Record Management System

Muhammad Zen S. Hadi, ST. MSc.

Penyimpanan Data

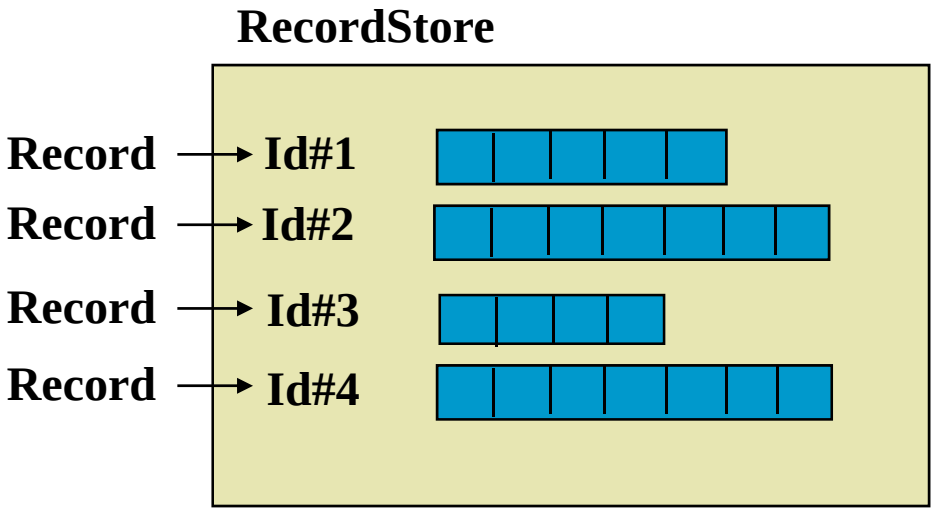
- MIDlet memerlukan penyimpanan data secara permanen.
- Perangkat mobile cukup terbatas fasilitasnya.
- Tidak ada filesystem atau relasi database dalam MIDP-based environment.
- MIDP menyediakan paket javax.microedition.rms untuk penyimpanan data.
- RMS menyediakan database berbasis record yang sederhana.

Record

- RMS adalah sistem untuk mengatur record.
- Record adalah item data tunggal.
- Tidak ada tipe data. Record dinyatakan dalam array of bytes.
- Record dapat terdiri dari bilangan, string, array, gambar – segala sesuatu yg dapat dinyatakan dalam urutan byte.

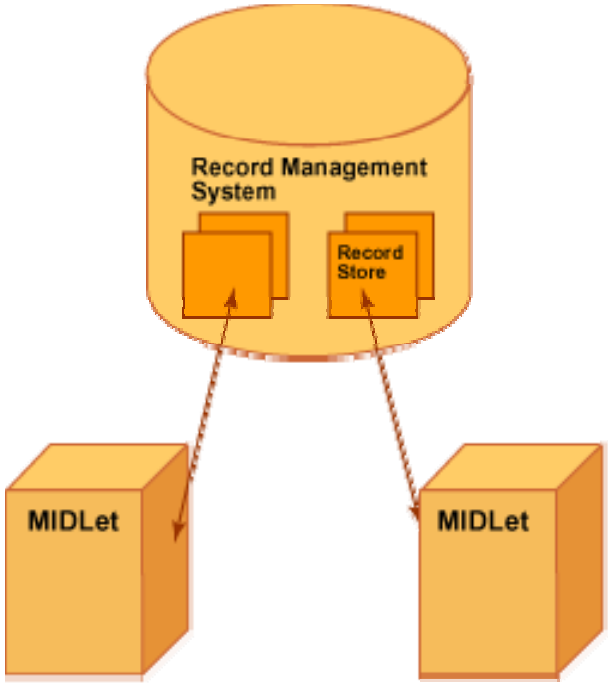
Letak Field

- Dalam RMS record tidak mempunyai field.
- Record terdiri dari array dari field tunggal yang diidentifikasi oleh recordId.
- Hal ini menjaga RMS tetap kecil dan fleksibe.



Record Stores

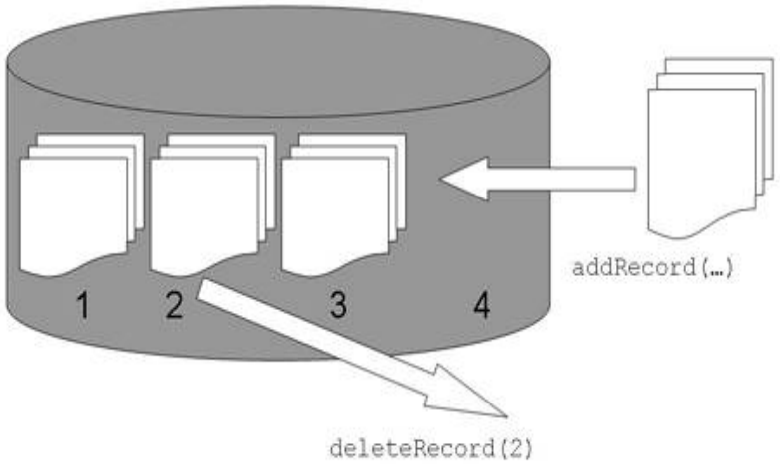
- Record store adalah sekumpulan record yang tersusun secara rapi.
- Masing-masing record milik dari record store.
- Record store akan memastikan bahwa record dapat dibaca dan ditulisi dan tanpa ada data yg hilang.



J2ME Persistent Storage

Record Stores

- Ketika record dibuat, record store membuat identifikasi unik berupa bil. integer yang disebut recordID.



- Record pertama ditambahkan ke record store akan mempunyai record ID 1, 2, dst.

Mengatur Record Stores

- Untuk membuka record store, gunakan method openRecordStore
  - public static RecordStore openRecordStore(
    String recordStoreName,
    boolean createIfNecessary)
    throws RecordStoreException, RecordStoreFullException,
    RecordStoreNotFoundException
- Jika tidak ditemukan maka akan menghasilkan : RecordStoreNotFoundException
- Untuk membuka record store dengan nama “Alamat”

```
RecordStore rs = RecordStore.openRecordStore("Alamat", true);
```

- Jika record store blm ada, maka akan dibuat dulu.

Mengatur Record Stores

- closeRecordStore() method menutup record store yang terbuka.

```
rs.closeRecordStore();
```

- Untuk menghapus sebuah record store yang berisi record, gunakan method deleteRecordStore()

```
RecordStore.deleteRecordStore(“Alamat");
```

Menambah Records

- MIDlet menggunakan method addRecord() dari class RecordStore untuk menambah sebuah record baru ke record store.
  - public int addRecord(byte[] data, int offset, int numBytes)

    menambah record yang dinyatakan dalam array of bytes data dengan offset adalah start index dan numBytes adalah panjangnya.

```
String appt = "new record";
byte bytes[] = appt.getBytes();
int recID = rs.addRecord(bytes,0,bytes.length);
```

Mengambil Records

- Ada 2 metode untuk mengambil record :

  public int getRecord(int recordId, byte[] buffer, int offset)

  – Mengkopi data yang tersimpan dalam record ke byte array yang dinyatakan dalam buffer.

  public byte[] getRecord(int recordId)

  – Mengembalikan copy data yang baru dari data yang dinyatakan oleh recordID.

```
byte[] retrieved = new byte[rs.getRecordSize(recID)];
rs.getRecord(id, retrieved, 0);
String retrievedString = new String(retrieved);
```

```
byte[] retrieved = rs.getRecord(recID);
String retrievedString = new String(retrieved);
```

Update Records

- Untuk mengupdate record gunakan method setRecord:

```
public void setRecord(int recordId,
                      byte[] newData, int offset, int numBytes)
```

  – Set informasi baru, data baru dengan offset sebagai start index, numBytes sebagai panjangnya dan lokasi record ditentukan lewat recordID.

```
String newappt = "update record";
byte data[] = newappt.getBytes();
rs.setRecord(recID, data, 0, data.length());
```

Deleting Records

- MIDlet menggunakan deleteRecord() untuk menghapus record dari recordstore.

  public void deleteRecord(int recordId)

  – Menghapus record yang dinyatakan dalam recordID. RecordID tidak dapat digunakan lagi.

```
rs.deleteRecord(1);
```

Contoh Aplikasi (PhoneBook)

- Aplikasi phone book untuk menambah data phone dan menampilkan semua hasilnya.

Deklarasi variabel

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

public class cgRMS extends MIDlet implements CommandListener {
  private Display display;
  private Form form, ftampil;
  private RecordStore rs;
  private RecordEnumeration re;
  private ChoiceGroup choicegroup;
  private Alert alert;
  //untuk proses entri data
  private Form entri;
  private TextField tfNama, tfNoTelp;

  private Command cmdKeluar = new Command("Keluar", Command.EXIT, 1);
  private Command cmdPilih =  new Command("Pilih", Command.OK, 1);
  private Command cmdSimpan = new Command("Simpan", Command.OK, 1);
  private Command cmdKembali = new Command("Kembali", Command.BACK, 1);
```

Open RMS dan konstruktor

```
public cgRMS() {
  display = Display.getDisplay(this);
  alert = new Alert("Info DB");

  rs = null;
  // membuat atau membuka record store
  try {
    rs = RecordStore.openRecordStore("phoneDB", true);
  } catch (RecordStoreException rse) {
    alert.setString("Record store tidak dapat dibuka. " +
                    "Aplikasi akan dihentikan");
    alert.setType(AlertType.ERROR);
    display.setCurrent(alert, null);
    System.exit(1); //keluar dari aplikasi
  }
}
```

Tampilan awal

```
public void startApp() {
  form = new Form("Aplikasi PhoneBook RMS");

  choicegroup = new ChoiceGroup("Menu:", Choice.EXCLUSIVE);
  choicegroup.append("Tambah Record", null);
  choicegroup.append("Daftar Record", null);

  form.append(choicegroup);
  form.addCommand(cmdKeluar);
  form.addCommand(cmdPilih);
  form.setCommandListener(this);
  display.setCurrent(form);
}

public void pauseApp() {
}

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

Pengaturan menu

```
public void commandAction(Command c, Displayable s) {
  if (c == cmdKeluar) {
    destroyApp(false);
    notifyDestroyed();
  } else if (c == cmdPilih) {
    if (choicegroup.isSelected(0))
        entriData();
    else if (choicegroup.isSelected(1))
        lihatRecord();
  } else if (c == cmdKembali) {
    display.setCurrent(form);
  } else if (c == cmdSimpan) {
    alert.setType(AlertType.INFO);
    tambahRecord(tfNama.getString(), tfNoTelp.getString());
    alert.setString("Data baru telah berhasil disimpan");
    display.setCurrent(alert, form);
  }
}
```

Entry Data

```
public void entriData() {
  entri = new Form("Entri Data");
  tfNama = new TextField("Nama:", null, 25, TextField.ANY);
  tfNoTelp = new TextField("No. Telepon:", null, 15, TextField.PHONENUMBER);
  entri.append(tfNama);
  entri.append(tfNoTelp);
  entri.addCommand(cmdSimpan);
  entri.addCommand(cmdKembali);
  entri.setCommandListener(this);
  display.setCurrent(entri);
}
```

Proses Tambah Data

```
public void tambahRecord(String nama, String noTelp) {
  byte[] temp = null;
  try {
    ByteArrayOutputStream baos = new ByteArrayOutputStream();
    DataOutputStream dos = new DataOutputStream(baos);
    dos.writeUTF(nama);
    dos.writeUTF(noTelp);
    temp = baos.toByteArray();
  } catch (IOException ioe) {
    ioe.printStackTrace();
  }
  try {
    rs.addRecord(temp, 0, temp.length);
  } catch (RecordStoreNotOpenException rsnoe) {
    rsnoe.printStackTrace();
  } catch (RecordStoreException rse) {
    rse.printStackTrace();
  }
}
```

Tampil Data

```
public void lihatRecord() {
  byte[] temp = null;
  ftampil = new Form ("Daftar Record");
  try {
    re = rs.enumerateRecords(null, null, false);
    while (re.hasNextElement()) {
      int i = re.nextRecordId();
      temp = rs.getRecord(i);
      ByteArrayInputStream bais = new ByteArrayInputStream(temp);
      DataInputStream dis = new DataInputStream(bais);
      try {
        String nama = dis.readUTF();
        String noTelp = dis.readUTF();
        ftampil.append(nama + " [" + noTelp + "]"+"\n");
      } catch (IOException ioe) {
        ioe.printStackTrace();
      }
    }
    ftampil.addCommand(cmdKembali);
    ftampil.setCommandListener(this);
    display.setCurrent(ftampil);
  } catch (InvalidRecordIDException invID) {invID.printStackTrace();
  } catch (RecordStoreNotOpenException rsnoe) {rsnoe.printStackTrace();
  } catch (RecordStoreException rse) {rse.printStackTrace();
  }
 }  //end lihatRecord
}  //end program
```