

# ORGANISASI ARSITEKTUR KOMPUTER CENTRAL PROCESSING UNIT

**STT TERPADU NURUL FIKRI**  
**TEKNIK INFORMATIKA**  
**2017**

## 20. Thaahaa

قَالَ رَبِّ اشْرَحْ لِي صَدْرِي ﴿٢٥﴾

25. Berkata Musa: "Ya Tuhanku, lapangkanlah untukku dadaku<sup>[915]</sup>,

وَيَسِّرْ لِي أَمْرِي ﴿٢٦﴾

26. dan mudahkanlah untukku urusanku,

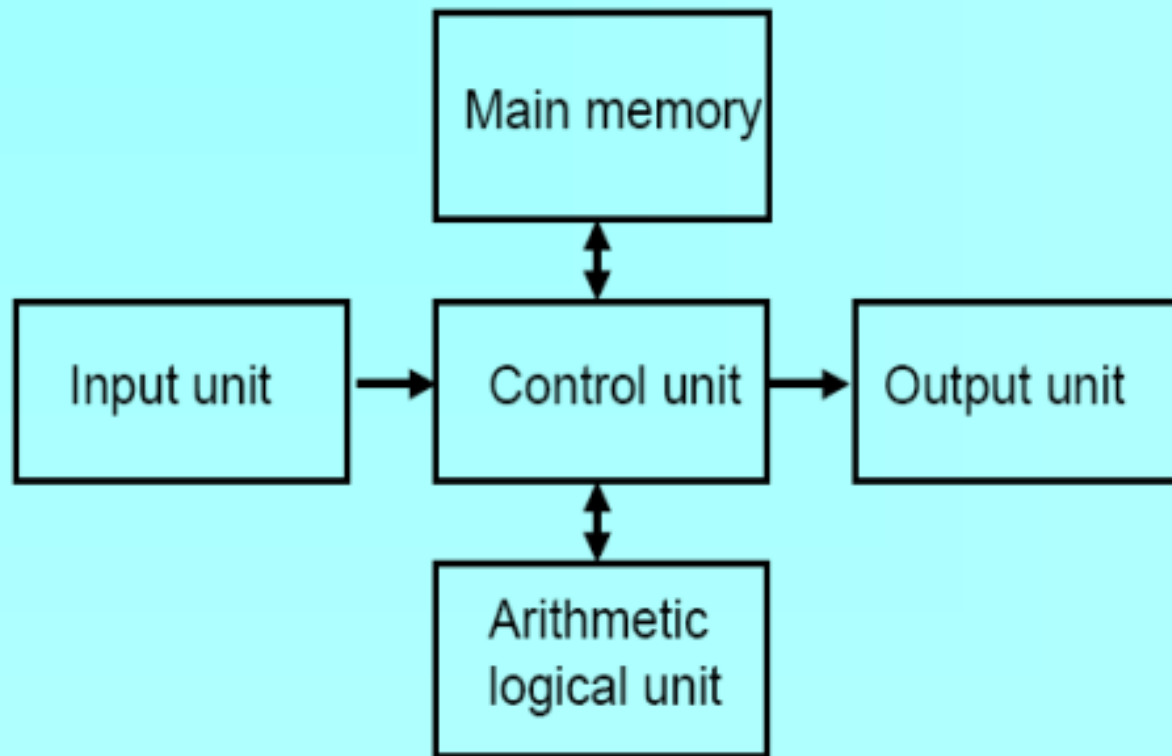
وَأَحْلِلْ عُقْدَةَ مِنِّ لِسَانِي ﴿٢٧﴾

27. dan lepaskanlah kekakuan dari lidahku,

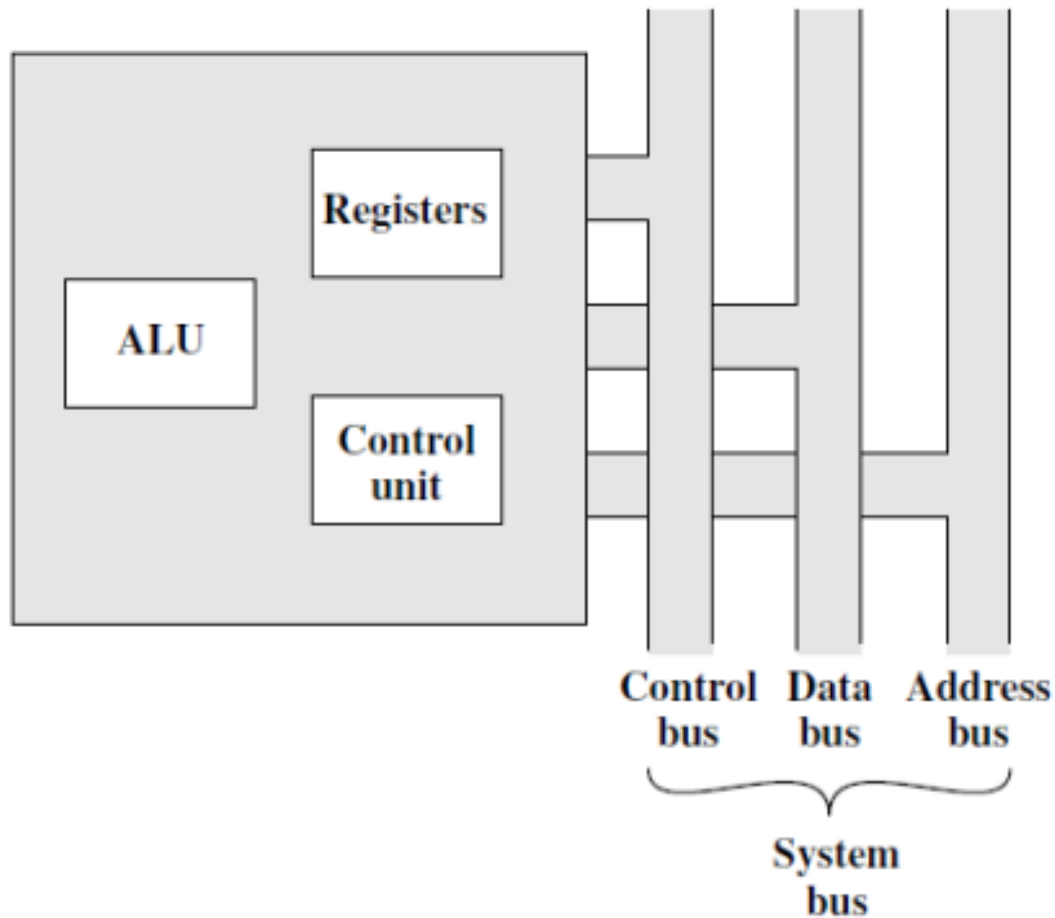
# CPU DI VON NEUMANN ARCHITECTURE

## von Neumann Architecture: Structure

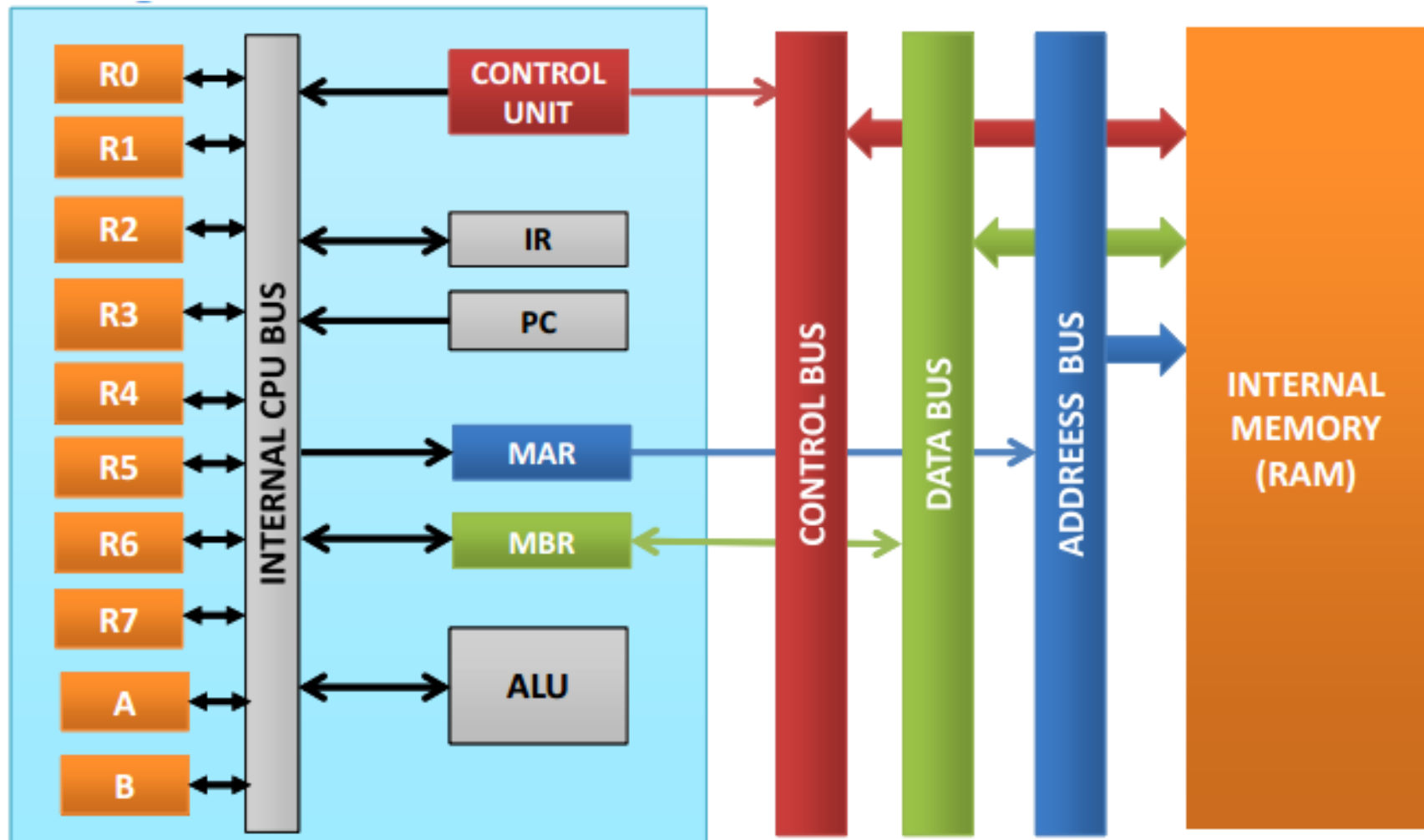
---



# BAGAN INTERKONEKSI CPU KE SISTEM BUS



# BAGAN STRUKTUR INTERNAL CPU & INTERKONEKSI



# BAGIAN-BAGIAN CPU

- Control Unit  
Bagian yang mengatur tahapan eksekusi instruksi dan mengirimkan sinyal kontrol ke bagian-bagian lain terkait
- ALU (Arithmetic and Logic Unit)  
Bagian yang melakukan pengolahan aritmatika dan logika
- IR (Instruction Registers)  
Tempat menyimpan instruksi yang akan dieksekusi
- PC (Program Counter)  
Tempat menyimpan alamat dari instruksi berikutnya
- GPR (General Purpose Registers): R0, R1, R2 ...  
Tempat-tempat penyimpanan untuk membantu pengolahan data
- MAR (Memory Address Register)  
Tempat menyimpan alamat memori untuk operasi READ / WRITE
- MBR (Memory Buffer Register)
  - Berisi data untuk operasi WRITE memori
  - Berisi data hasil operasi READ memori

- **Pemrograman**  
Adalah pembuatan/penyusunan kode program sebagai representasi instruksi-instruksi yang harus dilakukan komputer, agar komputer dapat melakukan sesuatu sesuai tujuan program.
- **Compile / Kompilasi**  
Adalah proses mengubah kode program ke kode mesin instruksi agar dapat dipahami komputer dan dilaksanakan / dieksekusi.



# JENIS BAHASA PEMROGRAMAN (1)

## 1. Bahasa Tingkat Tinggi

Adalah bahasa pemrograman yang lebih memudahkan manusia untuk memprogram, lebih mendekati bahasa manusia, namun mengurangi efisiensi ketika harus diterjemahkan ke kode mesin untuk dieksekusi.

Misal: Bahasa C, Basic, Java. Contoh:

```
If X=Y then X=X+2
```

## 2. Bahasa Tingkat Rendah

Adalah bahasa pemrograman yang lebih dekat ke kode mesin, sehingga sangat efisien ketika diterjemahkan ke kode mesin untuk dieksekusi komputer. Misal: Bahasa Assembly. Contoh:

```
MOV A,X  
MOV B,Y  
CJNE A,B,NEXT  
ADD A,#2  
MOV X,A
```

```
NEXT:  ...
```



# JENIS BAHASA PEMROGRAMAN (2)

Bahasa Tingkat Tinggi	Bahasa Tingkat Rendah
<ul style="list-style-type: none"><li>+ Lebih mudah digunakan manusia dalam memprogram</li><li>+ Lebih mudah dipahami dan didokumentasikan</li><li>+ Lebih mudah dikembangkan dan diteruskan dari waktu ke waktu, dari programmer yang satu ke programmer yang lain</li><li>- Lebih banyak pemakaian memori</li><li>- Lebih besar ukuran file program hasil kompilasi</li><li>- Lebih lambat ketika dieksekusi</li></ul>	<ul style="list-style-type: none"><li>- Lebih mendekati kode mesin, banyak baris harus ditulis untuk suatu kerja</li><li>- Semakin besar program, semakin sulit dipahami dan didokumentasikan</li><li>- Ketika program kompleks, jumlah baris sangat banyak, tidak mudah diteruskan dari waktu ke waktu atau dilanjutkan dari programmer yang satu ke yang lain</li><li>+ Sangat efisien dalam pemakaian memori</li><li>+ Lebih kecil ukuran file program hasil kompilasi</li><li>+ Sangat cepat ketika dieksekusi</li></ul>

# CONTOH INSTRUKSI (BAHASA ASSEMBLY)

## ADD A,R0

adalah instruksi untuk menjumlahkan isi register A dengan register R0 dimana hasilnya penjumlahannya nanti disimpan di register A.

Misal sebelum instruksi, nilai di A=1 R0=2

Maka sesudah eksekusi instruksi, nilai di A=3 R0=2

Pekerjaan	Contoh	Penjelasan
Pengisian data	MOV B,#8	B <- 8
Penjumlahan	ADD A,#4	A <- A + 4
Pengurangan	SUBB A,#3	A <- A - 3
Perkalian	MUL AB	A <- A * B
Logika AND	ANL A,#01010101B	A <- A AND 01010101B
Logika OR	ORL A,#00110011B	A <- A OR 00110011B
Logika NOT	CPL A	A <- <u>A</u>
Percabangan	CJNE A,#0,MULAI	IF A≠0 GOTO LABEL MULAI

# TAHAPAN PROSES INSTRUKSI DI CPU

1. Fetch Instruction	2. Decode Instruction	3. Calculate Operand	4. Fetch Operand	5. Execute Instruction	6. Store Result
yaitu proses mengambil instruksi yang harus dieksekusi, dari memori ke IR	yaitu proses menerjemahkan isi instruksi untuk diketahui apa yang harus dilakukan oleh prosesor	yaitu proses menghitung / mengidentifikasi letak operand yang harus disiapkan untuk operasi instruksi	yaitu proses mengambil operand yang diperlukan untuk siap dioperasikan	yaitu proses melaksanakan operasi instruksi (biasanya di ALU)	yaitu proses menyimpan hasil operasi instruksi

# CONTOH PROSES INSTRUKSI: ADD A,R0

## **Fetch Instruction**

1. Isi alamat instruksi yang akan dieksekusi, disalurkan dari di PC ke MAR dan dari MAR ke jalur alamat memori melalui address bus.
2. Control Unit mengirimkan sinyal READ ke memori melalui control bus.
3. Memori menyalurkan instruksi yang ada di alamat (yang diberitahukan MAR) ke MBR melalui data bus.
4. Instruksi dibawa dari MBR ke IR.

## **Decode Instuction**

5. Instruksi di IR diterjemahkan sebagai operasi penjumlahan dengan 2 operand

## **Calculate Operand**

6. Operand instruksi diidentifikasi, yaitu register A dan R0.

## **Fetch Operand**

7. Isi register A dan R0 dibawa ke ALU.

## **Execute Instruction**

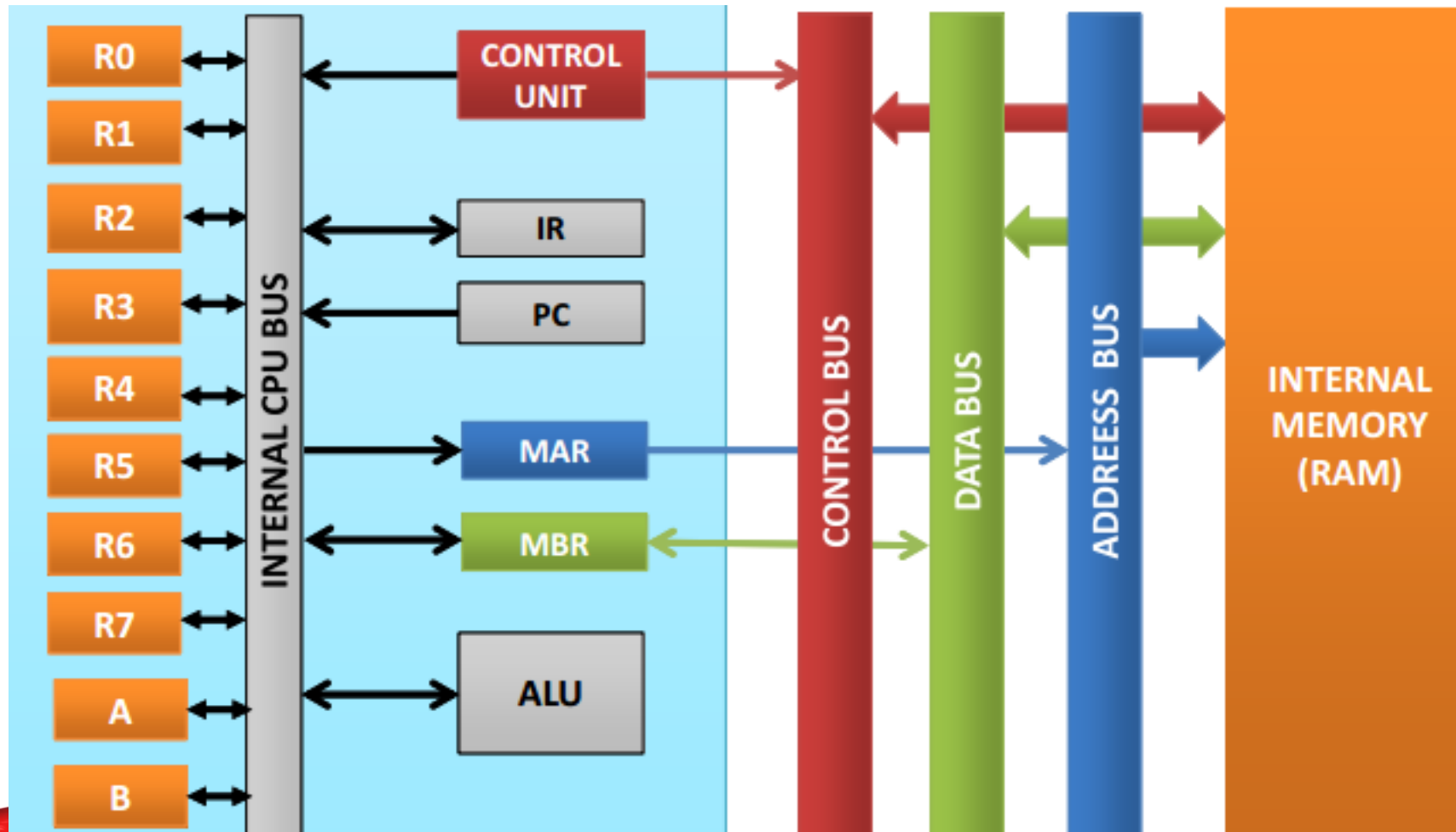
8. ALU mengeksekusi operasi penjumlahan isi A dengan isi R0.

## **Store Result**

9. Hasil penjumlahan dibawa dan disimpan ke register A.



# BAYANGKAN TAHAPAN PROSES DI BAGAN



# QUIZ

- Konversikan tiap angka-angka berikut ke format biner (8 bit) , Okta (2 digit), dan hexa (2 digit).
  - a. 212 (10)
  - b. 121 (10)
  - c. 192 (10)
- Konversikan tiap angka hexa berikut ke format biner, octal dan desimal
  - CA (16)
  - F4 (16)
  - 99 (16)
- Sederhanakan dan Buktikan
  - $Y = ABC + A\bar{B}\bar{C} + \bar{A}\bar{B}C$

# TERIMA KASIH



*Thank you very much for your kind attention*