

LAPORAN PRAKTIKUM – ANONYMOUS CLASS

Laporan ini disusun untuk memenuhi Tugass Ketujuh Praktikum

Pemprograman 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

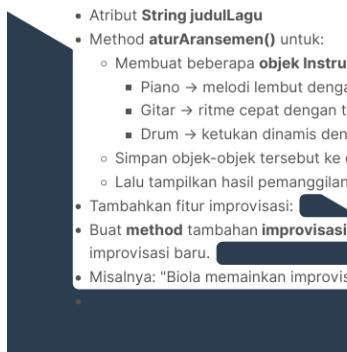
Individu



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

- 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: [REDACTED]
- Buat **method** tambahan **improvisasi()** yang **menerima parameter Instrumen (anonymous class lain)** dan **menjalankan mainkan()** dengan gaya improvisasi baru. [REDACTED]
- Misalnya: "Biola memainkan improvisasi dengan nada tinggi dan tempo 90 BPM"



Pengumpulan 1 zip (Program dan Laporan)

Di Laporan :

- Screenshot 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

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 AransemencoLagu {
12 private String judulLagu;
13 private ArrayList<Instrumen> daftarInstrumen = new ArrayList<>();
14
15 public AransemencoLagu(String judulLagu) {
16 this.judulLagu = judulLagu;
17 }
18
19 // Instrumen dengan Anonymous Class
20 public void aturAransemenco() {
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 tampilanAransemenco() {
60 System.out.println("==== Aрансемено Lagu: " + judulLagu + " ===");
61 for (Instrumen i : daftarInstrumen) {
62 i.mainkan(judulLagu);
63 }
64 }
65
66 // Main
67 public static void main(String[] args) {
68 AransemencoLagu lagu = new AransemencoLagu("Harmoni Senja");
69
70 lagu.aturAransemenco();
71 lagu.tampilanAransemenco();
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 the VS Code interface with the following details:

- File Explorer:** Shows the project structure with a folder named "ARASEMENLAGUDIGITAL".
- Terminal:** Displays the command-line session:

```
PUTRI WAWU@APTOP-3808050 MINGW64 /c/SEMESTER III/PRO-G/TUGAS_PRAK/Prak07_Pro_G_4524210080 PUTRI WAWU DNE NURFATIMAH/arasemenlagudigital
$ git init
Initialized empty Git repository in C:/SEMESTER III/PRO-G/TUGAS_PRAK/Prak07_Pro_G_4524210080/.git/
PUTRI WAWU@APTOP-3808050 MINGW64 /c/SEMESTER III/PRO-G/TUGAS_PRAK/Prak07_Pro_G_4524210080 PUTRI WAWU DNE NURFATIMAH/arasemenlagudigital (master)
$
```
- Bottom Status Bar:** Shows system information like battery level (28%), date (17/11/2025), and time (22:24).

The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows the project structure with a folder named "ARASEMENLAGUDIGITAL".
- Terminal:** Displays the command-line session:

```
PUTRI WAWU@APTOP-3808050 MINGW64 /c/SEMESTER III/PRO-G/TUGAS_PRAK/Prak07_Pro_G_4524210080 PUTRI WAWU DNE NURFATIMAH/arasemenlagudigital (master)
$ git commit -m "First commit"
[ArasemenLagu.java 1 file changed, 114 insertions(+)]
create mode 100644 .vscode/settings.json
create mode 100644 README.md
create mode 100644 bin/lagu/ArasemenLagu.class
create mode 100644 bin/lagu/ArasemenLagu1.class
create mode 100644 bin/lagu/ArasemenLagu2.class
create mode 100644 bin/lagu/ArasemenLagu3.class
create mode 100644 bin/lagu/ArasemenLagu4.class
create mode 100644 bin/lagu/ArasemenLagu5.class
create mode 100644 bin/lagu/Instrument.class
create mode 100644 src/App.java
create mode 100644 src/lagu/ArasemenLagu.java
PUTRI WAWU@APTOP-3808050 MINGW64 /c/SEMESTER III/PRO-G/TUGAS_PRAK/Prak07_Pro_G_4524210080 PUTRI WAWU DNE NURFATIMAH/arasemenlagudigital (main)
$ git checkout -b Putri
Switched to a new branch 'Putri'

PUTRI WAWU@APTOP-3808050 MINGW64 /c/SEMESTER III/PRO-G/TUGAS_PRAK/Prak07_Pro_G_4524210080 PUTRI WAWU DNE NURFATIMAH/arasemenlagudigital (Putri)
$ git remote add origin https://github.com/pawiyudinefatimah/creator/tugas07_Prak_Pro_G.git
PUTRI WAWU@APTOP-3808050 MINGW64 /c/SEMESTER III/PRO-G/TUGAS_PRAK/Prak07_Pro_G_4524210080 PUTRI WAWU DNE NURFATIMAH/arasemenlagudigital (Putri)
```
- Bottom Status Bar:** Shows system information like battery level (28%), date (17/11/2025), and time (22:29).

The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows the project structure with a folder named "ARASEMENLAGUDIGITAL".
- Terminal:** Displays the command-line session:

```
PUTRI WAWU@APTOP-3808050 MINGW64 /c/SEMESTER III/PRO-G/TUGAS_PRAK/Prak07_Pro_G_4524210080 PUTRI WAWU DNE NURFATIMAH/arasemenlagudigital
$ git push -u origin Putri
fatal: 'Putri' is a directory.
PUTRI WAWU@APTOP-3808050 MINGW64 /c/SEMESTER III/PRO-G/TUGAS_PRAK/Prak07_Pro_G_4524210080 PUTRI WAWU DNE NURFATIMAH/arasemenlagudigital (Putri)
$ git push -u origin main
fatal: 'main' is a directory.
PUTRI WAWU@APTOP-3808050 MINGW64 /c/SEMESTER III/PRO-G/TUGAS_PRAK/Prak07_Pro_G_4524210080 PUTRI WAWU DNE NURFATIMAH/arasemenlagudigital (main)
$ git pull origin main
fatal: 'origin' is not a valid option.
PUTRI WAWU@APTOP-3808050 MINGW64 /c/SEMESTER III/PRO-G/TUGAS_PRAK/Prak07_Pro_G_4524210080 PUTRI WAWU DNE NURFATIMAH/arasemenlagudigital (main)
$ git fetch origin
error: failed to push some refs to 'https://github.com/pawiyudinefatimah/creator/tugas07_Prak_Pro_G.git'
hint: Updates were rejected because they contain 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.
```
- Bottom Status Bar:** Shows system information like battery level (28%), date (17/11/2025), and time (22:35).

```
File Edit Selection View Go Run Terminal Help ← → 🔍 ArasementLaguDigital

EXPLORER ... ArasementLaguJava x
src > lagu > ArasementLaguJava > ArasementLagu > dataInstrumen
 +-- mfile.class ArasementLagu.java
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
$ git push -u origin Putri
Already up to date.

PUTRI WANIYU/APTOR:30RDVNOH MINGW4 /c/SEMESTER III/PRO-G/TUGAS_PRAK/Prak07_Pro_G_4524210080_PUTRI_WANIYU DM NURFATINAH/ArasementLaguDigital (main)
$ git push origin main
To https://github.com/padyuhdinurfatinah-creator/tugasprak_prak06_g.git
 error: failed to push some refs to 'https://github.com/padyuhdinurfatinah-creator/tugasprak_prak06_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 WANIYU/APTOR:30RDVNOH MINGW4 /c/SEMESTER III/PRO-G/TUGAS_PRAK/Prak07_Pro_G_4524210080_PUTRI_WANIYU DM NURFATINAH/ArasementLaguDigital (main)
$ 1000+git pull origin main --allow-unrelated-histories
bash: 1000+git: command not found

PUTRI WANIYU/APTOR:30RDVNOH MINGW4 /c/SEMESTER III/PRO-G/TUGAS_PRAK/Prak07_Pro_G_4524210080_PUTRI_WANIYU DM NURFATINAH/ArasementLaguDigital (main)
$ 5+git pull origin main --allow-unrelated-histories
bash: 5+git: command not found

PUTRI WANIYU/APTOR:30RDVNOH MINGW4 /c/SEMESTER III/PRO-G/TUGAS_PRAK/Prak07_Pro_G_4524210080_PUTRI_WANIYU DM NURFATINAH/ArasementLaguDigital (main)
$ 5+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)
remote: Unpacking objects: 100% (3/3), done.
From https://github.com/padyuhdinurfatinah-creator/tugasprak_prak06_g
 * [new 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 WANIYU/APTOR:30RDVNOH MINGW4 /c/SEMESTER III/PRO-G/TUGAS_PRAK/Prak07_Pro_G_4524210080_PUTRI_WANIYU DM NURFATINAH/ArasementLaguDigital (main/MERGEING)
$ 1cm hujan
Selasa
```

b. SS Hasil Running Program

### c. LINK REPOSITORY

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