



STIKOM BALI

# ORGANISASI KOMPUTER

## Materi 4: Assembly Language Programming

I Nyoman Kusuma Wardana  
Sistem Informasi STMIK STIKOM Bali

# MATERI PERKULIAHAN

---

- Pendahuluan
- Mesin sederhana
- Mnemonic dan sintaks



# MATERI PERKULIAHAN

---

- Pendahuluan
- Mesin sederhana
- Mnemonic dan sintaks



# PENDAHULUAN

- Wlpn bhs **tingkat tinggi** mengalami perkembangan signifikan → **bhs rakitan tetap diperlukan** pd bbrp kasus
- Bhs rakitan (**assembly**) → mnghasilkan **kode mesin** jauh **lebih kecil & lebih cepat** drpd kode yg dihasilkan compiler bhs tingkat tinggi
- **Kode kecil & cepat** → sangat penting terutama utk aplikasi **embedded & portable**

# PENDAHULUAN

- **Kode mesin** → kumpulan/koleksi instruksi2 mesin yg diwakili dlm bentuk bil. Biner
- Program yg ditulis **pd level yg lebih tinggi** dr bhs mesin harus **diterjemahkan** → **compiler**
- **Bhs rakitan** → perwakilan simbolik dr bhs mesin
- **Bhs mesin** → **murni biner**,
- namun **bhs rakitan** → **pemetaan langsung kode biner ke bentuk simbolik** agar mudah dipahami oleh programmer

# MATERI PERKULIAHAN

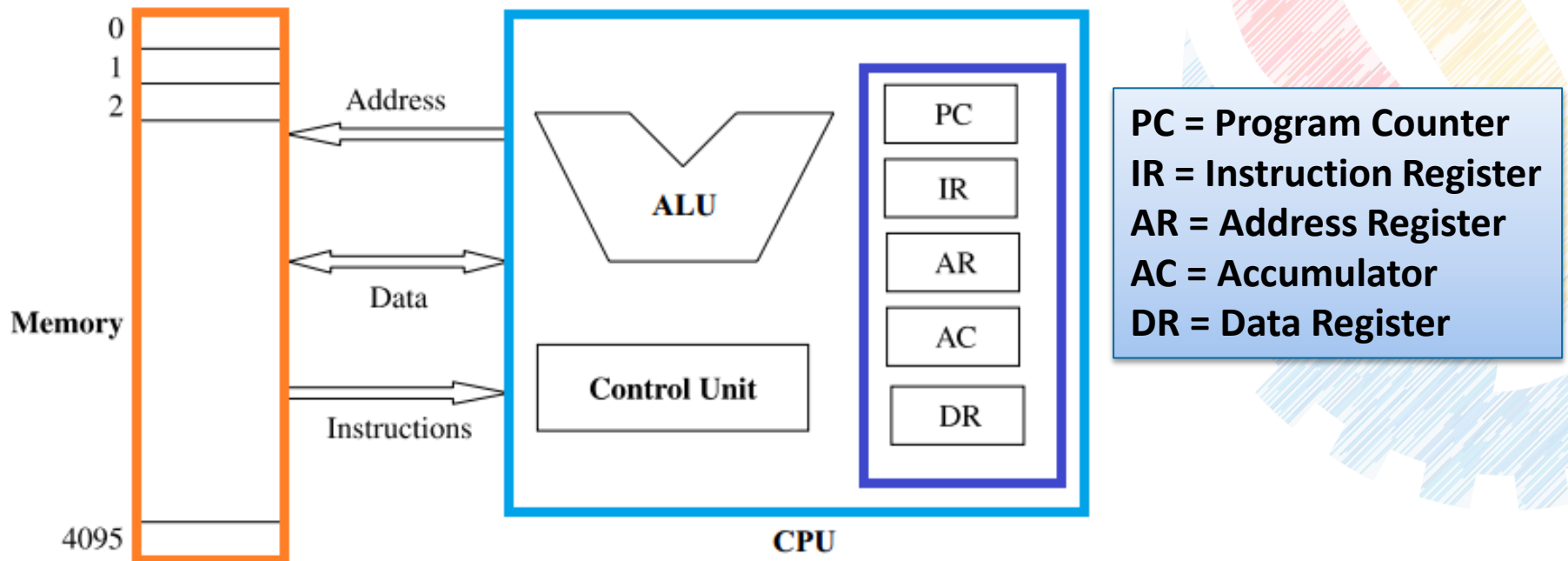
---

- Pendahuluan
- Mesin sederhana
- Mnemonic dan sintaks



# MESIN SEDERHANA

- **Bhs rakitan** → unik utk setiap mesin
- Amati **mesin sederhana**, sbb:





# MESIN SEDERHANA

- Asumsi semua **register** adlh **16-bit**
- Asumsikan jg **prosesor** sederhana kita memiliki **3 tipe instruksi**:
  1. **Data transfer** → *load, store, move* antara AC dan DR
  2. **Data processing** → *add, subtract, and, not*
  3. **Program control** → *jump*
- Instruksi → **16-bit**: **4-bit** = opcode, **12-bit** = alamat



# MESIN SEDERHANA

## ■ Tabel set instruksi sebuah prosesor sederhana

Operation code	Operand	Meaning of instruction
0000		Stop execution
0001	<i>adr</i>	Load operand from memory (location <i>adr</i> ) into AC
0010	<i>adr</i>	Store contents of AC in memory (location <i>adr</i> )
0011		Copy the contents AC to DR
0100		Copy the contents of DR to AC
0101		Add DR to AC
0110		Subtract DR from AC
0111		And bitwise DR to AC
1000		Complement contents of AC
1001	<i>adr</i>	Jump to instruction with address <i>adr</i>
1010	<i>adr</i>	Jump to instruction <i>adr</i> if AC = 0

# MESIN SEDERHANA

## Contoh:

- Tulislah sebuah **bahasa mesin** utk **menambahkan** isi dr memori yg beralamat di **12 (00C – hex)** dan isi dr memori yg beralamat di **14 (00E - hex)**, kemudian simpan hasilnya di alamat **16 (010 – hex)**
- Asumsi nilai awal di lokasi **12** adlh **350**, lokasi **14** adlh **96**, dan lokasi **16** adlh **0**.

# MESIN SEDERHANA

## ■ Jawab:

Memory location (bytes)	Binary instruction	Description
0000 0000 0000	0001 0000 0000 1100	Load the contents of location 12 in AC
0000 0000 0010	0011 0000 0000 0000	Move contents of AC to DR
0000 0000 0100	0001 0000 0000 1110	Load the contents of location 14 into AC
0000 0000 0110	0101 0000 0000 0000	Add DR to AC
0000 0000 1000	0010 0000 0001 0000	Store contents of AC in location 16
0000 0000 1010	0000 0000 0000 0000	Stop
0000 0000 1100	0000 0001 0101 1110	Data value 350
0000 0000 1110	0000 0000 0110 0000	Data value is 96
0000 0001 0000	0000 0000 0000 0000	Data value is 0



# MESIN SEDERHANA

## Contoh:

- Representasikan program tsb ke dlm bilangan **heksadesimal**!
- Jawab:

Memory location (bytes)	Hex instruction
000	100C
002	3000
004	100E
006	5000
008	2010
00A	0000
00C	015E
00E	0060
010	0000

# MATERI PERKULIAHAN

---

- Pendahuluan
- Mesin sederhana
- Mnemonic dan sintaks





# MNEMONIC DAN SINTAKS

- **Bhs rakitan** → bentuk **simbolik** dr **bahasa mesin**
- Bhs rakitan diprogram dlm bentuk singkatan → disbt sbg **mnemonic**
- **Mnemonic** → **mewakili instruksi mesin** yg aktual
- Mnemonic **mempermudah dlm mengingat** instruksi drpd bil biner/hex
- Bhs rakitan ditulis **baris per baris**

# MNEMONIC DAN SINTAKS

- Setiap baris program rakitan terdiri dpt dipecah mnjd 4:
  - *Label*
  - *Operation code (opcode)*
  - *Operand*
  - *Comment*

Label (Optional)	Operation Code (Required)	Operand (Required in some instructions)	Comment (Optional)
---------------------	------------------------------	---	-----------------------

Format bhs rakitan



# MNEMONIC DAN SINTAKS

Label (Optional)	Operation Code (Required)	Operand (Required in some instructions)	Comment (Optional)
---------------------	------------------------------	---	-----------------------

- **Label** → merupakan **identifier**, berguna jika ada percabangan atau utk mengakses data menggunakan nama simbolik
- **Opcode** → jenis **instruksi**
- **Operand** → **informasi tambahan** yg dibutuhkan opcode: constant, immediate data, register atau address
- **Comment** → komentar

# MNEMONIC DAN SINTAKS

- Asumsi terdapat bbrp instruksi utk sebuah prosesor sederhana kita, sbb:

**TABLE Assembly Language for the Simple Processor**

Mnemonic	Operand	Meaning of instruction
STOP		Stop execution
LD	<i>x</i>	Load operand from memory (location <i>x</i> ) into AC
ST	<i>x</i>	Store contents of AC in memory (location <i>x</i> )
MOVAC		Copy the contents AC to DR
MOV		Copy the contents of DR to AC
ADD		Add DR to AC
SUB		Subtract DR from AC
AND		And bitwise DR to AC
NOT		Complement contents of AC
BRA	<i>adr</i>	Jump to instruction with address <i>adr</i>
BZ	<i>adr</i>	Jump to instruction <i>adr</i> if AC = 0

# MNEMONIC DAN SINTAKS

- Ambil contoh berikut:

START	LD X	\salin isi dr lokasi X ke AC
	MOVAC	\salin isi AC ke DR
	BRA START	\lompat ke label START

- Program akan diulang terus-menerus dlm suatu loop dgn label **START**

# MNEMONIC DAN SINTAKS

## Contoh:

- Tulislah sebuah **bahasa rakitan** utk **menambahkan** isi dr memori yg beralamat di **12 (00C – hex)** dan isi dr memori yg beralamat di **14 (00E - hex)**, kemudian simpan hasilnya di alamat **16 (010 – hex)**
- Asumsi nilai awal di lokasi **12** adlh **350**, lokasi **14** adlh **96**, dan lokasi **16** adlh **0**.

# MNEMONIC DAN SINTAKS

- Program dalam **bhs rakitan** adlh sbb:

LD	X	\ AC ← X
MOVAC		\ DR ← AC
LD	Y	\ AC ← Y
ADD		\ AC ← AC + DR
ST	Z	\ Z ← AC
X	W 350	\ pesan tempat di X dgn nilai 350
Y	W 96	\ pesan tempat di Y dgn nilai 96
Z	W 0	\ simpan hasilnya disini

# MNEMONIC DAN SINTAKS

## Contoh:

- Tulislah sebuah **bahasa rakitan** utk melakukan operasi perkalian:  $Z \leftarrow X * Y$
- $X, Y, Z$  adlh suatu lokasi memori



# MNEMONIC DAN SINTAKS

## Jawab:

- Berdasarkan instruksi sebuah prosesor sederhana kita, **TIDAK** terdpt instruksi **perkalian**
- Solusi → lakukan operasi **penjumlahan berkali-kali**
- $Z \leftarrow X * Y$  : tambahkan **Y** dgn dirinya sendiri berkali-kali sebanyak **X**

Mnemonic	Operand
STOP	
LD	<i>x</i>
ST	<i>x</i>
MOVAC	
MOV	
ADD	
SUB	
AND	
NOT	
BRA	<i>adr</i>
BZ	<i>adr</i>



# MNEMONIC DAN SINTAKS

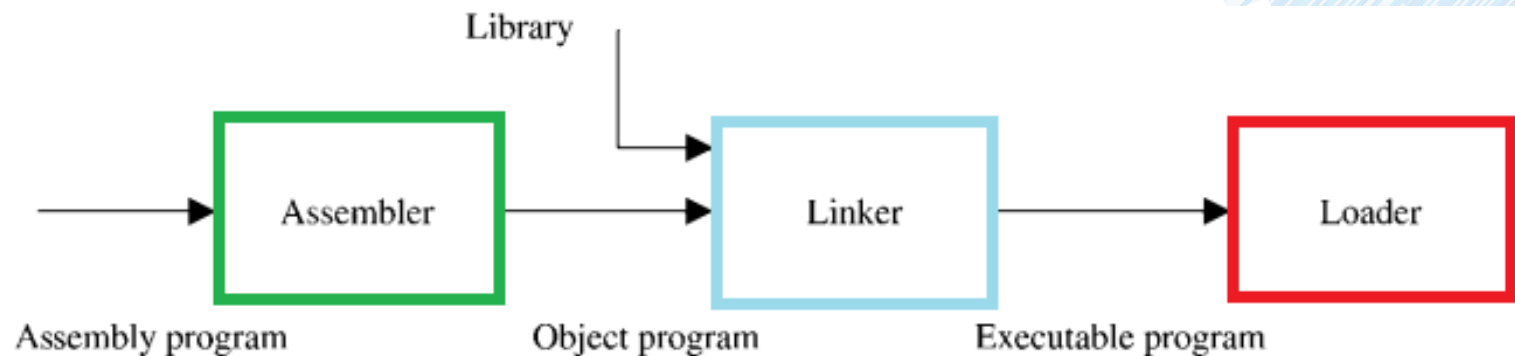
- Kita gunakan **N** sbg **counter** utk menginisialisasi **X** dan **dikurangi satu-per-satu** di setiap langkah
- Instruksi **BZ** digunakan sbg indikator ketika **N** mencapai **0**.
- Kita gunakan memori lokasi **ONE** utk menyimpan konstanta **1**
- Asumsi inisialisasi: **X = 5** dan **Y = 15**
- Selanjutnya simpan hasilnya di **Z**

# MNEMONIC DAN SINTAKS

	LD	X	\ isi AC dgn isi lokasi X
	ST	N	\ simpan AC di N
LOOP	LD	N	\ $AC \leftarrow N$
	BZ	EXIT	\ lompat ke EXIT jika $AC = 0$
	LD	ONE	\ $AC \leftarrow 1$
	MOVAC		\ $DR \leftarrow AC$
	LD	N	\ $AC \leftarrow N$
	SUB		\ $N = N - 1$
	ST	N	\ simpan hasil N
	LD	Y	\ $AC \leftarrow Y$
	MOVAC		\ $DR \leftarrow AC$
	LD	Z	\ $AC \leftarrow Z$
	ADD		\ tambahkan Y ke Z ( $Z = Z + Y$ )
	ST	Z	\ simpan nilai terbaru Z
	BRA	LOOP	
EXIT	STOP		
	X W	5	\ pesan tempat di X senilai 5
	Y W	15	\ pesan tempat di Y senilai 15
	Z W	0	\ pesan tempat di Z senilai 0
	ONE W	1	\ ONE bernilai 1
	N W	0	\ N bernilai 0

# ASSEMBLY & EXECUTION OF PROGRAMS

- **Bhs rakitan** → hrs diterjemahkan ke **bhs mesin**
- **Asembler** membaca **program sumber** dan menghasilkan **bil biner**
- **Linker** akan mengkombinasikan pustaka dan **object program** → menghasilkan **exe.**



# DAFTAR PUSTAKA

- Abd-El-Barr, M., El-Rewini, H., **Fundamentals of Computer Organization and Architecture**, John Wiley&Sons, Inc.
- Stallings, W., 2010, **Computer Organization and Architecture: Designing for Performance** 8<sup>th</sup> edition, Prentice Hall
- Hamacher, C., Vranesic, Z., Zaky, S., Manjikian, N., 2012, **Computer Organization and Embedded Systems** 6<sup>th</sup> edition, McGrawHill