

**UNIVERSITAS INDONESIA
FAKULTAS ILMU KOMPUTER**

STRUKTUR DATA & ALGORITMA (IKI10100)

UJIAN TENGAH SEMESTER
SEMESTER GENAP 2000/2001
KAMIS, 29 MARET 2001 PK: 10.00 – 12:00
KELAS B
Open Book

Nama	
NPM	
Kelas	B

- Semua pekerjaan dilakukan pada kertas ini pada tempat yang disediakan. Jika tempatnya tidak cukup, bisa menggunakan kertas tambahan.
- Boleh dikerjakan dengan menggunakan pensil.
- Baca soal dengan teliti. Jika anda mempunyai asumsi lain terhadap soal, tulis asumsi anda dan kerjakan.
- Kejujuran dan integritas harap menjadi perhatian utama setiap peserta. Contek-mencontek nilai E !
- Nilai total adalah 110. 10-nya adalah bonus. Masing-masing soal memiliki bobot yang berbeda.

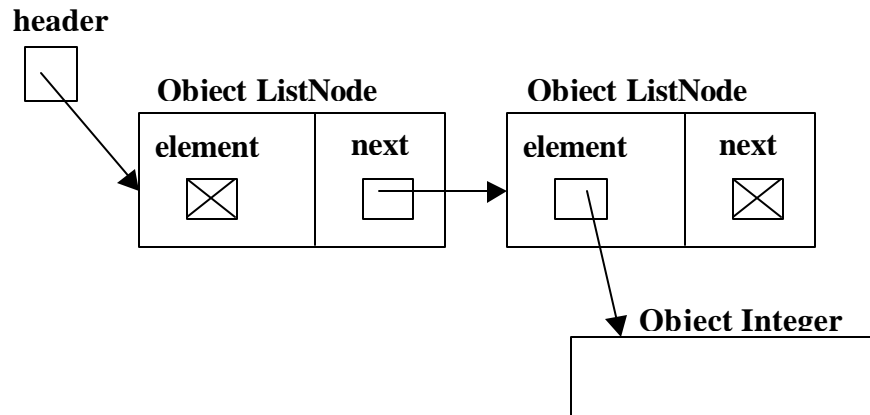
Kotak ini jangan diisi

Bagian	Max score	Score
Umum	20	
Analisa algoritme	20	
Linked-List	30	
Tree	40	
TOTAL	110	

Umum

1. (5) Nama class yang tidak mempunyai superclass adalah

2. (15) Gambarkan sebuah diagram *kotak dan pointer* (seperti linked-list yang digambar di kelas) untuk nilai variabel **x** yang dihasilkan oleh program di bawah. Contoh di bawah adalah diagram kotak dan pointer dari linked-list.



```
class Mid {
    private int i;
    private Mid mid1, mid2;
    public Mid (int i, Mid mid1, Mid mid2) {
        this.i = i;
        this.mid1 = mid1;
        this.mid2 = mid2;
    }

    public static void main (String[] args)
    {
        Mid x = new Mid (10, new Mid (5, null,
            new Mid (7, null, null)), new Mid (0, null, null));
    }
}
```

Analisa Algoritme

3. (5) Sebuah algoritme memerlukan waktu 1 detik untuk jumlah input 100. Jika algoritma tersebut n^3 , berapa kira-kira waktu yang dibutuhkan untuk memproses input sebanyak 10000?

4. Perhatikan fragmen program di bawah ini, kemudian hitung kompleksitasnya.

Running time untuk:

- method `panggilA (int n)` adalah $O(n)$

```
public static void methodA (int n)
{
    for (int ii = 0; ii < n; ii++) {
        panggilA (ii);
    }
}
```

```
public static int methodB (int n)
{
    int sum = 0;
    for (int ii = -n; ii < n; ii++) {
        for (int jj = 0; jj < 100; jj++) {
            sum++;
        }
    }
    return sum;
}

public static int methodC (int n)
{
    int sum = 0;
    methodA (n);
    for (int ii = -n; ii < n; ii++) {
        for (int jj = 0; jj < 100; jj++) {
            sum++;
        }
    }
    return sum;
}
```

a. (5) berapa kompleksitas (running time dalam notasi **big-Oh**) untuk **methodA**

b. (5) berapa kompleksitas (running time dalam notasi **big-Oh**) untuk **methodB**

c. (5) berapa kompleksitas (running time dalam notasi **big-Oh**) untuk **methodC**

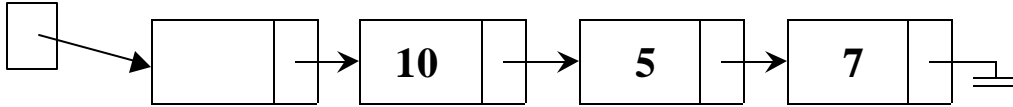
Linked-List

5. (10) apakah **binary search** lebih cepat dibandingkan dengan **sequential search** pada **doubly linked-list**? **Jelaskan dengan singkat! (Tanpa penjelasan, nilai 0)**

Mid Test SDA B

6. (20 pts) Menghitung total nilai integer dalam object **Integer** pada linked list dengan menggunakan versi **rekursif**. Contoh: untuk linked-list di bawah, pemanggilan terhadap method **totalList (header.next)** akan mengembalikan nilai **22**.

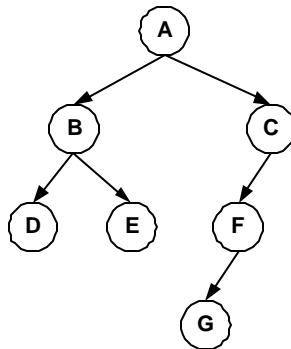
header



```
public static int totalList (ListNode node)
{
}
}
```

Tree

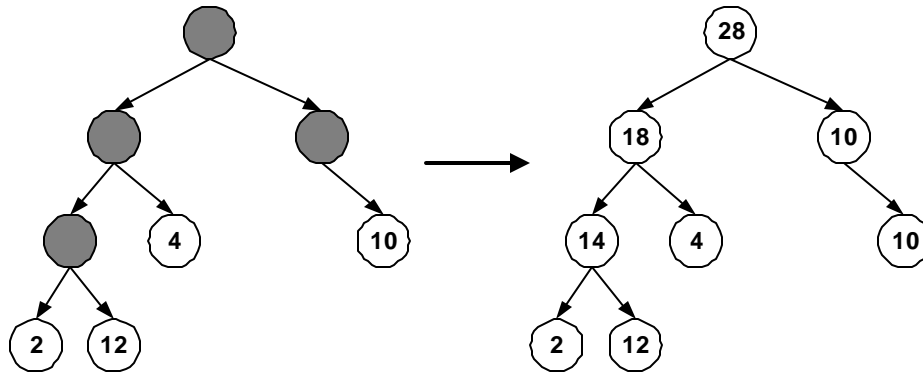
7. (20 pts) Buatlah sebuah class iterator untuk level-order traversal, tetapi dari bawah ke atas. Untuk tree (pohon) dibawah ini urutan kunjungannya adalah: **G D E F B C A**



Manfaatkan class Stack, Queue dan class-class Iterator yang sudah ada di buku / diajarkan di kelas!

Mid Test SDA B

8. (20 pts) Sebuah binary tree yang sudah terbentuk (sudah ada). Hanya pada leaf-nya saja yang telah berisi (menunjuk) sebuah object Integer. Buatlah sebuah method rekursif untuk mengisi node-node lainnya yang merupakan jumlah dari semua simpul anak-anaknya.



```
public static int melengkapiPohon (BinaryNode root)
{
}
}
```