



**PROGRAM STUDI**  
**TEKNIK INFORMATIKA**  
**FAKULTAS ILMU KOMPUTER**  
**UNIVERSITAS DIAN NUSWANTORO**

MATA KULIAH  
**ORGANISASI DAN ARSITEKTUR**  
**KOMPUTER**

# Input/ Output

- ✓ Konsep Dasar I/O,
- ✓ I/O Module,
- ✓ Programmed I/O,
- ✓ Interrupt-Driven I/O

*Tim pengampu*

*Sistem Komputer, Komunikasi dan Keamanan Data*

*T.A. 2020*

## Capaian Pembelajaran

Mahasiswa dapat memahami tentang *input/ output (i/o)*, konsep dasar *i/o*, *i/o module*, *programmed i/o*, *interrupt-driven i/o*



# Pengantar

Perangkat i/o memiliki peran penting dalam sistem komputer, dimana keberadaannya digunakan manusia dalam berkomunikasi dengan komputer. Dari *punch card* hingga *voice recognition command* digunakan manusia untuk mengoperasikan komputer dan mendapat respon dari komputer berupa tampilan monitor hingga *auto-pilot driving*.



Gambar 11.3  
Tesla auto-drive  
(nytimes.com)



Gambar 11.2  
Alexa, amazon echo  
(ahs.com)

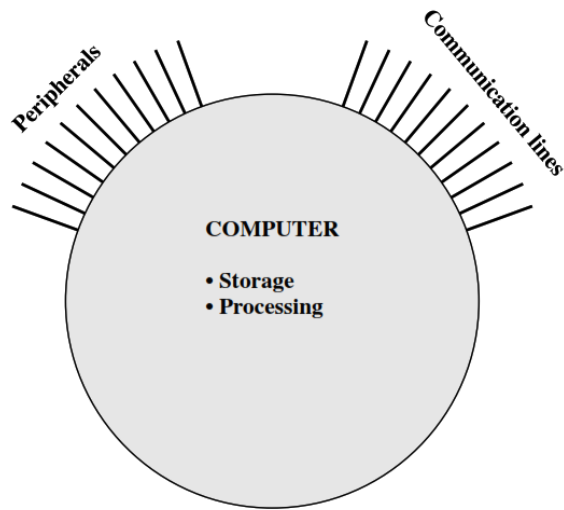


Gambar 11.1  
IBM punch card machine  
(columbia.edu)

I/O module merupakan salah satu komponen utama pada sebuah komputer (slide materi pertemuan 1), ia berfungsi sebagai kontrol dalam pengoperasian komputer. Dari memberikan masukan berupa data untuk diproses (punch card, kamera, finger print scanner, dsb.), menginterupsi jadwal proses pada processor, hingga memonitor hasil keluaran dari proses (monitor, printer, dsb.).

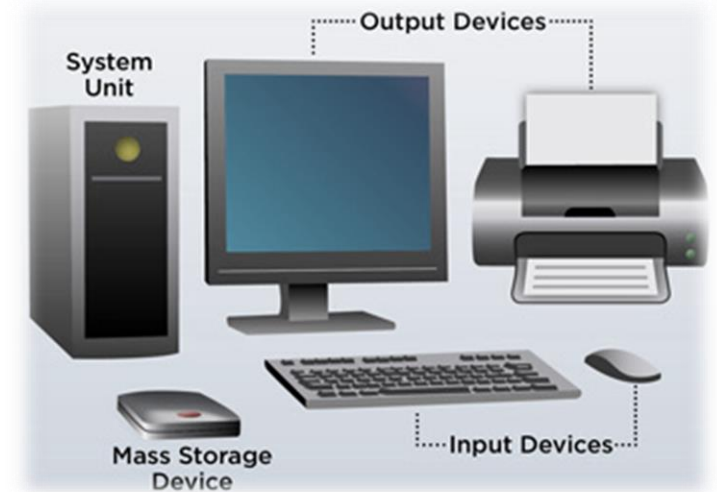
# Pengantar

Perangkat2 i/o memiliki berbagai karakteristik, salah satunya adalah *data processing rate*. Yaitu rata- rata data yang dapat diproses oleh sebuah perangkat tiap detik. Contoh : keyboard ( $\pm 10$ char/detik), scanner ( $\pm 200$ char/detik), monitor ( $\pm 30$ juta char/detik)



Gambar 11.5  
Computer peripherals and comm. Lines  
(Stallings)

Menekan sebuah *tuts* pada *keyboard* akan mengirimkan sebuah karakter (dalam bentuk ASCII) ke dalam komputer. Meski *keyboard* dapat memproses dan mengirimkan  $\pm 10$ char/ detik, biasanya ada jeda waktu diantara karakter yang dikirimkan *keyboard* ke komputer. Maka untuk mengakomodasi input dari *keyboard* sebuah mekanisme dibutuhkan untuk meng- interupsi *processor* manakala pengiriman dari *keyboard* telah siap. Mekanisme ini disebut komunikasi *interrupt-driven*, yaitu meng-interupsi daftar proses yang telah terjadwal pada *processor*.

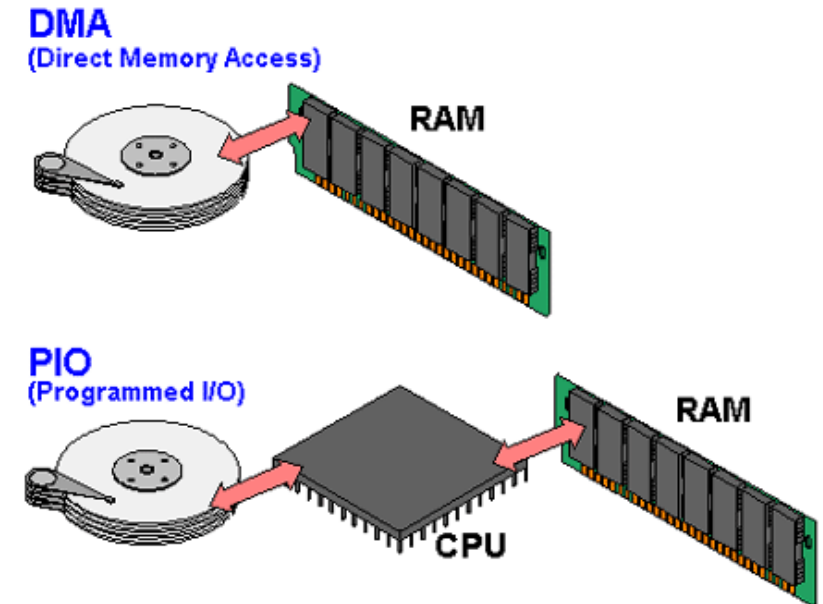


Gambar 11.4 - Computer I/O (webopedia.com)

## Pengantar (lanjt.)

*Hard-disk* yang mampu mengirimkan beberapa juta char/ detik, jika *processor* harus menunggu pengiriman data byte-per-byte maka akan membuang banyak waktu penjadwalan maka pengiriman data dikirimkan dalam bentuk blok- blok ke dalam *memory*. Sehingga dibutuhkan sebuah mekanisme yang memungkinkan *hard-disk* untuk mengirimkan data dalam volume besar ke *memory* tanpa intervensi CPU, sehingga CPU tetap dapat menjalankan jadwalnya untuk melakukan fungsi- fungsi lain. Mekanisme ini disebut DMA (*direct memory access*).

Pembahasan tentang DMA akan disampaikan pada pertemuan berikutnya.



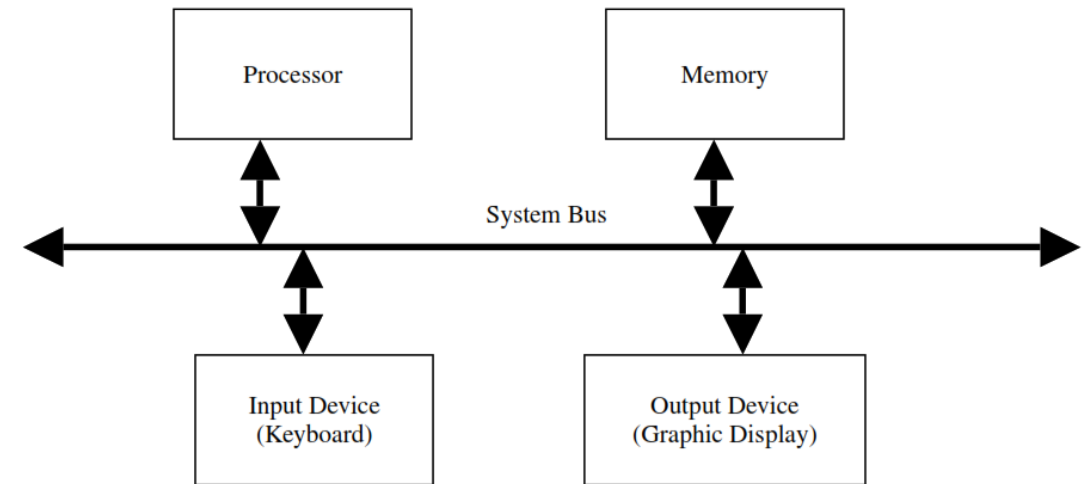
Gambar 11.6 - HDD with and without DMA  
([encyclopedia2.thefreedictionary.com](http://encyclopedia2.thefreedictionary.com))



## Konsep Dasar I/O

Gambar 11.1 menampilkan susunan komponen yang menghubungkan *processor*, *memory* dan perangkat input (keyboard) dan output (graphic display) melalui *single bus* (jalur koneksi tunggal).

Input (*keyboard*) berupa penekanan *key* akan dikirimkan ke cpu menggunakan *input register* dan hasil eksekusi dikirimkan menggunakan *output register* untuk dikirimkan kepada perangkat *output*.



Gambar 11.7 - *single bus system*

## Konsep Dasar I/O (lanjt.)

---

Telah disebutkan bahwa ada perbedaan besar pada '*data processing rate*' antara *processor* dan perangkat i/o.

Sebuah cara sederhana untuk mengakomodasi perbedaan kecepatan ini adalah dengan men-deposit-kan *char* ke sebuah *register* (*input register*), sehingga *char* tersebut 'tersedia' hingga *processor* membutuhkannya. Ketika *char* tersebut diakses oleh *processor* maka hal ini mengindikasikan bahwa perangkat *input* dapat mengirimkan *char* berikutnya.

Demikian juga ketika *processor* akan menampilkan sebuah *char* pada layar display (monitor), *processor* akan men-deposit-kan *char* pada *register* (*output register*) untuk kemudian dikirimkan kepada perangkat output. Dengan demikian *processor* dapat melanjutkan untuk mengirimkan *char* berikutnya.

## Konsep Dasar I/O (lanjt.)

---

Komunikasi sederhana antar *processor* dan perangkat i/o ini disebut *i/o protocol*, yang menuntut ketersediaan *input register* dan *output register*.

Secara umum, pada sistem komputer biasanya perangkat- perangkat i/o tertentu memiliki sejumlah *register i/o* untuk berkomunikasi dengan *processor*. Bahkan, sebuah mekanisme dibutuhkan agar processor dapat menangani register i/o dengan baik.

Setidaknya ada 2 teknik yang digunakan dalam komunikasi antara perangkat i/o dan processor.

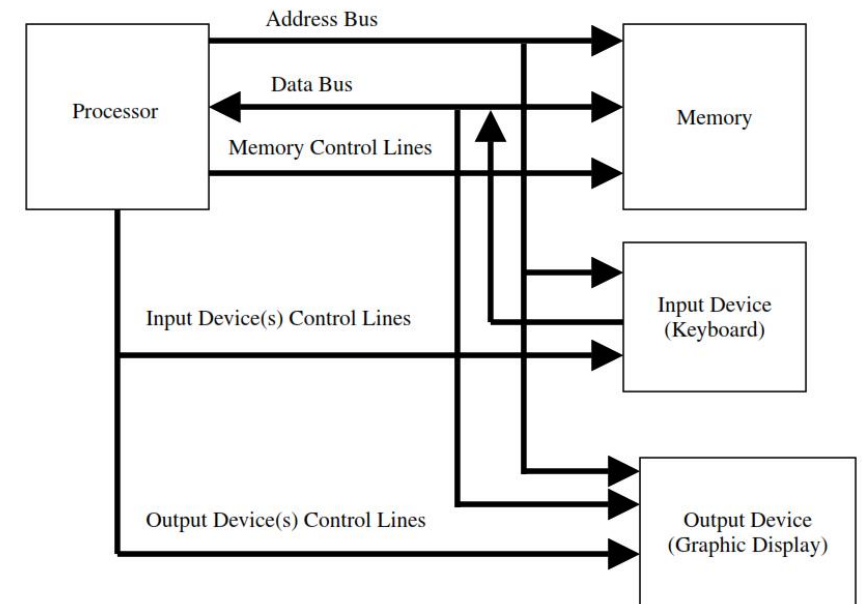


## Konsep Dasar I/O - *Shared I/O*

Pada *shared i/o*, perangkat i/o diberikan alamat tertentu, terpisah dari alokasi alamat *memory*. Eksekusi sebuah *input instruction* pada sebuah alamat perangkat input akan menyebabkan char disimpan ke dalam *register input* milik perangkat tersebut untuk kemudian dikirimkan kepada *register* tertentu didalam CPU.

Demikian juga bila terjadi eksekusi sebuah *output instruction* pada alamat perangkat output akan menyebabkan char yang tersimpan pada *register* didalam CPU dikirimkan kepada *output register*.

Keunggulan teknik ini terletak pada kepemilikan kapasitas alamat yang berbeda untuk memory dan perangkat i/o. Namun pada sisi lain, teknik ini memiliki kekurangan karena membutuhkan instuksi khusus untuk *input instruction* dan *output instruction* pada *processor instruction set* -nya.



Gambar 11.8 - *Shared I/O arrangement*

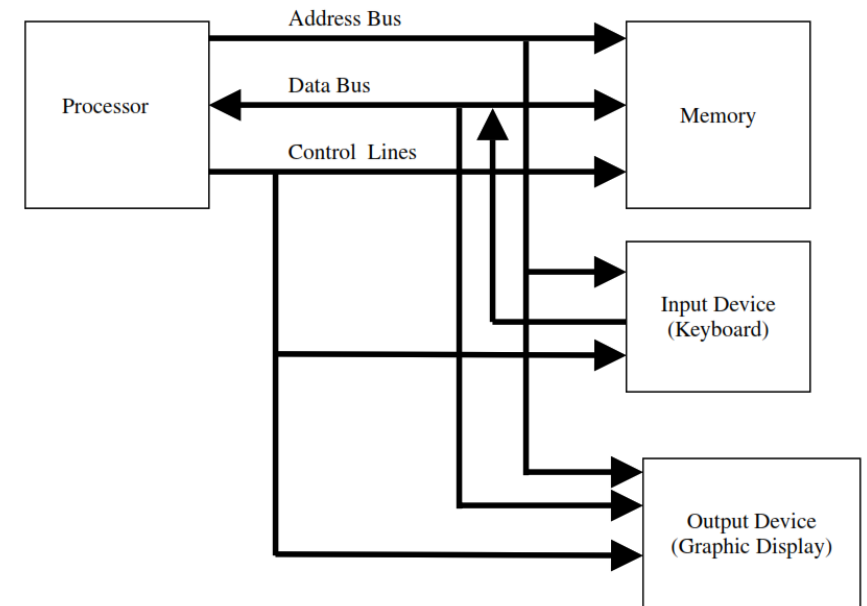
## Konsep Dasar I/O - *Memory-mapped I/O*

Pada *memory-mapped i/o*, *input register* dan *output register* diperlakukan sama dengan lokasi *memory* pada umumnya. Dengan demikian jika ditemukan perintah *read* dan melibatkan alamat *input register*, maka perintah tersebut melibatkan perangkat input.

Sehingga *input/ output instruction* akan diperlakukan sama baik saat melibatkan alamat pada *memory* maupun pada perangkat i/o.

Keunggulan teknik ini adalah tidak membutuhkannya perintah khusus *input/ output instruction* pada perangkat i/o, sehingga lebih banyak perintah- perintah yang bisa digunakan.

Dan kekurangannya adalah sebagian alokasi *memory* harus disisihkan untuk pengalamatan perangkat i/o, sehingga jumlah total alokasi *memory* yang tersedia akan berkurang.



Gambar 11.9 - Memory Mapped I/O arrangement

## ***I/O module***

### **Fungsi utama :**

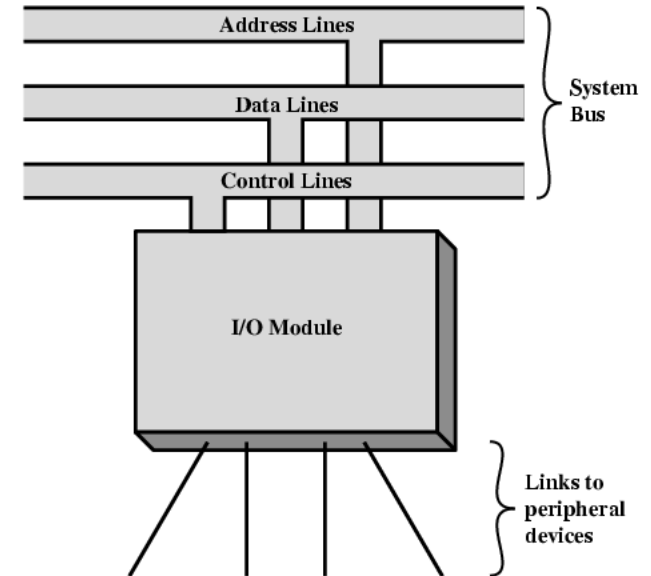
- Sebagai perantara CPU dan Memory
- Sebagai perantara antar satu peripheral dan peripheral lain

### **External devices :**

- Human readable (Screen, printer, keyboard)
- Machine readable (Monitoring and control)
- Communication (Modem, Network Interface Card)

### **Kategori fungsi *i/o module* :**

- Control & Timing
- CPU Communication
- Device Communication
- Data Buffering
- Error Detection



Gambar 11.10 - I/O Module

# Teknik operasi i/o

---

Ada 3 teknik operasi i/o yang bisa digunakan dalam mengatur akses perangkat i/o kepada *memory*.

- ***Programmed I/O***

Terjadi pertukaran data antara *processor* dan *i/o module*, *processor* mengeksekusi sebuah program yang memberikan *processor* akses langsung atas operasi i/o termasuk mengetahui status perangkat, mengirim perintah *read* dan *write* serta pengiriman data. Saat *processor* memberikan perintah kepada *i/o module*, *processor* harus menunggu hingga *i/o module* menyelesaikan operasi i/o sehingga *processor* akan membuang banyak waktu pada tahap ini.

- ***Interrupt-driven I/O***

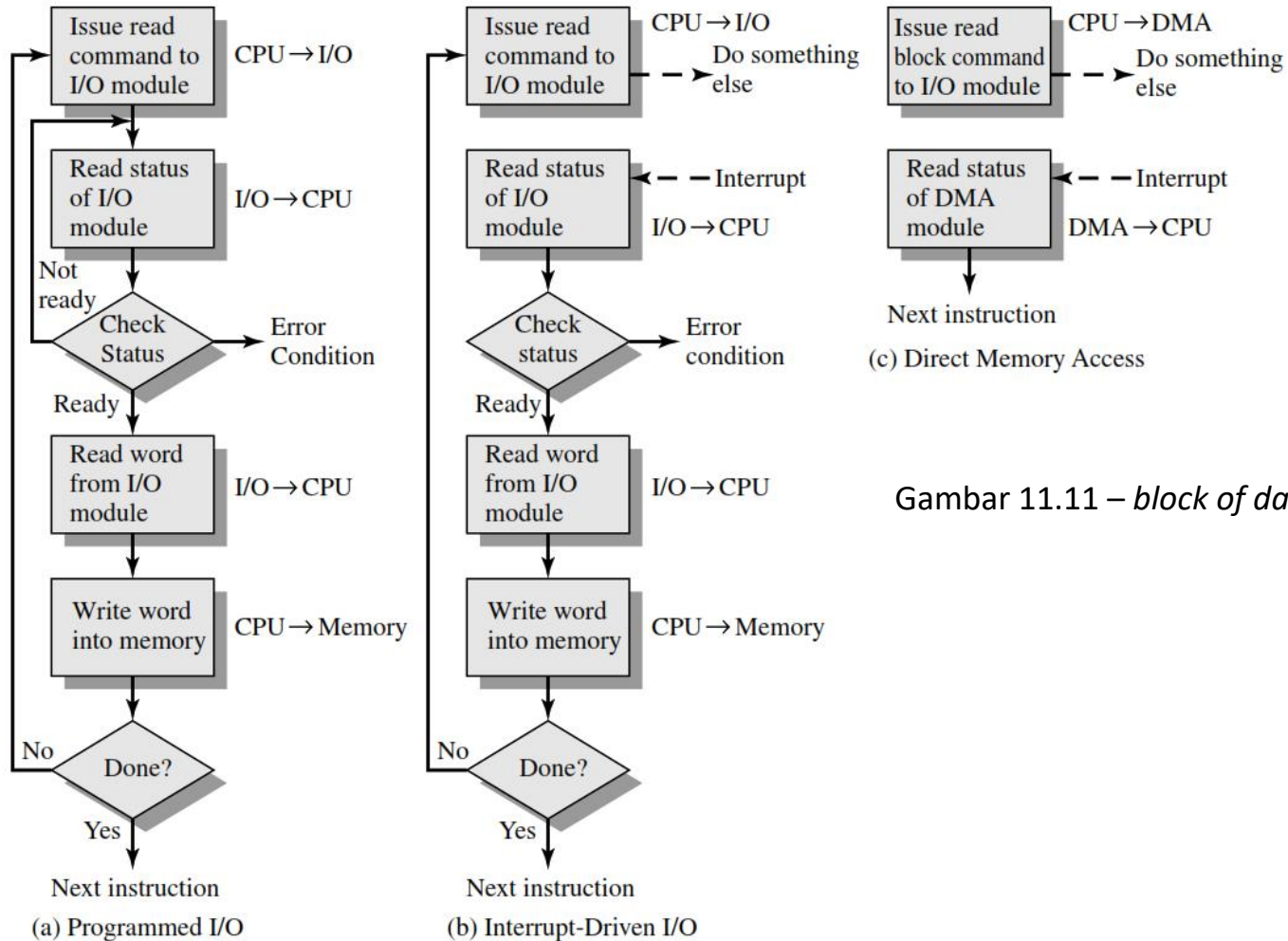
Pada teknik kedua, *processor* mengirimkan *i/o command* lalu melanjutkan eksekusi intruksi- instruksi lain dan saat *i/o module* menyelesaikan perintah ia meng-interupsi *processor*. Sehingga dengan teknik ini *processor* tidak membuang banyak waktu untuk menunggu respon *i/o module*.

Kedua teknik ini *i/o module* masih bergantung kepada *processor* untuk mengambil data dari *memory* sebagai output dan menyimpan data ke *memory* sebagai input.

- Direct memory access (DMA)

Dibahas pada pertemuan mendatang

# Blok data operasi i/o



Gambar 11.11 – *block of data input*

## SUMBER PUSTAKA

- Mostafa dan Hesham.2005.Fundamentals Of Computer Organization And Architecture.New Jersey : Wiley Interscience
- W. Stallings.2016. Computer Organization and Architecture.Hoboken:Pearson Education
- A.S. Tanenbaum..Structured Computer Organization.New Jersey : Pearson Prentice Hall





# ***THANKS***

ANY QUESTIONS?