

[Home](#) / [My courses](#) / [PROG. S1 FAK. REGULER](#) / [REG - Genap 2024/2025](#) / [\[Reg\] Dasar-Dasar Pemrograman 2 \(A s.d. G\) Genap 2...](#)
/ [Pekan 10 : Mid-Term Exam](#) / [Soal Bagian Programming 1-4 - UTS Sessi 2](#)

Started on Tuesday, 15 April 2025, 4:30 PM

State Finished

Completed on Tuesday, 15 April 2025, 6:15 PM

Time taken 1 hour 44 mins

Grade **0.00** out of 16.00 (0%)



Question 1

Not answered

Marked out of 4.00

Sebuah **kalimat** disebut **pangram** jika mengandung **semua huruf alfabet** dari 'a' hingga 'z' setidaknya satu kali, **tanpa memperhatikan huruf kapital**.

Buatlah sebuah program dalam **Java** untuk memeriksa apakah input string merupakan pangram.

Input:

Sebuah baris teks, berupa kalimat.

Output:

Cetak:

"Kalimat adalah pangram." jika kalimat memenuhi syarat pangram.

"Kalimat bukan pangram." jika tidak.

Test case:

Input: **The quick brown fox jumps over the lazy dog**

Output: **Kalimat adalah pangram.**

[Template](#)

[TestFile](#)

Hint:

Silahkan gunakan `java.util.Arrays`; dan method-method yang ada dibawah ini

- `replaceAll("\\s+", "")`: Menghapus semua spasi jika ada.
- `toLowerCase()`: Membuat perbandingan tidak peka huruf besar/kecil.
- `Arrays.sort()`: Mengurutkan karakter.
- `Arrays.equals()`: Memeriksa apakah kedua array karakter sama.
- `char[] arr1 = s1.toCharArray()`; // Ubah string s1 ke array of karakter

Acuan penilaian:

Poin 4 : Jika luaran program sudah tepat dan coverage dari test case = 100%

Poin 3: Jika luaran program sudah tepat dan coverage dari test case $\geq 80\%$

Poin 2: Jika luaran program sudah tepat namun test case tidak tersedia

Poin 1: Jika luaran program tidak tepat dan test case juga belum tersedia dengan benar

Poin 0: Compile Error

Response history

Step	Time	Action	State	Marks
1	15/04/25, 16:30	Started	Not yet answered	
2	16/04/25, 04:02	Attempt finished Administrator SCeLE	Not answered	



Question **2**

Not answered

Marked out of 4.00

Buatlah method yang menerima dua array dan akan mengembalikan true jika salah satu array "mendominasi" array lainnya. Array A mendominasi Array B jika semua elemen dalam A lebih besar dari semua elemen yang ada di B begitu juga sebaliknya.

```
public static boolean isArrayDominant(int[] arr1, int[] arr2){}
```

contoh eksekusi method:

- `isArrayDominant(new int[]{1, 2, 4, 6}, new int[]{2, 3, 4});` → false
- `isArrayDominant(new int[]{1, 2, 4}, new int[]{4, 5, 6});` → true
- `isArrayDominant(new int[]{9, 10, 11}, new int[]{4, 5, 6});` → true
- `isArrayDominant(new int[]{1, 2, 4, 9}, new int[]{4, 5, 6});` → false

$\min A \geq \max B$ artinya semua bilangan di A lebih dari semua bilangan di B

$\min B \geq \max A$ artinya semua bilangan di B lebih dari semua bilangan di A

Acuan penilaian:

Poin 4 : Jika luaran program sudah tepat dan coverage dari test case = 100% pada method `isArrayDominant`

Poin 3: Jika luaran program sudah tepat dan coverage dari test case $\geq 80\%$ pada method `isArrayDominant`

Poin 2: Jika luaran program sudah tepat namun test case tidak tersedia atau ada kesalahan minor

Poin 1: Jika luaran program tidak tepat dan test case juga belum tersedia dengan benar

Poin 0: Compile Error

Response history

Step	Time	Action	State	Marks
1	15/04/25, 16:30	Started	Not yet answered	
2	16/04/25, 04:02	Attempt finished Administrator SCeLE	Not answered	

Question 3

Not answered

Marked out of 4.00

```
public class EmployeeBukanOO
{
    public static void main (String[] args)
    {
        String nama = "Ucup";
        int umur = 36;
        String posisi = "Manajer";
        int gaji = 15000000;
        System.out.println(gaji);
        System.out.println(naikGaji(gaji, 250000));
        System.out.println(kenaPenalti(gaji, 500000));
        System.out.println(panggilEmployee(nama, posisi, umur));
    }
    public static int naikGaji(int gaji, int kenaikan)
    {
        return gaji+kenaikan;
    }
    public static int kenaPenalti (int gaji, int penalti)
    {
        return gaji-penalti;
    }
    public static String panggilEmployee(String nama, String posisi, int umur)
    {
        return "Employee: "+nama+" posisi "+posisi+ " berumur "+umur;
    }
}
```

Code di atas masih dalam bentuk prosedural, di mana code ini menunjukkan *state* dan *behaviour* dari sebuah obyek Employee. Lakukan proses refactor dari code di atas sehingga mengikuti aturan *object oriented* yang tepat. Anda bisa menggunakan template yang diberikan. Anda diwajibkan membuat test case dan pastikan *coverage* >= 85%. Luaran yang diharapkan ketika class HaloEmployee (HaloEmployee.java) dijalankan:

15000000

15250000

14750000

Employee: ucup posisi manajer berumur 36

Berkas template: [Template](#)

Acuan penilaian:

Poin 4 : Jika luaran class HaloEmployee (HaloEmployee.java) sudah tepat dan coverage dari test case $\geq 85\%$

Poin 3: Jika luaran class HaloEmployee (HaloEmployee.java) sudah tepat namun coverage dari test case $< 85\%$

Poin 2: Jika luaran class HaloEmployee (HaloEmployee.java) sudah tepat namun test case tidak tersedia

Poin 1: Jika luaran class HaloEmployee (HaloEmployee.java) tidak tepat dan test case juga belum tersedia dengan benar

Response history

Step	Time	Action	State	Marks
1	15/04/25, 16:30	Started	Not yet answered	



Step	Time	Action	State	Marks
2	16/04/25, 04:02	Attempt finished Administrator SCeLE	Not answered	

Question 4

Not answered

Marked out of 4.00

Pada [Tugas Pemrograman 2](#), anda belum diperkenalkan menggunakan [inheritance](#). Namun terlihat sekali sebenarnya ada beberapa hal yang bisa diperbaiki dengan menerapkan [inheritance](#).

Pada soal kali ini, coba anda perhatikan, ada beberapa **field username-password** yang sama pada beberapa class dalam class diagram TP2 tersebut. Hanya memperhatikan isu tersebut, implementasikan ulang atau perbaiki atau tambahkan, *class-class* yang terkait dengan menerapkan [inheritance](#). Silahkan anda memperbaiki beberapa hal lain terkait [inheritance](#) yang bisa diperbaiki sekalian. Perhatikan waktu dalam pengerjaannya.

Anda dapat meng-copy-paste dari submission TP2 anda dan lakukan perubahan secara minimal.

Yang harus anda lakukan dan submit:

- Perubahan code agar menerapkan [inheritance](#) untukantisipasi permasalahan tersebut diatas.
- Sebuah berkas Readme.md berisi secara singkat tapi jelas:
 - Berkas-berkas apa saja yang diubah
 - Perubahan apa yang dilakukan permasing-masing berkas
- Compress dengan format .zip, dengan penamaan: ProgrammingInheritance_NPM_namaSingkat.zip
- Submit pada tempat yang disediakan.

Acuan Penilaian:

- Nilai 4: Penjelasan pada Readme.md singkat dan jelas; dan sempurna penerapan [inheritance](#); tidak ada duplikasi; dan Unit test berjalan baik.
- Nilai 3: Penjelasan pada Readme.md singkat dan jelas, dan program berjalan, tapi penerapan tidak sempurna, atau masih ada duplikasi atau unit test tidak berjalan baik.
- Nilai 2: Readme.md kurang jelas. Atau ada upaya menerapkan [inheritance](#) yang benar, namun belum selesai atau program tidak bisa dcompile.
- Nilai 1: Tidak ada Readme.md. Atau Ada perubahan tapi tidak significant, masih terlalu sedikit perubahannya, atau terlalu jauh dari jawaban yang diharapkan (misalnya masih kurang dari 50%)

Response history



Step	Time	Action	State	Marks
1	15/04/25, 16:30	Started	Not yet answered	
2	16/04/25, 04:02	Attempt finished Administrator SCeLE	Not answered	

[◀ Bagian Objective Test - UTS Sessi 2](#)

Jump to...

[Submission UTS Pemrograman 1 \(20%\) Sessi 2 ▶](#)

