



INSTITUT TEKNOLOGI DEL
UJIAN TENGAH SEMESTER
Semester Gasal 2020/2021

Name	
NIM	
Tanggal	
Kode>Nama Mata Kuliah	1132102 / PBO
Tipe Ujian	TEORI (<i>online</i>)
Durasi	30 menit
Dosen	Teamsar Muliadi Panggabean (TMP)

NOTE:

- i. Anda tidak diperkenankan membuka barang-barang yang tidak diizinkan selama ujian berlangsung, seperti: catatan, kamus, *handphone*, maupun sumber lain yang berisikan materi perkuliahan.
- ii. Pastikan anda merekam sesi ujian daring anda dengan mengikuti instruksi yang telah diberikan sebelumnya.
- iii. Rekaman sesi ujian daring haruslah **selesai di-upload** paling lambat 3 jam setelah sesi ujian ini berakhir ke *YouTube channel* anda, dan *link* video tersebut haruslah dikirimkan ke WAG PBO. Bila tidak, maka ujian anda dianggap **batal**.
- iv. Ketika ujian dimulai, baca semua pertanyaan dengan hati-hati. Anda diperkenankan untuk bertanya 15 menit pertama ketika ujian dimulai lewat WAG PBO. Anda tidak akan diberikan *hint* ataupun *kata kunci* untuk menjawab pertanyaan yang diberikan.
- v. Disarankan men-*submit* hasil ujian 10 menit diawal untuk menghindari gagal *submit*.
- vi. Segala kegiatan yang teridentifikasi melakukan tindak kecurangan, akan diberi sanksi akademik, serta ujian dianggap **batal**.

AWAL SOAL

1. *Keyword* yang digunakan untuk menurunkan segala sifat dari kelas indukan ke kelas anakan adalah...
 - a. extend
 - b. extends**
 - c. implement
 - d. implements
 - e. overriding

2. Manakah *keyword* di bawah ini yang dapat digunakan untuk mengakses fitur-fitur dari sebuah *package*?
 - a. package
 - b. import**
 - c. extends
 - d. implements
 - e. extend

3. Sebuah *package* merupakan kumpulan...
 - a. classes
 - b. interfaces
 - c. editing tools
 - d. classes & interfaces**
 - e. editing tools & interfaces

4. Bila diberi sebuah kelas dengan nama Mahasiswa, manakah dari beberapa pernyataan di bawah ini yang menunjukkan pendeklarasian *constructor* yang benar?
 - a. Mahasiswa(mahasiswa m){ }**
 - b. Mahasiswa mahasiswa(){ }
 - c. private final mahasiswa(){ }
 - d. void mahasiswa(){ }
 - e. static void mahasiswa(){ }

5. Kode program Java ketika di-*compile* akan berubah menjadi...
 - a. obj
 - b. package

- c. class
- d. bytecode
- e. exe

6. Perhatikan beberapa pernyataan di bawah ini terkait konsep *multiple inheritances* di dalam pemrograman Java:

- I. Meng-extends dua atau lebih kelas
- II. Meng-extends satu kelas dan mengimplementasikan satu atau lebih *interfaces*.
- III. Mengimplementasikan dua atau lebih *interfaces*.

Berdasarkan beberapa pernyataan di atas maka yang **benar** adalah...

- a. II
- b. I dan II
- c. II dan III
- d. I
- e. III

7. Berikut ini pernyataan yang benar di *Java Programming* adalah...

- a. Sebuah kelas hanya memiliki *method* berupa *constructor*, *setter* dan *getter*.
- b. Sebuah kelas boleh memiliki sebuah *method* tanpa *body method*.
- c. *main method* akan di-eksekusi apabila sebuah kelas *static* sudah di-instansiasi terlebih dahulu.
- d. Sebuah program tidak boleh memiliki lebih dari satu *class*.
- e. Di pemrograman java *runtime error* akan muncul pada saat kompilasi.

8. Berikut *keyword* yang **tidak** dikenal di pemrograman Java ...

- a. goto
- b. throws
- c. transient
- d. instance
- e. volatile

9. Berikut ini cara yang benar untuk mendeklarasikan sebuah konstanta pi di *Java Programming* adalah...

- a. `consts float pi = 3.14`

- b. `final consts float pi = 3.14`
- c. `const final float pi = 3.14`
- d. `final float pi = 3.14`
- e. `float final pi = 3.14`

10. Berapakah keluaran dari kode program di bawah ini?

```
1. int k = 10;
2. return (k > 20/2) ? ((k*2) % 2) : (k & 2);
```

- a. 2
- b. 5
- c. 10
- d. 12
- e. 20

11. Berikut ini **dua** pernyataan yang benar terkait *class* di pemrograman java...

- a. Sebuah *class* merupakan objek dari *class* lain.
- b. **Struct** di dalam pemrograman C tidak sama dengan *class* yang ada di pemrograman java.
- c. **Interface** juga merupakan *class* yang ada di pemrograman Java.
- d. Sebuah *class* merupakan tipe di dalam pemrograman java.
- e. Penamaan kelas *inner class* di pemrograman Java harus dimulai dari nama *class* induk kemudian diikuti dengan nama kelas *inner class* itu sendiri.

12. Manakah **satu** dari beberapa pernyataan di bawah ini yang **benar**?

- a. Sebuah method dengan parameter **p** yang bertipe **X** (bukan tipe data primitif) dapat di isi dengan *null value* oleh method pemanggil (*invoker*).
- b. Dengan adanya *keyword new* sebuah kelas dapat menurunkan segala *property* nya ke kelas anakan.
- c. Sebuah *non-static class X* tidak harus di instansiasi untuk dapat digunakan segala *property* yang dimiliki nya.
- d. Jika kondisi repetisi *loop salah*, maka satu atau beberapa kode program di dalamnya akan dieksekusi.
- e. **X** dari kode program `X objX = new X();` **bukan** merupakan sebuah tipe data, melainkan sebuah *class*.

13. Manakah **satu** dari beberapa pernyataan di bawah ini yang **benar**?

- a. Jumlah parameter yang ada di dalam sebuah *constructor* **bukan** menjadi penentu banyaknya data yang harus dikirimkan ketika proses instansiasi sebuah kelas.
- b.** Jumlah parameter yang ada di dalam sebuah *constructor* **dapat** menjadi penentu banyaknya data yang harus dikirimkan ketika proses instansiasi sebuah kelas
- c. Sebuah *class* hanya memiliki satu *constructor*.
- d. *Overloading constructors* tidak diperkenankan di pemrograman java.
- e. Nama *constructor* boleh berbeda dengan nama kelas.

14. Manakah **dua** dari beberapa pernyataan di bawah ini yang **benar**?

- a. Sebuah kelas *X* tidak dapat dijadikan tipe data parameter dari sebuah *method*.
- b.** Apapun data yang dikirimkan oleh *method invoker* akan dapat diterima oleh *method* yang dipanggil (*invoked*) selama parameternya bertipekan *Object*.
- c. Bila sebuah *method* dengan nama "hitungLuasSegitiga" dengan 2 parameter nya bertipekan *Object*, maka data yang dikirimkan **tidak boleh** bertipe *float*.
- d. *Passing parameter* dengan konsep *call-by-reference* hanya dapat dilakukan dengan melempar *value* dari sebuah *variable*.
- e. *Passing parameter* dengan konsep *call-by-value* hanya dapat dilakukan dengan melempar objek dari sebuah kelas.

15. Apakah keluaran dari program di bawah ini?

```
class Test{
    int a, b;

    void meth(int i, int j){
        i = i*2;
        j = j/2;
    }
}

class X{
    public static void main(String args[]) {
        Test ob = new Test();
        ob.a = 6;
        ob.b = 8;
        System.out.println("ob.a and ob.b before call: " + ob.a + " " + ob.b);
        ob.meth(ob.a, ob.b);
        System.out.println("ob.a and ob.b after call: " + ob.a + " " + ob.b);
    }
}
```

- a. ob.a and ob.b before call: 6 8
ob.a and ob.b after call: 3 4
- b. ob.a and ob.b before call: 3 4
ob.a and ob.b after call: 6 8
- c.** ob.a and ob.b before call: 6 8
ob.a and ob.b after call: 6 8
- d. ob.a and ob.b before call: 8 6
ob.a and ob.b after call: 4 3
- e. ob.a and ob.b before call: 4 3
ob.a and ob.b after call: 8 6

16. Apakah keluaran dari program di bawah ini?

```
class Test{
    int a, b;

    Test(int i, int j){
        a = i; b = j;
    }

    void meth(Test o){
        int temp = o.a;
        o.a = o.b;
        o.b = temp;
    }
}

class X{
    public static void main(String a[]) {
        Test ob = new Test(7, 6);

        System.out.println("ob.a and ob.b before call: " + ob.a + " " + ob.b);
        ob.meth(ob);
        System.out.println("ob.a and ob.b after call: " + ob.a + " " + ob.b);
    }
}
```

- a. ob.a and ob.b before call: 6 7
ob.a and ob.b after call: 6 7
- b. ob.a and ob.b before call: 7 6
ob.a and ob.b after call: 7 6
- c. ob.a and ob.b before call: 6 7
ob.a and ob.b after call: 7 6
- d.** ob.a and ob.b before call: 7 6
ob.a and ob.b after call: 6 7
- e. ob.a and ob.b before call: 6 7
ob.a and ob.b after call: 6 6

17. Manakah **satu** dari beberapa pernyataan di bawah ini yang **benar**?

- a. Sebuah kelas anakan dapat meng-*inherit* lebih dari satu kelas induk.
- b. Sebuah kelas induk dapat meng-*inherit* lebih dari satu kelas anakan.
- c. Untuk dapat melakukan *inheritance* dibutuhkan kata kunci *extend*.
- d. Untuk dapat melakukan *inheritance* dibutuhkan kata kunci *extent*.
- e. Sebuah kelas induk dapat di-instansiasi di dalam "driver" java.

18. Manakah **satu** dari beberapa pernyataan di bawah ini yang **benar**?

- a. Semua *instance variables* dari kelas induk dapat diakses oleh kelas anakan.
- b. Salah satu akses *modifier* yang tepat untuk digunakan bila seseorang ingin memberi *accessibility* dari sebuah data member ke kelas *sub-class*nya adalah **protected**.
- c. Akses *modifier* **private** memberikan hak akses secara *partial* terhadap data member lainnya di dalam kelas yang sama.
- d. Akses *modifier* **friendly** (no modifier) memberikan hak akses kepada data member lainnya di dalam kelas anakan yang nama *package* nya berbeda.
- e. Akses *modifier* **public** hanya akan memberikan *accessibility* penuh, **jika** kelas anakan berada tepat berada di dalam *package* yang sama.

19. Manakah dari pernyataan berikut yang **benar**?

- a. Hanya melalui instansiasi kelaslah sebuah *variable* **static** dapat digunakan.
- b. Sebuah *method* **static** dapat di-*invoke* oleh semua *method*.
- c. *Variable* **static** tidak dapat dikatakan sebagai *global variable*.
- d. *Variable* **static** dapat langsung digunakan **tanpa** harus memerlukan instansiasi sebuah kelas terlebih dahulu.
- e. Harus *method* **non-static** lah yang dapat meng-*invoke* *method* **static**.

20. Kata kunci yang dipakai untuk merealisasikan sebuah *interface* oleh sebuah kelas adalah...

- a. *extend*
- b. *implement*
- c. *abstract*
- d. *implements*
- e. *super*

21. Sebuah kelas dapat merealisasikan lebih dari satu *interfaces*.

- a. True
- b. False

22. Sebuah *interface* dapat menggunakan sebuah kelas.

- a. True
- b. False

23. Sebuah *interface* dapat meng-*extend* kelas abstrak.

- a. True
- b. False

24. Sebuah *interface* dapat meng-*inherit* *interface* lainnya.

- a. True
- b. False

25. *Multiple methods* dengan nama yang sama namun memiliki *return type* dan urutan *argument* yang berbeda disebut...

- a. Overriding method
- b. Overloading method
- c. Abstract method
- d. Inheriting method
- e. Extending method

26. *Multiple methods* dengan nama yang sama namun memiliki jumlah parameter yang berbeda disebut...

- a. Overriding method
- b. Overloading method
- c. Abstract method
- d. Inheriting method
- e. Extending method

27. Manakah **satu** dari pernyataan di bawah ini yang **benar**?

- a. *Polymorphism* dapat tercapai apabila sebuah kelas memiliki nama *method* yang unik.
- b. Jika seseorang melakukan *overriding method* dari sebuah kelas, dimana kelas tersebut menggunakan sebuah *interface*, dapat dikatakan bahwa orang tersebut sudah melakukan konsep *polymorphism*.

- c. *Overloading constructors* tidak dapat disebut dengan *polymorphism*, karena *constructor* merupakan *method* spesial dari sebuah kelas.
- d. Konsep *polymorphism* tidak dapat diterapkan di dalam sebuah *interface*.
- e. Konsep *polymorphism* tidak dapat diterapkan di dalam sebuah *abstract class*.

28. Di dalam konsep *exception handling*, kata kunci *finally* selalu di-eksekusi di akhir *sub-routines* yang berada di *body try*.

- a. True
- b. False

29. Ketika seseorang menerapkan *multiple "catch"* di dalam programnya, level terakhir *catch* haruslah berisikan informasi yang spesifik terkait dengan *error*.

- a. True
- b. False

30. *Body* di dalam *catch* akan tetap di-eksekusi meskipun tidak ada kegagalan pada saat *runtime*.

- a. True
- b. False

AKHIR SOAL