

# **LAPORAN PRAKTIKUM – ANONYMOUS CLASS**

*Laporan ini disusun untuk memenuhi Tugas Ketujuh Praktikum*

*Pemrograman Berorientasi Objek*

**Dosen Pengampu : Dyah Sulistyowati R, S.Kom.,M.Kom**



Oleh :

Nama : Putri Wahyu Dwi Nurfatimah

NPM : 4524210080

Kelas : Praktikum Pemrograman Berorientasi Objek (G)

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS PANCASILA**

## 1. Soal

# Tugas 7 Praktikum Individu

Buat 1 projek Java program **aransemen lagu digital** yang memungkinkan sebuah lagu dimainkan oleh beberapa instrumen dengan gaya permainan yang berbeda-beda.  
Setiap instrumen (misalnya Piano, Gitar, Drum, Biola) memiliki gaya dan tempo unik, tergantung aransemen yang dipilih. Terapkan konsep Anonymous Class didalamnya

### Pengumpulan 1 zip (Program dan Laporan)

Di Laporan :

- Screenshoot code full layar
- Hasil RUN
- Beri Penjelasan
- Beri tanda pada bagian yang soal minta
- Kumpulkan **laporan** dan **program** pada drive : <https://bit.ly/Prak-PBO-G>
- Format penamaan file : Prak7\_PBO G\_NPM\_Nama

- Buat interface bernama Instrumen dengan method **void mainkan(String judulLagu);**
- Buat **class utama** bernama **AransemenLagu** yang memiliki:
- Atribut **String judulLagu**
- Method **aturAransemen()** untuk:
  - Membuat beberapa **objek Instrumen (minimal 3)** menggunakan **anonymous class**, misalnya:
    - Piano → melodi lembut dengan tempo 80 BPM
    - Gitar → ritme cepat dengan tempo 120 BPM
    - Drum → ketukan dinamis dengan tempo 100 BPM
  - Simpan objek-objek tersebut ke dalam **array atau ArrayList**.
  - Lalu tampilkan hasil pemanggilan **mainkan(judulLagu)** untuk setiap instrumen.
- Tambahkan fitur improvisasi:
- Buat **method** tambahan **improvisasi()** yang menerima parameter **Instrumen (anonymous class lain)** dan menjalankan **mainkan()** dengan gaya improvisasi baru.
- Misalnya: "Biola memainkan improvisasi dengan nada tinggi dan tempo 90 BPM"

## Jawaban:

### a. Source Code (AransemenLagu.java)

**Lokasi:** src/lagu/AransemenLagu.java

Kelas AransemenLagu.java adalah kelas utama yang digunakan untuk mengatur aransemen sebuah lagu dan instrumen-instrumen yang memainkannya. Kelas ini juga memanfaatkan konsep interface, anonymous class, dan ArrayList untuk menyimpan daftar instrumen yang digunakan.

```
package lagu;
import java.util.ArrayList;
- Bagian ini menunjukkan bahwa file ini berada dalam package bernama lagu, dan
  membutuhkan ArrayList untuk menyimpan banyak instrumen.

interface Instrumen
    void mainkan(String judulLagu);
end
- Interface ini mendefinisikan satu metode, yaitu mainkan().
  Tujuannya supaya semua instrumen yang dibuat nanti punya perilaku (method) yang
  sama, yaitu kemampuan memainkan lagu, tetapi dengan cara yang berbeda-beda sesuai
  instrumennya.

public class AransemenLagu
    private String judulLagu;
    private ArrayList<Instrumen> daftarInstrumen = new ArrayList<>();
- Di sini terdapat dua atribut penting:
    • judulLagu nama lagu yang akan dimainkan
    • daftarInstrumen list instrumen yang nanti akan memainkan lagu
  Keduanya dibuat private supaya hanya bisa diakses lewat method tertentu.

public AransemenLagu(String judulLagu)
    this.judulLagu = judulLagu;
end
```

- Constructor ini akan otomatis dijalankan saat objek AransemenLagu dibuat. Parameter judulLagu langsung disimpan ke atribut kelas.

```

Instrumen piano = new Instrumen()
    @Override
    public void mainkan(String judulLagu)
        System.out.println("Piano memainkan melodi lembut ...");
    end

```

end

- Metode ini membuat instrumen-instrumen awal untuk aransemen lagu. Instrumen dibuat menggunakan anonymous class, yaitu kelas tanpa nama yang langsung mengimplementasikan interface Instrumen. Begitu juga gitar dan drum. Setiap anonymous class punya isi method sendiri sesuai karakter instrumennya. Setelah itu instrumen dimasukkan ke ArrayList:  
 daftarInstrumen.add(piano);  
 daftarInstrumen.add(gitar);  
 daftarInstrumen.add(drum);

```

public void improvisasi(Instrumen instrumenImprovisasi)
    System.out.println("\n=== Bagian Improvisasi ===");
    instrumenImprovisasi.mainkan(judulLagu);

```

end

- Metode ini menerima instrumen tambahan biasanya juga anonymous class untuk memainkan improvisasi. Konsepnya fleksibel, karena kita bisa memasukkan instrumen apa pun sepanjang ia mengimplementasikan Instrumen.

```

public void tampilkanAransemen()
    System.out.println("=== Aransemen Lagu: " + judulLagu + " ===");
    for (Instrumen i : daftarInstrumen)
        i.mainkan(judulLagu);
    end

```

end

- Method ini menampilkan semua instrumen yang ada dalam aransemen, lalu masing-masing instrumen akan memainkan lagu sesuai gaya mereka.

```
AransemenLagu lagu = new AransemenLagu("Harmoni Senja");
```

- Ini adalah bagian program yang benar-benar dijalankan ketika program start. Membuat objek baru dengan judul lagu.  
 lagu.aturAransemen();  
 lagu.tampilkanAransemen();  
 Menyusun instrumen lalu menampilkan aransemen.

```

Instrumen improvisasiBiola = new Instrumen()
    @Override
    public void mainkan(String judulLagu) {
        System.out.println("Biola memainkan improvisasi ...");
    }
    end

```

end

- Sama seperti instrumen lainnya, tetapi ini spesifik untuk improvisasi. Kemudian dipanggil melalui:  
 lagu.improvisasi(improvisasiBiola);

- Secara keseluruhan program ini:

- Menggunakan interface untuk menyamakan perilaku instrumen
- Menggunakan anonymous class agar tidak perlu membuat class baru untuk tiap instrumen
- Menggunakan ArrayList untuk menyimpan instrumen yang memainkan aransemen
- Memiliki fitur improvisasi instrumen yang bisa ditambahkan kapan saja
- Menerapkan konsep OOP seperti polimorfisme, enkapsulasi, dan dynamic binding

```

1 package lagu;
2
3 import java.util.ArrayList;
4
5 // Interface Instrumen
6 interface Instrumen {
7     void mainkan(String judulLagu);
8 }
9
10 // Class Utama
11 public class AransemenLagu {
12     private String judulLagu;
13     private ArrayList<Instrumen> daftarInstrumen = new ArrayList<>();
14
15     public AransemenLagu(String judulLagu) {
16         this.judulLagu = judulLagu;
17     }
18
19     // Instrumen dengan Anonymous Class
20     public void aturAransemen() {
21
22         Instrumen piano = new Instrumen() {
23             @Override
24             public void mainkan(String judulLagu) {
25                 System.out.println("Piano memainkan melodi lembut dengan tempo 80 BPM untuk lagu \""
26                     + judulLagu + "\"");
27             }
28         };
29
30         Instrumen gitar = new Instrumen() {
31             @Override
32             public void mainkan(String judulLagu) {
33                 System.out.println("Gitar memainkan ritme cepat dengan tempo 120 BPM untuk lagu \""
34                     + judulLagu + "\"");
35             }
36         };
37
38         Instrumen drum = new Instrumen() {
39             @Override
40             public void mainkan(String judulLagu) {
41                 System.out.println("Drum memainkan ketukan dinamis dengan tempo 100 BPM untuk lagu \""
42                     + judulLagu + "\"");
43             }
44         };
45
46         // Masukkan ke ArrayList
47         daftarInstrumen.add(piano);
48         daftarInstrumen.add(gitar);
49         daftarInstrumen.add(drum);
50     }
51
52     // Improvisasi dengan anonymous class lain
53     public void improvisasi(Instrumen instrumenImprovisasi) {
54         System.out.println("\n== Bagian Improvisasi ==");
55         instrumenImprovisasi.mainkan(judulLagu);
56     }
57
58     // Menampilkan semua instrumen
59     public void tampilkanAransemen() {
60         System.out.println("== Aransemen Lagu: " + judulLagu + " ==");
61         for (Instrumen i : daftarInstrumen) {
62             i.mainkan(judulLagu);
63         }
64     }
65
66     // Main
67     public static void main(String[] args) {
68         AransemenLagu lagu = new AransemenLagu("Harmoni Senja");
69
70         lagu.aturAransemen();
71         lagu.tampilkanAransemen();
72
73         // Anonymous class improvisasi (contoh: Biola)
74         Instrumen improvisasiBiola = new Instrumen() {
75             @Override
76             public void mainkan(String judulLagu) {
77                 System.out.println("Biola memainkan improvisasi dengan nada tinggi dan tempo 90 BPM untuk lagu \""
78                     + judulLagu + "\"");
79             }
80         };
81
82         lagu.improvisasi(improvisasiBiola);
83     }
84 }
85

```

The screenshot shows an IDE with a project named 'AransementLaguDigital'. The terminal window displays the following commands and output:

```
src > lagu > AransementLaguJava > daftarinstrument
11 ~ public class AransementLagu {
}

PUTRI WAWU@LAPTOP-S8R0NDQ MING64 /C:/SEMPSTER III/PRO-G/TUGAS_PRAK/Prak07_PRO_G_4524210000_PUTRI WAWU DMT NURFATIMAH/AransementLaguDigital:
$ git init
Initialized empty Git repository in C:/SEMPSTER III/PRO-G/TUGAS_PRAK/Prak07_PRO_G_4524210000_PUTRI WAWU DMT NURFATIMAH/AransementLaguDigital/.git/
$ git add .
PUTRI WAWU@LAPTOP-S8R0NDQ MING64 /C:/SEMPSTER III/PRO-G/TUGAS_PRAK/Prak07_PRO_G_4524210000_PUTRI WAWU DMT NURFATIMAH/AransementLaguDigital: (master)
$
```

The screenshot shows the IDE with the following commands and output in the terminal:

```
PUTRI WAWU@LAPTOP-S8R0NDQ MING64 /C:/SEMPSTER III/PRO-G/TUGAS_PRAK/Prak07_PRO_G_4524210000_PUTRI WAWU DMT NURFATIMAH/AransementLaguDigital: (master)
$ git commit -m "first commit"
[master (root commit) f4d807] first commit
11 files changed, 114 insertions(+)
create mode 100644 src/daftarinstrument.json
create mode 100644 bin/app.class
create mode 100644 bin/lagu/AransementLagu01.class
create mode 100644 bin/lagu/AransementLagu02.class
create mode 100644 bin/lagu/AransementLagu03.class
create mode 100644 bin/lagu/AransementLagu04.class
create mode 100644 bin/lagu/AransementLagu05.class
create mode 100644 src/app.java
create mode 100644 src/lagu/AransementLagu.java

PUTRI WAWU@LAPTOP-S8R0NDQ MING64 /C:/SEMPSTER III/PRO-G/TUGAS_PRAK/Prak07_PRO_G_4524210000_PUTRI WAWU DMT NURFATIMAH/AransementLaguDigital: (master)
$ git branch -M main

PUTRI WAWU@LAPTOP-S8R0NDQ MING64 /C:/SEMPSTER III/PRO-G/TUGAS_PRAK/Prak07_PRO_G_4524210000_PUTRI WAWU DMT NURFATIMAH/AransementLaguDigital: (main)
$ git checkout -b Putri
Switched to a new branch "Putri"

PUTRI WAWU@LAPTOP-S8R0NDQ MING64 /C:/SEMPSTER III/PRO-G/TUGAS_PRAK/Prak07_PRO_G_4524210000_PUTRI WAWU DMT NURFATIMAH/AransementLaguDigital: (Putri)
$ git remote add origin https://github.com/pwawu07/pwawu07-Pro-G_4524210000_PUTRI WAWU DMT NURFATIMAH/AransementLaguDigital.git
PUTRI WAWU@LAPTOP-S8R0NDQ MING64 /C:/SEMPSTER III/PRO-G/TUGAS_PRAK/Prak07_PRO_G_4524210000_PUTRI WAWU DMT NURFATIMAH/AransementLaguDigital: (Putri)
$
```

The screenshot shows the IDE with the following commands and output in the terminal:

```
PUTRI WAWU@LAPTOP-S8R0NDQ MING64 /C:/SEMPSTER III/PRO-G/TUGAS_PRAK/Prak07_PRO_G_4524210000_PUTRI WAWU DMT NURFATIMAH/AransementLaguDigital: (Putri)
$ git push -u origin Putri
bash: git push -u origin Putri: Is a directory

PUTRI WAWU@LAPTOP-S8R0NDQ MING64 /C:/SEMPSTER III/PRO-G/TUGAS_PRAK/Prak07_PRO_G_4524210000_PUTRI WAWU DMT NURFATIMAH/AransementLaguDigital: (main)
$ git push -u origin Putri
Enumerating objects: 18, done.
Counting objects: 100% (18/18), done.
Delta compression using up to 12 threads
Compressing objects: 100% (16/16), done.
Writing objects: 100% (18/18), 5.02 KiB | 1.25 MiB/s, done.
Total 18 (delta 3), reused 0 (delta 0), packed-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), done.
PUTRI WAWU@LAPTOP-S8R0NDQ MING64 /C:/SEMPSTER III/PRO-G/TUGAS_PRAK/Prak07_PRO_G_4524210000_PUTRI WAWU DMT NURFATIMAH/AransementLaguDigital: (main)
$ -git pull origin main
bash: -git: command not found
$ git merge Putri
Already up to date.

PUTRI WAWU@LAPTOP-S8R0NDQ MING64 /C:/SEMPSTER III/PRO-G/TUGAS_PRAK/Prak07_PRO_G_4524210000_PUTRI WAWU DMT NURFATIMAH/AransementLaguDigital: (main)
$ git push origin main
To https://github.com/pwawu07/pwawu07-Pro-G_4524210000_PUTRI WAWU DMT NURFATIMAH/AransementLaguDigital.git
! [rejected] main -> main (fetch first)
error: failed to push some refs to 'https://github.com/pwawu07/pwawu07-Pro-G_4524210000_PUTRI WAWU DMT NURFATIMAH/AransementLaguDigital.git'
hint: updates were rejected because the remote contains work that you do not
hint: have locally. This is usually caused by another repository pushing to
hint: the same ref; if you want to integrate the remote changes, use
hint: 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push -help' for details.
PUTRI WAWU@LAPTOP-S8R0NDQ MING64 /C:/SEMPSTER III/PRO-G/TUGAS_PRAK/Prak07_PRO_G_4524210000_PUTRI WAWU DMT NURFATIMAH/AransementLaguDigital: (main)
$
```

```

src> lagu > ArasemenLagu.java > ArasemenLagu > dafarInstrument
11 public class ArasemenLagu {

$ git push -u origin Putri
Already up to date.

PUTRI WAWU@LAPTOP-380XNDP MINGGA /C:/SEPESTER III/PRO-G/TUGAS_PRAK/Prak07_PBO_G_4524210080_PUTRI WAWU DME NURFATIMAH/ArasemenLagu01git (main)
$ git push origin main
to https://github.com/pwahyudwinurfatimah-creator/Tugas07_Prak_PBO_G.git
! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'https://github.com/pwahyudwinurfatimah-creator/Tugas07_Prak_PBO_G.git'
hint: updates were rejected because the remote contains work that you do not
hint: have locally. This is usually caused by another repository pushing to
hint: the same ref. If you want to integrate the remote changes, use
hint: 'git pull' before pushing again.
hint: see the 'Note about fast-forwards' in 'git push --help' for details.

PUTRI WAWU@LAPTOP-380XNDP MINGGA /C:/SEPESTER III/PRO-G/TUGAS_PRAK/Prak07_PBO_G_4524210080_PUTRI WAWU DME NURFATIMAH/ArasemenLagu01git (main)
$ [no-git pull origin main --allow-unrelated-histories
bash: [no-git: command not found

PUTRI WAWU@LAPTOP-380XNDP MINGGA /C:/SEPESTER III/PRO-G/TUGAS_PRAK/Prak07_PBO_G_4524210080_PUTRI WAWU DME NURFATIMAH/ArasemenLagu01git (main)
$ git pull origin main --allow-unrelated-histories
bash: git: command not found

PUTRI WAWU@LAPTOP-380XNDP MINGGA /C:/SEPESTER III/PRO-G/TUGAS_PRAK/Prak07_PBO_G_4524210080_PUTRI WAWU DME NURFATIMAH/ArasemenLagu01git (main)
$ git pull origin main --allow-unrelated-histories
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack reused 0 (from 0)
unpacking objects: 100% (3/3), 880 bytes | 80.00 KiB/s, done.
from https://github.com/pwahyudwinurfatimah-creator/Tugas07_Prak_PBO_G
* branch                main       -> FETCH HEAD
* [new branch]          main       -> origin/main
Auto-merging README.md
CONFLICT (add/add): merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.

PUTRI WAWU@LAPTOP-380XNDP MINGGA /C:/SEPESTER III/PRO-G/TUGAS_PRAK/Prak07_PBO_G_4524210080_PUTRI WAWU DME NURFATIMAH/ArasemenLagu01git (main/PUSING)
$

```

## b. SS Hasil Running Program

```

src> lagu > ArasemenLagu.java > ArasemenLagu > dafarInstrument
11 public class ArasemenLagu {

PS C:\SEPESTER III\PRO-G\TUGAS_PRAK\Prak07_PBO_G_4524210080_PUTRI WAWU DME NURFATIMAH\ArasemenLagu01git> "C:\Program Files\Java\jdk-24\bin\java.exe" "-Xms128M" "-Xmx512M" "-Duser.dir=C:\SEPESTER III\PRO-G\TUGAS_PRAK\Prak07_PBO_G_4524210080_PUTRI WAWU DME NURFATIMAH\ArasemenLagu01git\bin" "lagu.ArasemenLagu"

Piano memainkan melodi lembut dengan tempo 80 BPM untuk lagu "harmoni Senja"
Gitar memainkan ritme cepat dengan tempo 120 BPM untuk lagu "harmoni Senja"
Drum memainkan ketukan dinamis dengan tempo 100 BPM untuk lagu "harmoni Senja"

--- Bagian Improvisasi ---
Biola memainkan improvisasi dengan nada tinggi dan tempo 90 BPM untuk lagu "harmoni Senja"
PS C:\SEPESTER III\PRO-G\TUGAS_PRAK\Prak07_PBO_G_4524210080_PUTRI WAWU DME NURFATIMAH\ArasemenLagu01git>

```

## c. LINK REPOSITORY

[https://github.com/pwahyudwinurfatimah-creator/Tugas07\\_Prak\\_PBO\\_G.git](https://github.com/pwahyudwinurfatimah-creator/Tugas07_Prak_PBO_G.git)