CHAPTER 5
IMPLEMENTATION AND TESTING

5.1 Implementation

Create GerakKupu class

```
34  public GerakKupu(JPanel panel, int x, int y, int z) {
35    this.panel = panel;
36    this.x = x;
37    this.y = y;
38    this.z = z;
```

Illustration 5.1: The constructor of the class GerakKupu

Illustration 5.1: GerakKupu class has four parameters. The parameters are: panel, x, y, and z. Parameter panel is to set JPanel. Parameter set x dan y are the coordinates location. Parameter z is to set images.

```
42  public void GerakA() {
43    if(z == 1) {
44      if(x < 1800) {
45        x = x + 15;
46        z = 2;
47      }
48    }
49    else {
50      if(x < 1800) {
51        x = x + 15;
52        z = 1;
53      }
54    }
```

Illustration 5.2: Control the direction of movement

Illustration 5.2: GerakA method controls direction of movement.

```
181  public static int randInt(int min, int max) {
182    Random rand = new Random();
183    int randomNum = rand.nextInt((max - min) + 1) + min;
184    return randomNum;
185  }
```

Illustration 5.3: Function random

Illustration 5.3: RandInt method gets random the ranges of a numbers.
public void run() {
    int randomGerak = randint(1, 3);
    System.out.println(randomGerak);
    while(true) {
        try {
            if(randomGerak == 1) {
                GerakA();
            } else if(randomGerak == 2) {
                GerakB();
            } else if(randomGerak == 3) {
                GerakC();
            }
            Thread.sleep(460);
        } catch(InterruptedException e) {};
        panel.repaint();
    }
}

Illustration 5.4: Variable randomGerak is to set length of a number. If the number equals to 1 then it executes GerakA method. If the number equals to 2 then it executes GerakB method. If the number equals to 3 then it executes. Thread.sleep is to set delay. Repaint is to draw the graphics panel.

Create GambarKupu class

Illustration 5.5: Create a new object and invoke the parent GerakKupu class to set a parameters.
Illustration 5.6: Create a new Thread, and start the thread.

```java
GambarKupu() {
    setPreferredSize(new Dimension(1000, 500));
    Thread thread = new Thread();
    thread.start();
    kupuku.start();
}
```

Illustration 5.7: Create a component

Illustration 5.7: The paintComponent method sets components to be used for drawing 2D animation.

```java
public void paintComponent(Graphics g){
    super.paintComponent(g);
    Graphics2D g2 = (Graphics2D)g;
    drawBackground(g2, 0, 0);
    xKupu = kupuku.getX();
    yKupu = kupuku.getY();
    zKupu = kupuku.getZ();
    drawKupu(g2, xKupu, yKupu, zKupu);
}
```

Illustration 5.8: Draw 2D animation

Illustration 5.8: DrawKupu method contains draw 2D animation using graphics panel. GeneralPath is to combine the lines make the image pattern.
Create DemoKupu class

```java
6 class DemoKupu extends JFrame{
7     DemoKupu() {
8         setTitle("Kupu - kupu");
9         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
10        setSize(1000,500);
11        setLocationRelativeTo(null);
12        setVisible(true);
13    }
14    JPanel panel = new JPanel();
15  ```
16     panel.add(new GambarKupu());
17    }
18    }
19    get getContentPane().add(panel);
20    }
21  ```
22  ```
23  ```
24  ```
25 public static void main(String[] args) {
26     new DemoKupu();
27  }
28  ```
29  ```
30  ```
Illustration 5.9: DemoKupu class
Illustration 5.9: DemoKupu class to testing the program.
5.2 Testing

Testing this program

Illustration 5.10: Displays terminal

Illustration 5.10: This terminal displays the results of random.

Illustration 5.11: Result value 2

Illustration 5.11: Butterfly animation moves from left to top right.
Illustration 5.12: Butterfly animation moves left to right.

Illustration 5.13: Butterfly animation moves from left to bottom right.
Illustration 5.14: This program is continues from illustration 5.14. After the butterfly gets to the bottom right, the butterfly moves to the upper right.