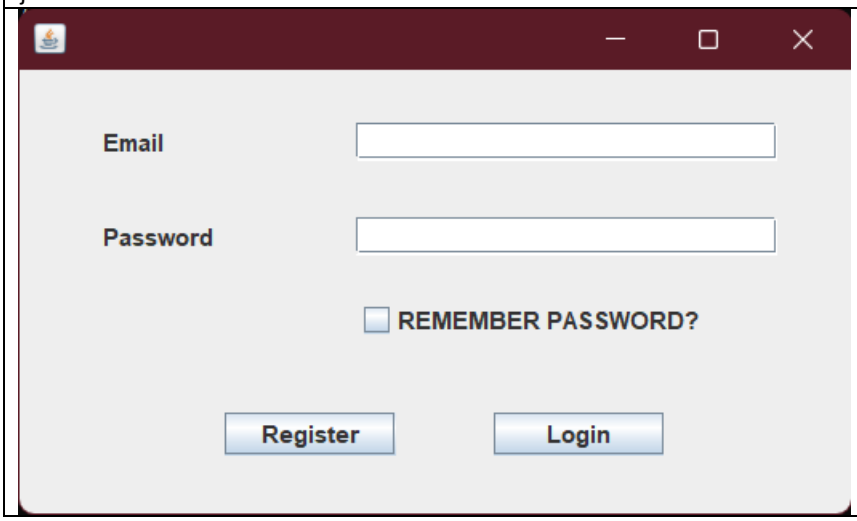
Login Source code:

```
package com.ibik.pbo.praktikum;

import java.awt.EventQueue;
import java.awt.event.*;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JCheckBox;
import javax.swing.JTextField;
import javax.swing.JButton;
import javax.swing.JPasswordField;

@SuppressWarnings("serial")
public class Login extends JFrame{

	private JPanel contentPane;
	private final JCheckBox chckbxNewCheckBox = new JCheckBox("REMEMBER
PASSWORD?");
	private JTextField textemail;
	private JPasswordField textpass;

	/**
	 * Launch the application.
	 */
	public static void main(String[] args) {
		EventQueue.invokeLater(new Runnable() {
			public void run() {
				try {
					Login frame = new Login();
					frame.setVisible(true);
				} catch (Exception e) {
					e.printStackTrace();
				}
			}
		});
	}

	/**
	 * Create the frame.
	 */
	public Login() {
		setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		setBounds(100, 100, 430, 259);
		contentPane = new JPanel();
		contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

		setContentPane(contentPane);
```

```
		contentPane.setLayout(null);

		JLabel lblNewLabel = new JLabel("Email");
		lblNewLabel.setBounds(42, 30, 45, 13);
		contentPane.add(lblNewLabel);

		JLabel lblNewLabel_1 = new JLabel("Password");
		lblNewLabel_1.setBounds(42, 77, 64, 13);
		contentPane.add(lblNewLabel_1);

		textemail = new JTextField();
		textemail.setBounds(168, 27, 211, 19);
		contentPane.add(textemail);
		textemail.setColumns(10);
		chckbxNewCheckBox.setBounds(168, 107, 188, 36);
		contentPane.add(chckbxNewCheckBox);

		final JButton register = new JButton("Register");
		register.setBounds(103, 172, 85, 21);
		contentPane.add(register);


		final JButton login = new JButton("Login");
		login.setBounds(237, 172, 85, 21);
		contentPane.add(login);

		textpass = new JPasswordField();
		textpass.setBounds(168, 74, 211, 19);
		contentPane.add(textpass);

		register.addActionListener(new ActionListener() {

			@Override
			public void actionPerformed(ActionEvent e) {
				Register.main(null);
			}

		});

		login.addActionListener(new ActionListener() {

			@Override
			public void actionPerformed(ActionEvent e) {
				if(textemail.getText().isEmpty() || new String
(textpass.getPassword()).isEmpty()) {
					JOptionPane.showMessageDialog(null, "Masukkan Email
dan Password terlebih dahulu",null, JOptionPane.WARNING_MESSAGE);
				}
				else {
					JOptionPane.showMessageDialog(null, "Welcome USER",
null, JOptionPane.INFORMATION_MESSAGE);
```

```
                                        Latihan05.main(null);
                                    }
                                }
                            });

                }


}
```



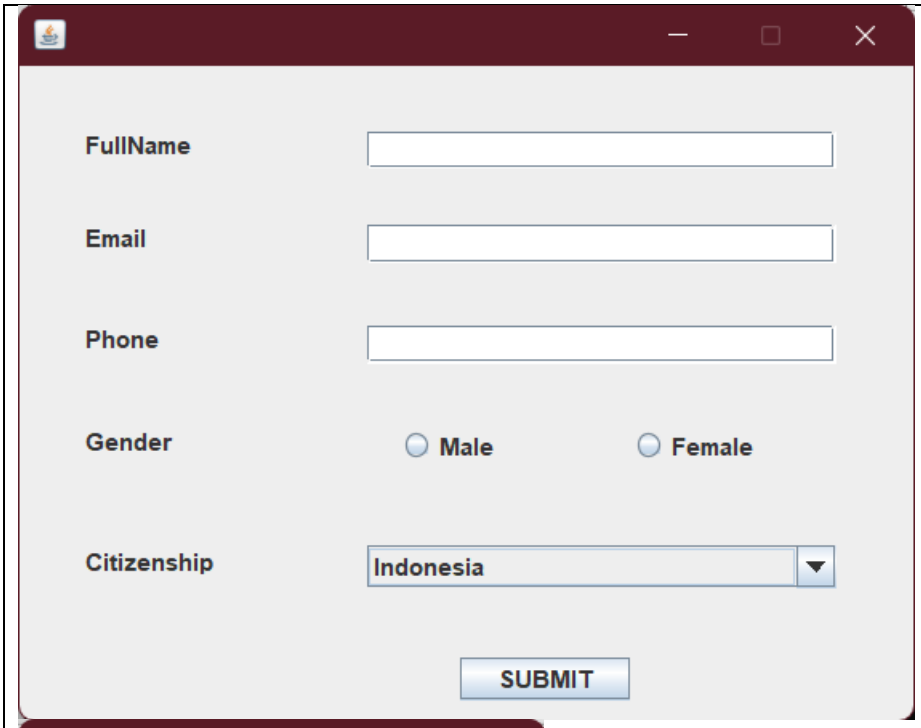
Register Source code:

```
package com.ibik.pbo.praktikum;

import java.awt.EventQueue;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JButton;
import javax.swing.JTextField;
import javax.swing.JRadioButton;
import javax.swing.JComboBox;
import javax.swing.ButtonGroup;
import javax.swing.DefaultComboBoxModel;

public class Register extends JFrame {

        private JPanel contentPane;
        private JTextField textfull;
        private JTextField textemail;
        private JTextField textphone;
```

```
	/**
	 * Launch the application.
	 */
	public static void main(String[] args) {
		EventQueue.invokeLater(new Runnable() {
			public void run() {
				try {
					Register frame = new Register();
					frame.setVisible(true);
				} catch (Exception e) {
					e.printStackTrace();
				}
			}
		});
		new Register();
	}

	/**
	 * Create the frame.
	 */
	public Register() {
		setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		setBounds(100, 100, 461, 364);
		contentPane = new JPanel();
		contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

		setContentPane(contentPane);
		contentPane.setLayout(null);

		JLabel lblNewLabel = new JLabel("FullName");
		lblNewLabel.setBounds(33, 33, 85, 13);
		contentPane.add(lblNewLabel);

		JLabel lblNewLabel_1 = new JLabel("Email");
		lblNewLabel_1.setBounds(33, 80, 45, 13);
		contentPane.add(lblNewLabel_1);

		JLabel lblNewLabel_2 = new JLabel("Phone");
		lblNewLabel_2.setBounds(33, 130, 45, 13);
		contentPane.add(lblNewLabel_2);

		JLabel lblNewLabel_3 = new JLabel("Gender");
		lblNewLabel_3.setBounds(33, 181, 45, 13);
		contentPane.add(lblNewLabel_3);

		JLabel lblNewLabel_4 = new JLabel("Citizenship");
		lblNewLabel_4.setBounds(33, 241, 85, 13);
		contentPane.add(lblNewLabel_4);

		final JButton submit = new JButton("SUBMIT");
		submit.setBounds(220, 296, 85, 21);
```

```
                contentPane.add(submit);

                textfull = new JTextField();
                textfull.setBounds(174, 33, 234, 19);
                contentPane.add(textfull);
                textfull.setColumns(10);

                textemail = new JTextField();
                textemail.setColumns(10);
                textemail.setBounds(174, 80, 234, 19);
                contentPane.add(textemail);

                textphone = new JTextField();
                textphone.setColumns(10);
                textphone.setBounds(174, 130, 234, 19);
                contentPane.add(textphone);

                final JRadioButton male_btn = new JRadioButton("Male");
                male_btn.setBounds(189, 180, 103, 21);
                contentPane.add(male_btn);

                final JRadioButton female_btn = new JRadioButton("Female");
                female_btn.setBounds(305, 180, 103, 21);
                contentPane.add(female_btn);

                final ButtonGroup group = new ButtonGroup();
                group.add(male_btn);
                group.add(female_btn);

                final JComboBox citizen = new JComboBox();
                citizen.setModel(new DefaultComboBoxModel(new String[] {"Indonesia", "Japan",
"India", "USA", "England"}));
                citizen.setBounds(174, 240, 234, 21);
                contentPane.add(citizen);

                setResizable(false);

                submit.addActionListener(new ActionListener() {

                        @Override
                        public void actionPerformed(ActionEvent e) {
                                if(textfull.getText().isEmpty() || textemail.getText().isEmpty() ||
textphone.getText().isEmpty() && !group.getSelection().isSelected()) {
                                        JOptionPane.showMessageDialog(null, "masukkan data
dengan benar",null, JOptionPane.WARNING_MESSAGE);
                                }
                                else {
                                        if(e.getSource() == submit) {
                                                String output = "Data anda : \n \n";
                                                String outputKelamin = "";
```

```
                                                    if(male_btn.isSelected()) {
                                                            outputKelamin = male_btn.getText();
                                                            output += "Fullname : " +
textfull.getText() + "\n" + "Email : " + textemail.getText() + "\n" + "Phone : " + textphone.getText()
+ "\n" + "Gender : " + outputKelamin + "\n" + "Citizenship : " + citizen.getSelectedItem();
                                                    }
                                                    else {
                                                            outputKelamin = female_btn.getText();
                                                            output += "Fullname : " +
textfull.getText() + "\n" + "Email : " + textemail.getText() + "\n" + "Phone : " + textphone.getText()
+ "\n" + "Gender : " + outputKelamin + "\n" + "Citizenship : " + citizen.getSelectedItem();

                                                    }

                                                    JOptionPane.showMessageDialog(null, output);
                                                    Login.main(null);
                                            }
                                    }
                            }

                    });

                    setDefaultCloseOperation(HIDE_ON_CLOSE);
            }
}
```



X & O Source code:

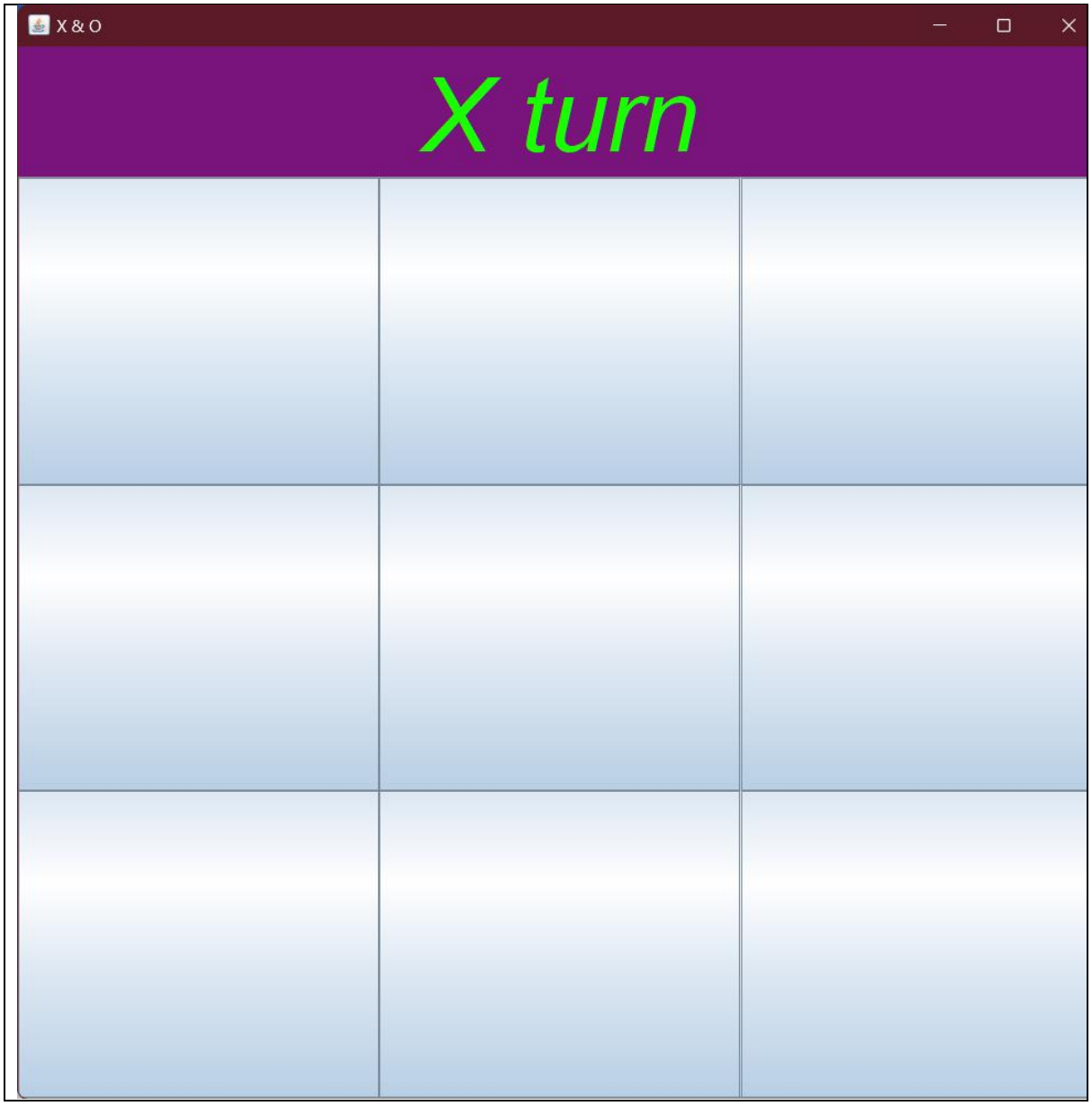
```
package com.ibik.pbo.praktikum;
import java.awt.*;
import java.awt.event.*;
import java.util.*;
import javax.swing.*;

public class XO implements ActionListener {

        JFrame frame = new JFrame();
        JPanel ypanel = new JPanel();
        JPanel xpanel = new JPanel();
        JLabel textfield = new JLabel();
        JButton[] btn = new JButton[9];
        int chance_flag = 0;
        Random random = new Random();
```

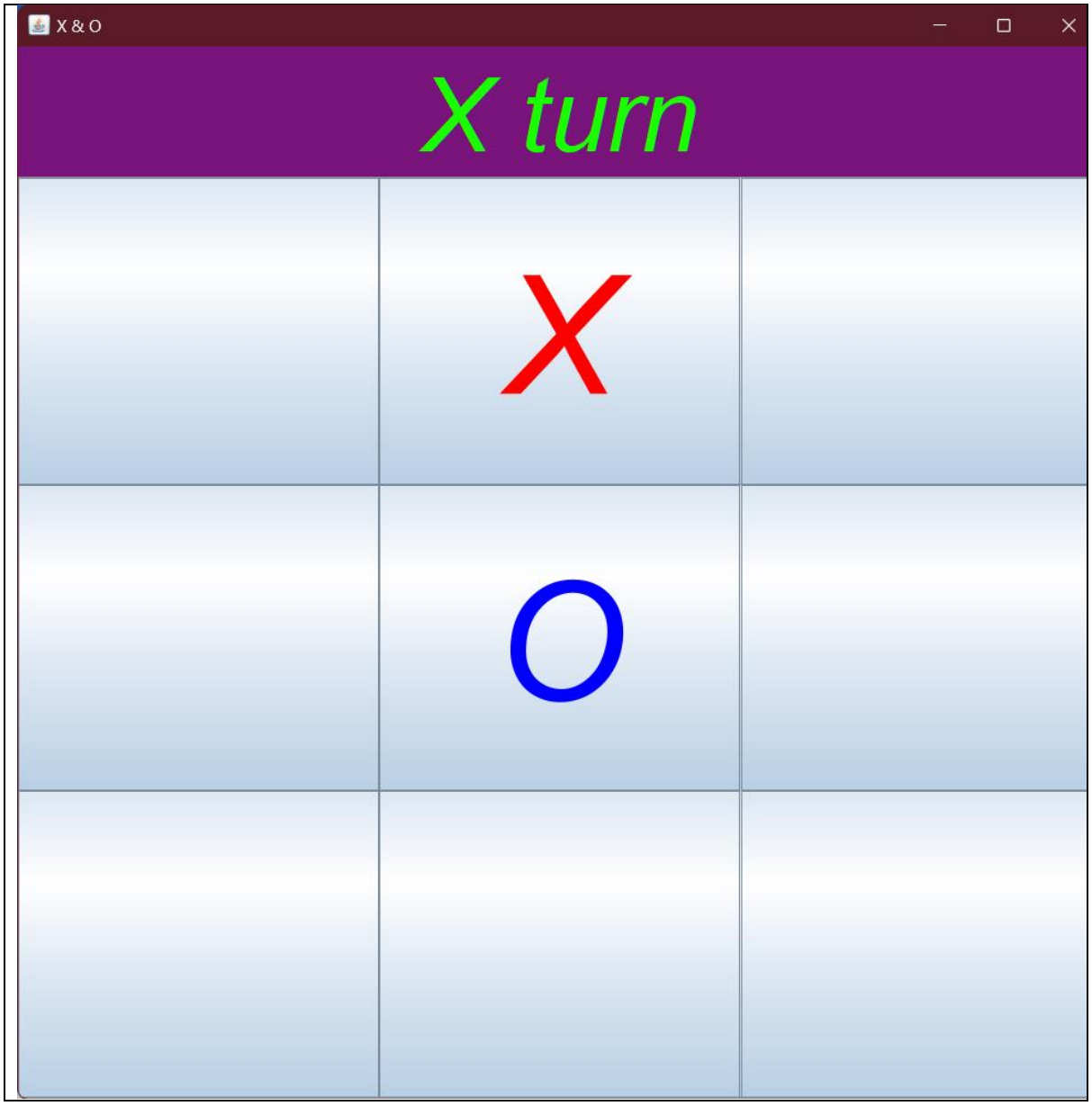
```
    boolean pl1_chance;

    XO(){

            frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            frame.setSize(800, 800);
            frame.getContentPane().setBackground(new Color(51, 53, 48));
            frame.setTitle("X & O");
            frame.setLayout(new BorderLayout());
            frame.setVisible(true);

            textfield.setBackground(new Color(120, 20, 124));
textfield.setForeground(new Color(25, 255, 0));
textfield.setFont(new Font("Start", Font.ITALIC, 75));
textfield.setHorizontalAlignment(JLabel.CENTER);
textfield.setText("X & O");
textfield.setOpaque(true);

ypanel.setLayout(new BorderLayout());
ypanel.setBounds(0, 0, 800, 800);

xpanel.setLayout(new GridLayout(3,3));
xpanel.setBackground(new Color(150, 150, 150));

for (int i = 0; i < 9; i++) {
    btn[i] = new JButton();
    xpanel.add(btn[i]);
    btn[i].setFont(new Font("Start", Font.ITALIC, 120 ));
    btn[i].setFocusable(false);
    btn[i].addActionListener(this);

}

ypanel.add(textfield);
frame.add(ypanel, BorderLayout.NORTH);
frame.add(xpanel);

startGame();
    }



    public static void main(String[] args) {
            new XO();

    }

    public void startGame() {

            try {
```

```
                        textfield.setText("Loading ...");
                        Thread.sleep(4000);
                } catch (InterruptedException e) {
                        e.printStackTrace();
                }

                int chance = random.nextInt(100);

                if (chance%2 == 0) {
                        pl1_chance = true;
                        textfield.setText("X turn");
                } else {
                        pl1_chance = false;
                        textfield.setText("O turn");
                }
        }

        public void gameOver(String s) {
                chance_flag = 0;
                Object[] option = {"Restart", "Exit"};
                int n = JOptionPane.showOptionDialog(frame, "Game Over\n"+s, "Game Over",
JOptionPane.YES_NO_CANCEL_OPTION, JOptionPane.QUESTION_MESSAGE, null, option,
option[0]);

                if (n==0) {
                        frame.dispose();
                        new XO();

                } else {
                        frame.dispose();
                }
        }

        public void matchCheck() {
    if ((btn[0].getText() == "X") && (btn[1].getText() == "X") && (btn[2].getText() == "X")) {
      xWins(0, 1, 2);
    }
    else if ((btn[0].getText() == "X") && (btn[4].getText() == "X") && (btn[8].getText() == "X")) {
      xWins(0, 4, 8);
    }
    else if ((btn[0].getText() == "X") && (btn[3].getText() == "X") && (btn[6].getText() == "X")) {
      xWins(0, 3, 6);
    }
    else if ((btn[1].getText() == "X") && (btn[4].getText() == "X") && (btn[7].getText() == "X")) {
      xWins(1, 4, 7);
    }
    else if ((btn[2].getText() == "X") && (btn[4].getText() == "X") && (btn[6].getText() == "X")) {
      xWins(2, 4, 6);
    }
    else if ((btn[2].getText() == "X") && (btn[5].getText() == "X") && (btn[8].getText() == "X")) {
      xWins(2, 5, 8);
```

```
    }
   else if ((btn[3].getText() == "X") && (btn[4].getText() == "X") && (btn[5].getText() == "X")) {
      xWins(3, 4, 5);
    }
   else if ((btn[6].getText() == "X") && (btn[7].getText() == "X") && (btn[8].getText() == "X")) {
      xWins(6, 7, 8);
    }

    else if ((btn[0].getText() == "O") && (btn[1].getText() == "O") && (btn[2].getText() == "O")) {
      oWins(0, 1, 2);
    }
    else if ((btn[0].getText() == "O") && (btn[3].getText() == "O") && (btn[6].getText() == "O")) {
      oWins(0, 3, 6);
    }
    else if ((btn[0].getText() == "O") && (btn[4].getText() == "O") && (btn[8].getText() == "O")) {
      oWins(0, 4, 8);
    }
    else if ((btn[1].getText() == "O") && (btn[4].getText() == "O") && (btn[7].getText() == "O")) {
      oWins(1, 4, 7);
    }
    else if ((btn[2].getText() == "O") && (btn[4].getText() == "O") && (btn[6].getText() == "O")) {
      oWins(2, 4, 6);
    }
    else if ((btn[2].getText() == "O") && (btn[5].getText() == "O") && (btn[8].getText() == "O")) {
      oWins(2, 5, 8);
    }
    else if ((btn[3].getText() == "O") && (btn[4].getText() == "O") && (btn[5].getText() == "O")) {
      oWins(3, 4, 5);
    } else if ((btn[6].getText() == "O") && (btn[7].getText() == "O") && (btn[8].getText() == "O")) {
      oWins(6, 7, 8);
    }
    else if(chance_flag==9) {
      textfield.setText("Match Tie");
       gameOver("Match Tie");
    }
  }

        public void xWins(int x1, int x2, int x3) {
    btn[x1].setBackground(Color.RED);
    btn[x2].setBackground(Color.RED);
    btn[x3].setBackground(Color.RED);

    for (int i = 0; i < 9; i++) {
      btn[i].setEnabled(false);
    }
    textfield.setText("X wins");
    gameOver("X Wins");
         }

        public void oWins(int x1, int x2, int x3) {
    btn[x1].setBackground(Color.RED);
```

```
        btn[x2].setBackground(Color.RED);
        btn[x3].setBackground(Color.RED);

        for (int i = 0; i < 9; i++) {
            btn[i].setEnabled(false);
        }
        textfield.setText("O Wins");
        gameOver("O Wins");
    }

            @Override
    public void actionPerformed(ActionEvent e) {

        for (int i = 0; i < 9; i++) {
            if (e.getSource() == btn[i]) {
                if (pl1_chance) {
                    if (btn[i].getText() == "") {
                        btn[i].setForeground(new Color(255, 0, 0));
                        btn[i].setText("X");
                        pl1_chance = false;
                        textfield.setText("O turn");
                        chance_flag++;
                        matchCheck();
                    }
                } else {
                    if (btn[i].getText() == "") {
                        btn[i].setForeground(new Color(0, 0, 255));
                        btn[i].setText("O");
                        pl1_chance = true;
                        textfield.setText("X turn");
                        chance_flag++;
                        matchCheck();
                    }
                }
            }
        }
    }


}
```


X & O
X turn



KeyL Source code:

```
package com.ibik.pbo.praktikum;

import java.awt.EventQueue;
import javax.swing.JFrame;

public class KeyL extends JFrame {

	public KeyL() {
		initUI();
	}


	private void initUI() {
```

```
                add(new Board());

                setTitle("Contoh KeyListener");
                setSize(400, 300);

                setLocationRelativeTo(null);
                setResizable(false);
                setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        }

        public static void main(String[] args) {

                        KeyL l = new KeyL();
                        l.setVisible(true);

        }

}
```

Board Source code:

```
package com.ibik.pbo.praktikum;

import java.awt.Color;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.Toolkit;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;

import javax.swing.JPanel;
import javax.swing.Timer;
```

```
public class Board extends JPanel implements ActionListener {

	private Timer timer;
	private Sprite sprite;
	private final int DELAY = 10;

	public Board() {
		initBoard();
	}

	private void initBoard() {

		addKeyListener(new TAdapter());
		setBackground(Color.black);
		setFocusable(true);

		sprite = new Sprite();

		timer = new Timer(DELAY, this);
		timer.start();

	}

	@Override
	public void paintComponent(Graphics g) {
		super.paintComponent(g);

		doDrawing(g);

		Toolkit.getDefaultToolkit().sync();
	}

	private void doDrawing(Graphics g) {
		Graphics2D g2d = (Graphics2D) g;

		g2d.drawImage(sprite.getImage(), sprite.getX(), sprite.getY(), this);
	}

	@Override
	public void actionPerformed(ActionEvent e) {

		step();
	}

	private void step() {
		sprite.move();

		repaint(sprite.getX()-1, sprite.getY()-1, sprite.getWidth()+2, sprite.getHeight()+2);
	}
```

```
	private class TAdapter extends KeyAdapter {

		@Override
		public void keyReleased(KeyEvent e) {
			sprite.KeyReleased(e);
		}

		@Override
		public void keyPressed(KeyEvent e) {
			sprite.KeyPressed(e);
		}
	}
}
```

Sprite Source code:

```
package com.ibik.pbo.praktikum;

import java.awt.Image;
import java.awt.event.KeyEvent;

import javax.swing.ImageIcon;

public class Sprite {

	private int dx;
	private int dy;
	private int x = 40;
	private int y = 60;
	private int w;
	private int h;
	private Image image;

	public Sprite() {

		loadImage();
	}

	private void loadImage() {
		ImageIcon ii = new ImageIcon("C:/Users/Daphne Holy/Pictures/butterfly.png");
		image = ii.getImage();

		w = image.getWidth(null);
		h = image.getHeight(null);
	}

	public void move() {

		x += dx;
		y += dy;
	}

	public int getX() {
```

```
        return x;
    }

    public int getY() {
        return y;
    }

    public int getWidth() {
        return w;
    }

    public int getHeight() {
        return h;
    }

    public Image getImage() {
        return image;
    }


    public void KeyPressed(KeyEvent e) {
        int key = e.getKeyCode();

        if(key == KeyEvent.VK_LEFT) {
            dx = -4;
        }

        if(key == KeyEvent.VK_RIGHT) {
            dx = 4;
        }

        if(key == KeyEvent.VK_UP) {
            dy = -4;
        }

        if(key == KeyEvent.VK_DOWN) {
            dy = 4;
        }
    }

    public void KeyReleased(KeyEvent e) {
        int key = e.getKeyCode();

        if(key == KeyEvent.VK_LEFT || key == KeyEvent.VK_RIGHT) {
            dx = 0;
        }

        if(key == KeyEvent.VK_UP || key == KeyEvent.VK_DOWN) {
            dy = 0;
        }
    }
```

```
}
```