1. Tujuan

Setelah mempelajari bab ini, Pelajar diharapkan mampu untuk :

- Memahami dan menggunakan Player
- Dapat menggunakan method-method dalam class player.
- Dapat membuat dan memainkan audio (*.wav dan *.midi)

2. Latar Belakang

Mobile Media API (MMAPI) memberikan dukungan lebih dari MIDlet dalam penggunaan media seperti audio dan video. MMAPI sangat digunakan untuk pembuatan suara dalam sebuah game yang akan menambah daya tarik sebuah game selain grafik.

3. Percobaan

Percobaan 1: Membuat class MidletPlaywav

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

public class MidletPlayWav extends MIDlet implements PlayerListener,
   CommandListener {

   private Display display;
   private static Player player;
   private VolumeControl vc;
   private List list;
   private Command cmExit, cmStop;

   public MidletPlayWav() {
      display = Display.getDisplay(this);
   }
   public void startApp() {
```

```
      try {
         playMedia("wav.wav", "wav");
      } catch (Exception e) {
         e.printStackTrace();
      }
   }

   public void pauseApp() { }

   public void destroyApp(boolean unconditional) { }

   public void commandAction(Command c, Displayable s){}

   public void exitMIDlet() {
      destroyApp(false);
      notifyDestroyed();
   }

   private void playMedia(String file, String p) throws Exception {
      player = Manager.createPlayer(getClass().getResourceAsStream(file),
               "audio/x-wav");
      player.addPlayerListener(this);
      player.setLoopCount(-1);
      player.prefetch();
      player.realize();

      vc = (VolumeControl)player.getControl("VolumeControl");
      if (vc != null) {
         vc.setLevel(100);
      }
      player.start();
   }

   public void playerUpdate(Player player, String event, Object eventData){ }
}
```

Hasil :

Mengeluarkan suara dengan extension wav

Percobaan 2 : Membuat MidletPlayMidi

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

public class MidletPlayMidi extends MIDlet implements PlayerListener,
   CommandListener {

   private Display display;
   private static Player player;
   private VolumeControl vc;
   private List list;
   private Command cmExit, cmStop;

   public MidletPlayMidi() {
      display = Display.getDisplay(this);
   }
   public void startApp() {
      try {
         playMedia("mid.mid", "mid");
      } catch (Exception e) {
         e.printStackTrace();
```

```
      }
   }

   public void pauseApp() { }

   public void destroyApp(boolean unconditional) { }

   public void commandAction(Command c, Displayable s){ }

   public void exitMIDlet() {
      destroyApp(false);
      notifyDestroyed();
   }

   private void playMedia(String file, String p) throws Exception {
      player = Manager.createPlayer(getClass().getResourceAsStream(file),
               "audio/midi");
      player.addPlayerListener(this);
      player.setLoopCount(-1);
      player.prefetch();
      player.realize();

      vc = (VolumeControl)player.getControl("VolumeControl");
      if (vc != null) {
         vc.setLevel(100);
      }
      player.start();
   }

   public void playerUpdate(Player player, String event, Object eventData){ }
}
```

Hasil :

Pilih midi mengeluarkan suara dengan extension midi

Percobaan 3 : Membuat class MidletPlayWavMidi

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

public class MidletPlayWavMidi extends MIDlet implements PlayerListener,
   CommandListener {

   private Display display;
   private static Player player;
   private VolumeControl vc;
   private List list;
   private Command cmExit, cmStop;

   public MidletPlayWavMidi() {
      display = Display.getDisplay(this);
   }

   public void startApp() {
      cmExit = new Command("Keluar", Command.EXIT, 1);
      cmStop = new Command("Stop", Command.OK, 1);
      list = new List("Menu", List.IMPLICIT);
      list.append("wav", null);
```

```
        list.append("midi", null);
        list.addCommand(cmExit);
        list.setCommandListener(this);
        display.setCurrent(list);
    }

    public void pauseApp() { }

    public void destroyApp(boolean unconditional) { }

    public void commandAction(Command c, Displayable s){
        if (c == List.SELECT_COMMAND){
            switch (list.getSelectedIndex()){
                case 0:
                        list.addCommand(cmStop);
                        try {
                                playMedia("wav.wav", "wav");
                        } catch (Exception e) {
                                e.printStackTrace();
                        }
                        break;
                case 1:
                        list.addCommand(cmStop);
                        try {
                                playMedia("mid.mid", "mid");
                        } catch (Exception e) {
                                e.printStackTrace();
                        }
                        break;
            }
        }else if(c == cmExit){
            exitMIDlet();
        }else if(c == cmStop){
            try{
                player.stop();
                if(player != null){
                        player.close();
                }
            }catch(Exception e){ }
            list.removeCommand(cmStop);
        }
```

```
    }

    public void exitMIDlet() {
        destroyApp(false);
        notifyDestroyed();
        player = null;
    }

    private void playMedia(String file, String p) throws Exception {
        if(p.equals("wav")){
            player = Manager.createPlayer(getClass().getResourceAsStream(file),
                     "audio/x-wav");
        }else if(p.equals("mid")){
            player = Manager.createPlayer(getClass().getResourceAsStream(file),
                     "audio/midi");
        }

        player.addPlayerListener(this);
        player.setLoopCount(-1);
        player.prefetch();
        player.realize();

        vc = (VolumeControl)player.getControl("VolumeControl");
        if (vc != null) {
            vc.setLevel(100);
        }
        player.start();
    }

    public void playerUpdate(Player player, String event, Object eventData){ }
}
```

Hasil :

Pilih wav mengeluarkan suara dengan extension wav
Pilih midi mengeluarkan suara dengan extension midi