

# Mengenal Bitmap Index

**Djoni Darmawikarta**

djoni\_darmawikarta@yahoo.ca

## ***Lisensi Dokumen:***

*Copyright © 2003 IlmuKomputer.Com*

*Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.*

Fasilitas bitmap index di database relasional relatif baru. Oracle database misalnya baru memiliki fasilitas ini mulai release 7 (release termutakhir adalah 10) Microsoft SQL Server dan MySQL belum menyediakannya.

Secara umum, index digunakan untuk mempercepat pencarian data didalam tabel database relasional. Didalam index disimpan dan disusun key dari data untuk mencapai tujuan tersebut. Didalam bitmap index yang disimpan adalah bitmap dari setiap key, bukan key-nya sendiri seperti pada sistem index yang umum kita temui.

Untuk memperjelas kita gunakan contoh tabel data berikut. (kotak biru dan merah akan dibahas dibawah).

## **PENJUALAN**

BULAN	TAHUN	KODE INDUSTRI	TIPE PELANGGAN	KODE PRODUK	PROPINSI	NEGARA	PENJUALAN	LABA PEJUALAN
JAN	2002	313	PBRK	WDM	KALTENG	INDONESIA	1500000000	1170000000
JAN	2002	313	PBRK	WTP	KALTENG	INDONESIA	600000000	480000000
JAN	2002	313	PBRK	PE1	KALTENG	INDONESIA	600000000	447000000
JAN	2002	313	PBRK	PE2	KALTENG	INDONESIA	1200000000	897000000
JAN	2002	321	IMPR	WDM		JAPAN	1000000000	400000000
JAN	2002	321	IMPR	PE2		JAPAN	4200000000	3190000000
APR	2003	313	PBRK	WTP	KALTENG	INDONESIA	15000000000	12450000000

Andaikan KODE INDUSTRI dan KODE PRODUK adalah key yang ingin kita buat bitmap index-nya, kita gunakan SQL statement berikut:

```
CREATE BITMAP INDEX penjualan_bmi ON penjualan(kode_industri, kode_produk);
```

Dimana penjualan\_bmi adalah nama yang kita berikan untuk bitmap index ini.

KODE INDUSTRI memiliki dua buah nilai yang berbeda, yaitu 313 dan 321. Sedang KODE PRODUK empat nilai: WDM, WTP, PE1, dan PE2. Total 6 buah bitmap (6 kolom)

Maka untuk 7 baris data tabel PENJUALAN diatas bitmap index-nya akan berisi nilai-nilai bit sebagai berikut: 1 bila nilai key benar untuk baris bersangkutan, 0 bila nilai key tidak benar untuk baris bersangkutan. Lihat dua buah contoh: kode industri 313 (kotak biru) dan kode produk PE1 (kotak merah)

313	321	WDM	WTP	PE1	PE2
1	0	1	0	0	0
1	0	0	1	0	0
1	0	0	0	1	0
1	0	0	0	0	1
0	1	1	0	0	0
0	1	0	0	0	1
1	0	0	1	0	0

Bila ada SQL query misalnya:

SELECT \* FROM penjualan WHERE kode\_industri = '313' AND kode\_produk = 'PE1'

Database memanfaatkan bitmap index untuk mencari baris data dimana kolom 313 dan PE1 bernilai 1.

Hasilnya adalah baris ke-3.

BULAN	TAHUN	KODE INDUSTRI	TIPE PELANGGAN	KODE PRODUK	PROPINSI	NEGARA	PENJUALAN	LABA PEJUALAN
JAN	2002	313	PBRK	PE1	KALTENG	INDONESIA	600000000	447000000

Untuk SQL query:

SELECT \* FROM penjualan WHERE kode\_industri = '313' OR kode\_produk = 'PE1'

Database memanfaatkan bitmap index untuk mencari baris data dimana kolom salah satu atau kedua kolom 313 dan PE1 bernilai 1.

Hasilnya adalah:

BULAN	TAHUN	KODE INDUSTRI	TIPE PELANGGAN	KODE PRODUK	PROPINSI	NEGARA	PENJUALAN	LABA PEJUALAN
JAN	2002	313	PBRK	WDM	KALTENG	INDONESIA	1500000000	1170000000
JAN	2002	313	PBRK	WTP	KALTENG	INDONESIA	600000000	480000000
JAN	2002	313	PBRK	PE1	KALTENG	INDONESIA	600000000	447000000
JAN	2002	313	PBRK	PE2	KALTENG	INDONESIA	1200000000	897000000
APR	2003	313	PBRK	WTP	KALTENG	INDONESIA	1500000000	1245000000

Kecepatan pencarian data menggunakan bitmap index sangat tinggi, karena dalam melaksanakan condition WHERE dari query hanyalah pada dua nilai bit 1 dan 0.

Makin banyak nilai suatu key, bitmap index akan makin besar (meskipun tetap lebih kecil dibanding menyimpan key sendiri), dan tentu saja akibatnya selain tempat yang makin besar diperlukan untuk menyimpannya juga kecepatan akan makin menurun.

Oracle memberikan petunjuk sebagai berikut:

Kecepatan pelaksanaan query menggunakan bitmap index kemungkinan lebih tinggi dibanding jenis index yang lain (misalnya yang umum disediakan adalah B\*tree index)

- Bila banyaknya nilai key kurang dari 1% banyaknya baris data atau suatu nilai key dipakai lebih dari 100 kali (100 baris data)
- Bila makin kompleks WHERE condition didalam SQL query kita.

Selain kedua petunjuk diatas, tentu saja kalau tabel data kita dinamik (berubah isinya), overhead pemeliharaan bitmap index harus diperhatikan; jangan sampai mengganggu kecepatan databasenya secara menyeluruh.