

1. Mengapa pemakaian operasi join perlu dievaluasi ? pendekatan apa yang diperlukan dalam evaluasi tersebut ?
2. Jelaskan mengapa perbedaan dalam hal biaya diantara rencana alternatif untuk query dapat menjadi besar ? berikan contoh kasus khusus untuk mengilustrasikan pengaruh push selection, pemilihan metode join dan ketersediaan index yang tepat.
3. Rencana apa yang digunakan untuk sub query dengan satu relasi ? , rencana apa yang digunakan dalam pendekatan pemrograman dinamis untuk mengumpulkan rencana left-deep ? jelaskan metode akses dan output order dalam jawaban anda. Secara khusus jelaskan rencana index-only dan mengapa rencana tersebut menarik ?
4. Perhatikan tabel relasi siswa (yang terurut berdasarkan umur) berikut ini :

| <i>nim</i> | <i>nama</i> | <i>alamat</i> | <i>umur</i> | <i>ipk</i> | |
|------------|-------------|---------------|-------------|------------|--------|
| 53831 | Haikal | ambarawa | 16 | 1.8 | Page 1 |
| 53832 | Tiara | bogor | 17 | 2.0 | Page 1 |
| 53666 | Jenar | jogja | 18 | 3.4 | Page 1 |
| 53688 | Sunu | solo | 19 | 3.2 | Page 2 |
| 53650 | Sanaz | surabaya | 19 | 3.8 | Page 2 |

Asumsikan tuple-tuple pada tabel di atas tersimpan dalam urutan file sebagai berikut : tuple pertama ada pada halaman 1, tuple kedua juga pada halaman 1, dan seterusnya. Setiap halaman dapat menyimpan sampai dengan 3 data record. Sehingga tuple keempat akan berada pada halaman 2.

Jelaskan data-data entri yang bagaimanakah yang berada di dalam indeks-indeks berikut. Jika entri-entrinya signifikan, jelaskan alasannya. Jika terdapat indeks yang tidak dapat dikonstruksi, jelaskan alasannya.

- a. Sebuah unclustered index pada *umur* menggunakan Alternatif(1)
- b. Sebuah unclustered index pada *umur* menggunakan Alternatif(2)
- c. Sebuah unclustered index pada *umur* menggunakan Alternatif(3)
- d. Sebuah clustered index pada *ipk* menggunakan Alternatif(1)
- e. Sebuah clustered index pada *ipk* menggunakan Alternatif(2)
- f. Sebuah clustered index pada *ipk* menggunakan Alternatif(3)

5. Asumsikan anda baru saja membuat membuat B+ tree index menggunakan alyternatif (2) pada sebuah heap file yang berisi 20.000 record . Key field pada B+ tree ini adalah string 40-byte dan merupakan candidate key. Pointer (missal , id record dan id page) adalah paling banyak 10 byte value. Jika ukuran disk sebuah disk page adalah 1000 byte. Sedangkan index terbentuk melalui pendekatan bottom up dan algoritma yang digunakan bulk loading, maka :
- Berapa banyak level yang dimiliki oleh tree yang dihasilkan tersebut ?
 - Untuk tiap level tree berapa banyak node pada masing-masing level ?

(*Slamat Mengerjakan*)