

Business Application Development

Kelompok 14 :

- Wisnu Ramadhan - 2440106693
 - Nifsu Damareodito Ismawan - 2440017416
 - Muhammad Farrel Hakim - 2440017441
 - Steven Gozali - 2440040085

Contoh Codingan

1. Login Page

eclipse-workspace - Project_BA11_BAD/src/pokemonNK/Login.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer Project Explorer

Login.java

```
1 package pokemonNK;
2 // Notes : untuk roles kami random Kak, oleh karena itu, untuk mendapatkan role antara member dan admin, bisa untuk melakukan login hingga mendapatkan
3 
4 import java.awt.*;
5 
6 public class Login extends JFrame {
7 
8     static String role = "";
9     // HashMap<String, String> login = new HashMap<String, String>();
10 
11     DBConnect con = DBConnect.getConnection();
12 
13     public Login() {
14 
15         JPanel panel = new JPanel(new GridLayout(9,1));
16         panel.setBackground(Color.CYAN);
17         JLabel loginLbl = new JLabel("Login");
18         loginLbl.setHorizontalAlignment(JLabel.CENTER);
19         panel.setBorder(new EmptyBorder(30, 100, 30, 100));
20         panel.add(loginLbl);
21 
22         //username
23         JPanel usernamePanel = new JPanel(new GridLayout(2,1));
24         usernamePanel.setBackground(Color.CYAN);
25         JLabel usernameLbl = new JLabel("Username");
26         usernameLbl.setHorizontalAlignment(JLabel.LEADING);
27         usernamePanel.add(usernameLbl);
28 
29         JTextField usernameTF = new JTextField();
30         usernamePanel.add(usernameTF);
31         panel.add(usernamePanel);
32 
33         //pass
34         JPanel passwordPanel = new JPanel(new GridLayout(2,1));
35         passwordPanel.setBackground(Color.CYAN);
36         JLabel passwordLbl = new JLabel("Password");
37         passwordPanel.add(passwordLbl);
38 
39         JButton loginBt = new JButton("Login");
40         JButton registerBt = new JButton("Register");
41         JButton exitBt = new JButton("Exit");
42         passwordPanel.add(loginBt);
43         passwordPanel.add(registerBt);
44         passwordPanel.add(exitBt);
45         panel.add(passwordPanel);
46 
47         loginBt.addActionListener(e -> {
48             String username = usernameTF.getText();
49             String password = passwordTF.getText();
50             if (password.equals("admin")) {
51                 role = "Admin";
52             } else {
53                 role = "Member";
54             }
55             System.out.println("Role: " + role);
56         });
57     }
58 }
```

Problems Javadoc Declaration Coverage Console

<terminated> Login (1) [Java Application] C:\Program Files\Java\jre1.8.0_291\bin\javaw.exe (Jan 12, 2022 7:47:51 AM - 7:48:00 AM)

eclipse-workspace - Project_BA11_BAD/src/pokemonNK/Login.java - Eclipse IDE

```
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer Project Explorer Login.java
52 JPanel passpn = new JPanel(new GridLayout(2,1));
53 passpn.setBackground(Color.CYAN);
54 JLabel passlbl = new JLabel("Password");
55 passlbl.setHorizontalAlignment(JLabel.LEADING);
56 passpn.add(passlbl);
57
58 JPasswordField passTA = new JPasswordField();
59 passpn.add(passTA);
60 panel1.add(passpn);
61
62 //button
63 JPanel bpanel = new JPanel();
64 bpanel.setBackground(Color.CYAN);
65 JButton blogin = new JButton("Login");
66
67 JButton bRegist = new JButton("Register");
68 blogin.addActionListener(new ActionListener() {
69
70     @Override
71     public void actionPerformed(ActionEvent e) {
72         String username = usernameTF.getText();
73         String password = String.valueOf(passTA.getPassword());
74
75         if (username.isEmpty() || password.isEmpty()) {
76             JOptionPane.showMessageDialog(null, "Username or Password cannot be Empty!");
77         } else {
78
79             String query = "SELECT Username, Name, Password FROM user WHERE Username = '"+username+"' AND Password = '"+password+"'";
80             ResultSet rs = con.executeQuery(query);
81
82             usernameTF.setText("");
83             passTA.setText("");
84
85             try {
86                 if (rs.next()) {
87                     username = rs.getString(1);
88                     String name = rs.getString(2);
89                     Random rd = new Random();
90                     String token = "token"+rd.nextInt(1000000);
91                     rs.close();
92                     rs = null;
93                     rs2 = null;
94                     rs3 = null;
95                     rs4 = null;
96                     rs5 = null;
97                     rs6 = null;
98                     rs7 = null;
99                     rs8 = null;
100
101                     bRegist.addActionListener(new ActionListener() {
102
103                         @Override
104                         public void actionPerformed(ActionEvent e) {
105                             dispose();
106                             Register r = new Register();
107                         }
108                     });
109
110                     bPanel.add(bLogin);
111                     bPanel.add(bRegist);
112                     panel1.add(bPanel);
113
114                     add(panel1);
115                     init();
116
117                 }
118             }
119         }
120     }
121
122     bPanel.add(bLogin);
123     bPanel.add(bRegist);
124     panel1.add(bPanel);
125
126     add(panel1);
127
128 }
129
130
131
132 }
133
134 public void init() {
135     setPreferredSize(new Dimension(600, 400));
136     setTitle("Pokemon");
137     this.setLocationRelativeTo(null);
138     this.setDefaultCloseOperation(EXIT_ON_CLOSE);
139     this.setVisible(true);
140 }
141
142 public static void main(String[] args) {
143     new Login();
144 }
145
146 }
```

Problems Javadoc Declaration Coverage Console
<terminated> Login (1) [Java Application] C:\Program Files\Java\jre1.8.0_291\bin\javaw.exe (Jan 12, 2022 7:47:51 AM - 7:48:00 AM)

Type here to search

eclipse-workspace - Project_BA11_BAD/src/pokemonNK/Login.java - Eclipse IDE

```
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer Project Explorer Login.java
109     });
110
111     bRegist.addActionListener(new ActionListener() {
112
113         @Override
114         public void actionPerformed(ActionEvent e) {
115             dispose();
116             Register r = new Register();
117         }
118     });
119
120     });
121
122     bPanel.add(bLogin);
123     bPanel.add(bRegist);
124     panel1.add(bPanel);
125
126     add(panel1);
127
128 }
129
130
131
132 }
133
134 public void init() {
135     setPreferredSize(new Dimension(600, 400));
136     setTitle("Pokemon");
137     this.setLocationRelativeTo(null);
138     this.setDefaultCloseOperation(EXIT_ON_CLOSE);
139     this.setVisible(true);
140 }
141
142 public static void main(String[] args) {
143     new Login();
144 }
145
146 }
```

Problems Javadoc Declaration Coverage Console
<terminated> Login (1) [Java Application] C:\Program Files\Java\jre1.8.0_291\bin\javaw.exe (Jan 12, 2022 7:47:51 AM - 7:48:00 AM)

Type here to search

```

eclipse-workspace - Project_BA11_BAD/src/pokemonNK/Login.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer Project Explorer
Login.java
88     String name = rs.getString(2);
89     Random rd = new Random();
90     String [] arr = {"Admin", "Player"};
91     // randomly selects an index from the arr
92     int select = rd.nextInt(arr.length);
93     role = arr[select];
94
95     JOptionPane.showMessageDialog(null, "Welcome, " + name);
96     dispose();
97     MainForm mf = new MainForm();
98 } else {
99     JOptionPane.showMessageDialog(null, "Incorrect Username or Password");
100 }
101
102 } catch (SQLException e2) {
103     e2.printStackTrace();
104 }
105 }
106 }
107 }
108 }
109 }
110 }
111 }
112 bRegist.addActionListener(new ActionListener() {
113     @Override
114     public void actionPerformed(ActionEvent e) {
115         dispose();
116         Register r = new Register();
117     }
118 }
119 );
120 });
121 bPanel.add(bLogin);
122 bPanel.add(bRegist);
123 panel1.add(bPanel);
124
125

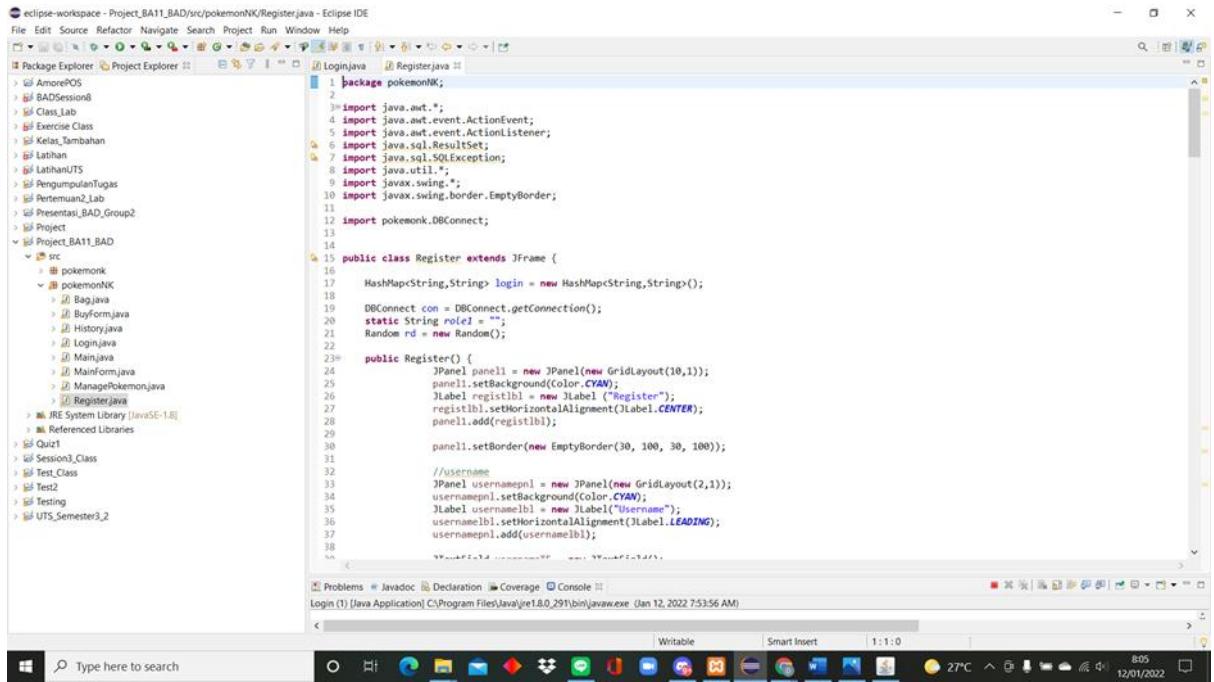
```

Problems Javadoc Declaration Coverage Console

<terminated> Login (1) [Java Application] C:\Program Files\Java\jre1.8.0_291\bin\java.exe (Jan 12, 2022 7:47:51 AM - 7:48:00 AM)

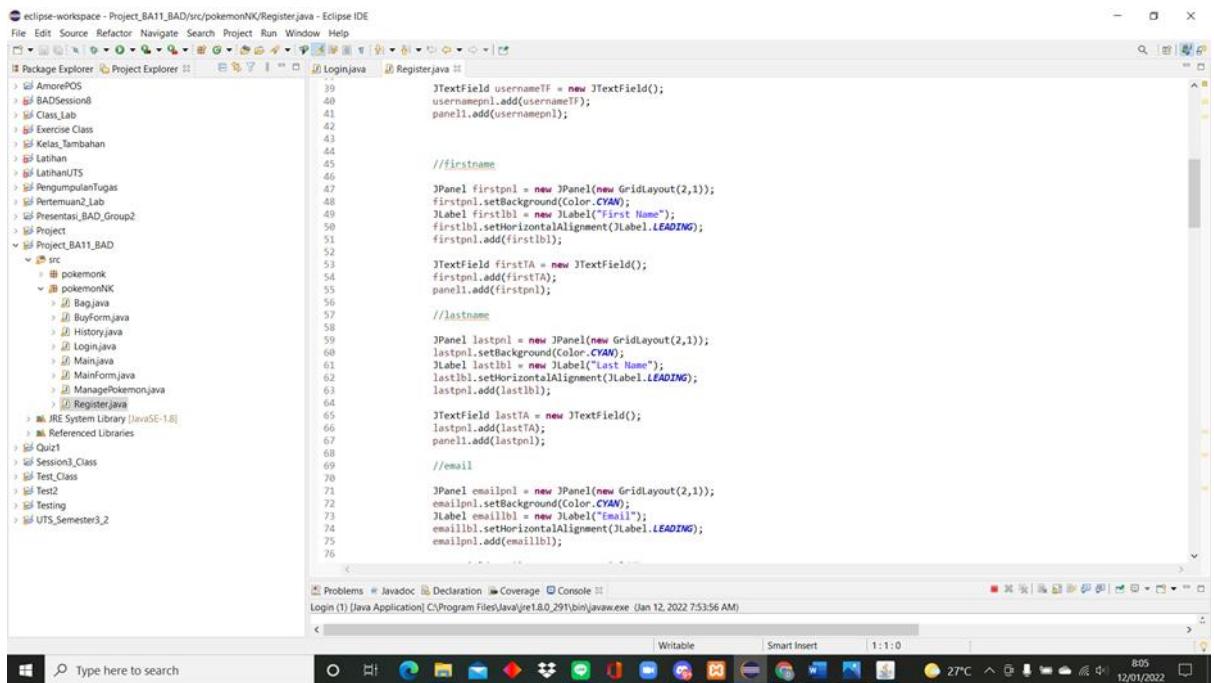
- Pembuatan panel dengan gridlayout 9 x 1 (9 vertikal dan 1 horizontal) dengan warna background Cyan
- Dimana ‘Login’ Label berada di tengah
- Setelah itu dibawah label ‘Login’. User diminta untuk input Username dan password.
- Sesudah menginput username dan password, jika user mengklik button ‘Login’ akan ada 3 kemungkinan :
 - 1) Jika username dan password tidak ada inputannya, maka sistem menunjukkan **"Username or Password cannot be Empty!"**
 - 2) Jika username dan password yang diinput tidak ada/salah, maka sistem akan menunjukkan **"Incorrect Username or Password"**
 - 3) Jika Username dan password yang diinput benar, maka akan dilanjutkan ke bagian *MainForm*, disini kami menggunakan metode random, dimana Ketika user login, ada 2 kemungkinan:
 - A. User login menjadi **"Admin"**
 - B. user login menjadi **"Player"**
- Jika user mengklik button ‘Register’ maka user akan berpindah ke register
- Untuk ukuran dari frame tersebut adalah 600 x 400 dengan lokasi di tengah layer
- Jika user mengklik symbol ‘ X ‘ maka sistem akan keluar dan berhenti bekerja

2. Register



The screenshot shows the Eclipse IDE interface with the 'Register.java' file open in the editor. The code implements a registration logic for a user. It starts by importing necessary Java packages and classes. It then defines a 'Register' class that extends 'JFrame'. Inside the class, it initializes a 'HashMap' for login details and a database connection. A random role is assigned to the user. The constructor sets up the main form with a grid layout and adds a registration label to it. The code then creates a panel for the username input, setting its background color to CYAN and adding a label and text field. Similar panels are created for first name, last name, and email. Finally, the code adds all components to the main panel.

```
1 package pokemonNK;
2
3 import java.awt.*;
4 import java.awt.event.ActionEvent;
5 import java.awt.event.ActionListener;
6 import java.sql.ResultSet;
7 import java.sql.SQLException;
8 import java.util.*;
9 import javax.swing.*;
10 import javax.swing.border.EmptyBorder;
11
12 import pokemon.NK.DBConnect;
13
14
15 public class Register extends JFrame {
16
17     HashMap<String, String> login = new HashMap<String, String>();
18
19     DBConnect con = DBConnect.getConnection();
20     static String role = "";
21     Random rd = new Random();
22
23     public Register() {
24         JPanel panel1 = new JPanel(new GridLayout(10,1));
25         panel1.setBackground(Color.CYAN);
26         JLabel registlbl = new JLabel("Register");
27         registlbl.setHorizontalTextAlignment(JLabel.CENTER);
28         panel1.add(registlbl);
29
30         panel1.setBorder(new EmptyBorder(30, 100, 30, 100));
31
32         //username
33         JPanel usernamepln = new JPanel(new GridLayout(2,1));
34         usernamepln.setBackground(Color.CYAN);
35         JLabel usernamelbl = new JLabel("Username");
36         usernamelbl.setHorizontalAlignment(JLabel.LEADING);
37         usernamepln.add(usernamelbl);
38
39         JTextField usernameTF = new JTextField();
40         usernamepln.add(usernameTF);
41         panel1.add(usernamepln);
42
43         //firstname
44
45         JPanel firstpln = new JPanel(new GridLayout(2,1));
46         firstpln.setBackground(Color.CYAN);
47         JLabel firstlbl = new JLabel("First Name");
48         firstlbl.setHorizontalAlignment(JLabel.LEADING);
49         firstpln.add(firstlbl);
50
51         JTextField firstTA = new JTextField();
52         firstpln.add(firstTA);
53         panel1.add(firstpln);
54
55         //lastname
56
57         JPanel lastpln = new JPanel(new GridLayout(2,1));
58         lastpln.setBackground(Color.CYAN);
59         JLabel lastlbl = new JLabel("Last Name");
60         lastlbl.setHorizontalAlignment(JLabel.LEADING);
61         lastpln.add(lastlbl);
62
63         JTextField lastTA = new JTextField();
64         lastpln.add(lastTA);
65         panel1.add(lastpln);
66
67         //email
68
69         JPanel emailpln = new JPanel(new GridLayout(2,1));
70         emailpln.setBackground(Color.CYAN);
71         JLabel emaillbl = new JLabel("Email");
72         emaillbl.setHorizontalAlignment(JLabel.LEADING);
73         emailpln.add(emaillbl);
74
75         JTextField emailTF = new JTextField();
76         emailpln.add(emailTF);
77
78         panel1.add(emailpln);
79
80         this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
81         this.setTitle("Register");
82         this.setContentPane(panel1);
83         this.pack();
84         this.setVisible(true);
85     }
86
87     public static void main(String[] args) {
88         new Register();
89     }
90 }
```



This screenshot shows the same Eclipse IDE session with the 'Register.java' code. The code is identical to the one in the previous screenshot, but the cursor is currently positioned on the line 'label emaillbl = new JLabel("Email");' in the 'lastpln' section. The rest of the code and interface are identical to the first screenshot.

```
1 package pokemonNK;
2
3 import java.awt.*;
4 import java.awt.event.ActionEvent;
5 import java.awt.event.ActionListener;
6 import java.sql.ResultSet;
7 import java.sql.SQLException;
8 import java.util.*;
9 import javax.swing.*;
10 import javax.swing.border.EmptyBorder;
11
12 import pokemon.NK.DBConnect;
13
14
15 public class Register extends JFrame {
16
17     HashMap<String, String> login = new HashMap<String, String>();
18
19     DBConnect con = DBConnect.getConnection();
20     static String role = "";
21     Random rd = new Random();
22
23     public Register() {
24         JPanel panel1 = new JPanel(new GridLayout(10,1));
25         panel1.setBackground(Color.CYAN);
26         JLabel registlbl = new JLabel("Register");
27         registlbl.setHorizontalTextAlignment(JLabel.CENTER);
28         panel1.add(registlbl);
29
30         panel1.setBorder(new EmptyBorder(30, 100, 30, 100));
31
32         //username
33         JPanel usernamepln = new JPanel(new GridLayout(2,1));
34         usernamepln.setBackground(Color.CYAN);
35         JLabel usernamelbl = new JLabel("Username");
36         usernamelbl.setHorizontalAlignment(JLabel.LEADING);
37         usernamepln.add(usernamelbl);
38
39         JTextField usernameTF = new JTextField();
40         usernamepln.add(usernameTF);
41         panel1.add(usernamepln);
42
43         //firstname
44
45         JPanel firstpln = new JPanel(new GridLayout(2,1));
46         firstpln.setBackground(Color.CYAN);
47         JLabel firstlbl = new JLabel("First Name");
48         firstlbl.setHorizontalAlignment(JLabel.LEADING);
49         firstpln.add(firstlbl);
50
51         JTextField firstTA = new JTextField();
52         firstpln.add(firstTA);
53         panel1.add(firstpln);
54
55         //lastname
56
57         JPanel lastpln = new JPanel(new GridLayout(2,1));
58         lastpln.setBackground(Color.CYAN);
59         JLabel lastlbl = new JLabel("Last Name");
60         lastlbl.setHorizontalAlignment(JLabel.LEADING);
61         lastpln.add(lastlbl);
62
63         JTextField lastTA = new JTextField();
64         lastpln.add(lastTA);
65         panel1.add(lastpln);
66
67         //email
68
69         JPanel emailpln = new JPanel(new GridLayout(2,1));
70         emailpln.setBackground(Color.CYAN);
71         JLabel emaillbl = new JLabel("Email");
72         emaillbl.setHorizontalAlignment(JLabel.LEADING);
73         emailpln.add(emaillbl);
74
75         JTextField emailTF = new JTextField();
76         emailpln.add(emailTF);
77
78         panel1.add(emailpln);
79
80         this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
81         this.setTitle("Register");
82         this.setContentPane(panel1);
83         this.pack();
84         this.setVisible(true);
85     }
86
87     public static void main(String[] args) {
88         new Register();
89     }
90 }
```

The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** eclipse-workspace - Project_BA11_BAD/src/pokemonNK/Register.java - Eclipse IDE
- Menu Bar:** File Edit Source Refactor Navigate Search Project Run Window Help
- Left Sidebar (Package Explorer):** Shows the project structure with packages like AmorePOS, BADSession8, Class_Lab, Exercise_Class, Kelas_Tambahan, LathanJTS, PengumpulanTugas, Pertemuan2Lab, Presentasi_BAD_Group2, Project, and Project_BA11_BAD.
- Central Editor:** Displays the `Register.java` file content. The code is a Java Swing application for user registration. It includes imports for `java.awt.GridLayout`, `java.awt.Label`, `java.awt.PasswordField`, `java.awt.Panel`, `java.awt.event.ActionListener`, and `java.awt.event.ActionEvent`. The class `Register` extends `JFrame` and contains methods for handling login and register actions. The `register()` method is annotated with `@Override` and uses `ActionEvent` parameters to get user input from `usernameTF`, `firstnameTA`, and `lastnameTA`.
- Bottom Status Bar:** Shows the current file is `Login.java`, the Java version is JavaSE-1.8, and the build path includes JRE System Library [JavaSE-1.8].
- Bottom Taskbar:** Includes icons for Problems, Javadoc, Declaration, Coverage, Console, and a search bar.

```
116     //pass
117
118     JPanel passPanel = new JPanel(new GridLayout(2,1));
119     passPanel.setBackground(Color.CYAN);
120     JLabel passlbl = new JLabel("Password");
121     passlbl.setHorizontalTextPosition(JLabel.LEADING);
122     passPanel.add(passlbl);
123
124     JTextField passTA = new JTextField();
125     passPanel.add(passTA);
126     panel1.add(passPanel);
127
128     //cpass
129
130     JPanel cpassPanel = new JPanel(new GridLayout(2,1));
131     cpassPanel.setBackground(Color.CYAN);
132     JLabel cpasslbl = new JLabel("Confirm Password");
133     cpasslbl.setHorizontalTextPosition(JLabel.LEADING);
134     cpassPanel.add(cpasslbl);
135
136     JTextField cpassTA = new JTextField();
137     cpassPanel.add(cpassTA);
138     panel1.add(cpassPanel);
139
140     //button
141     JPanel bPanel = new JPanel();
142     bPanel.setBackground(Color.CYAN);
143     JButton bregister = new JButton("Register");
144
145     JButton bBack = new JButton("Back to login");
146
147     bregister.addActionListener(new ActionListener() {
148
149         @Override
150         public void actionPerformed(ActionEvent e) {
151             String username = usernameTF.getText();
152             String firstname = firstnameTA.getText();
153             String lastname = lastnameTA.getText();
154         }
155     });
156 }
```

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "Project_BA11_BAD". The "src" folder contains packages like "AmorePOS", "BADSession8", "Class_Lab", "Exercise Class", "Kelas_Tambah", "Latihan", "LatihanUTS", "PengumpulanTugas", "Presentasi_Lab", and "Presentasi_BAD_Group2".
- Register.java:** The code is displayed in the main editor. It is a Java class with an overridden actionPerformed method. The code handles user input from various JTextFields and JSpinners, performs validation (e.g., password confirmation, age range), and inserts data into a database using JDBC. It also generates random roles ("Admin" or "Player").
- Bottom Status Bar:** Shows the current file is "Login.java", the path is "C:\Program Files\Java\jre1.8.0_291\bin\javaw.exe", and the date is "Jan 12, 2022 7:53:56 AM".

```
149 @Override
150     public void actionPerformed(ActionEvent e) {
151         String username = usernameTF.getText();
152         String firstname = firstTA.getText();
153         String lastname = lastTA.getText();
154         String nameITA = nameITA.getText();
155         String email = emailTA.getText();
156         Integer age = (Integer) ageSpinner.getValue();
157         String gender = getSelectedButtonText(genderRG);
158         String password = String.valueOf(passTA.getPassword());
159         String cpassword = String.valueOf(cpassTA.getPassword());
160         Integer userRole = randomNum();
161
162         if (username.isEmpty() || password.isEmpty() || email.isEmpty() || firstname.isEmpty() || lastname.isEmpty() || gender.isEmpty()
163             || !password.equals(cpassword)) {
164             JOptionPane.showMessageDialog(null, "User should fill all the field!");
165         } else if (password == cpassword) {
166             JOptionPane.showMessageDialog(null, "Password and confirm password should be the same value!");
167         } else if (age < 11) {
168             JOptionPane.showMessageDialog(null, "Age must more than equal to 11");
169         } else if (!(email.contains("@")) || !(email.contains(".com"))) {
170             JOptionPane.showMessageDialog(null, "Email should use @ and .com");
171         } else {
172             String query = "INSERT INTO user VALUES ('" + userId + "', '" + username + "', '" + nameITA + "', '" + age + "', '" + email + "', '" + gender + "', "
173             + userRole + ")";
174             con.executeUpdate(query);
175
176             usernameTF.setText("");
177             firstTA.setText("");
178             lastTA.setText("");
179             nameITA.setText("");
180             emailTA.setText("");
181             passTA.setText("");
182             cpassTA.setText("");
183
184             Random rd = new Random();
185             String[] arr = {"Admin", "Player"};
186             // randomly selects an index from the arr
187             int select = rd.nextInt(arr.length);
188             roleTA.setText(select);
```

eclipse-workspace - Project_BA11_BAD/src/pokemonNK/Register.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer Project Explorer Login.java Register.java

```
188 JOptionPane.showMessageDialog(null, "Register Success");
189 JOptionPane.showMessageDialog(null, "Welcome, " + name);
190 dispose();
191 MainForm mf = new MainForm();
192
193 }
194 }
195 }
196 }
197 });
198 });
199 bBack.addActionListener(new ActionListener() {
200
201     @Override
202     public void actionPerformed(ActionEvent e) {
203         dispose();
204         Login l = new Login();
205
206     }
207 });
208 });
209 bPanel.add(bregist);
210 bPanel.add(bBack);
211 panel1.add(bPanel);
212
213 add(panel1);
214 init();
215
216 }
217
218 }
219 }
220 }
221 Integer randomNum() {
222
223     Integer random, j;
224     return random = randomInteger(0, 100);
225 }
226
```

Problems Javadoc Declaration Coverage Console

Login (1) [Java Application] C:\Program Files\Java\jre1.8.0_291\bin\javaw.exe (Jan 12, 2022 7:53:56 AM)

Type here to search

```

eclipse-workspace - Project_BA11_BAD/src/pokemonNK/Register.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer Project Explorer
Login.java Register.java
1 220
2 Integer randomNum() {
3     221=         Integer random, j;
4     222=         return random = randomInteger(0, 100);
5     223=     }
6     224=     int randomInteger(int min, int max) {
7     225=         225=             return min + rd.nextInt(max - min + 1);
8     226=     }
9     227= }
10    228=     229=         public String getSelectedButtonText(ButtonGroup buttonGroup) {
11        230=             for (Enumeration<AbstractButton> buttons = buttonGroup.getElements(); buttons.hasMoreElements();) {
12            231=                 AbstractButton button = buttons.nextElement();
13            232=                     if (button.isSelected()) {
14            233=                         return button.getText();
15            234=                     }
16            235=                 return null;
17        236=     }
18        237=     238=         public void init() {
19        239=             setTitle("PokemonNK");
20        240=             this.setSize(600, 400);
21        241=             this.setLocationRelativeTo(null);
22        242=             this.setDefaultCloseOperation(EXIT_ON_CLOSE);
23        243=             this.setVisible(true);
24    244= }
25    245=     256=         public static void main(String[] args) {
26    257=             258=         }
27    259=     }
28    260= }
29    261= }
30    262= }
31    263= }
32    264= }
33    265= }
34    266= }
35    267= }
36    268= }
37    269= }
38    270= }
39    271= }
40    272= }
41    273= }
42    274= }
43    275= }
44    276= }
45    277= }
46    278= }
47    279= }
48    280= }
49    281= }
50    282= }
51    283= }
52    284= }
53    285= }
54    286= }
55    287= }
56    288= }
57    289= }
58    290= }
59    291= }
60    292= }
61    293= }
62    294= }
63    295= }
64    296= }
65    297= }
66    298= }
67    299= }
68    300= }
69    301= }
70    302= }
71    303= }
72    304= }
73    305= }
74    306= }
75    307= }
76    308= }
77    309= }
78    310= }
79    311= }
80    312= }
81    313= }
82    314= }
83    315= }
84    316= }
85    317= }
86    318= }
87    319= }
88    320= }
89    321= }
90    322= }
91    323= }
92    324= }
93    325= }
94    326= }
95    327= }
96    328= }
97    329= }
98    330= }
99    331= }
100   332= }
101   333= }
102   334= }
103   335= }
104   336= }
105   337= }
106   338= }
107   339= }
108   340= }
109   341= }
110   342= }
111   343= }
112   344= }
113   345= }
114   346= }
115   347= }
116   348= }
117   349= }
118   350= }
119   351= }
120   352= }
121   353= }
122   354= }
123   355= }
124   356= }
125   357= }
126   358= }
127   359= }
128   360= }
129   361= }
130   362= }
131   363= }
132   364= }
133   365= }
134   366= }
135   367= }
136   368= }
137   369= }
138   370= }
139   371= }
140   372= }
141   373= }
142   374= }
143   375= }
144   376= }
145   377= }
146   378= }
147   379= }
148   380= }
149   381= }
150   382= }
151   383= }
152   384= }
153   385= }
154   386= }
155   387= }
156   388= }
157   389= }
158   390= }
159   391= }
160   392= }
161   393= }
162   394= }
163   395= }
164   396= }
165   397= }
166   398= }
167   399= }
168   400= }
169   401= }
170   402= }
171   403= }
172   404= }
173   405= }
174   406= }
175   407= }
176   408= }
177   409= }
178   410= }
179   411= }
180   412= }
181   413= }
182   414= }
183   415= }
184   416= }
185   417= }
186   418= }
187   419= }
188   420= }
189   421= }
190   422= }
191   423= }
192   424= }
193   425= }
194   426= }
195   427= }
196   428= }
197   429= }
198   430= }
199   431= }
200   432= }
201   433= }
202   434= }
203   435= }
204   436= }
205   437= }
206   438= }
207   439= }
208   440= }
209   441= }
210   442= }
211   443= }
212   444= }
213   445= }
214   446= }
215   447= }
216   448= }
217   449= }
218   450= }
219   451= }
220   452= }
221   453= }
222   454= }
223   455= }
224   456= }
225   457= }
226   458= }
227   459= }
228   460= }
229   461= }
230   462= }
231   463= }
232   464= }
233   465= }
234   466= }
235   467= }
236   468= }
237   469= }
238   470= }
239   471= }
240   472= }
241   473= }
242   474= }
243   475= }
244   476= }
245   477= }
246   478= }
247   479= }
248   480= }
249   481= }
250   482= }
251   483= }
252   484= }
253   485= }
254   486= }
255   487= }
256   488= }
257   489= }

```

- Ketika user mengklik Button ‘Register’ maka user akan berpindah ke halaman register
- Disini user diminta untuk input ‘Username, First Name, Last Name, Email, Gender, Age, Password, dan Confirm Password
- Jika user mengklik button ‘Register’ tanpa menginput data, maka sistem akan memunculkan “User should fill all the field”
- Jika user menginput password dan Confirm password dengan berbeda kata, maka user harus mengulangi pengisianya lagi dan memunculkan “Password and confirm password should be the same value!”
- Jika user menginput umur kurang dari 11 tahun, maka sistem akan memunculkan “Age must more than equal to 11”
- Disini user wajib menggunakan “@” dan “.com” apabila tidak menggunakan akan muncul “Email should use "@" and ".com""

3. MainForm (Menu)

A screenshot of the Eclipse IDE interface. The top bar shows the title "eclipse-workspace - Project_BA11_BAD/src/pokemonNK/MainForm.java - Eclipse IDE" and the system tray with icons for battery, signal, and date/time (1047 12/01/2022). The main window has a toolbar with icons for file operations, search, and help. Below the toolbar is a menu bar with File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help. The left side features two view panes: "Package Explorer" and "Project Explorer". The "Package Explorer" pane lists several Java packages and their contents, including "Pertemuan9", "Exercise Class", "Kelas_Tambah", "Latihan", "LatihanNTS", "PengumpulanTugas", "Pertemuan2_Lab", "Presentasi_BAD_Group2_1", and "Project_BA11_BAD". The "Project Explorer" pane shows the "src" folder containing "DBConnector", "Main.java", and "membermaker.java". The right side of the screen contains a large code editor window displaying Java code for "MainForm.java". The code includes imports for JPanel, JPopupMenu, and JLabel, and defines a main method and a setMenu() method for creating a menu bar with various items like User, Adventure, Transaction History, Manage, Logout, and Manage Pokemon. At the bottom, there are tabs for Problems, Javadoc, Declaration, Search, Coverage, and Console, along with a status bar showing "Writable", "Smart Insert", "1:1:0", and "28°C".

eclipse-workspace - Project_BA11_BAD/src/pokemonNK/MainForm.java - Eclipse IDE

```

71 userMenu.setMnemonic('U');
72 adventureMenu.setMnemonic('A');
73 transactionMenu.setMnemonic('T');
74 manageMenu.setMnemonic('M');

75 milogout.setMnemonic('L');
76 miPokemonMarket.setMnemonic('P');
77 miBag.setMnemonic('B');
78 miViewHistory.setMnemonic('H');
79 miManagePokemon.setMnemonic('J');

80 milogout.addActionListener(this);
81 userMenu.add(milogout);
82 adventureMenu.add(miPokemonMarket);
83 transactionMenu.add(miBag);
84 manageMenu.add(miManagePokemon);
85 transactionMenu.add(miViewHistory);
86 userMenu.addMenuItemListener(this);
87 adventureMenu.addMenuItemListener(this);
88 transactionMenu.addMenuItemListener(this);
89 manageMenu.addMenuItemListener(this);
90 milogout.addActionListener(this);
91 userMenu.addMenuItemListener(this);
92 transactionMenu.addMenuItemListener(this);
93 milogout.addActionListener(this);
94 miPokemonMarket.addActionListener(this);
95 miBag.addActionListener(this);
96 miManagePokemon.addActionListener(this);
97 miViewHistory.addActionListener(this);
98 ...

```

Problems Javadoc Declaration Search Coverage Console

setupTable() - 0 references in workspace (0 matches filtered from view)

Type here to search

Writable Smart Insert 1:1:0 1047 12/01/2022

eclipse-workspace - Project_BA11_BAD/src/pokemonNK/MainForm.java - Eclipse IDE

```

98 role();
99 role();
100 setJMenuBar(menuBar);
101 JPanel mainPanel = new JPanel();
102 JLabel pokemonNK = new JLabel("PokemonNK");
103 mainPanel.add(pokemonNK);
104 ...
105 }
106 }
107 }
108 private void role() {
109 if (role.equals("Player")) {
110 menuBar.add(userMenu);
111 menuBar.add(adventureMenu);
112 menuBar.add(transactionMenu);
113 setTitle("Welcome User");
114 } else if (role.equals("Admin")) {
115 menuBar.add(userMenu);
116 menuBar.add(manageMenu);
117 setTitle("Welcome Admin");
118 }
119 }
120 private void role1() {
121 if (role1.equals("Player")) {
122 menuBar.add(userMenu);
123 menuBar.add(adventureMenu);
124 menuBar.add(transactionMenu);
125 ...

```

Problems Javadoc Declaration Search Coverage Console

setupTable() - 0 references in workspace (0 matches filtered from view)

Type here to search

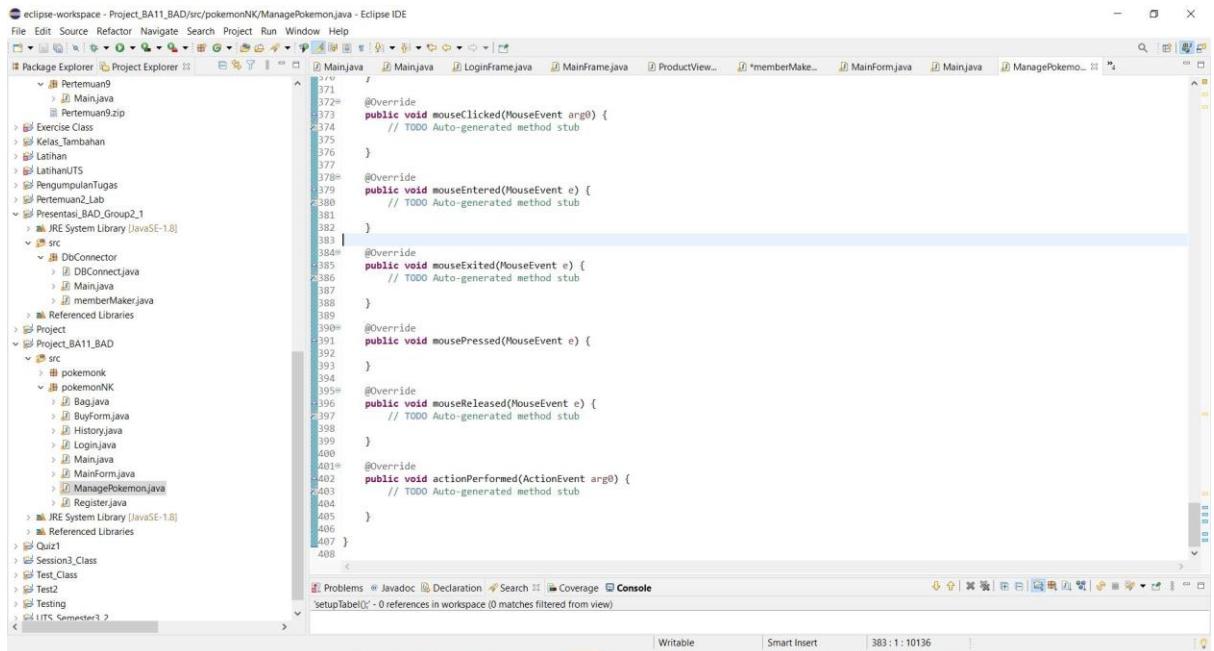
Writable Smart Insert 1:1:0 1047 12/01/2022

The screenshot shows the Eclipse IDE interface. The top bar displays the title "eclipse-workspace - Project_BA11_BAD/src/pokemonNK/MainForm.java - Eclipse IDE". The menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help. The toolbar contains various icons for file operations like Open, Save, Cut, Copy, Paste, Find, and Delete. The left-hand sidebar is the Package Explorer, listing projects, source files, and Java classes. The main workspace shows the code for `MainForm.java` with syntax highlighting and line numbers. The bottom status bar shows "Writable", "Smart Insert", and "1:1:0". The taskbar at the bottom includes icons for File, Open, Save, Cut, Copy, Paste, Find, Delete, and others.

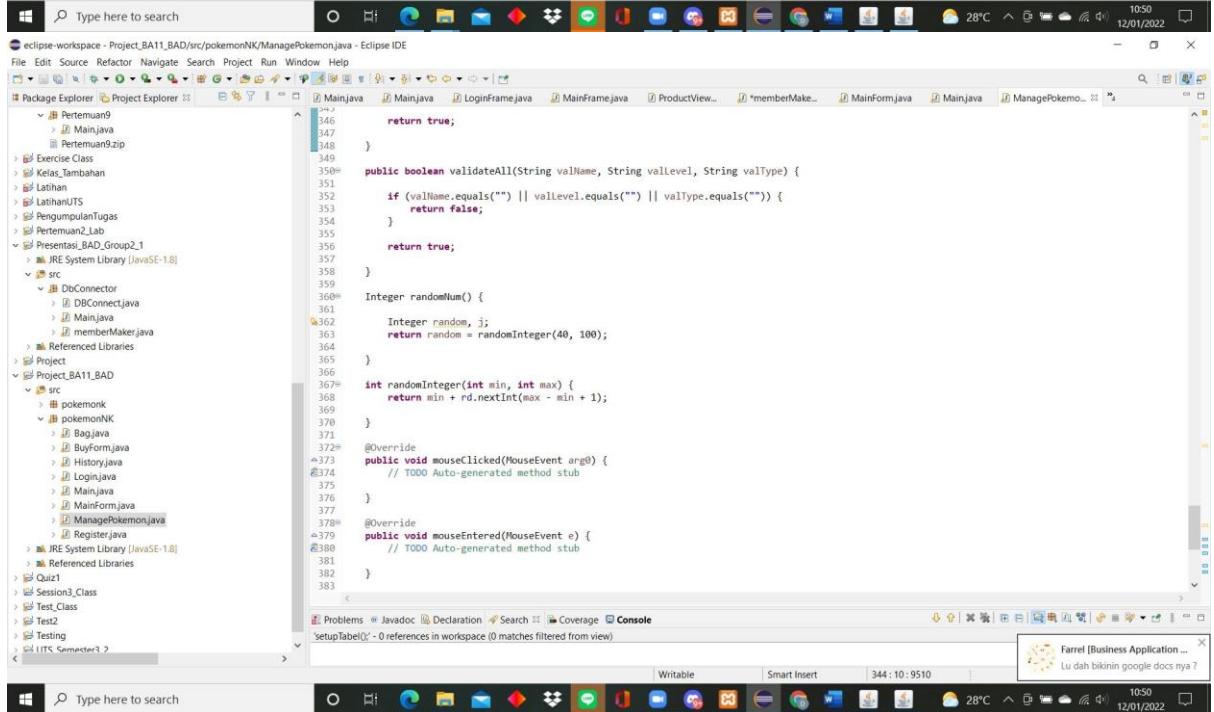
```
149@ Override  
150    public void menuSelected(MenuEvent e) {  
151        }  
152    }  
153  
154@ Override  
155    public void actionPerformed(ActionEvent arg0) {  
156        if (arg0.getSource() == logout) {  
157            this.dispose();  
158            Login lf = new Login();  
159  
160        } else if (arg0.getSource() == miPokemonMarket) {  
161            this.dispose();  
162            BuyForm bf = new BuyForm();  
163  
164        } else if (arg0.getSource() == miManagePokemon) {  
165            this.dispose();  
166            ManagePokemon mp = new ManagePokemon();  
167        } else if (arg0.getSource() == miBag) {  
168            this.dispose();  
169            Bag b = new Bag();  
170        } else if (arg0.getSource() == miViewHistory) {  
171            this.dispose();  
172            History h = new History();  
173        }  
174    }  
175  
176    public static void main(String[] args) {  
177        //  
178    }  
179}
```

- Ketika User berhasil masuk dan memiliki role sebagai ‘Player’, maka akan muncul dibelakangnya kata ‘PokemoNK’
 - user dapat memilih berbagai menu, mulai dari menu ‘Pokemon Market, Bag, dan View Transaction History’
 - Jika User berhasil masuk dan memiliki role sebagai ‘Admin’, maka user hanya memiliki menu ‘Manage Pokemon’

4. Manage pokemon role : admin



```
 372@override
 373    public void mouseClicked(MouseEvent arg0) {
 374        // TODO Auto-generated method stub
 375    }
 376
 377@override
 378    public void mouseEntered(MouseEvent e) {
 379        // TODO Auto-generated method stub
 380    }
 381
 382}
 383
 384@override
 385    public void mouseExited(MouseEvent e) {
 386        // TODO Auto-generated method stub
 387    }
 388
 389
 390@override
 391    public void mousePressed(MouseEvent e) {
 392    }
 393
 394
 395@override
 396    public void mouseReleased(MouseEvent e) {
 397        // TODO Auto-generated method stub
 398    }
 399
 400
 401@override
 402    public void actionPerformed(ActionEvent arg0) {
 403        // TODO Auto-generated method stub
 404
 405    }
 406
 407}
 408
```



```
 346    return true;
 347}
 348}
 349
 350public boolean validateAll(String valName, String valLevel, String valType) {
 351    if (valName.equals("") || valLevel.equals("") || valType.equals("")) {
 352        return false;
 353    }
 354
 355    return true;
 356}
 357
 358}
 359
 360Integer randomNum() {
 361    Integer random, j;
 362    random = randomInteger(40, 100);
 363
 364}
 365
 366int randomInteger(int min, int max) {
 367    return min + rd.nextInt(max - min + 1);
 368}
 369
 370
 371@Override
 372    public void mouseClicked(MouseEvent arg0) {
 373        // TODO Auto-generated method stub
 374    }
 375
 376
 377@override
 378    public void mouseEntered(MouseEvent e) {
 379        // TODO Auto-generated method stub
 380    }
 381
 382}
 383
```

eclipse-workspace - Project_BA11_BAD/src/pokemonNK/ManagePokemon.java - Eclipse IDE

```

try {
    data.clear();
    while (rs.next()) {
        String pokemonid = rs.getString(1);
        String name = rs.getString(2);
        String level = rs.getString(3);
        String type = rs.getString(4);

        Vector<String> vecRead = new Vector<String>();

        vecRead.add(pokemonid);
        vecRead.add(name);
        vecRead.add(level);
        vecRead.add(type);

        data.add(vecRead);
    }
    dtmManage.fireTableDataChanged();
} catch (SQLException e) {
    e.printStackTrace();
}

public boolean validateNumber(String valNumb) {
    char[] arraylevel = valNumb.toCharArray();
    for (char i : arraylevel) {
        if (!Character.isDigit(i)) {
            return false;
        }
    }
}

```

Writable | Smart Insert | 306 : 8815 | 1050 | 12/01/2022

eclipse-workspace - Project_BA11_BAD/src/pokemonNK/ManagePokemon.java - Eclipse IDE

```

else if (sLevel.equals("0")) {
    JOptionPane.showMessageDialog(null, "Level must more than equal to 1");
} else if (validateNumber(sLevel) == false) {
    JOptionPane.showMessageDialog(null, "Level must be a number!");
} else {
    String query = "UPDATE pokemon SET PokemonName = '" + sName + "', PokemonLevel = '" +
        + Integer.parseInt(sLevel) + "', PokemonType = '" + sType + "' WHERE PokemonId = '" + Id +
        + "'";
    con.executeUpdate(query);
    setupTable();
    JOptionPane.showMessageDialog(null, "Update Success!");
}

```

Writable | Smart Insert | 266 : 81 : 7880 | 1050 | 12/01/2022

eclipse-workspace - Project_BA11_BAD/src/pokemonNK/ManagePokemon.java - Eclipse IDE

```

154 JTextField UnameTF = new JTextField();
155 UnamePanel.add(UnameTF);
156 updatePanel.add(UnamePanel);
157
158 JPanel UlevelPanel = new JPanel(new GridLayout(1, 1));
159 UlevelPanel.setBackground(Color.CYAN);
160 JLabel UlevelLabel = new JLabel("Pokemon Level: ");
161 UlevelLabel.setHorizontalAlignment(JLabel.LEADING);
162 UlevelPanel.add(UlevelLabel);
163
164 JTextField UlevelTF = new JTextField();
165 UlevelPanel.add(UlevelTF);
166 updatePanel.add(UlevelPanel);
167
168 JPanel UtpePanel = new JPanel(new GridLayout(1, 1));
169 UtpePanel.setBackground(Color.CYAN);
170 JLabel UtpeLabel = new JLabel("Pokemon Type: ");
171 UtpeLabel.setHorizontalAlignment(JLabel.LEADING);
172 UtpePanel.add(UtpeLabel);
173
174 JTextField UtpeTF = new JTextField();
175 UtpePanel.add(UtpeTF);
176 updatePanel.add(UtpePanel);
177
178 JButton blupdate = new JButton("Update");
179 updatePanel.add(blupdate);
180
181 midPanel.add(insertPanel, BorderLayout.WEST);
182 midPanel.add(deletePanel, BorderLayout.CENTER);
183 midPanel.add(updatePanel, BorderLayout.EAST);
184
185 // Lower Panel
186 bPanel = new JPanel();
187 bPanel.setBackground(Color.CYAN);
188 bBack = new JButton("Back to Menu");
189 lowerPanel.add(bBack);
190
191 // Main Panel

```

Problems Javadoc Declaration Search Coverage Console

setupTable() - 0 references in workspace (0 matches filtered from view)

Writable Smart Insert 155 : 33 : 4574 1049 12/01/2022

eclipse-workspace - Project_BA11_BAD/src/pokemonNK/ManagePokemon.java - Eclipse IDE

```

118 JPanel insertPanel = new JPanel(new GridLayout(1, 1));
119 JButton binsert = new JButton("Insert");
120 insertPanel.add(binsert);
121
122 // Delete
123 JPanel DidPanel = new JPanel(new GridLayout(1, 1));
124 DidPanel.setBackground(Color.CYAN);
125 JLabel DidLabel = new JLabel("Pokemon ID: ");
126 DidLabel.setHorizontalAlignment(JLabel.LEADING);
127 DidPanel.add(DidLabel);
128
129 JTextField DidTF = new JTextField();
130 DidPanel.add(DidTF);
131 deletePanel.add(DidPanel);
132
133 JButton bDelete = new JButton("Delete");
134 deletePanel.add(bDelete);
135
136 // Update
137 JPanel UidPanel = new JPanel(new GridLayout(1, 1));
138 UidPanel.setBackground(Color.CYAN);
139 JLabel UidLabel = new JLabel("Pokemon ID: ");
140 UidLabel.setHorizontalAlignment(JLabel.LEADING);
141 UidPanel.add(UidLabel);
142
143 JTextField UidTF = new JTextField();
144 UidPanel.add(UidTF);
145 updatePanel.add(UidPanel);
146
147 JPanel UnamePanel = new JPanel(new GridLayout(1, 1));
148 UnamePanel.setBackground(Color.CYAN);
149 JLabel UnameLabel = new JLabel("Pokemon Name: ");
150 UnameLabel.setHorizontalAlignment(JLabel.LEADING);
151 UnamePanel.add(UnameLabel);
152
153 JTextField UnameTF = new JTextField();
154 UnamePanel.add(UnameTF);
155

```

Problems Javadoc Declaration Search Coverage Console

setupTable() - 0 references in workspace (0 matches filtered from view)

Writable Smart Insert 118 : 37 : 3448 1049 12/01/2022

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure with packages like `MainJava`, `MemberMak...`, and `ManagePoko...`.
- Code Editor:** Displays the `Main.java` file content, which includes imports for `java.awt`, `java.awt.GridLayout`, `java.awt.BorderLayout`, and `java.awt.Color`. The code itself creates panels, labels, and text fields for a user interface.
- Bottom Bar:** Includes tabs for `Problems`, `Javadoc`, `Declaration`, `Search`, `Coverage`, and `Console`. The `Console` tab is active, showing the message: `setupTable(): - 0 references in workspace (0 matches filtered from view)`.
- System Tray:** Shows system icons for battery, network, and date/time (10:49, 28°C, 12/01/2022).

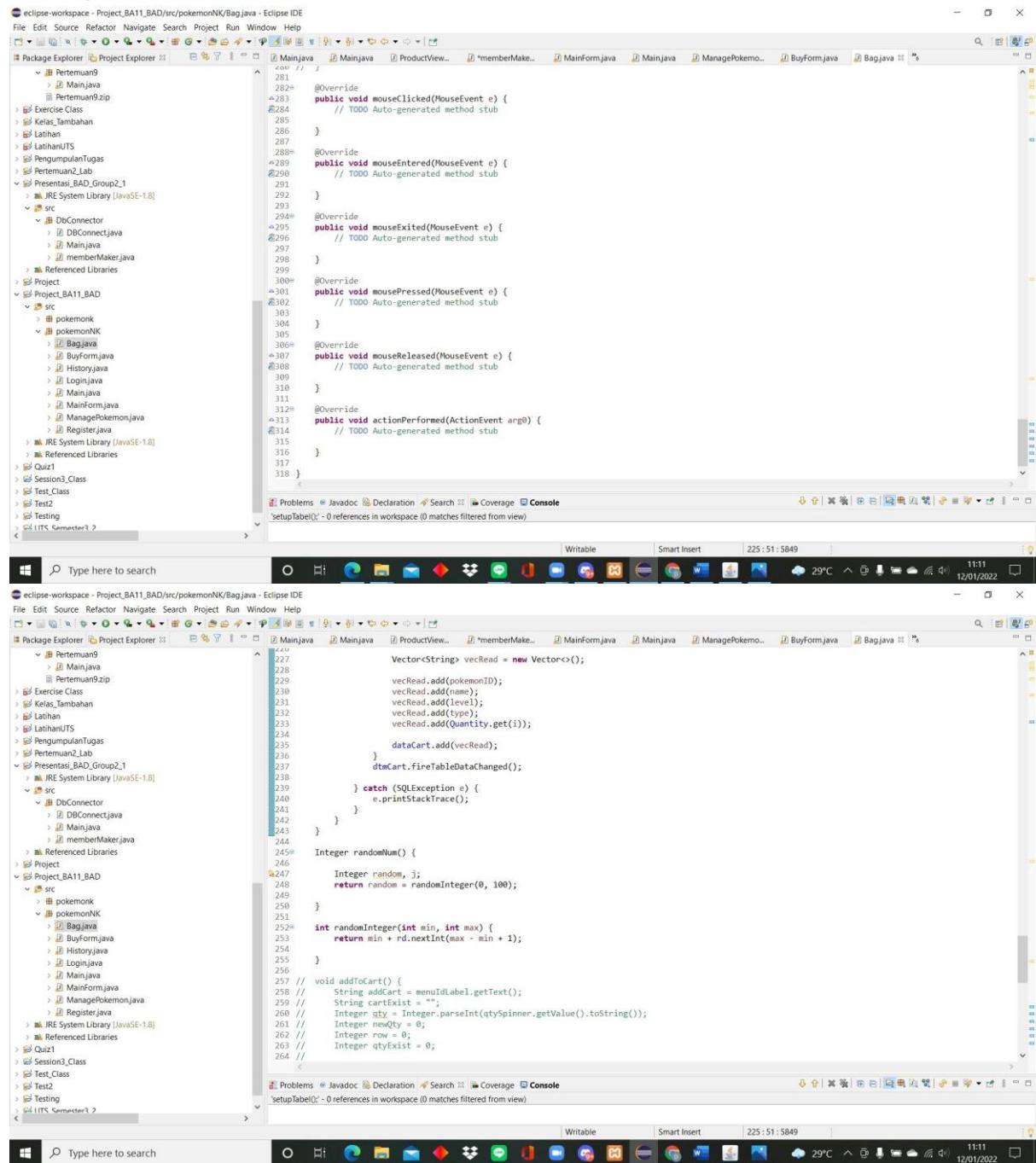
The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "Project_BA11_BAD". It includes packages like "Permenuan" and "Exercise Class", and classes such as "MainJava", "ManagePokemon", "LoginFramejava", "MainFramejava", "ProductView...", "memberMake...", "MainForm.java", "Main.java", and "ManagePoko...".
- Editor:** The main window displays the code for "ManagePokemon.java". The code implements a JFrame and extends ActionListener and MouseListener. It initializes panels, adds components like JTable and JButton, and sets up a DefaultTableModel. A JTable named "menuTable" is populated with data from a Vector and a Random object.
- Bottom Bar:** Includes tabs for "Problems", "Javadoc", "Declaration", "Search", "Coverage", and "Console". The "Console" tab is active, showing the message: "setupTable(): - 0 references in workspace (0 matches filtered from view)".
- System Tray:** Shows the date and time as "12/01/2022 10:49".

- Ketika user terpilih dan memakai role sebagai Admin, maka user dapat melakukan manage pokemon dengan cara menekan menu ‘Manage Pokemon’
 - Disini user dapat melakukan ‘Insert’, ‘Delete’, dan ‘Update’
 - Jika User ingin melakukan ‘Insert’, maka user cukup input nama, level, dan tipe pokemon yang ingin diinput. Kemudian sistem secara otomatis memilih ID, pokemon yang akan di insert
 - Jika User ingin melakukan ‘Delete’, maka user cukup input ID pokemonnya saja, kemudian tekan tombol ‘Delete’ untuk menghilangkan data pokemon tersebut

- Jika User ingin melakukan ‘Update’, maka user cukup input id pokemon dan merubah nama, level, dan tipe pokemon yang sudah ada di dalam tabel. Ketika sudah diinput, maka user cukup menekan tombol ‘Update’. Sistem akan secara otomatis merubah nama, level, dan tipe pokemon tersebut

5. Bag Pokemon (Role : Member)



The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "Project_BA11_BAD".
- Code Editor:** Displays the Java code for the `Bag.java` class.
- Code Snippet:**

```
    281     @Override
    282     public void mouseClicked(MouseEvent e) {
    283         // TODO Auto-generated method stub
    284     }
    285
    286     @Override
    287     public void mouseEntered(MouseEvent e) {
    288         // TODO Auto-generated method stub
    289     }
    290
    291     @Override
    292     public void mouseExited(MouseEvent e) {
    293         // TODO Auto-generated method stub
    294     }
    295
    296     @Override
    297     public void mousePressed(MouseEvent e) {
    298         // TODO Auto-generated method stub
    299     }
    300
    301     @Override
    302     public void mouseReleased(MouseEvent e) {
    303         // TODO Auto-generated method stub
    304     }
    305
    306     @Override
    307     public void actionPerformed(ActionEvent arg0) {
    308         // TODO Auto-generated method stub
    309     }
    310
    311     @Override
    312     public void actionPerformed(ActionEvent arg0) {
    313         // TODO Auto-generated method stub
    314     }
    315
    316     }
    317
    318 }
```
- Bottom Status Bar:** Shows the date (12/01/2022), time (11:11), and system temperature (29°C).

eclipse-workspace - Project_BA11_BAD/src/pokemonNK/Bag.java - Eclipse IDE

```
188     }
189     dataCart.removeAllElements();
190     dtmCart.fireTableDataChanged();
191   }
192 }
193 }
194
195   return bottomPanel;
196 }
197
198 void showCart() {
199   cartTable.setBackground(Color.CYAN);
200
201   for (int i = 0; i < ID.size(); i++) {
202     for (int j = 0; j < ID.get(i).size(); j++) {
203       if (ID.get(i).containsIgnoreCase(ID.get(j)) > 0) {
204         String tempID = ID.get(j);
205         ID.set(i, ID.get(j));
206         ID.set(j, tempID);
207
208         String tempQty = Quantity.get(i);
209         Quantity.set(i, Quantity.get(j));
210         Quantity.set(j, tempQty);
211
212       }
213     }
214   }
215   String query = "SELECT PokemondId, PokemonName, PokemonLevel, PokemonType FROM Pokemon WHERE PokemondId = "
216   + ID.get(i) + "";
217   ResultSet rs = con.executeQuery(query);
218
219   try {
220
221     while (rs.next()) {
222       String pokemondId = rs.getString(1);
223       String name = rs.getString(2);
224       String level = rs.getString(3);
225       String type = rs.getString(4);
226
227     }
228   }
229 }
```

eclipse-workspace - Project_BA11_BAD/src/pokemonNK/Bag.java - Eclipse IDE

```
149 private JPanel setBottomPanel() {
150   bottomPanel = new JPanel(new GridLayout(1, 1));
151   bottomPanel.setBorder(new EmptyBorder(10, 10, 10, 10));
152   bottomPanel.setBackground(Color.CYAN);
153
154   JButton bCheck = new JButton("Checkout");
155   JButton bBack = new JButton("Back to Main");
156
157   bottomPanel.add(bCheck);
158   bottomPanel.add(bBack);
159
160   bBack.addActionListener(new ActionListener() {
161
162     @Override
163     public void actionPerformed(ActionEvent e) {
164       dispose();
165       MainForm mf = new MainForm();
166
167     }
168   });
169
170   bCheck.addActionListener(new ActionListener() {
171
172     @Override
173     public void actionPerformed(ActionEvent e) {
174       Integer transactionId = 0;
175       Integer id = 1;
176       Integer userId = 1;
177       LocalDateTime transactionTime = LocalDateTime.now();
178
179       String queryTH = "INSERT INTO headertransaction VALUES ('" + transactionId + "', '" + userId + "', '"
180       + transactionTime + ")";
181       con.executeUpdate(queryTH);
182       for (int i = 0; i < cartTable.getRowCount(); i++) {
183         String queryTD = "INSERT INTO detailtransaction VALUES ('" + id + "', '" + cartTable.getValueAt(i, 0) + "', '"
184         + "", "' + cartTable.getValueAt(i, 4) + ')";
185         con.executeUpdate(queryTD);
186
187     }
188   }
189 }
```

The screenshot shows the Eclipse IDE interface with the following details:

- Project Structure:** The left sidebar displays the project structure under "Project_BA11_BAD". It includes packages like "Perlemen9", "Exercise Class", "Lathian", "PengumpulanTugas", "Perfeman9_Lan", "Presentasi_BAD_Group_2_1", and "src" which contains "DbConnector", "Main.java", and "memberMaker.java".
- Main.java Code:** The main code editor window shows the implementation of the `actionPerformed` method. The code iterates through a table, checks if the value in column 0 matches a remove value, and if so, removes the row from a data cart.
- Bottom Bar:** The Eclipse toolbar at the bottom includes icons for Problems, Javadoc, Declaration, Search, Coverage, Console, and others.

The screenshot shows the Eclipse IDE interface. The top bar displays the title "eclipse-workspace - Project_BA11_BAD/src/pokemonNK/Bag.java - Eclipse IDE" and system status like "11:11 12/01/2022". The menu bar includes File, Edit, Source, Refactor, Navigate, Project, Run, Window, Help. The left sidebar shows the Package Explorer with a tree view of projects and files, including "Permentuan" (Main.java, Permentuan9.zip), "Exercise Class" (Kelas_Tambahan), "Latihan" (LatihanUTS), "PengumpulanTugas", "Permentuan2_Lab", "Presentasi_BAO_Group_2_1" (JRE System Library [JavaSE-1.8]), "src" (DbConnector, Main.java, member.java), "Referenced Libraries", "Project_BA11_BAD" (src/pokemonN, src/pokemonNK, Bag.java, BuyForm.java, History.java, Login.java, Main.java, MainForm.java, ManagePokemon.java, Register.java), and "JRE System Library [JavaSE-1.8]" (Referenced Libraries). The main workspace shows the code editor with Bag.java open, displaying methods for setting up top and center panels. The bottom status bar shows "Writable", "Smart Insert", "11:11 12/01/2022", "SetupTable(0) - 0 references in workspace (0 matches filtered from view)", and "75 : 1 : 1914".

```

eclipse-workspace - Project_BA11_BAD/src/pokemonNK/Bag.java - Eclipse IDE
File Edit Source Refactor Navigate Project Run Window Help
Main.java Main.java ProductView... *memberMake... MainForm.java Main.java ManagePokemo... BuyForm.java Bag.java
Package Explorer Project Explorer
Main.java
37 Vector<String> Quantity = BuyForm.vQuantity;
38 DBConnect con = DBConnect.getConnection();
39 public Bag() {
40     initFrame();
41     mainPanel = setMainPanel();
42     add(mainPanel);
43     showCart();
44 }
45 public static void main(String[] args) {
46     // TODO Auto-generated method stub
47 }
48 private void initFrame() {
49     setTitle("Pemang Bag");
50     setSize(900, 700);
51     setDefaultCloseOperation(EXIT_ON_CLOSE);
52     setIconImage(null);
53     setResizable(true);
54     setVisible(true);
55 }
56 private JPanel setMainPanel() {
57     topPanel = setTopPanel();
58     centerPanel = setCenterPanel();
59     bottomPanel = setBottomPanel();
60     mainPanel = new JPanel(new BorderLayout());
61     mainPanel.add(topPanel, BorderLayout.NORTH);
62     mainPanel.add(centerPanel, BorderLayout.CENTER);
63     mainPanel.add(bottomPanel, BorderLayout.SOUTH);
64 }
65 JPanel topPanel = new JPanel();
66 JPanel centerPanel = new JPanel();
67 JPanel bottomPanel = new JPanel();
68 mainPanel.setLayout(new BorderLayout());
69 topPanel.setLayout(new BorderLayout());
70 centerPanel.setLayout(new BorderLayout());
71 bottomPanel.setLayout(new BorderLayout());
72 topPanel.add(centerPanel, BorderLayout.CENTER);
73 centerPanel.add(bottomPanel, BorderLayout.SOUTH);
74 bottomPanel.add(mainPanel, BorderLayout.CENTER);
75
76 JButton addBtn = new JButton("Add");
77 JButton updateBtn = new JButton("Update");
78 JButton removeBtn = new JButton("Remove");
79 JButton cancelBtn = new JButton("Cancel");
80 JButton finishBtn = new JButton("Finish");
81
82 JTextField idLabel = new JTextField("ID");
83 JTextField nameLabel = new JTextField("Name");
84 JTextField levelLabel = new JTextField("Level");
85 JTextField typeLabel = new JTextField("Type");
86 JTextField quantityLabel = new JTextField("Quantity");
87 JTextField valueTotalLabel = new JTextField("Value Total");
88
89 JSpinner qtySpinner = new JSpinner();
90
91 JPanel panel = new JPanel();
92 panel.setLayout(new GridLayout(2, 2));
93 panel.add(idLabel);
94 panel.add(nameLabel);
95 panel.add(levelLabel);
96 panel.add(typeLabel);
97 panel.add(quantityLabel);
98 panel.add(valueTotalLabel);
99
100 panel.add(addBtn);
101 panel.add(updateBtn);
102 panel.add(removeBtn);
103 panel.add(cancelBtn);
104
105 JPanel panel2 = new JPanel();
106 panel2.setLayout(new GridLayout(1, 1));
107 panel2.add(panel);
108
109 JPanel panel3 = new JPanel();
110 panel3.setLayout(new GridLayout(1, 1));
111 panel3.add(panel2);
112
113 JPanel panel4 = new JPanel();
114 panel4.setLayout(new GridLayout(1, 1));
115 panel4.add(panel3);
116
117 JPanel panel5 = new JPanel();
118 panel5.setLayout(new GridLayout(1, 1));
119 panel5.add(panel4);
120
121 JPanel panel6 = new JPanel();
122 panel6.setLayout(new GridLayout(1, 1));
123 panel6.add(panel5);
124
125 JPanel panel7 = new JPanel();
126 panel7.setLayout(new GridLayout(1, 1));
127 panel7.add(panel6);
128
129 JPanel panel8 = new JPanel();
130 panel8.setLayout(new GridLayout(1, 1));
131 panel8.add(panel7);
132
133 JPanel panel9 = new JPanel();
134 panel9.setLayout(new GridLayout(1, 1));
135 panel9.add(panel8);
136
137 JPanel panel10 = new JPanel();
138 panel10.setLayout(new GridLayout(1, 1));
139 panel10.add(panel9);
140
141 JPanel panel11 = new JPanel();
142 panel11.setLayout(new GridLayout(1, 1));
143 panel11.add(panel10);
144
145 JPanel panel12 = new JPanel();
146 panel12.setLayout(new GridLayout(1, 1));
147 panel12.add(panel11);
148
149 JPanel panel13 = new JPanel();
150 panel13.setLayout(new GridLayout(1, 1));
151 panel13.add(panel12);
152
153 JPanel panel14 = new JPanel();
154 panel14.setLayout(new GridLayout(1, 1));
155 panel14.add(panel13);
156
157 JPanel panel15 = new JPanel();
158 panel15.setLayout(new GridLayout(1, 1));
159 panel15.add(panel14);
160
161 JPanel panel16 = new JPanel();
162 panel16.setLayout(new GridLayout(1, 1));
163 panel16.add(panel15);
164
165 JPanel panel17 = new JPanel();
166 panel17.setLayout(new GridLayout(1, 1));
167 panel17.add(panel16);
168
169 JPanel panel18 = new JPanel();
170 panel18.setLayout(new GridLayout(1, 1));
171 panel18.add(panel17);
172
173 JPanel panel19 = new JPanel();
174 panel19.setLayout(new GridLayout(1, 1));
175 panel19.add(panel18);
176
177 JPanel panel20 = new JPanel();
178 panel20.setLayout(new GridLayout(1, 1));
179 panel20.add(panel19);
180
181 JPanel panel21 = new JPanel();
182 panel21.setLayout(new GridLayout(1, 1));
183 panel21.add(panel20);
184
185 JPanel panel22 = new JPanel();
186 panel22.setLayout(new GridLayout(1, 1));
187 panel22.add(panel21);
188
189 JPanel panel23 = new JPanel();
190 panel23.setLayout(new GridLayout(1, 1));
191 panel23.add(panel22);
192
193 JPanel panel24 = new JPanel();
194 panel24.setLayout(new GridLayout(1, 1));
195 panel24.add(panel23);
196
197 JPanel panel25 = new JPanel();
198 panel25.setLayout(new GridLayout(1, 1));
199 panel25.add(panel24);
200
201 JPanel panel26 = new JPanel();
202 panel26.setLayout(new GridLayout(1, 1));
203 panel26.add(panel25);
204
205 JPanel panel27 = new JPanel();
206 panel27.setLayout(new GridLayout(1, 1));
207 panel27.add(panel26);
208
209 JPanel panel28 = new JPanel();
210 panel28.setLayout(new GridLayout(1, 1));
211 panel28.add(panel27);
212
213 JPanel panel29 = new JPanel();
214 panel29.setLayout(new GridLayout(1, 1));
215 panel29.add(panel28);
216
217 JPanel panel30 = new JPanel();
218 panel30.setLayout(new GridLayout(1, 1));
219 panel30.add(panel29);
220
221 JPanel panel31 = new JPanel();
222 panel31.setLayout(new GridLayout(1, 1));
223 panel31.add(panel30);
224
225 JPanel panel32 = new JPanel();
226 panel32.setLayout(new GridLayout(1, 1));
227 panel32.add(panel31);
228
229 JPanel panel33 = new JPanel();
230 panel33.setLayout(new GridLayout(1, 1));
231 panel33.add(panel32);
232
233 JPanel panel34 = new JPanel();
234 panel34.setLayout(new GridLayout(1, 1));
235 panel34.add(panel33);
236
237 JPanel panel35 = new JPanel();
238 panel35.setLayout(new GridLayout(1, 1));
239 panel35.add(panel34);
240
241 JPanel panel36 = new JPanel();
242 panel36.setLayout(new GridLayout(1, 1));
243 panel36.add(panel35);
244
245 JPanel panel37 = new JPanel();
246 panel37.setLayout(new GridLayout(1, 1));
247 panel37.add(panel36);
248
249 JPanel panel38 = new JPanel();
250 panel38.setLayout(new GridLayout(1, 1));
251 panel38.add(panel37);
252
253 JPanel panel39 = new JPanel();
254 panel39.setLayout(new GridLayout(1, 1));
255 panel39.add(panel38);
256
257 JPanel panel40 = new JPanel();
258 panel40.setLayout(new GridLayout(1, 1));
259 panel40.add(panel39);
260
261 JPanel panel41 = new JPanel();
262 panel41.setLayout(new GridLayout(1, 1));
263 panel41.add(panel40);
264
265 JPanel panel42 = new JPanel();
266 panel42.setLayout(new GridLayout(1, 1));
267 panel42.add(panel41);
268
269 JPanel panel43 = new JPanel();
270 panel43.setLayout(new GridLayout(1, 1));
271 panel43.add(panel42);
272
273 JPanel panel44 = new JPanel();
274 panel44.setLayout(new GridLayout(1, 1));
275 panel44.add(panel43);
276
277 JPanel panel45 = new JPanel();
278 panel45.setLayout(new GridLayout(1, 1));
279 panel45.add(panel44);
280
281 JPanel panel46 = new JPanel();
282 panel46.setLayout(new GridLayout(1, 1));
283 panel46.add(panel45);
284
285 JPanel panel47 = new JPanel();
286 panel47.setLayout(new GridLayout(1, 1));
287 panel47.add(panel46);
288
289 JPanel panel48 = new JPanel();
290 panel48.setLayout(new GridLayout(1, 1));
291 panel48.add(panel47);
292
293 JPanel panel49 = new JPanel();
294 panel49.setLayout(new GridLayout(1, 1));
295 panel49.add(panel48);
296
297 JPanel panel50 = new JPanel();
298 panel50.setLayout(new GridLayout(1, 1));
299 panel50.add(panel49);
300
301 JPanel panel51 = new JPanel();
302 panel51.setLayout(new GridLayout(1, 1));
303 panel51.add(panel50);
304
305 JPanel panel52 = new JPanel();
306 panel52.setLayout(new GridLayout(1, 1));
307 panel52.add(panel51);
308
309 JPanel panel53 = new JPanel();
310 panel53.setLayout(new GridLayout(1, 1));
311 panel53.add(panel52);
312
313 JPanel panel54 = new JPanel();
314 panel54.setLayout(new GridLayout(1, 1));
315 panel54.add(panel53);
316
317 JPanel panel55 = new JPanel();
318 panel55.setLayout(new GridLayout(1, 1));
319 panel55.add(panel54);
320
321 JPanel panel56 = new JPanel();
322 panel56.setLayout(new GridLayout(1, 1));
323 panel56.add(panel55);
324
325 JPanel panel57 = new JPanel();
326 panel57.setLayout(new GridLayout(1, 1));
327 panel57.add(panel56);
328
329 JPanel panel58 = new JPanel();
330 panel58.setLayout(new GridLayout(1, 1));
331 panel58.add(panel57);
332
333 JPanel panel59 = new JPanel();
334 panel59.setLayout(new GridLayout(1, 1));
335 panel59.add(panel58);
336
337 JPanel panel60 = new JPanel();
338 panel60.setLayout(new GridLayout(1, 1));
339 panel60.add(panel59);
340
341 JPanel panel61 = new JPanel();
342 panel61.setLayout(new GridLayout(1, 1));
343 panel61.add(panel60);
344
345 JPanel panel62 = new JPanel();
346 panel62.setLayout(new GridLayout(1, 1));
347 panel62.add(panel61);
348
349 JPanel panel63 = new JPanel();
350 panel63.setLayout(new GridLayout(1, 1));
351 panel63.add(panel62);
352
353 JPanel panel64 = new JPanel();
354 panel64.setLayout(new GridLayout(1, 1));
355 panel64.add(panel63);
356
357 JPanel panel65 = new JPanel();
358 panel65.setLayout(new GridLayout(1, 1));
359 panel65.add(panel64);
360
361 JPanel panel66 = new JPanel();
362 panel66.setLayout(new GridLayout(1, 1));
363 panel66.add(panel65);
364
365 JPanel panel67 = new JPanel();
366 panel67.setLayout(new GridLayout(1, 1));
367 panel67.add(panel66);
368
369 JPanel panel68 = new JPanel();
370 panel68.setLayout(new GridLayout(1, 1));
371 panel68.add(panel67);
372
373 JPanel panel69 = new JPanel();
374 panel69.setLayout(new GridLayout(1, 1));
375 panel69.add(panel68);
376
377 JPanel panel70 = new JPanel();
378 panel70.setLayout(new GridLayout(1, 1));
379 panel70.add(panel69);
380
381 JPanel panel71 = new JPanel();
382 panel71.setLayout(new GridLayout(1, 1));
383 panel71.add(panel70);
384
385 JPanel panel72 = new JPanel();
386 panel72.setLayout(new GridLayout(1, 1));
387 panel72.add(panel71);
388
389 JPanel panel73 = new JPanel();
390 panel73.setLayout(new GridLayout(1, 1));
391 panel73.add(panel72);
392
393 JPanel panel74 = new JPanel();
394 panel74.setLayout(new GridLayout(1, 1));
395 panel74.add(panel73);
396
397 JPanel panel75 = new JPanel();
398 panel75.setLayout(new GridLayout(1, 1));
399 panel75.add(panel74);
400
401 JPanel panel76 = new JPanel();
402 panel76.setLayout(new GridLayout(1, 1));
403 panel76.add(panel75);
404
405 JPanel panel77 = new JPanel();
406 panel77.setLayout(new GridLayout(1, 1));
407 panel77.add(panel76);
408
409 JPanel panel78 = new JPanel();
410 panel78.setLayout(new GridLayout(1, 1));
411 panel78.add(panel77);
412
413 JPanel panel79 = new JPanel();
414 panel79.setLayout(new GridLayout(1, 1));
415 panel79.add(panel78);
416
417 JPanel panel80 = new JPanel();
418 panel80.setLayout(new GridLayout(1, 1));
419 panel80.add(panel79);
420
421 JPanel panel81 = new JPanel();
422 panel81.setLayout(new GridLayout(1, 1));
423 panel81.add(panel80);
424
425 JPanel panel82 = new JPanel();
426 panel82.setLayout(new GridLayout(1, 1));
427 panel82.add(panel81);
428
429 JPanel panel83 = new JPanel();
430 panel83.setLayout(new GridLayout(1, 1));
431 panel83.add(panel82);
432
433 JPanel panel84 = new JPanel();
434 panel84.setLayout(new GridLayout(1, 1));
435 panel84.add(panel83);
436
437 JPanel panel85 = new JPanel();
438 panel85.setLayout(new GridLayout(1, 1));
439 panel85.add(panel84);
440
441 JPanel panel86 = new JPanel();
442 panel86.setLayout(new GridLayout(1, 1));
443 panel86.add(panel85);
444
445 JPanel panel87 = new JPanel();
446 panel87.setLayout(new GridLayout(1, 1));
447 panel87.add(panel86);
448
449 JPanel panel88 = new JPanel();
450 panel88.setLayout(new GridLayout(1, 1));
451 panel88.add(panel87);
452
453 JPanel panel89 = new JPanel();
454 panel89.setLayout(new GridLayout(1, 1));
455 panel89.add(panel88);
456
457 JPanel panel90 = new JPanel();
458 panel90.setLayout(new GridLayout(1, 1));
459 panel90.add(panel89);
460
461 JPanel panel91 = new JPanel();
462 panel91.setLayout(new GridLayout(1, 1));
463 panel91.add(panel90);
464
465 JPanel panel92 = new JPanel();
466 panel92.setLayout(new GridLayout(1, 1));
467 panel92.add(panel91);
468
469 JPanel panel93 = new JPanel();
470 panel93.setLayout(new GridLayout(1, 1));
471 panel93.add(panel92);
472
473 JPanel panel94 = new JPanel();
474 panel94.setLayout(new GridLayout(1, 1));
475 panel94.add(panel93);
476
477 JPanel panel95 = new JPanel();
478 panel95.setLayout(new GridLayout(1, 1));
479 panel95.add(panel94);
480
481 JPanel panel96 = new JPanel();
482 panel96.setLayout(new GridLayout(1, 1));
483 panel96.add(panel95);
484
485 JPanel panel97 = new JPanel();
486 panel97.setLayout(new GridLayout(1, 1));
487 panel97.add(panel96);
488
489 JPanel panel98 = new JPanel();
490 panel98.setLayout(new GridLayout(1, 1));
491 panel98.add(panel97);
492
493 JPanel panel99 = new JPanel();
494 panel99.setLayout(new GridLayout(1, 1));
495 panel99.add(panel98);
496
497 JPanel panel100 = new JPanel();
498 panel100.setLayout(new GridLayout(1, 1));
499 panel100.add(panel99);
500
501 JPanel panel101 = new JPanel();
502 panel101.setLayout(new GridLayout(1, 1));
503 panel101.add(panel100);
504
505 JPanel panel102 = new JPanel();
506 panel102.setLayout(new GridLayout(1, 1));
507 panel102.add(panel101);
508
509 JPanel panel103 = new JPanel();
510 panel103.setLayout(new GridLayout(1, 1));
511 panel103.add(panel102);
512
513 JPanel panel104 = new JPanel();
514 panel104.setLayout(new GridLayout(1, 1));
515 panel104.add(panel103);
516
517 JPanel panel105 = new JPanel();
518 panel105.setLayout(new GridLayout(1, 1));
519 panel105.add(panel104);
520
521 JPanel panel106 = new JPanel();
522 panel106.setLayout(new GridLayout(1, 1));
523 panel106.add(panel105);
524
525 JPanel panel107 = new JPanel();
526 panel107.setLayout(new GridLayout(1, 1));
527 panel107.add(panel106);
528
529 JPanel panel108 = new JPanel();
530 panel108.setLayout(new GridLayout(1, 1));
531 panel108.add(panel107);
532
533 JPanel panel109 = new JPanel();
534 panel109.setLayout(new GridLayout(1, 1));
535 panel109.add(panel108);
536
537 JPanel panel110 = new JPanel();
538 panel110.setLayout(new GridLayout(1, 1));
539 panel110.add(panel109);
540
541 JPanel panel111 = new JPanel();
542 panel111.setLayout(new GridLayout(1, 1));
543 panel111.add(panel110);
544
545 JPanel panel112 = new JPanel();
546 panel112.setLayout(new GridLayout(1, 1));
547 panel112.add(panel111);
548
549 JPanel panel113 = new JPanel();
550 panel113.setLayout(new GridLayout(1, 1));
551 panel113.add(panel112);
552
553 JPanel panel114 = new JPanel();
554 panel114.setLayout(new GridLayout(1, 1));
555 panel114.add(panel113);
556
557 JPanel panel115 = new JPanel();
558 panel115.setLayout(new GridLayout(1, 1));
559 panel115.add(panel114);
560
561 JPanel panel116 = new JPanel();
562 panel116.setLayout(new GridLayout(1, 1));
563 panel116.add(panel115);
564
565 JPanel panel117 = new JPanel();
566 panel117.setLayout(new GridLayout(1, 1));
567 panel117.add(panel116);
568
569 JPanel panel118 = new JPanel();
570 panel118.setLayout(new GridLayout(1, 1));
571 panel118.add(panel117);
572
573 JPanel panel119 = new JPanel();
574 panel119.setLayout(new GridLayout(1, 1));
575 panel119.add(panel118);
576
577 JPanel panel120 = new JPanel();
578 panel120.setLayout(new GridLayout(1, 1));
579 panel120.add(panel119);
580
581 JPanel panel121 = new JPanel();
582 panel121.setLayout(new GridLayout(1, 1));
583 panel121.add(panel120);
584
585 JPanel panel122 = new JPanel();
586 panel122.setLayout(new GridLayout(1, 1));
587 panel122.add(panel121);
588
589 JPanel panel123 = new JPanel();
590 panel123.setLayout(new GridLayout(1, 1));
591 panel123.add(panel122);
592
593 JPanel panel124 = new JPanel();
594 panel124.setLayout(new GridLayout(1, 1));
595 panel124.add(panel123);
596
597 JPanel panel125 = new JPanel();
598 panel125.setLayout(new GridLayout(1, 1));
599 panel125.add(panel124);
500
501 JPanel panel126 = new JPanel();
502 panel126.setLayout(new GridLayout(1, 1));
503 panel126.add(panel125);
504
505 JPanel panel127 = new JPanel();
506 panel127.setLayout(new GridLayout(1, 1));
507 panel127.add(panel126);
508
509 JPanel panel128 = new JPanel();
510 panel128.setLayout(new GridLayout(1, 1));
511 panel128.add(panel127);
512
513 JPanel panel129 = new JPanel();
514 panel129.setLayout(new GridLayout(1, 1));
515 panel129.add(panel128);
516
517 JPanel panel130 = new JPanel();
518 panel130.setLayout(new GridLayout(1, 1));
519 panel130.add(panel129);
520
521 JPanel panel131 = new JPanel();
522 panel131.setLayout(new GridLayout(1, 1));
523 panel131.add(panel130);
524
525 JPanel panel132 = new JPanel();
526 panel132.setLayout(new GridLayout(1, 1));
527 panel132.add(panel131);
528
529 JPanel panel133 = new JPanel();
530 panel133.setLayout(new GridLayout(1, 1));
531 panel133.add(panel132);
532
533 JPanel panel134 = new JPanel();
534 panel134.setLayout(new GridLayout(1, 1));
535 panel134.add(panel133);
536
537 JPanel panel135 = new JPanel();
538 panel135.setLayout(new GridLayout(1, 1));
539 panel135.add(panel134);
540
541 JPanel panel136 = new JPanel();
542 panel136.setLayout(new GridLayout(1, 1));
543 panel136.add(panel135);
544
545 JPanel panel137 = new JPanel();
546 panel137.setLayout(new GridLayout(1, 1));
547 panel137.add(panel136);
548
549 JPanel panel138 = new JPanel();
550 panel138.setLayout(new GridLayout(1, 1));
551 panel138.add(panel137);
552
553 JPanel panel139 = new JPanel();
554 panel139.setLayout(new GridLayout(1, 1));
555 panel139.add(panel138);
556
557 JPanel panel140 = new JPanel();
558 panel140.setLayout(new GridLayout(1, 1));
559 panel140.add(panel139);
560
561 JPanel panel141 = new JPanel();
562 panel141.setLayout(new GridLayout(1, 1));
563 panel141.add(panel140);
564
565 JPanel panel142 = new JPanel();
566 panel142.setLayout(new GridLayout(1, 1));
567 panel142.add(panel141);
568
569 JPanel panel143 = new JPanel();
570 panel143.setLayout(new GridLayout(1, 1));
571 panel143.add(panel142);
572
573 JPanel panel144 = new JPanel();
574 panel144.setLayout(new GridLayout(1, 1));
575 panel144.add(panel143);
576
577 JPanel panel145 = new JPanel();
578 panel145.setLayout(new GridLayout(1, 1));
579 panel145.add(panel144);
580
581 JPanel panel146 = new JPanel();
582 panel146.setLayout(new GridLayout(1, 1));
583 panel146.add(panel145);
584
585 JPanel panel147 = new JPanel();
586 panel147.setLayout(new GridLayout(1, 1));
587 panel147.add(panel146);
588
589 JPanel panel148 = new JPanel();
590 panel148.setLayout(new GridLayout(1, 1));
591 panel148.add(panel147);
592
593 JPanel panel149 = new JPanel();
594 panel149.setLayout(new GridLayout(1, 1));
595 panel149.add(panel148);
596
597 JPanel panel150 = new JPanel();
598 panel150.setLayout(new GridLayout(1, 1));
599 panel150.add(panel149);
500
501 JPanel panel151 = new JPanel();
502 panel151.setLayout(new GridLayout(1, 1));
503 panel151.add(panel150);
504
505 JPanel panel152 = new JPanel();
506 panel152.setLayout(new GridLayout(1, 1));
507 panel152.add(panel151);
508
509 JPanel panel153 = new JPanel();
510 panel153.setLayout(new GridLayout(1, 1));
511 panel153.add(panel152);
512
513 JPanel panel154 = new JPanel();
514 panel154.setLayout(new GridLayout(1, 1));
515 panel154.add(panel153);
516
517 JPanel panel155 = new JPanel();
518 panel155.setLayout(new GridLayout(1, 1));
519 panel155.add(panel154);
520
521 JPanel panel156 = new JPanel();
522 panel156.setLayout(new GridLayout(1, 1));
523 panel156.add(panel155);
524
525 JPanel panel157 = new JPanel();
526 panel157.setLayout(new GridLayout(1, 1));
527 panel157.add(panel156);
528
529 JPanel panel158 = new JPanel();
530 panel158.setLayout(new GridLayout(1, 1));
531 panel158.add(panel157);
532
533 JPanel panel159 = new JPanel();
534 panel159.setLayout(new GridLayout(1, 1));
535 panel159.add(panel158);
536
537 JPanel panel160 = new JPanel();
538 panel160.setLayout(new GridLayout(1, 1));
539 panel160.add(panel159);
540
541 JPanel panel161 = new JPanel();
542 panel161.setLayout(new GridLayout(1, 1));
543 panel161.add(panel160);
544
545 JPanel panel162 = new JPanel();
546 panel162.setLayout(new GridLayout(1, 1));
547 panel162.add(panel161);
548
549 JPanel panel163 = new JPanel();
550 panel163.setLayout(new GridLayout(1, 1));
551 panel163.add(panel162);
552
553 JPanel panel164 = new JPanel();
554 panel164.setLayout(new GridLayout(1, 1));
555 panel164.add(panel163);
556
557 JPanel panel165 = new JPanel();
558 panel165.setLayout(new GridLayout(1, 1));
559 panel165.add(panel164);
560
561 JPanel panel166 = new JPanel();
562 panel166.setLayout(new GridLayout(1, 1));
563 panel166.add(panel165);
564
565 JPanel panel167 = new JPanel();
566 panel167.setLayout(new GridLayout(1, 1));
567 panel167.add(panel166);
568
569 JPanel panel168 = new JPanel();
570 panel168.setLayout(new GridLayout(1, 1));
571 panel168.add(panel167);
572
573 JPanel panel169 = new JPanel();
574 panel169.setLayout(new GridLayout(1, 1));
575 panel169.add(panel168);
576
577 JPanel panel170 = new JPanel();
578 panel170.setLayout(new GridLayout(1, 1));
579 panel170.add(panel169);
580
581 JPanel panel171 = new JPanel();
582 panel171.setLayout(new GridLayout(1, 1));
583 panel171.add(panel170);
584
585 JPanel panel172 = new JPanel();
586 panel172.setLayout(new GridLayout(1, 1));
587 panel172.add(panel171);
588
589 JPanel panel173 = new JPanel();
590 panel173.setLayout(new GridLayout(1, 1));
591 panel173.add(panel172);
592
593 JPanel panel174 = new JPanel();
594 panel174.setLayout(new GridLayout(1, 1));
595 panel174.add(panel173);
596
597 JPanel panel175 = new JPanel();
598 panel175.setLayout(new GridLayout(1, 1));
599 panel175.add(panel174);
500
501 JPanel panel176 = new JPanel();
502 panel176.setLayout(new GridLayout(1, 1));
503 panel176.add(panel175);
504
505 JPanel panel177 = new JPanel();
506 panel177.setLayout(new GridLayout(1, 1));
507 panel177.add(panel176);
508
509 JPanel panel178 = new JPanel();
510 panel178.setLayout(new GridLayout(1, 1));
511 panel178.add(panel177);
512
513 JPanel panel179 = new JPanel();
514 panel179.setLayout(new GridLayout(1, 1));
515 panel179.add(panel178);
516
517 JPanel panel180 = new JPanel();
518 panel180.setLayout(new GridLayout(1, 1));
519 panel180.add(panel179);
520
521 JPanel panel181 = new JPanel();
522 panel181.setLayout(new GridLayout(1, 1));
523 panel181.add(panel180);
524
525 JPanel panel182 = new JPanel();
526 panel182.setLayout(new GridLayout(1, 1));
527 panel182.add(panel181);
528
529 JPanel panel183 = new JPanel();
530 panel183.setLayout(new GridLayout(1, 1));
531 panel183.add(panel182);
532
533 JPanel panel184 = new JPanel();
534 panel184.setLayout(new GridLayout(1, 1));
535 panel184.add(panel183);
536
537 JPanel panel185 = new JPanel();
538 panel185.setLayout(new GridLayout(1, 1));
539 panel185.add(panel184);
540
541 JPanel panel186 = new JPanel();
542 panel186.setLayout(new GridLayout(1, 1));
543 panel186.add(panel185);
544
545 JPanel panel187 = new JPanel();
546 panel187.setLayout(new GridLayout(1, 1));
547 panel187.add(panel186);
548
549 JPanel panel188 = new JPanel();
550 panel188.setLayout(new GridLayout(1, 1));
551 panel188.add(panel187);
552
553 JPanel panel189 = new JPanel();
554 panel189.setLayout(new GridLayout(1, 1));
555 panel189.add(panel188);
556
557 JPanel panel190 = new JPanel();
558 panel190.setLayout(new GridLayout(1, 1));
559 panel190.add(panel189);
560
561 JPanel panel191 = new JPanel();
562 panel191.setLayout(new GridLayout(1, 1));
563 panel191.add(panel190);
564
565 JPanel panel192 = new JPanel();
566 panel192.setLayout(new GridLayout(1, 1));
567 panel192.add(panel191);
568
569 JPanel panel193 = new JPanel();
570 panel193.setLayout(new GridLayout(1, 1));
571 panel193.add(panel192);
572
573 JPanel panel194 = new JPanel();
574 panel194.setLayout(new GridLayout(1, 1));
575 panel194.add(panel193);
576
577 JPanel panel195 = new JPanel();
578 panel195.setLayout(new GridLayout(1, 1));
579 panel195.add(panel194);
580
581 JPanel panel196 = new JPanel();
582 panel196.setLayout(new GridLayout(1, 1));
583 panel196.add(panel195);
584
585 JPanel panel197 = new JPanel();
586 panel197.setLayout(new GridLayout(1, 1));
587 panel197.add(panel196);
588
589 JPanel panel198 = new JPanel();
590 panel198.setLayout(new GridLayout(1, 1));
591 panel198.add(panel197);
592
593 JPanel panel199 = new JPanel();
594 panel199.setLayout(new GridLayout(1, 1));
595 panel199.add(panel198);
596
597 JPanel panel200 = new JPanel();
598 panel200.setLayout(new GridLayout(1, 1));
599 panel200.add(panel199);
500
501 JPanel panel201 = new JPanel();
502 panel201.setLayout(new GridLayout(1, 1));
503 panel201.add(panel200);
504
505 JPanel panel202 = new JPanel();
506 panel202.setLayout(new GridLayout(1, 1));
507 panel202.add(panel201);
508
509 JPanel panel203 = new JPanel();
510 panel203.setLayout(new GridLayout(1, 1));
511 panel203.add(panel202);
512
513 JPanel panel204 = new JPanel();
514 panel204.setLayout(new GridLayout(1, 1));
515 panel204.add(panel203);
516
517 JPanel panel205 = new JPanel();
518 panel205.setLayout(new GridLayout(1, 1));
519 panel205.add(panel204);
520
521 JPanel panel206 = new JPanel();
522 panel206.setLayout(new GridLayout(1, 1));
523 panel206.add(panel205);
524
525 JPanel panel207 = new JPanel();
526 panel207.setLayout(new GridLayout(1, 1));
527 panel207.add(panel206);
528
529 JPanel panel208 = new JPanel();
530 panel208.setLayout(new GridLayout(1, 1));
531 panel208.add(panel207);
532
533 JPanel panel209 = new JPanel();
534 panel209.setLayout(new GridLayout(1, 1));
535 panel209.add(panel208);
536
537 JPanel panel210 = new JPanel();
538 panel210.setLayout(new GridLayout(1, 1));
539 panel210.add(panel209);
540
541 JPanel panel211 = new JPanel();
542 panel211.setLayout(new GridLayout(1, 1));
543 panel211.add(panel210);
544
545 JPanel panel212 = new JPanel();
546 panel212.setLayout(new GridLayout(1, 1));
547 panel212.add(panel211);
548
549 JPanel panel213 = new JPanel();
550 panel213.setLayout(new GridLayout(1, 1));
551 panel213.add(panel212);
552
553 JPanel panel214 = new JPanel();
554 panel214.setLayout(new GridLayout(1, 1));
555 panel214.add(panel213);
556
557 JPanel panel215 = new JPanel();
558 panel215.setLayout(new GridLayout(1, 1));
559 panel215.add(panel214);
560
561 JPanel panel216 = new JPanel();
562 panel216.setLayout(new GridLayout(1, 1));
563 panel216.add(panel215);
564
565 JPanel panel217 = new JPanel();
566 panel217.setLayout(new GridLayout(1, 1));
567 panel217.add(panel216);
568
569 JPanel panel218 = new JPanel();
570 panel218.setLayout(new GridLayout(1, 1));
571 panel218.add(panel217);
572
573 JPanel panel219 = new JPanel();
574 panel219.setLayout(new GridLayout(1, 1));
575 panel219.add(panel218);
576
577 JPanel panel220 = new JPanel();
578 panel220.setLayout(new GridLayout(1, 1));
579 panel220.add(panel219);
580
581 JPanel panel221 = new JPanel();
582 panel221.setLayout(new GridLayout(1, 1));
583 panel221.add(panel220);
584
585 JPanel panel222 = new JPanel();
586 panel222.setLayout(new GridLayout(1, 1));
587 panel222.add(panel221);
588
589 JPanel panel223 = new JPanel();
590 panel223.setLayout(new GridLayout(1, 1));
591 panel223.add(panel222);
592
593 JPanel panel224 = new JPanel();
594 panel224.setLayout(new GridLayout(1, 1));
595 panel224.add(panel223);
596
597 JPanel panel225 = new JPanel();
598 panel225.setLayout(new GridLayout(1, 1));
599 panel225.add(panel224);
500
501 JPanel panel226 = new JPanel();
502 panel226.setLayout(new GridLayout(1, 1));
503 panel226.add(panel225);
504
505 JPanel panel227 = new JPanel();
506 panel227.setLayout(new GridLayout(1, 1));
507 panel227.add(panel226);
508
509 JPanel panel228 = new JPanel();
510 panel228.setLayout(new GridLayout(1, 1));
511 panel228.add(panel227);
512
513 JPanel panel229 = new JPanel();
514 panel229.setLayout(new GridLayout(1, 1));
515 panel229.add(panel228);
516
517 JPanel panel230 = new JPanel();
518 panel230.setLayout(new GridLayout(1, 1));
519 panel230.add(panel229);
520
521 JPanel panel231 = new JPanel();
522 panel231.setLayout(new GridLayout(1, 1));
523 panel231.add(panel230);
524
525 JPanel panel232 = new JPanel();
526 panel232.setLayout(new GridLayout(1, 1));
527 panel232.add(panel231);
528
529 JPanel panel233 = new JPanel();
530 panel233.setLayout(new GridLayout(1, 1));
531 panel233.add(panel232);
532
533 JPanel panel234 = new JPanel();
534 panel234.setLayout(new GridLayout(1, 1));
535 panel235 panel234.add(panel233);
536
537 JPanel panel236 = new JPanel();
538 panel236.setLayout(new GridLayout(1, 1));
539 panel236.add(panel235);
540
541 JPanel panel237 = new JPanel();
542 panel237.setLayout(new GridLayout(1, 1));
543 panel237.add(panel236);
544
545 JPanel panel238 = new JPanel();
546 panel238.setLayout(new GridLayout(1, 1));
547 panel238.add(panel237);
548
549 JPanel panel239 = new JPanel();
550 panel239.setLayout(new GridLayout(1, 1));
551 panel239.add(panel238);
552
553 JPanel panel240 = new JPanel();
554 panel240.setLayout(new GridLayout(1, 1));
555 panel240.add(panel239);
556
557 JPanel panel241 = new JPanel();
558 panel241.setLayout(new GridLayout(1, 1));
559 panel241.add(panel240);
560
561 JPanel panel242 = new JPanel();
562 panel242.setLayout(new GridLayout(1, 1));
563 panel242.add(panel241);
564
565 JPanel panel243 = new JPanel();
566 panel243
```

6. Pokemon Market (Role : Member)

The screenshot shows two instances of the Eclipse IDE interface. Both instances display the same Java code for the `BuyForm` class, which is part of the `pokemonNK` package. The code implements `ActionListener` and `MouseListener` interfaces. It initializes a frame with panels for upper, middle, and lower sections, and adds components like a scroll pane, a table, and buttons. The code uses various Java Swing and JDBC libraries.

```
1 package pokemonNK;
2
3 import java.awt.*;
4 import java.awt.event.*;
5 import java.sql.*;
6 import java.util.*;
7 import javax.swing.*;
8 import javax.swing.border.*;
9 import javax.swing.table.*;
10
11 import pokemon.NBConnect;
12
13 public class BuyForm extends JFrame implements ActionListener, MouseListener {
14     JPanel upperPanel, midPanel, lowerPanel, mainPanel;
15     JPanel IDPanel, QTYPanel, IDTP, QTYTP, bPanel;
16     JLabel IDLabel, QTYLabel;
17     JTextField inputID, inputQTY;
18     JButton bInsert, bBack;
19
20     JTable Table = new JTable();
21     JScrollPane scrollPane = new JScrollPane();
22     DefaultTableModel dtmBuy;
23
24     Vector<Vector> data = new Vector<>();
25     Vector<String> header = new Vector<>();
26
27     String ID = "";
28     String Quantity = "";
29
30     static Vector<String> vID = new Vector<>();
31     static Vector<String> vQuantity = new Vector<>();
32
33     DBConnect con = DBConnect.getConnection();
34
35     public BuyForm() {
36         initFrame();
37         initComp();
38     }
39
40     private void initFrame() {
41         setTitle("Buy Form");
42         setSize(600, 700);
43         setDefaultCloseOperation(EXIT_ON_CLOSE);
44         setLocationRelativeTo(null);
45         setResizable(true);
46         setVisible(true);
47     }
48
49     private void initComp() {
50         mainPanel = new JPanel(new BorderLayout());
51
52         dtmBuy = new DefaultTableModel(data, header);
53         header.add("Pokemon ID");
54         header.add("Pokemon Name");
55         header.add("Pokemon Level");
56         header.add("Pokemon Type");
57
58         Table.setModel(dtmBuy);
59         Table.addMouseListener(this);
60         scrollPane.setViewportView(Table);
61
62         setupTable();
63     }
64
65     private void setupTable() {
66         upperPanel = new JPanel();
67         upperPanel.setLayout(new EmptyBorder(20, 20, 20, 20));
68
69         midPanel = new JPanel(new GridLayout(6, 1));
70         midPanel.setBorder(new EmptyBorder(0, 60, 10, 60));
71
72         bInsert = new JButton("Insert");
73         bInsert.addActionListener(this);
74         bInsert.setBorder(new EmptyBorder(0, 0, 0, 0));
75
76         bBack = new JButton("Back");
77         bBack.addActionListener(this);
78         bBack.setBorder(new EmptyBorder(0, 0, 0, 0));
79
80         upperPanel.add(bInsert);
81         upperPanel.add(bBack);
82
83         midPanel.add(inputID);
84         midPanel.add(inputQTY);
85         midPanel.add(IDLabel);
86         midPanel.add(QTYLabel);
87
88         mainPanel.add(upperPanel, "North");
89         mainPanel.add(midPanel, "Center");
90         mainPanel.add(scrollPane, "South");
91
92         setLayout(mainPanel);
93     }
94
95     public static void main(String[] args) {
96         new BuyForm();
97     }
98 }
```

This screenshot shows the same Java code for the `BuyForm` class, identical to the one in the first instance. The code defines a frame with panels for upper, middle, and lower sections, and adds components like a scroll pane, a table, and buttons. The code uses various Java Swing and JDBC libraries.

```
1 package pokemonNK;
2
3 import java.awt.*;
4 import java.awt.event.*;
5 import java.sql.*;
6 import java.util.*;
7 import javax.swing.*;
8 import javax.swing.border.*;
9 import javax.swing.table.*;
10
11 import pokemon.NBConnect;
12
13 public class BuyForm extends JFrame implements ActionListener, MouseListener {
14     JPanel upperPanel, midPanel, lowerPanel, mainPanel;
15     JPanel IDPanel, QTYPanel, IDTP, QTYTP, bPanel;
16     JLabel IDLabel, QTYLabel;
17     JTextField inputID, inputQTY;
18     JButton bInsert, bBack;
19
20     JTable Table = new JTable();
21     JScrollPane scrollPane = new JScrollPane();
22     DefaultTableModel dtmBuy;
23
24     Vector<Vector> data = new Vector<>();
25     Vector<String> header = new Vector<>();
26
27     String ID = "";
28     String Quantity = "";
29
30     static Vector<String> vID = new Vector<>();
31     static Vector<String> vQuantity = new Vector<>();
32
33     DBConnect con = DBConnect.getConnection();
34
35     public BuyForm() {
36         initFrame();
37         initComp();
38     }
39
40     private void initFrame() {
41         setTitle("Buy Form");
42         setSize(600, 700);
43         setDefaultCloseOperation(EXIT_ON_CLOSE);
44         setLocationRelativeTo(null);
45         setResizable(true);
46         setVisible(true);
47     }
48
49     private void initComp() {
50         mainPanel = new JPanel(new BorderLayout());
51
52         dtmBuy = new DefaultTableModel(data, header);
53         header.add("Pokemon ID");
54         header.add("Pokemon Name");
55         header.add("Pokemon Level");
56         header.add("Pokemon Type");
57
58         Table.setModel(dtmBuy);
59         Table.addMouseListener(this);
60         scrollPane.setViewportView(Table);
61
62         setupTable();
63     }
64
65     private void setupTable() {
66         upperPanel = new JPanel();
67         upperPanel.setLayout(new EmptyBorder(20, 20, 20, 20));
68
69         midPanel = new JPanel(new GridLayout(6, 1));
70         midPanel.setBorder(new EmptyBorder(0, 60, 10, 60));
71
72         bInsert = new JButton("Insert");
73         bInsert.addActionListener(this);
74         bInsert.setBorder(new EmptyBorder(0, 0, 0, 0));
75
76         bBack = new JButton("Back");
77         bBack.addActionListener(this);
78         bBack.setBorder(new EmptyBorder(0, 0, 0, 0));
79
80         upperPanel.add(bInsert);
81         upperPanel.add(bBack);
82
83         midPanel.add(inputID);
84         midPanel.add(inputQTY);
85         midPanel.add(IDLabel);
86         midPanel.add(QTYLabel);
87
88         mainPanel.add(upperPanel, "North");
89         mainPanel.add(midPanel, "Center");
90         mainPanel.add(scrollPane, "South");
91
92         setLayout(mainPanel);
93     }
94
95     public static void main(String[] args) {
96         new BuyForm();
97     }
98 }
```

eclipse-workspace - Project_BA11_BAD/src/pokemonNK/BuyForm.java - Eclipse IDE

```
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer Project Explorer
Main.java Main.java MainFrame.java ProductView... *memberMake... MainForm.java MainJava ManagePoko... BuyForm.java
1 MainJava
2 Perteman9
3 > MainJava
4 > Perteman9.zip
5 > Exercise Class
6 > Kelas_Tambah
7 > Lathan
8 > LathanJTS
9 > PengumpulanTugas
10 > Pertemuan_2_Lab
11 > Presentasi_BAD_Group2_1
12 > IRE System Library [JavaSE-1.8]
13 > src
14 > DbConnector
15 > JDBCConnect.java
16 > MainJava
17 > memberMaker.java
18 > Referenced Libraries
19 > Project
20 > Project_BA11_BAD
21 > src
22 > pokemon
23 > pokemonNK
24 > Bag.java
25 > BuyForm.java
26 > History.java
27 > Login.java
28 > MainJava
29 > MainForm.java
30 > ManagePokemon.java
31 > Register.java
32 > IRE System Library [JavaSE-1.8]
33 > Referenced Libraries
34 > Quiz1
35 > Session3_Class
36 > Test_Class
37 > Test2
38 > Testing
39 > ITS_Semester3_2
40 >
41 >
42 >
43 >
44 >
45 >
46 >
47 >
48 >
49 >
50 >
51 >
52 >
53 >
54 >
55 >
56 >
57 >
58 >
59 >
60 >
61 >
62 >
63 >
64 >
65 >
66 >
67 >
68 >
69 >
70 >
71 >
72 >
73 >
74 > muraria = new JFrame();
75 > muraria.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
76 > midPanel.setBorder(new EmptyBorder(0, 60, 10, 60));
77 > midPanel.setBackground(Color.CYAN);
78 > lowerPanel = new JPanel(new GridLayout(1, 2));
79 > lowerPanel.setBorder(new EmptyBorder(0, 60, 10, 60));
80 > lowerPanel.setBackground(Color.CYAN);
81 > // Upper Panel
82 > upperPanel.add(scrollPane);
83 > // Mid Panel
84 > // ID
85 > JPanel IDPanel = new JPanel(new GridLayout(1, 1));
86 > IDPanel.setBackground(Color.CYAN);
87 > JLabel IDLabel = new JLabel("Insert Pokemon ID");
88 > IDLabel.setHorizontalTextPosition(JLabel.LEADING);
89 > IDPanel.add(IDLabel);
90 > JTextField IDTF = new JTextField();
91 > IDPanel.add(IDTF);
92 > IDPanel.add(IDTF);
93 > IDPanel.add(IDTF);
94 > IDPanel.add(IDTF);
95 > midPanel.add(IDPanel);
96 > // Quantity
97 > JPanel QTYPanel = new JPanel(new GridLayout(1, 1));
98 > QTYPanel.setBackground(Color.CYAN);
99 > JLabel QTYLabel = new JLabel("Quantity");
100 > QTYLabel.setHorizontalTextPosition(JLabel.LEADING);
101 > QTYPanel.add(QTYLabel);
102 > JTextField QTYTF = new JTextField();
103 > QTYPanel.add(QTYTF);
104 > QTYPanel.add(QTYTF);
105 > QTYPanel.add(QTYTF);
106 > QTYPanel.add(QTYTF);
107 > // Lower Panel
108 > bPanel = new JPanel();
109 > bPanel.setBackground(Color.CYAN);
110 > bInsert = new JButton("Insert");
111 >
```

eclipse-workspace - Project_BA11_BAD/src/pokemonNK/BuyForm.java - Eclipse IDE

```
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer Project Explorer
Main.java Main.java MainFrame.java ProductView... *memberMake... MainForm.java MainJava ManagePoko... BuyForm.java
1 MainJava
2 Perteman9
3 > MainJava
4 > Perteman9.zip
5 > Exercise Class
6 > Kelas_Tambah
7 > Lathan
8 > LathanJTS
9 > PengumpulanTugas
10 > Pertemuan_2_Lab
11 > Presentasi_BAD_Group2_1
12 > IRE System Library [JavaSE-1.8]
13 > src
14 > DbConnector
15 > JDBCConnect.java
16 > MainJava
17 > memberMaker.java
18 > Referenced Libraries
19 > Project
20 > Project_BA11_BAD
21 > src
22 > pokemon
23 > pokemonNK
24 > Bag.java
25 > BuyForm.java
26 > History.java
27 > Login.java
28 > MainJava
29 > MainForm.java
30 > ManagePokemon.java
31 > Register.java
32 > IRE System Library [JavaSE-1.8]
33 > Referenced Libraries
34 > Quiz1
35 > Session3_Class
36 > Test_Class
37 > Test2
38 > Testing
39 > ITS_Semester3_2
40 >
41 >
42 >
43 >
44 >
45 >
46 >
47 >
48 >
49 >
50 >
51 >
52 >
53 >
54 >
55 >
56 >
57 >
58 >
59 >
60 >
61 >
62 >
63 >
64 >
65 >
66 >
67 >
68 >
69 >
70 >
71 >
72 >
73 >
74 > muraria = new JFrame();
75 > muraria.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
76 > bInsert = new JButton("Insert");
77 > bBack = new JButton("Back to Main");
78 > lowerPanel.add(bInsert);
79 > lowerPanel.add(bBack);
80 > // Main Panel
81 > mainPanel.add(upperPanel, BorderLayout.NORTH);
82 > mainPanel.add(midPanel, BorderLayout.CENTER);
83 > mainPanel.add(lowerPanel, BorderLayout.SOUTH);
84 > bInsert.addActionListener(new ActionListener() {
85 >     @Override
86 >     public void actionPerformed(ActionEvent e) {
87 >         ID = IDTF.getText();
88 >         Quantity = QTYTF.getText();
89 >         vID.add(ID);
90 >         vQuantity.add(Quantity);
91 >         if (validateNumber(Quantity) == false) {
92 >             JOptionPane.showMessageDialog(null, "Quantity must be a number!");
93 >         } else {
94 >             String query = "SELECT PokemonId FROM Pokemon WHERE PokemonId = '" + ID + "'";
95 >             ResultSet rs = con.executeQuery(query);
96 >             IDTF.setText("");
97 >             QTYTF.setText("");
98 >             try {
99 >                 if (rs.next()) {
100 >                     ID = rs.getString(1);
101 >                     JOptionPane.showMessageDialog(null, "Insert Success!");
102 >                 } else {
103 >                     JOptionPane.showMessageDialog(null, "PokemonId not found!");
104 >                 }
105 >             } catch (SQLException ex) {
106 >                 JOptionPane.showMessageDialog(null, "Error: " + ex.getMessage());
107 >             }
108 >         }
109 >     }
110 > }
```

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the workspace structure. Projects include "Perlemen9", "Exercise Class", "Kelas_Tambah", "Latihan", "LatihanUTS", "PengumpulanTugas", "Perfemanan_Lab", "Presentasi_BAD_Group_2", and "Project_BA11_BAD".
- Package Explorer:** Shows the file structure for "Project_BA11_BAD/src/pokemonNK/BuyForm.java".
- Editor:** Displays the Java code for the `BuyForm.java` class.
- Bottom Bar:** Includes tabs for Problems, Javadoc, Declaration, Search, Coverage, and Console.

```
146     JOptionPane.showMessageDialog(null, "Insert Success!");
147 } else {
148     JOptionPane.showMessageDialog(null, "ID is not exist!");
149 }
150 }
151 } catch (SQLException e2) {
152     e2.printStackTrace();
153 }
154 }
155 }
156 }
157 });
158 bBack.addActionListener(new ActionListener() {
159     @Override
160     public void actionPerformed(ActionEvent e) {
161         dispose();
162         Mainform mf = new Mainform();
163     }
164 });
165 add(mainPanel);
166 }
167 });
168 });
169 });
170 add(mainPanel);
171 }
172 }
173 void setupTable() {
174     Table.setBackground(Color.CYAN);
175 }
176 String query = "SELECT * FROM pokemon";
177 ResultSet rs = con.executeQuery(query);
178 try {
179     data.clear();
180 }
```

The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** eclipse-workspace - Project_BA11_BAD/src/pokemonNK/BuyForm.java - Eclipse IDE
- Toolbar:** Includes standard Eclipse icons for File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help.
- Project Explorer:** Shows the project structure:
 - Exercise Class
 - Kelas_Jambahan
 - Latihan
 - LatihanITS
 - PengumpulanTugas
 - Perfumanii_2_Lab
 - Presentasi_BAD_Group_2_1
 - JRE System Library [JavaSE-1.8]
 - src
 - DbConnector
 - DBConnector.java
 - Mainjava
 - membermaker.java
 - member.java
 - Referenced Libraries
 - Project_BA11_BAD
 - src
 - pokemon
 - pokemonNK
 - Bag.java
 - BuyForm.java
 - History.java
 - Login.java
 - Main.java
 - MainForm.java
 - ManagePokemon.java
 - Register.java
 - JRE System Library [JavaSE-1.8]
 - Referenced Libraries
 - Quiz1
 - Session3_Class
 - Test_Class
 - Test2
 - Testing
 - UTS_Semester2 ?
- Code Editor:** Displays the Java code for `BuyForm.java`. The code includes methods for reading data from a database and validating input numbers.
- Bottom Bar:** Includes tabs for Problems, Javadoc, Declaration, Search, Coverage, and Console. The status bar at the bottom shows the file path as `SetupTable(x) - 0 references in workspace (0 matches filtered from view)`.

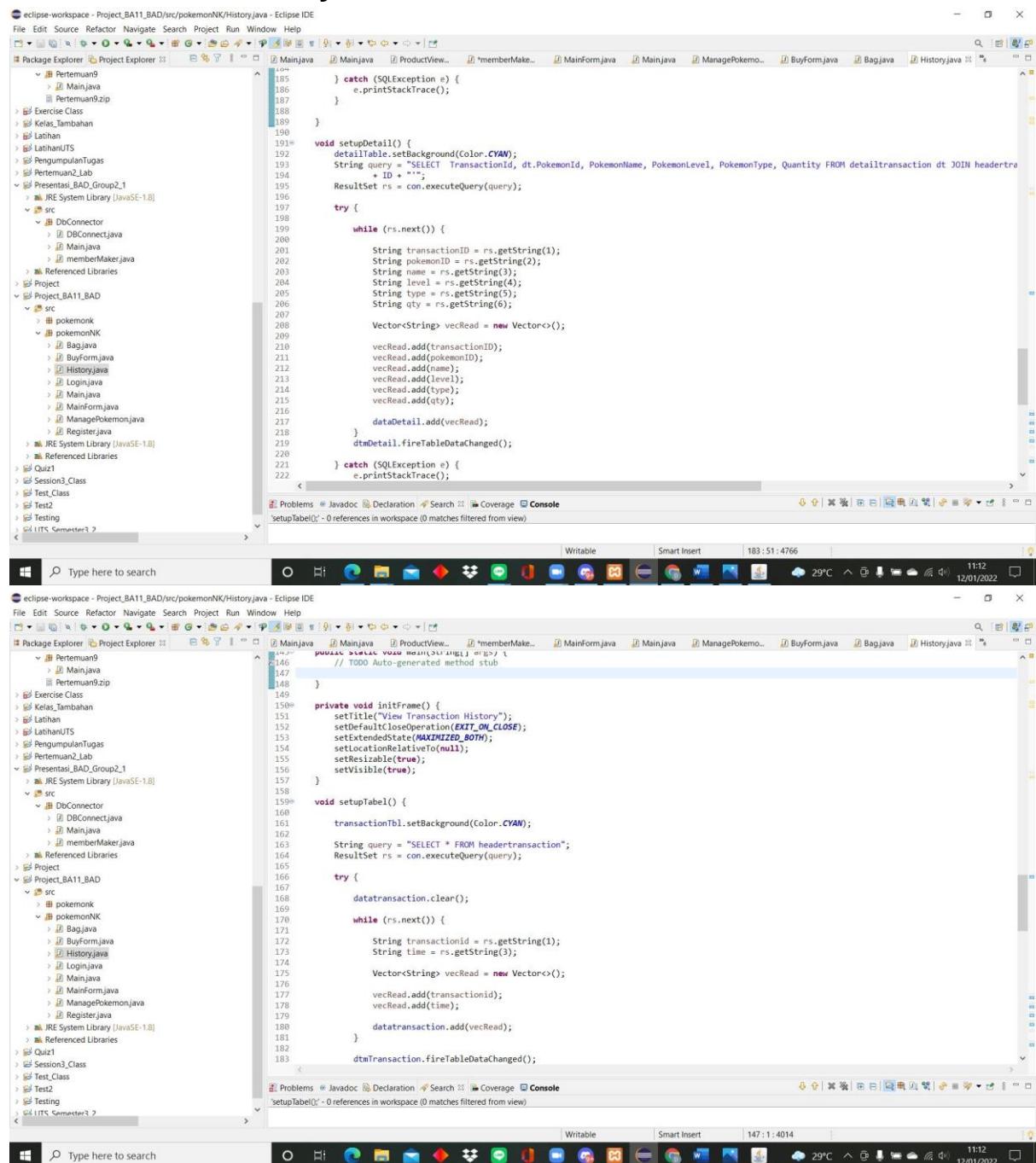
The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure:
 - Project_BA11_BAD
 - src
 - com.permanuan
 - com.permanuan.Bag
 - com.permanuan.BuyForm
 - com.permanuan.History
 - com.permanuan.Login
 - com.permanuan.Main
 - com.permanuan.MainForm
 - com.permanuan.ManagePokemon
 - com.permanuan.Register
 - JRE System Library [JavaSE-1.8]
 - Referenced Libraries
- Code Editor:** Displays the `MouseListener` implementation in the `BuyForm.java` file. The code includes several overridden methods:

```
public void mouseClicked(MouseEvent arg0) {  
    // TODO Auto-generated method stub  
}  
  
@Override  
public void mouseEntered(MouseEvent e) {  
    // TODO Auto-generated method stub  
}  
  
@Override  
public void mouseExited(MouseEvent e) {  
    // TODO Auto-generated method stub  
}  
  
@Override  
public void mousePressed(MouseEvent e) {  
    // TODO Auto-generated method stub  
}  
  
@Override  
public void mouseReleased(MouseEvent e) {  
    // TODO Auto-generated method stub  
}  
  
@Override  
public void actionPerformed(ActionEvent arg0) {  
    // TODO Auto-generated method stub  
}
```
- Bottom Bar:** Includes tabs for Problems, Javadoc, Declaration, Search, Coverage, and Console. The status bar at the bottom shows "SetupTable(0)- 0 references in workspace (0 matches filtered from view)".

- Ketika User mendapatkan role ‘Player’, maka user dapat menenkan menu ‘Pokemon Market’ untuk membeli pokemon yang diinginkan
 - Ketika di menu ‘Pokemon Market’, user dapat membeli pokemon apa yang diinginkan dengan menginput ID Pokemon yang tertera, dan berapa jumlah pokemon yang diinginkan
 - Sesudah menginput datanya, User dapat menenkan tombol ‘Insert’ untuk mendapatkan pokemon yang diinginkan
 - Untuk mengecek pokemon apa saja yang user miliki, cukup pergi ke menu ‘Bag’ untuk melihat apa saja pokemon yang user miliki

7. Transaction History



The screenshot shows the Eclipse IDE interface with two windows side-by-side, both displaying the same Java code for the `History.java` file.

The code is as follows:

```
187     } catch (SQLException e) {
188         e.printStackTrace();
189     }
190
191     void setupDetail() {
192         detailTable.setBackground(Color.CYAN);
193         String query = "SELECT TransactionId, dt.PokemonId, PokemonName, PokemonLevel, PokemonType, Quantity FROM detailtransaction dt JOIN headertransaction ht ON ht.TransactionId = dt.TransactionId";
194         + " WHERE ht.MemberId = ? ";
195         ResultSet rs = con.executeQuery(query);
196
197         try {
198             while (rs.next()) {
199                 String transactionID = rs.getString(1);
200                 String pokemonID = rs.getString(2);
201                 String name = rs.getString(3);
202                 String level = rs.getString(4);
203                 String type = rs.getString(5);
204                 String qty = rs.getString(6);
205
206                 Vector<String> vecRead = new Vector<>();
207
208                 vecRead.add(transactionID);
209                 vecRead.add(pokemonID);
210                 vecRead.add(name);
211                 vecRead.add(level);
212                 vecRead.add(type);
213                 vecRead.add(qty);
214
215                 dataDetail.addRow(vecRead);
216             }
217             dtmDetail.fireTableDataChanged();
218         } catch (SQLException e) {
219             e.printStackTrace();
220         }
221     }
222 }
```

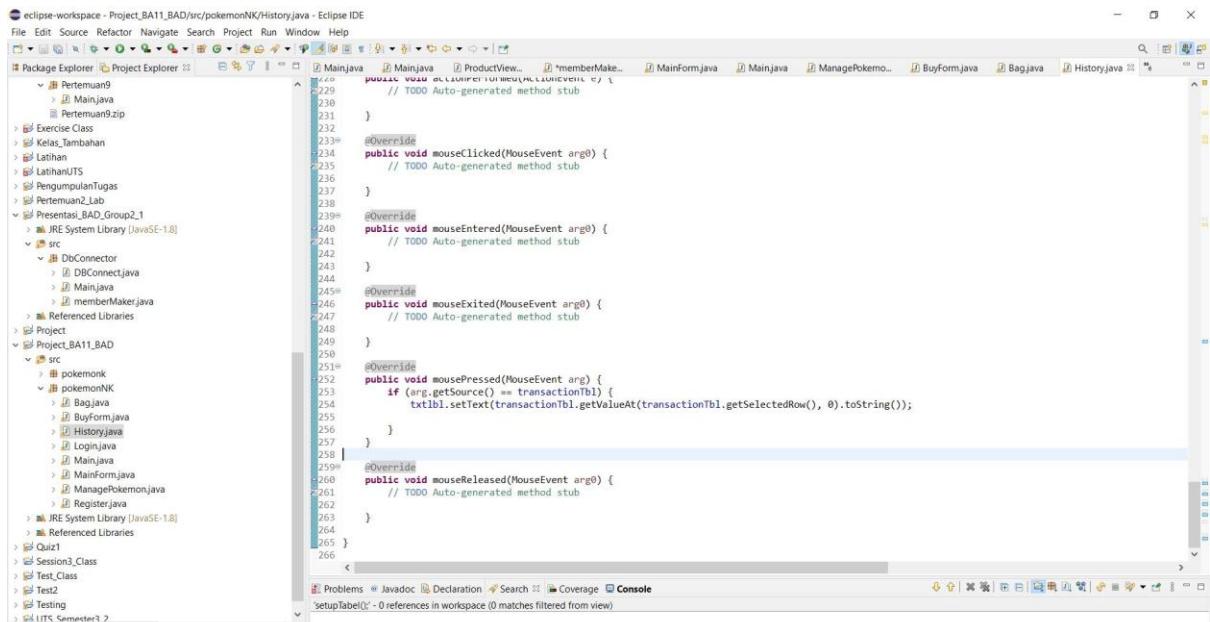
The code is used to set up a table named `detailTable` with columns corresponding to the database schema. It retrieves data from the `detailtransaction` and `headertransaction` tables, joining them on the `TransactionId` field. The resulting rows are added to a `Vector` and then inserted into the table.

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer View:** Shows the project structure. The `src` folder contains packages like `Permenan9`, `Exercise Class`, `Kelas_Tambahan`, `Lathian`, `LathianUTS`, `PengumpulanTugas`, `Permenan9_Latihan`, `Presentasi_BAD_Group_2_1`, and `JRE System Library [JavaSE-1.8]`. The `Project_BA11_BAD` project has a `src` folder containing `pokemon` and `pokemonNK` packages, which in turn contain classes like `Bag.java`, `BuyForm.java`, `History.java`, `Login.java`, `Main.java`, `MainForm.java`, `ManagePokemon.java`, and `Register.java`.
- Code Editor View:** Displays the `History.java` file. The code implements a `TableModel` interface and defines a `DetailPanel` class. It includes methods for setting up the top panel, creating a detail table, and adding listeners to the scroll pane.
- Bottom Status Bar:** Shows the status bar with "Writable", "Smart Insert", "111:44:3007", and other system icons.

The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** eclipse-workspace - Project_BA11_BAD/src/pokemonNK/History.java - Eclipse IDE
- Toolbar:** File Edit Source Refactor Navigate Project Run Window Help
- Project Explorer:** Shows the project structure with packages like Pertemanan, Kelas_Tambahan, Lathian, LathianJTS, PengumpulanTugas, Pertemanan2_Lab, and Presentasi_BAO_Group2_1.
- Code Editor:** Displays the `History.java` file content, which includes imports for `java.awt`, `java.awt.event`, `java.sql`, and `javax.swing`. The code implements a `JFrame` for a history application, setting up panels for bottom, center, and detail sections, and handling action events for a dispose button.
- Bottom Bar:** Shows tabs for Problems, Javadoc, Declaration, Search, Coverage, and Console. The status bar indicates "SetupTable(0) - 0 references in workspace (0 matches filtered from view)" and the date/time as 75-22-1976.



```

eclipse-workspace - Project_BA11_BAD/src/pokemonNK/History.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer Project Explorer MainJava MainJava ProductView_ *memberMake... MainForm.java MainJava ManagePoko... BuyForm.java Bag.java History.java
Perfuman9
Main.java
Perfuman9.zip
Exercise Class
Kelas_Tambahan
Lathan
LathanJTS
PengumpulanTugas
Perfuman2_Lab
Presentasi_BAD_Group2_1
JRE System Library [JavaSE-1.8]
src
DbConnector
DBCConnect.java
Main.java
memberMaker.java
Referenced Libraries
Project
Project_BA11_BAD
src
pokemon
pokemonNK
Bag.java
BuyForm.java
History.java
Login.java
Main.java
MainForm.java
ManagePokemon.java
Register.java
JRE System Library [JavaSE-1.8]
Referenced Libraries
Quiz1
Session3_Class
Test_Class
Test2
Testing
ITS_Semester3

```

```

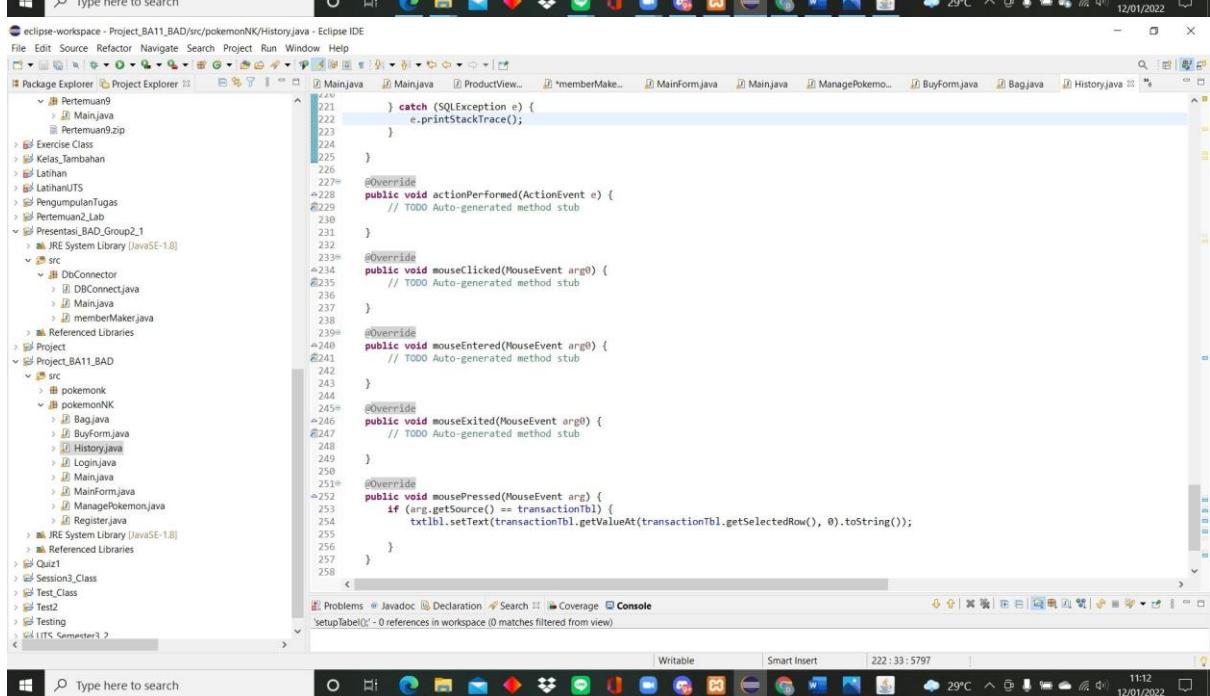
public void mouseClicked(MouseEvent arg0) {
    // TODO Auto-generated method stub
}
@Override
public void mouseEntered(MouseEvent arg0) {
    // TODO Auto-generated method stub
}
@Override
public void mouseExited(MouseEvent arg0) {
    // TODO Auto-generated method stub
}
@Override
public void mousePressed(MouseEvent arg0) {
    if (arg.getSource() == transactionTbl) {
        txtlbl.setText(transactionTbl.getValueAt(transactionTbl.getSelectedRow(), 0).toString());
    }
}
@Override
public void mouseReleased(MouseEvent arg0) {
    // TODO Auto-generated method stub
}

221 } catch (SQLException e) {
222     e.printStackTrace();
223 }
224
225 }
226
227 @Override
228 public void actionPerformed(ActionEvent e) {
229     // TODO Auto-generated method stub
230 }
231
232 @Override
233 public void mouseClicked(MouseEvent arg0) {
234     // TODO Auto-generated method stub
235 }
236
237 @Override
238 public void mouseEntered(MouseEvent arg0) {
239     // TODO Auto-generated method stub
240 }
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266

```

Problems Javadoc Declaration Search Coverage Console

11:12 12/01/2022



```

eclipse-workspace - Project_BA11_BAD/src/pokemonNK/History.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer Project Explorer MainJava MainJava ProductView_ *memberMake... MainForm.java MainJava ManagePoko... BuyForm.java Bag.java History.java
Perfuman9
Main.java
Perfuman9.zip
Exercise Class
Kelas_Tambahan
Lathan
LathanJTS
PengumpulanTugas
Perfuman2_Lab
Presentasi_BAD_Group2_1
JRE System Library [JavaSE-1.8]
src
DbConnector
DBCConnect.java
Main.java
memberMaker.java
Referenced Libraries
Project
Project_BA11_BAD
src
pokemon
pokemonNK
Bag.java
BuyForm.java
History.java
Login.java
Main.java
MainForm.java
ManagePokemon.java
Register.java
JRE System Library [JavaSE-1.8]
Referenced Libraries
Quiz1
Session3_Class
Test_Class
Test2
Testing
ITS_Semester3

```

```

public void mouseClicked(MouseEvent arg0) {
    // TODO Auto-generated method stub
}
@Override
public void mouseEntered(MouseEvent arg0) {
    // TODO Auto-generated method stub
}
@Override
public void mouseExited(MouseEvent arg0) {
    // TODO Auto-generated method stub
}
@Override
public void mousePressed(MouseEvent arg0) {
    if (arg.getSource() == transactionTbl) {
        txtlbl.setText(transactionTbl.getValueAt(transactionTbl.getSelectedRow(), 0).toString());
    }
}
@Override
public void mouseReleased(MouseEvent arg0) {
    // TODO Auto-generated method stub
}

221 } catch (SQLException e) {
222     e.printStackTrace();
223 }
224
225 }
226
227 @Override
228 public void actionPerformed(ActionEvent e) {
229     // TODO Auto-generated method stub
230 }
231
232 @Override
233 public void mouseClicked(MouseEvent arg0) {
234     // TODO Auto-generated method stub
235 }
236
237 @Override
238 public void mouseEntered(MouseEvent arg0) {
239     // TODO Auto-generated method stub
240 }
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258

```

Problems Javadoc Declaration Search Coverage Console

11:12 12/01/2022

- Ketika User mendapatkan role ‘Player’, maka user dapat menekan menu ‘View Transaction History’ untuk melihat kapan saja user membeli pokemon
- Ketika sudah berada di menu ‘View Transaction History’, user dapat menekan beberapa id transaksi dan waktu transaksi yang ada
- Kemudian, sistem akan menunjukkan berbagai informasi, mulai dari ‘TransactionId, PokemonId, PokemonName, PokemonLevel, PokemonType, dan Quantity’.