LAMPIRAN C

PROGRAM PADA PERANGKAT PENERIMA (J2ME)

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

import java.io.DataInputStream;
import java.util.Vector;
import javax.bluetooth.DeviceClass;
import javax.bluetooth.DiscoveryAgent;
import javax.bluetooth.DiscoveryListener;
import javax.bluetooth.LocalDevice;
import javax.bluetooth.RemoteDevice;
import javax.bluetooth.ServiceRecord;
import javax.bluetooth.UUID;
import javax.microedition.io.Connector;
import javax.microedition.io.StreamConnection;
import javax.microedition.lcdui.Display;
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

/**
 * @author Mulyawan
 */
public class SPP extends MIDlet implements CommandListener, Runnable,
DiscoveryListener{
    private Display display;
    private Form mainForm;
    private List mainList;
    private static boolean  Busy=false;
    private static String   sURL="";

    private static final Command CMD_FIND = new Command("Search",
Command.ITEM, 1);
    private static final Command CMD_CONN = new Command("Connect",
Command.ITEM, 1);
    private static final Command CMD_EXIT = new Command("Exit",
Command.EXIT, 1);
    private static final Command CMD_BACK = new Command("Back",
Command.BACK, 1);

    private static Vector vecDevices=new Vector();

    private static Object lock=new Object();
```

```
public void exitApp(){
   destroyApp(false);
   notifyDestroyed();
}

public SPP(){
   display = Display.getDisplay(this);

   //Simpan list bluetooth yang terdeteksi
   mainList = new List("Bluetooth SPP", List.EXCLUSIVE);
   mainList.addCommand(CMD_EXIT);
   mainList.addCommand(CMD_FIND);
   mainList.setCommandListener(this);

   //Simpan list bluetooth yang berhasil terkoneksi
   mainForm = new Form("Bluetooth");
   mainForm.addCommand(CMD_BACK);
   mainForm.addCommand(CMD_EXIT);
   mainForm.setCommandListener(this);

   //Data bluetooth
   vecDevices=new Vector();
}

public void startApp() {
   display.setCurrent(mainList);
}

public void pauseApp() {
}

public void destroyApp(boolean unconditional) {
}

public void commandAction(Command c, Displayable d) {
   if (c==CMD_FIND) {
      if (Busy==false) {
         new Thread(this).start();

      }
   }else  if (c==CMD_CONN) {
      if (vecDevices.size()>=1) {
        display.setCurrent(mainForm);
```

```
            new Thread(new Runnable() {
                public void run() {
                    try {
                        mainForm.setTitle("connecting");
                        StreamConnection
streamConnection=Connect((RemoteDevice)vecDevices.elementAt(mainList.getS
electedIndex()));
                        if (streamConnection!=null) {
                            mainForm.removeCommand(CMD_BACK);
                            mainForm.addCommand(CMD_EXIT);
                            mainForm.setTitle("connected");
                            DataInputStream dis=
streamConnection.openDataInputStream();
                            int c=0; StringBuffer sb = new StringBuffer("");
                            while ( (c=dis.read())!=-1 ) {
                                if (c=='\r'){
                                    mainForm.append(sb.toString()+'\n');
                                    sb = new StringBuffer("");
                                }else if (c=='\n'){
                                }else{
                                    sb.append( ((char)c) );
                                }
                            }
                            mainForm.append(sb.toString()+'\n');

                        }else{
                            mainForm.removeCommand(CMD_EXIT);
                            mainForm.addCommand(CMD_BACK);
                            mainForm.setTitle("bukan SPP");
                        }
                    } catch (Exception e) {}
                }
            }).start();
        }
    }else  if (c==CMD_BACK) {
        display.setCurrent(mainList);
    }else  if (c==CMD_EXIT) {
        destroyApp(false);
        notifyDestroyed();
    }
  }
```

```
   public void run() {
      try {
            Busy=true;
            LocalDevice localDevice = LocalDevice.getLocalDevice();
            DiscoveryAgent agent = localDevice.getDiscoveryAgent();
            agent.startInquiry(DiscoveryAgent.GIAC, this);
                  //lakukan pencarian bluetooth

            mainList.removeCommand(CMD_FIND);
            mainList.removeCommand(CMD_CONN);
            mainList.setTitle("SPP searching");

            try {
               synchronized(lock){
                  lock.wait();
                  //berhenti disini sampai proses pencarian bluetooth selesai
               }
            }catch (InterruptedException e) {}

            int deviceCount=vecDevices.size();
            mainList.deleteAll();
            if(deviceCount <= 0){
               System.out.println("No Devices Found .");
               mainList.addCommand(CMD_FIND);
            }else{
               mainList.addCommand(CMD_CONN);
               System.out.println("Bluetooth Devices: ");
               for (int i = 0; i <deviceCount; i++) {
                  RemoteDevice
remoteDevice=(RemoteDevice)vecDevices.elementAt(i);
                  mainList.append(remoteDevice.getFriendlyName(true), null);
                  System.out.println((i+1)+".
"+remoteDevice.getBluetoothAddress()+"
("+remoteDevice.getFriendlyName(true)+")");
               }
            }

       } catch (Exception e) {mainList.addCommand(CMD_FIND);}
       mainList.setTitle("Bluetooth SPP");
       Busy=false;
   }

   public void deviceDiscovered(RemoteDevice btDevice, DeviceClass cod) {
      if(!vecDevices.contains(btDevice)){
         vecDevices.addElement(btDevice);
      }
```

I

II

```
    }
    public void inquiryCompleted(int discType) {
        synchronized(lock){
            lock.notify();
            //pencarian bluetooth selesai, beri notifikasi agar dapat melanjutkan ke
baris bawahnya
        }
    }

    public void servicesDiscovered(int transID, ServiceRecord[] servRecord) {
        if(servRecord!=null && servRecord.length>0){
            sURL=servRecord[0].getConnectionURL(0,false);
        }
        synchronized(lock){
            lock.notify();
        }
    }
    public void serviceSearchCompleted(int transID, int respCode) {
        synchronized(lock){
            lock.notify();
            //beri notifikasi ke pemeriksa service, kalau pemeriksaan service benar
        }
    }


    public StreamConnection Connect(RemoteDevice btDevice) {
        try {
            sURL="";
            UUID[] uuidSet = new UUID[1];
            uuidSet[0]=new UUID("1101",false); //UUID untuk SSP bluetooth
            LocalDevice localDevice = LocalDevice.getLocalDevice();
            DiscoveryAgent agent = localDevice.getDiscoveryAgent();
            agent.searchServices(null,uuidSet,btDevice,this);
            //periksa dulu apakah bluetooth target membuka/support SSP
            try {
                  synchronized(lock){
                     lock.wait();
                     //tunggu disini sampai proses pemeriksaan service selesai
                  }
            }catch (InterruptedException e) {}
            return (StreamConnection)Connector.open(sURL);
            //sekarang lakukan koneksi yang sebenarnnya

        } catch (Exception e) {return null;}
    }
}
```

III

IV

V