## Soal Praktikum

1. Dalam mengelola file, apa saja yang bisa di lakukan oleh Java Files? [Wajib]

## Soal Praktikum

2. Buatlah program Java File untuk membuat file ‘Biodata.txt’! [Wajib]

## Source Code

*import* java.io.File;  
*import* java.io.FileWriter;  
*import* java.io.IOException;  
  
*//TIP To <b>Run</b> code, press <shortcut actionId="Run"/> or  
// click the <icon src="AllIcons.Actions.Execute"/> icon in the gutter.  
class* fileCreator{  
 *public void* creatFile(){  
 String directoryPath = "D:/ngoding adam/Java/Pratikum 5/pertemuan5\_07720";  
 String fileName = "Biodata.txt";  
 String fileContent = "Membuat file menggunakan java";  
  
 *try* {  
 File file = *new* File(directoryPath, fileName);  
 *if*(file.createNewFile()){  
 System.out.println("File" + fileName + " Berhasil dibuat di folder" + directoryPath);  
 *try* (FileWriter writer = *new* FileWriter(file)) {  
 writer.write(fileContent);  
 System.out.println("File Berhasil diisi.");  
 }  
 }*else* {  
 System.out.println("File sudah ada: " + fileName );  
 }  
 } *catch* (IOException e) {  
 System.out.println("Terjadi kesalahan saat membuat file");  
 e.printStackTrace();  
 }  
 }  
}  
  
  
*public class* Main {  
 *public static void* main(String[] args) {  
 fileCreator FileCreator = *new* fileCreator();  
 FileCreator.creatFile();  
 }  
}

**Penjelasan**

Program Java ini membuat file teks baru dengan nama Biodata.txt di direktori yang ditentukan (D:/ngoding adam/Java/Pratikum 5/pertemuan5\_07720). Program pertama-tama memeriksa apakah file tersebut sudah ada di lokasi yang ditentukan. Jika file belum ada, maka file baru akan dibuat, dan konten berupa string "Membuat file menggunakan java" akan ditulis ke dalam file menggunakan FileWriter. Jika file sudah ada, program akan memberi tahu pengguna bahwa file tersebut sudah ada. Program juga menangani kesalahan dengan blok try-catch, yang akan mencetak pesan kesalahan jika terjadi masalah dalam pembuatan atau penulisan file. Program ini dijalankan dari kelas Main yang memanggil metode creatFile() dari kelas fileCreator.

**Output**

|  |
| --- |
|  |

## Soal Praktikum

3. Jelaskan apa itu Jframe pada Component Java Swing serta berikan

contohnya penggunaannya dengan ukuran frame 400 x 400 [Wajib]

## Jawaban

Jframe adalah bingkai utama dalam aplikasi Swing. Ini mewakili jendela yang dapat diubah ukurannya dan berfungsi sebagai wadah untuk menamoung komponen2 lain

## Source Code

*package* soalNo3;  
  
  
*import* javax.swing.\*;  
  
*public class* main {  
 *public static void* main(String[] args) {  
 JFrame frame = *new* JFrame();  
 frame.setSize(400,400);  
  
 frame.setVisible(*true*);  
  
 }  
}

**Penjelasan**

Program Java ini membuat sebuah jendela GUI (Graphical User Interface) menggunakan kelas JFrame dari paket javax.swing. Dalam program ini, kelas main berfungsi sebagai titik masuk (entry point). Di dalam metode main, objek JFrame bernama frame dibuat, yang kemudian ukurannya diatur menjadi 400x400 piksel dengan menggunakan frame.setSize(400, 400). Setelah itu, frame.setVisible(true) dipanggil untuk menampilkan jendela tersebut. Dengan demikian, ketika program dijalankan, sebuah jendela kosong dengan ukuran 400x400 piksel akan muncul di layar. Program ini tidak memiliki elemen lain di dalam jendela, seperti tombol atau teks, hanya sebuah jendela yang ditampilkan.

**Output**

|  |
| --- |
|  |

## Soal Praktikum

Buatlah program Java Swing yang memiliki input berupa JTextField untuk

memasukkan username dan JPasswordField untuk memasukkan password.

Tambahkan tombol "Login". Ketika tombol "Login" ditekan, lakukan validasi

dengan ketentuan berikut:

- Panjang password minimal 8 karakter.

- Password harus mengandung huruf besar, huruf kecil, dan angka.

- Jika validasi berhasil, tampilkan pesan "Login Berhasil" di JLabel.

- Jika validasi gagal, tampilkan pesan "Password tidak valid" di JLabel.

## Jawaban

Ketik jawaban disini …

## Source Code

*package* soalNo4;  
  
*import* javax.swing.\*;  
*import* java.awt.\*;  
*import* java.awt.event.ActionEvent;  
*import* java.awt.event.ActionListener;  
  
*public class* main {  
 *public static void* main(String[] args) {  
 JFrame frame = *new* JFrame("PRAKTIKUM 5");  
  
 frame.setLayout(*null*);  
 frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  
 frame.setLocationRelativeTo(*null*);  
 frame.setSize(400, 300);  
  
 JLabel username = *new* JLabel("Username");  
 JLabel password = *new* JLabel("Password");  
  
 username.setBounds(50, 50, 100, 30);  
 frame.add(username);  
  
 password.setBounds(50, 100, 100, 30);  
 frame.add(password);  
  
 JTextField usernameField = *new* JTextField();  
 usernameField.setBounds(150, 55, 165, 20);  
 frame.add(usernameField);  
  
 JPasswordField passwordField = *new* JPasswordField();  
 passwordField.setBounds(150, 105, 165, 20);  
 frame.add(passwordField);  
  
 JButton loginButton = *new* JButton("LOGIN");  
 loginButton.setBounds(150, 150, 100, 30);  
 frame.add(loginButton);  
  
 JLabel validationMessage = *new* JLabel();  
 validationMessage.setBounds(50, 200, 300, 30);  
 frame.add(validationMessage);  
  
 loginButton.addActionListener(*new* ActionListener() {  
 *public void* actionPerformed(ActionEvent e) {  
 String password = *new* String(passwordField.getPassword());  
 *try* {  
 validatePassword(password);  
 validationMessage.setText("Login Berhasil");  
 validationMessage.setForeground(Color.GREEN);  
 } *catch* (Exception ex) {  
 validationMessage.setText("Password tidak valid. " + ex.getMessage());  
 validationMessage.setForeground(Color.RED);  
 }  
 }  
 });  
  
 frame.setVisible(*true*);  
 }  
  
 *private static void* validatePassword(String password) *throws* Exception {  
 *if* (password.length() < 8) {  
 *throw new* Exception("Panjang password minimal 8 karakter.");  
 }  
 *if* (!password.matches(".\*[A-Z].\*")) {  
 *throw new* Exception("Password harus mengandung huruf besar.");  
 }  
 *if* (!password.matches(".\*[a-z].\*")) {  
 *throw new* Exception("Password harus mengandung huruf kecil.");  
 }  
 *if* (!password.matches(".\*\\d.\*")) {  
 *throw new* Exception("Password harus mengandung angka.");  
 }  
 }  
}

**Penjelasan**

Program Java ini adalah aplikasi login sederhana dengan GUI menggunakan JFrame dan Swing, yang memvalidasi password yang dimasukkan oleh pengguna. Program ini menciptakan sebuah jendela dengan dua label untuk "Username" dan "Password", dua field input (satu JTextField untuk username dan satu JPasswordField untuk password), dan sebuah tombol "LOGIN". Ketika tombol login ditekan, aplikasi akan memvalidasi password yang dimasukkan melalui metode validatePassword(). Jika password memenuhi syarat (minimal 8 karakter, mengandung huruf besar, huruf kecil, dan angka), maka akan muncul pesan "Login Berhasil" berwarna hijau. Jika tidak, aplikasi akan menampilkan pesan kesalahan yang sesuai dengan alasan kegagalan validasi password (misalnya "Panjang password minimal 8 karakter" atau "Password harus mengandung huruf besar") dengan teks berwarna merah. Jendela aplikasi ini juga disesuaikan ukurannya menjadi 400x300 piksel dan diposisikan di tengah layar. Program ini menggunakan null layout untuk menempatkan elemen-elemen GUI secara manual menggunakan metode setBounds.

**Output**

|  |
| --- |
|  |

**Soal Praktikum**

Melanjutkan dari soal no 3, tambahkan komponen JComboBox dengan

beberapa pilihan image. Ketika pengguna memilih image dari JComboBox, latar belakang JFrame harus berubah sesuai dengan image yang dipilih. Seperti

contoh berikut:

## Source Code

*package* soalNo5;  
  
*import* javax.swing.\*;  
*import* java.awt.\*;  
*import* java.awt.event.ActionEvent;  
*import* java.awt.event.ActionListener;  
  
*public class* Main {  
 *public static void* main(String[] args) {  
 JFrame frame = *new* JFrame("Frame dengan Background");  
 frame.setSize(400, 400);  
 frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  
 frame.setLocationRelativeTo(*null*);  
  
 BackgroundPanel backgroundPanel = *new* BackgroundPanel();  
 backgroundPanel.setLayout(*null*);  
  
 String[] imageOptions = {"Image 1", "Image 2", "Image 3"};  
 JComboBox<String> comboBox = *new* JComboBox<>(imageOptions);  
 comboBox.setBounds(10, 10, 150, 30);  
 backgroundPanel.add(comboBox);  
  
 comboBox.addActionListener(*new* ActionListener() {  
 *@Override  
 public void* actionPerformed(ActionEvent e) {  
 String selectedImage = (String) comboBox.getSelectedItem();  
 String imagePath = "";  
  
 *switch* (selectedImage) {  
 *case* "Image 1":  
 imagePath = "D:/ngoding adam/Java/Pratikum 5/pertemuan5\_07720/Pertemuan5\_07720/src/soalNo4/image/image1.jpeg";  
 *break*;  
 *case* "Image 2":  
 imagePath = "D:/ngoding adam/Java/Pratikum 5/pertemuan5\_07720/Pertemuan5\_07720/src/soalNo4/image/image2.jpeg";  
 *break*;  
 *case* "Image 3":  
 imagePath = "D:/ngoding adam/Java/Pratikum 5/pertemuan5\_07720/Pertemuan5\_07720/src/soalNo4/image/image3.jpeg";  
 *break*;  
 }  
  
 backgroundPanel.setBackgroundImage(imagePath);  
 backgroundPanel.repaint();  
 }  
 });  
  
 frame.setContentPane(backgroundPanel);  
 frame.setVisible(*true*);  
 }  
}  
  
*class* BackgroundPanel *extends* JPanel {  
 *private* Image backgroundImage;  
  
 *public void* setBackgroundImage(String imagePath) {  
 *this*.backgroundImage = *new* ImageIcon(imagePath).getImage();  
 }  
  
 *@Override  
 protected void* paintComponent(Graphics g) {  
 *super*.paintComponent(g);  
 *if* (backgroundImage != *null*) {  
 g.drawImage(backgroundImage, 0, 0, getWidth(), getHeight(), *this*);  
 }  
 }  
}

**Penjelasan**

Program Java ini membuat aplikasi GUI menggunakan JFrame dan JPanel, yang memungkinkan pengguna memilih gambar latar belakang dari dropdown (JComboBox). Sebuah jendela (JFrame) berukuran 400x400 piksel berjudul "Frame dengan Background" dibuat, dengan panel kustom bernama BackgroundPanel yang dapat mengganti latar belakang berdasarkan pilihan gambar yang dipilih dari dropdown. Tiga pilihan gambar tersedia (Image 1, Image 2, Image 3), dan ketika pengguna memilih salah satu, aplikasi akan mengganti gambar latar belakang sesuai dengan path gambar yang ditentukan. Gambar tersebut digambar menggunakan metode paintComponent(Graphics g) yang otomatis dipanggil untuk menggambar ulang panel. Program juga mengatur jendela agar muncul di tengah layar dan berhenti saat jendela ditutup dengan frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE). Setelah pilihan gambar dibuat, repaint() dipanggil untuk menggambar latar belakang yang baru, dan jendela akan menampilkan gambar yang dipilih sebagai latar belakang panel.

**Output**

|  |
| --- |
|  |

## Soal Praktikum

Ketik soal disini …

## Jawaban

Ketik jawaban disini …

## Source Code

Tulis kode program dikotak ini...

1 kotak dan 1 Penjelasan untuk 1 Class

**Penjelasan**

Tulis Penjelasan disini …

**Output**

|  |
| --- |
| Masukan screenshot output disini |