

**LAPORAN UJIAN TENGAH SEMESTER
COMPUTER DAN BAHASA PEMOGRAMAN (CODING)**



DISUSUN OLEH:

- 1. Aulia Dwi Rahmadhani
NPM: G1A023043**
- 2. Merischa Theresia Hutauruk
NPM: G1A023071**

KELAS: A

Asisten Dosen :

Randi julian Saputra (G1A019066)

Dosen Pengampu :

Arie Vatesia,ST. M.TI.,P.Hd

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS BENGKULU
2023**

LANDASAN TEORI

A. Pemrograman Berbasis Web

Web atau istilah lengkapnya World Wide Web (WWW) dikembangkan pertama kali oleh Tim Berners-Lee pada tahun 1989. Tim merupakan bagian dari CERN, sebuah laboratorium penelitian ilmiah di Genewa, Swiss. Pada awalnya, Tim mengusulkan WWW sebagai suatu cara berbagi dokumen diantara para peneliti. Dokumen online dapat diakses melalui alamat unik yang disebut Universal Resources Locator (URL). Dokumen dapat saling berkaitan melalui suatu link. WWW menggunakan protokol HTTP (Hypertext Transfer Protocol) untuk berkomunikasi di antara komputer dalam suatu jaringan. Selanjutnya WWW tidak hanya dikembangkan untuk keperluan para peneliti, namun juga dikembangkan untuk kalangan pendidikan, bisnis dan perorangan. Disisi lain, CERN dan National Center for Supercomputer Applications (NCSA) di University of Illinois juga mengembangkan suatu perangkat lunak yang berjalan di atas jaringan internet dan berfungsi sebagai penyedia dokumen online. Perangkat lunak tersebut dikenal dengan web server. Selain itu, untuk membuat dan menghubungkan halamanhalaman web Tim juga mengembangkan bahasa mark up yang disebut HTML (Hypertext Markup Language

B. PHP

PHP merupakan singkatan dari Hypertext Preprocessor yang merupakan salah satu bahasa pemrograman yng digunakan dalam pembuatan dan pengembangan situs web yang dikombinasikan dengan HTML. Serverside merupakan sifat PHP, yang artinya adalah bahasa yang berbentuk script yang disimpan dan dijalankan melalui komputer server ataupun web server dan hasilnya akan dikirim ke komputer client ataupun web browser yang di dalamnya terdapat script HTML atau Hypertext Mark up Language.

Dalam pembuatan sebuah web,script PHP seringkali digunakan untuk mengatur perumusan logika Di dalam PHP terdapat variabel, yang mana variabel tersebut merupakan tempat untuk menampung sebuah data. Variabel didefenisikan dengan simbol dollar (\$) yang diikuti dengan nama variabelnya. Dalam menentukan simbol pada PHP dapat menggunakan operator. Yang mana operator merupakan suatu simbol yang berfungsi untuk menyusun sebuah ekspresi dan operasi. Operator PHP diantaranya meliputi operator aritmatika, operator perbandingan, dan operator logika. Operator aritmatika adalah operator yang berfungsi untuk melakukan operasi aritmatika. Seperti penjumlahan, pengurangan, pembagian, perkalian dan sisa pembagian. Dan selanjutnya operator perbandingan, yang merupakan sebuah simbol untuk melakukan perbandingan antara dua buah operand. Misalnya dalam penulisam a lebih besar dari b dapat dituliskan

menjadi $a > b$. Dan yang terakhir adalah operator logika. Yang dimaksud operator logika merupakan simbol operator dalam menyusun kalimat berlogika. Misalnya operator AND dituliskan menjadi simbol `&&`, OR taupun menadi simbol `||`, dan XOR dituliskan menjadi simbol `!`. Di dalam PHP juga terdapat pernyataan suatu kondisi, seperti: [1] pernyataan If – Else, [2] pernyataan If – Else If dan [3] pernyataan Switce

C. Komputer dan Arsitektur Komputer

Komputer adalah mesin elektronik yang dapat mengolah data. Arsitektur komputer mencakup komponen-komponen utama seperti CPU (Central Processing Unit), RAM (Random Access Memory), penyimpanan sekunder, dan periferal.

D. Sistem Operasi

Sistem operasi adalah perangkat lunak yang mengelola sumber daya komputer, seperti manajemen memori, sistem berkas, dan antarmuka pengguna. Contoh sistem operasi meliputi Windows, macOS, dan Linux.

E. Bahasa Pemrograman

Bahasa pemrograman adalah cara untuk berkomunikasi dengan komputer melalui kode. Contoh bahasa pemrograman termasuk Python, Java, C++, dan JavaScript.

F. Algoritma dan Struktur Data

Algoritma adalah langkah-langkah yang diperlukan untuk menyelesaikan tugas tertentu. Struktur data adalah cara penyimpanan dan pengorganisasian data dalam program.

G. Konsep Pemrograman

Ini mencakup konsep seperti variabel, loop, percabangan, fungsi, dan objek. Paradigma pemrograman termasuk pemrograman prosedural, pemrograman berorientasi objek, dan lainnya.

H. Pemecahan Masalah

Pemrograman melibatkan pemecahan masalah, yaitu mengidentifikasi masalah, merancang solusi, dan menerapkannya dalam kode.

I. Kompilasi vs. Interpretasi

Beberapa bahasa pemrograman dikompilasi menjadi bahasa mesin sebelum dijalankan, sementara yang lain diinterpretasi baris demi baris saat dijalankan.

J. Pengembangan Perangkat Lunak

Proses pengembangan perangkat lunak melibatkan analisis kebutuhan, perancangan, implementasi, pengujian, dan pemeliharaan.

K. Keamanan Komputer

Keamanan komputer mencakup prinsip-prinsip dan praktik untuk melindungi sistem dan data dari ancaman seperti malware dan peretas.

L. Pengembangan Web

Konsep dasar pengembangan web mencakup HTML (Hypertext Markup Language) untuk struktur halaman web, CSS (Cascading Style Sheets) untuk tampilan, dan JavaScript untuk interaktivitas.

M. Jaringan dan Protokol

Jaringan komputer adalah infrastruktur yang memungkinkan komunikasi antara komputer. Protokol seperti TCP/IP digunakan untuk mengatur komunikasi di internet.

N. Pengembangan Perangkat Bergerak (Mobile Development)

Ini mencakup pengembangan aplikasi untuk perangkat seluler, seperti Android (menggunakan bahasa Java atau Kotlin) dan iOS (menggunakan bahasa Swift).

O. Pemrograman Paralel dan Berdistribusi

Ini melibatkan pengembangan aplikasi yang dapat menjalankan tugas-tugas secara paralel di beberapa komputer atau inti CPU.

P. Bahasa Pemrograman

Anda perlu memilih bahasa pemrograman untuk mengembangkan permainan Tic-Tac-Toe. Beberapa bahasa yang umum digunakan untuk pengembangan permainan termasuk Python, Java, JavaScript, dan C++.

Q. Logika Tic-Tac-Toe

Anda harus memahami aturan dasar permainan Tic-Tac-Toe, seperti papan 3x3, giliran pemain, cara memenangkan permainan (tiga tanda 'X' atau 'O' dalam baris, kolom, atau diagonal), serta cara mengidentifikasi permainan seri atauimbang.

R. Struktur Data

Anda akan menggunakan struktur data, seperti matriks atau array, untuk merepresentasikan papan permainan. Ini akan memungkinkan Anda melacak status setiap sel di papan.

S. Algoritma Permainan

Anda perlu mengembangkan algoritma yang memungkinkan pemain untuk bermain secara cerdas. Ini mungkin melibatkan pemikiran ke depan dan penentuan langkah terbaik.

T. Interaksi Pengguna

Anda perlu memahami cara berinteraksi dengan pengguna, baik melalui antarmuka teks di terminal atau grafis jika Anda mengembangkan permainan dengan antarmuka pengguna berbasis grafis.

U. Loop dan Percabangan

Anda akan menggunakan loop untuk mengatur giliran pemain, dan percabangan untuk memeriksa kapan permainan telah selesai dan menentukan pemenangnya.

V. Fungsi

Mengorganisasi kode Anda dalam fungsi-fungsi yang logis akan membuat kode lebih mudah dikelola dan memahami.

W. Pengujian dan Debugging

Penting untuk menguji permainan secara menyeluruh dan mengatasi masalah (bug) yang mungkin muncul selama pengembangan.

X. Grafika (Opsional)

Jika Anda ingin membuat versi Tic-Tac-Toe yang lebih visual, Anda perlu memahami konsep-konsep dasar dalam grafika komputer, seperti rendering objek dan antarmuka pengguna grafis (GUI).

Y. Kreativitas

Anda memiliki kebebasan untuk menambahkan elemen kreatif ke dalam permainan, seperti animasi, suara, atau mode permainan tambah

CODING

```
1.  /*
2.   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
   * default.txt to change this license
3.   * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to
   * edit this template
4.   */
5.
6.  import javax.swing.JOptionPane; /* Memanggil Library javax.swing */
7.
8.  public class TicTacToe extends javax.swing.JFrame { //kelas TicTacToeToe
   yang dibuat dari JFrame
9.
10.     int nilai, nilaiX, nilaiO; //Mendelrasaikan variabel nilai, nilaiX,
    nilaiO
11.
12.     public TicTacToe() { //Kelas TicTacToeToe
13.         initComponents();
14.         nilai = 1; //Menentukan value variabel nilai
15.         nilaiX = 0; //Menentukan value variabel nilaiX
16.         nilaiO = 0; //Menentukan value variabel nilaiO
17.     }
18.
19.     void pemenang() {
20.         if ("X".equals(btn1.getText()) && "X".equals(btn2.getText()) &&
            "X".equals(btn3.getText())) {
21.             JOptionPane.showMessageDialog(rootPane, "Pemenangnya adalah X,
                Selamat X !!!");
22.         } else if ("X".equals(btn4.getText()) &&
            "X".equals(btn5.getText()) && "X".equals(btn6.getText())) {
23.             JOptionPane.showMessageDialog(rootPane, "Pemenangnya adalah X,
                Selamat X !!!");
24.         } else if ("X".equals(btn7.getText()) &&
            "X".equals(btn8.getText()) && "X".equals(btn9.getText())) {
25.             JOptionPane.showMessageDialog(rootPane, "Pemenangnya adalah X,
                Selamat X !!!");
26.         } else if ("X".equals(btn1.getText()) &&
            "X".equals(btn5.getText()) && "X".equals(btn9.getText())) {
27.             JOptionPane.showMessageDialog(rootPane, "Pemenangnya adalah X,
                Selamat X !!!");
28.         } else if ("X".equals(btn3.getText()) &&
            "X".equals(btn5.getText()) && "X".equals(btn7.getText())) {
29.             JOptionPane.showMessageDialog(rootPane, "Pemenangnya adalah X,
                Selamat X !!!");
30.         } else if ("X".equals(btn1.getText()) &&
            "X".equals(btn4.getText()) && "X".equals(btn7.getText())) {
31.             JOptionPane.showMessageDialog(rootPane, "Pemenangnya adalah X,
                Selamat X !!!");
32.         }
```

```

33.         } else if ("X".equals(btn2.getText()) &&
    "X".equals(btn5.getText()) && "X".equals(btn8.getText())) {
34.             JOptionPane.showMessageDialog(rootPane, "Pemenangnya adalah X,
    Selamat X !!!");
35.         } else if ("X".equals(btn3.getText()) &&
    "X".equals(btn6.getText()) && "X".equals(btn9.getText())) {
36.             JOptionPane.showMessageDialog(rootPane, "Pemenangnya adalah X,
    Selamat X !!!");
37.         }
38.
39.         if ("O".equals(btn1.getText()) && "O".equals(btn2.getText()) &&
    "O".equals(btn3.getText())) {
40.             JOptionPane.showMessageDialog(rootPane, "Pemenangnya adalah O,
    Selamat O !!!");
41.         } else if ("O".equals(btn4.getText()) &&
    "O".equals(btn5.getText()) && "O".equals(btn6.getText())) {
42.             JOptionPane.showMessageDialog(rootPane, "Pemenangnya adalah O,
    Selamat O !!!");
43.         } else if ("O".equals(btn7.getText()) &&
    "O".equals(btn8.getText()) && "O".equals(btn9.getText())) {
44.             JOptionPane.showMessageDialog(rootPane, "Pemenangnya adalah O,
    Selamat O !!!");
45.         } else if ("O".equals(btn1.getText()) &&
    "O".equals(btn5.getText()) && "O".equals(btn9.getText())) {
46.             JOptionPane.showMessageDialog(rootPane, "Pemenangnya adalah O,
    Selamat O !!!");
47.         } else if ("O".equals(btn3.getText()) &&
    "O".equals(btn5.getText()) && "O".equals(btn7.getText())) {
48.             JOptionPane.showMessageDialog(rootPane, "Pemenangnya adalah O,
    Selamat O !!!");
49.         } else if ("O".equals(btn1.getText()) &&
    "O".equals(btn4.getText()) && "O".equals(btn7.getText())) {
50.             JOptionPane.showMessageDialog(rootPane, "Pemenangnya adalah O,
    Selamat O !!!");
51.         } else if ("O".equals(btn2.getText()) &&
    "O".equals(btn5.getText()) && "O".equals(btn8.getText())) {
52.             JOptionPane.showMessageDialog(rootPane, "Pemenangnya adalah O,
    Selamat O !!!");
53.         } else if ("O".equals(btn3.getText()) &&
    "O".equals(btn6.getText()) && "O".equals(btn9.getText())) {
54.             JOptionPane.showMessageDialog(rootPane, "Pemenangnya adalah O,
    Selamat O !!!");
55.         }
56.
57.
58.     }
59.
60.     void draw () {
61.         if ("O".equals(btn2.getText()) && "X".equals(btn1.getText()) &&
    "X".equals(btn3.getText()) && "O".equals(btn4.getText()) &&
    "O".equals(btn5.getText()) && "X".equals(btn6.getText()) &&

```

```

        "X".equals(btn7.getText()) && btn8.getText() == "X" && btn9.getText() ==
        "O"
62.    ) {
63.    } else if ("O".equals(btn1.getText()) &&
        "X".equals(btn2.getText()) && "O".equals(btn3.getText()) &&
        "X".equals(btn4.getText())
64.    && btn5.getText() == "X" && btn6.getText() == "O" &&
        btn7.getText() == "X" && btn8.getText() == "O" && btn9.getText() == "X"
65.    ) {
66.        JOptionPane.showMessageDialog(rootPane, "Game Draw !!!");
67.    }
68.
69.    else if ("X".equals(btn1.getText()) && "X".equals(btn2.getText())
        && "O".equals(btn3.getText()) && "O".equals(btn4.getText())
70.    && "O".equals(btn5.getText()) &&
        "X".equals(btn6.getText()) && btn7.getText() == "X" && btn8.getText() ==
        "X" && btn9.getText() == "O"
71.    ) {
72.        JOptionPane.showMessageDialog(rootPane, "Game Draw !!!");
73.    }
74.
75.    else if ("X".equals(btn5.getText())
        && "O".equals(btn1.getText()) &&
        "O".equals(btn2.getText()) && "X".equals(btn3.getText()) &&
        "X".equals(btn4.getText()) && btn6.getText() == "O" && btn7.getText() ==
        "O" && btn8.getText() == "X" && btn9.getText() == "X"
77.    ) {
78.        JOptionPane.showMessageDialog(rootPane, "Game Draw !!!");
79.    }
80.
81.    else {
82.        JOptionPane.showMessageDialog(rootPane, "Game Draw !!!");
83.    }
84.
85. }
86.
87. void reset() {
88.     nilai = 1;
89.     nilaiX = 0;
90.     nilaiO = 0;
91.
92.     lblX.setText("0");
93.     lblO.setText("0");
94.
95.     btn1.setText("");
96.     btn1.setEnabled(true);
97.     btn2.setText("");
98.     btn2.setEnabled(true);
99.     btn3.setText("");
100.    btn3.setEnabled(true);
101.    btn4.setText("");

```



```

102.         btn4.setEnabled(true);
103.         btn5.setText("");
104.         btn5.setEnabled(true);
105.         btn6.setText("");
106.         btn6.setEnabled(true);
107.         btn7.setText("");
108.         btn7.setEnabled(true);
109.         btn8.setText("");
110.         btn8.setEnabled(true);
111.         btn9.setText("");
112.         btn9.setEnabled(true);
113.     }
114.
115.     /**
116.      * This method is called from within the constructor to initialize
117.      * the form.
118.      * WARNING: Do NOT modify this code. The content of this method is
119.      * always
120.      * regenerated by the Form Editor.
121.      */
122.     @SuppressWarnings("unchecked")
123.     // <editor-fold defaultstate="collapsed" desc="Generated
124.     Code">
125.     private void initComponents() {
126.
127.         jLabel1 = new javax.swing.JLabel();
128.         jPanel2 = new javax.swing.JPanel();
129.         jPanel1 = new javax.swing.JPanel();
130.         btn3 = new javax.swing.JButton();
131.         btn1 = new javax.swing.JButton();
132.         btn4 = new javax.swing.JButton();
133.         btn5 = new javax.swing.JButton();
134.         btn6 = new javax.swing.JButton();
135.         btn7 = new javax.swing.JButton();
136.         lbl0 = new javax.swing.JLabel();
137.         lblX = new javax.swing.JLabel();
138.         resetbtn = new javax.swing.JButton();
139.         btn2 = new javax.swing.JButton();
140.         btn8 = new javax.swing.JButton();
141.         btn9 = new javax.swing.JButton();
142.         jPanel3 = new javax.swing.JPanel();
143.         jLabel2 = new javax.swing.JLabel();
144.
145.         jLabel1.setText("jLabel1");
146.
147.         javax.swing.GroupLayout jPanel2Layout = new
148.         javax.swing.GroupLayout(jPanel2);
149.         jPanel2.setLayout(jPanel2Layout);
150.         jPanel2Layout.setHorizontalGroup(
151.             jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

147.         .addGap(0, 100, Short.MAX_VALUE)
148.     );
149.     JPanel12Layout.setVerticalGroup(
150.         JPanel12Layout.createParallelGroup(javax.swing.GroupLayout.AL
            ignment.LEADING)
151.         .addGap(0, 100, Short.MAX_VALUE)
152.     );
153.
154.     setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLO
        SE);
155.
156.     JPanel11.setBackground(new java.awt.Color(204, 255, 255));
157.     JPanel11.setBorder(javax.swing.BorderFactory.createBevelBorder(ja
        vax.swing.border.BevelBorder.LOWERED));
158.
159.     btn3.setBackground(new java.awt.Color(0, 102, 51));
160.     btn3.setFont(new java.awt.Font("Javanese Text", 1, 24)); //
        NOI18N
161.     btn3.setForeground(new java.awt.Color(0, 102, 102));
162.     btn3.addActionListener(new java.awt.event.ActionListener() {
163.         public void actionPerformed(java.awt.event.ActionEvent evt)
            {
164.             btn3ActionPerformed(evt);
165.         }
166.     });
167.
168.     btn1.setBackground(new java.awt.Color(0, 102, 51));
169.     btn1.setFont(new java.awt.Font("Javanese Text", 1, 24)); //
        NOI18N
170.     btn1.setForeground(new java.awt.Color(0, 102, 102));
171.     btn1.addActionListener(new java.awt.event.ActionListener() {
172.         public void actionPerformed(java.awt.event.ActionEvent evt)
            {
173.             btn1ActionPerformed(evt);
174.         }
175.     });
176.
177.     btn4.setBackground(new java.awt.Color(0, 102, 51));
178.     btn4.setFont(new java.awt.Font("Javanese Text", 1, 24)); //
        NOI18N
179.     btn4.setForeground(new java.awt.Color(0, 102, 102));
180.     btn4.addActionListener(new java.awt.event.ActionListener() {
181.         public void actionPerformed(java.awt.event.ActionEvent evt)
            {
182.             btn4ActionPerformed(evt);
183.         }
184.     });
185.
186.     btn5.setBackground(new java.awt.Color(0, 102, 51));
187.     btn5.setFont(new java.awt.Font("Javanese Text", 1, 24)); //
        NOI18N

```

```

188.         btn5.setForeground(new java.awt.Color(0, 102, 102));
189.         btn5.addActionListener(new java.awt.event.ActionListener() {
190.             public void actionPerformed(java.awt.event.ActionEvent evt)
191.             {
192.                 btn5ActionPerformed(evt);
193.             }
194.         });
195.         btn6.setBackground(new java.awt.Color(0, 102, 51));
196.         btn6.setFont(new java.awt.Font("Javanese Text", 1, 24)); //
    NOI18N
197.         btn6.setForeground(new java.awt.Color(0, 102, 102));
198.         btn6.addActionListener(new java.awt.event.ActionListener() {
199.             public void actionPerformed(java.awt.event.ActionEvent evt)
200.             {
201.                 btn6ActionPerformed(evt);
202.             }
203.         });
204.         btn7.setBackground(new java.awt.Color(0, 102, 51));
205.         btn7.setFont(new java.awt.Font("Javanese Text", 1, 24)); //
    NOI18N
206.         btn7.setForeground(new java.awt.Color(0, 102, 102));
207.         btn7.addActionListener(new java.awt.event.ActionListener() {
208.             public void actionPerformed(java.awt.event.ActionEvent evt)
209.             {
210.                 btn7ActionPerformed(evt);
211.             }
212.         });
213.         lbl0.setFont(new java.awt.Font("Arial", 1, 24)); // NOI18N
214.         lbl0.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
215.         lbl0.setText("0 kali");
216.
217.         lblX.setFont(new java.awt.Font("Arial", 1, 24)); // NOI18N
218.         lblX.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
219.         lblX.setText("0 kali");
220.
221.         resetbtn.setBackground(new java.awt.Color(0, 102, 0));
222.         resetbtn.setFont(new java.awt.Font("Bahnschrift", 1, 24)); //
    NOI18N
223.         resetbtn.setForeground(new java.awt.Color(255, 255, 255));
224.         resetbtn.setText("RESET");
225.         resetbtn.setHorizontalTextPosition(javax.swing.SwingConstants.RI
    GHT);
226.         resetbtn.addActionListener(new java.awt.event.ActionListener() {
227.             public void actionPerformed(java.awt.event.ActionEvent evt)
228.             {
229.                 resetbtnActionPerformed(evt);
230.             }
231.         });

```

```

231.
232.         btn2.setBackground(new java.awt.Color(0, 102, 51));
233.         btn2.setFont(new java.awt.Font("Javanese Text", 1, 24)); //
    NOI18N
234.         btn2.setForeground(new java.awt.Color(0, 102, 102));
235.         btn2.addActionListener(new java.awt.event.ActionListener() {
236.             public void actionPerformed(java.awt.event.ActionEvent evt)
237.             {
238.                 btn2ActionPerformed(evt);
239.             }
240.         });
241.         btn8.setBackground(new java.awt.Color(0, 102, 51));
242.         btn8.setFont(new java.awt.Font("Javanese Text", 1, 24)); //
    NOI18N
243.         btn8.setForeground(new java.awt.Color(0, 102, 102));
244.         btn8.addActionListener(new java.awt.event.ActionListener() {
245.             public void actionPerformed(java.awt.event.ActionEvent evt)
246.             {
247.                 btn8ActionPerformed(evt);
248.             }
249.         });
250.         btn9.setBackground(new java.awt.Color(0, 102, 51));
251.         btn9.setFont(new java.awt.Font("Javanese Text", 1, 24)); //
    NOI18N
252.         btn9.setForeground(new java.awt.Color(0, 102, 102));
253.         btn9.addActionListener(new java.awt.event.ActionListener() {
254.             public void actionPerformed(java.awt.event.ActionEvent evt)
255.             {
256.                 btn9ActionPerformed(evt);
257.             }
258.         });
259.         jPanel3.setBackground(new java.awt.Color(153, 153, 255));
260.
261.         jLabel2.setFont(new java.awt.Font("Kristen ITC", 1, 36)); //
    NOI18N
262.         jLabel2.setForeground(new java.awt.Color(255, 255, 255));
263.         jLabel2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER
264.         );
265.         jLabel2.setText("Let's Play Tic Tac Toe");
266.         javax.swing.GroupLayout jPanel3Layout = new
267.         javax.swing.GroupLayout(jPanel3);
268.         jPanel3.setLayout(jPanel3Layout);
269.         jPanel3Layout.setHorizontalGroup(
270.             jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Al
271.             ignment.LEADING)

```

```
272.                .addComponent(jLabel12)
273.                .addContainerGap(280, Short.MAX_VALUE))
274.            );
275.            jPanel3Layout.setVerticalGroup(
276.                jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.AL
                ignment.LEADING)
277.                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
                jPanel3Layout.createSequentialGroup())
278.                .addContainerGap(40, Short.MAX_VALUE)
279.                .addComponent(jLabel12)
280.                .addGap(31, 31, 31))
281.            );
282.
283.            javax.swing.GroupLayout jPanel1Layout = new
                javax.swing.GroupLayout(jPanel1);
284.            jPanel1.setLayout(jPanel1Layout);
285.            jPanel1Layout.setHorizontalGroup(
286.                jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.AL
                ignment.LEADING)
287.                .addGroup(jPanel1Layout.createSequentialGroup()
288.                    .addGap(163, 163, 163)
289.                    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.
                GroupLayout.Alignment.LEADING)
290.                        .addGroup(jPanel1Layout.createSequentialGroup()
291.                            .addComponent(btn7,
                javax.swing.GroupLayout.PREFERRED_SIZE, 56,
                javax.swing.GroupLayout.PREFERRED_SIZE)
292.                            .addPreferredGap(javax.swing.LayoutStyle.Compone
                ntPlacement.RELATED)
293.                            .addComponent(btn8,
                javax.swing.GroupLayout.PREFERRED_SIZE, 56,
                javax.swing.GroupLayout.PREFERRED_SIZE)
294.                            .addPreferredGap(javax.swing.LayoutStyle.Compone
                ntPlacement.RELATED)
295.                            .addComponent(btn9,
                javax.swing.GroupLayout.PREFERRED_SIZE, 56,
                javax.swing.GroupLayout.PREFERRED_SIZE))
296.                        .addGroup(jPanel1Layout.createParallelGroup(javax.sw
                ینگ.GroupLayout.Alignment.TRAILING)
297.                            .addGroup(jPanel1Layout.createSequentialGroup()
298.                                .addComponent(btn4,
                javax.swing.GroupLayout.PREFERRED_SIZE, 56,
                javax.swing.GroupLayout.PREFERRED_SIZE)
299.                                .addPreferredGap(javax.swing.LayoutStyle.Com
                ponentPlacement.RELATED)
300.                                .addComponent(btn5,
                javax.swing.GroupLayout.PREFERRED_SIZE, 56,
                javax.swing.GroupLayout.PREFERRED_SIZE)
301.                                .addPreferredGap(javax.swing.LayoutStyle.Com
                ponentPlacement.RELATED))
```

```

302.                .addComponent(btn6,
    javax.swing.GroupLayout.PREFERRED_SIZE, 56,
    javax.swing.GroupLayout.PREFERRED_SIZE))
303.                .addGroup(jPanel1Layout.createSequentialGroup())
304.                .addComponent(btn1,
    javax.swing.GroupLayout.PREFERRED_SIZE, 56,
    javax.swing.GroupLayout.PREFERRED_SIZE)
305.                .addPreferredGap(javax.swing.LayoutStyle.Com
    ponentPlacement.RELATED)
306.                .addComponent(btn2,
    javax.swing.GroupLayout.PREFERRED_SIZE, 56,
    javax.swing.GroupLayout.PREFERRED_SIZE)
307.                .addPreferredGap(javax.swing.LayoutStyle.Com
    ponentPlacement.RELATED)
308.                .addComponent(btn3,
    javax.swing.GroupLayout.PREFERRED_SIZE, 56,
    javax.swing.GroupLayout.PREFERRED_SIZE))))
309.                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.
    GroupLayout.Alignment.LEADING)
310.                .addGroup(jPanel1Layout.createSequentialGroup())
311.                .addGap(102, 102, 102)
312.                .addComponent(resetbtn,
    javax.swing.GroupLayout.PREFERRED_SIZE, 129,
    javax.swing.GroupLayout.PREFERRED_SIZE))
313.                .addGroup(jPanel1Layout.createSequentialGroup())
314.                .addGap(125, 125, 125)
315.                .addGroup(jPanel1Layout.createParallelGroup(java
    x.swing.GroupLayout.Alignment.LEADING)
316.                .addComponent(lbl0,
    javax.swing.GroupLayout.PREFERRED_SIZE, 83,
    javax.swing.GroupLayout.PREFERRED_SIZE)
317.                .addComponent(lblX,
    javax.swing.GroupLayout.PREFERRED_SIZE, 83,
    javax.swing.GroupLayout.PREFERRED_SIZE))))
318.                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
    Short.MAX_VALUE))
319.                .addGroup(jPanel1Layout.createSequentialGroup())
320.                .addComponent(jPanel13,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)
321.                .addGap(0, 0, Short.MAX_VALUE))
322.            );
323.            jPanel1Layout.setVerticalGroup(
324.                jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Al
    ignment.LEADING)
325.                .addGroup(jPanel1Layout.createSequentialGroup())
326.                .addComponent(jPanel13,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)

```

```
327.         .addGap(25, 25, 25)
328.         .addGroup(jPanel1Layout.createParallelGroup(javax.swing.
    GroupLayout.Alignment.TRAILING)
329.         .addGroup(jPanel1Layout.createSequentialGroup())
330.         .addGroup(jPanel1Layout.createParallelGroup(java
    x.swing.GroupLayout.Alignment.LEADING, false)
331.         .addComponent(btn3,
    javax.swing.GroupLayout.DEFAULT_SIZE, 58, Short.MAX_VALUE)
332.         .addComponent(btn2,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
333.         .addComponent(btn1,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
334.         .addPreferredGap(javax.swing.LayoutStyle.Compone
    ntPlacement.RELATED)
335.         .addGroup(jPanel1Layout.createParallelGroup(java
    x.swing.GroupLayout.Alignment.LEADING)
336.         .addComponent(btn4,
    javax.swing.GroupLayout.PREFERRED_SIZE, 58,
    javax.swing.GroupLayout.PREFERRED_SIZE)
337.         .addComponent(btn5,
    javax.swing.GroupLayout.PREFERRED_SIZE, 58,
    javax.swing.GroupLayout.PREFERRED_SIZE)
338.         .addComponent(btn6,
    javax.swing.GroupLayout.PREFERRED_SIZE, 58,
    javax.swing.GroupLayout.PREFERRED_SIZE))
339.         .addPreferredGap(javax.swing.LayoutStyle.Compone
    ntPlacement.RELATED)
340.         .addGroup(jPanel1Layout.createParallelGroup(java
    x.swing.GroupLayout.Alignment.TRAILING)
341.         .addComponent(btn7,
    javax.swing.GroupLayout.PREFERRED_SIZE, 58,
    javax.swing.GroupLayout.PREFERRED_SIZE)
342.         .addComponent(btn8,
    javax.swing.GroupLayout.PREFERRED_SIZE, 58,
    javax.swing.GroupLayout.PREFERRED_SIZE)
343.         .addComponent(btn9,
    javax.swing.GroupLayout.PREFERRED_SIZE, 58,
    javax.swing.GroupLayout.PREFERRED_SIZE))
344.         .addContainerGap(70, Short.MAX_VALUE))
345.         .addGroup(jPanel1Layout.createSequentialGroup())
346.         .addComponent(lblX,
    javax.swing.GroupLayout.PREFERRED_SIZE, 39,
    javax.swing.GroupLayout.PREFERRED_SIZE)
347.         .addPreferredGap(javax.swing.LayoutStyle.Compone
    ntPlacement.UNRELATED)
348.         .addComponent(lbl0,
    javax.swing.GroupLayout.PREFERRED_SIZE, 39,
    javax.swing.GroupLayout.PREFERRED_SIZE)
349.         .addGap(42, 42, 42)
```



```

350.                .addComponent(resetbtn,
    javax.swing.GroupLayout.PREFERRED_SIZE, 46,
    javax.swing.GroupLayout.PREFERRED_SIZE)
351.                .addGap(60, 60, 60))))
352.            );
353.
354.            javax.swing.GroupLayout layout = new
    javax.swing.GroupLayout(getContentPane());
355.            getContentPane().setLayout(layout);
356.            layout.setHorizontalGroup(
357.                layout.createParallelGroup(javax.swing.GroupLayout.Alignment
    .LEADING)
358.                .addComponent(jPanel1,
    javax.swing.GroupLayout.PREFERRED_SIZE, 761,
    javax.swing.GroupLayout.PREFERRED_SIZE)
359.                );
360.            layout.setVerticalGroup(
361.                layout.createParallelGroup(javax.swing.GroupLayout.Alignment
    .LEADING)
362.                .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
363.                );
364.
365.            pack();
366.        }// </editor-fold>
367.
368.        private void btn4ActionPerformed(java.awt.event.ActionEvent evt)
    {
369.            // TODO add your handling code here:
370.            if (nilai == 1) {
371.                btn4.setText("X");
372.                btn4.setEnabled(false);
373.                nilai -= 1;
374.                nilaiX += 1;
375.                lblX.setText(String.valueOf(nilaiX + " klik"));
376.                pemenang();
377.                draw ();
378.            } else {
379.                btn4.setText("O");
380.                btn4.setEnabled(false);
381.                nilai += 1;
382.                nilaiO += 1;
383.                lblO.setText(String.valueOf(nilaiO + " klik"));
384.                pemenang();
385.                draw ();
386.            }
387.        }
388.
389.        private void btn5ActionPerformed(java.awt.event.ActionEvent evt)
    {
390.            // TODO add your handling code here:

```



```

391.         if (nilai == 1) {
392.             btn5.setText("X");
393.             btn5.setEnabled(false);
394.             nilai -= 1;
395.             nilaiX += 1;
396.             lblX.setText(String.valueOf(nilaiX + " klik"));
397.             pemenang();
398.             draw ();
399.         } else {
400.             btn5.setText("0");
401.             btn5.setEnabled(false);
402.             nilai += 1;
403.             nilai0 += 1;
404.             lbl0.setText(String.valueOf(nilai0 + " klik"));
405.             pemenang();
406.             draw ();
407.         }
408.     }
409.
410.     private void btn6ActionPerformed(java.awt.event.ActionEvent evt)
411.     {
412.         // TODO add your handling code here:
413.         if (nilai == 1) {
414.             btn6.setText("X");
415.             btn6.setEnabled(false);
416.             nilai -= 1;
417.             nilaiX += 1;
418.             lblX.setText(String.valueOf(nilaiX + " klik"));
419.             pemenang();
420.             draw ();
421.         } else {
422.             btn6.setText("0");
423.             btn6.setEnabled(false);
424.             nilai += 1;
425.             nilai0 += 1;
426.             lbl0.setText(String.valueOf(nilai0 + " klik"));
427.             pemenang();
428.             draw ();
429.         }
430.     }
431.
432.     private void btn3ActionPerformed(java.awt.event.ActionEvent evt)
433.     {
434.         // TODO add your handling code here:
435.         if (nilai == 1) {
436.             btn3.setText("X");
437.             btn3.setEnabled(false);
438.             nilai -= 1;
439.             nilaiX += 1;
440.             lblX.setText(String.valueOf(nilaiX + " klik"));
441.             pemenang();

```

```
440.         draw ();
441.     } else {
442.         btn3.setText("0");
443.         btn3.setEnabled(false);
444.         nilai += 1;
445.         nilai0 += 1;
446.         lbl0.setText(String.valueOf(nilai0 + " klik"));
447.         pemenang();
448.         draw ();
449.     }
450. }
451.
452. private void resetbtnActionPerformed(java.awt.event.ActionEvent evt)
453. {
454.     // TODO add your handling code here:
455.     reset();
456. }
457.
458. private void btn7ActionPerformed(java.awt.event.ActionEvent evt)
459. {
460.     // TODO add your handling code here:
461.     if (nilai == 1) {
462.         btn7.setText("X");
463.         btn7.setEnabled(false);
464.         nilai -= 1;
465.         nilaiX += 1;
466.         lblX.setText(String.valueOf(nilaiX + " klik"));
467.         pemenang();
468.         draw ();
469.     } else {
470.         btn7.setText("0");
471.         btn7.setEnabled(false);
472.         nilai += 1;
473.         nilai0 += 1;
474.         lbl0.setText(String.valueOf(nilai0 + " klik"));
475.         pemenang();
476.         draw ();
477.     }
478. }
479.
480. private void btn1ActionPerformed(java.awt.event.ActionEvent evt)
481. {
482.     // TODO add your handling code here:
483.     if (nilai == 1) {
484.         btn1.setText("X");
485.         btn1.setEnabled(false);
486.         nilai -= 1;
487.         nilaiX += 1;
488.         lblX.setText(String.valueOf(nilaiX + " klik"));
489.         pemenang();
490.         draw ();
491.     }
```

```

488.         } else {
489.             btn1.setText("0");
490.             btn1.setEnabled(false);
491.             nilai += 1;
492.             nilai0 += 1;
493.             lbl0.setText(String.valueOf(nilai0 + " klik"));
494.             pemenang();
495.             draw ();
496.         }
497.     }
498.
499.     private void btn2ActionPerformed(java.awt.event.ActionEvent evt)
    {
500.         // TODO add your handling code here:
501.         if (nilai == 1) {
502.             btn2.setText("X");
503.             btn2.setEnabled(false);
504.             nilai -= 1;
505.             nilaiX += 1;
506.             lblX.setText(String.valueOf(nilaiX + " klik"));
507.             pemenang();
508.             draw ();
509.         } else {
510.             btn2.setText("0");
511.             btn2.setEnabled(false);
512.             nilai += 1;
513.             nilai0 += 1;
514.             lbl0.setText(String.valueOf(nilai0 + " klik"));
515.             pemenang();
516.             draw ();
517.         } // TODO add your handling code here:
518.     }
519.
520.     private void btn8ActionPerformed(java.awt.event.ActionEvent evt)
    {
521.         // TODO add your handling code here:
522.         if (nilai == 1) {
523.             btn8.setText("X");
524.             btn8.setEnabled(false);
525.             nilai -= 1;
526.             nilaiX += 1;
527.             lblX.setText(String.valueOf(nilaiX + " klik"));
528.             pemenang();
529.             draw ();
530.         } else {
531.             btn8.setText("0");
532.             btn8.setEnabled(false);
533.             nilai += 1;
534.             nilai0 += 1;
535.             lbl0.setText(String.valueOf(nilai0 + " klik"));
536.             pemenang();

```

```

537.         draw ();
538.     }
539. }
540.
541. private void btn9ActionPerformed(java.awt.event.ActionEvent evt)
542. {
543.     // TODO add your handling code here:
544.     if (nilai == 1) {
545.         btn9.setText("X");
546.         btn9.setEnabled(false);
547.         nilai -= 1;
548.         nilaiX += 1;
549.         lblX.setText(String.valueOf(nilaiX + " klik"));
550.         pemenang();
551.         draw ();
552.     } else {
553.         btn9.setText("0");
554.         btn9.setEnabled(false);
555.         nilai += 1;
556.         nilaiO += 1;
557.         lblO.setText(String.valueOf(nilaiO + " klik"));
558.         pemenang();
559.         draw ();
560.     }
561. }
562. /**
563.  * @param args the command line arguments
564.  */
565. public static void main(String args[]) {
566.     /* Set the Nimbus look and feel */
567.     //<editor-fold defaultstate="collapsed" desc=" Look and feel
568.     setting code (optional) ">
569.     /* If Nimbus (introduced in Java SE 6) is not available, stay
570.     with the default look and feel.
571.     * For details see
572.     http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
573.     */
574.     try {
575.         for (javax.swing.UIManager.LookAndFeelInfo info :
576.             javax.swing.UIManager.getInstalledLookAndFeels()) {
577.             if ("Nimbus".equals(info.getName())) {
578.                 javax.swing.UIManager.setLookAndFeel(info.getClassName());
579.                 break;
580.             }
581.         }
582.     } catch (ClassNotFoundException | InstantiationException |
583.         IllegalAccessException | javax.swing.UnsupportedLookAndFeelException ex) {
584.         java.util.logging.Logger.getLogger(TicTacToe.class.getName())
585.             .log(java.util.logging.Level.SEVERE, null, ex);

```

```
580.     }
581.     //</editor-fold>
582.
583.     //</editor-fold>
584.
585.     /* Create and display the form */
586.     java.awt.EventQueue.invokeLater(() -> {
587.         new TicTacToe().setVisible(true);
588.     });
589. }
590.
591. // Variables declaration - do not modify
592. private javax.swing.JButton btn1;
593. private javax.swing.JButton btn2;
594. private javax.swing.JButton btn3;
595. private javax.swing.JButton btn4;
596. private javax.swing.JButton btn5;
597. private javax.swing.JButton btn6;
598. private javax.swing.JButton btn7;
599. private javax.swing.JButton btn8;
600. private javax.swing.JButton btn9;
601. private javax.swing.JLabel jLabel1;
602. private javax.swing.JLabel jLabel2;
603. private javax.swing.JPanel jPanel1;
604. private javax.swing.JPanel jPanel2;
605. private javax.swing.JPanel jPanel3;
606. private javax.swing.JLabel lbl0;
607. private javax.swing.JLabel lblX;
608. private javax.swing.JButton resetbtn;
609. // End of variables declaration
610. }
```


Gambar 2.2 javax.swing

`javax.swing.GroupLayout jPanel1Layout`: Ini adalah deklarasi variabel `jPanel1Layout` yang bertipe `javax.swing.GroupLayout`. Variabel ini akan digunakan untuk menyimpan objek `GroupLayout` yang akan mengatur tata letak komponen dalam `jPanel1`.

`new javax.swing.GroupLayout(jPanel1)`: Di sini, Anda membuat sebuah objek `GroupLayout` baru dengan parameter `jPanel1`. Ini mengaitkan `GroupLayout` dengan panel (`jPanel1`) yang akan mengatur tata letaknya. `jPanel1` adalah panel GUI yang mungkin sudah Anda buat sebelumnya sebagai wadah untuk komponen-komponen GUI.

`javax.swing.GroupLayout.Alignment.LEADING`: Ini adalah argumen yang menentukan bagaimana komponen-komponen dalam grup paralel ini akan diatur. `LEADING` mengacu pada tata letak komponen yang berada di bagian kiri (atau paling atas dalam tata letak vertikal) dari grup paralel ini.

3. Void Pemenang

```
18
19 void pemenang() {
20     if ("X".equals(btn1.getText()) && "X".equals(btn2.getText()) && "X".equals(btn3.getText())) {
21         JOptionPane.showMessageDialog(rootPane, message:"Pemenangnya adalah X, Selamat X !!!");
22     } else if ("X".equals(btn4.getText()) && "X".equals(btn5.getText()) && "X".equals(btn6.getText())) {
23         JOptionPane.showMessageDialog(rootPane, message:"Pemenangnya adalah X, Selamat X !!!");
24     } else if ("X".equals(btn7.getText()) && "X".equals(btn8.getText()) && "X".equals(btn9.getText())) {
25         JOptionPane.showMessageDialog(rootPane, message:"Pemenangnya adalah X, Selamat X !!!");
26     } else if ("X".equals(btn1.getText()) && "X".equals(btn5.getText()) && "X".equals(btn9.getText())) {
27         JOptionPane.showMessageDialog(rootPane, message:"Pemenangnya adalah X, Selamat X !!!");
28     } else if ("X".equals(btn3.getText()) && "X".equals(btn5.getText()) && "X".equals(btn7.getText())) {
29         JOptionPane.showMessageDialog(rootPane, message:"Pemenangnya adalah X, Selamat X !!!");
30     } else if ("X".equals(btn1.getText()) && "X".equals(btn4.getText()) && "X".equals(btn7.getText())) {
31         JOptionPane.showMessageDialog(rootPane, message:"Pemenangnya adalah X, Selamat X !!!");
32     } else if ("X".equals(btn2.getText()) && "X".equals(btn5.getText()) && "X".equals(btn8.getText())) {
33         JOptionPane.showMessageDialog(rootPane, message:"Pemenangnya adalah X, Selamat X !!!");
34     } else if ("X".equals(btn3.getText()) && "X".equals(btn6.getText()) && "X".equals(btn9.getText())) {
35         JOptionPane.showMessageDialog(rootPane, message:"Pemenangnya adalah X, Selamat X !!!");
36     }
37 }
```

Gambar 2.3 void pemenang

Void Pemenang pada Kode ini memeriksa kombinasi tombol ("X") pada papan permainan dan menampilkan pesan dialog jika ada pemenang. Anda dapat mengganti pesan "Pemenangnya adalah X, Selamat X !!!" sesuai dengan keinginan Anda dalam terjemahan tersebut.

4. Kode Kombinasi O

```
37
38 xt() && "O".equals(btn2.getText()) && "O".equals(btn3.getText())) {
39     geDialog(rootPane, message:"Pemenangnya adalah O, Selamat O !!!");
40     4.getText() && "O".equals(btn5.getText()) && "O".equals(btn6.getText()) {
41     geDialog(rootPane, message:"Pemenangnya adalah O, Selamat O !!!");
42     7.getText() && "O".equals(btn8.getText()) && "O".equals(btn9.getText()) {
43     geDialog(rootPane, message:"Pemenangnya adalah O, Selamat O !!!");
44     1.getText() && "O".equals(btn5.getText()) && "O".equals(btn9.getText()) {
45     geDialog(rootPane, message:"Pemenangnya adalah O, Selamat O !!!");
46     3.getText() && "O".equals(btn5.getText()) && "O".equals(btn7.getText()) {
47     geDialog(rootPane, message:"Pemenangnya adalah O, Selamat O !!!");
48     1.getText() && "O".equals(btn4.getText()) && "O".equals(btn7.getText()) {
49     geDialog(rootPane, message:"Pemenangnya adalah O, Selamat O !!!");
50     2.getText() && "O".equals(btn5.getText()) && "O".equals(btn8.getText()) {
51     geDialog(rootPane, message:"Pemenangnya adalah O, Selamat O !!!");
52     3.getText() && "O".equals(btn6.getText()) && "O".equals(btn9.getText()) {
53     geDialog(rootPane, message:"Pemenangnya adalah O, Selamat O !!!");
54
```

Gambar 2.4 code kombinasi O

Kode ini juga memeriksa kombinasi tombol ("O") pada papan permainan dan menampilkan pesan dialog jika ada pemenang pemain "O". kita dapat mengganti pesan "Pemenangnya adalah O, Selamat O !!!" sesuai dengan keinginan kita dalam terjemahan tersebut.

5. Void Draw

```
void draw () {
    if ("O".equals(btn2.getText()) && "X".equals(btn1.getText()) &&
    "X".equals(btn3.getText()) && "O".equals(btn4.getText()) &&
    "O".equals(btn5.getText()) && "X".equals(btn6.getText()) &&
    "X".equals(btn7.getText()) && btn8.getText() == "X" && btn9.getText() == "O"
    ) {
    } else if ("O".equals(btn1.getText()) && "X".equals(btn2.getText()) &&
    "O".equals(btn3.getText()) && "X".equals(btn4.getText())
    && btn5.getText() == "X" && btn6.getText() == "O" &&
    btn7.getText() == "X" && btn8.getText() == "O" && btn9.getText() == "X"
    ) {
        JOptionPane.showMessageDialog(rootPane, "Game Draw !!!");
    }
}
```

Gambar 2.5 void draw

Fungsi draw() digunakan untuk memeriksa kondisi tertentu dalam permainan, dan mungkin digunakan untuk menentukan apakah permainan berakhir dengan seri (draw). Ada dua blok if-else dalam fungsi ini:

- a. Blok pertama (if) adalah untuk memeriksa kondisi tertentu di papan permainan, tetapi tidak melakukan tindakan apa pun jika kondisinya terpenuhi.

- b. Blok kedua (else if) digunakan untuk memeriksa kondisi lain di papan permainan, dan jika kondisi ini terpenuhi, maka akan menampilkan pesan dialog "Permainan Seri!!!" menggunakan `JOptionPane.showMessageDialog()`. Kode ini terutama fokus pada kasus di mana permainan berakhir dengan hasil seri (draw) dan menampilkan pesan yang sesuai jika kondisi tersebut terpenuhi.

6. Void Reset

```
cTacToe.java > TicTacToe > draw()

void reset() {
    nilai1 = 1;
    nilai1X = 0;
    nilai1O = 0;

    lblX.setText(text:"0");
    lblO.setText(text:"0");

    btn1.setText(text:"");
    btn1.setEnabled(b:true);
    btn2.setText(text:"");
    btn2.setEnabled(b:true);
    btn3.setText(text:"");
    btn3.setEnabled(b:true);
    btn4.setText(text:"");
    btn4.setEnabled(b:true);
    btn5.setText(text:"");
    btn5.setEnabled(b:true);
    btn6.setText(text:"");
    btn6.setEnabled(b:true);
    btn7.setText(text:"");
    btn7.setEnabled(b:true);
    btn8.setText(text:"");
    btn8.setEnabled(b:true);
    btn9.setText(text:"");
    btn9.setEnabled(b:true);
}
```

Gambar 2.6 void reset

metode `reset()` digunakan untuk mengatur ulang permainan atau aplikasi, termasuk mengatur ulang giliran pemain, nilai kemenangan, dan teks yang ditampilkan pada tombol-tombol permainan. Ini mempersiapkan permainan untuk dimulai kembali dengan kondisi awal y `lblO` dan `lblX` adalah variabel yang digunakan untuk merujuk ke objek `JLabel`. Dalam konteks ini, objek `JLabel` digunakan untuk menampilkan teks atau label yang akan ditampilkan dalam antarmuka pengguna.

`new javax.swing.JLabel()` adalah perintah untuk membuat objek baru dari kelas `JLabel` yang termasuk dalam paket (package) `javax.swing`. Kelas `JLabel` adalah komponen antarmuka pengguna yang digunakan untuk menampilkan teks. Dengan menggunakan `new`, Anda menciptakan objek baru dari kelas tersebut.

Setelah objek `JLabel` dibuat, Anda dapat menggunakannya untuk mengatur dan menampilkan teks dalam elemen GUI Anda. Biasanya, Anda akan mengatur teks yang akan ditampilkan pada objek `JLabel` dengan menggunakan metode seperti `setText()`.

7. Source Code

```

142     javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
143     jPanel2.setLayout(jPanel2Layout);
144     jPanel2Layout.setHorizontalGroup(
145         jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
146             .addGroup(jPanel2Layout.createSequentialGroup()
147                 .addGap(10, 10, 10)
148                 .addGap(min:0, pref:100, Short.MAX_VALUE)
149             );
150     jPanel2Layout.setVerticalGroup(
151         jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
152             .addGroup(jPanel2Layout.createSequentialGroup()
153                 .addGap(min:0, pref:100, Short.MAX_VALUE)
154             );
155     setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
156     jPanel1.setBackground(new java.awt.Color(204, 255, 255));
157     jPanel1.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.LOWERED));
158
159     btn3.setBackground(new java.awt.Color(0, 102, 51));
160     btn3.setFont(new java.awt.Font("Javanese Text", style:1, size:24)); // NOI18N
161     btn3.setForeground(new java.awt.Color(0, 102, 102));
162     btn3.addActionListener(new java.awt.event.ActionListener() {
163         public void actionPerformed(java.awt.event.ActionEvent evt) {
164             btn3ActionPerformed(evt);
165         }
166     });

```

Gambar 2.7 source code

- a. `javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);`

Ini adalah deklarasi dan inisialisasi objek `jPanel2Layout` dari kelas `GroupLayout`. `GroupLayout` digunakan untuk mengatur tata letak komponen dalam kontainer, dalam hal ini, `jPanel2`. `jPanel2` adalah komponen antarmuka pengguna yang akan memiliki tata letak yang diatur.

- b. `jPanel2.setLayout(jPanel2Layout);`

Baris ini mengatur tata letak komponen `jPanel2` dengan objek `jPanel2Layout` yang telah dibuat. Dengan kata lain, objek `jPanel2Layout` akan mengendalikan tata letak komponen di dalam `jPanel2`.

- c. `setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);`

Ini adalah pengaturan perilaku jendela. Dalam hal ini, itu mengatur perilaku jendela saat tombol penutup jendela diklik. `EXIT_ON_CLOSE` mengarahkan aplikasi untuk berhenti ketika jendela ditutup.

- d. `jPanel1.setBackground(new java.awt.Color(204, 255, 255));`

Ini mengatur warna latar belakang untuk `jPanel1`, yang merupakan komponen GUI. Warna yang digunakan dalam kode ini adalah biru muda (RGB 204, 255, 255).

- e. `jPanel1.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.LOWERED));`

Baris ini mengatur tepi (border) komponen `jPanel1` dengan pengaturan border yang telah dibuat. Border ini dibuat menggunakan `createBevelBorder`, dan di sini, jenis border yang digunakan adalah `BevelBorder.LOWERED`, yang memberikan efek penurunan pada tepi komponen.

8. Reset Button

```

220
221     resetbtn.setBackground(new java.awt.Color(r:0, g:102, b:0));
222     resetbtn.setFont(new java.awt.Font(name:"Bahnschrift", style:1, size:24)); // NOI18N
223     resetbtn.setForeground(new java.awt.Color(r:255, g:255, b:255));
224     resetbtn.setText(text:"RESET");
225     resetbtn.setHorizontalTextPosition(javax.swing.SwingConstants.RIGHT);
226     resetbtn.addActionListener(new java.awt.event.ActionListener() {
227         public void actionPerformed(java.awt.event.ActionEvent evt) {
228             resetbtnActionPerformed(evt);
229         }
230     });
231

```

Gambar 2.8 resetbtn

- Baris ini mengatur latar belakang tombol (resetbtn) dengan warna tertentu. Dalam hal ini, warna latar belakang diatur sebagai warna hijau gelap dengan nilai RGB (0, 102, 0). Ini mengatur font (jenis huruf) tombol. Font yang digunakan adalah "Bahnschrift" dengan gaya huruf tebal (1) dan ukuran huruf 24.
- Baris ini mengatur warna teks pada tombol, yang diatur sebagai warna putih (RGB 255, 255, 255). Teks tombol akan ditampilkan dengan warna ini. Ini mengatur teks yang akan ditampilkan di tombol, yaitu "RESET".
- Baris ini mengatur posisi teks dalam tombol. Dalam hal ini, teks akan ditempatkan sejajar dengan sebelah kanan tombol. Ini menambahkan seorang pendengar (listener) ke tombol (resetbtn). Ketika tombol ini ditekan (di-click), metode resetbtnActionPerformed(evt) akan dipanggil. Metode ini akan menangani tindakan yang harus dilakukan ketika tombol "RESET" ditekan.

9. Javax.swing.grouplayout

```

352 );
353
354 javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
355 getContentPane().setLayout(layout);
356 layout.setHorizontalGroup(
357     layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
358     .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE, pref:761, javax.swing.GroupLayout.PREFERRED_SIZE)
359 );
360 layout.setVerticalGroup(
361     layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
362     .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
363 );
364

```

Gambar 2.9 javax.swing.grouplayout

- `javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());`
Ini adalah deklarasi dan inisialisasi objek layout dari kelas GroupLayout. Objek ini akan digunakan untuk mengatur tata letak komponen dalam kontainer utama (`getContentPane()`). `getContentPane()` mengembalikan panel akar yang digunakan sebagai kontainer utama di dalam jendela utama.

b. `getContentPane().setLayout(layout);`

Baris ini mengatur tata letak komponen kontainer utama (biasanya jendela). Dengan kata lain, objek layout akan mengendalikan tata letak komponen dalam kontainer utama.

c. `layout.setHorizontalGroup(...)`

Ini adalah pengaturan tata letak horizontal. Ini mengatur bagaimana komponen akan ditata secara horizontal dalam kontainer. Dalam hal ini, `JPanel1` adalah komponen yang akan ditambahkan ke kontainer utama dengan lebar yang diinginkan (761 piksel).

d. `layout.setVerticalGroup(...)`

Ini adalah pengaturan tata letak vertikal. Ini mengatur bagaimana komponen akan ditata secara vertikal dalam kontainer. Dalam hal ini, `JPanel1` adalah komponen yang akan ditambahkan ke kontainer utama dengan tinggi yang diatur agar mengisi ruang secara vertikal dengan ukuran yang sesuai.

10. Privat Void

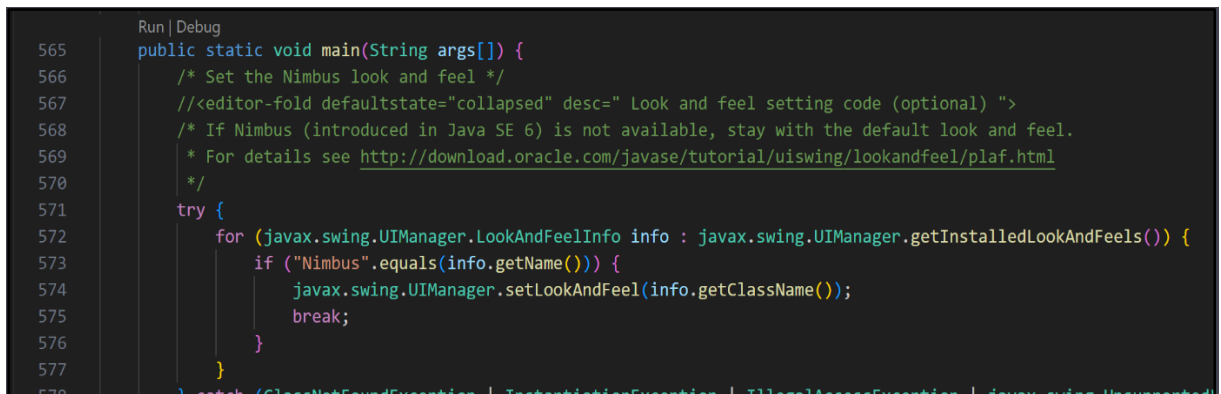
```
477
478     private void btn1ActionPerformed(java.awt.event.ActionEvent evt) {
479         // TODO add your handling code here:
480         if (nilai == 1) {
481             btn1.setText(text:"X");
482             btn1.setEnabled(b:false);
483             nilai -= 1;
484             nilaiX += 1;
485             lblX.setText(String.valueOf(nilaiX + " klik"));
486             pemenang();
487             draw ();
488         } else {
489             btn1.setText(text:"O");
490             btn1.setEnabled(b:false);
491             nilai += 1;
492             nilaiO += 1;
493             lblO.setText(String.valueOf(nilaiO + " klik"));
494             pemenang();
495             draw ();
496         }
497     }
```

Gambar 2.10 private void

- a. Ini adalah deklarasi dari metode `btn1ActionPerformed` yang disebut sebagai penanganan acara. Metode ini akan dieksekusi ketika tombol `btn1` ditekan. Komentar `TODO` menandakan bahwa di sini Anda biasanya akan menambahkan logika penanganan acara sesuai dengan kebutuhan aplikasi Anda.
- b. Ini adalah logika utama penanganan acara yang terjadi ketika tombol `btn1` ditekan. Pertama, kondisi `if (nilai == 1)` memeriksa nilai dari variabel `nilai`. Jika nilainya adalah 1, maka akan dilakukan tindakan yang sesuai.
 1. Jika kondisi terpenuhi, maka teks pada tombol `btn1` diubah menjadi "X" (kemungkinan ini adalah giliran pemain X), tombol dinonaktifkan (tidak dapat

- diklik lagi), nilai nilai dikurangi 1, dan jumlah klik pemain X (nilaiX) ditambah 1. Label lblX juga diperbarui untuk menampilkan jumlah klik pemain X.
2. Selanjutnya, dua metode dipanggil: pemenang() dan draw(). pemenang() kemungkinan akan memeriksa apakah pemain X telah memenangkan permainan, dan draw() mungkin memeriksa apakah permainan berakhir dengan seri (draw).
 3. Jika kondisi if tidak terpenuhi (yaitu nilai bukan 1), maka akan dieksekusi bagian else.
 4. Dalam bagian else, teks pada tombol btn1 diubah menjadi "O" (kemungkinan ini adalah giliran pemain O), tombol dinonaktifkan, nilai nilai ditambah 1, dan jumlah klik pemain O (nilaiO) ditambah 1. Label lblO juga diperbarui untuk menampilkan jumlah klik pemain O.
 5. Kembali, dua metode pemenang() dan draw() dipanggil untuk memeriksa status permainan.

11. Public Static



```
Run | Debug
565 public static void main(String args[]) {
566     /* Set the Nimbus look and feel */
567     //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
568     /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
569      * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
570      */
571     try {
572         for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
573             if ("Nimbus".equals(info.getName())) {
574                 javax.swing.UIManager.setLookAndFeel(info.getClassName());
575                 break;
576             }
577         }
578     } catch (ClassNotFoundException | InstantiationException | IllegalAccessException | javax.swing.UnsupportedLookAndFeelException ex) {
579         // Do nothing
580     }
581 }
```

Gambar 2.11 public static

- a. Ini adalah deklarasi metode main(), yang merupakan metode utama yang akan dieksekusi saat aplikasi Java dijalankan. Metode main() adalah pintu masuk utama ke program
- b. Ini adalah komentar dalam kode yang memberikan penjelasan bahwa kode berikutnya akan mengatur tampilan antarmuka pengguna dengan menggunakan "look and feel" Nimbus. "Look and feel" mengacu pada penampilan dan gaya visual aplikasi.
- c. Ini adalah komentar yang memberi petunjuk kepada lingkungan pengembangan (IDE) untuk menghasilkan kode dengan "fold" atau pengelompokan opsional. Ini memungkinkan bagian kode ini dilipat atau di-expand secara opsional dalam editor.
- d. Ini adalah komentar yang memberikan penjelasan tentang apa yang akan terjadi dalam kode berikutnya. Kode ini mencoba mengatur "look and feel" Nimbus jika tersedia, dan

jika tidak, akan tetap menggunakan "look and feel" default yang tersedia di Java SE 6 atau versi yang lebih baru.

e. Ini adalah blok try-catch yang mencoba mengatur "look and feel" Nimbus. Ini melakukan langkah-langkah berikut:

1. Iterasi melalui daftar "look and feel" yang diinstal di lingkungan aplikasi.
2. Memeriksa apakah ada "look and feel" dengan nama "Nimbus".
3. Jika ada, maka mengatur "look and feel" aplikasi ke "Nimbus" dengan mengganti UIManager dengan kelas "look and feel" yang sesuai.
4. Break dari iterasi jika "look and feel" Nimbus ditemukan.
5. Kode ini akan menangkap pengecualian jika "look and feel" Nimbus tidak ditemukan atau gagal diatur.

12. Variables Declaration

```
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
    /* Create and display the form */
    java.awt.EventQueue.invokeLater(() -> {
        new TicTacToe().setVisible(true);
    });
}

// Variables declaration - do not modify
private javax.swing.JButton btn1;
private javax.swing.JButton btn2;
private javax.swing.JButton btn3;
private javax.swing.JButton btn4;
private javax.swing.JButton btn5;
private javax.swing.JButton btn6;
private javax.swing.JButton btn7;
private javax.swing.JButton btn8;
private javax.swing.JButton btn9;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
private javax.swing.JLabel lbl0;
private javax.swing.JLabel lblX;
private javax.swing.JButton resetbtn;
// End of variables declaration
}
```

Gambar 2.12 variables declaration

- a. Kode ini adalah deklarasi variabel-variabel yang akan digunakan dalam aplikasi Anda. Setiap variabel memiliki tipe data javax.swing yang sesuai dengan jenis komponen atau elemen antarmuka pengguna yang direferensikan.
- b. Variabel-variabel ini mencakup tombol (JButton), label (JLabel), dan panel (JPanel) yang akan digunakan dalam aplikasi. Misalnya, btn1 hingga btn9 adalah variabel yang digunakan untuk merujuk ke sembilan tombol dalam antarmuka pengguna.
- c. Penamaan variabel ini mengikuti konvensi yang biasanya digunakan dalam Java Swing, di mana jenis komponen diberi nama dengan awalan sesuai dengan jenisnya (contoh: btn untuk tombol, lbl untuk label, dan jPanel untuk panel).
- d. Komentar // Variables declaration - do not modify memberi petunjuk bahwa bagian ini adalah hasil dari pembuatan otomatis oleh alat pengembangan (seperti IDE) dan tidak seharusnya diubah secara manual oleh pengembang.

- e. Bagian End of variables declaration menandai akhir dari blok deklarasi variabel. Variabel-variabel ini nantinya akan digunakan untuk mengakses dan mengendalikan komponen-komponen dalam antarmuka pengguna Anda.

HASIL GAME TIC TAC TOE

1. Opening



Gambar 3.1 opening game tic tac toe

Gambar 3.1 yang merupakan opening dari game tic tac toe, game tersebut merupakan game yang tadi kita buat menggunakan code coding pada tahapan no 2 tepatnya pada no 1

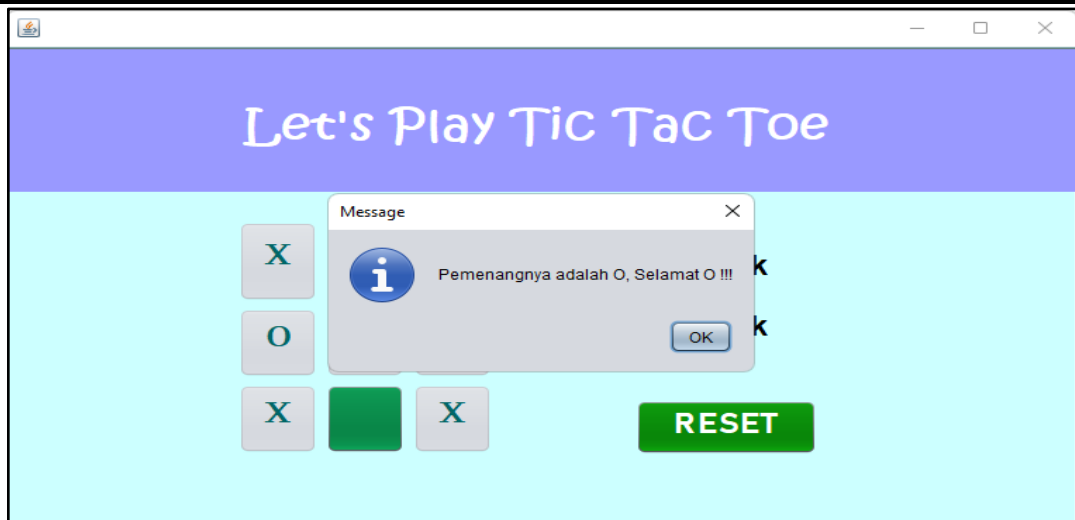
2. Pelacak Pembagian Giliran Pemain



Gambar 3.2 pelacak pembagian giliran permainan

Pada gambar 3.2 yaitu pelacak pembagian giliran permainan pada game tictac toe, pada bagian ini merupakan hasil dari code coding pada gambar satu, bagian b yaitu tentang pemain 1 dan pemain 2 yang di simbolkan dengan X dan O, serta jumlah klik yang dapat dilakukan oleh dua orang pemain nantinya.

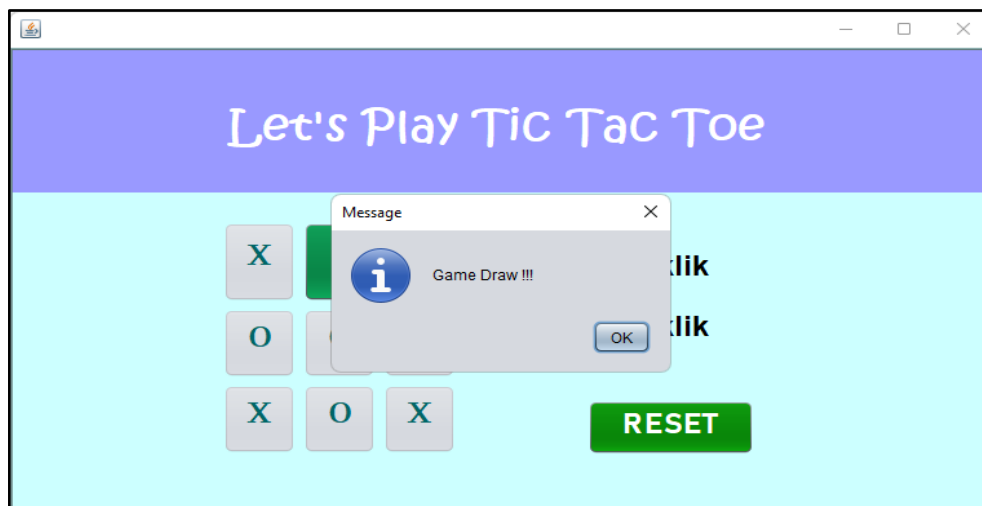
3. Hasil Void Reset



Gambar 3.3 hasil dari void reset

Pada gambar 3.3 ,yaitu merupakan hasil dari private void, pada bagian no 2 tepatnya pada gambar 2.11 , setelah pada tahap sebelumnya setiap Langkah dalam permainan tic tac toe di lihat atau di atur dalam bagian pelacak, lalu sekarang, pada bagian ini kita dapat memeriksa apakah pada permainan ini ada pemenang atau tidak. Jika pada permainan tersebut di temukannya pemenang maka ,pesan yang akan timbul pada permainan tersebut akan sama seperti pada di atas.

4. Game Draw



Gambar 3.4 game draw

Gambar 3.4 yaitu pada gambar draw ,bagian game draw ini merupakan hasil dari void draw pada gambar 5 pada bagian di atas, pada bagian draw ini tidak semua akan muncul pada saat kita bermain game tic tac toe ini, kondisi seperti ini hanya akan muncul jika seluruh papan permainan tersebut terisi, namun tidak di temukannya pemenang dari permainan gamesnya. Hal ini adalah salah satu bagian penting dari logika dari permainan game tic tac toe agar dapat memastikan bahwa permainan

berakhir dengan benar ketika tidak ada pemenang yang muncul pada game tersebut, dan tidak ada langkah lain yang dapat dilakukan.

5. Closing



Gambar 3. 5 closing tic tac toe

Pada gambar 3.5 merupakan closing dari game tic tac toe, pada tahap ini merupakan hasil dari gambar 13 yang merupakan gambar atau hasil akhir pada permainan tic tac toe ini, kode reset ini nantinya sangat penting dalam permainan tic tac toe karena hal tersebut dapat memungkinkan pemain untuk dapat memulai lagi permainan yang baru dengan sangat mudah setelah setelah melakukan permainan tersebut sebelumnya sudah selesai dimainkan atau di gunakan. Hal itu secara tidak langsung juga dapat memastikan bahwa semua variabel dan elemen dalam tampilan tersebut di kelola dengan sangat baik dan pastinya benar, sehingga mendapatkan awal permainan yang bersih.

KESIMPULAN DAN SARAN

A. Kesimpulan

Kesimpulan dari coding (pemrograman) dan pembuatan permainan Tic-Tac-Toe adalah sebagai berikut:

1. Coding adalah Alat untuk Mewujudkan Ide: Coding adalah proses menerjemahkan konsep dan ide ke dalam bahasa yang bisa dimengerti oleh komputer. Dalam konteks pembuatan permainan Tic-Tac-Toe, coding memungkinkan kita untuk menciptakan permainan sesuai dengan aturan dan logika yang kita inginkan.
2. Pemahaman Konsep Dasar Pemrograman Penting: Pembuatan permainan Tic-Tac-Toe memerlukan pemahaman konsep dasar pemrograman, seperti variabel, loop, percabangan, fungsi, dan struktur data. Ini adalah dasar yang diperlukan untuk mengembangkan permainan dan membuatnya berfungsi.
3. Pemahaman Logika Permainan: Penting untuk memahami logika dan aturan permainan secara menyeluruh sebelum mulai coding. Dalam hal ini, kita perlu memahami aturan Tic-Tac-Toe, termasuk cara menentukan pemenang dan mengidentifikasiimbang.
4. Kreativitas dalam Pemrograman: Coding juga merupakan wadah untuk kreativitas. Anda memiliki fleksibilitas untuk menambahkan elemen-elemen kreatif ke dalam permainan, seperti grafik, animasi, atau fitur tambahan, untuk membuat permainan lebih menarik.
5. Pengujian dan Debugging: Bagian penting dari coding adalah pengujian dan debugging. Penting untuk menguji permainan secara menyeluruh untuk memastikan bahwa itu berfungsi dengan baik dan mengatasi masalah yang mungkin muncul selama pengembangan.
6. Bahasa Pemrograman dan Alat yang Tepat: Pemilihan bahasa pemrograman dan alat yang sesuai sangat penting. Ini akan memengaruhi bagaimana Anda mengimplementasikan permainan dan berinteraksi dengan pengguna.
7. Pengembangan Permainan adalah Proses Pembelajaran: Mengembangkan permainan Tic-Tac-Toe adalah peluang untuk belajar pemrograman dan mengasah keterampilan dalam memahami konsep-konsep dasar pemrograman.
8. Pada intinya, coding dan pembuatan permainan Tic-Tac-Toe adalah contoh praktis dari bagaimana pemrograman dapat digunakan untuk menciptakan perangkat lunak yang memiliki tujuan tertentu, dalam hal ini, permainan yang bisa

dimainkan oleh pengguna. Ini juga menggambarkan cara pemrograman melibatkan pemahaman konsep dan kreativitas untuk menciptakan sesuatu yang bermanfaat atau hiburan bagi orang lain.

B. Saran

Saran dari pembuatan tugas Tengah semester kami,yaitu tentang Bahasa pemograman (coding) dan game tic tac toe adalah,sebelum kita dapat membangun game tic tia toe sebaiknya kita memahami dahulu tentang Bahasa pemograman,terutama jav,karena dari Bahasa pemograman java ini lah kita dapat membuat code code yang nanti nya akan kita gunakan di setiap tahap dalam permainan tic tac toc tersebut dari awal sampe pada akhir,semuannya menggunakan unsur – unsur ,yang kita buat pada setiap langkah maupun tahapan yang sesuai ,pada coding yang kita buatr pada bagian sebelumnya.

DAFTAR PUSTAKA

A.Chanhar,K Sunanto Husin,and R. Angreni, “Penerapan Algoritma MTD (f) pada permainan TIC TAC TOE dengan variasi area”

Donald H. Sanders, “pengertian computer”, Google 2011

M.A Dalffa, B . S . Abu Nasser , and S . S . Abu Naser, “ Tic Tac Toe Learning Using Artifical

Neural Networks,” int J. Eng. Inf . Syst ., vol. 3, no 2 , 2019.

Nugroho, Andi. Analisis Dan Perancangan system dengan metodologi Berorientasi Objek. Bandung: Informatika Bandung, 2002