

## 1. Tabel pemetaan memori pada PC

### a. Pemetaan Langsung (Direct Mapping)

Pemetaan langsung adalah teknik yang paling sederhana, yaitu teknik ini memetakan blok memori utama hanya ke sebuah saluran cache saja. Jika suatu blok ada di cache, maka tempatnya sudah tertentu. Keuntungan dari direct mapping adalah sederhana dan murah. Sedangkan kerugian dari direct mapping adalah suatu blok memiliki lokasi yang tetap (jika program mengakses 2 blok yang di map ke line yang sama secara berulang-ulang, maka cache-miss sangat tinggi).

Berikut penjelasan lebih detail :

§ Setiap blok pada main memory dipetakan dengan line tertentu pada *cache*.  $i = j \text{ modulo } C$  di mana  $i$  adalah nomor line pada cache yang digunakan untuk meletakkan blok main memory ke- $j$ .

§ Jika  $M = 64$  dan  $C = 4$ , maka pemetaan antara line dengan blok menjadi seperti berikut :

Line 0 can hold blocks 0, 4, 8, 12, ...

Line 1 can hold blocks 1, 5, 9, 13, ...

Line 2 can hold blocks 2, 6, 10, 14, ...

Line 3 can hold blocks 3, 7, 11, 15, ...

§ Pada cara ini, *address* pada main memory dibagi 3 *field* atau bagian, yaitu:

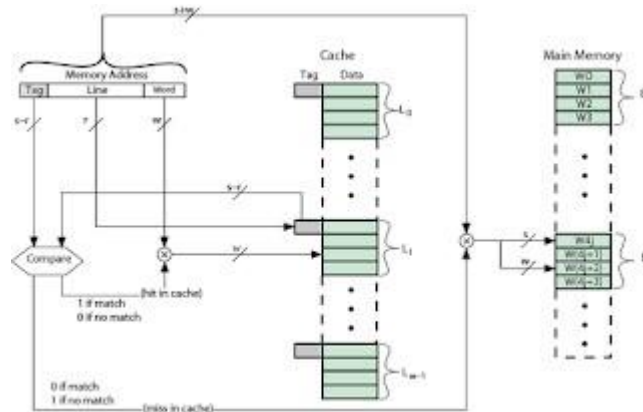
- o Tag identifier.
- o Line number identifier
- o Word identifier (offset)

§ *Word identifier* berisi informasi tentang lokasi word atau unit *addressable* lainnya dalam line tertentu pada cache.

§ *Line identifier* berisi informasi tentang nomor fisik (bukan logika) line pada chace

§ *Tag identifier* disimpan pada cache bersama dengan blok pada *line*.

- o Untuk setiap alamat memory yang dibuat oleh CPU, line tertentu yang menyimpan copy alamat tsb ditentukan, jika blok tempat lokasi data tersebut sudah dikopi dari main memory ke *cache*.
  - o *Tag* yang ada pada line akan dicek untuk melihat apakah benar blok yang dimaksud ada line tsb.



Gambar 2.1 : Gambar Organisasi Direct Mapping.

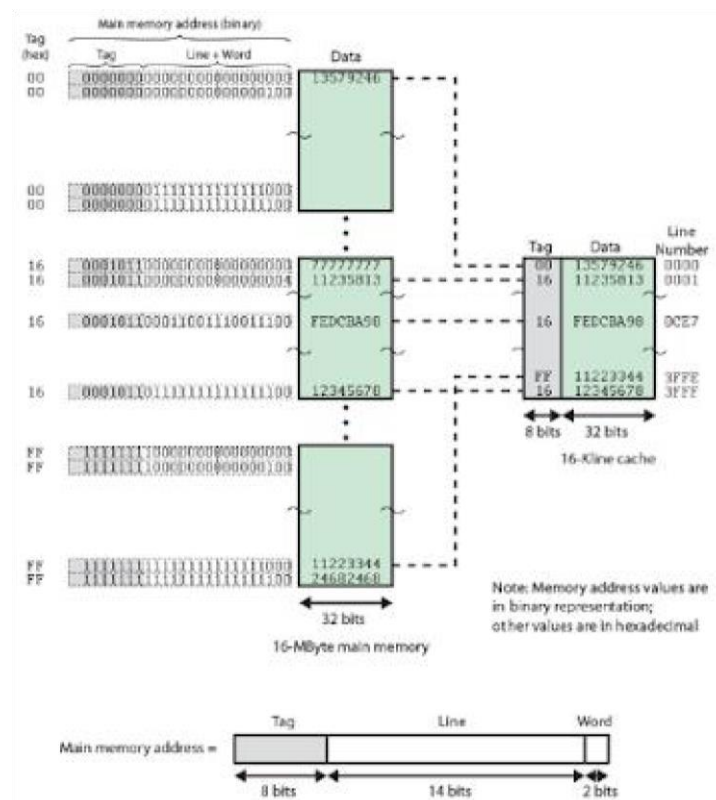
2.

Keuntungan Menggunakan Direct Mapping antara lain :

1. Mudah dan Murah diimplementasikan
2. Mudah untuk menentukan letak salinan data main memory pada chace.

Kerugian menggunakan Direct Mapping antara lain :

1. Setiap blok *main memory* hanya dipetakan pada 1 line saja.
2. Terkait dengan sifat lokal pada *main memory*, sangat mungkin mengakses blok yang dipetakan pada *line* yang sama pada *cache*. Blok seperti ini akan menyebabkan seringnya sapu masuk dan keluar data ke/dari *cache*, sehingga *hit ratio* mengecil. *Hit ratio* adalah perbandingan antara jumlah ditemukannya data pada *cache* dengan jumlah usaha mengakses *cache*.



Gambar 2.2 : Gambar Contoh Pengalamatan Direct Mapping.

Ringkasan *direct mapping* nampak pada tabel berikut:

Item	Keterangan
Panjang alamat	$(s+w)$ bits
Jumlah unit yang dapat dialamati	$2^{s+w}$ words or bytes
Ukuran Bloks sama dengan ukuran Line	$2^w$ words or bytes
Jumlah blok memori utama	$2^s + w/2^w = 2^s$
Jumlah line di chace	$M = 2^r$
Besarnya tag	$(s - r)$ bits

## b. Pemetaan Asosiatif (Associative Mapping)

Pemetaan asosiatif mengatasi kekurangan pemetaan langsung dengan cara mengizinkan setiap blok memori utama untuk dimuatkan ke sembarang saluran cache. Dengan pemetaan asosiatif, terdapat fleksibilitas penggantian blok ketika blok baru dibaca ke dalam cache. Kekurangan pemetaan asosiatif yang utama adalah kompleksitas rangkaian yang diperlukan untuk menguji tag seluruh saluran cache secara parallel, sehingga pencarian data di cache menjadi lama.

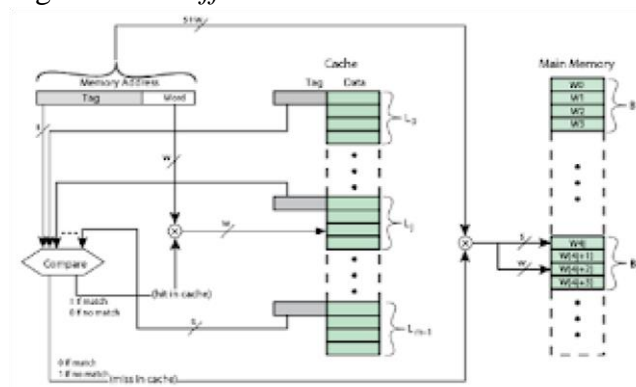
§ Memungkinkan blok diletakkan di sebarang line yang sedang tidak terpakai.

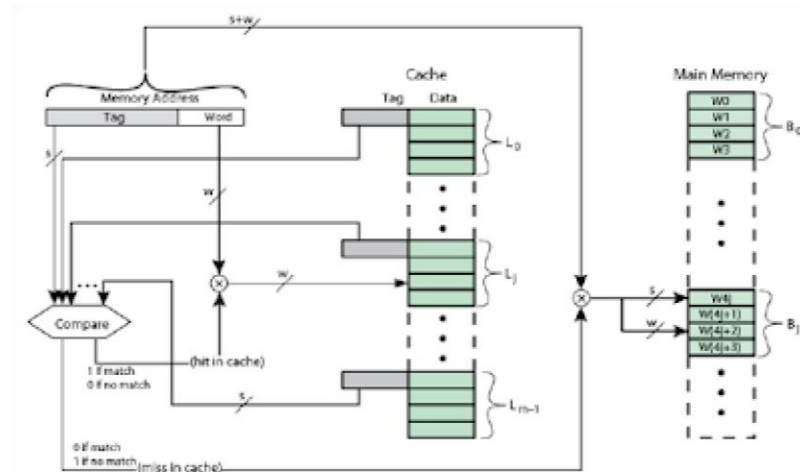
§ Diharapkan akan mengatasi kelemahan utama *Direct Mapping*.

§ Harus menguji setiap *cache* untuk menemukan blok yang diinginkan.

§ Mengecek setiap tag pada line o Sangat lambat untuk *cache* berukuran besar.

§ Nomor line menjadi tidak berarti. *Address main memory* dibagi menjadi 2 field saja, yaitu tag dan *word offset*.

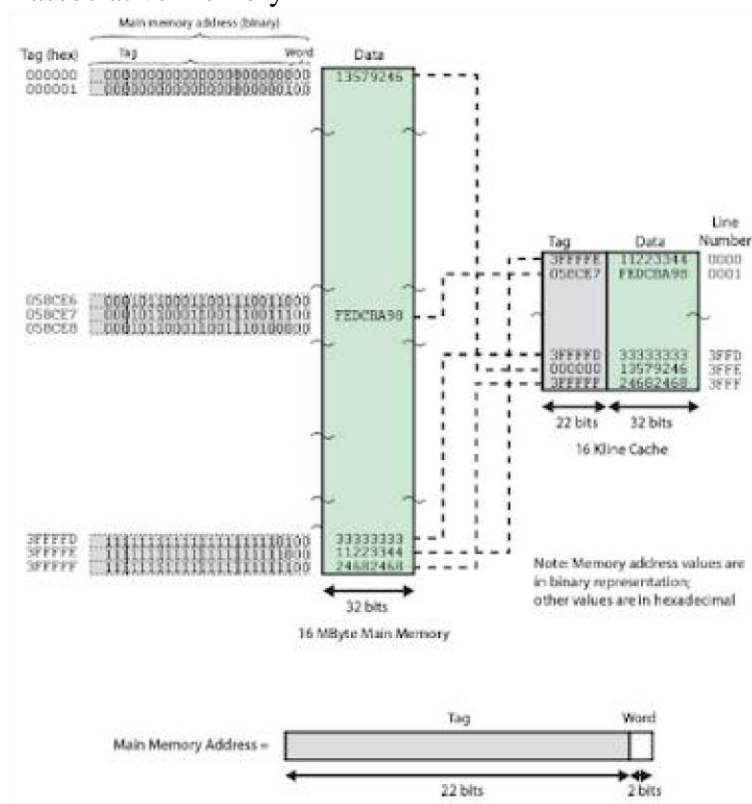




Gambar 2.3 : Gambar Organisasi Associative Mapping.

Cache dibagi menjadi 2 bagian :

- o lines dalam SRAM
- o tag dalam associative memory



Gambar 2.4 : Gambar Contoh Pengalamatan Associative Mapping

Keuntungan Associative Mapping : Cepat dan fleksibel.

Kerugian *Associative Mapping* : Biaya Implementasi, misalnya untuk cache ukuran 8 kbyte dibutuhkan 1024 x 17 bit *associative memory* untuk menyimpan *tag identifier*.

Ringkasan *Associative Mapping* nampak pada tabel berikut:

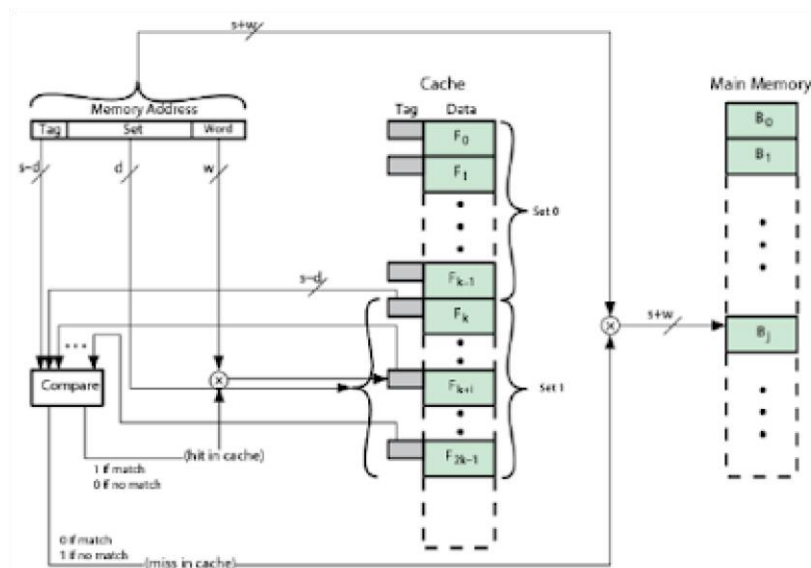
Item	Keterangan
Panjang alamat	$(s+w)$ bits
Jumlah unit yang dapat dialamati	$2s+w$ words or bytes
Ukuran Bloks sama dengan ukuran Line	$2w$ words or bytes
Jumlah blok memori utama	$2s + w/2w = 2s$
Jumlah line di chace	Undetermined
Besarnya tag	$s$ bits

### c. Pemetaan Asosiatif Set (Set Associative Mapping)

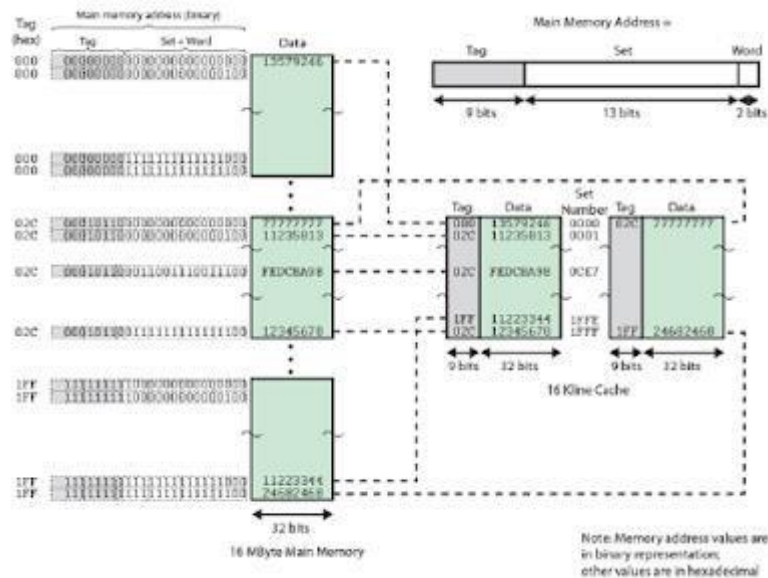
Pada pemetaan ini, cache dibagi dalam sejumlah sets. Setiap set berisi sejumlah line. Pemetaan asosiatif set memanfaatkan kelebihan-kelebihan pendekatan pemetaan langsung dan pemetaan asosiatif.

§ Merupakan kompromi antara *Direct* dengan *Full Associative Mapping*. § Membagi cache menjadi sejumlah set ( $v$ ) yang masing-masing memiliki sejumlah line ( $k$ )

§ Setiap blok dapat diletakkan di sebarang line dengan nomor set: ***nomor set =  $j$  modulo  $v$***



Gambar 2.5 : Gambar Organisasi K-Way Set Associative Mapping.



Gambar 2.6 : Gambar Contoh Pengalamatan 2-Way Associative Mapping.

Ringkasan *Set Associative Mapping* nampak pada tabel berikut:

Item	Keterangan
Panjang alamat	$(s+w)$ bits
Jumlah unit yang dapat dialamati	$2s+w$ words or bytes
Ukuran Bloks sama dengan ukuran Line	$2w$ words or bytes
Jumlah blok memori utama	$2d$
Jumlah line dalam set	$k$
Jumlah set	$V=2d$
Jumlah line di chace	$K_v = k*2d$
Besarnya tag	$(s - d)$ bits

Tabel Memori pada IMB PC

Blok Memori	Alokasi Pemakaian
F0000	ROM BIOS, Diagnostic, BASIC
E0000	ROM program
D0000	ROM program
C0000	Perluasan BIOS untukhardisk XT
B0000	Monokrom Monitor
A0000	Monitor EGA, VGS, dll

90000	Daerah kerjapemakai s/d 640 KB
80000	Daerah kerjapemakai s/d 576 KB
70000	Daerah kerjapemakai s/d 512 KB
60000	Daerah kerjapemakai s/d 448 KB
50000	Daerah kerjapemakai s/d 384 KB
40000	Daerah kerjapemakai s/d 320 KB
30000	Daerah kerjapemakai s/d 256 KB
20000	Daerah kerjapemakai s/d 192 KB
10000	Daerah kerjapemakai s/d 128 KB
00000	Daerah kerjapemakai s/d 64 KB

## 2. Perbedaan antara mode kerja 'Real-Mode' dan mode kerja 'Protect-Mode' pada PC IBM Compatible a. Real-Mode

Real-Mode adalah sebuah modus di mana prosesor Intel x86 berjalan seolaholah dirinya adalah sebuah prosesor Intel 8085 atau Intel 8088, meski ia merupakan prosesor Intel 80286 atau lebih tinggi. Karenanya, modus ini juga disebut sebagai modus 8086 (8086 Mode). Dalam modus ini, prosesor hanya dapat mengeksekusi instruksi 16-bit saja dengan menggunakan register internal yang berukuran 16-bit, serta hanya dapat mengakses hanya 1024 KB dari memori karena hanya menggunakan 20-bit jalur bus alamat. Semua program DOS berjalan pada modus ini.

### b. Protected Mode

Modus terproteksi (protected mode) adalah sebuah modus di mana terdapat proteksi ruang alamat memori yang ditawarkan oleh mikroprosesor untuk digunakan oleh sistem operasi. Modus ini datang dengan mikroprosesor Intel 80286 atau yang lebih tinggi. Karena memiliki proteksi ruang alamat memori, maka dalam modus ini sistem operasi dapat melakukan multitasking.