

Indexing

Monday, September 15, 2025

4:33 PM

Yang mana sajakah pernyataan yang benar?

- ☒ Search key pada primary index selalu merupakan primary key. *↳ biasanya, tapi tidak harus*
- ☒ Index file adalah file berisi record index yang terdiri atas search key dan pointer.
- ☒ Kegunaan index adalah untuk mendukung akses yang efisien ke data.
- ☒ Jika data pada suatu atribut tabel yang diberi index diubah, maka index file tidak akan berubah.
- ☒ File index biasanya berukuran lebih kecil dari file data yang di-index.

Primary index dan secondary index adalah tipe dari:

- ☒ Ordered index *↳ search-key value*
- ☐ Unordered index
- ☐ Linear index
- ☐ Relative search index
- ☐ Multi table clustering index

Parameter yang dapat digunakan untuk mengevaluasi kinerja index adalah:

- ☒ Waktu yang dibutuhkan untuk pengaksesan data *access time*
- ☒ Waktu yang dibutuhkan untuk menambahkan data *insertion time*
- ☒ Waktu yang dibutuhkan untuk menghapus data *deletion time*
- ☒ Space overhead
- ☐ Jumlah record yang disimpan

Manakah dari pernyataan berikut ini yang benar?

- ☒ Sparse index files memiliki maintenance overhead untuk proses insert dan delete yang lebih sedikit dibanding dengan dense index
- ☒ Sparse index files membutuhkan space yang lebih sedikit dibanding dengan dense index
- ☒ Sparse index files membutuhkan waktu yang lebih relative sedikit dalam menemukan lokasi records dibandingkan dengan dense index *↳ generally slower*
- ☒ Sparse index files membutuhkan waktu yang relative lebih banyak dalam menemukan lokasi records dibandingkan dengan dense index
- ☒ Tidak ada kelebihan dari sparse index files dibandingkan dengan dense index

- ☒ Tertiary index
- ☒ Secondary index
- ☒ Primary index
- ☒ Non-clustered index
- ☒ Unary index

] alias

Diketahui B+tree index sebagai berikut:

75 < 90 (1)

(2)

(3)

(4)

(5)

batas STOP

75 ada yg memenuhi, lanjut ke pointer

Berapa banyak jumlah nodes minimum (termasuk *root node*) yang perlu dilewati untuk menjawab query “Carilah semua record yang memiliki search key lebih besar atau sama dengan 75 dan kurang dari 105”.

write your answer here

5 nodes

[Save Answer](#)

(17, 3, 2, 11, 19, 7, 29, 23, 31, 5)

1. Buatlah B+-tree indeks untuk file di atas dengan jumlah pointer dalam 1 node adalah 5. Asumsikan bahwa awalnya pohon kosong.
2. Buatlah struktur hash index untuk file tersebut jika fungsi hash $h(x) = x \bmod 8$, dan bucket dapat menyimpan 3 records.

Paragraph

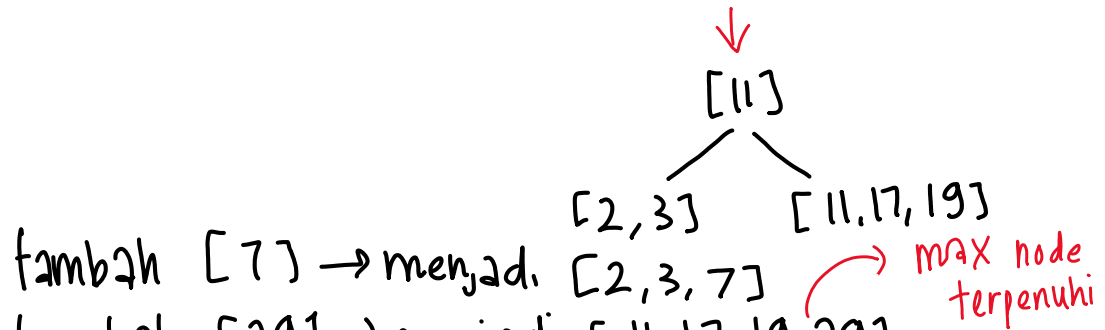
① jumlah pointer dalam 1 node = 5
 jumlah maksimum key dalam 1 node = $p - 1 = 4$
 urutan search-key = (17, 3, 2, 11, 19, 7, 29, 23, 31, 5)
 root = tree kosong \rightarrow [] \rightarrow leaf \rightarrow max node

urutan search key = (11, 3, 2, 11, 17, 19, 23, 29, 31)

awal = tree kosong $\rightarrow [17]$ leaf max node terpenuhi

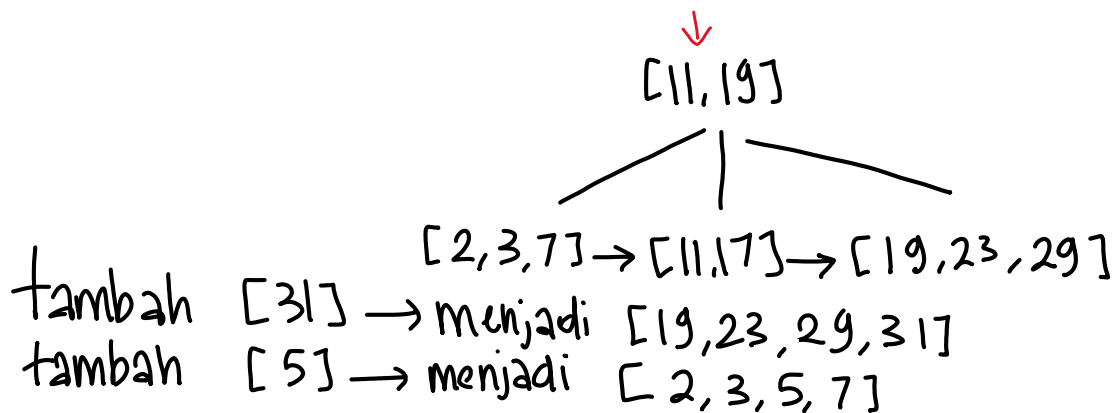
tambah [3, 2, 11] $\rightarrow [2, 3, 11, 17]$

tambah [19] \rightarrow melebihi kapasitas, dilakukan split dari [2, 3, 11, 17, 19]

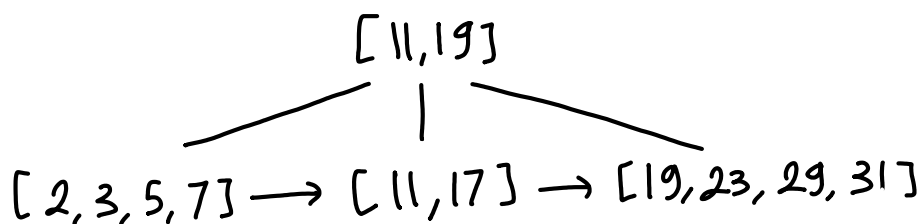


tambah [29] \rightarrow menjadi [11, 17, 19, 29]

tambah [23] \rightarrow melebihi kapasitas, dilakukan split dari [11, 17, 19, 23, 29]



Struktur B+ - index.



② fungsi hash $h(x) : x \bmod 8$

bucket menyimpan 3 record
perhitungan hash.

$17 \bmod 8 \rightarrow 1$

$3 \bmod 8 \rightarrow 3$

struktur hash index.

$1 \mod 8 \rightarrow 1$
 $3 \mod 8 \rightarrow 3$
 $2 \mod 8 \rightarrow 2$
 $11 \mod 8 \rightarrow 3$
 $19 \mod 8 \rightarrow 3$
 $7 \mod 8 \rightarrow 7$
 $29 \mod 8 \rightarrow 5$
 $23 \mod 8 \rightarrow 7$
 $31 \mod 8 \rightarrow 7$
 $5 \mod 8 \rightarrow 5$

STRUKTUR RIJSEN INDEX.

Bucket 0 = []

Bucket 1 = [17]

Bucket 2 = [2]

Bucket 3 = [3, 11, 19]

Bucket 4 = []

Bucket 5 = [29, 5]

Bucket 6 = []

Bucket 7 = [7, 23, 31]

Bucket 8 = []