


# Mode Pengalaman

**D4 Teknik Komputer  
Politeknik Elektronika Negeri Surabaya**

# Pokok Bahasan (SAP)

- ▶ Pengantar Desain dan Organisasi Komputer
  - ▶ Evolusi dan Kinerja Komputer
  - ▶ Komponen dan Struktur CPU
  - ▶ Memori Komputer
  - ▶ Peralatan Penyimpanan Data
  - ▶ Unit Masukan dan Keluaran
  - ▶ Sistem Bus
  - ▶ Set Instruksi
  - ▶ Mode Pengalamatan
  - ▶ Unit Kontrol
  - ▶ Pipelining dan Branch Prediction
- 

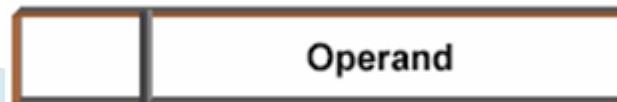
# Mode Pengalamatan

- ▶ 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
  - Immediate Addressing
  - Direct Addressing
  - Indirect Addressing
  - Register Addressing
  - Register Indirect Addressing
  - Displacement Addressing
  - 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 komplement 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

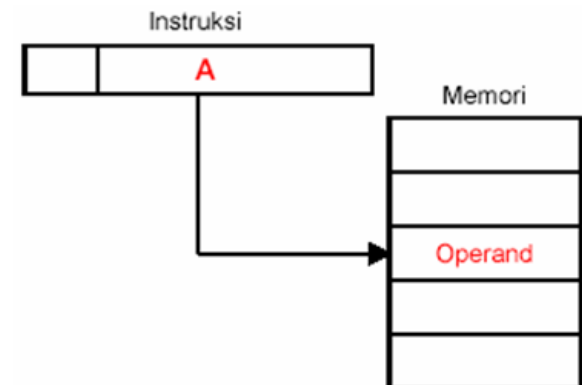
- Ukuran bilangan dibatasi oleh ukuran field alamat

# *Direct Addressing (2)*

## ► Pengalamatan langsung

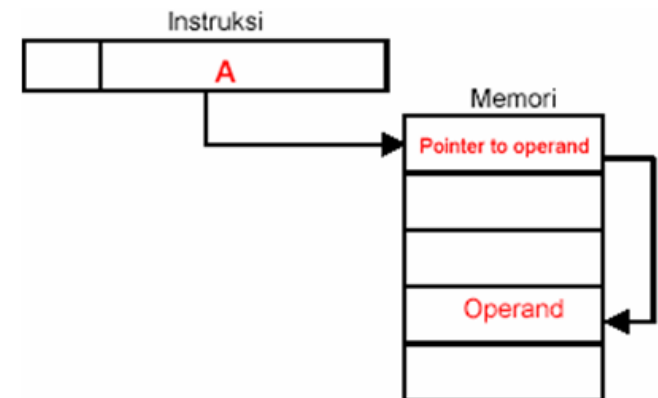
- Kelebihan :
  - Field alamat berisi efektif address sebuah operand.
- Teknik ini banyak digunakan pada komputer lama dan komputer kecil.
- Hanya memerlukan sebuah referensi memori dan tidak memerlukan kalkulasi khusus.
- Kelemahan :
  - Keterbatasan field alamat karena panjang field alamat biasanya 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



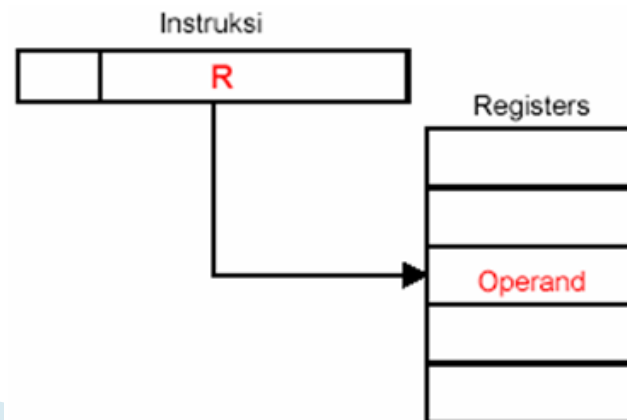
# *Indirect Addressing (+)&(-)*

- ▶ 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.

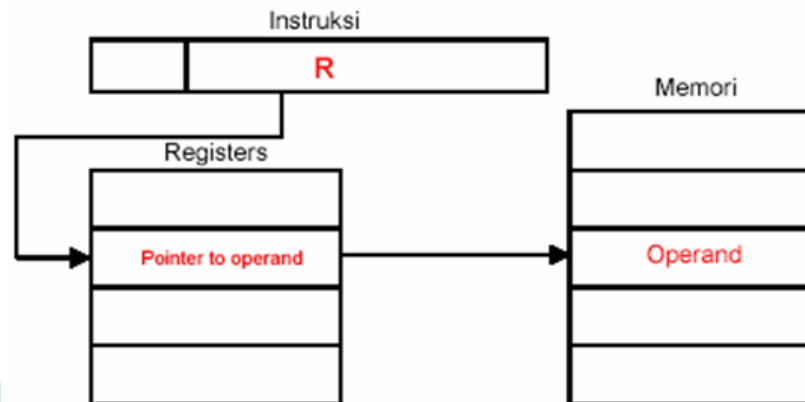


# *Register Addressing (+)&(-)*

- ▶ 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)

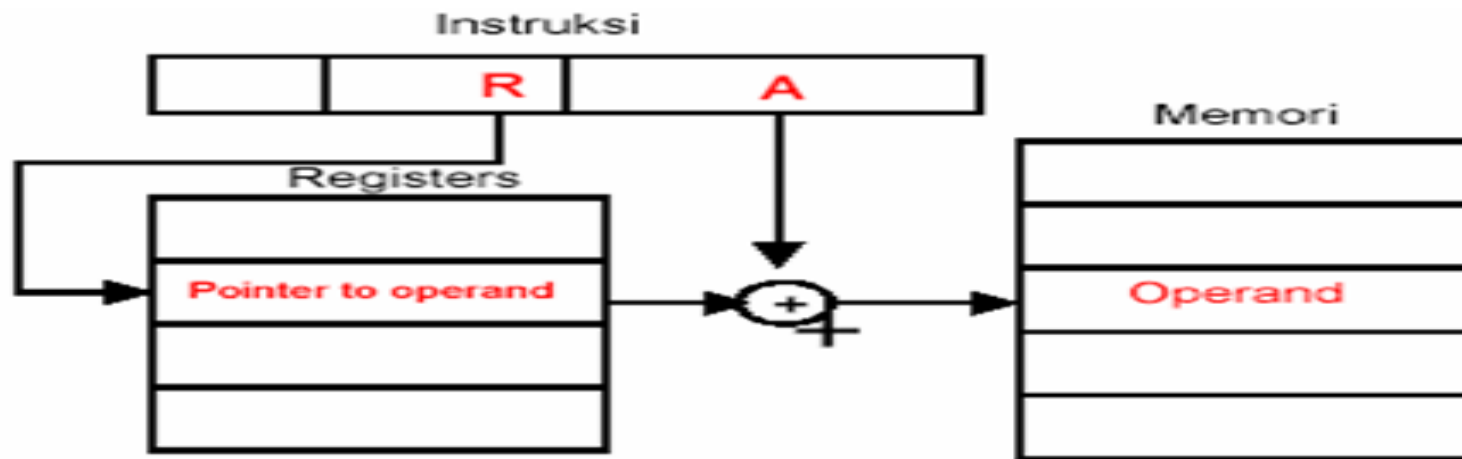
- ▶ Metode pengalamatan register tidak langsung mirip dengan mode pengalamatan tidak langsung.
- ▶ Perbedaannya adalah field alamat mengacu pada alamat register. Letak operand berada pada memori yang ditunjuk oleh isi register.
- ▶ Keuntungan dan keterbatasan pengalamatan register tidak langsung pada 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, setidaknya 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*

- ▶ 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 pointer 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

# Mode pengalamatan (tabel)

Mode	Algoritma	Keuntungan	Kerugian
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 ganda
Register	$eA = R$	Tidak ada referensi memori	Ruang alamat terbatas
Register Indirect	$eA = (R)$	Ruang alamat besar	Referensi memori ekstra
Displacement	$eA = A + (R)$	Fleksibel	Kompleks
Stack	$eA = \text{top of stack}$	Tidak ada referensi	Aplikasi memori terbatas



# Mode Pengalamatan Pentium

- ▶ Pentium dilengkapi bermacam – macam mode pengalamatan untuk memudahkan bahasa – bahasa tingkat tinggi mengeksekusinya secara efisien.

# 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	$eA = (SR) + (I) \times S + (B) + A$
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 pengalamatan pentium

## ▶ 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

# Mode pengalamatan pentium

## ► Mode base

- Pengalamatan indirect yang menspesifikasi satu register 8, 16 atau 32 bit berisi alamat efektifnya.
- Base with displacement mode
  - Instruksi mempunyai displacement yang akan ditambahkan ke register basis.
  - Umumnya termasuk general purpose register.
  - Contoh penggunaan mode ini adalah digunakan kompiler untuk menunjuk awal daerah variabel, untuk mengindeks suatu larik, dan digunakan untuk mengakses field sebuah record

# Mode pengalamatan pentium

- ▶ Mode scaled index with displacement
  - Instruksi mengandung displacement yang akan ditambahkan ke register indeks.
  - Register indeks dapat berupa sembarang register kecuali ES yang umumnya untuk pengolahan stack.
  - Dalam perhitungan alamat efektif, isi register indeks dikalikan dengan 1, 2, 4, atau 8 dan kemudian ditambahkan ke displacement.
  - Mode ini sangat cocok untuk pengindekan larik.
  - Faktor skala 2 digunakan untuk larik integer 16 bit, skala 4 untuk larik integer 32 bit dan faktor skala 8 untuk bilangan floating point

# Mode pengalamatan pentium

- ▶ Base with index and displacement mode
  - menjumlahkan isi register basis, register indeks, dan displacement untuk mendapatkan alamat efektifnya.
  - Register basis dan register indeks dapat berupa sembarang register, kecuali ESP.
  - Contoh :
    - Untuk mengakses larik lokal pada stack frame.
    - Mode ini juga dapat digunakan untuk mendukung larik dua dimensi, displacement menunjuk awal larik dan setiap register menangani satu dimensi larik

# Mode pengalamatan pentium

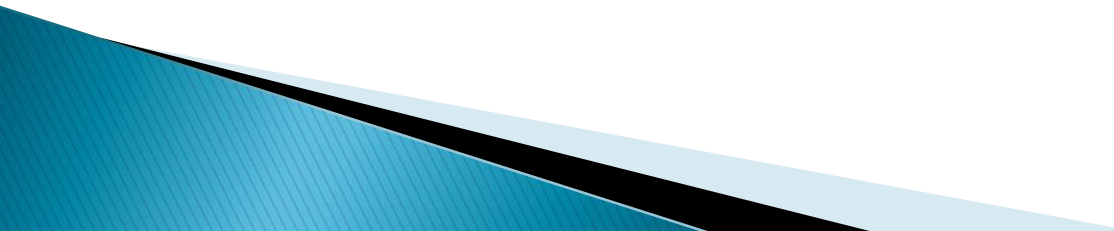
- ▶ Base scaled index with displacement mode
  - Alamat efektif diperoleh dari penjumlahan isi register indeks yang dikalikan dengan faktor skala
  - Isi register basis, dan displacement.
  - Mode ini sangat berguna untuk pengaksesan larik pada stack frame



# Mode pengalamatan pentium

- ▶ Mode relative addressing
  - Digunakan dalam instruksi – instruksi tranfer kontrol.
  - Displacement ditambahkan ke program counter (PC), yang menunjuk ke instruksi berikutnya

# Format Instruksi

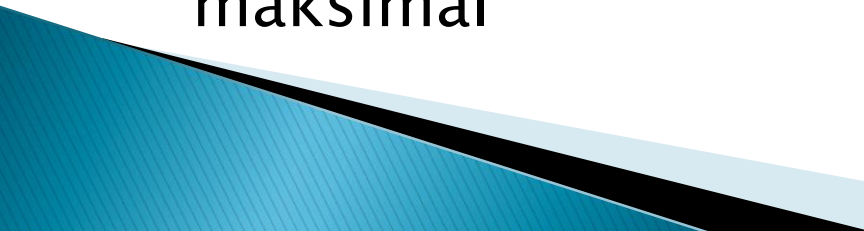
- ▶ Format instruksi menentukan susunan dan tata letak bit suatu instruksi.
  - ▶ Format intruksi harus mencakup opcode serta implisit dan eksplisit operand.
  - ▶ Biasanya set instruksi memiliki lebih dari satu format instruksi.
  - ▶ Inti dari format instruksi adalah menentukan panjang instruksi dan alokasi bit dalam instruksi tersebut
- 

# Panjang Instruksi

- ▶ Penentuan panjang intruksi mempengaruhi dan dipengaruhi oleh
  - Ukuran memori
  - Organisasi memori
  - Struktur bus
  - Kompleksitas CPU
  - Kecepatan CPU
- Bahasan RISC –

# Penentuan panjang instruksi ?

Pertimbangan : (INSTRUKSI)

- ▶ Instruksi yang kompleks mempengaruhi perancangan perangkat keras prosesor, karena fungsi – fungsi yang disajikan CPU harus diimplementasikan dalam perangkat keras.
  - ▶ Semakin kompleks perangkat keras, tentunya akan meningkatkan faktor biaya walau belum tentu meningkatkan kinerja komputer secara keseluruhan.
  - ▶ Penentuan panjang instruksi menjadi sangat essensi untuk mencapai kinerja komputer yang maksimal
- 

# Penentuan panjang instruksi ?

Pertimbangan : (PROGRAMMER)

- ▶ Menginginkan opcode, operand, dan mode pengalamatan yang lebih banyak serta range alamat yang lebih besar karena semua itu akan mempermudah pemrogram mengimplementasikan keinginannya dalam program.
- ▶ Pertimbangannya bahwa opcode, operand dan mode pengalamatan yang lebih banyak akan membutuhkan ruang yang lebih besar.
- ▶ Instruksi 32 bit akan menempati ruang dua kali lebih banyak daripada instruksi 16 bit, namun kegunaannya mungkin tidak akan dua kali lebih banyak

# Penentuan panjang instruksi ?

Pertimbangan lain :

- ▶ Panjang instruksi harus sama dengan panjang perpindahan memori (pada sistem bus, panjang bus data) dan panjang instruksi seharusnya merupakan kelipatan panjang instruksi lainnya.
- ▶ Hal ini harus dipertimbangkan untuk mendapatkan optimalisasi proses eksekusi instruksi nantinya, baik kecepatan perpindahan maupun alokasi memorinya.
- ▶ Kecepatan perpindahan data tidak dapat diatasi dengan menambah kecepatan prosesor.
- ▶ Kecepatan prosesor hanya berhubungan dengan eksekusi internalnya, sedangkan kecepatan perpindahan tergantung bus, memori, dan data itu sendiri.
- ▶ Cara meningkatkan kecepatan perpindahan data adalah dengan menggunakan cache memori dan menggunakan instruksi - instruksi yang lebih pendek

# Penentuan panjang instruksi ?

- ▶ Panjang instruksi harus merupakan kelipatan panjang karakter, yang umumnya 8 bit, dan kelipatan panjang bilangan fixed point.
- ▶ Diabaikan ?
  - Terjadi pemborosan bit pada setiap word ketika sejumlah karakter disimpan di dalamnya
- ▶ Keputusan salah yang pernah diambil IBM
  - Mengeluarkan arsitektur prosesor 36 bit, terjadi banyak pemborosan karena ukuran **karakter 8** bit.
  - Arsitektur tersebut diganti dengan arsitektur 32 bit

# Bagaimana dengan *Alokasi Bit*?

- ▶ Inti dalam alokasi bit adalah berada pada untung – rugi antara jumlah opcode dengan kemampuan pengalamatannya.
- ▶ Opcode yang banyak akan menyebabkan bit yang lebih banyak pada field opcode, yang secara otomatis akan mengurangi jumlah bit untuk pengalamatan.
- ▶ Faktor yang merupakan hal – hal yang penting dalam menentukan penggunaan bit – bit pengalamatan :
  - Jumlah mode pengalamatan
  - Jumlah operand
  - Register vs memori
  - Jumlah set register
  - Jangkauan alamat
  - Granularitas alamat



## ▶ Jumlah mode pengalamatan

- Mode pengalamatan dapat dilakukan secara implisit atau eksplisit, yang kesemuanya memerlukan jumlah bit yang berbeda

## ▶ Jumlah operand

- Jumlah operand sangat mempengaruhi kemampuan instruksi.
- Jumlah operand yang sedikit biasanya akan menjadikan instruksi yang panjang dalam suatu fungsi

## ▶ Register vs memori

- Penggunaan register maupun memori membutuhkan jumlah bit yang berbeda.
- Pada pengalamatan implisit dengan register akan dibutuhkan bit lebih kecil dari pada mode pengalamatan langsung ke memori

## ▶ Jumlah set register

- Jumlah set register juga mempengaruhi penggunaan bit – bit instruksi.
- General purpose register yang umumnya dimiliki hampir seluruh arsitektur komputer dapat digunakan untuk register alamat maupun register instruksi

## ▶ Jangkauan alamat

- untuk alamat – alamat yang mereferensi memori secara eksplisit, jangkauan ditentukan oleh jumlah bit yang digunakan untuk pengalamatan.
- Pertimbangan menggunakan mode displacement patut dipertimbangkan untuk memiliki jangkauan pengalamatan yang besar

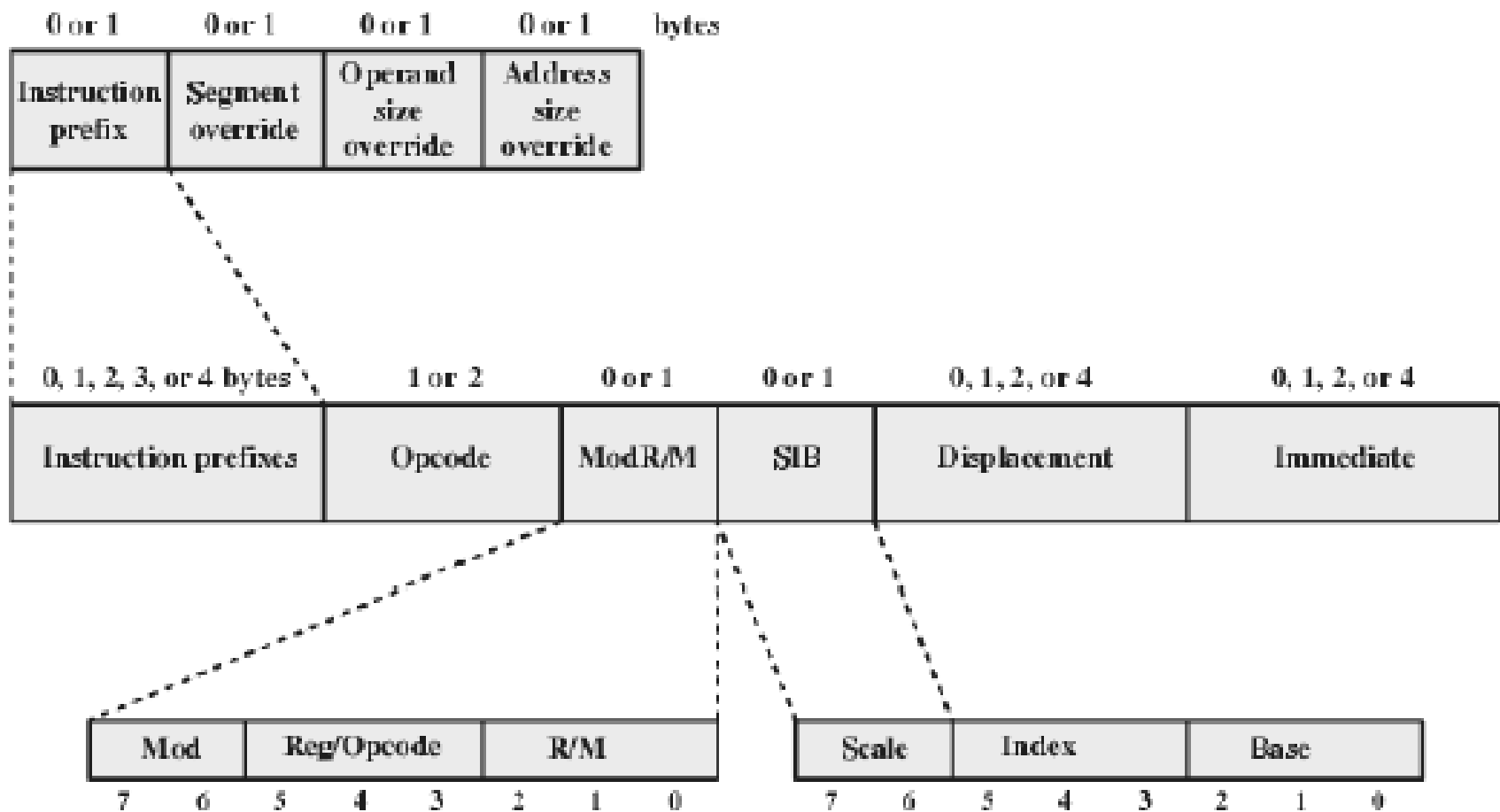
## ▶ Granularitas alamat :

- Pengalamatan yang mereferansi memori dapat digunakan pengalamatan yang mereferensi word atau byte

# Format Instruksi Pentium

- ▶ Arsitektur Pentium dilengkapi bermacam – macam format instruksi.
- ▶ Instruksi – instruksinya dibangun mulai dari nol hingga empat prefiks instruksi opsional, sebuah opcode satu atau dua byte, specifier alamat opsional, yang terdiri dari Mod r/m byte dan scale index byte (SIB), sebuah opsional displacement, dan opsional immediate

# Format Instruksi Pentium



# Bagian prefix byte

## ▶ Instruction Prefixes Biasa

- Berisi : Prefiks Lock dan Prefiks perulangan.
- Prefiks Lock yang digunakan untuk keamanan penggunaan shared memory yang eksklusif dalam lingkungan multiprosesor.
- Prefiks perulangan berguna untuk operasi perulangan yang dapat diproses lebih cepat daripada menggunakan loop perangkat lunak biasa

## ▶ Segment Override

- Menspesifikasi register segmen yang harus dipakai instruksi
- Mengesampingkan (override) pilihan register segmen default yang dihasilkan Pentium untuk instruksi tersebut

# Bagian prefix byte

## ▶ Address Size

- Prosesor dapat mengamati memori dengan menggunakan alamat 16 bit atau 32 bit.
- Ukuran alamat menentukan ukuran displacement dalam instruksi dan ukuran offset alamat yang dihasilkan selama perhitungan alamat efektif berlangsung.
- Prefiks ukuran alamat digunakan untuk mengubah alamat 16 bit ke 32 bit dan sebaliknya

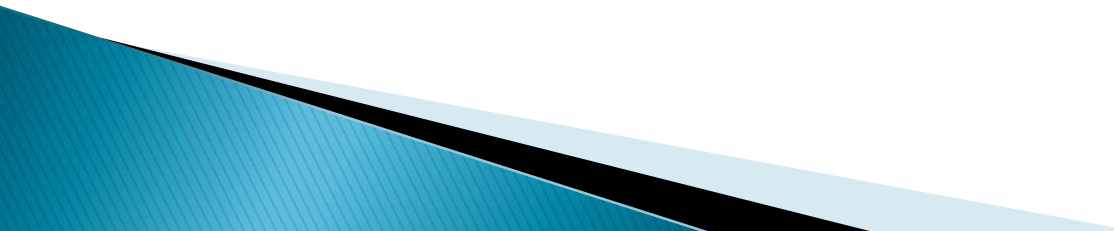
## ▶ Operand Size

- Instruksi memiliki ukuran operand default 16 bit dan 32 bit
- Prefiks operand mengubah operand 16 bit ke 32 bit dan sebaliknya

# Field Instruksi

- ▶ Opcode
  - Opcode dapat mencakup bit – bit yang menspesifikasikan apakah suatu data merupakan byte atau full-size, arah operasi data, dan apakah immediate data field harus merupakan sign-extended
- ▶ Mod r/m
  - Memberikan informasi pengalamatan.
  - Byte Mod r/m menspesifikasikan apakah operand berada di dalam register atau berada di dalam memori. Apabila operand berada di dalam memori, maka field – field yang berada di dalam byte akan menspesifikasi mode pengalamatan yang akan dipakai
- ▶ SIB
  - Berisi skala indeks register dan base register
- ▶ Displacement
  - Bila mode pengalamatan menggunakan mode ini maka akan ditambahkan field displacement integer bertanda 8 bit, 16 bit atau 32 bit
- ▶ Immediate
  - Memberikan nilai operand 8 bit, 16 bit atau 32 bit

# Kesimpulan

- ▶ Instruksi = biner
    - Bagian opcode
    - Bagian alamat
  - ▶ Tipe data dan jenis instruksi digolongkan kebeberapa kelompok
  - ▶ Panjang bit Opcode mempengaruhi jumlah jenis instruksi
  - ▶ Jumlah bit Alamat mempengaruhi jangkauan alamat yang bisa digunakan
  - ▶ Terdapat berbagai macam mode pengalamatan digunakan sesuai dengan kondisi
- 



# Soal – Soal

- ▶ Jelaskan hubungan antara jumlah bit pada opcode dengan jumlah instruksi yang ada !
  - ▶ Jelaskan hubungan antara jumlah bit pada Alamat yang ada di set instruksi dengan jumlah alamat yang bisa di jangkau !
  - ▶ Bagaimana cara agar set instruksi jumlah dapat menambah jangkauan pada memori!
  - ▶ Jelaskan kapan dan pada saat apa mode pengalamatan digunakan
- 