PERTEMUAN 10

SET INSTRUKSI MODE DAN FORMAT PENGALAMATAN

Karakteristik Instruksi Mesin

- Instruksi mesin (machine intruction) yang dieksekusi membentuk suatu operasi dan berbagai macam fungsi CPU.
- Kumpulan fungsi yang dapat dieksekusi CPU disebut set instruksi (instruction set) CPU.
- Mempelajari karakteristik instruksi mesin, meliputi
  - Elemen – elemen intruksi mesin
  - Representasi instruksinya
  - Jenis – jenis instruksi
  - Penggunaan alamat
  - Rancangan set instruksi

Elemen Instruksi Mesin

Untuk dapat dieksekusi suatu instruksi harus berisi elemen informasi yang diperlukan CPU secara lengkap dan jelas , Apa saja elemennya ?

1. Operation code (Op code)

   Menspesifikasi operasi yang akan dilakukan. Kode operasi berbentuk kode biner

2. Source Operand reference

   Operasi dapat berasal dari lebih satu sumber. Operand adalah input operasi

3. Result Operand reference

   Merupakan hasil atau keluaran operasi

4. Next Instruction Reference

   Elemen ini menginformasikan CPU posisi instruksi berikutnya yang harus diambil dan dieksekusi

Operand dari Operasi

Melihat dari sumbernya, operand suatu operasi dapat berada di salah satu dari ketiga daerah berikut ini :

- Memori utama atau memori virtual
- Register CPU
- Perangkat I/O

Representasi Instruksi

- Instruksi komputer direpresentasikan oleh sekumpulan bit. Instruksi dibagi menjadi beberapa field.
- Field – field ini diisi oleh elemen – elemen instruksi yang membawa informasi bagi operasi CPU.
- Layout instruksi dikenal dengan format instruksi

Format Instruksi #1

Opcode	Alamat

- Kode operasi (opcode) direpresentasi kan dengan singkatan – singkatan, yang disebut mnemonic.
- Mnemonic mengindikasikan suatu operasi bagi CPU.
- Contoh mnemonic adalah :
  - ADD = penambahan
  - SUB = substract (pengurangan)
  - LOAD = muatkan data ke memori

Format Instruksi #2

- Contoh representasi operand secara simbolik :
  - ADD X, Y artinya : tambahkan nilai yang berada pada lokasi Y ke isi register X, dan simpan hasilnya di register X.
- Programer dapat menuliskan program bahasa mesin dalam bentuk simbolik.
- Setiap opcode simbolik memiliki representasi biner yang tetap dan programer dapat menetapkan lokasi masing – masing operand

Jenis – Jenis Instruksi

Contoh suatu ekspresi bilangan :

X = X + Y ;

X dan Y berkorespondensi dengan lokasi 513 dan 514.

Pernyataan dalam bahasa tingkat tinggi tersebut mengintruksikan komputer untuk melakukan langkah berikut ini :

- Muatkan sebuah register dengan isi lokasi memori 513.
- Tambahkan isi lokasi memori 514 ke register.
- Simpan isi register ke lokasi memori 513

Korelasi #1

- Terlihat hubungan antara ekspresi bahasa tingkat tinggi dengan bahasa mesin.
- Dalam bahasa tingkat tinggi, operasi dinyatakan dalam bentuk aljabar singkat menggunakan variabel.
- Dalam bahasa mesin hal tersebut diekpresikan dalam operasi perpindahan antar register

Korelasi #2

- Dapat ditarik kesimpulan bahwa instruksi – instruksi mesin harus mampu mengolah data sebagai implementasi keinginan – keinginan kita.
- Terdapat kumpulan unik set instruksi, yang dapat digolongkan dalam jenis–jenisnya, yaitu
  - Pengolahan data (data processing),
    meliputi operasi – operasi aritmetika dan logika. Operasi aritmetika memiliki kemampuan komputasi untuk pengolahan data numerik. Sedangkan instruksi logika beroperasi terhadap bit – bit word sebagai bit, bukannya sebagai bilangan, sehingga instruksi ini memiliki kemampuan untuk pengolahan data lain
  - Perpindahan data (data movement),
    berisi instruksi perpindahan data antar register maupun modul I/O. Untuk dapat diolah oleh CPU maka diperlukan instruksi - instruksi yang bertugas memindahkan data operand yang diperlukan

Korelasi #3

- Penyimpanan data (data storage),

  berisi instuksi – instruksi penyimpanan ke memori. Instruksi penyimpanan sangat penting dalam operasi komputasi, karena data tersebut akan digunakan untuk operasi berikutnya, minimal untuk ditampilkan pada layar harus diadakan penyimpanan walaupun sementara
- Kontrol aliran program (program flow control),

  berisi instruksi pengontrolan operasi dan pencabangan. Instruksi ini berguna untuk pengontrolan status dan mengoperasikan pencabangan ke set instruksi lain

Jumlah Alamat #1

- Jumlah register atau alamat yang digunakan dalam operasi CPU tergantung format operasi masing – masing CPU.
- Ada format operasi yang menggunakan 3, 2, 1 dan 0 register.
- Umumnya yang digunakan adalah 2 register dalam suatu operasi. Desain CPU saat ini telah menggunakan 3 alamat dalam suatu operasi, terutama dalam MIPS (million instruction per secon).

Jumlah Alamat #2

- Alamat per instruksi yang lebih sedikit akan membuat instruksi lebih sederhana dan pendek, tetapi lebih sulit mengimplementasikan fungsi-fungsi yang kita inginkan.
- Karena instruksi CPU sederhana maka rancangan CPU juga lebih sederhana.
- Jumlah bit dan referensi per instruksi lebih sedikit sehingga fetch dan eksekusi lebih cepat.
- Jumlah instruksi per program biasanya jauh lebih banyak
- Pada jumlah alamat per instruksi banyak, jumlah bit dan referensi instruksi lebih banyak sehingga waktu eksekusi lebih lama.
- Diperlukan register CPU yang banyak, namun operasi antar register lebih cepat.

Jumlah Alamat #3

- Lebih mudah mengimplementasikan fungsi – fungsi yang kita inginkan.
- Jumlah instruksi per program jauh lebih sedikit.
- Untuk lebih jelas perhatikan contoh instruksi – instruksi dengan jumlah register berbeda untuk menyelesaikan persoalan yang sama
- Contoh penggunaan set instruksi dengan alamat 1, 2, dan 3 untuk menyelesaikan operasi hitungan

  Y = (A – B) ÷ (C + D* E)

Contoh instruksi 2 dan 3 alamat

Instruksi 3 alamat :

Instruksi		Komentar
SUB	Y, A, B	Y =A – B
MPY	T, D, E	T= D x E
ADD	T, T, C	T=T + C
DIV	Y, Y, T	Y= Y ÷ T

Instruksi 2 alamat :

Instruksi		Komentar
MOVE	Y, A	Y = A
SUB	Y, B	Y = Y – B
MOVE	T, D	T =D
MPY	T, E	T = T E
ADD	T, C	T=T + C
DIV	Y, T	Y = Y ÷ T

Instruksi 1 alamat

Instruksi		Komentar
LOAD	D	AC =D
MPY	E	AC = AC E
ADD	C	AC= AC + C
STOR	Y	Y = AC
LOAD	A	AC = A
SUB	B	AC = AC – B
DIV	Y	AC =AC ÷ Y
STOR	Y	Y= AC

Instruksi	Keterangan	isi stack
PUSH	B	B
PUSH	A	B,A
SUB	A-B	(A-B)
PUSH	E	(A-B),E
PUSH	D	(A-B),E,D
MUL	D*E	(A-B),(D*E)
PUSH	C	(A-B),(D*E),C
ADD	C+(D*E)	(A-B),(C+D*E)
DIV	(A-B)/(C+(D*E))	(A-B)/(C+(D*E))

Spesifikasi instruksi 3 alamat :

- Simbolik : a = b + c.
- Format alamat : hasil, operand 1, operand 2
- Digunakan dalam arsitektur MIPS.
- Memerlukan word panjang dalam suatu instruksi.

Spesifikasi instruksi 2 alamat :

- Simbolik : a = a + b.
- Satu alamat diisi operand terlebih dahulu kemudian digunakan untuk menyimpan hasilnya.
- Tidak memerlukan instruksi yang panjang.
- Jumlah instruksi per program akan lebih banyak dari pada 3 alamat.
- Diperlukan penyimpanan sementara untuk menyimpan hasil.

Spesifikasi instruksi 1 alamat :

- Memerlukan alamat implisit untuk operasi.
- Menggunakan register akumulator (AC) dan digunakan pada mesin lama.

Spesifikasi instruksi 0 alamat :

- Seluruh alamat yang digunakan implisit.
- Digunakan pada organisasi memori, terutama operasi stack

Rancangan Set Instruksi

- Aspek paling menarik dalam arsitektur komputer adalah perancangan set instruksi, karena rancangan ini berpengaruh banyak pada aspek lainnya.
- Set instruksi menentukan banyak fungsi yang harus dilakukan CPU.
- Set instruksi merupakan alat bagi para pemrogram untuk mengontrol kerja CPU.
- Pertimbangan : Kebutuhan pemrogram menjadi bahan pertimbangan dalam merancang set instruksi

Masalah rancangan yang fundamental meliputi :

Masalah rancangan yang fundamental meliputi :

- Operation repertoire :
  - Berapa banyak dan operasi – operasi apa yang harus tersedia
  - Sekompleks apakah operasi itu seharusnya
- Data types :
  - Jenis data
  - Format data
- Instruction format
  - Panjang instruksi,
  - Jumlah alamat,
  - Ukuran field
- Registers
  - Jumlah register CPU yang dapat direferensikan oleh instruksi, dan fungsinya
- Addressing
  - mode untuk menspesifikasi alamat suatu operand

Tipe Operasi

- Dalam perancangan arsitektur komputer, jumlah kode operasi akan sangat berbeda untuk masing – masing komputer, tetapi terdapat kemiripan dalam jenis operasinya

Jenis Operasi Komputer

- Transfer data. – Konversi
- Aritmetika. - Input/Output
- Logika. - Kontrol sistem dan transfer kontrol

Operasi set instruksi secara umum

Jenis	Nama Operasi	Keterangan
Pemindahan Data	Move	Memindahkan word atau blok dari sumber ke tujuan.
	Store	Memindahkan word dari prosesor ke memori.
	Load	Memindahkan word dari memori ke prosesor.
	Exchange	Menukar isi sumber dengan tujuan.
	Clear (reset)	Memindahkan word 0 ke tujuan.
	Set	Memindahkan word 1 ke tujuan.
	Push	Memindahkan word dari sumber ke bagian paling atas stack.
	Pop	Memindahkan word dari stack teratas ke tujuan.

• Operasi set instruksi secara umum

Logika	AND	Melakukan operasi logika tertentu terhadap bit.
	OR	
	NOT	
	Exclusive-OR	
	Test	Menguji kondisi tertentu.
	Compare	Membandingkan dua operand (secara logika maupun aritmetika).
	Set Variabel Kontrol	Instruksi untuk menyetel kontrol bagi keperluan proteksi, interrupt, kontrol timer.
	Shift	Melakukan penggeseran bit – bit operand.
	Rotate	Melakukan pemutaran bit – bit operand.

• Operasi set instruksi secara umum

Pemindahan Kontrol	Jump (cabang)	Perpindahan tidak bersyarat; memuatkan PC dengan alamat tertentu.	Pemindahan Kontrol	Execute	Mengambil operand dari lokasi tertentu dan mengeksekusinya sebagai instruksi.
	Jump bersyarat	Menguji persyaratan tertentu; melakukan aktivitas tergantung persyaratannya.		Skip	Menambah PC sehingga melompati instruksi berikutnya.
	Jump ke subrutin	Menempatkan informasi data kontrol program saat itu di lokasi yang ditentukan; melompat ke alamat tertentu.		Skip bersyarat	Melompat berdasarkan syarat tertentu.
	Return	Menggantiini PC dan register lainnya yang berasal dari lokasi tertentu.		Halt	Menghentikan eksekusi program.
				Wait (hold)	Menghentikan eksekusi program dan menguji persyaratan.
				No operation	Tidak ada operasi yang dilakukan, namun eksekusi program dilanjutkan.

Operasi set instruksi secara umum

Input/Output	Input (read)	Memindahkan data dari port I/O ke tujuan.
	Output (write)	Memindahkan data dari prosesor ke port atau modul I/O.
	Start I/O	Memindahkan instruksi ke prosesor I/O untuk memulai proses I/O.
	Test I/O	Memindahkan informasi status dari sistem I/O ke tujuan.

- **Operasi set instruksi secara umum**

Konversi	Translate	Menerjemahkan nilai – nilai dalam suatu bagian memori berdasarkan tabel korespondensi.
	Convert	Mengkonversi isi word ke bentuk lain.

Transfer Data

Instruksi tranfer data harus menetapkan :

- Lokasi operand sumber
- Lokasi operand tujuan
- Panjang data yang akan dipindahkan
- Mode pengalamatannya

Apabila sebuah atau kedua operand berada di dalam memori, maka CPU harus melakukan sebagian atau seluruh tindakan berikut :

1. Menghitung alamat memori, yang didasarkan pada mode alamatnya.
2. Apabila alamat mengacu pada virtual memori harus dicari alamat memori sebenarnya.
3. Menentukan apakah alamat berada dalam cache memori.
4. Bila di cache tidak ada, dikeluarkan perintah ke modul memori

Mode Pengalamatan

Mengatasi keterbatasan format instruksi

- Dapat mereferensi lokasi memori yang besar
- Mode pengalamatan yang mampu menangani keterbatasan tersebut

- Masing – masing prosesor menggunakan mode pengalamatan yang berbeda – beda.
- Memiliki pertimbangan dalam penggunaannya.
- Ada beberapa teknik pengalamatan
    a. Immediate Addressing
    b. Direct Addressing
    c. Indirect Addressing
    d. Register Addressing
    e. Register Indirect Addressing
    f. Displacement Addressing
    g. Stack Addressing

Immediate Addressing (1)

Bentuk pengalamatan ini yang paling sederhana ?

- Operand benar – benar ada dalam instruksi atau bagian dari instruksi = Operand sama dengan field alamat.
- Umumnya bilangan akan disimpan dalam bentuk komplemen dua.
- Bit paling kiri sebagai bit tanda.
- Ketika operand dimuatkan ke dalam register data, bit tanda akan digeser ke kiri hingga maksimum word data
- Contoh :

ADD 5 ; tambahkan 5 pada akumulator

Immediate Addressing (+)&(-)

Keuntungan

- Mode ini adalah tidak adanya referensi memori selain dari instruksi yang diperlukan untuk memperoleh operand.
- Menghemat siklus instruksi sehingga proses keseluruhan akan cepat.

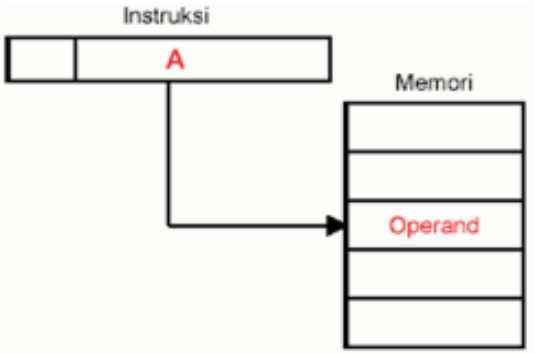
Kerugiannya

- Ukuran bilangan dibatasi oleh ukuran field alamat

Direct Addressing (2)

Pengalamatan langsung

- Kelebihan :
  - Field alamat berisi efektif address sebuah operand.
- Teknik ini banyak digunakan pada komputer lama dan komputer kecil.
- Hanya memerlukan sebuah referensi memori dan tidak memerlukan kalkulasi khusus.
- Kelemahan :
  - Keterbatasan field alamat karena panjang field alamat biasanya lebih kecil dibandingkan panjang word
- Contoh :
  - ADD A ; tambahkan isi pada lokasi alamat A ke akumulator



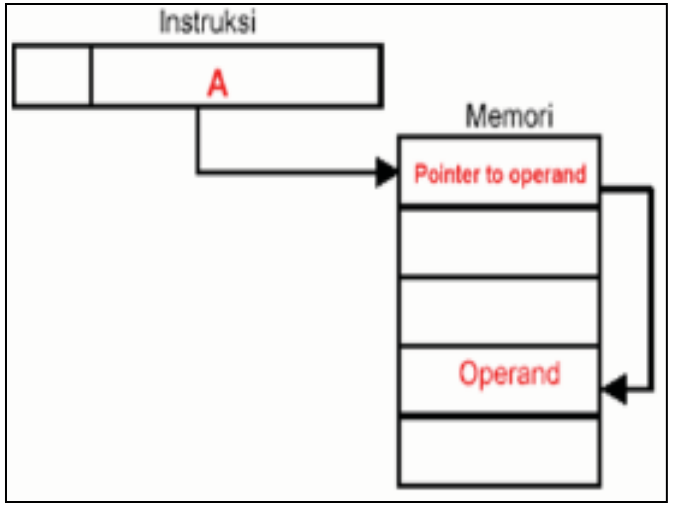
Indirect Addressing (3)

- Mode pengalamatan tak langsung
  - Field alamat mengacu pada alamat word di dalam memori, yang pada gilirannya akan berisi alamat operand yang panjang
  - Contoh :

    ADD (A) ; tambahkan isi memori

    yang ditunjuk oleh

    isi alamat A ke akumulator



Indirect Addressing (+)&(-)

Keuntungan

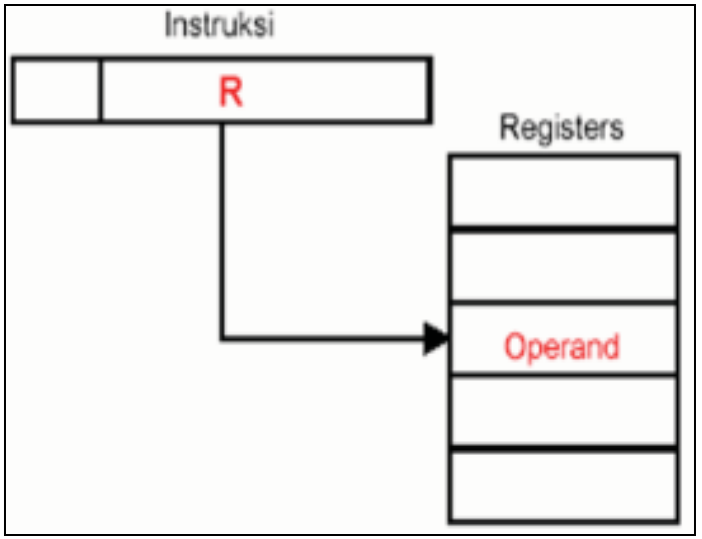
- Ruang bagi alamat menjadi besar sehingga semakin banyak alamat yang dapat referensi.

Kerugian

- Diperlukan referensi memori ganda dalam satu fetch sehingga memperlambat proses operasi

Register Addressing (4)

- Metode pengalamatan register mirip dengan mode pengalamatan langsung.
- Perbedaannya terletak pada field alamat yang mengacu pada register, bukan pada memori utama.
- Field yang mereferensi register memiliki panjang 3 atau 4 bit, sehingga dapat mereferensi 8 atau 16 register general purpose.



Register Addressing (+)&(-)

Keuntungan pengalamatan register

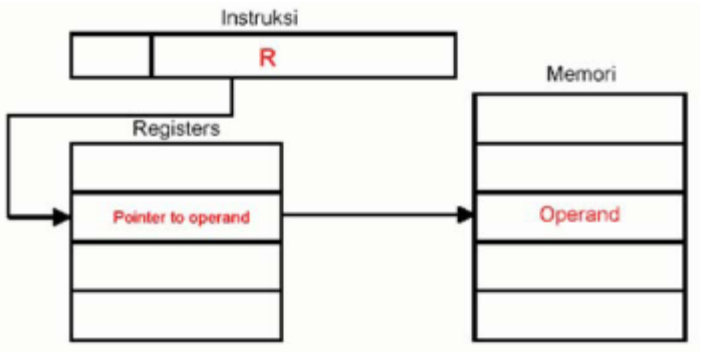
- Diperlukan field alamat berukuran kecil dalam instruksi dan tidak diperlukan referensi memori.
- Akses ke register lebih cepat daripada akses ke memori, sehingga proses eksekusi akan lebih cepat.

Kerugian

- Ruang alamat menjadi terbatas

Register Indirect Addressing (5)

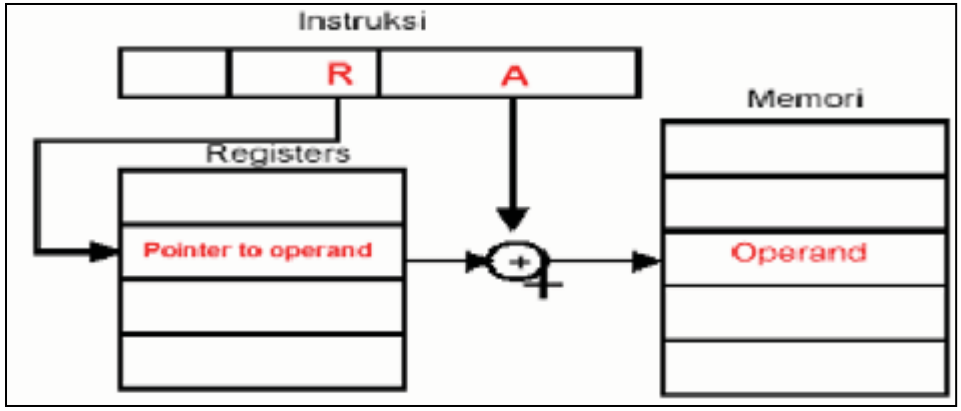
- Metode pengalamatan register tidak langsung mirip dengan mode pengalamatan tidak langsung.
- Perbedaannya adalah field alamat mengacu pada alamat register.
- Letak operand berada pada memori yang ditunjuk oleh isi register.
- Keuntungan dan keterbatasan pengalamatan register tidak langsung pada dasarnya sama dengan pengalamatan tidak langsung.
  - Keterbatasan field alamat diatasi dengan pengaksesan memori yang tidak langsung sehingga alamat yang dapat direferensi makin banyak.
  - Dalam satu siklus pengambilan dan penyimpanan, mode pengalamatan register tidak langsung hanya menggunakan satu referensi memori
  - utama sehingga lebih cepat daripada mode pengalamatan tidak langsung



Displacement Addressing (6)

- Menggabungkan kemampuan pengalamatan langsung dan pengalamatan register tidak langsung.
- Mode ini mensyaratkan instruksi memiliki dua buah field alamat, sedikitnya sebuah field yang eksplisit.
  - Field eksplisit bernilai A dan field implisit mengarah pada register

Displacement Addressing (6)



- Operand berada pada alamat A ditambah isi register.
- Tiga model displacement
  - Relative Addressing
  - Base Register Addressing
  - Indexing

Displacement Addressing (6)

- Relative addressing, register yang direferensi secara implisit adalah program counter (PC).
  - Alamat efektif didapatkan dari alamat instruksi saat itu ditambahkan ke field alamat.
  - Memanfaatkan konsep lokalitas memori untuk menyediakan operand-operand berikutnya.
- Base register addressing, register yang direferensikan berisi sebuah alamat memori, dan field alamat berisi perpindahan dari alamat itu.
  - Referensi register dapat eksplisit maupun implisit.
  - Memanfaatkan konsep lokalitas memori.
- Indexing adalah field alamat mereferensi alamat memori utama, dan register yang direferensikan berisi pemindahan positif dari alamat tersebut.
  - Merupakan kebalikan dari model base register.
  - Field alamat dianggap sebagai alamat memori dalam indexing.
  - Manfaat penting dari indexing adalah untuk eksekusi program-program iteratif

Stack Addressing (7)

- Stack adalah array lokasi yang linier = pushdown list = last-in-first-out-queue.
- Stack merupakan blok lokasi yang terbalik.
  - Butir ditambahkan ke puncak stack sehingga setiap saat blok akan terisi secara parsial.
- Yang berkaitan dengan stack adalah pointer yang nilainya merupakan alamat bagian paling atas stack.
- Dua elemen teratas stack dapat berada di dalam register CPU, yang dalam hal ini stack ponter mereferensi ke elemen ketiga stack.
- Stack pointer tetap berada di dalam register.
- Dengan demikian, referensi – referensi ke lokasi stack di dalam memori pada dasarnya merupakan pengalamatan register tidak langsung

Perbandingan Mode pengalamatan

Mode	Algoritma	Keuntungan Utama	Kerugian utama
Immediate	Operand = A	Tidak ada referensi memori	Besaran operand terbatas
Direct	EA = A	Sederhana	Ruang alamat terbatas
Indirect	EA = (A)	Ruang alamat besar	Referensi memori berganda
Register	EA = R	Tidak ada referensi memori	Ruang alamat terbatas
Register Indirect	EA = (R)	Ruang alamat besar	Referensi memori ekstra
Displacement	EA = A+ (R)	Fleksibelitas	Kompleksitas
Stack	EA = Puncak Stack	Tidak ada referensi	Aplikasi memori terbatas

Keterangan Perbandingan Mode Pengalamatan

Keterangan :

- A = isi suatu field alamat dalam instruksi
- EA = alamat aktual (efektif) sebuah lokasi yang berisi operasi yang di referensikan
- (X) = isi lokasi X

Mode Pengalamatan Pentium

- Pentium dilengkapi bermacam – macam mode pengalamatan untuk memudahkan bahasa – bahasa tingkat tinggi mengeksekusinya secara efisien (C/Fortran)

Mode pengalamatan pentium

Mode	Algoritma
Immediate	Operand = A
Register	eA = R
Displacement	eA = (SR) + A
Base	eA = (SR) + (B)
Base with displacement	eA = (SR) + (B) + A
Scaled index with displacement	eA = (SR) + (B) + (I) + A
Base with scaled index and displacement	eA = (SR) + (I) x S + (B) + A
Relative	eA = (PC) + A

Keterangan :

- SR = register segment
- PC = program counter
- A = isi field alamat
- B = register basis
- I = register indeks
- S = faktor skala

Mode pengalamatan pentium

- Mode immediate
  - Operand berada di dalam instruksi.
  - Operand dapat berupa data byte, word maupun doubleword
- Mode operand register, operand adalah isi register.
  - Beberapa macam jenis register
    - register 8 bit (AH, BH, CH, DH, AL, BL, CL, DL)
    - register 16 bit (AX, BX, CX, DX, SI, DI, SP, BP)
    - register 32 bit (EAX, EBX, ECX, EDX, ESI, EDI, ESP, EBP)
    - register 64 bit yang dibentuk dari register 32 bit secara berpasangan.
    - register 8, 16 dan 32 merupakan register untuk penggunaan umum
    - (general purpose register).
    - register 64 bit biasanya untuk operasi floating point.
    - register segmen (CS, DS, ES, SS, FS, GS)

Mode Displacement

- Mode displacement
  - ❑ Alamat efektif berisi bagian – bagian instruksi dengan displacement 8, 16, atau 32 bit.
  - ❑ Dengan segmentasi, seluruh alamat dalam instruksi mengacu ke sebuah offset di dalam segmen.
  - ❑ Dalam Pentium, mode ini digunakan untuk mereferensi variabel – variabel global

2. Format-Format Instruksi #1

- Format instruksi menentukan layout bit suatu instruksi.
- Format instruksi harus mencakup opcode dan secara implisit atau eksplisit, nol operand atau lebih

Format-Format Instruksi #2

- Seluruh operand eksplisit direferensikan dengan menggunakan salah satu mode pengalamatan yang ada
- Secara implisit atau eksplisit format harus dapat mengindikasikan mode pemngalamata seluruh operandnya.
- Pada sebagian besar set instruksi digunakan lebih dari satu format instruksi.

A. Panjang Instruksi #1

- Umumnya pemograman menginginkan opcode, operand, dan mode pengalamatan yang lebih banyak serta range alamat yang lebih besar
- Dengan adanya opcode dan operand yang lebih banyak akan memudahkan pekerjaan pemograman

Panjang Instruksi #2

- Mode pengalamatan yang lebih banyak akan memberikan fleksibilitas yang lebih besar terhadap pemogram dalam mengimplementasikan fungsi-fungsi tertentu, seperti manipulasi table dan pencabangan yang berjumlah banyak.
- Dengan bertambahnya ukuran memori utama dan semakin banyaknya pemakaian memori virtual, pemogram akan dapat mengalamati jangkauan memori yang lebih besar.
- Kecepatan perpindahan memori tidak dapat diatasi dengan penambahan kecepatan processor

Panjang Instruksi #3

- Karena memori akan dapat menjadi sebuah bottleneck apabila prosessor dapat mengeksekusi instruksi lebih cepat dari pada kecepatan untuk mengambil instruksi itu.
- Salah satu cara mengatasi masalah ini adalah dengan menggunakan cache memori atau dengan menggunakan instruksi-instruksi yang lebih pendek
- Instruksi 16 bit akan dapat diambil dua kali lebih cepat di bandingkan instruksi 32 bit namun mungkin akan dieksekusi dua kali lebih lambat

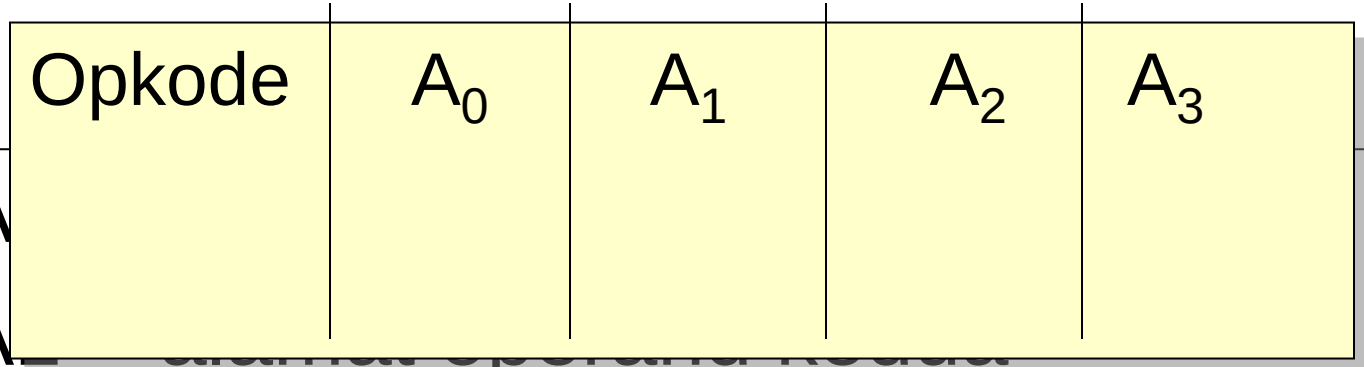
B. Format Instruksi

- Organisasi internal pada komputer dinyatakan oleh instruksi-instruksi yang dapat dijalankannya.
- Suatu instruksi merupakan suatu tata cara yang digunakan oleh komputer untuk menyatakan operasi-operasi seperti ADD, STORE, LOAD, MOVE dan BRANCH beserta untuk menentukan lokasi data di mana suatu operasi akan dikerjakan.

C. Format Alamat

- Pengkodean biner pada setiap komputer memiliki format kode instruksi tersendiri.
- Pada komputer terdahulu, setiap instruksi terdiri atas sebuah upcode dan empat field alamat.

Opkode



- A
- A
- A2 = alamat di mana hasil operand disimpan
- A3 = Alamat dari instruksi berikutnya

D. Mode Pengalamatan #1

- Suatu mode pengalamatan dapat digunakan untuk menentukan suatu alamat tempat untuk dimana operand akan di fetch
- Beberapa teknik semacam ini dapat meningkatkan kecepatan pelaksanaan instruksi dengan menurunkan jumlah referensi pada memori utama dan meningkatkan jumlah referensi pada register kecepatan tinggi

Mode Pengalamatan #2

- Mode pengalamatan ini menjabarkan suatu aturan untuk menginterpresetasikan atau memodifikasi field alamat dari instruksi sebelum operand di referensikan
- Pada semua mode pengalamatan lainnya, operand yang sesungguhnya tidak disimpan pada field alamat tetapi beberapa nilai di jabarkan dan di gunakan untuk menentukan operasi operand.

E. Kode Instruksi (KI) #1

- Selain dari representasi data, kode biner juga digunakan untuk membuat instruksi kontrol dalam komputer, yang disebut kode instruksi.
- Kode instruksi merupakan kelompok bit yang memberitahukan kepada komputer untuk menunjukan suatu operasi tertentu.

Kode Instruksi (KI) #2

- Kode Instruksi dibagi dalam bagian-bagian, yang masing-masing bagian mempunyai interpretasi sendiri
- Bagian yang paling pokok adalah kode operasi (Operation Code / Opcode)
- Opcode adalah sekelompok bit yang menunjukan operasi seperti ADD, SUBTRACT, SHIFT, dan COMPLEMENT
- Bagian lain dari instruksi mencakup satu operasi (operand) atau lebih

Kode Instruksi (KI) #3

- Operand adalah suatu nama yang digunakan untuk obyek instruksi dan mungkin data atau alamat yang mengatakan dimana data tersebut
- Untuk membuat kode instruksi dalam komputer harus kode biner. (seperti operasi LOAD dan Store)
- Load adalah meng-copy bilangan dari lokasi memori kedalam register
- Strore adalah meng-copy bilangan dari register kedalam lokasi memori

Selesai