STIKOM BALI

ORGANISASI KOMPUTER

Materi 3: Instruction Set

I Nyoman Kusuma Wardana

Sistem Informasi STMIK STIKOM Bali

Materi Perkuliahan

- Lokasi dan Pengalamatan Memori
- Tipe-tipe Instruksi
- Mode Pengalamatan

Materi Perkuliahan

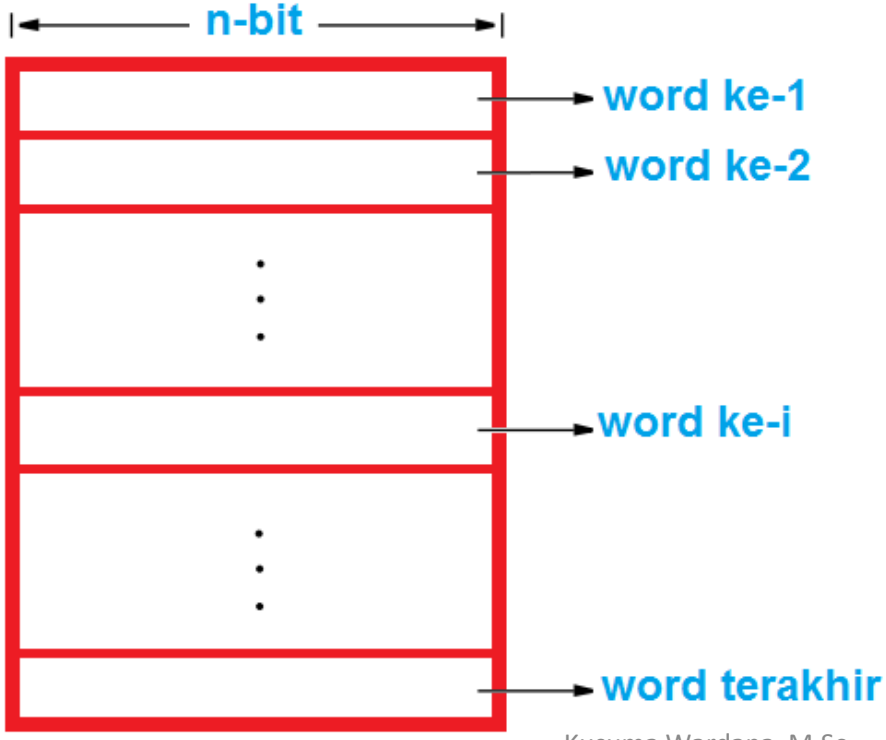
- **Lokasi dan Pengalamatan Memori**
- Tipe-tipe Instruksi
- Mode Pengalamatan

Lokasi & Pengalamatan Memori

- **Memori** → terdiri dr jutaan **sel** penyimpanan
- Setiap sel dpt **menyimpan** informasi → berupa data **0** atau **1**
- **1 bit tunggal** → **tdk cukup** utk menyimpan informasi
- Oleh karena itu, **memori** bekerja dgn **total bit tertentu**

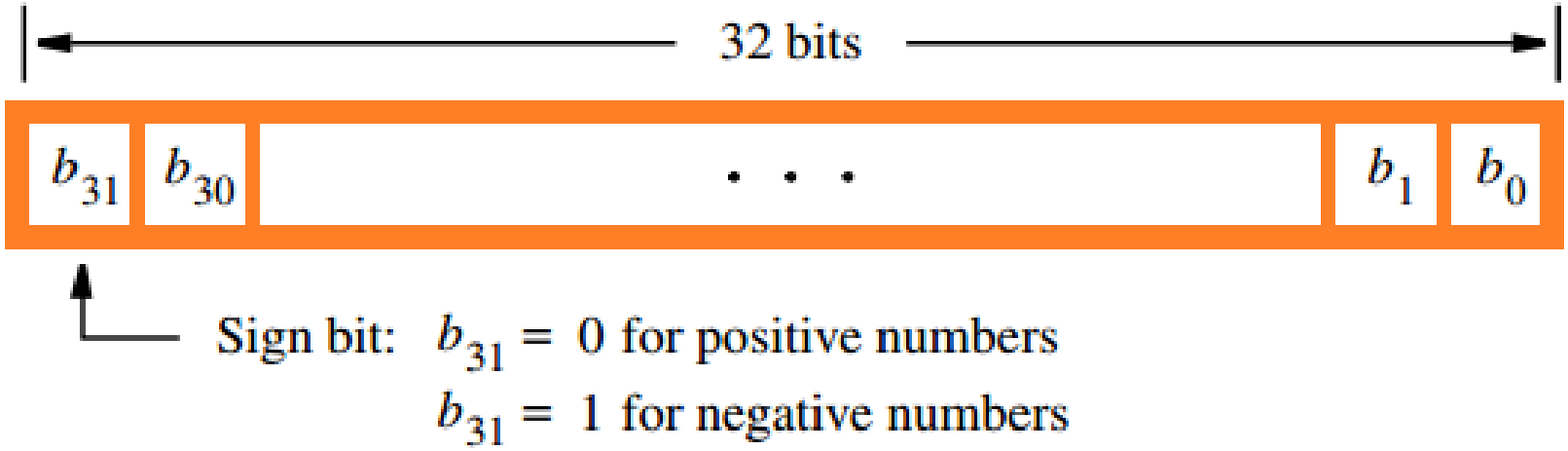
Lokasi & Pengalamatan Memori

- **Asumsi** → memori diorganisasikan dgn **n-bit**
- Setiap grup **n**-bit disebut → **word**
- **n** → **panjang word** (umumnya **16 -64**)



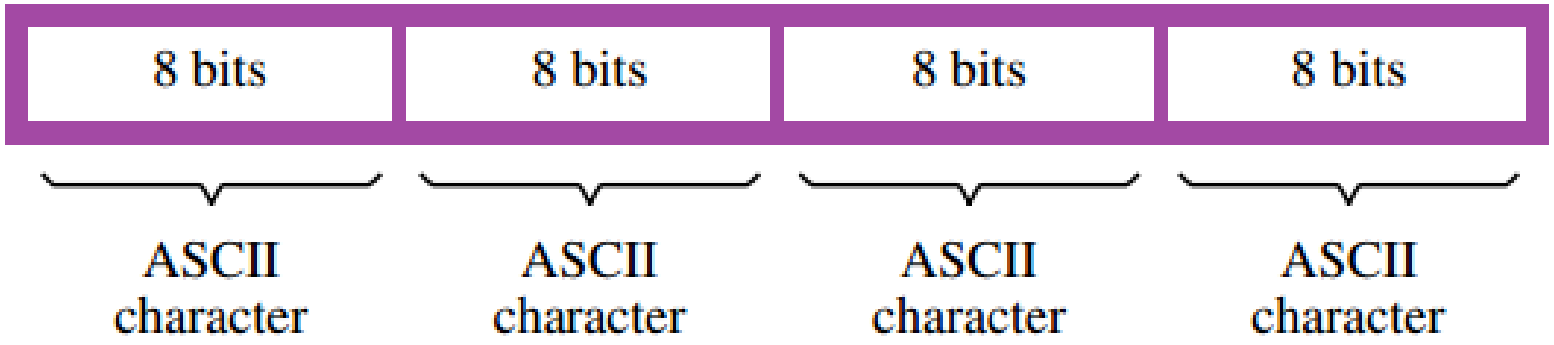
Lokasi & Pengalamatan Memori

- Asumsi panjang word adlh **32**
- Jadi, **1 word** → dpt menyimpan **32-bit bilangan bertanda**



Lokasi & Pengalamatan Memori

- **8-bit = 1byte**
- Kalau **n = 32-bit**, maka dpt menyimpan **4** karakter ASCII



Lokasi & Pengalamatan Memori

- Utk **mengakses memori** → digunakan **alamat** (***address***) utk setiap lokasi
- Asumsi jika kita mempunyai ***k*-bit** alamat, maka:
- Banyak ruang yg bisa dialamati: 2^k
- Dgn alokasi alamat: **0** sampai $2^k - 1$

Lokasi & Pengalamatan Memori

- **Penyederhanaan**:

  $2^{10} = 1024 \approx 10^3$ → **1 kilobyte** (1 KB)

  $2^{11} = 2048 \approx 2 \times 10^3$ → **2 kilobyte** (2 KB)

- **Contoh**: tanpa menggunakan kalkulator, estimasi nilai **2^{24} byte**.
- **Jawab** :

  $2^{24} = 2^{20} \times 2^4$

  $2^{20} \approx 1$ juta sedangkan $2^4=16$

  Jadi sekitar 16 juta → **16 MB**

Lokasi & Pengalamatan Memori

- Mohon diingat:

  2^{10} = 1K (Kilo)

  2^{20} = 1M (Mega)

  2^{30} = 1G (Giga)

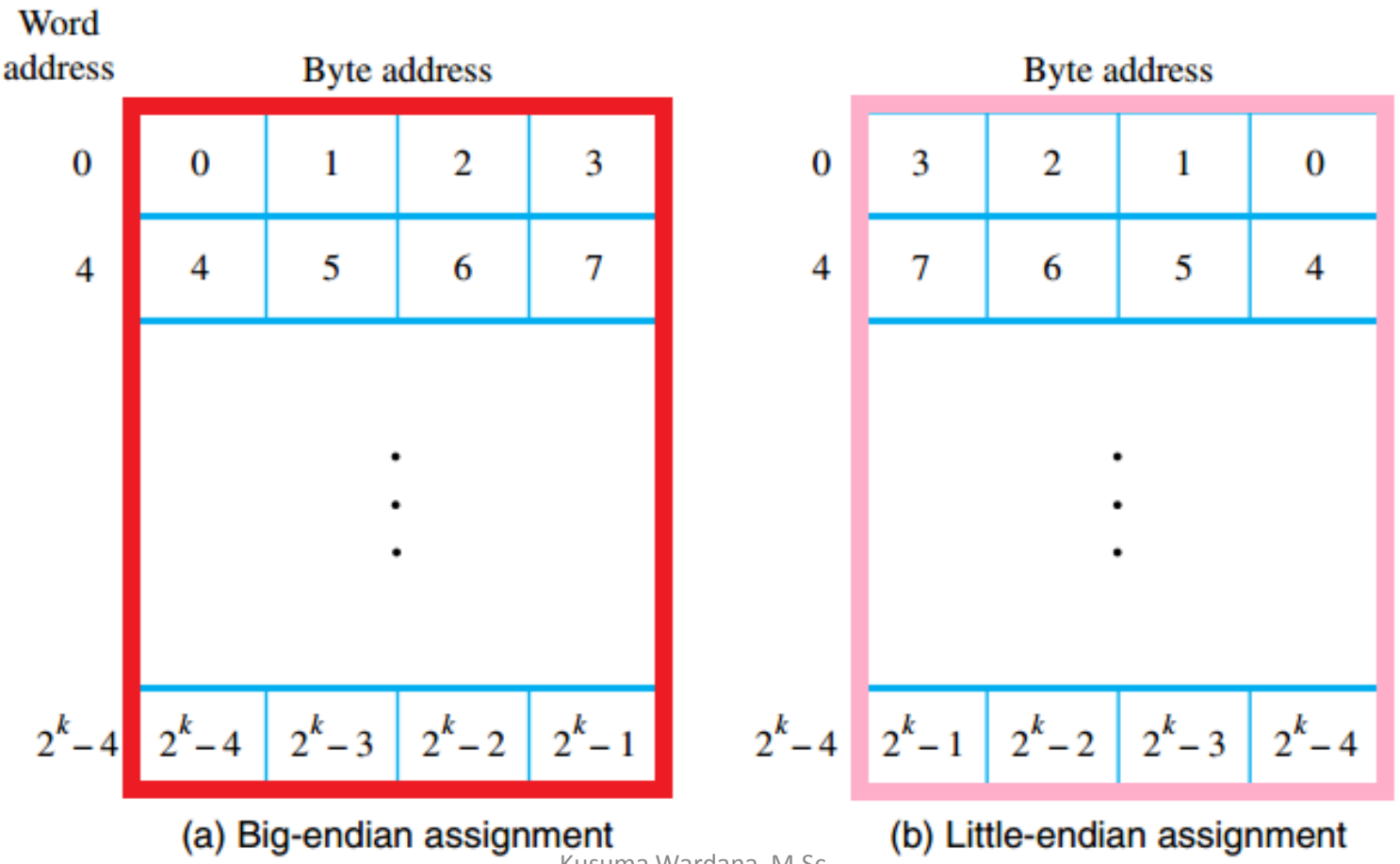
  2^{40} = 1T (Tera)

Contoh: estimasikan 2^{35}

Jawab: 32 Giga

Lokasi & Pengalamatan Memori

- **Big-Endian & Little-Endian**



Materi Perkuliahan

- Lokasi dan Pengalamatan Memori
- **Tipe-tipe Instruksi**
- Mode Pengalamatan

Tipe Instruksi

- Informasi yg terlibat dlm berbagai bentuk operasi CPU harus di**alamat**i
- Informasi ini dsbt→ operand
- Paling tidak informasi ini terdiri dr **2** jenis:
  1. **Opcode**: operasi yg akan dijalankan
  2. **Address**: alamat dlm proses operasi tsb

TIPE INSTRUKSI

- Instruksi dpt digolongkan berdasarkan **jumlah** operand:
  1. ***Three*-address**
  2. ***Two*-address**
  3. ***One-and-half*-address**
  4. ***One*-address**
  5. ***Zero*-address**

Tipe Instruksi

- Dalam suatu instruksi, terdapat istilah:
  1. ***Operation***
  2. ***Source***
  3. ***Destination***
- Sedangkan **jenis operasi** yg mungkin dijalankan, misalnya: **add**, **subtract**, **write** atau **read**
- ***Source*** bisa berupa:
  1. **Konstanta**
  2. **Nilai** yg tersimpan di **register**
  3. **Nilai** yg tersimpan di **memori**

TIPE INSTRUKSI

- **Three-address**

```
operation add-1, add-2, add-3
          source   destination
```

Penjelasan:

- add-1, add-2, add-3 → **alamat** register atau memori

Contoh:

```
ADD R1, R2, R3
```

Jumlahkan isi **R1** dengan **R2** kemudian simpan hasilnya di **R3**

TIPE INSTRUKSI

- **Two-address**

```
operation add-1, add-2
```

Penjelasan:

- add-1, add-2 → **alamat** register atau memori

Contoh:

```
ADD R1, R2
```

Jumlahkan isi **R1** dengan **R2** kemudian simpan hasilnya di **R2**

- Ekuivalen dgn: **`ADD R1, R2, R2`**

TIPE INSTRUKSI

- **One-address**

```
operation add-1
```

Penjelasan:

- add-1 → **alamat** register atau memori

Contoh:

```
ADD R1
```

Jumlahkan isi **R1** dengan **Racc** **(akumulator)** kemudian simpan hasilnya di **Racc**

- Ekuivalen dgn: **`ADD R1, Racc, Racc`**
- atau: **`ADD R1, Racc`**

TIPE INSTRUKSI

- **One-and-half-address**

```
operation add-1,add-2
```

Penjelasan:

- add-1 → **alamat** register atau memori

Contoh:

```
ADD B, R1
```

Jumlahkan isi **register R1** dgn isi dr **memori** yg berlokasi di **B** dan simpan hasilnya di **R1**

- Memakai 2 pengalamatan yg **berbeda**, yaitu **register** dan **memori**

TIPE INSTRUKSI

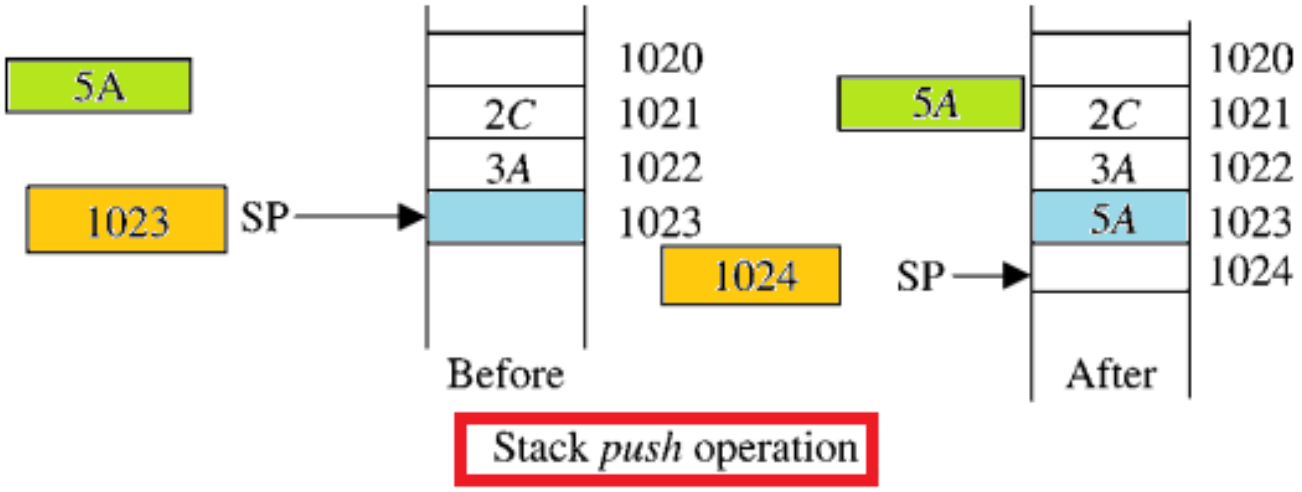
- **Zero-address**

```
operation (SP)+,(SP)
```

Penjelasan:

- SP→ **stack pointer**
- Operasi ini melibatkan proses: **push** dan **pop**
- **Push** → nilai SP mengindikasikan **lokasi** dimana nilai tertentu akan **disimpan**
- **Pop** → nilai SP **dikurangi** satu dulu, kemudian nilai pd alamat SP tersebut selanjutnya **diambil**

Tipe Instruksi



TIPE INSTRUKSI

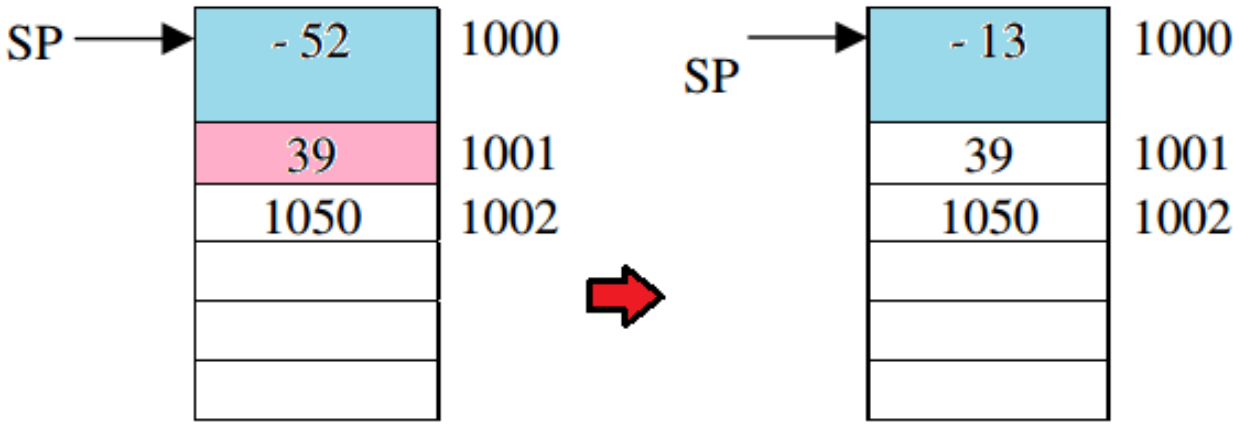
- **Zero-address**

Contoh:

```
ADD (SP)+, (SP)
```

Jumlahkan isi pd lokasi **SP** tersebut dgn isi lokasi **SP+1** dan simpan kembali di **SP**.

Misal, isi SP = -52, isi SP+1 = 39. Hasil -52+39 = -13



TIPE INSTRUKSI

- Rangkuman tipe instruksi

klasifikasi instruksi	Contoh
Three-address	$ADD\ R_1,R_2,R_3$
	$ADD\ A,B,C$
Two-address	$ADD\ R_1,R_2$
	$ADD\ A,B$
One-and-half-address	$ADD\ B,R_1$
One-address	$ADD\ R_1$
Zero-address	$ADD\ (SP)+, (SP)$

Materi Perkuliahan

- Lokasi dan Pengalamatan Memori
- Tipe-tipe Instruksi
- **Mode Pengalamatan**

Mode Pengalamatan

- Berdasarkan bagaimana cara alamat dirujuk, mode pengalamatan dapat dibagi menjadi:
  - ***Immediate*** mode
  - ***Direct(Absolute)*** mode
  - ***Indirect*** mode
  - Mode lainnya (tidak dibahas)

Mode Pengalamatan

Immediate mode

- Nilai dr operand secara langsung sudah tersedia

Contoh:

```
LOAD #1000, R1
```

- Masukkan nilai desimal 1000 ke R1
- **R1** scr langsung (***immediate***) dimasukkan dgn nilai **1000**
- Tanda **#** menunjukkan **nilai dlm desimal**, **bukan** mrpkn **nilai lokasi tertentu**

Mode Pengalamatan

Direct mode

- Nilai dr operand diambil dari lokasi tertentu di memori

Contoh:

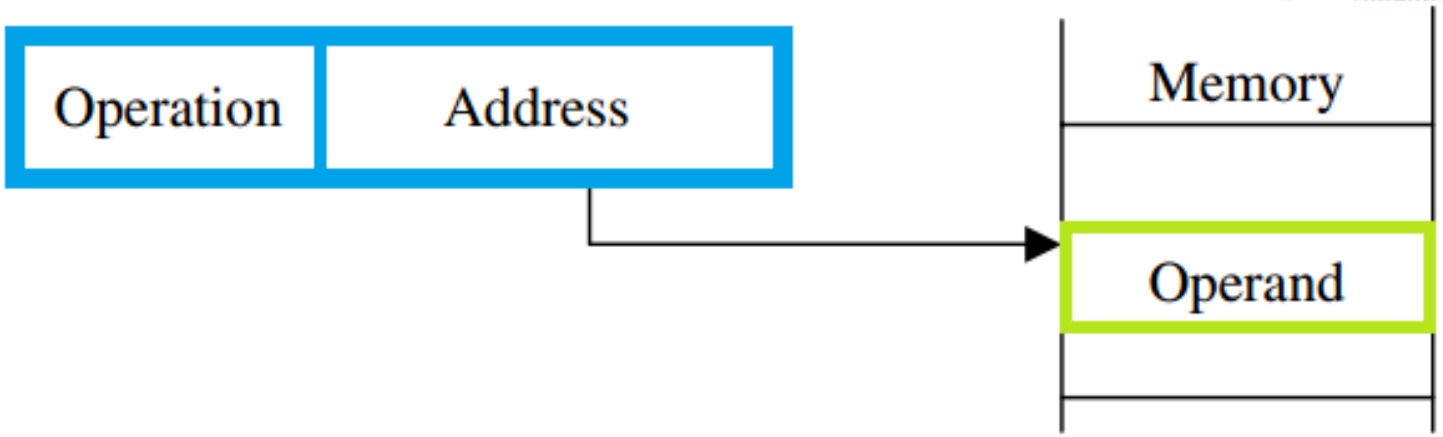
```
LOAD 1000, R1
```

- Masukkan **isi** dari memori yg **berlokasi 1000** ke **R1**
- Misal, isi dr lokasi 1000 adalah -345, maka nilai -345 akan dimasukkan ke R1

Mode Pengalamatan

Direct mode

- Nilai dr operand diambil dari lokasi tertentu di memori



Mode Pengalamatan

Indirect mode

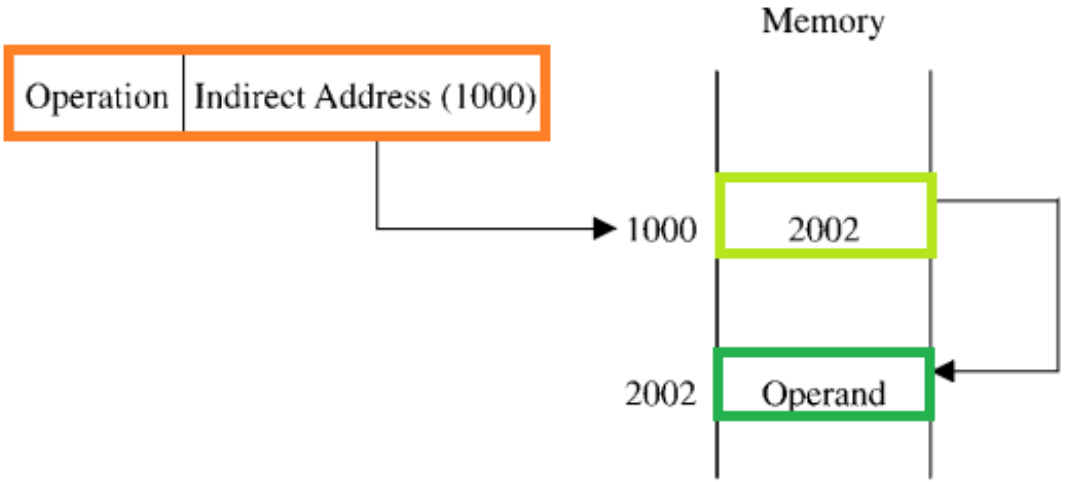
- Informasi yg terdpt dlm instruksi tidak menujukkan alamat dr operand, tapi menunjukkan alamat register atau memori yg menyimpan alamat dr operand

Contoh:

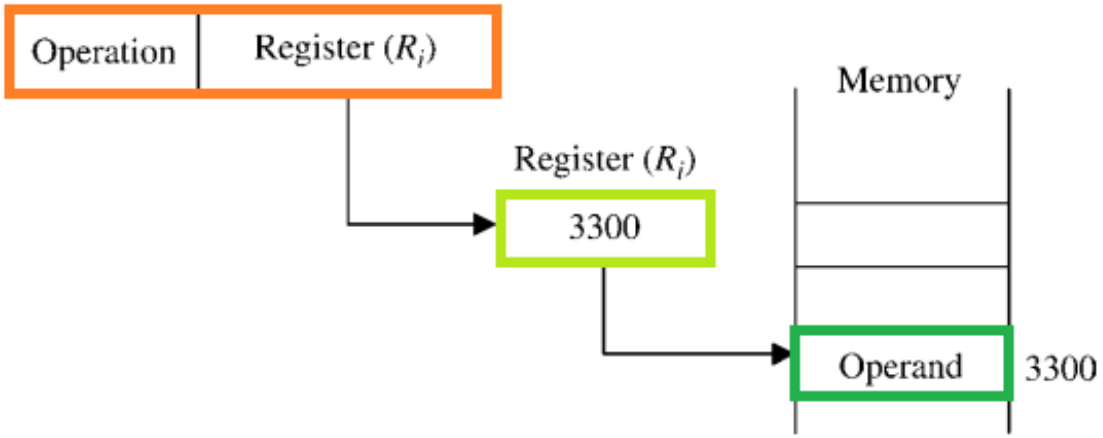
```
LOAD (1000), R1
```

- Pergi ke memori dgn alamat 1000, kemudian cek disana alamat apa yg menunjukkan operand sebenarnya
- Bisa berupa: ***register indirect addressing*** atau ***memory indirect addressing***

Mode Pengalamatan



Memory indirect addressing



Register indirect addressing

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*** 8th edition, Prentice Hall
- Hamacher,C., Vranezic, Z., Zaky, S., Manjikian, N., 2012, ***Computer Organization and Embedded Systems*** 6th edition, McGrawHill