

**LAPORAN TUGAS BESAR**  
**BBK1JAB3 - PEMROGRAMAN BERORIENTASI OBJEK**  
**SEMESTER GANJIL 2024/2025**

**Nama Studi Kasus:**  
**Sistem Peminjaman Ruangan**



**Disusun oleh:**  
**Kelompok 3**

**BINTANG TOPA KURNIAWAN - 102022300073**  
**BAHAA AZZIKRA AZHAR ARMA - 102022300404**  
**M. FACHRI HAFIZHAN - 102022300441**

**Program Studi S1 Sistem Informasi**  
**Fakultas Rekayasa Industri**  
**Telkom University**  
**2024**

Use Case (1/N)	
Code	<i>Paste all the code related to the practical assignment in this section.</i>
	<pre>@FXML     private void handleLoginButton(ActionEvent event) {</pre>

```

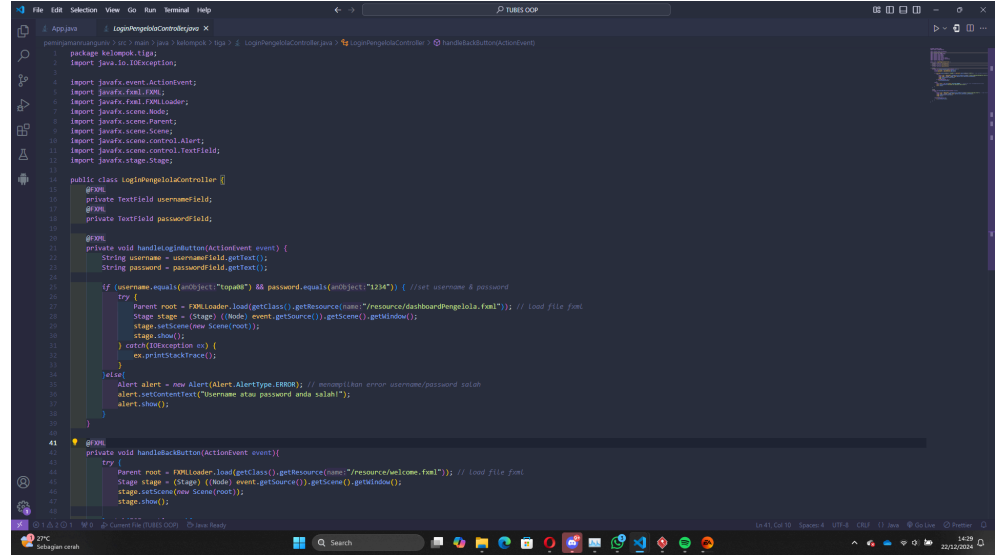
String username = usernameField.getText();
String password = passwordField.getText();

if (username.equals("topa08") &&
password.equals("1234")) { //set username & password
    try {
        Parent root =
FXMLLoader.load(getClass().getResource("/resource/dashboar
dPengelola.fxml")); // load file fxml
        Stage stage = (Stage) ((Node)
event.getSource()).getScene().getWindow();
        stage.setScene(new Scene(root));
        stage.show();
    } catch (IOException ex) {
        ex.printStackTrace();
    }
} else {
    Alert alert = new
Alert(Alert.AlertType.ERROR); // menampilkan error
username/password salah
    alert.setContentText("Username atau password
anda salah!");
    alert.show();
}
}

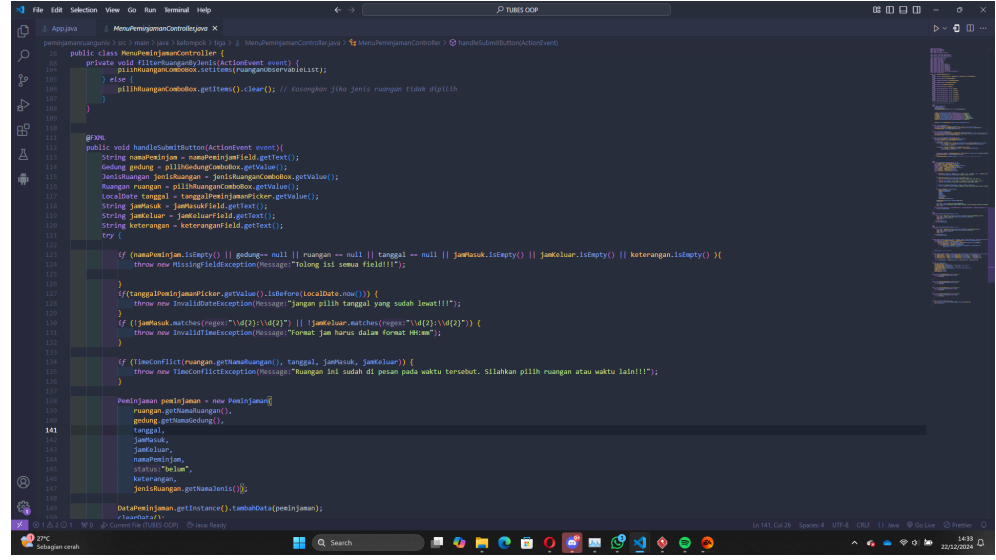
```

**Screenshot**

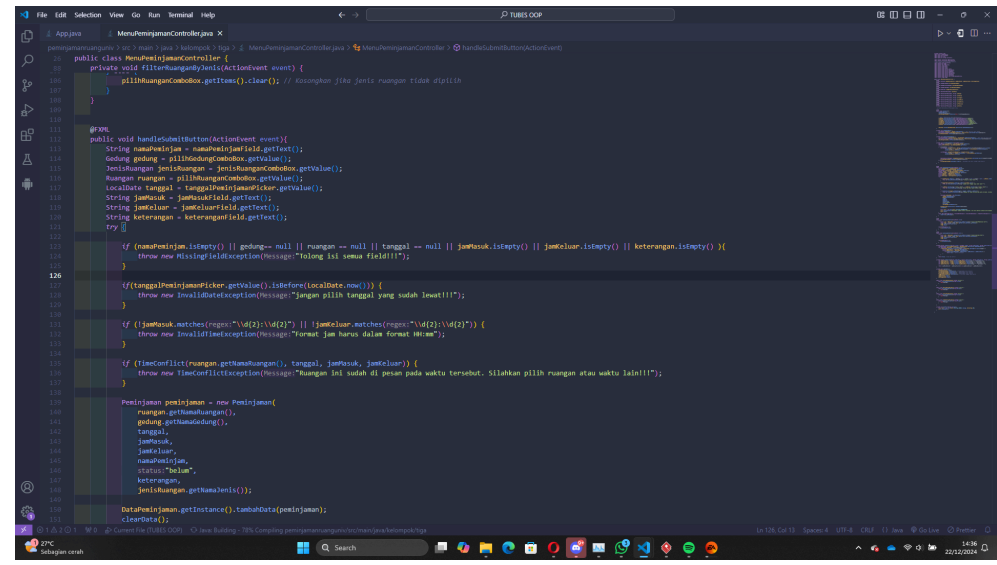
*Capture one screenshots of the program execution results in this section.*

	
Notes	<i>Alert error ketika login dengan username/password yang salah di login pengelola ataupun peminjam</i>

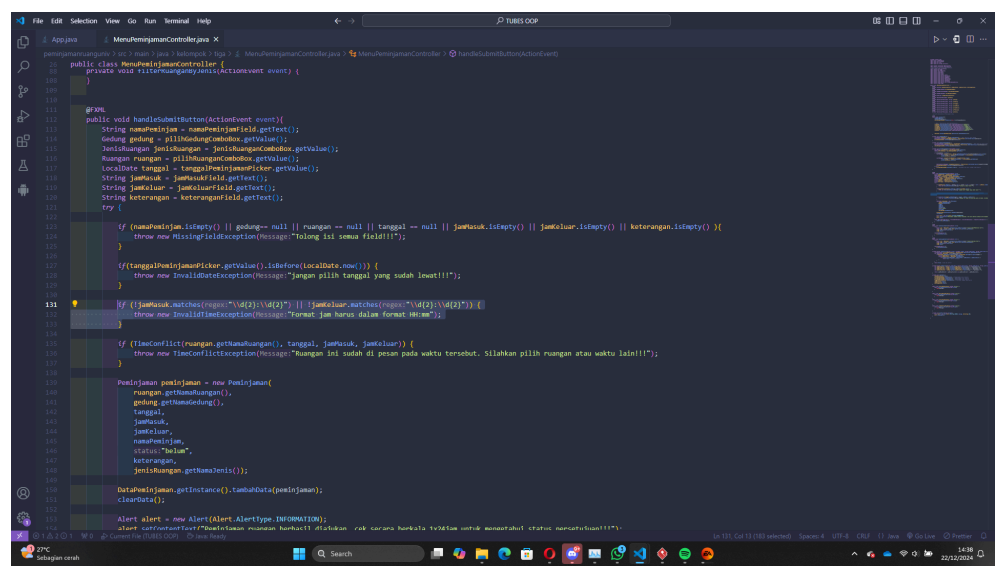
Use Case (2/N)	
Code	<p><i>Paste all the code related to the practical assignment in this section.</i></p> <pre>if (namaPeminjam.isEmpty()    gedung== null    ruangan == null    tanggal == null    jamMasuk.isEmpty()    jamKeluar.isEmpty()    keterangan.isEmpty() ){     throw new MissingFieldException("Tolong isi semua field!!!"); }</pre>
Screenshot	<p><i>Capture one screenshots of the program execution results in this section.</i></p>

	
Notes	Alert error ketika ada field yang belum diisi pada menu peminjaman ruang

Use Case (3/N)	
Code	<p>Paste all the code related to the practical assignment in this section.</p> <pre>if(tanggalPeminjamanPicker.getValue().isBefore(LocalDate.now())) {     throw new InvalidDateException("jangan pilih tanggal yang sudah lewat!!!"); }</pre>
Screenshot	Capture one screenshots of the program execution results in this section.

	
Notes	<i>Alert error ketika tanggal yang dipilih adalah tanggal yang sudah lampau/sudah lewat</i>

Use Case (4/N)	
Code	<p><i>Paste all the code related to the practical assignment in this section.</i></p> <pre>if (!jamMasuk.matches("\\d{2}:\\d{2}")    !jamKeluar.matches("\\d{2}:\\d{2}")) {     throw new InvalidTimeException("Format jam harus dalam format HH:mm"); }</pre>
Screenshot	<p><i>Capture one screenshots of the program execution results in this section.</i></p>

	 <pre> 111 public void handleTimeConflictException(Exception event) { 112     // ... 113     // (namaPeminjaman.isEmpty()    gedung == null    ruangan == null    tanggal == null    jamMasuk.isEmpty()    jamKeluar.isEmpty()    keterangan.isEmpty()) { 114         // throw new MissingFieldException("Tolong isi semua field!!!"); 115     // } 116     // (tanggalPeminjamanField.getValue().isBefore(LocalDate.now())) { 117         // throw new InvalidDateException("Tanggal harus lebih dari sekarang!!!"); 118     // } 119     // (jamMasuk.matches(regex)    jamKeluar.matches(regex)) { 120         // throw new InvalidTimeException("Format jam harus dalam format HH:mm"); 121     // } 122     // (TimeConflict(ruangan.getNamaRuangan(), tanggal, jamMasuk, jamKeluar)) { 123         // throw new TimeConflictException("Ruangan ini sudah di pesan pada waktu tersebut. Silahkan pilih ruangan atau waktu lain!!!"); 124     // } 125     Peminjaman peminjaman = new Peminjaman( 126         ruangan.getNamaRuangan(), 127         gedung.getNamaGedung(), 128         tanggal, 129         jamMasuk, 130         jamKeluar, 131         namaPeminjaman, 132         status, 133         keterangan, 134         jamMasuk, 135         jamKeluar); 136     DataPeminjaman.getInstance().tambahData(peminjaman); 137     clearData(); 138     Alert alert = new Alert(Alert.AlertType.INFORMATION); 139     alert.setTitle("Peminjaman Berhasil"); 140     alert.setContentText("Peminjaman berhasil dilakukan. Silahkan pilih ruangan atau waktu lain!!!"); 141     alert.show(); 142 } </pre>
Notes	Alert error ketika format penulisan waktu/jam salah

Use Case (5/N)	
Code	<p>Paste all the code related to the practical assignment in this section.</p> <pre> if (TimeConflict(ruangan.getNamaRuangan(), tanggal, jamMasuk, jamKeluar)) {     throw new TimeConflictException("Ruangan ini sudah di pesan pada waktu tersebut. Silahkan pilih ruangan atau waktu lain!!!"); }  private boolean TimeConflict(String ruangan, LocalDate tanggal, String jamMasukBaru, String jamKeluarBaru) {     List&lt;Peminjaman&gt; existingPeminjamanList = DataPeminjaman.getInstance().getDataRuangList();     for (Peminjaman peminjaman : existingPeminjamanList) {         if(peminjaman.getRuangan().equals(ruangan) &amp;&amp; peminjaman.getTanggal().equals(tanggal)) {             String jamMasukExist = peminjaman.getJamMasuk();             String jamKeluarExist = peminjaman.getJamKeluar();             if (TimeOverlap(jamMasukBaru, jamKeluarBaru, jamMasukExist, jamKeluarExist)) { </pre>

```

        return true; // Konflik ditemukan
    }
}

return false; //tidak ada konflik
}

private boolean TimeOverlap(String jamMasukBaru,
String jamKeluarBaru, String jamMasukExist, String
jamKeluarExist) {
    int jamMasukBaruInt =
Integer.parseInt(jamMasukBaru.replace(":", ""));
    int jamKeluarBaruInt =
Integer.parseInt(jamKeluarBaru.replace(":", ""));
    int jamMasukExistInt =
Integer.parseInt(jamMasukExist.replace(":", ""));
    int jamKeluarExistInt =
Integer.parseInt(jamKeluarExist.replace(":", ""));

    return (jamMasukBaruInt < jamKeluarExistInt) &&
(jamKeluarBaruInt > jamMasukExistInt);
}

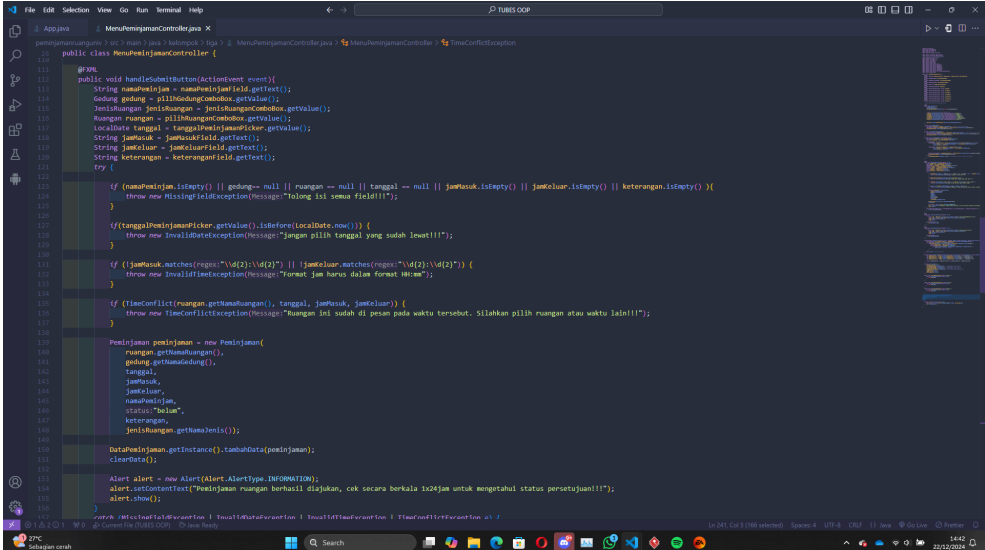
public class TimeConflictException extends Exception {
    public TimeConflictException(String Message){
        super(Message);
    }
}
}

```

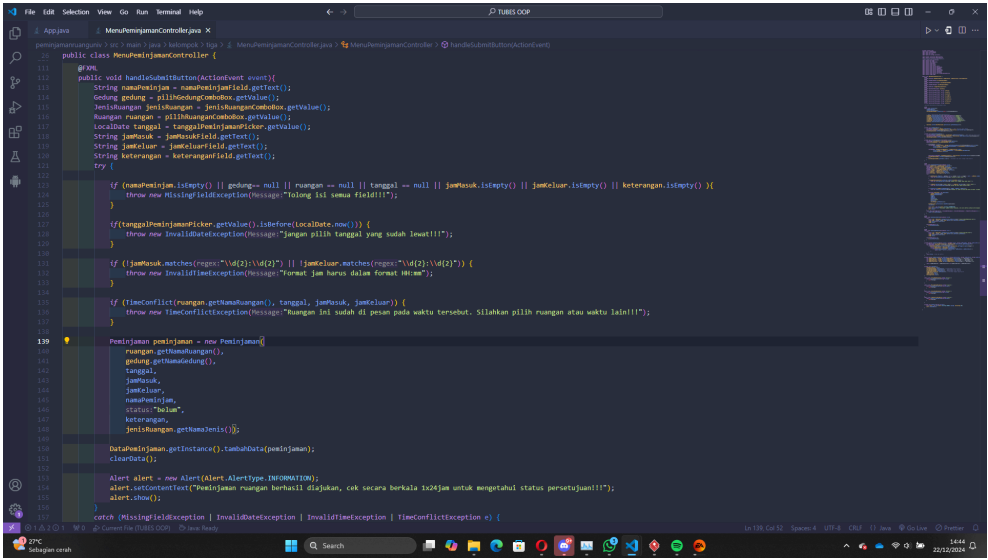
**Screenshot**

*Capture one screenshots of the program execution results in this section.*

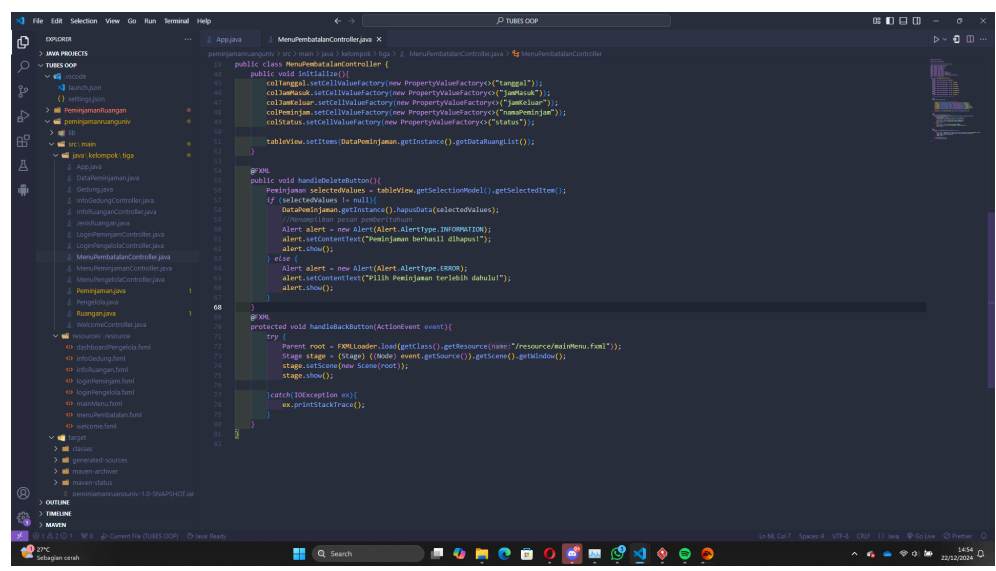


	 <pre> 111 public void handleButtonClick(ActionEvent event) { 112     String roomId = roomIdField.getText(); 113     Gedung gedung = pilihGedungComboBox.getValue(); 114     JmlRuangon pilihRuangon = pilihRuangonComboBox.getValue(); 115     Ruangon ruangan = pilihRuangonComboBox.getValue(); 116     LocalDate tanggal = tanggalPicker.getValue(); 117     String jamMasuk = jamMasukField.getText(); 118     String jamKeluar = jamKeluarField.getText(); 119     String keterangan = keteranganField.getText(); 120     try { 121         if (roomId.isEmpty()    gedung == null    ruangan == null    tanggal == null    jamMasuk.isEmpty()    jamKeluar.isEmpty()    keterangan.isEmpty()) { 122             throw new IllegalArgumentException("Tolong isi semua field!!!"); 123         } 124         if (tanggalPicker.getValue().isBefore(LocalDate.now())) { 125             throw new IllegalArgumentException("Jangan pilih tanggal yang sudah lewat!!!"); 126         } 127         if (!jamMasuk.matches(regex "\\d{2}:\\d{2}")    !jamKeluar.matches(regex "\\d{2}:\\d{2}")) { 128             throw new IllegalArgumentException("Format jam harus dalam format HH:mm"); 129         } 130         if (!timeConflict(ruangan.getRuangon(), tanggal, jamMasuk, jamKeluar)) { 131             throw new IllegalArgumentException("Ruangon ini sudah di pesan pada waktu tersebut. Silahkan pilih ruangan atau waktu lain!!!"); 132         } 133         Peminjaman peminjaman = new Peminjaman( 134             ruangan.getRuangon(), 135             gedung.getGedung(), 136             tanggal, 137             jamMasuk, 138             jamKeluar, 139             roomId, 140             keterangan, 141             pilihRuangon.getRuangon()); 142         DataPeminjaman.getInstance().tambahData(peminjaman); 143         clearData(); 144         Alert alert = new Alert(Alert.AlertType.INFORMATION); 145         alert.setContentText("Peminjaman ruangan berhasil diujikan, cek secara berkala is24jam untuk mengetahui status persetujuan!!!"); 146         alert.show(); 147     } catch (Exception e) { 148         e.printStackTrace(); 149     } 150 } 151 152 private boolean timeConflict(String ruangan, LocalDate tanggal, String jamMasukBaru, String jamKeluarBaru) { 153     List&lt;Peminjaman&gt; existingPeminjamanList = DataPeminjaman.getInstance().getPeminjamanList(); 154     for (Peminjaman peminjaman : existingPeminjamanList) { 155         if (peminjaman.getRuangon().equals(ruangan) &amp;&amp; peminjaman.getTanggal().equals(tanggal)) { 156             String jamMasukExist = peminjaman.getJamMasuk(); 157             String jamKeluarExist = peminjaman.getJamKeluar(); 158             if (timeOverlap(jamMasukBaru, jamKeluarBaru, jamMasukExist, jamKeluarExist)) { 159                 return true; // Konflik ditemukan 160             } 161         } 162     } 163     return false; // Tidak ada konflik 164 } 165 166 private boolean timeOverlap(String jamMasukBaru, String jamKeluarBaru, String jamMasukExist, String jamKeluarExist) { 167     int jamMasukBaruInt = Integer.parseInt(jamMasukBaru.replace(":", "")); 168     int jamKeluarBaruInt = Integer.parseInt(jamKeluarBaru.replace(":", "")); 169     int jamMasukExistInt = Integer.parseInt(jamMasukExist.replace(":", "")); 170     int jamKeluarExistInt = Integer.parseInt(jamKeluarExist.replace(":", "")); 171     return (jamMasukBaruInt &lt; jamKeluarExistInt &amp;&amp; (jamKeluarBaruInt &gt; jamMasukExistInt)); 172 } 173 174 private void clearData() { 175     roomIdField.clear(); 176     pilihGedungComboBox.setValue(null); // Mengembalikan nilai terpilih 177     pilihRuangonComboBox.setValue(null); // Mengembalikan nilai terpilih 178     tanggalPicker.setValue(null); // Bersihkan tanggal yang dipilih 179     jamMasukField.clear(); 180     jamKeluarField.clear(); 181     keteranganField.clear(); 182 } </pre>
Notes	<p>Alert error ketika tanggal,waktu, dan ruangan yang dipilih untuk dipinjam bertabrakan dengan ruangan yang sudah dipinjam</p>

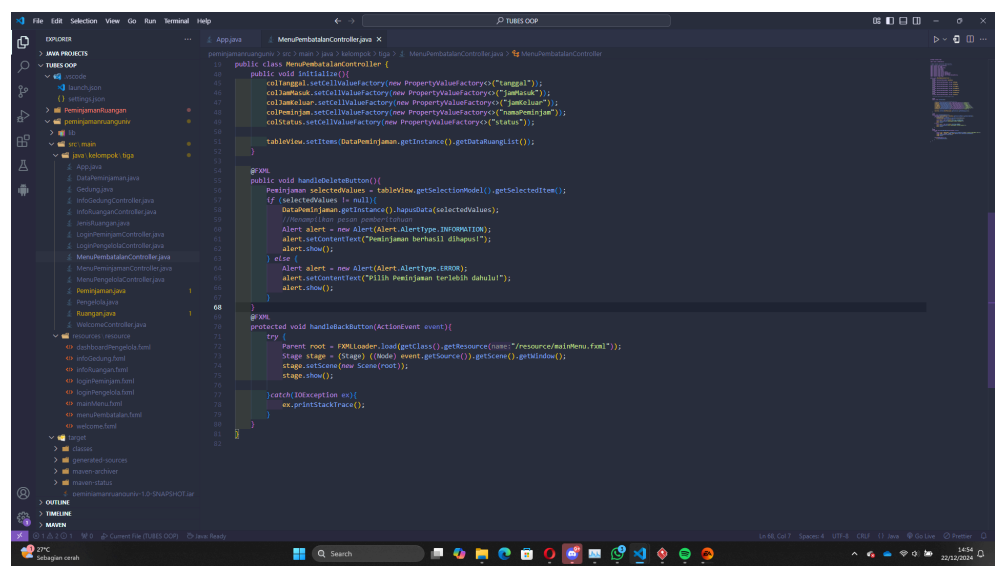
Use Case (6/N)	
Code	<p>Paste all the code related to the practical assignment in this section.</p> <pre> DataPeminjaman.getInstance().tambahData(peminjaman);  clearData();  Alert alert = new Alert(Alert.AlertType.INFORMATION);  alert.setContentText("Peminjaman ruangan </pre>

	<pre>berhasil diajukan, cek secara berkala 1x24jam untuk mengetahui status persetujuan!!!");  alert.show();</pre>
Screenshot	<p><i>Capture one screenshots of the program execution results in this section.</i></p> 
Notes	<p><i>Alert information ketika berhasil melakukan submit peminjaman</i></p>

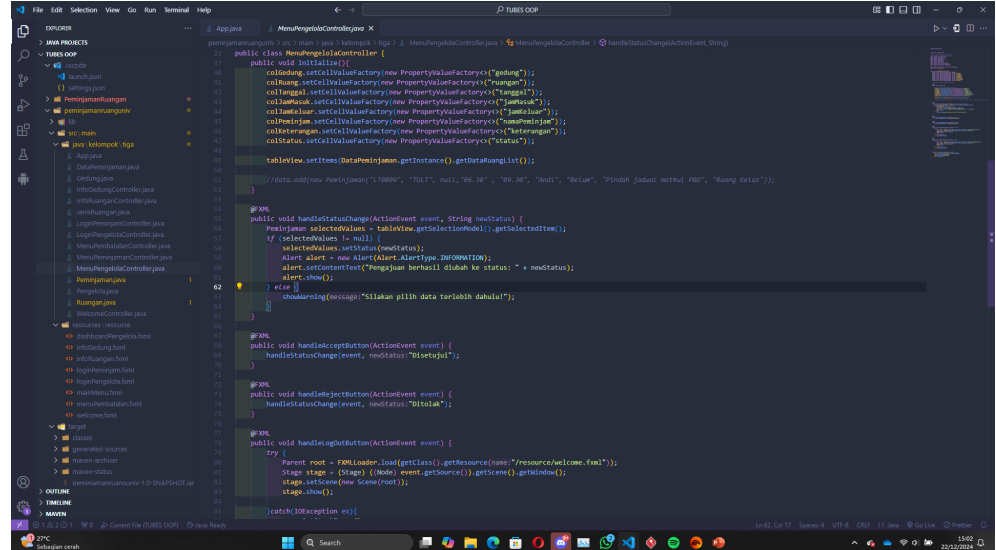
Use Case (7/N)	
Code	<p><i>Paste all the code related to the practical assignment in this section.</i></p> <pre>if (selectedValues != null){  DataPeminjaman.getInstance().hapusData(selectedValues);  //Menampilkan pesan pemberitahuan Alert alert = new Alert(Alert.AlertType.INFORMATION);  alert.setContentText("Peminjaman berhasil dihapus!");  alert.show(); }</pre>
Screenshot	<p><i>Capture one screenshots of the program execution results in this section.</i></p>

	
Notes	<i>Alert information ketika berhasil menghapus peminjaman</i>

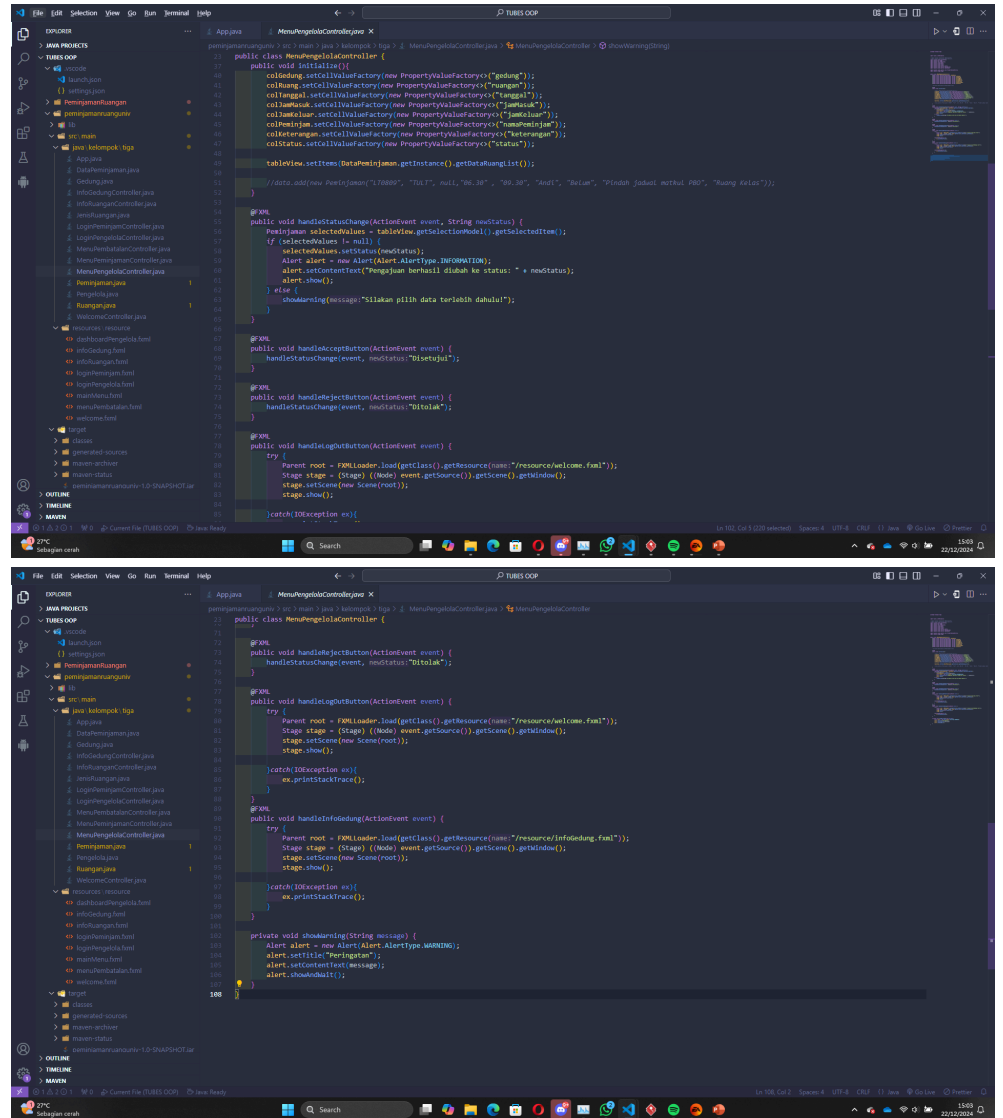
Use Case (8/N)	
Code	<p><i>Paste all the code related to the practical assignment in this section.</i></p> <pre>    } else {          Alert alert = new Alert(Alert.AlertType.ERROR);          alert.setContentText("Pilih Peminjaman terlebih dahulu!");          alert.show();      }</pre>
Screenshot	<p><i>Capture one screenshots of the program execution results in this section.</i></p>

	
Notes	Alert error ketika belum memilih data peminjaman di tabel untuk menghapusnya

Use Case (9/N)	
Code	<p>Paste all the code related to the practical assignment in this section.</p> <pre>if (selectedValues != null) {     selectedValues.setStatus(newStatus);     Alert alert = new Alert(Alert.AlertType.INFORMATION);     alert.setContentText("Pengajuan berhasil diubah ke status: " + newStatus);     alert.show(); }</pre>
Screenshot	Capture one screenshots of the program execution results in this section.

	
Notes	Alert information ketika berhasil mengubah status peminjaman

Use Case (10/N)	
Code	<p>Paste all the code related to the practical assignment in this section.</p> <pre>    } else {          showWarning("Silakan pilih data terlebih dahulu!");     }  private void showWarning(String message) {     Alert alert = new Alert(Alert.AlertType.WARNING);     alert.setTitle("Peringatan");     alert.setContentText(message);     alert.showAndWait(); } }</pre>
Screenshot	Capture one screenshots of the program execution results in this section.



Notes *Alert warning ketika belum memilih peminjaman saat ingin mengubah status peminjaman*

## Use Case (11/N)

### Code

Paste all the code related to the practical assignment in this section.

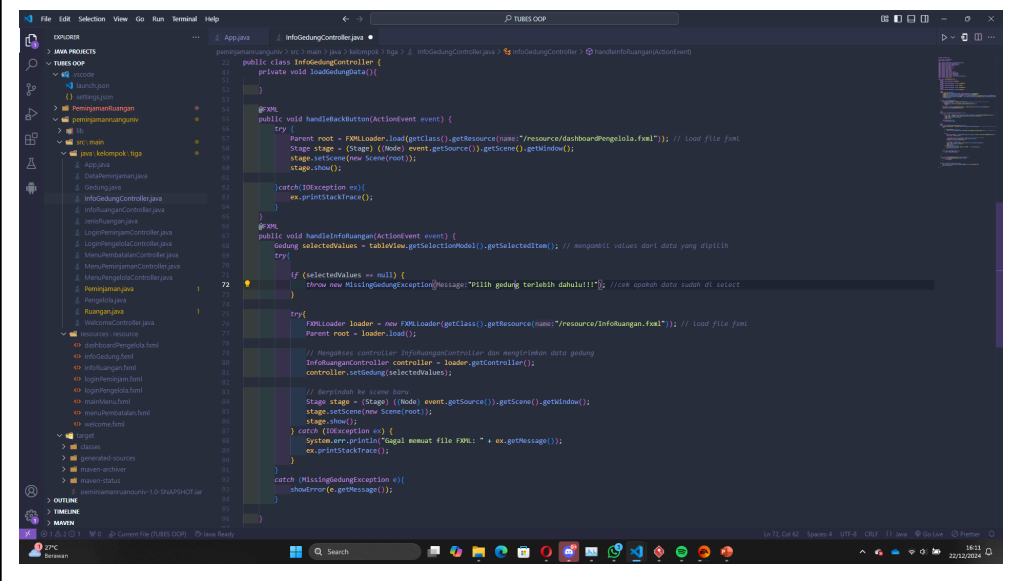
```

if (selectedValues == null) {
    throw new MissingGedungException("Pilih
gedung terlebih dahulu!!!"); //cek apakah data sudah di
select
}

```

## Screenshot

<p><i>Capture one screenshots of the program execution results in this section.</i></p>	



## Notes

*Alert error ketika belum memilih gedung ketika ingin melihat informasi ruangan*

Link youtube video demo: <https://youtu.be/R0TNc4gM9ko>