

LAMPIRAN A SOURCE CODE

A.1 Source Code MQTT Publisher

```
1 import org.eclipse.paho.client.mqttv3.*;
2 import java.util.Calendar;
3 import java.util.Scanner;
4
5 public class Publisher_Input {
6
7     static MqttClient c;
8     static MqttConnectOptions conOptions;
9     static Callback callback = new Callback();
10    static Scanner input = new Scanner(System.in);
11
12    public static void main(String[] args) {
13        try {
14            System.out.println("Starting demo.");
15            String haURIs[] = new
16String[]{"tcp://0.0.0.0:1883", "tcp://0.0.0.0:1883"};
17            c = new MqttClient("tcp://0.0.0.0:1883",
18MqttClient.generateClientId(), null);
19            c.setCallback(callback);
20            conOptions = new MqttConnectOptions();
21            conOptions.setServerURIs(haURIs);
22            connection();
23        } catch (MqttException e) {
24        }
25        publishAMessage();
26        try {
27            c.disconnect();
28        } catch (MqttException e) {
29        }
30    }
31
32    private static void publishAMessage() {
33        c.setCallback(callback);
34        try {
35            String topik, pesan;
36            System.out.print("Masukkan Topik : ");
37            topik = input.nextLine();
38            System.out.print("Masukkan Pesan : ");
39            pesan = input.nextLine();
40            c.publish(topik, ("Message " +
41pesan).getBytes(), 2, false);
42        } catch (MqttException e) {
43        }
44    }
45
46    private static void connection() {
47        boolean tryConnecting = true;
48        while (tryConnecting) {
49            try {
50                c.connect(conOptions);
51            } catch (Exception e1) {
52                System.out.println("Connection attempt
53failed with '" + e1.getCause() + "'. Retrying.");
54            }
55        }
56    }
57 }
```

```

50         }
51         if (c.isConnected()) {
52             String serverURI = c.getCurrentServerURI();
53             System.out.println("Connected To " +
serverURI + ". Up Timestamp : " +
Calendar.getInstance().getTimeInMillis());
54             tryConnecting = false;
55         } else {
56             pause();
57         }
58     }
59 }
60
61 private static void pause() {
62     try {
63         Thread.sleep(1000);
64     } catch (InterruptedException e) {
65     }
66 }
67
68 private static class Callback implements MqttCallback {
69
70     public void connectionLost(Throwable cause) {
71         System.out.println("Connection lost -
attempting reconnect. Down Timestamp : " +
Calendar.getInstance().getTimeInMillis());
72         connection();
73     }
74
75     @Override
76     public void deliveryComplete(IMqttDeliveryToken
arg0) {
77         System.out.println("Message " + arg0 + "
published. Timestamp : " +
Calendar.getInstance().getTimeInMillis());
78     }
79
80     @Override
81     public void messageArrived(String arg0, MqttMessage
arg1) throws Exception {
82
83     }
84 }
85 }

```

A.2 Source Code MQTT Subscriber

```

1 import org.eclipse.paho.client.mqttv3.*;
2 import java.util.Calendar;
3 import java.util.Scanner;
4
5 public class Subscriber {
6
7     static MqttClient c;
8     static MqttConnectOptions conOptions;
9     static Callback callback = new Callback();
10    static Scanner input = new Scanner(System.in);
11
12    public static void main(String[] args) {

```

```

13         try {
14             System.out.println("Starting demo.");
15             String haURIs[] = new
String[]{"tcp://0.0.0.0:1883", "tcp://0.0.0.0:1883"};
16             c = new MqttClient("tcp://0.0.0.0:1883",
MqttClient.generateClientId(), null);
17             c.setCallback(callback);
18             conOptions = new MqttConnectOptions();
19             conOptions.setServerURIs(haURIs);
20             connection();
21         } catch (MqttException e) {
22         }
23         subscribeATopic();
24         try {
25             c.disconnect();
26         } catch (MqttException e) {
27         }
28     }
29
30     private static void subscribeATopic() {
31         String topik;
32         c.setCallback(new Callback());
33         try {
34             System.out.print("Masukkan Topik : ");
35             topik = input.nextLine();
36             c.subscribe(topik, 2);
37         } catch (MqttException e) {
38         }
39     }
40
41     private static void connection() {
42         boolean tryConnecting = true;
43         while (tryConnecting) {
44             try {
45                 c.connect(conOptions);
46             } catch (Exception e1) {
47                 System.out.println("Connection attempt
failed with '" + e1.getCause() + "'. Retrying.");
48             }
49             if (c.isConnected()) {
50                 String serverURI = c.getCurrentServerURI();
51                 System.out.println("Connected To " +
serverURI + ". Up Timestamp : " +
Calendar.getInstance().getTimeInMillis());
52
53                 tryConnecting = false;
54             } else {
55                 pause();
56             }
57         }
58     }
59
60     private static void pause() {
61         try {
62             Thread.sleep(10000);
63         } catch (InterruptedException e) {
64         }
65     }
66

```

```

67     private static class Callback implements MqttCallback {
68
69         public void connectionLost(Throwable cause) {
70             System.out.println("Connection lost -
attempting reconnect. Down Timestamp : " +
Calendar.getInstance().getTimeInMillis());
71             connection();
72         }
73
74         @Override
75         public void deliveryComplete(IMqttDeliveryToken
arg0) {
76
77         }
78
79         @Override
80         public void messageArrived(String s, MqttMessage
mqttMessage) throws Exception {
81             System.out.println("Message received:\t" + new
String(mqttMessage.getPayload()) + " Message Arrived : " +
Calendar.getInstance().getTimeInMillis());
82         }
83     }
84 }

```