1.Tujuan

- Mengetahui fungsionalitas yang disediakan oleh Mobile Media API (MMAPI)
- Memainkan nada sederhana
- Menjalankan file audio dari jaringan dan file JAR
- Mengirim dan menerima pesan SMS
- Berkomunikasi wireless menggunakan protokol bluetooth

2. Latar Belakang

Tidak seluruh device terbuat sama dan tiap class device memiliki fitur yang berbeda–beda pula. Sangatlah sulit untuk membuat spesifikasi standar yang meliputi seluruh device yang telah ada.
Untuk mengakomodasi perbedaan kemampuan dari device, MIDP memiliki beberapa optional packages. Packages – packages tersebut adalah spesifik dan memenuhi fitur – fitur umum spesifik.
Bab ini akan membahas bagaimana memulai penggunaan Mobile Media API (MMAPI) dan Wireless Messaging API (WMA).

3. Percobaan

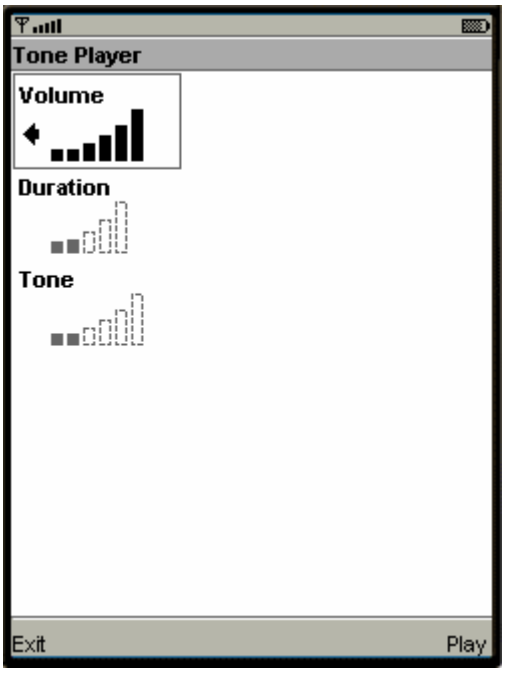
Percobaan 1 Pembuatan Nada :

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
import javax.microedition.media.*;
import javax.microedition.media.control.*;
import java.io.*;
```

```
public class ToneMIDlet extends MIDlet implements CommandListener{
    private Command exitCommand, playCommand;
    private Form form;
    private Gauge volumeGauge;
    private Gauge durationGauge;
    private Gauge toneGauge;
    private Display display;
    private int duration = 2; // seconds
    private int volume = 100;
    private int tone = ToneControl.C4;
    private static int MAX_VOLUME = 100;
    private static int MAX_TONE = 127;
    private static int MAX_DURATION = 5;

    public ToneMIDlet() {
        playCommand = new Command("Play", Command.OK, 1);
        exitCommand = new Command("Exit", Command.EXIT, 1);
        volumeGauge = new Gauge("Volume", true, MAX_VOLUME, volume);
        toneGauge = new Gauge("Tone", true, MAX_TONE, tone);
        durationGauge = new Gauge("Duration", true, MAX_DURATION,
duration);
        form = new Form("Tone Player");
        form.addCommand(playCommand);
        form.addCommand(exitCommand);
        form.append(volumeGauge);
        form.append(durationGauge);
        form.append(toneGauge);
    }

    public void startApp() {
        display = Display.getDisplay(this);
        form.setCommandListener(this);
        display.setCurrent(form);
    }

    public void pauseApp() {}
    public void destroyApp(boolean unconditional) {}
```

```
        public void startApp() {
            display = Display.getDisplay(this);
            form.setCommandListener(this);
            display.setCurrent(form);
        }

        public void pauseApp() {}
        public void destroyApp(boolean unconditional) {}

        public void commandAction(Command c, Displayable d) {
            if (c == exitCommand) {
                notifyDestroyed();
            }
            if (c == playCommand){
                try {
                    volume = volumeGauge.getValue();
                    tone = toneGauge.getValue();
                    duration = durationGauge.getValue();
                    Manager.playTone(tone, duration*1000, volume);
                } catch (MediaException mex){}
            }
        }
}
```

Output:



Percobaan 2 Memainkan Nada Dari URI :

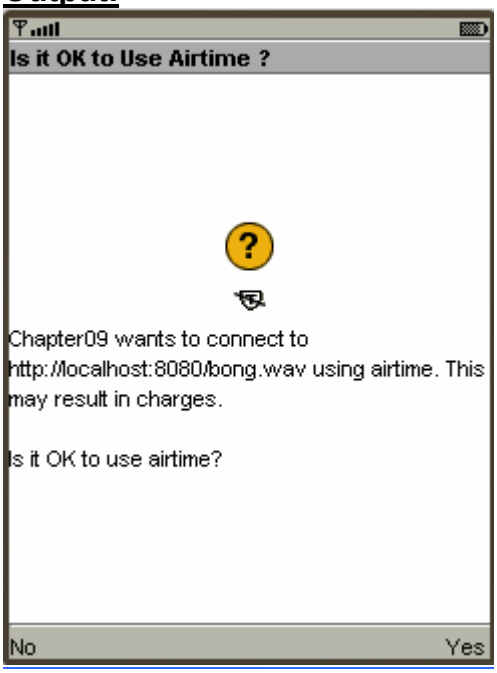
```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
import javax.microedition.media.*;
import javax.microedition.media.control.*;

import java.io.*;
public class NetAudioMidlet extends MIDlet implements CommandListener{

    private Command exitCommand, playCommand;
    private Form form;
```

```
    private Gauge volumeGauge;
        private Display display;
        private int volume = 100;
        private static int MAX_VOLUME = 100;
        Player player;

        public NetAudioMidlet() {
            playCommand = new Command("Play", Command.OK, 1);
            exitCommand = new Command("Exit", Command.EXIT, 1);
            volumeGauge = new Gauge("Volume", true, MAX_VOLUME, volume);

            form = new Form("Audio Player");
            form.addCommand(playCommand);
            form.addCommand(exitCommand);
            form.append(volumeGauge);
        }

        public void startApp() {
            display = Display.getDisplay(this);
            form.setCommandListener(this);
            display.setCurrent(form);

            try {
                player =
    Manager.createPlayer("http://localhost:8080/bong.wav");
                player.realize();
```

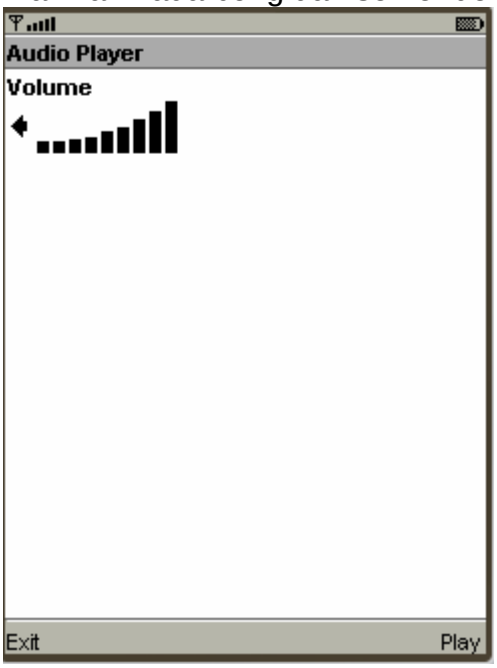
```
// pre-fetch media to reduce latency
            player.prefetch();
        } catch (IOException ioex) {
            display.setCurrent(new Alert("IO Exception",
ioex.getMessage(),  null, AlertType.ERROR));
        }  catch (MediaException mex) {
            display.setCurrent(new Alert("Media Exception",
mex.getMessage(),  null, AlertType.ERROR));
        }
    }

    public void pauseApp() {}
 public void destroyApp(boolean unconditional) {}
    public void commandAction(Command c, Displayable d) {
        if (c == exitCommand) {
            notifyDestroyed();
        }
        if (c == playCommand){
            try {
                VolumeControl control = (VolumeControl)
player.getControl("VolumeControl");
                if (control != null){
                    control.setLevel(volumeGauge.getValue());
                }
                player.start();
            }  catch (MediaException mex) {
                display.setCurrent(new Alert("Media Exception",
mex.getMessage(), null, AlertType.ERROR));
} catch (Exception ex){
                display.setCurrent(new Alert("Exception",
ex.getMessage(), null, AlertType.ERROR));
            }
        }
    }
}
```

Catatan:
Sebelum Anda dapat memainkan nada bong.wav dari URI, Anda harus menjalankan server terlebih dahulu dan kemudian letakkanlah nada bong.wav disana.
Pada percobaan kali ini kami menggunakan Tomcat Server dengan default URI: http://localhost:8080, kemudian kami letakkan bong.wav dalam folder root, sehingga bisa kami akses pada http://localhost:8080/bong.wav
Running Program:
Pada saat program dijalankan terlebih dahulu Anda akan mendapatkan interface mengenai kemungkinan penggunaan airtime. Oleh karena kita akan menggunakan koneksi http yang biasanya dipungut biaya airtime, maka klik yes pada interface tersebut.

Output:



Mainkan nada bong dari server dengan cara memilih command play



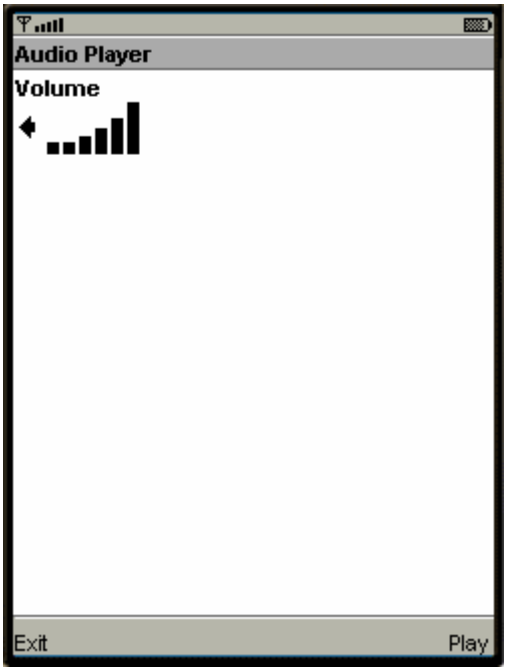
Percobaan 3 Memainkan Media Dari File JAR :

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
import javax.microedition.media.*;
import javax.microedition.media.control.*;
import java.io.*;

public class AudioMidlet extends MIDlet implements CommandListener{
    private Command exitCommand, playCommand;
    private Form form;
    private Gauge volumeGauge;
    private Display display;
    private int volume = 100;
    private static int MAX_VOLUME = 100;
    Player player;
```

```
       public void startApp() {
           display = Display.getDisplay(this);
           form.setCommandListener(this);
           display.setCurrent(form);
       }

       public void pauseApp() {}
    public void destroyApp(boolean unconditional) {}
       public void commandAction(Command c, Displayable d) {
           if (c == exitCommand) {
               notifyDestroyed();
           }
           if (c == playCommand){
               try {
                   InputStream stream =
   getClass().getResourceAsStream("bong.wav");
                   player = Manager.createPlayer(stream, "audio/x-wav");
                   player.realize();

                VolumeControl control = (VolumeControl)
player.getControl("VolumeControl");
          if (control != null){
                       control.setLevel(volumeGauge.getValue());
                   }
                player.start();
```

```
                }  catch (MediaException mex) {
                    display.setCurrent(new Alert("Media Exception",
    mex.getMessage(),  null, AlertType.ERROR));
                } catch (Exception ex){
                    display.setCurrent(new Alert("Exception",
    ex.getMessage(),  null, AlertType.ERROR));
                }
            }
        }
}
```

Output:



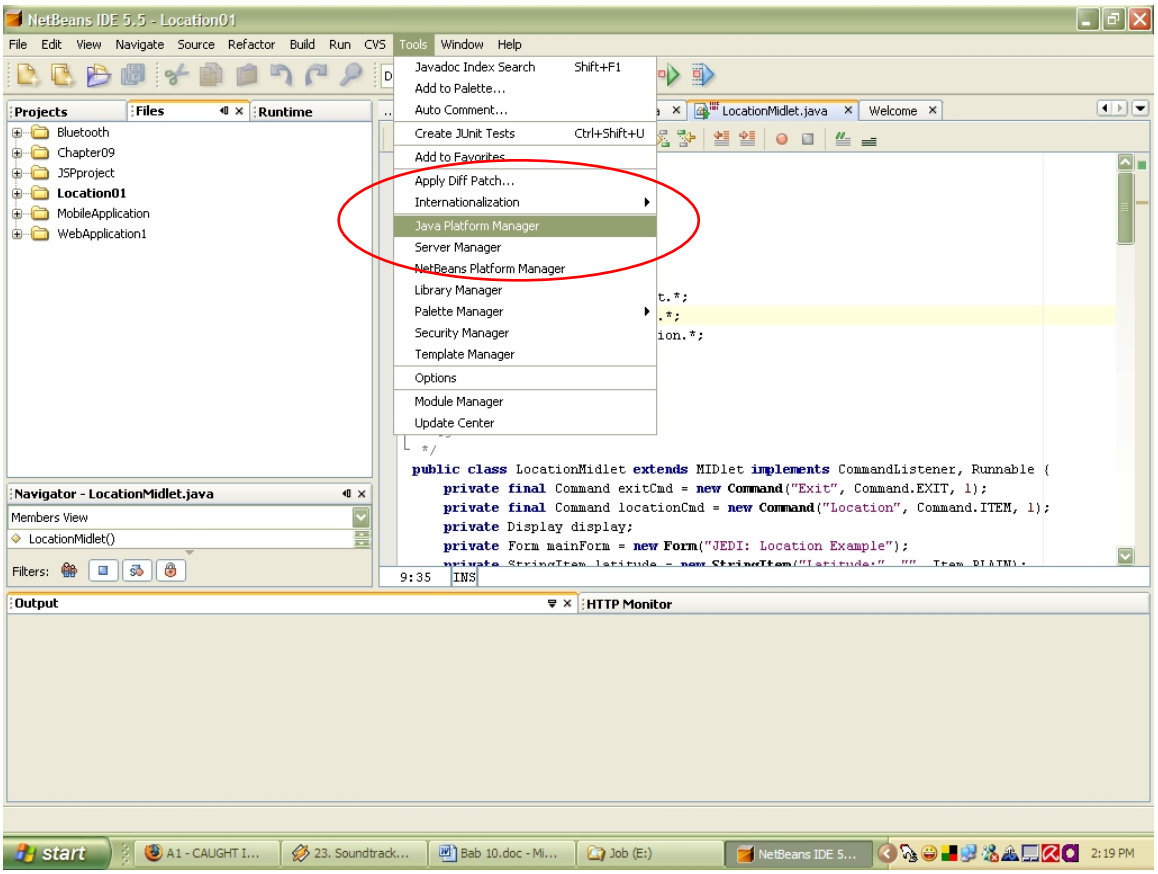
Percobaan 4 Mengirimkan Sms :

```
   import javax.microedition.lcdui.*;
   import javax.microedition.io.*;
   import javax.wireless.messaging.*;

   public class SMSMidlet extends MIDlet implements CommandListener,
   Runnable {

       private Command exitCommand, sendCommand;
       private Form form;
       private TextField addressField, mesgField;
       private Display display;
    private Thread thread;
       public SMSMidlet() {
           sendCommand = new Command("Send", Command.OK, 1);
           exitCommand = new Command("Exit", Command.EXIT, 1);

           addressField = new TextField("Phone Number", "+5550000", 32,
   TextField.ANY);
            mesgField = new TextField("Message", "hello, world!", 160,
TextField.ANY);
           form = new Form("SMS Message");
           form.append(addressField);
           form.append(mesgField);
          form.addCommand(sendCommand);
           form.addCommand(exitCommand);
       }
```

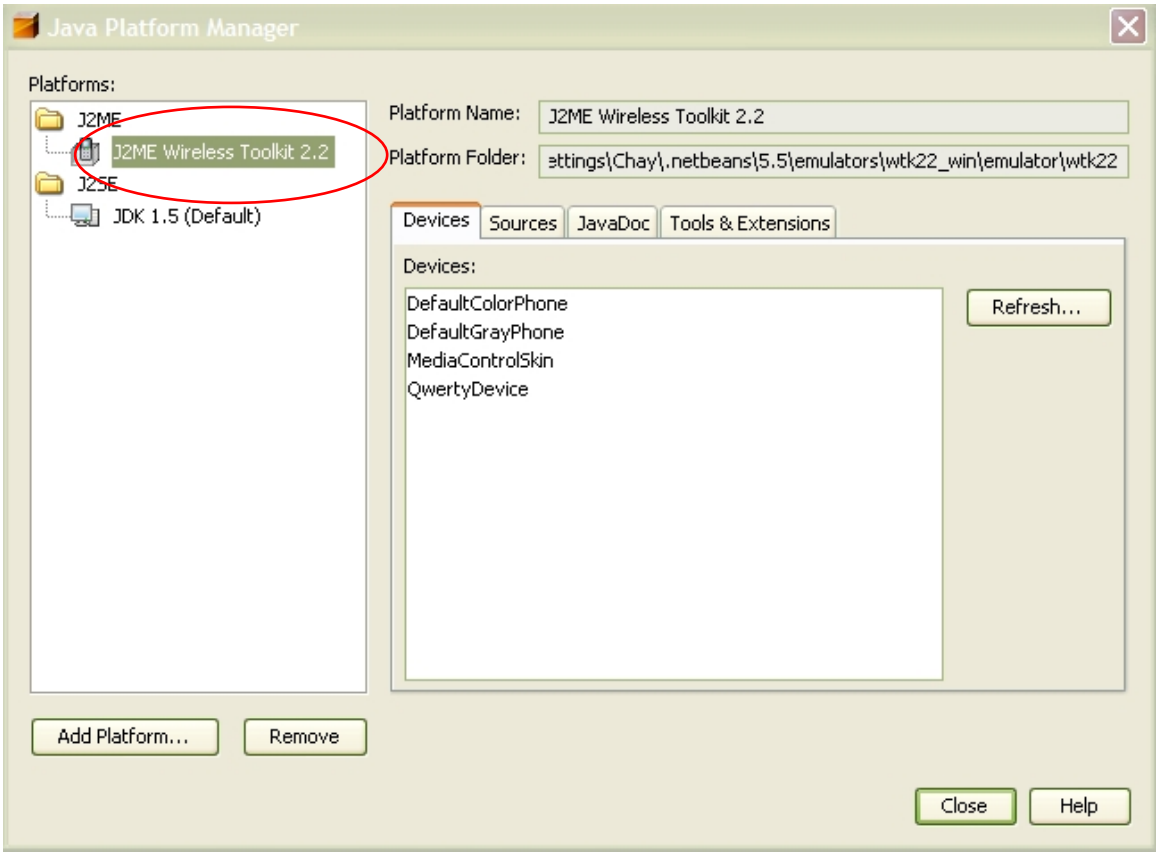
```
public void startApp() {
    display = Display.getDisplay(this);
    form.setCommandListener(this);
    display.setCurrent(form);
}
public void pauseApp() {
    thread = null;
}
public void destroyApp(boolean unconditional) {
    thread = null;
}
public void commandAction(Command c, Displayable d) {
    if (c == exitCommand) {
        notifyDestroyed();
    }
    if (c == sendCommand) {
        thread = new Thread( this );
        thread.start();
    }
}
public void sendSMS(String number, String message) throws Exception{
    String url = "sms://" +  number;
    MessageConnection connection =
         (MessageConnection) Connector.open(url);
```

```
            TextMessage msg = (TextMessage) connection.newMessage(
                    MessageConnection.TEXT_MESSAGE);
            msg.setPayloadText(message);
            connection.send(msg);
            connection.close();
        }

        public void run() {
            try {
                String address = addressField.getString();
                String message = mesgField.getString();
             sendSMS(address, message);
    display.setCurrent(new Alert("SMS Message",
                        "Message Sent\n"
                        + "To: " + address + "\n"
                        + "Message: " + message,
                        null, AlertType.INFO));
            } catch (Exception ex) {
                display.setCurrent(new Alert("SMS Error", ex.getMessage(),
                        null, AlertType.ERROR));
            }
        }
     }
```

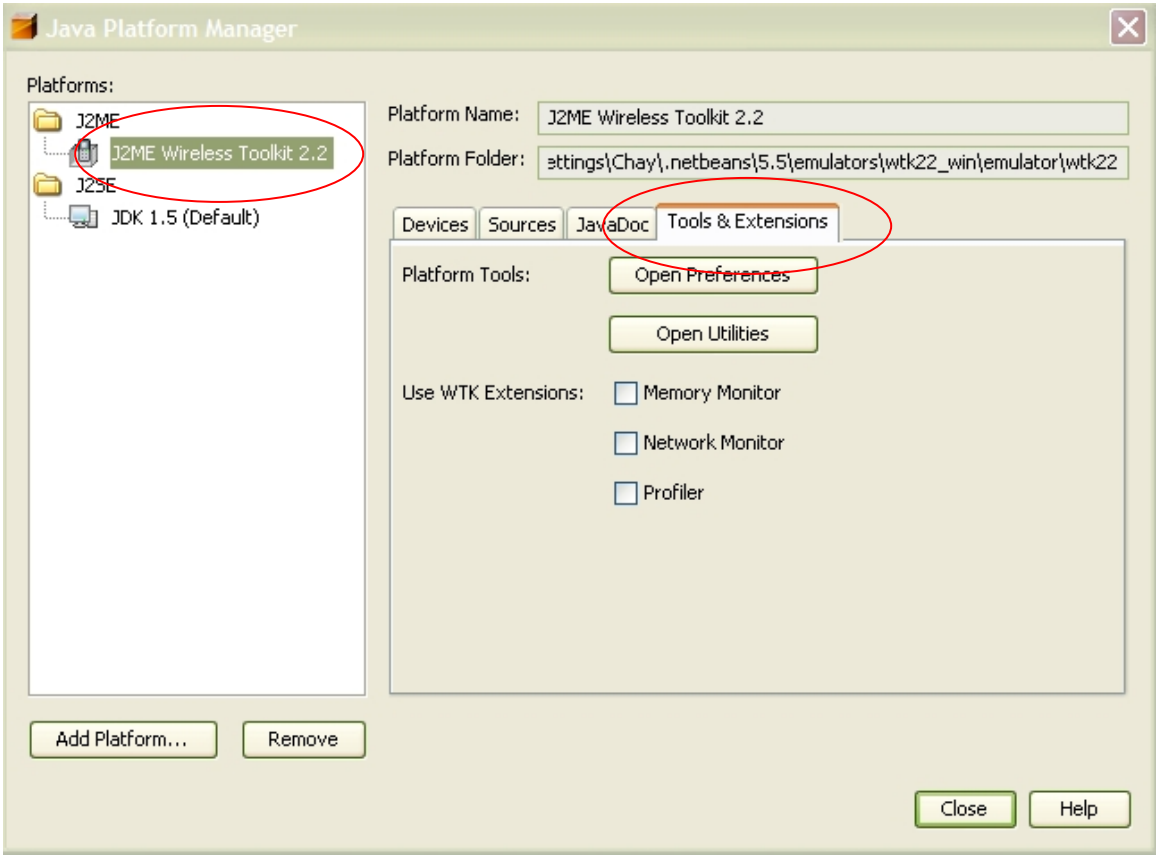
Untuk mengkondisikan sehingga emulator Anda mampu mensimulasikan pengiriman dan penerimaan SMS,ikutilah langkah-langkah berikut ini:
a. Pilih Tools -> Java Platform Manager



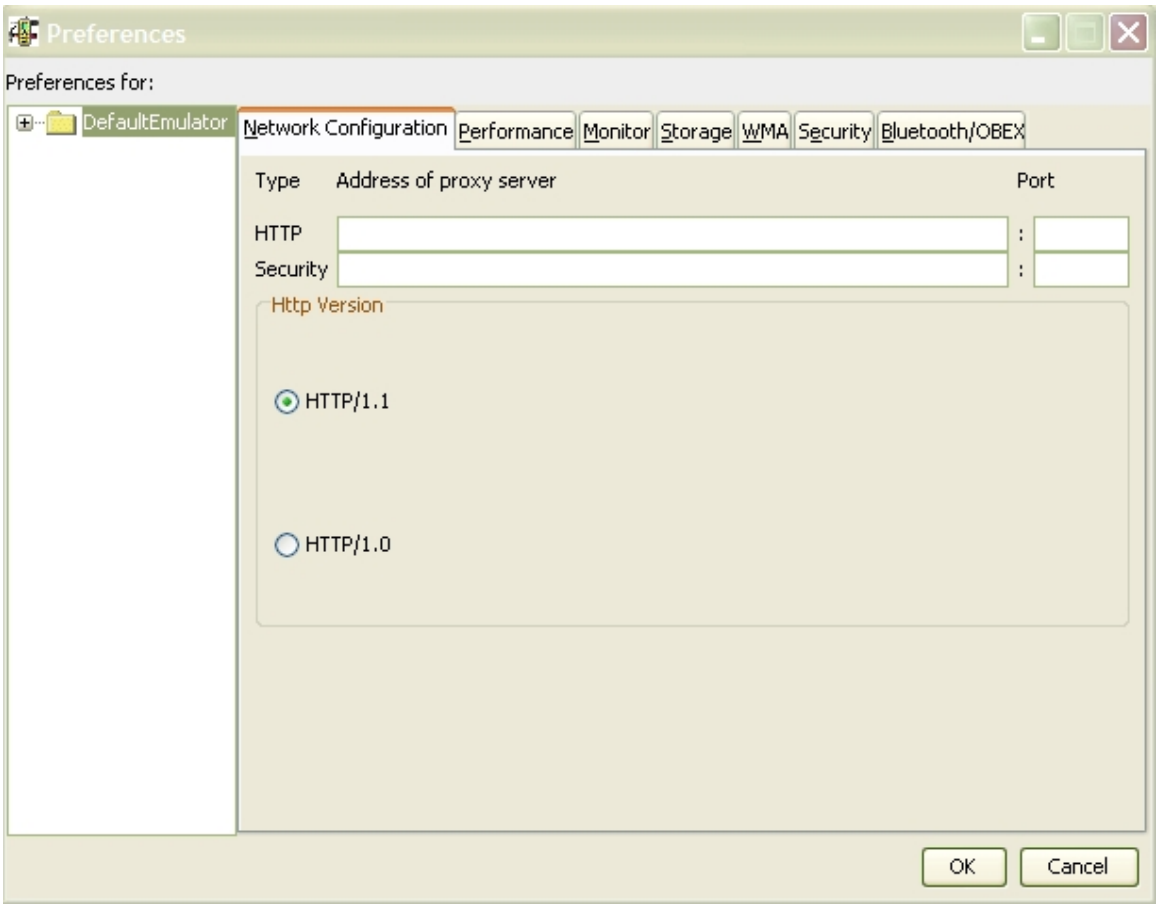
b. Java Platform Manager -> J2ME Wireles Toolkit 2.2



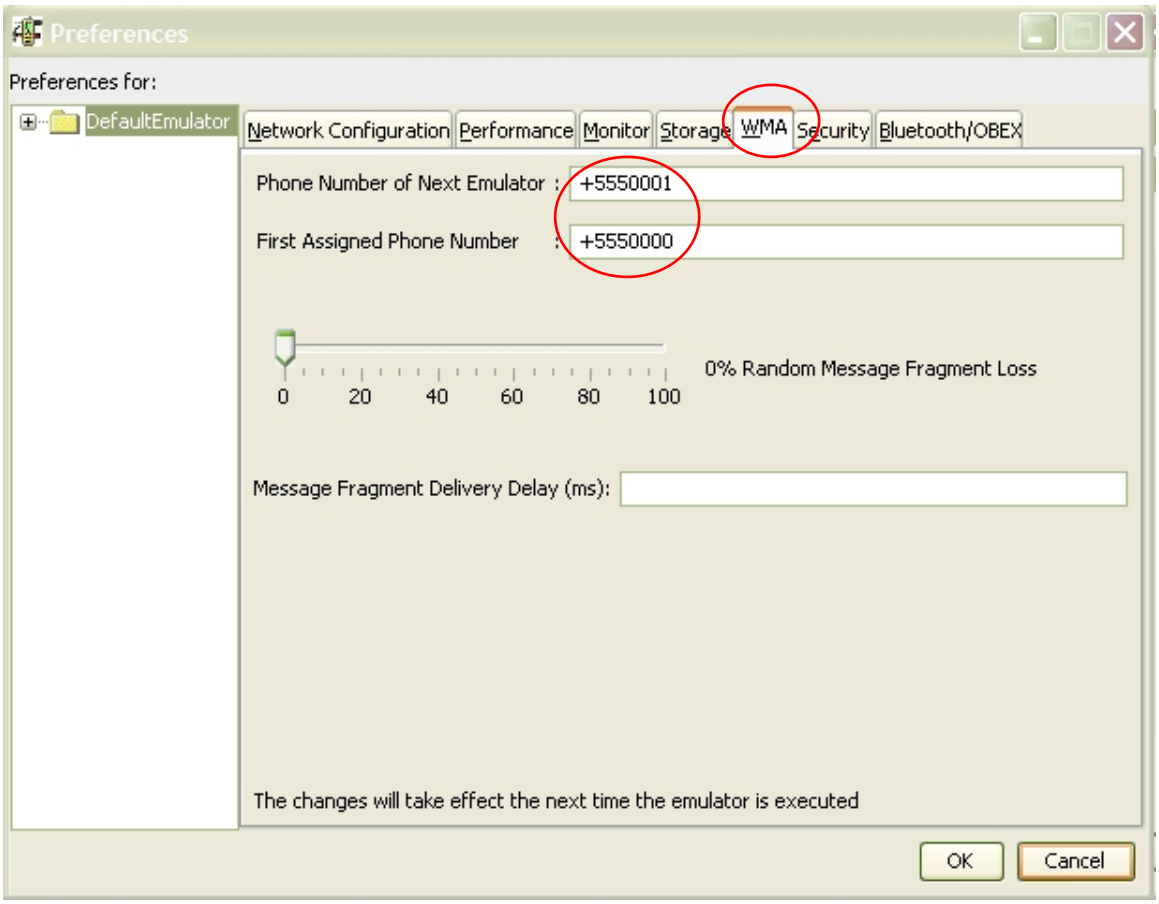
c. Pilih tab Tools & Extensions



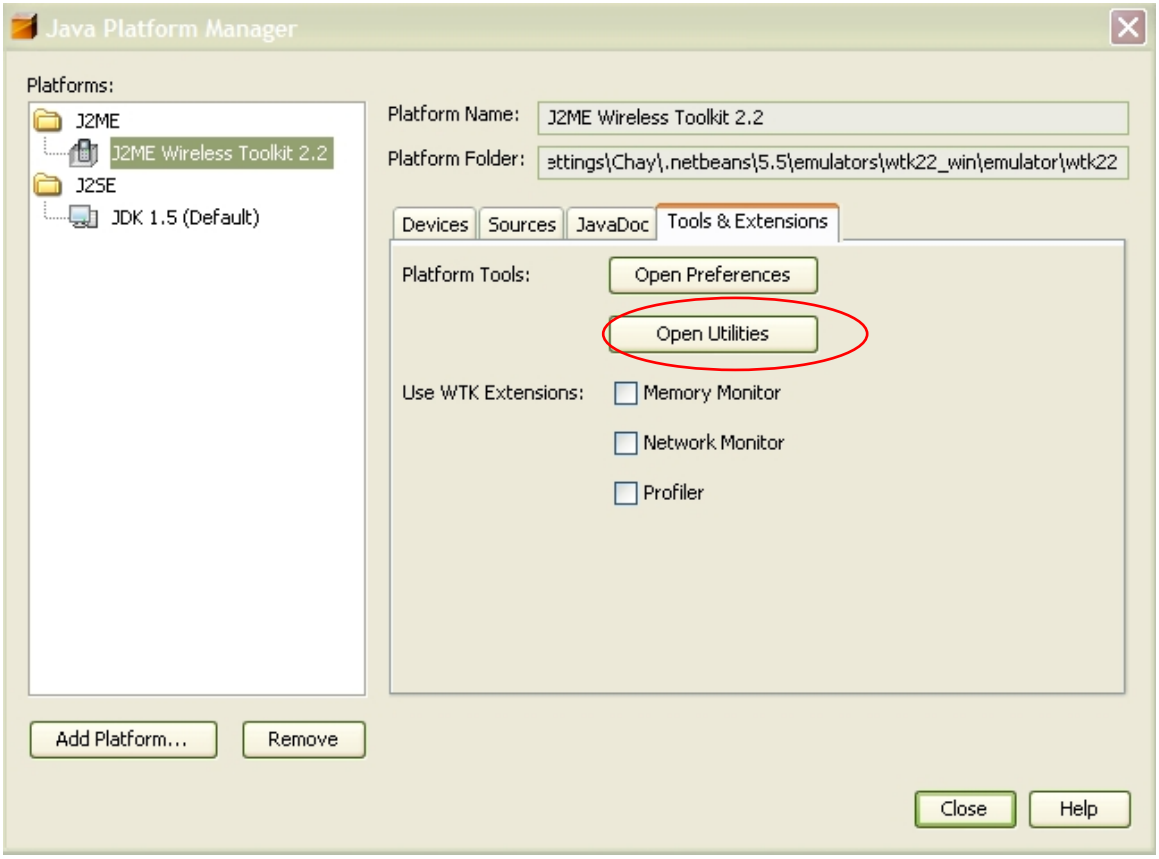
d. Pilih button Open Preferences, sehingga Anda mendapatkan interface sbb :



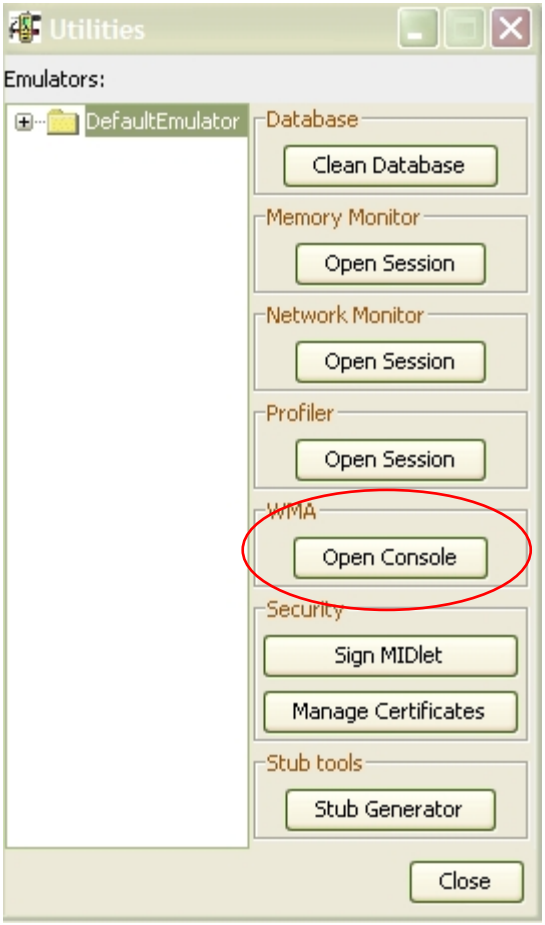
e. Pilih tab WMA untuk melaksanakan setting port yang digunakan untuk komunikasi



f. Tekan “Ok” untuk menyelesaikan setting port yang telah Anda lakukan
g. Kembalilah pada tab tools & Extention pada Java Platform Manager, kemudian pilih tab open utilites



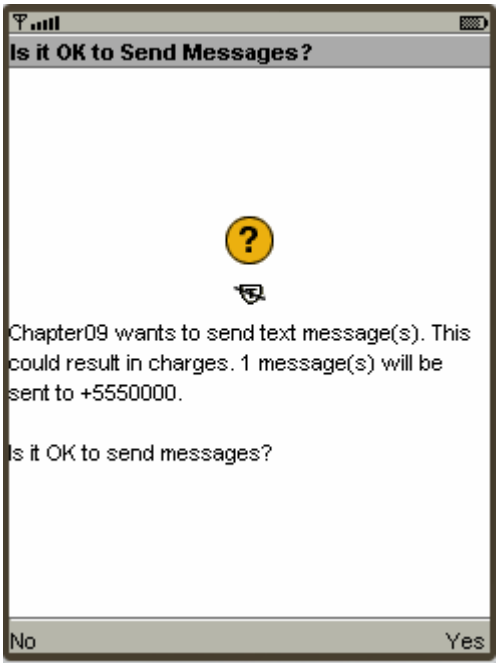
h. Pilihlah button open Console, untuk mencoba menerima SMS dari HP



i. Interface dari WMA Console yang bekerja pada +5550000

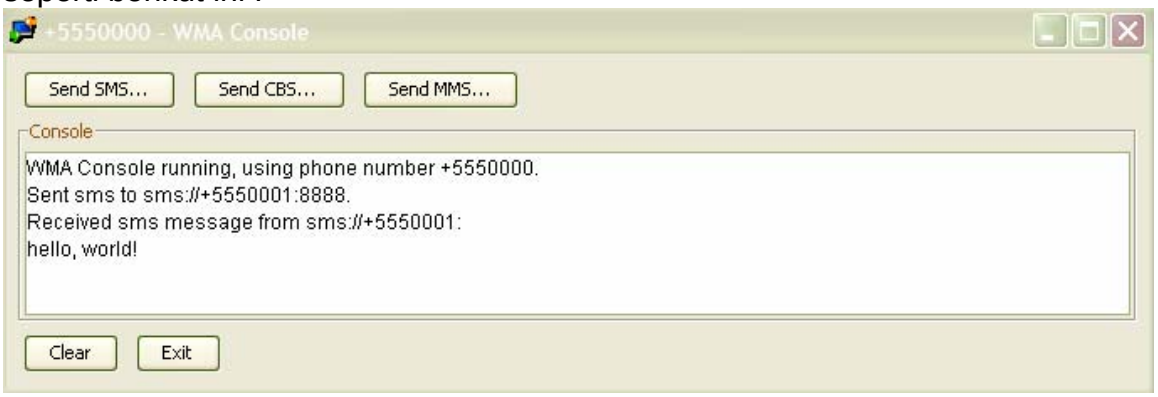
j. Emulator Hand Phone yang bekerja pada nomor +5550001 dan digunakan untuk mengirimkan pesan ke WMA Console

k. Alert pengiriman pesan akan tampil apabila command Send ditekan. Klik command yes untuk mengirimkan pesan



l. Apabila pesan telah terkirim maka akan muncul seperti berikut ini di emulator Hand phone (+5550001)

m. Pesan yang telah terkirim akan diterima oleh WMA Console, sehingga terdapat pesan seperti berikut ini :



Percobaan 5 Menerima Sms :

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
import javax.microedition.io.*;
import javax.wireless.messaging.*;

public class SMSReceiverMidlet extends MIDlet
     implements CommandListener, MessageListener, Runnable  {
    private Command exitCommand, sendCommand;
    private Form form;
    private StringItem statusField, addressField, mesgField, dateField;
    private Display display;
    private MessageConnection conn;
    private Thread thread;
    private String port = "8888";
  public SMSReceiverMidlet() {
     exitCommand = new Command("Exit", Command.EXIT, 1);
     statusField = new StringItem("Status:", "");
     addressField = new StringItem("From:", "");
     mesgField = new StringItem("Message:", "");
     dateField = new StringItem("Timestamp:", "");

     form = new Form("SMS Receiver");
     form.append(statusField);
     form.append(addressField);
```

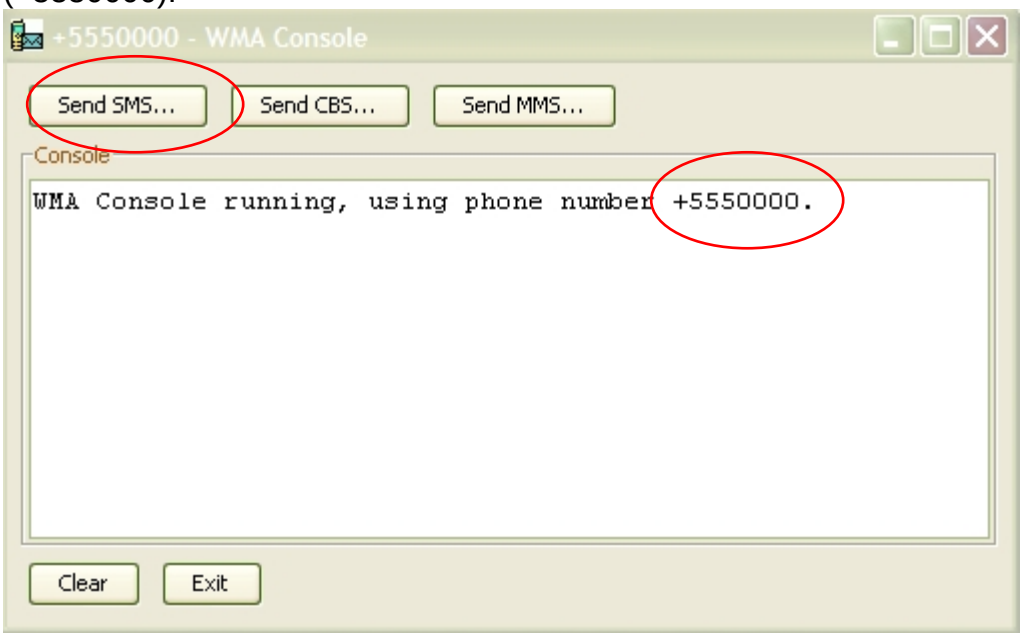
```
            form.append(mesgField);
            form.append(dateField);
            form.addCommand(exitCommand);
        }
        public void startApp() {
            display = Display.getDisplay(this);
            form.setCommandListener(this);
            startReceiver();
            display.setCurrent(form);
    }
        public void pauseApp() {
            thread = null;
        }
        public void destroyApp(boolean unconditional) {
            thread = null;
            if (conn != null){
                try {
                    conn.close();
                } catch (Exception ex){}
            }
    }
        public void commandAction(Command c, Displayable d) {
            if (c == exitCommand) {
                notifyDestroyed();
            }
    }
```

```
private void startReceiver(){
        try {
            String addr = "sms://:" + port;
            if (conn == null){
                conn = (MessageConnection) Connector.open(addr);

                conn.setMessageListener(this);
                statusField.setText(
                        "waiting for message at port " + port);
            }
        } catch (Exception ex){
            statusField.setText("Cannot open connection on port "
                    + port + ":" + ex.getMessage());
        }
        thread = new Thread(this);
        thread.start();
 }
    public void notifyIncomingMessage(MessageConnection
messageConnection) {
        if (thread == null){
            thread = new Thread(this);
            thread.start();
        }
 }
```

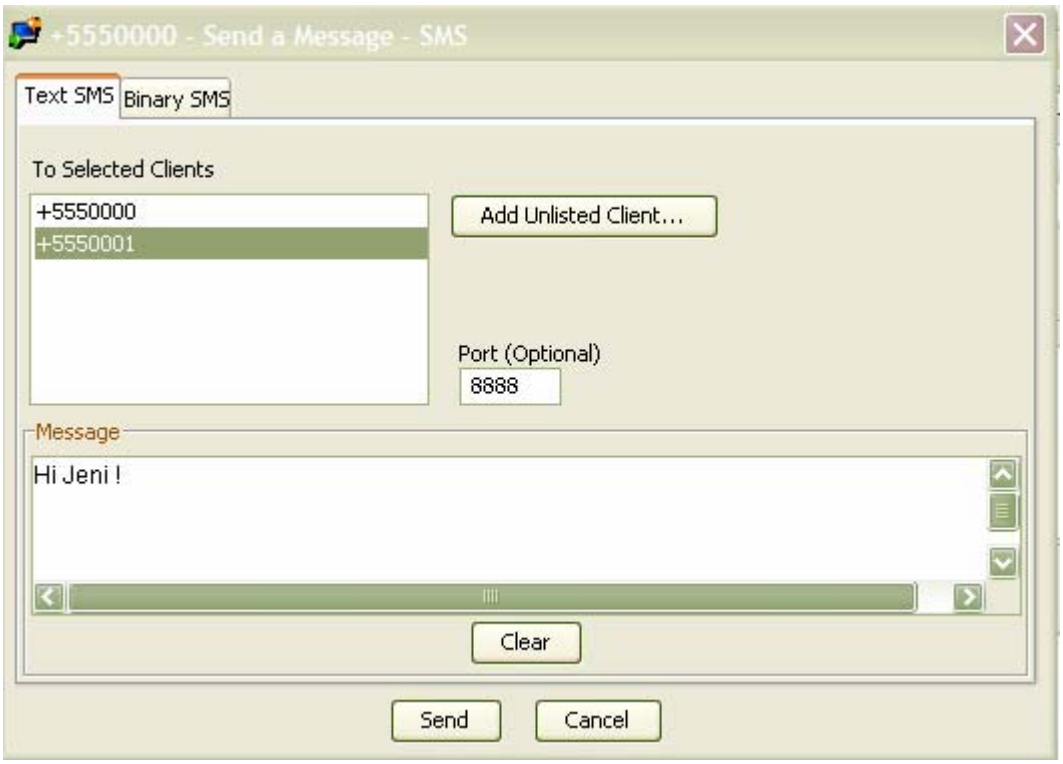

jeNi
CENTER
> > > Java Education Network Indonesia

```
   public void run(){
           try {
               // Menanti pesan diterima
               Message mesg = conn.receive();

               // Pesan telah diterima
               // Periksa apakah pesan tersebut berupa SMS (bukan MMS)
               if (mesg != null && mesg instanceof TextMessage) {
                   TextMessage text = (TextMessage) mesg;
                   addressField.setText(text.getAddress());
                   mesgField.setText(text.getPayloadText());
                   dateField.setText("" + text.getTimestamp());
                   statusField.setText("Message received.");
               } else {
                   statusField.setText("Non-text message received: " +
   mesg.getClass().toString());

               }
           } catch (Exception e) {
               statusField.setText("Error: " + e.getMessage());
           }
           thread = null;
       }
}
```

a. Bukalah console WMA (sama dengan langkah-langkah diatas). Kemudian tekan tombol send sms, untuk mencoba mengirimkan SMS ke emulator yang dituju. Perlu diperhatikan juga, keterangan pada layar mengenai nomor telephone yang digunakan oleh console (+5550000).

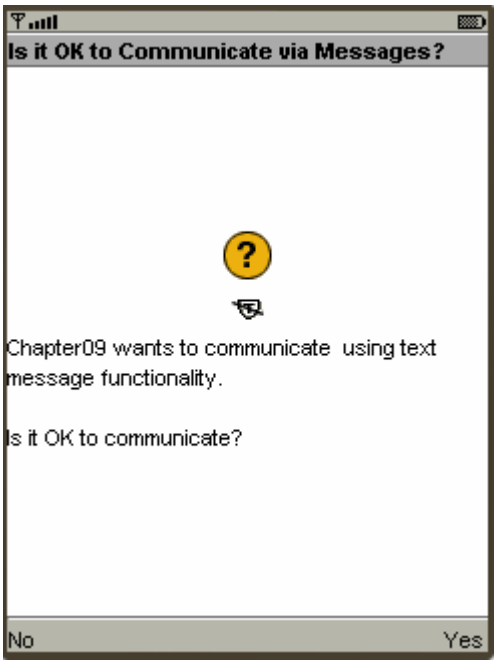


b. Berikut ini adalah interface yang muncul setelah Anda menekan tombol “Send SMS” pada console WMA. Pilihlah nomor emulator yang dituju (+5550001),tuliskanlah pesan yang Anda inginkan pada text box “Message”, dan juga tuliskanlah nomor port yang akan digunakan untuk berkomunikasi, kemudian klik tombol “Send”.



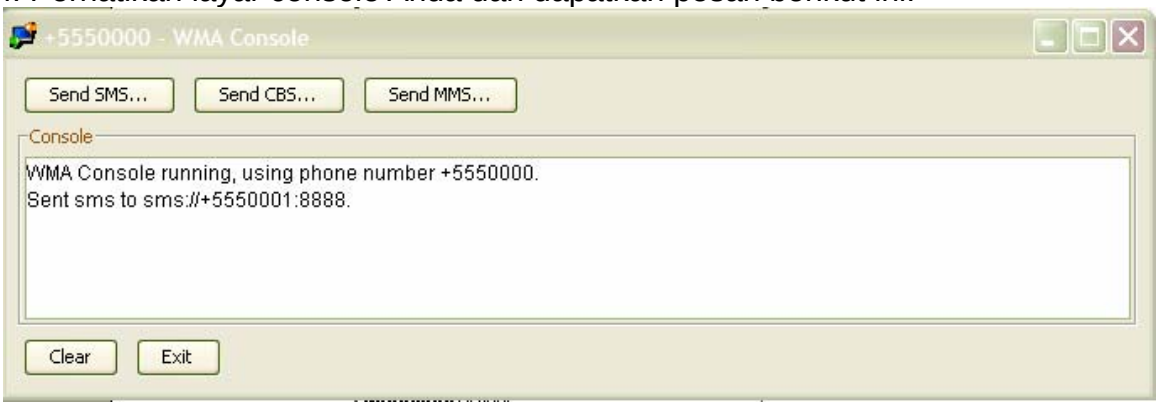
c. Berikut ini adalah tampilan dari emulator untuk menerima pesan. Sebelum ada pesan yang dikirimkan dari console, maka status dari emulator tersebut adalah menunggu pesan pada port 8888.

d. Apabila Ada pesan yang masuk, maka secara otomatis pesan berikut ini akan muncul sebagai penanda bahwa Emulator Anda akan berkomunikasi menggunakan SMS. Klik “Yes” untuk meneruskan penerimaan pesan dari WMA Console



e. Apabila pesan telah diterima, maka akan muncul interface berikut ini pada layar emulator Anda.

f. Perhatikan layar console Anda dan dapatkan pesan berikut ini:



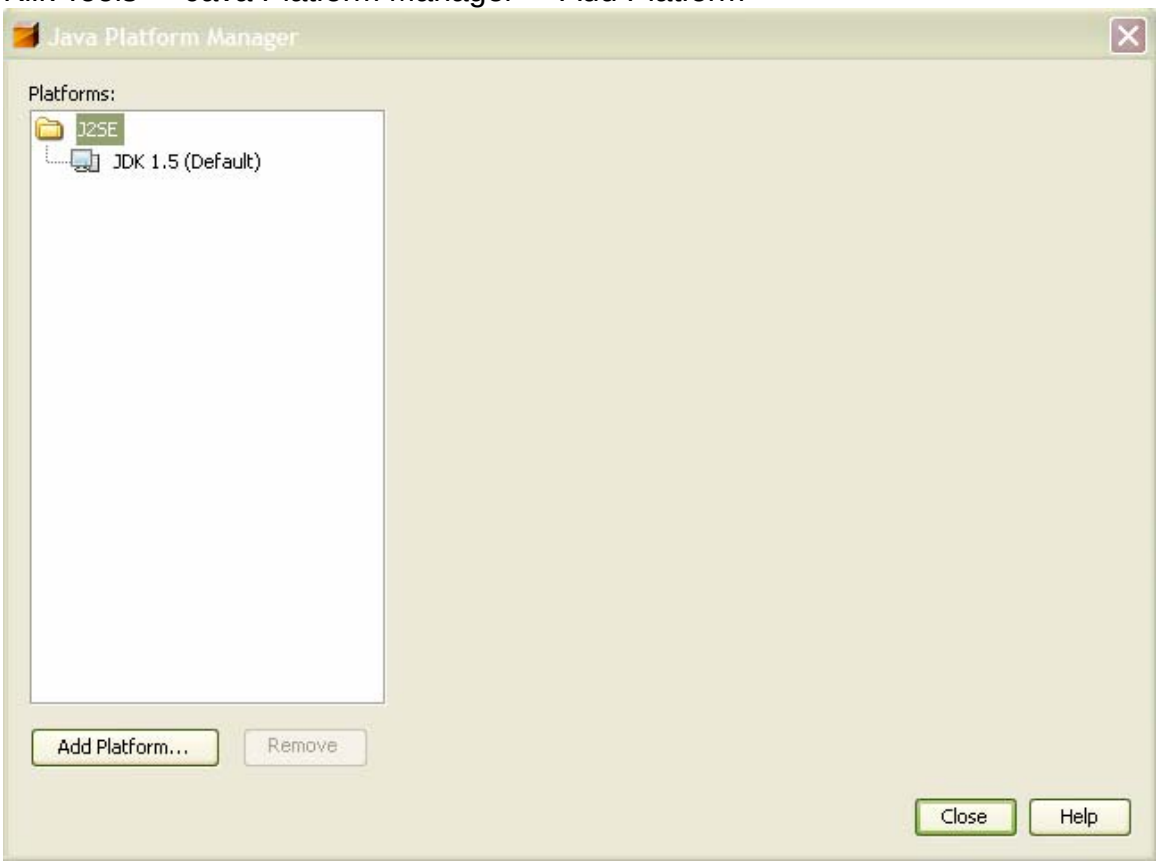
Materi Tambahan:
Mengenali Environment Sun Java Wireless Toolkit 2.5

Untuk mensimulasikan komunikasi Bluetooth dan Location API, Anda membutuhkan Sun Java Wireless Toolkit yang harus diintegrasikan kedalam Netbeans

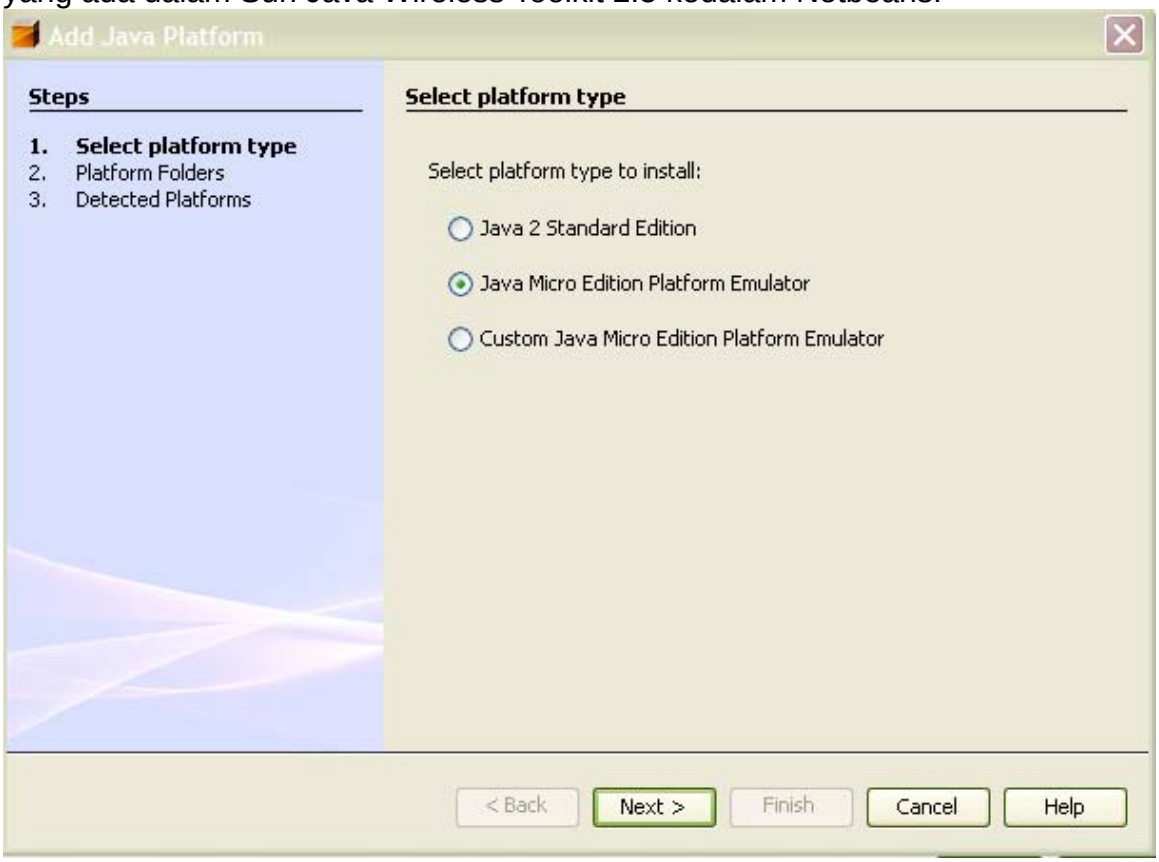
1. Install Sun Java Wireless Toolkit 2.5 kedalam Netbeans

Untuk menggunakan Sun Java Wireless Toolkit 2.5 sebagai bagian terintegrasi dalam Netbeans(dengan catatan bahwa Anda telah meng-install toolkit tersebut kedalam sistem Anda), Anda harus melaksanakan langkah sebagai berikut:

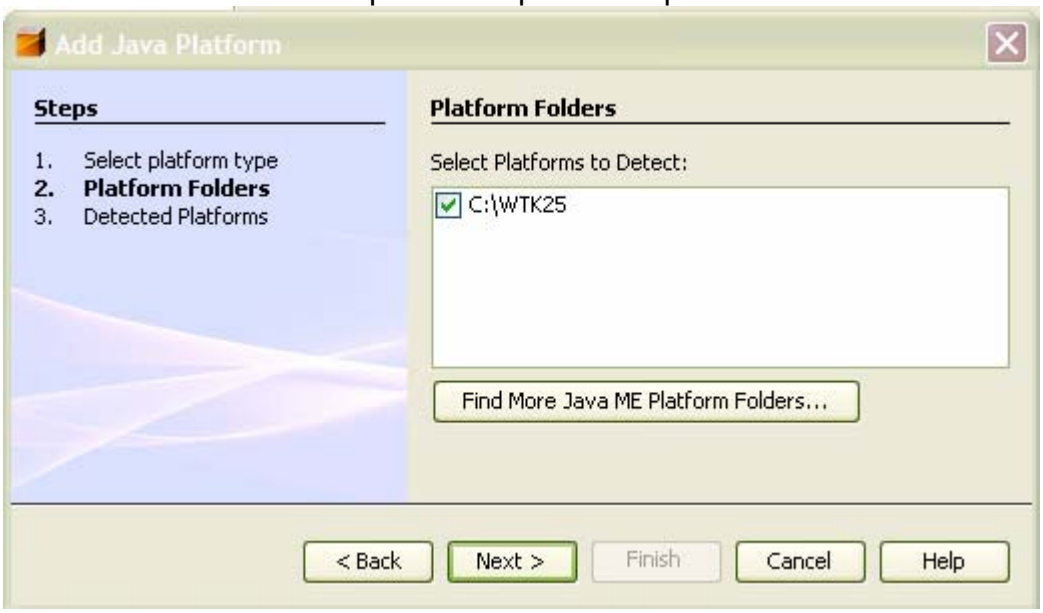
Klik Tools -> Java Platform Manager -> Add Platform



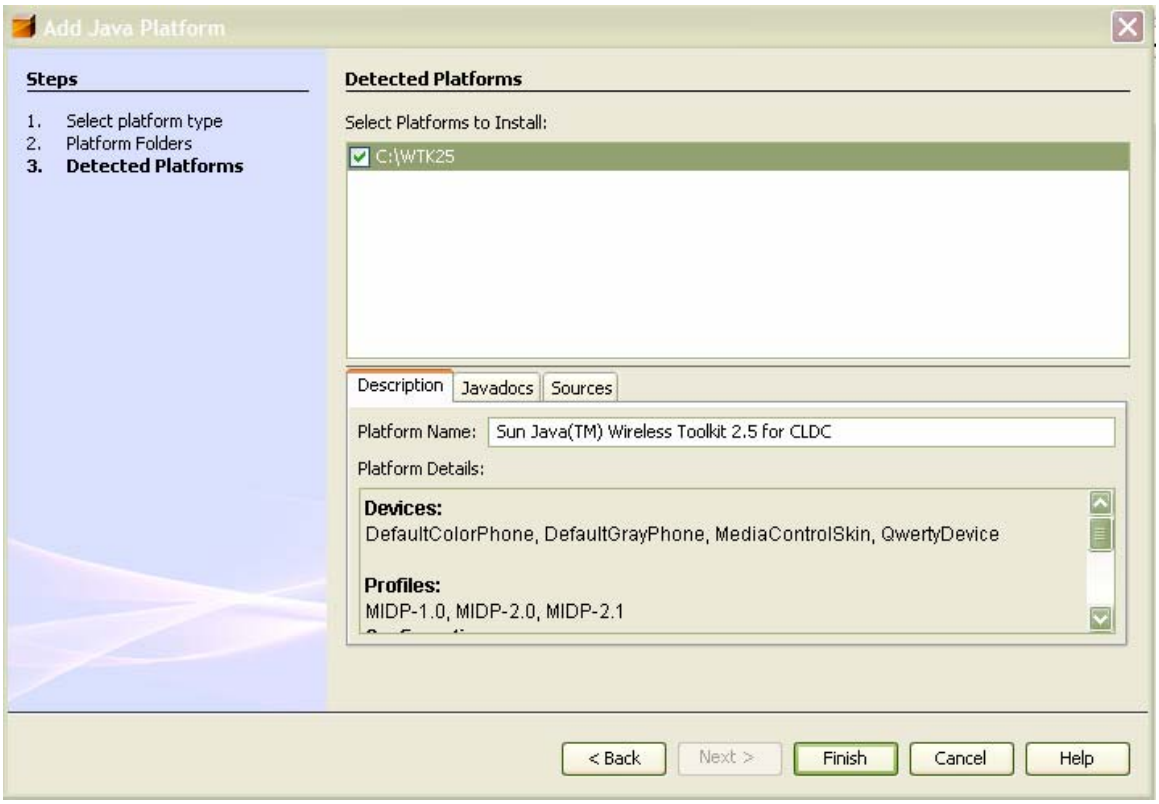
Pilih radio button Java Micro Edition Platform Emulator untuk menambahkan segala fasilitas yang ada dalam Sun Java Wireless Toolkit 2.5 kedalam Netbeans.



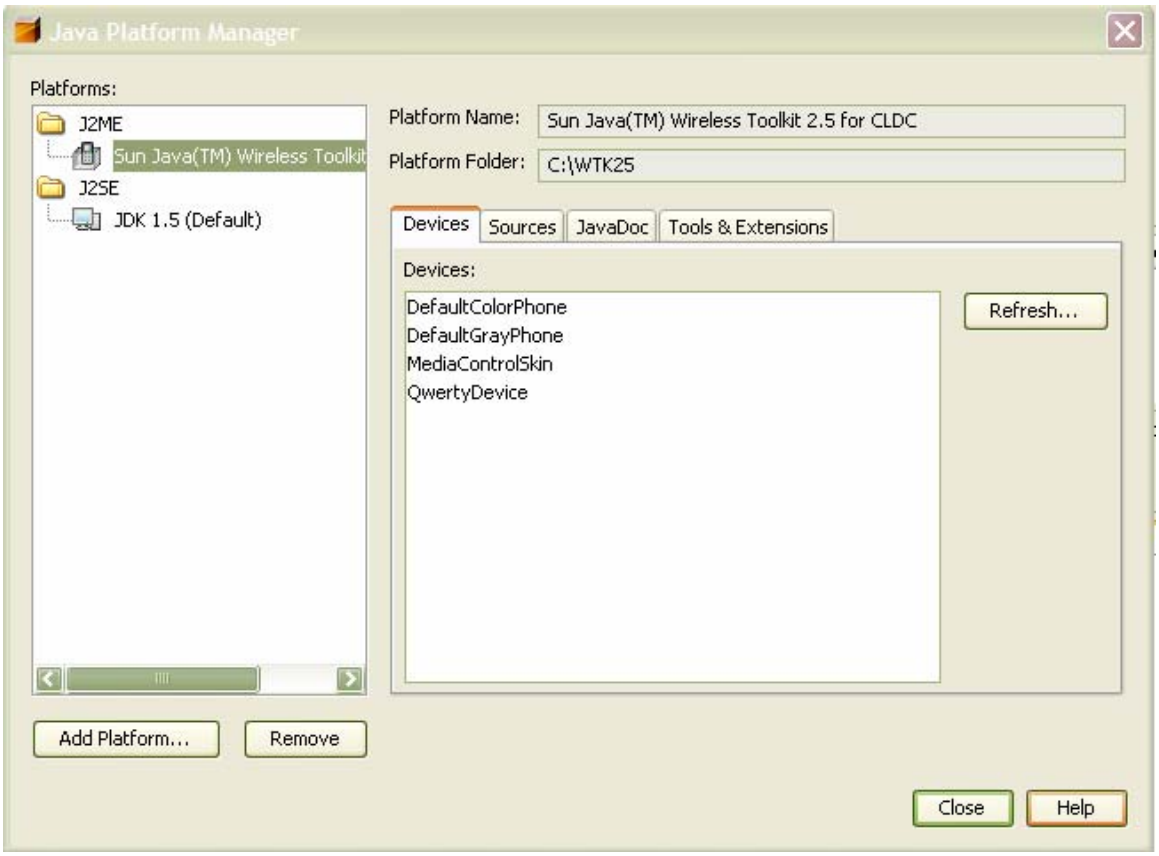
Apabila Anda telah meng-install Sun Java Wireless Toolkit 2.5, maka Anda tinggal memberikan tanda check pada saat pemilihan platform folders



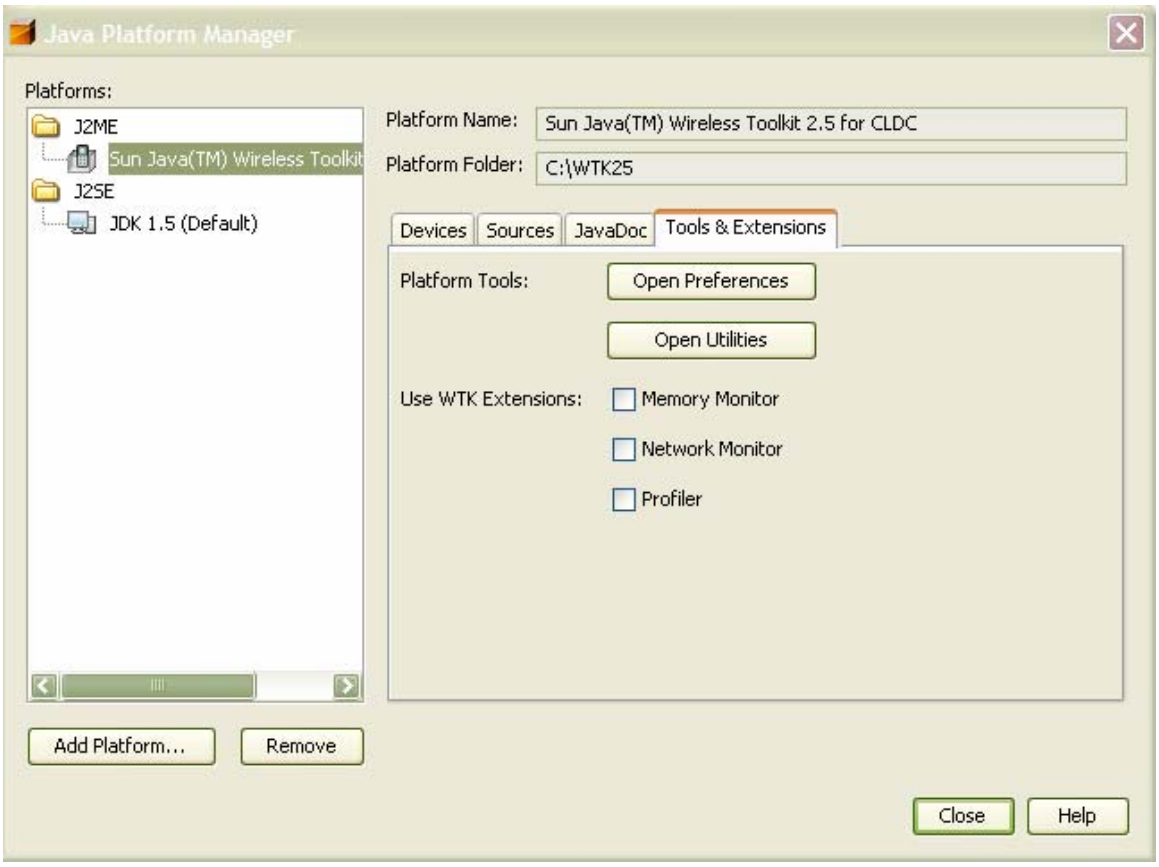
Secara otomatis Netbeans akan mendeteksi Platform WTK 2.5 tersebut. Klik “next” untuk memulai integrasi kedalam Netbeans



Setelah Anda mengintegrasikan WTK 2.5 maka akan muncul tampilan sbb, pada Java Platfor Manager Anda.

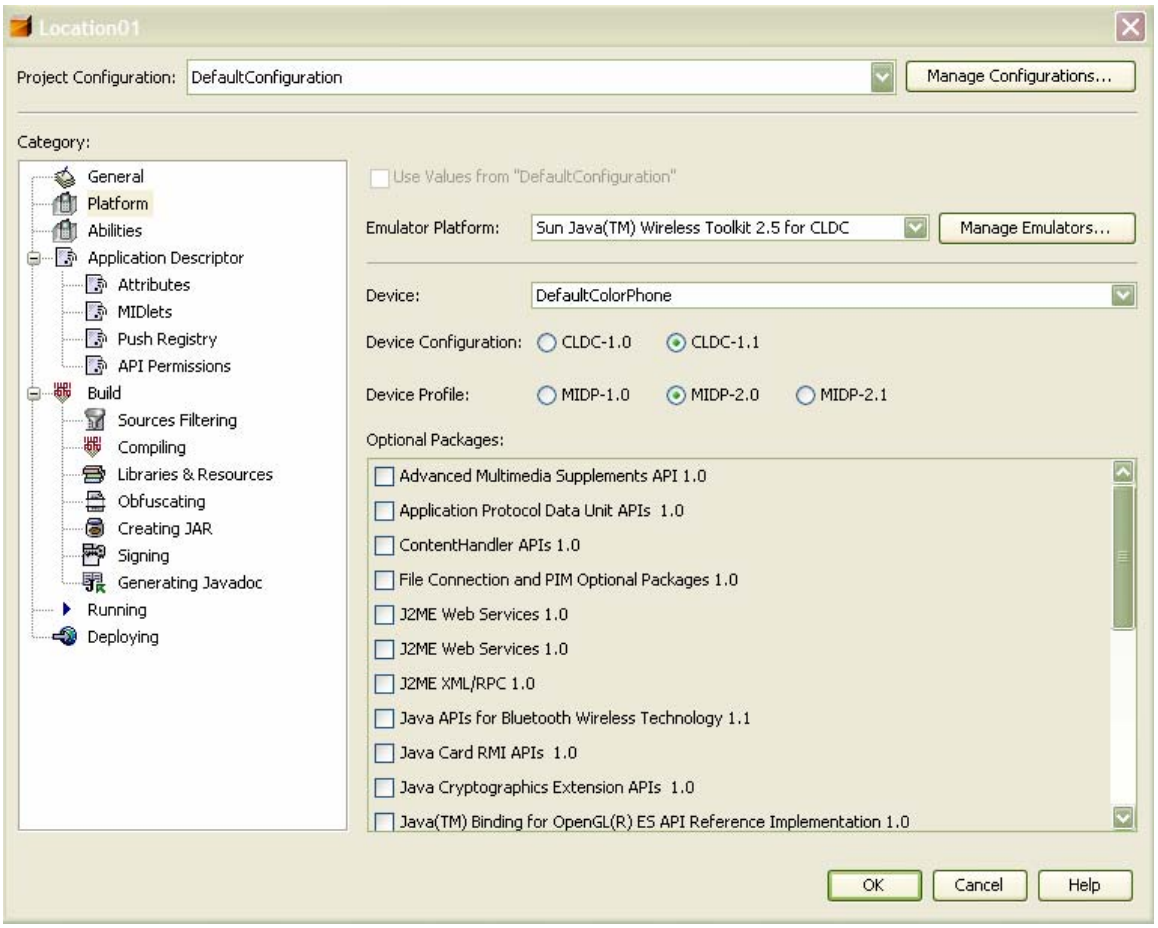


Pilih tab Tools dan Extensions untuk setting preferences dan utitlities dari suatu project.



2. Setting *Project Configuration*

Untuk mengubah konfigurasi pada project, maka pada tab projects, lakukanlah klik kanan, kemudian pilih properties dan dapatkan interface seperti dibawah ini:



Pada interface tersebut diatas, kita dapat memilih JSR (optional package API) yang dibutuhkan oleh masing-masing project. Misalnya: untuk komunikasi Bluetooth kita membutuhkan Java APIs for Bluetooth Wireless Technology, untuk Location API kita membutuhkan Location Based APIs 1.0.1

Percobaan 6 Koneksi Bluetooth :

Bluetooth Server

```
import javax.bluetooth.*;
import javax.microedition.lcdui.*;
import javax.microedition.io.*;
import java.io.*;

public class InfoServer implements Runnable {
    InputStream input;
    OutputStream output;
    StreamConnectionNotifier notifier;
    StreamConnection conn;
    LocalDevice localDevice;
 ServiceRecord serviceRecord;
public static String SERVICE_NAME = "chat";
    public static UUID PORT = new UUID(0x0518);

    private boolean isRunning = false;
    private static String URL = "btspp://localhost:" + PORT +
         ";name=" + SERVICE_NAME + ";authorize=true";
public InfoServer() {
     isRunning = false;
Thread thread = new Thread(this);
        thread.start();
 }
```

```
public void run() {
        if (!isRunning) {
            try {
                conn = null;
                localDevice = LocalDevice.getLocalDevice();
                localDevice.setDiscoverable(DiscoveryAgent.GIAC);

                notifier = (StreamConnectionNotifier)
Connector.open(URL);

            } catch (BluetoothStateException e) {
                System.err.println("Bluetooth Exception: " +
e.getMessage());
            } catch (IOException e) {
                System.err.println( "IO Exception: " + e.getMessage());
            }
            isRunning = true;
     }
while (true) {
            try {
                System.out.println("Waiting for connection...\n");

                // Menanti koneksi
             conn = notifier.acceptAndOpen();
```

```
   String msg = BluetoothMidlet.read(conn);
                    System.out.println("Received from Client: " + msg);

                    // Mengirimkan pesan balasan
                    msg = "InfoServer: Your command was: " + msg;
                    output = conn.openOutputStream();
                    output.write(msg.length());
                    output.write(msg.getBytes());
                    output.close();
                } catch (Exception ex) {
                    System.err.println("Bluetooth Server Exception: " + ex);
                }
            }
        }
}
```

Bluetooth Client

```
   import javax.bluetooth.*;
   import javax.microedition.io.*;
   import java.io.*;

class InfoClient implements DiscoveryListener {
    private DiscoveryAgent discoveryAgent;
    private RemoteDevice[] remoteDevices;
    private UUID[] UUIDSet;
    private String URL;
```

```
   public InfoClient() {
           try {
            LocalDevice localDevice = LocalDevice.getLocalDevice();
            discoveryAgent = localDevice.getDiscoveryAgent();
            discoveryAgent.startInquiry(DiscoveryAgent.GIAC, this);
           } catch (Exception e) {
               System.out.println(e);
           }
    }
   public void deviceDiscovered(RemoteDevice btDevice, DeviceClass cod) {
           try {

               // Dapatkan informasi mengenai device
               System.out.println("deviceDiscovered()");
               System.out.println("Address: " +
   btDevice.getBluetoothAddress());
               System.out.println("Major Device Class: " +
   cod.getMajorDeviceClass());
               System.out.println("Minor Device Class: " +
   cod.getMinorDeviceClass());
            System.out.println("Friendly Name: " +
btDevice.getFriendlyName(true));

               UUIDSet = new UUID[1];
               UUIDSet[0] = InfoServer.PORT;
               int searchID = discoveryAgent.searchServices(null, UUIDSet,
btDevice, this);
```

```
} catch (Exception e) {
            System.out.println("Exception: " + e);
        }
 }

public void servicesDiscovered(int transID, ServiceRecord[] servRecord)
{
        System.out.println("servicesDiscovered()");
        for (int i=0; i<servRecord.length; i++) {
            URL = servRecord[i].getConnectionURL(0, false);
        }
 }
public void serviceSearchCompleted(int transID, int responseCode) {
        switch (responseCode) {
            case SERVICE_SEARCH_COMPLETED:
                System.out.println("SERVICE_SEARCH_COMPLETED\n");
                System.out.println("Service URL: " + URL);
                StreamConnection conn = null;
                try {
                    String msg = "INFO";
                    conn = (StreamConnection)Connector.open(URL);
                    OutputStream output = conn.openOutputStream();
                    output.write(msg.length());
                    output.write(msg.getBytes());
                    output.close();
                    System.out.println(BluetoothMidlet.read(conn));
```

```
} catch (Exception ex) {
  System.out.println(ex);
} finally {
  try {
       conn.close();
  } catch (IOException ioe) {
       System.out.println("Error Closing connection " + ioe);
  }
}
break;
case SERVICE_SEARCH_ERROR:
               System.out.println("SERVICE_SEARCH_ERROR\n");
break;
            case SERVICE_SEARCH_TERMINATED:
                System.out.println("SERVICE_SEARCH_TERMINATED");
                break;
            case SERVICE_SEARCH_DEVICE_NOT_REACHABLE:

System.out.println("SERVICE_SEARCH_DEVICE_NOT_REACHABLE");
                break;
            case SERVICE_SEARCH_NO_RECORDS:
                System.out.println("SERVICE_SEARCH_NO_RECORDS");
                break;
            default: break;
        }
    }
```

```
    public void inquiryCompleted(int discType) {
            System.out.println("inquiryCompleted()");
        }
}
```

Bluetooth MIDlet

```
    import javax.microedition.midlet.*;
    import javax.microedition.lcdui.*;
    import javax.microedition.io.*;
    import javax.bluetooth.*;
    import java.io.*;

    public final class BluetoothMidlet extends MIDlet implements
    CommandListener {

     private final Command okCmd = new Command("Start", Command.OK, 1);
     private final Command exitCmd = new Command("Exit", Command.EXIT, 1);
     private static final String[] commands = { "Server", "Client" };
     private final List menu = new List("Bluetooth Application",
             List.IMPLICIT, commands, null);

    Display display;
     private InfoClient chatClient;
     private InfoServer chatServer;
```

```
public BluetoothMidlet() {
        menu.addCommand(exitCmd);
        menu.addCommand(okCmd);
        menu.setCommandListener(this);
    }
public void startApp() {
        display = Display.getDisplay(this);
        display.setCurrent(menu);
}
protected void destroyApp(boolean unconditional) {}
protected void pauseApp() {}
public void commandAction(Command c, Displayable d) {
        if (c == exitCmd) {
            destroyApp(true);
            notifyDestroyed();
            return;
        }
switch (menu.getSelectedIndex()) {
case 1: chatClient = new InfoClient(); break;
            default: break;
        };
}
```

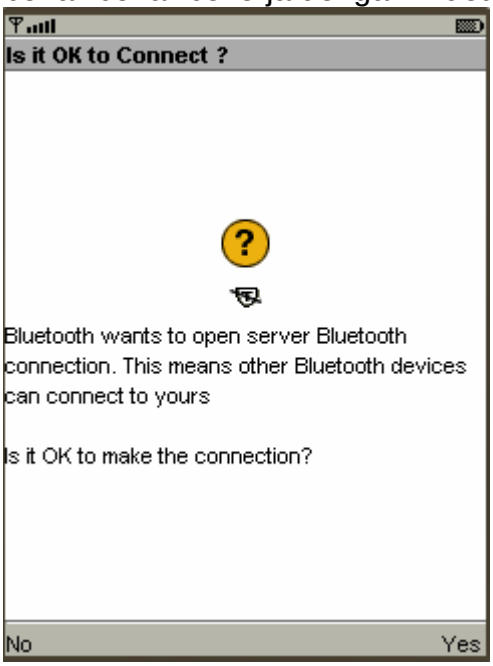
```
public final static String read(StreamConnection conn) {

        InputStream is = null;
        byte[] dataBytes = null;
        int len;

        try {
            is = conn.openInputStream();

            len = is.read();
            dataBytes = new byte[len];
            len = 0;

            while (len != dataBytes.length) {
                int readLen = is.read(dataBytes, len, dataBytes.length -
len);
                if (readLen == -1) {
                    System.err.println("Error reading data.");
                }
                len += readLen;
         }
} catch (IOException ex) {
         System.err.println(ex);
} finally {
         if (is != null) {
```

```
    try {
                        is.close();
                    } catch (IOException ex) { }
                }
            }
            return new String(dataBytes);
        }
}
```

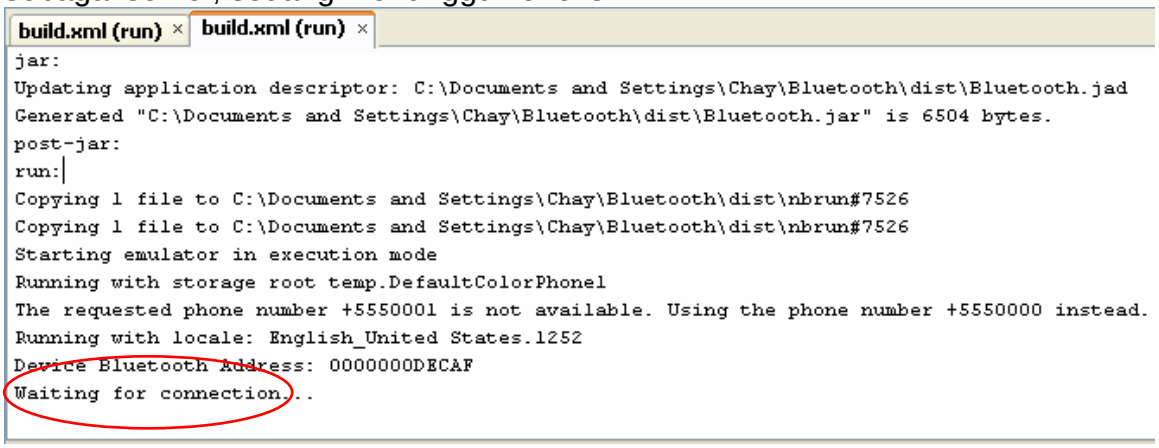
Untuk mendemonstrasikan bagaimana Simulator Bluetooth bekerja, Anda memerlukan dua running emulator yang bekerja bersamaan. Satu sebagai server (misal: +5550001) dan satu sebagai client (missal: +555000).

Pada saat server diaktifkan, akan muncul sebuah pesan untuk memastikan apakah user

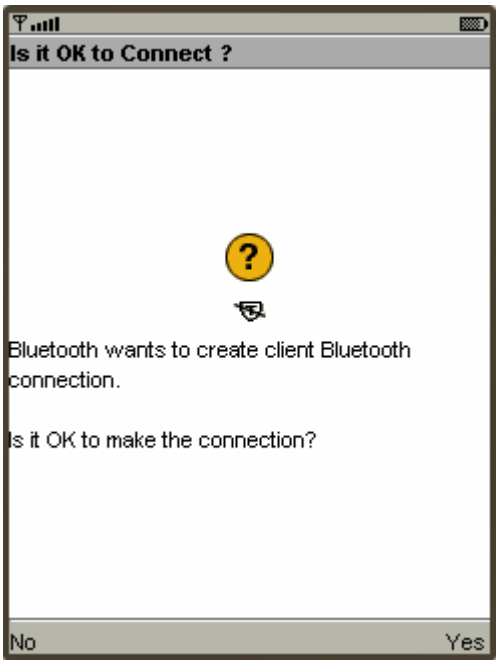
benar-benar bekerja dengan Bluetooth. Pilih command “yes” untuk mengaktifkan server.



Perhatikan console Anda, dan dapatkan pesan bahwa Bluetooth device yang bertindak sebagai server, sedang menunggu koneksi

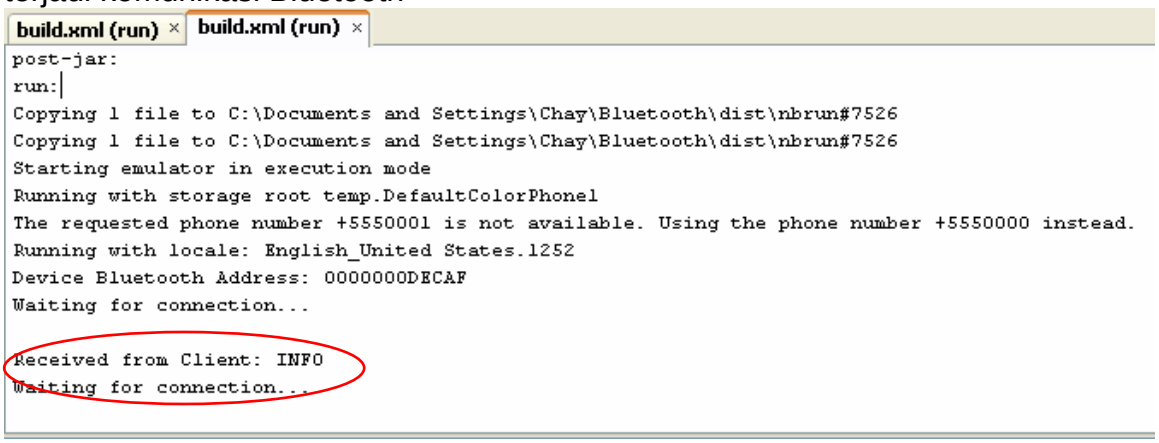


Perhatikan apa yang terjadi pada saat client Bluetooth diaktifkan. Sebuah pesan untuk komunikasi Bluetooth kembali tampil. Klik yes untuk membuat koneksi Bluetooth



Jaringan yang kita gunakan untuk berkomunikasi tidak dijamin keamanannya. Oleh karena itu, pada saat client memilih untuk melakukan komunikasi dengan server, maka akan ada sebuah pesan yang tampil pada server untuk meyakinkan server bahwa ia akan menerima sebuah pesan yang tidak ter-authorisasi dengan komunikasi Bluetooth

Perhatikan console Anda, sehingga tampil pesan berikut ini yang membuktikan bahwa telah terjadi komunikasi Bluetooth

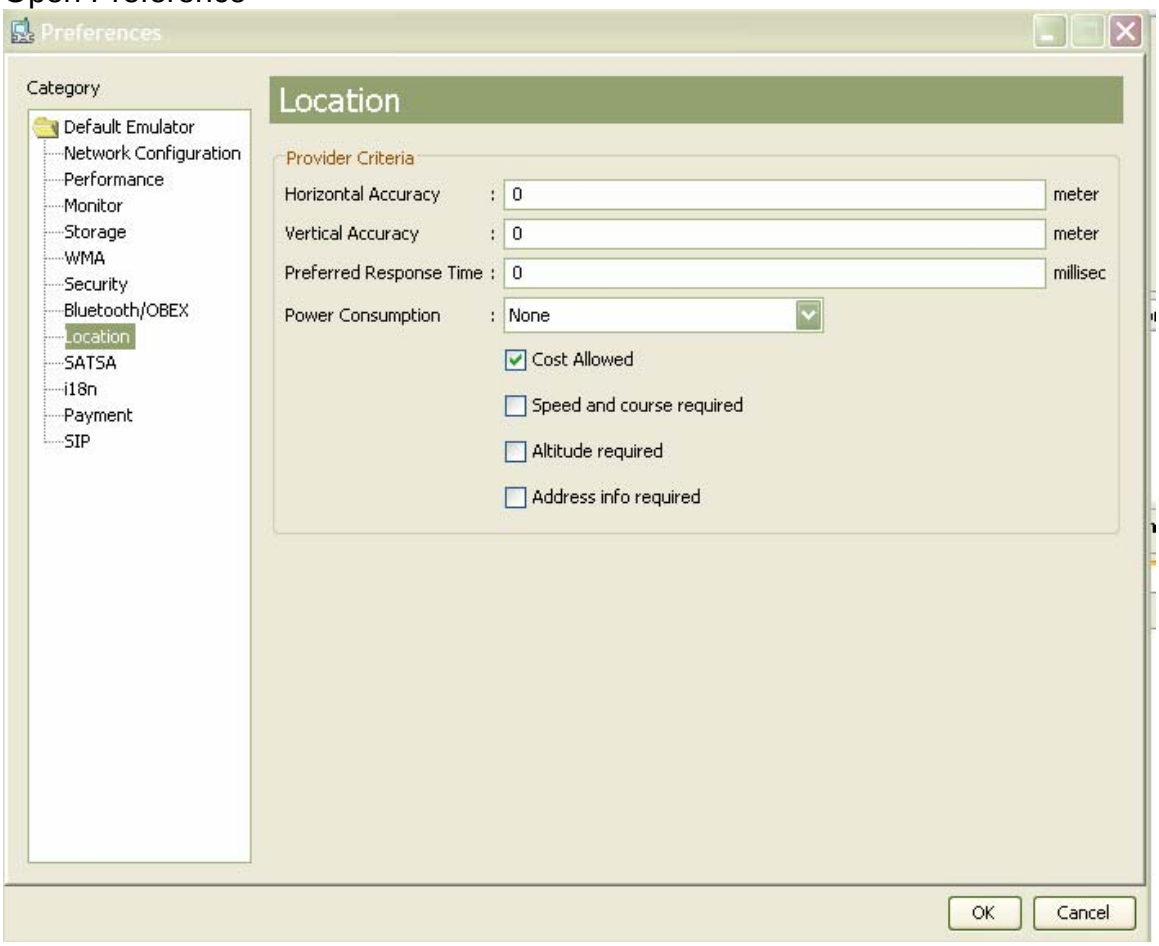


Percobaan 7 Location API :

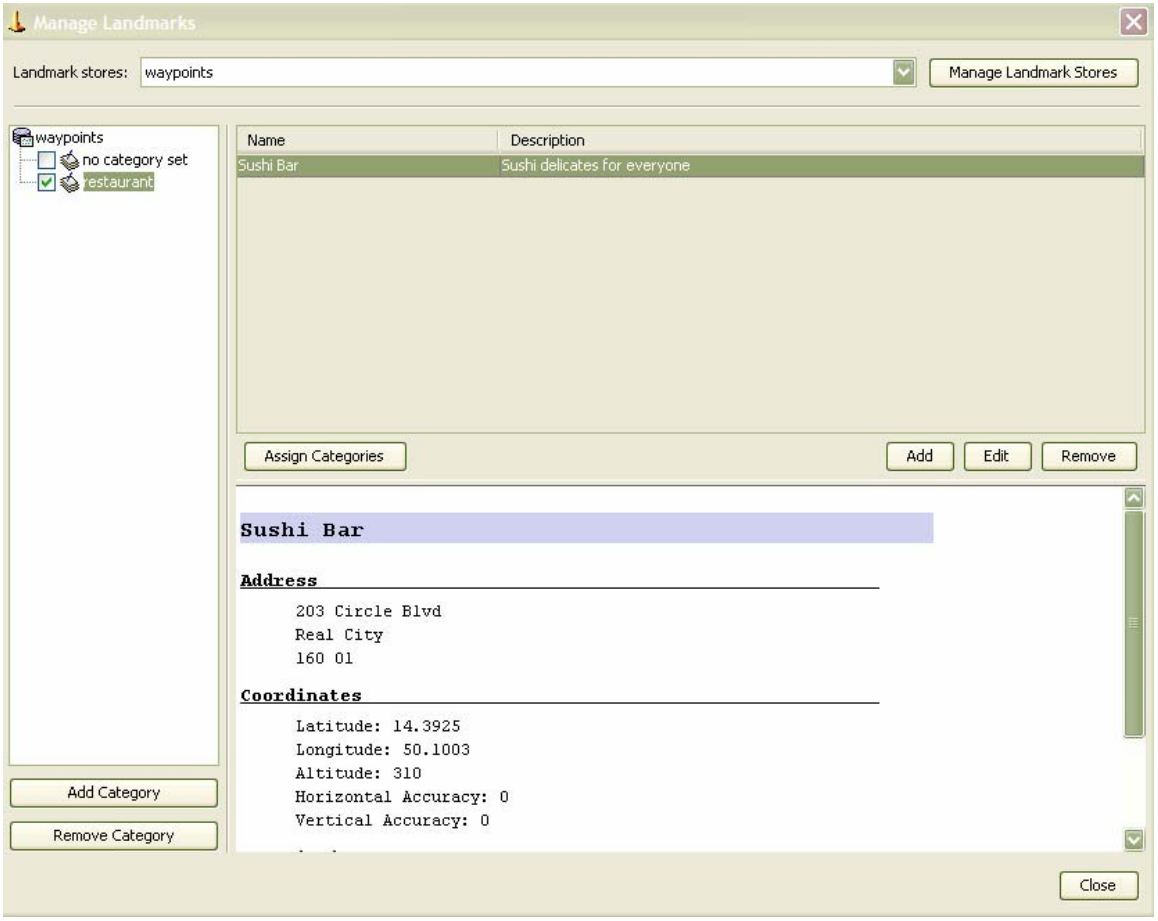
Hal-hal yang perlu disiapkan :

a. Ikuti langkah-langkah berikut ini untuk mendapatkan interface sbb:

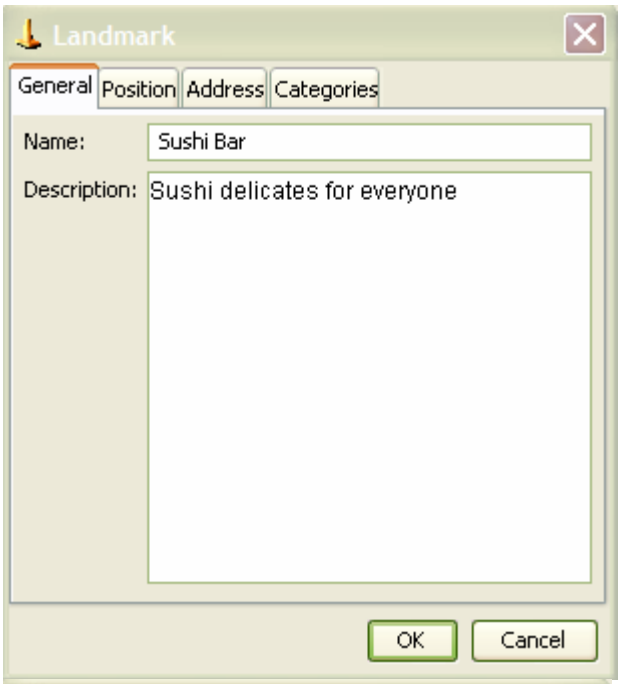
Tools -> Java Platform Manager -> Sun Java Wireless Toolkit 2.5 -> Tools and Extensions -> Open Preference

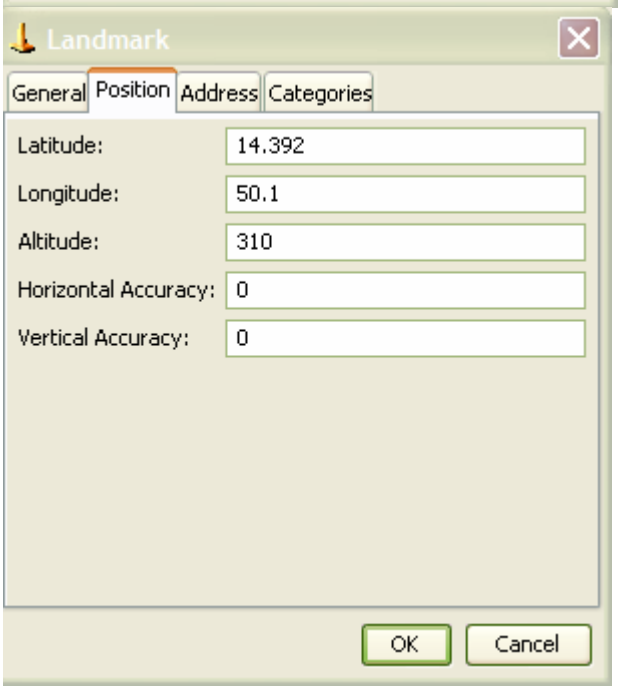


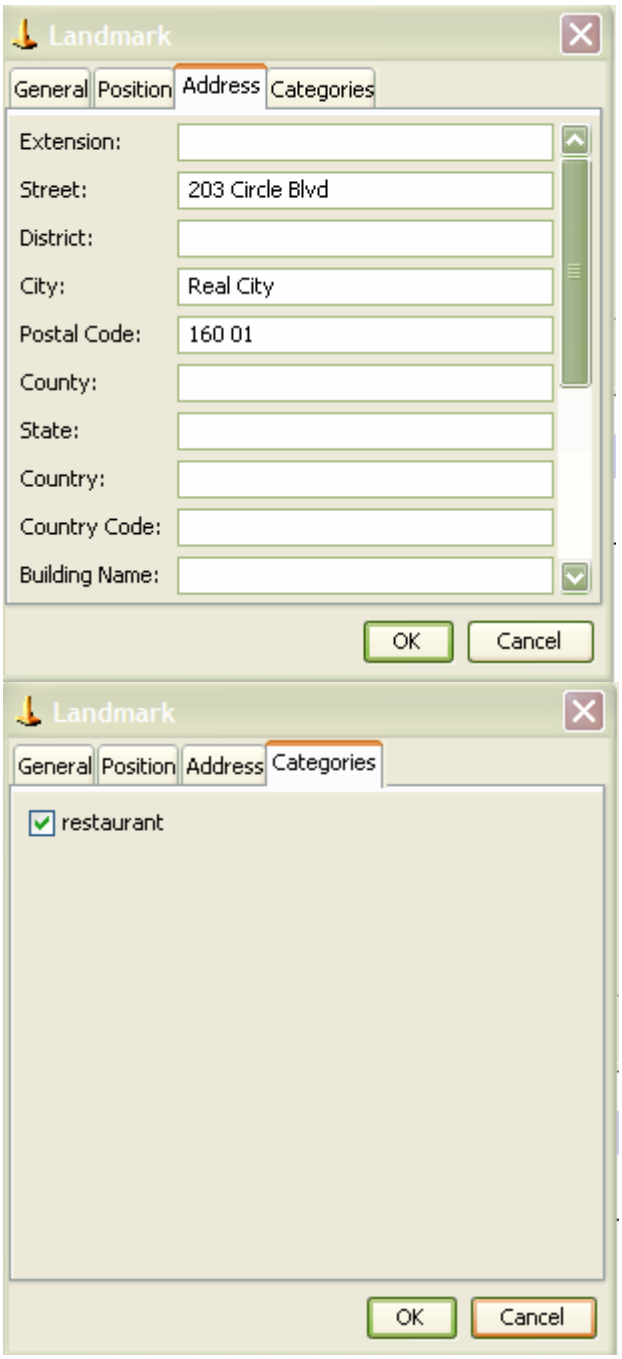
b. Pilih juga Open Utilities kemudian Manage *Landmark* untuk menambahkan sebuah category dari *landmark*



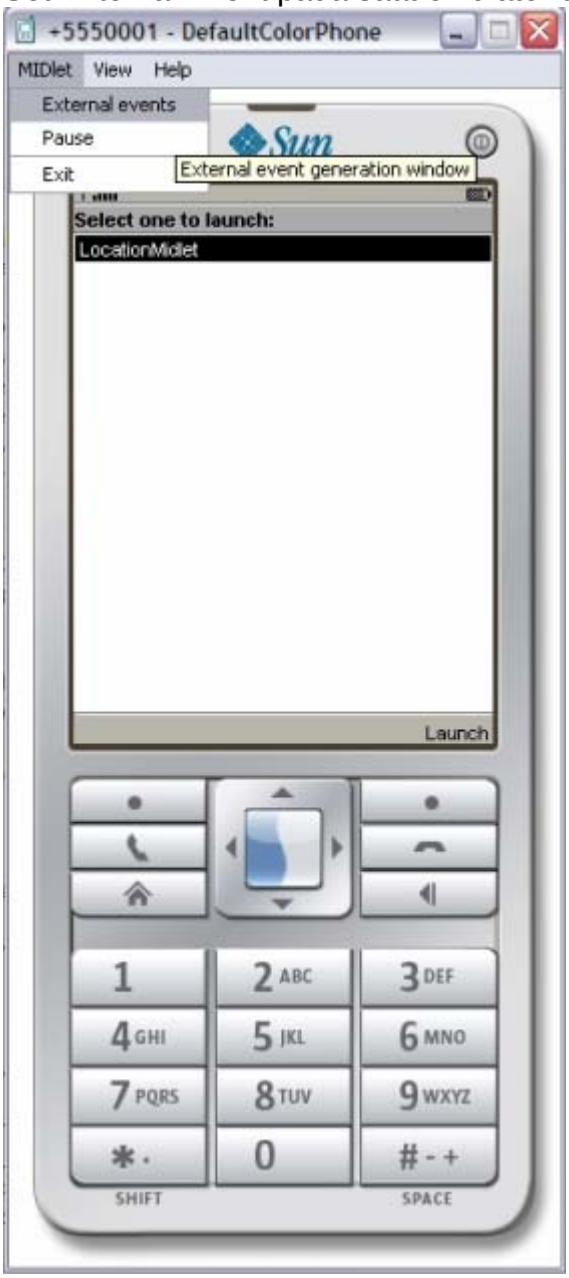
c. Setelah menambahkan kategori, tambahkan juga sebuah *landmark* beserta deskripsi-nya seperti nama, alamat, koordinat, dan juga kategori dari *landmark* tersebut.

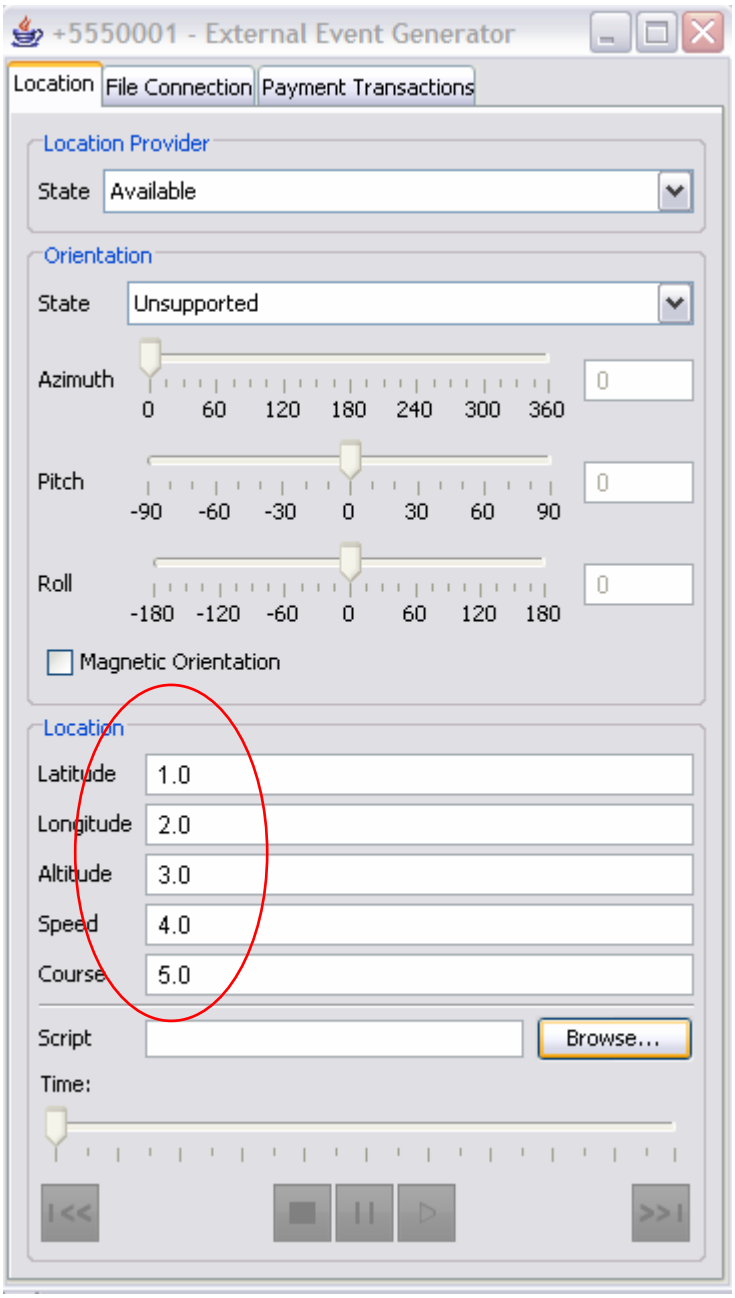

Landmark
General
Position
Address
Categories
Name:
Sushi Bar
Description:
Sushi delicates for everyone
OK
Cancel


Landmark
General
Position
Address
Categories
Latitude:
14.392
Longitude:
50.1
Altitude:
310
Horizontal Accuracy:
0
Vertical Accuracy:
0
OK
Cancel


Landmark
General
Position
Address
Categories
Extension:
Street:
203 Circle Blvd
District:
City:
Real City
Postal Code:
160 01
County:
State:
Country:
Country Code:
Building Name:
OK
Cancel
Landmark
General
Position
Address
Categories
restaurant
OK
Cancel

Set External Event pada saat emulator dijalankan:




+5550001 - External Event Generator
Location
File Connection
Payment Transactions
Location Provider
State
Available
Orientation
State
Unsupported
Azimuth
0 60 120 180 240 300 360
Pitch
-90 -60 -30 0 30 60 90
Roll
-180 -120 -60 0 60 120 180
Magnetic Orientation
Location
Latitude
1.0
Longitude
2.0
Altitude
3.0
Speed
4.0
Course
5.0
Script
Browse...
Time:


jeNi
CENTER
> > > Java Education Network Indonesia

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
import javax.microedition.location.*;

public class LocationMidlet extends MIDlet implements CommandListener,
Runnable {
    private final Command exitCmd = new Command("Exit", Command.EXIT,
1);
    private final Command locationCmd = new Command("Location",
Command.ITEM, 1);
    private Display display;
 private Form mainForm = new Form("JENI: Location Example");
private StringItem latitude = new StringItem("Latitude:", "",
Item.PLAIN);
    private StringItem longtitude = new StringItem("Longtitude:", "",
Item.PLAIN);
    private StringItem altitude = new StringItem("Altitude:", "",
Item.PLAIN);
    private StringItem speed = new StringItem("Speed:", "", Item.PLAIN);
    private StringItem course = new StringItem("Course:", "",
Item.PLAIN);
    private StringItem locMethod = new StringItem("Method:", "",
Item.PLAIN);
    private StringItem timestamp = new StringItem("Timestamp:", "",
Item.PLAIN);
    private StringItem status = new StringItem("Status:", "",
Item.PLAIN);
    private StringItem version = new StringItem("Version:", "Unknown",
Item.PLAIN);
 private LocationProvider locationProvider;
```

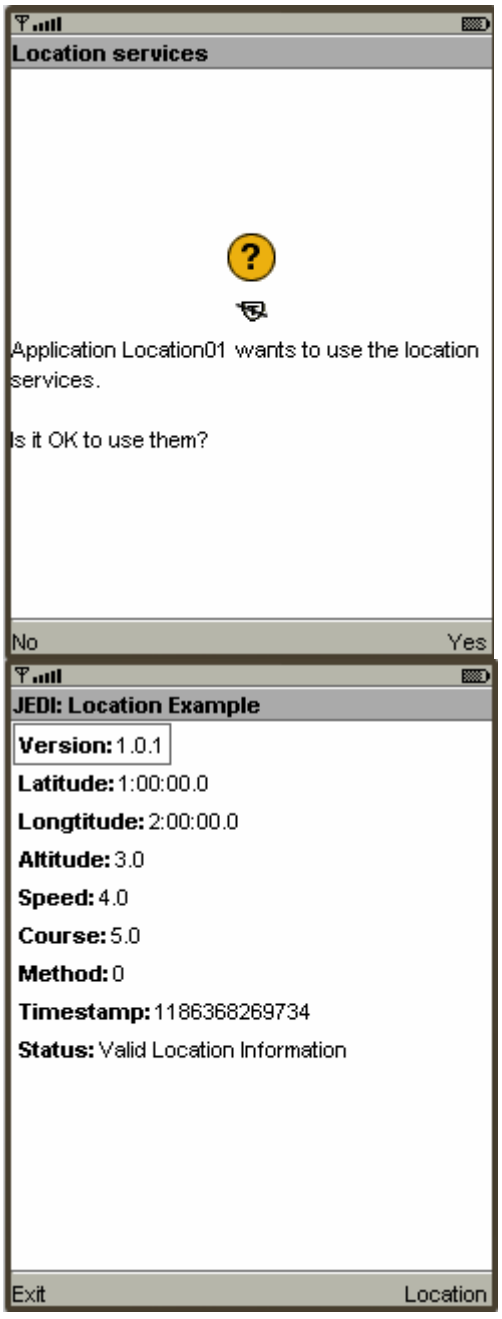
```
public LocationMidlet() {
        mainForm.addCommand(exitCmd);
        mainForm.addCommand(locationCmd);
        mainForm.setCommandListener(this);
   version.setText(System.getProperty("microedition.location.version"));
        mainForm.append(version);
        mainForm.append(latitude);
        mainForm.append(longtitude);
        mainForm.append(altitude);
        mainForm.append(speed);
        mainForm.append(course);
        mainForm.append(locMethod);
        mainForm.append(timestamp);
        mainForm.append(status);
 }
public void startApp() {
        display = Display.getDisplay(this);
        display.setCurrent(mainForm);
}
public void pauseApp() {}
public void destroyApp(boolean unconditional) {}
public void commandAction(Command c, Displayable d) {
        if (c == locationCmd) {
            Thread thread = new Thread(this);
            thread.start();
        }
```

```
if (c == exitCmd) {
            destroyApp(true);
            notifyDestroyed();
            return;
        }
 }
public void run() {
        try {
            Criteria cr = new Criteria();
            // Set titik horizontal sampai 1 km
            cr.setHorizontalAccuracy(0);
            cr.setVerticalAccuracy(0);
            LocationProvider provider =
LocationProvider.getInstance(cr);
            // Timeout setelah 5 detik
            Location loc = provider.getLocation(5);

            Coordinates coord = loc.getQualifiedCoordinates();
            if (coord != null) {
             if (loc.isValid()) {
latitude.setText(Coordinates.convert(coord.getLatitude(),
Coordinates.DD_MM_SS));

longtitude.setText(Coordinates.convert(coord.getLongitude(),
Coordinates.DD_MM_SS));
altitude.setText(Float.toString(coord.getAltitude()));
timestamp.setText(Long.toString(loc.getTimestamp()));
```

```
                        speed.setText(Float.toString(loc.getSpeed()));

                        locMethod.setText(Integer.toString(loc.getLocationMe
                        thod()));
                       course.setText(Float.toString(loc.getCourse()));

                        status.setText("Valid Location Information");
                    } else {
                        status.setText("Invalid Location Information");
                    }

                }
            } catch (LocationException e) {
                status.setText("Exception:" + e.getMessage());
            } catch (InterruptedException e) {
                status.setText("Timeout: " + e.getMessage());
            }
        }
}
```

Saat pertama kali Midlet dijalankan, akan tampil interface sbb:

Saat command *Location* dipilih, maka akan tampil sebuah pesan yang menyatakan bahwa Midlet akan menggunakan *location service* pada saat dijalankan :


Location services
Application Location01 wants to use the location services.
Is it OK to use them?
No
Yes
JEDI: Location Example
Version: 1.0.1
Latitude: 1:00:00.0
Longtitude: 2:00:00.0
Altitude: 3.0
Speed: 4.0
Course: 5.0
Method: 0
Timestamp: 1186368269734
Status: Valid Location Information
Exit
Location

Latihan :

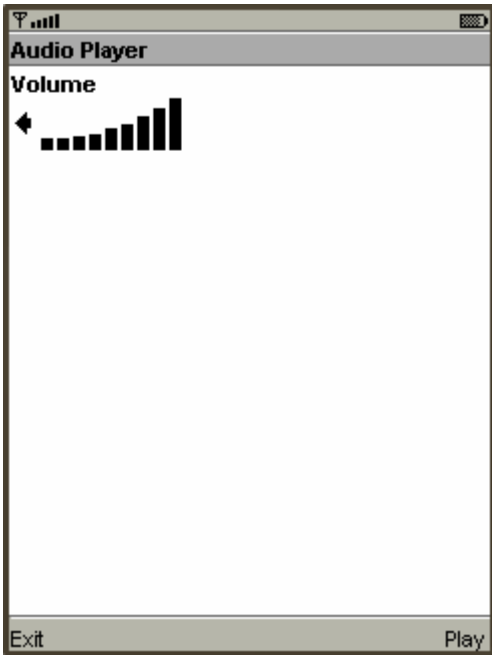
1. Audio Player

Buatlah sebuah MIDlet yang dapat memainkan file audio secara berulang-ulang sampai command “stop” dipilih dimana audio tersebut dibaca dari JAR file.

2. SMS Auto-Responder

Buatlah sebuah MIDlet yang secara otomatis akan me-reply apabila ia menerima sebuah text message. Petunjuk: Anda dapat memodifikasi SMSReceiverMidlet dan menggunakan koneksi yang sama untuk me-reply pesan.

Jawaban 1:



```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
import javax.microedition.media.*;
import javax.microedition.media.control.*;

import java.io.*;

public class Exercise1 extends MIDlet
        implements CommandListener{
  private Command exitCommand, playCommand, stopCommand;
    private Form form;
    private Gauge volumeGauge;
    private Display display;
    private int volume = 100;
    private static int MAX_VOLUME = 100;
   Player player;
    public Exercise1() {
        playCommand = new Command("Play", Command.OK, 1);
        stopCommand = new Command("Stop", Command.OK, 1);
        exitCommand = new Command("Exit", Command.EXIT, 1);
       volumeGauge = new Gauge("Volume", true, MAX_VOLUME, volume);
        form = new Form("Audio Player");
        form.addCommand(playCommand);
        form.addCommand(exitCommand);
        form.append(volumeGauge);
```

```
public void commandAction(Command c, Displayable d) {
        if (c == exitCommand) {
            notifyDestroyed();
        }
        if (c == playCommand){
            form.removeCommand(playCommand);
            form.addCommand(stopCommand);
            try {
                InputStream stream =
getClass().getResourceAsStream("bong.wav");
                player = Manager.createPlayer(stream,
"audio/x-wav");
                player.realize();


                VolumeControl control = (VolumeControl)
player.getControl("VolumeControl");
                if (control != null){
                    control.setLevel(volumeGauge.getValue());
                }
                player.setLoopCount(-1);


              player.start();
          } catch (MediaException mex) {
                display.setCurrent(new Alert("Media
Exception", mex.getMessage(),  null, AlertType.ERROR));
            } catch (Exception ex){
                display.setCurrent(new Alert("Exception",
ex.getMessage(),  null, AlertType.ERROR));
```

```
          try {
                player.stop();
            } catch (MediaException mex) {}
        }
    }
}
```

Jawaban 2:

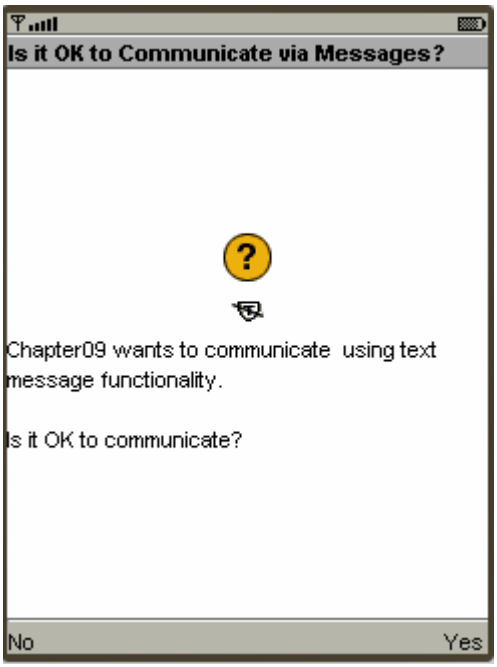
```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
import javax.microedition.io.*;
import javax.wireless.messaging.*;

public class Exercise2 extends MIDlet
        implements CommandListener, MessageListener, Runnable
{

    private Command exitCommand, sendCommand;
    private Form form;
    private StringItem statusField, addressField, mesgField,
dateField;
    private Display display;
    private MessageConnection conn;
    private Thread thread;
   private String port = "8888";
public Exercise2() {
        exitCommand = new Command("Exit", Command.EXIT, 1);
```

```
mesgField = new StringItem("Message:", "");
        dateField = new StringItem("Timestamp:", "");


       form = new Form("SMS Receiver");
form.append(statusField);
        form.append(addressField);
        form.append(mesgField);
        form.append(dateField);
        form.addCommand(exitCommand);
    }

    public void startApp() {
        display = Display.getDisplay(this);
        form.setCommandListener(this);

        startReceiver();
        display.setCurrent(form);
    }

    public void pauseApp() {
        thread = null;
   }
public void destroyApp(boolean unconditional) {
        thread = null;
        if (conn != null){
```

```
public void commandAction(Command c, Displayable d) {
        if (c == exitCommand) {
            notifyDestroyed();
        }
    }

    private void startReceiver(){
        try {
            String addr = "sms://:" + port;
            if (conn == null){
                conn = (MessageConnection)
Connector.open(addr);
                conn.setMessageListener(this);
              statusField.setText(
                          "waiting for message at port " +
port);
            }
        } catch (Exception ex){
            statusField.setText("Cannot open connection on
port "
                    + port + ":" + ex.getMessage());
        }
        thread = new Thread(this);
        thread.start();
   }
 public void notifyIncomingMessage(MessageConnection
messageConnection) {
        if (thread == null){
            thread = new Thread(this);
```

```
                TextMessage text = (TextMessage) mesg;
                addressField.setText(text.getAddress());
                mesgField.setText(text.getPayloadText());
                dateField.setText("" + text.getTimestamp());
                statusField.setText("Message received.");
                text.setPayloadText("Thank You.");
                conn.send(text);
            } else {
                statusField.setText("Non-text message
received: " + mesg.getClass().toString());

            }
        } catch (Exception e) {
            statusField.setText("Error: " + e.getMessage());
        }
        thread = null:
```

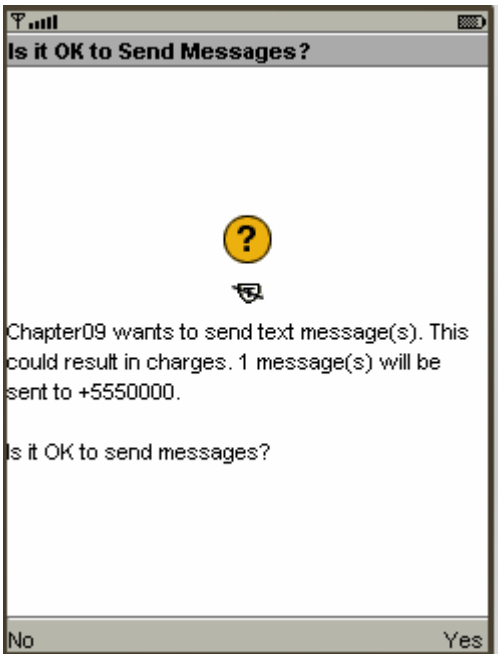
Klik yes pada interface berikut ini sehingga kita dapat mensimulasikan komunikasi SMS



Pada saat program dijalankan, maka ia akan menunggu sampai ada pesan yang dikirim kepada SMS Receiver yang ditunjukkan oleh interface berikut ini:

Pengiriman SMS akan dilaksanakan oleh WMA Console dengan memilih tombol Send SMS

Pada saat pesan dikirimkan kepada SMS Receiver maka akan tampil sebuah Alert seperti dibawah ini untuk memberitahukan kepada user bahwa ia juga secara otomatis akan mengirimkan pesan kepada WMA Console (+5550000)



Setelah pesan diterima oleh SMS Receiver (+5550001), maka akan tampil pesan seperti berikut ini:

Secara otomatis pesan yang dikirimkan oleh +555001 juga diterima oleh WMA Console, sehingga muncul pesan “thank you” pada layar.