

Dasar-Dasar Assembly 64-bit

Register

Nama	Fungsi Utama		Deskripsi
RAX	Accumulator		Digunakan untuk menyimpan hasil operasi aritmetika/logika, sering jadi register default untuk instruksi tertentu
RBX	Base Register		Digunakan sebagai penyimpan data umum, kadang sebagai alamat basis dalam akses memori
RCX	Counter Register		Sering dipakai sebagai penghitung dalam loop atau pergeseran bit
RDX	Data Register		Digunakan bersama RAX pada operasi perkalian/pembagian, juga pada instruksi I/O
RSP	Stack Pointer		Menunjuk posisi teratas stack, otomatis berubah ketika PUSH atau POP
RBP	Base Pointer		Digunakan untuk menandai awal stack frame fungsi, memudahkan akses parameter/variabel lokal
RSI	Source Index		Digunakan sebagai penunjuk sumber data dalam operasi string/memori
RDI	Destination Index		Digunakan sebagai penunjuk tujuan data dalam operasi string/memori

Instruksi Sederhana

Instruksi	Deskripsi
MOV	Instruksi untuk memindahkan (menyalin) data dari satu lokasi ke lokasi lain
CMP	Instruksi untuk membandingkan dua operand dengan cara menghitung selisihnya (Operand1 – Operand2), tapi hasilnya tidak disimpan, hanya memengaruhi flag CPU
JMP	Instruksi untuk melompat ke alamat instruksi lain
CALL	Instruksi untuk memanggil subrutin atau fungsi

Instruksi	Deskripsi
MOV RAX, RBX	Menyalin isi register RBX ke register RAX
MOV RAX, 0x42	Menyalin nilai konstanta 0x42 ke register RAX
CMP RAX, RBX	Membandingkan isi RAX dengan RBX (hasil perbandingan disimpan di flag CPU, bukan di register)
JMP loop_start	Lompat ke instruksi di label loop_start (mengatur alur program)
CALL print_data	Memanggil fungsi print_data, alamat instruksi berikutnya disimpan di stack

Instruksi	Deskripsi
ADD	Instruksi untuk menjumlahkan dua nilai
SUB	Instruksi untuk mengurangi operand tujuan dengan operand sumber
INC	Instruksi untuk menambah nilai operand sebesar 1
DEC	Instruksi untuk mengurangi nilai operand sebesar 1
MUL	Instruksi untuk melakukan operasi perkalian
DIV	Instruksi untuk melakukan operasi pembagian

Instruksi	Deskripsi
ADD RAX, RBX	Menambahkan isi RBX ke RAX dan menyimpan hasilnya di RAX
SUB RAX, 0x10	Mengurangi nilai RAX dengan 0x10
INC RDX	Menambahkan nilai RDX sebesar 1
DEC RCX	Mengurangi nilai RCX sebesar 1
MUL 0x50	Mengalikan isi RAX dengan 0x50 dan menyimpan hasilnya pada pasangan register RDX:RAX
DIV 0x75	Membagi isi RDX:RAX dengan 0x75, hasil pembagian disimpan di RAX, sisanya disimpan di RDX

Assembly tidak punya if/for secara langsung, semua dibentuk dari kombinasi CMP + JMP/Jcc (conditional jump)

- JE/JZ → lompat jika sama
- JNE/JNZ → lompat jika tidak sama
- JG/JGE → lompat jika lebih besar/lebih besar sama dengan
- JL/JLE → lompat jika lebih kecil/lebih kecil sama dengan

If Statement (C → Assembly)

```
int a = 5, b = 5;
if (a == b) {
    a = a + 1;
}
```

```
MOV RAX, 5      ; a = 5
MOV RBX, 5      ; b = 5
CMP RAX, RBX    ; bandingkan a dengan b
JE equal_label   ; lompat jika sama
JMP end_if       ; kalau tidak sama, lompat keluar
equal_label:
ADD RAX, 1      ; a = a + 1
end_if:
```

For Loop (C → Assembly)

```
for (int i = 0; i < 3; i++) {
    sum = sum + i;
}
```

```
MOV RCX, 0      ; i = 0
MOV RAX, 0      ; sum = 0
loop_start:
    CMP RCX, 3    ; i < 3 ?
    JGE loop_end   ; keluar jika i >= 3
    ADD RAX, RCX    ; sum = sum + i
    INC RCX        ; i++
    JMP loop_start
loop_end:
```