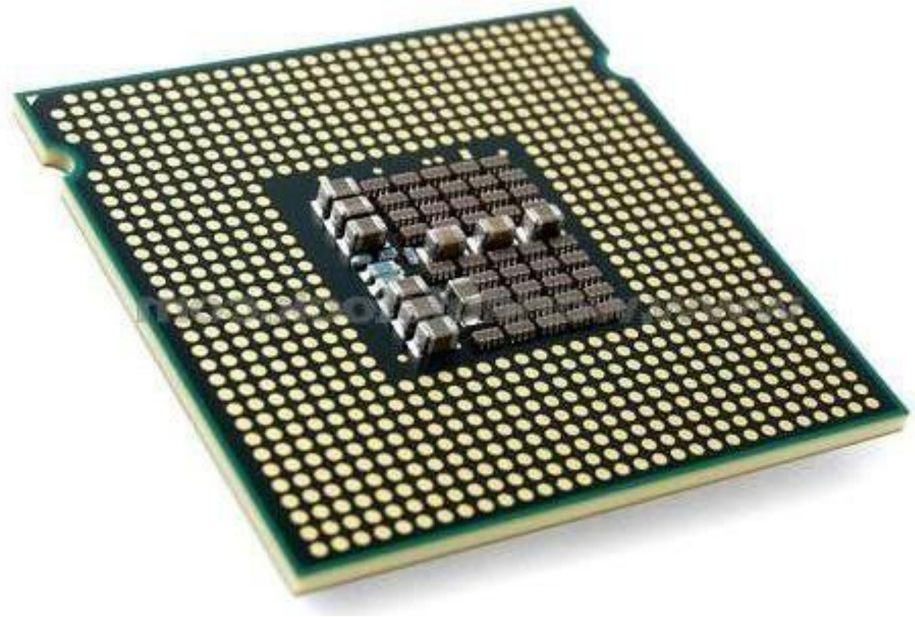
```
void setup() {
    // initialize digital pin LED_BUILTIN as an output
    pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
    digitalWrite(LED_BUILTIN, HIGH); // turn the LED on
    delay(1000); // wait for a second
    digitalWrite(LED_BUILTIN, LOW); // turn the LED off
    delay(1000); // wait for a second
}
```

B. Microprocessor

1. Pengenalan *Microprocessor*

Microprocessor adalah sebuah *Chip IC* (Sirkuit Terintegrasi) yang menggabungkan fungsi inti dari unit pemrosesan pusat (CPU/*Central Processing Unit*) komputer. *Chip IC* (*Microprocessor*) ini merupakan perangkat multiguna yang dapat di program untuk menerima data digital sebagai *input*, memprosesnya sesuai dengan instruksi yang tersimpan dalam memorinya, dan memberikan hasil sebagai *output*. Fungsi utamanya yaitu sebagai unit kontrol yang dapat mengendalikan seluruh kerja sistem *microprocessor*.



Adapun fungsi lainnya antara lain sebagai berikut :

- Mengambil instruksi dan data dari memori.
- Memindah data dari dan ke memori.
- Mengirim sinyal kendali dan melayani sinyal interupsi.
- Menyediakan pewaktuan untuk siklus kerja sistem *microprocessor*.
- Mengerjakan operasi logika dan aritmatika.

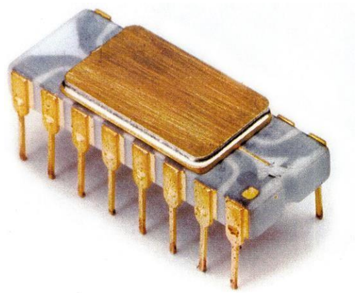
2. Sejarah *Microprocessor*

Sekitar tahun 1642, Blaise Pascal menciptakan mesin hitung yang menggunakan prinsip gigi roda (cikal bakal kalkulator sekarang). Perkembangan berikutnya adalah penciptaan mesin hitung raksasa (1940-1950) yang dibangun dari relai-relai dan tabung-hampa (*vacum-tube*) berukuran raksasa. Perkembangan selanjutnya adalah pemanfaatan transistor dan komponen zat padat (*solid-state electronic*) untuk membangun mesin serupa yang berukuran lebih kecil. Akhirnya, perkembangan rangkaian terpadu atau terintegrasi (*Integrated Circuit*) sekitar 1960 itu telah mengantar perkembangan *microprocessor* dan sistem komputer berbasis *microprocessor* ke arah yang lebih baik.

Microprocessor atau *processor* merupakan bagian yang sangat penting dari sebuah komputer, yang berfungsi sebagai otak dari komputer. Tanpa *processor*, komputer hanyalah sebuah mesin dungu yang tak bisa apa-apa. Nah berikut perkembangan *processor*, mulai dari generasi 4004 *microprocessor* yang dipakai pada mesin penghitung *Busicom* sampai dengan *intel Quad-core Xeon*.

Perkembangan *processor* diawali oleh *processor intel*, yang pada saat itu hanya satu-satunya *microprocessor* yang ada. Tetapi saat ini sudah banyak beredar *processor* dari *produsen* yang lain, sehingga *processor* yang ada cukup beragam. Konsumen atau pengguna dapat memilih sesuai dengan kebutuhan.

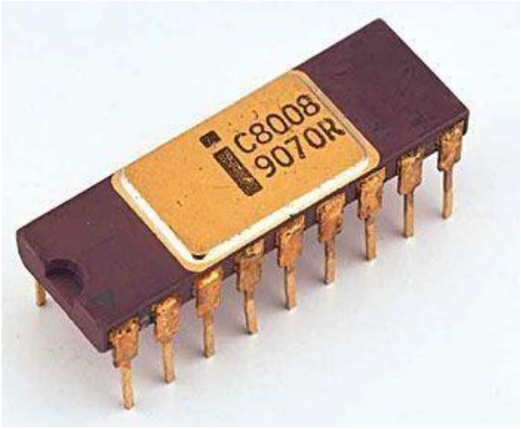
A. *Microprocessor* 4004 (1971)



Pada awal tahun 1971, Intel Corporation & Marcion G. Hoff memperkenalkan *microprocessor* pertama kali yaitu *microprocessor* 4-*bit* seri 4004, yang memiliki 4096 alamat, masing-masing 4-*bit* memori dan memiliki 45 buah instruksi yang berbeda serta hanya digunakan untuk keperluan terbatas, misal *video game* dan beberapa alat kendali (*controller*) sederhana. *Chip intel 4004* ini mengawali perkembangan CPU dengan mempelopori peletakan seluruh komponen mesin hitung dalam satu IC.

B. *Microprocessor* 8008 (1971)

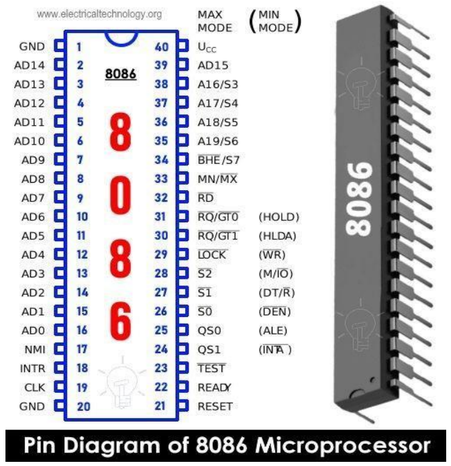
Pada akhir tahun 1971, dengan menyadari bahwa *microprocessor* adalah produk komersial, *Intel Corporation* kembali meluncurkan *microprocessor* 8-*bit* pertama yaitu 8008, dengan memori yang lebih besar (16 KB x 8-*bit*) dan jumlah instruksi yang lebih banyak (48 instruksi). Oleh karena pemakaian *microprocessor* 8008 yang makin meningkat dan meluas maka memori dan jumlah instruksi tersebut menjadi kurang memadai, sehingga *Intel Corp* tahun 1973 meluncurkan *microprocessor* 8080 (*microprocessor* 8-*bit* modern pertama). Setelah itu banyak perusahaan-perusahaan lain yang juga mengeluarkan *microprocessor* 4-*bit* & 8-*bit* mereka yang pertama. Keistimewaan *microprocessor* ini, selain alamat memori dan jumlah instruksinya yang lebih besar, peningkatan juga terasa pada aspek kecepatan (*speed*) yang fantastis dibanding dengan *microprocessor* 8008. *Microprocessor* 8008 dalam melakukan satu operasi internal membutuhkan waktu eksekusi sebesar 20mS, sedangkan *microprocessor* 8080 hanya membutuhkan waktu eksekusi sebesar 2,0mS saja.



C. *Microprocessor* 8080 (1974)

Pada tahun 1974, *intel* kembali mengeluarkan *microprocessor* terbaru dengan seri 8080. Pada seri ini, *intel* melakukan perubahan dari *microprocessor multivoltage* menjadi *triple voltage*, teknologi yang dipakai NMOS, lebih cepat dari seri sebelumnya yang memakai teknologi PMOS. *Microprocessor* ini adalah otak pertama bagi komputer yang bernama *ALTAIR*. Pada saat ini pengalamatan memori sudah sampai 64 *kilobyte*. Kecepatannya sampai 10X *microprocessor* sebelumnya. Tahun ini juga muncul *microprocessor* dari produsen lain seperti MC6800 dari Motorola-1974, Z80 dari Zilog-1976 (merupakan dua rival berat), dan *processor-processor* lain seperti seri 6500 buatan MOST, Rockwell, Hyundai, WDC, NCR dst.

D. *Microprocessor* 8086 (1978)



Pin Diagram of 8086 Microprocessor

Pada tahun 1978, *Intel Corp* meluncurkan *microprocessor* 8086. *Processor* 8086 adalah CPU pertama yang memiliki cakupan 16-*bit*, tetapi pada saat ini masih banyak digunakan di *mainboard* standar 8-*bit*, karena *mainboard* 16-*bit* merupakan suatu hal yang mahal. Akhirnya pada tahun 1979, *intel* merancang ulang *processor* ini sehingga *compatible* dengan *mainboard* 8-*bit* yang diberi nama 8088 tetapi secara logika bisa dinamakan 8086sx. Perusahaan komputer *IBM* menggunakan *processor* 8086sx ini untuk komputernya karena lebih murah dari harga 8086, dan juga bisa menggunakan *mainboard* bekas dari *processor* 8080. Teknologi yang digunakan pada *processor* ini juga berbeda dari seri 8080, dimana pada seri 8086 dan 8086sx *intel* menggunakan teknologi HMOS.

E. *Microprocessor* 286 (1982)

Intel 286 atau yang lebih dikenal dengan nama 80286 adalah sebuah *processor* yang pertama kali dapat mengenali dan menggunakan *software* yang digunakan untuk *processor* sebelumnya. 286 (1982) juga merupakan *processor* 16-*bit*. *Processor* ini mempunyai kemajuan yang relatif besar dibanding *chip-chip* generasi pertama. Frekuensi *clock* ditingkatkan, tetapi perbaikan yang utama adalah optimasi penanganan perintah. *Intel* 286 menghasilkan kerja lebih banyak tiap detik *clock* daripada 8088/8086.

F. *Microprocessor* (1992)

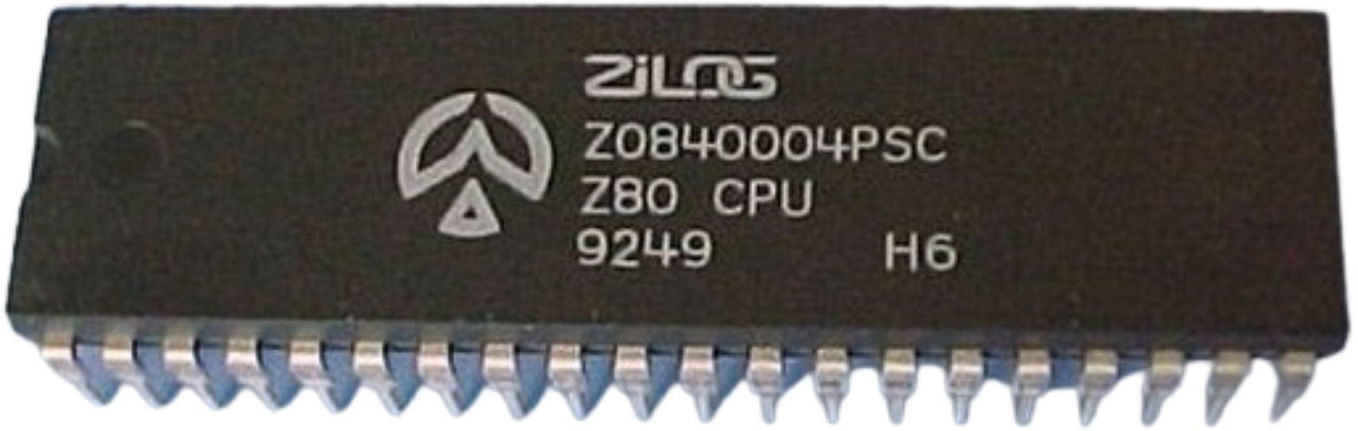
Processor 64-*bit* telah ada diantara kita sejak 1992, dan pada abad ke-21 mereka semakin populer. *Intel* dan *AMD* telah memperkenalkan *chip* 64-*bit*, dan *Mac G5* merupakan *processor* 64-*bit*. *Processor* 64-*bit* mempunyai ALU 64-*bit*, *register* 64-*bit*, *bus* 64-*bit*, dan seterusnya. Yang menjadi alasan mengapa perlu *processor* 64-*bit* adalah karena ruang pengalamatan mereka yang besar. *Microprocessor* 32-*bit* mempunyai akses RAM maksimum 2GB atau 4GB. Kedengarannya mungkin banyak, apalagi kebanyakan komputer rumahan biasanya hanya menggunakan RAM 256 MB sampai 512 MB. Namun, *limit* 4GB bisa menjadi masalah berat bagi mesin *server* dan mesin yang menjalankan *database* besar.

3. Arsitektur *Microprocessor* Z80

Setiap *microprocessor* mempunyai bentuk dan desain arsitektur yang berbeda antara satu dengan yang lainnya dan masing-masing mempunyai keunggulan dan kelemahan.

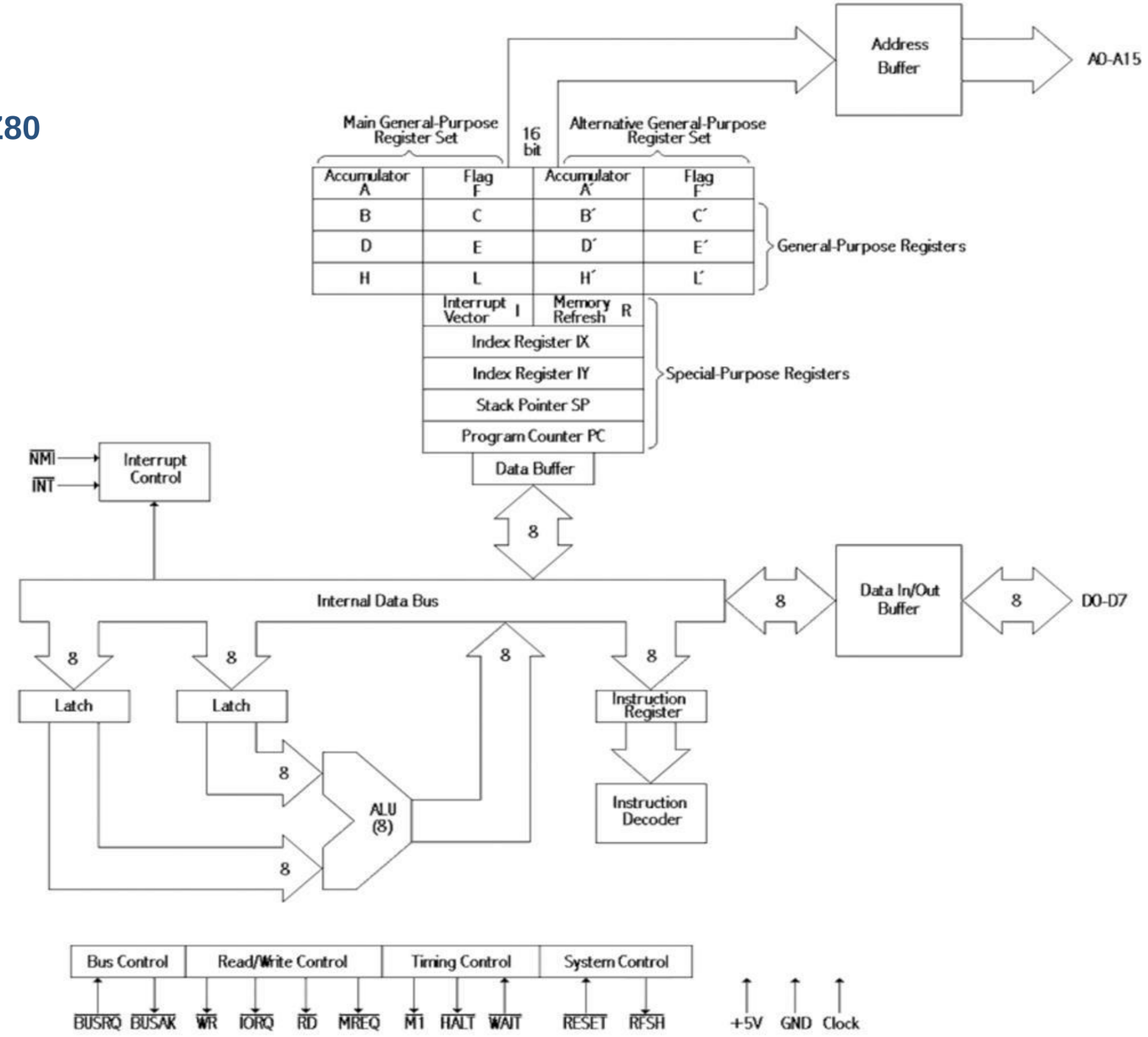
Arsitektur µP yang diperkenalkan pada bahasan ini, yaitu Z80, 8085, 6800, 68020, 80486, dan pentium. Tetapi yang akan dijelaskan secara mendalam yaitu *microprocessor* Z80.

Di dalam CPU Z80 terdapat 22 *register*, yaitu 18 *register* yang berkapasitas 8-*bit* dan 4 *register* berkapasitas 16-*bit*. *Register* ini dapat dipakai dan di program oleh pemakai. Susunan dari *register-register* ini dapat dilihat pada gambar di samping.



INDOBOT ACADEMY
THE FASTEST WAY TO IOT MASTERY

4. Blok Diagram Arsitektur Z80



- Akumulator (*register* A dan A’ = 8 *bit*) digunakan untuk menyimpan data sementara dari hasil perhitungan ALU (*Arithmetic Logic Unit*).
- *Flag register* (*register* F dan F’ = 8 *bit*) digunakan untuk menunjuk kondisi-kondisi yang terjadi sebagai hasil operasi *arithmetic* dan *logic*.
- *Register* B dan C, *register* 8 *bit* yang dapat disambungkan menjadi *register* pasangan BC dengan lebar 16 *bit*, dapat digunakan untuk menyimpan cacahan. Demikian juga dengan *register* HL dan DE berlaku untuk tujuan umum.
- *Register index* Ix dan Iy, *register* 16 *bit* yang digunakan untuk menangani lokasi memori eksternal dalam instruksi-instruksi pengalamatan tak langsung.
- *Register stack pointer* (SP), *register* 16 *bit* digunakan untuk menangani *register* 2 *byte* (16 *bit*) untuk menyimpan alamat 16 *bit* dari suatu tumpukan (*stack*) dalam memori luar yang bersifat *filo*, melalui instruksi *push* dan *pop*. *Push* digunakan untuk menyimpan data ke dalam *stack*. *Pop* digunakan untuk mengambil data dari *stack*.
- *Register program counter* (PC), *register* 16 *bit* digunakan sebagai penghitung program yang berisi instruksi berikutnya akan dilaksanakan oleh CPU.
- *Register interupsi* (I), *register* 8 *bit* digunakan untuk melayani interupsi yang berasal dari suatu alat *peripheral*, CPU akan loncat ke suatu lokasi memori yang mengandung *subroutine* yang melayani alat *peripheral* tersebut.
- *Register memory refresh*, *register* 8 *bit* digunakan untuk menyegarkan memori dinamik selama waktu CPU sedang mendekode dan melaksanakan pengambilan instruksi dari memori.

5. Bahasa *Assembly*

Bahasa mesin adalah bahasa dalam bentuk kode-kode *biner* sebagai sandi operasi (*operation code*) dari sebuah *microprocessor*. Bahasa mesin adalah bahasa yang langsung berhubungan dengan *microprocessor* yang ditulis dan dikembangkan dari *set* instruksi. Tanpa bantuan *set* instruksi, bahasa mesin sangat sulit dimengerti atau dipahami. Untuk dapat menulis bahasa mesin, maka penguasaan *set* instruksi sebuah *microprocessor* adalah wajib. Untuk membuat proses pemrograman menjadi lebih mudah, kebanyakan *programmer* menuliskan program dalam bentuk bahasa *assembly* lalu mereka menerjemahkan bahasa *assembly* yang ditulisnya menjadi bahasa mesin sehingga dapat di *download* ke memori dan di *run* (dijalankan). Penerjemahan bahasa *assembly* menjadi kode *biner* (bahasa mesin) dapat dilakukan secara manual atau menggunakan program yang disebut dengan *assembler*. Bahasa *assembly* menggunakan sejumlah *mnemonik* untuk merepresentasikan instruksi–instruksi. *Mnemonik* adalah singkatan dari suatu perintah atau instruksi sebagai piranti untuk membantu ingatan.

Sebagai contoh :

- *Load* (LD).
- *Add* (ADD).
- *Add With Carry* (ADC).
- *Subtract* (SUB).
- *Subtract With Carry* (SBC).
- *Complement* (CPL).

Pernyataan bahasa *assembly* biasanya ditulis dalam bentuk standar seperti pola tabel di bawah.

Label	Mnemonik	Operand	Komentar
Mulai	LD	A, 3F	Isi register A dengan data 3Fh
	LD	B, 5D	Isi register B dengan data 5Dh
	ADD	A, B	Jumlahkan data A dengan data B

Tabel 1. Format program assembly

Dari tabel terlihat label. Label adalah simbol atau kelompok simbol yang digunakan untuk merepresentasikan alamat yang tidak diketahui secara spesifik pada saat pernyataan-pernyataan ditulis. Label tidak dipersyaratkan dalam setiap pernyataan. Label dimasukkan bila diperlukan saja.

Mnemonik adalah kolom singkatan dari setiap perintah bahasa *assembly*. Bagian *operand* dapat memuat nama *Register*, alamat memori, alamat *port*, atau data *immediate* dari sebuah instruksi. *Operand* adalah sasaran dari instruksi. Pada bagian *operand* terbagi menjadi dua bagian yaitu sumber data yang disebut dengan *source* dan tujuan data atau destinasi. Pada umumnya, *source* ada di sebelah kanan dari destinasi. Bagian komentar biasanya digunakan untuk memberi penjelasan singkat terkait maksud atau sasaran dari instruksi di sebelah kirinya.

INDOBOT ACADEMY
THE FASTEST WAY TO IOT MASTERY

Sekian Materi

Pengenalan Microcontroller dan Microprocessor

Sampai Jumpa di Materi Berikutnya