LAPORAN AWAL

LAPORAN PEMROGRAMAN I

LAPORAN KE-2



Disusun Oleh :

Nama :Febiyanto Rizki Qurbandi

NIM :231011450284

Kelas :04TPLP003

TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS PAMULANG

Jl. Surya Kencana No. 1 Pamulang Telp (021)7412566, Fax. (021)7412566

Tangerang Selatan – Banten

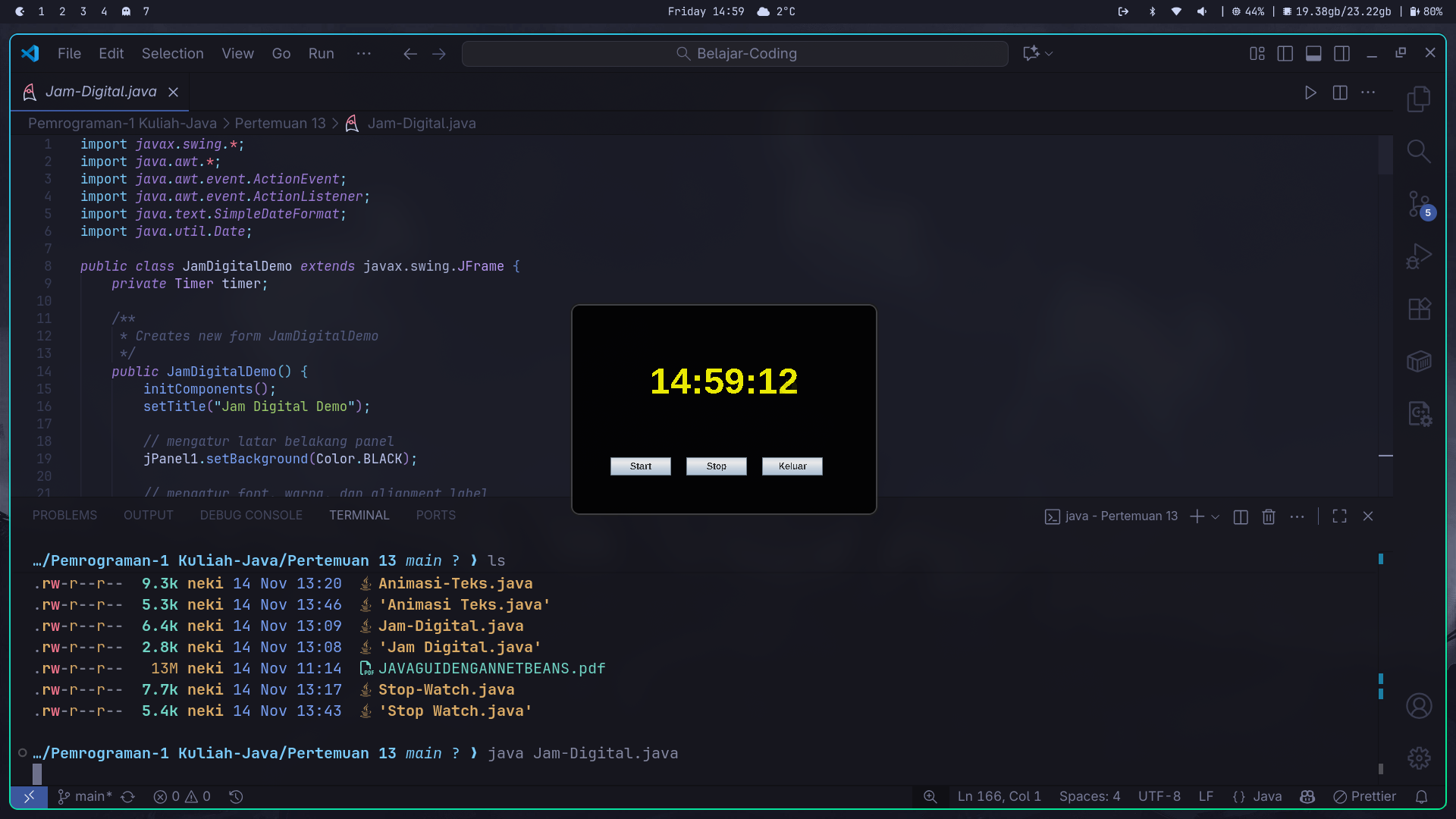
Link GITHUB Repo:

[https://github.com/NeiaKI/Belajar-Coding/tree/main/Pemrograman-1%20Kuliah-Java/Pertemuan%2013](https://github.com/NeiaKI/Belajar-Coding/tree/main/Pemrograman-1 Kuliah-Java/Pertemuan 13)

Code JAM DIGITAL



OUTPUT:



PENJELASAN:

* GUI Java Swing: Kode ini mendefinisikan sebuah kelas TampilanTimer yang merupakan turunan dari JFrame, digunakan untuk membuat jendela aplikasi berbasis grafis.
* Objek Timer: Di dalam konstruktor, sebuah objek Timer dibuat dan diatur dengan interval 1000 milidetik (1 detik).
* Logika Counter: Setiap kali timer berdetak (per detik), program akan:
* Menambah variabel counter sebesar 1.
* Memperbarui teks pada label (jLabel1) untuk menampilkan nilai counter terbaru ("Counter : [angka]").
* Tombol Mulai: Terdapat event handler jButton1ActionPerformed. Saat tombol diklik, perintah timer.start() dijalankan untuk memulai proses penghitungan waktu.
* Pengaturan Layout: Metode initComponents() (kode yang digenerate otomatis oleh IDE seperti NetBeans) bertugas mengatur posisi tombol, label, ukuran jendela, dan tata letak (layout) aplikasi.
* Metode Main: Titik awal program yang membuat dan menampilkan jendela aplikasi TampilanTimer agar terlihat di layar.
* Format Waktu: Waktu mentah diformat menjadi pola "HH:mm:ss" (Jam:Menit:Detik) menggunakan kelas SimpleDateFormat sebelum ditampilkan ke label.
* Fungsi Tombol:
* Start: Memanggil timer.start() untuk mulai menjalankan jam.
* Stop: Memanggil timer.stop() untuk membekukan waktu (jeda).
* Keluar: Memanggil System.exit(0) untuk menutup program sepenuhnya.

Code STOP WATCH:

import *javax.swing.\**;

import *java.awt.\**;

import *java.awt.event.ActionEvent*;

import *java.awt.event.ActionListener*;

import *java.util.concurrent.TimeUnit*;

*public* *class* StopWatchDemo *extends* javax.swing.JFrame {

*private* *static* *final* String Start = "Start";

*private* *static* *final* String Pause = "Pause";

*private* *static* *final* String Reset = "Reset";

*private* boolean isRunning;

*private* Timer timer = new Timer(100, new WaktuListener());

*private* long initTime = System.currentTimeMillis();

*private* long startTime;

*private* long pauseTime;

*/\*\**

*\* Creates new form StopWatchDemo*

*\*/*

*public* StopWatchDemo() {

initComponents();

setTitle("Stop Watch Demo");

jPanel1.setBackground(Color.BLACK);

btnReset.setEnabled(false);

setWaktu();

}

*/\*\**

*\* Mengatur format waktu dan tampilan label*

*\*/*

*private* void setWaktu() {

lblWaktu.setFont(new Font("Arial", Font.BOLD, 48));

lblWaktu.setForeground(Color.YELLOW);

lblWaktu.setText("00:00:00.0"); *// Mulai dari 00:00:00.0*

lblWaktu.setHorizontalAlignment(JLabel.CENTER);

}

*/\*\**

*\* Memulai stop watch*

*\*/*

*public* void start() {

if (isRunning == false) {

startTime = System.currentTimeMillis();

} else {

startTime = System.currentTimeMillis() - (pauseTime - startTime);

}

isRunning = true;

timer.start();

}

*/\*\**

*\* Menjeda stop watch*

*\*/*

*public* void pause() {

pauseTime = System.currentTimeMillis();

timer.stop();

}

*/\*\**

*\* Mereset stop watch*

*\*/*

*public* void reset() {

startTime = 0;

isRunning = false;

timer.stop();

lblWaktu.setText("00:00:00.0"); *// Reset ke 00:00:00.0*

btnStart.setText(Start);

}

*/\*\**

*\* Memformat waktu menjadi format HH:MM:SS.ms*

*\*/*

*private* String formatWaktu(*final* long time) {

*final* long hr = TimeUnit.MILLISECONDS.toHours(time);

*final* long min = TimeUnit.MILLISECONDS.toMinutes(time -

TimeUnit.HOURS.toMillis(hr));

*final* long sec = TimeUnit.MILLISECONDS.toSeconds(time -

TimeUnit.HOURS.toMillis(hr) -

TimeUnit.MINUTES.toMillis(min));

*final* long ms = TimeUnit.MILLISECONDS.toMillis(time -

TimeUnit.HOURS.toMillis(hr) -

TimeUnit.MINUTES.toMillis(min) -

TimeUnit.SECONDS.toMillis(sec));

return String.format("%02d:%02d:%02d.%01d", hr, min, sec, ms/100);

}

*/\*\**

*\* Mendapatkan waktu saat ini*

*\*/*

*private* String getCurrentTime(long time) {

return formatWaktu(time);

}

*/\*\**

*\* Inner class untuk mendengarkan event Timer*

*\*/*

*class* WaktuListener *implements* ActionListener {

*public* void actionPerformed(ActionEvent e) {

lblWaktu.setText(getCurrentTime(System.currentTimeMillis() - startTime));

}

}

*/\*\**

*\* This method is called from within the constructor to initialize the form.*

*\*/*

*private* void initComponents() {

jPanel1 = new javax.swing.JPanel();

lblWaktu = new javax.swing.JLabel();

btnReset = new javax.swing.JButton();

btnStart = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.JFrame.EXIT\_ON\_CLOSE);

setResizable(false);

jPanel1.setBackground(new java.awt.Color(0, 0, 0));

lblWaktu.setFont(new java.awt.Font("Arial", Font.BOLD, 48));

lblWaktu.setForeground(new java.awt.Color(255, 255, 0));

lblWaktu.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

lblWaktu.setText("00:00:00.0");

btnStart.setText("Start");

btnStart.addActionListener(new java.awt.event.ActionListener() {

*public* void actionPerformed(java.awt.event.ActionEvent evt) {

btnStartActionPerformed(evt);

}

});

btnReset.setText("Reset");

btnReset.addActionListener(new java.awt.event.ActionListener() {

*public* void actionPerformed(java.awt.event.ActionEvent evt) {

btnResetActionPerformed(evt);

}

});

*// layout menggunakan GroupLayout*

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(lblWaktu, javax.swing.GroupLayout.DEFAULT\_SIZE, 400, Short.MAX\_VALUE)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(100, 100, 100)

.addComponent(btnStart, javax.swing.GroupLayout.PREFERRED\_SIZE, 80, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(50, 50, 50)

.addComponent(btnReset, javax.swing.GroupLayout.PREFERRED\_SIZE, 80, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(90, Short.MAX\_VALUE))

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(50, 50, 50)

.addComponent(lblWaktu, javax.swing.GroupLayout.PREFERRED\_SIZE, 100, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(50, 50, 50)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(btnStart)

.addComponent(btnReset))

.addContainerGap(50, Short.MAX\_VALUE))

);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

);

pack();

setLocationRelativeTo(null);

}

*/\*\**

*\* Event handler untuk tombol Start/Pause*

*\*/*

*private* void btnStartActionPerformed(java.awt.event.ActionEvent evt) {

if (btnStart.getText().equals(Start)) {

btnStart.setText(Pause);

btnReset.setEnabled(false);

start();

} else {

btnStart.setText(Start);

btnReset.setEnabled(true);

pause();

}

}

*/\*\**

*\* Event handler untuk tombol Reset*

*\*/*

*private* void btnResetActionPerformed(java.awt.event.ActionEvent evt) {

reset();

}

*/\*\**

*\* Main method untuk menjalankan aplikasi*

*\*/*

*public* *static* void main(String args[]) {

java.awt.EventQueue.invokeLater(new Runnable() {

*public* void run() {

new StopWatchDemo().setVisible(true);

}

});

}

*// Deklarasi komponen*

*private* javax.swing.JPanel jPanel1;

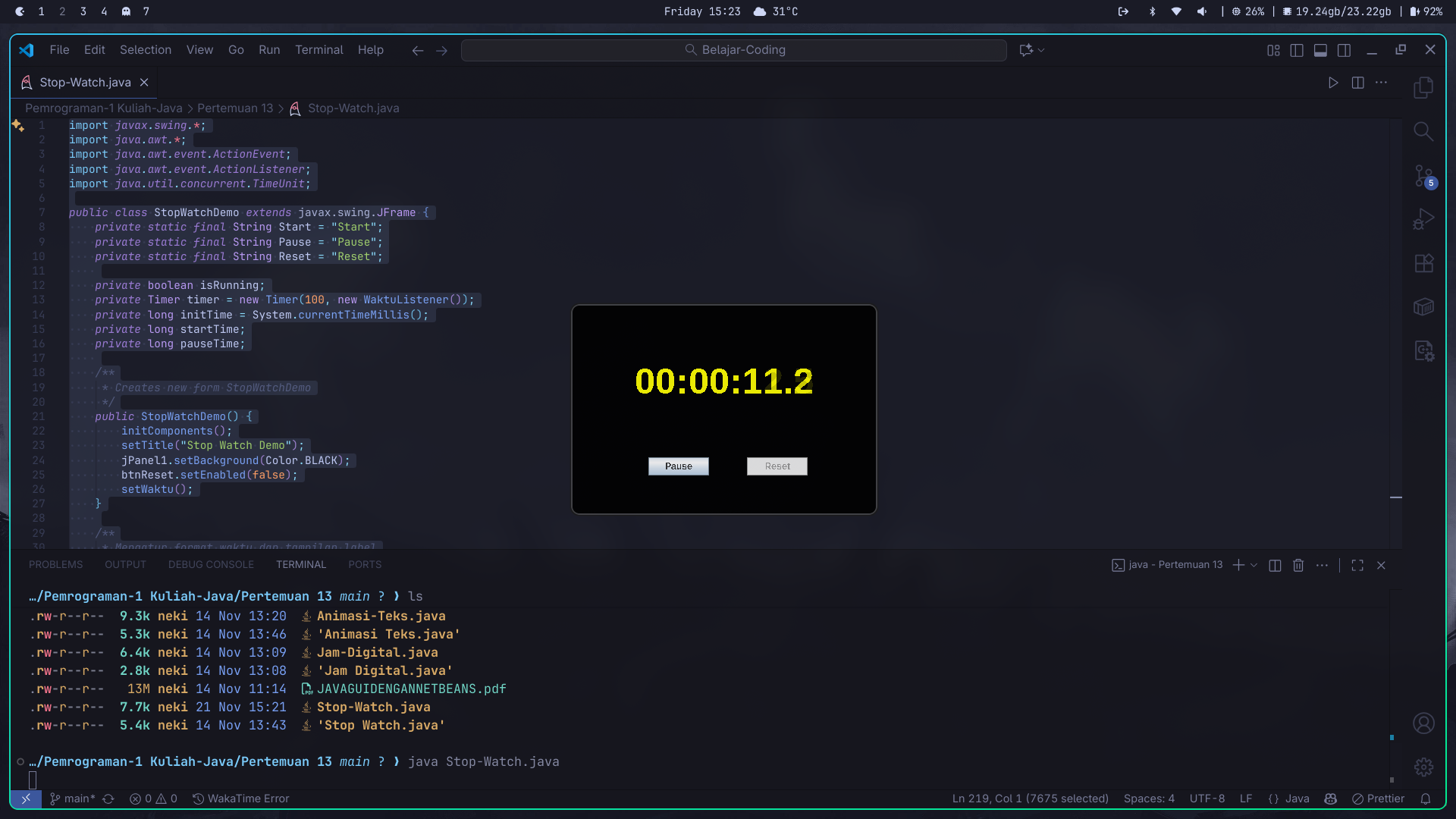
*private* javax.swing.JLabel lblWaktu;

*private* javax.swing.JButton btnReset;

*private* javax.swing.JButton btnStart;

}

Output :



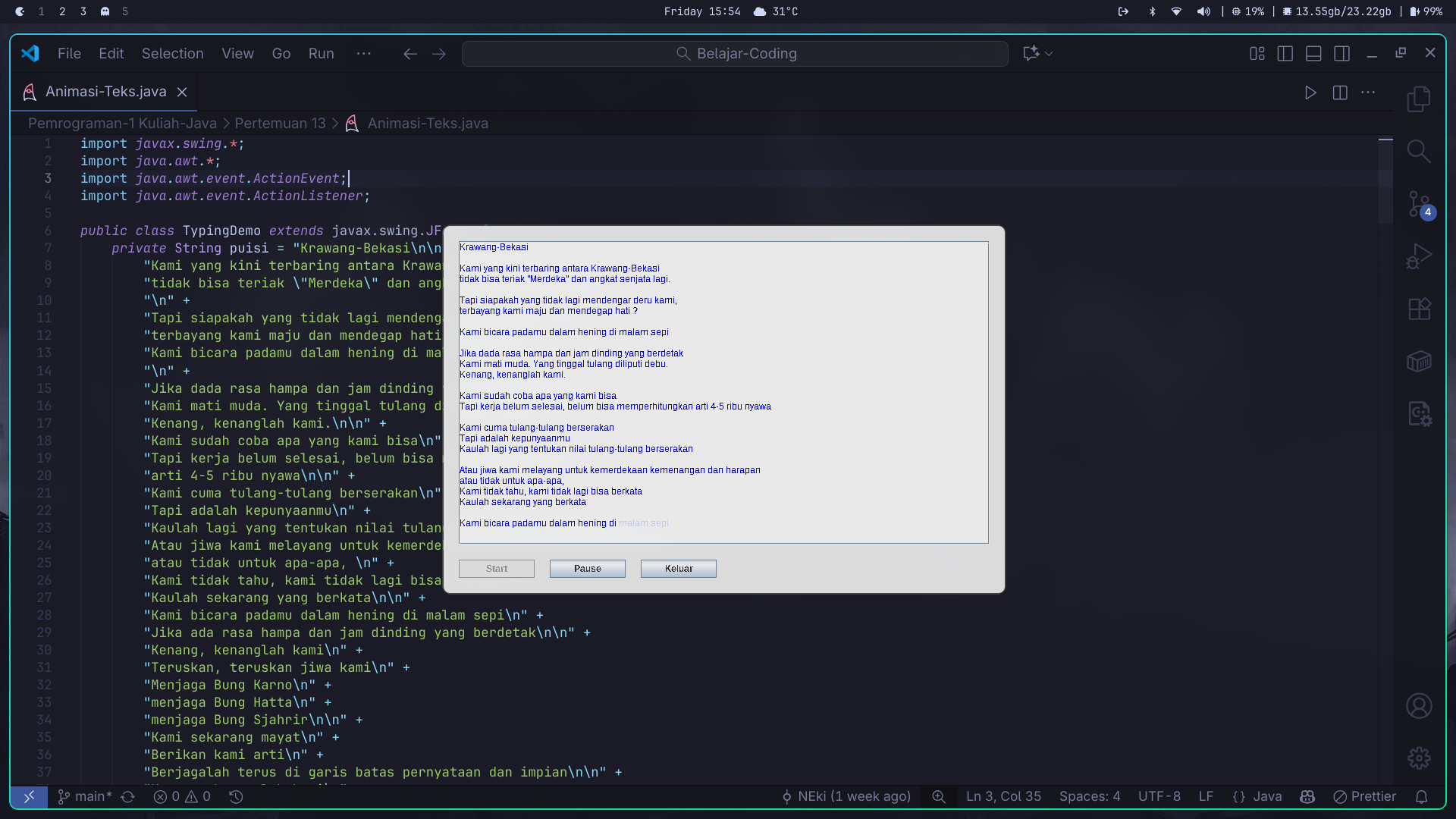
Penjelasan:

* Tujuan Program: Membuat aplikasi Stopwatch yang memiliki fungsi Start (Mulai), Pause (Jeda), dan Reset.
* Logika Perhitungan Waktu (Time Delta): Berbeda dengan program counter sebelumnya yang hanya menambah angka (+1), kode ini menghitung selisih waktu nyata.
* Rumusnya: Waktu Sekarang - Waktu Mulai.
* Ini membuat perhitungan waktu tetap akurat sesuai jam sistem komputer.
* Logika Tombol Toggle (Start/Pause): Tombol btnStart memiliki fungsi ganda:
* Jika tulisan "Start": Timer berjalan, tombol berubah jadi "Pause", tombol Reset dimatikan (disable).
* Jika tulisan "Pause": Timer berhenti, tombol berubah jadi "Start", tombol Reset diaktifkan (enable).
* Mekanisme Pause yang Cerdas:
* Saat tombol "Start" ditekan kembali setelah Pause, program tidak memulai dari nol.
* Program menggunakan rumus startTime = System.currentTimeMillis() - (pauseTime - startTime) untuk memastikan waktu berlanjut mulus dari angka terakhir, seolah-olah waktu "berhenti" saat dipause.
* Format Tampilan (TimeUnit):
* Menggunakan TimeUnit untuk mengonversi ribuan milidetik menjadi format Jam, Menit, Detik, dan Milidetik.
* Hasilnya diformat string menjadi 00:00:00.0.
* Timer UI: Objek Timer diset dengan interval 100ms (0.1 detik) hanya untuk memperbarui tampilan label agar terlihat berjalan mulus, bukan untuk menghitung logika waktunya.

Code:



Output:



Penjelasan:

* **Tujuan Program:** Membuat efek "mesin ketik" (typing animation) di mana teks puisi muncul karakter demi karakter secara otomatis.
* **Data Teks:** Sebuah variabel String puisi menampung seluruh teks puisi "Karawang-Bekasi" karya Chairil Anwar yang akan ditampilkan.
* **Variabel Index:** int charIndex digunakan sebagai penanda posisi karakter yang sedang ditampilkan saat ini (dimulai dari 0).
* **Mekanisme Timer:**
  + Objek Timer diset dengan interval **100 ms**.
  + Setiap detak timer, program mengambil satu karakter dari string puisi berdasarkan charIndex.
  + Karakter tersebut ditambahkan ke JTextArea (txaPuisi.setText(...)).
  + charIndex ditambah 1 (charIndex++) untuk persiapan karakter selanjutnya.
* **Kondisi Berhenti:** Jika charIndex sudah mencapai panjang total string puisi (puisi.length()), timer dimatikan (timer.stop()) dan tombol Start diaktifkan kembali.
* **Logika Tombol:**
  + **Start:** Memulai timer. Jika puisi sudah selesai, teks direset kosong dan indeks dikembalikan ke 0 untuk mulai ulang.
  + **Pause/Continue:** Tombol ini berfungsi sebagai *toggle*.
    - Jika diklik saat berjalan ("Pause"), timer dihentikan (stop).
    - Jika diklik saat berhenti ("Continue"), timer dilanjutkan (start).