

Membuat Aplikasi Java dan Object Database sederhana dengan Database db4o

Mukhamad Nurkamid

nurkamid@umk.ac.id

Object-oriented database adalah salah satu jenis database dimana data direpresentasikan dalam bentuk object. Object Oriented Database merupakan sebuah sistem data base yang menggabungkan semua konsep penting dari object oriented.

Pendekatan ini sangat dipengaruhi oleh bahasa pemrograman object-oriented dan dapat dipahami sebagai usaha untuk menambah fungsionalitas DBMS pada lingkup bahasa pemrograman.

Pada aplikasi ini, kita akan mencoba mendemonstrasikan bagaimana **Menyimpan, me-retrieve—meng-Query, meng-update dan menghapus instance** dari satu single class yang hanya terdiri dari anggota primitif dan String. Pada contoh kali ini, kita akan membuat aplikasi Data **Mahasiswa** di mana atributnya antara lain, **nama**, **nim** dan **ipk**.

Pertama-tama kita membuat suatu class untuk menangani data kita.

```
package prjmahasiswa;

/**
 * Mahasiswa.java
 * Created on 4 Desember 2010 *
 * @author kamid
 */
public class Mahasiswa {
    private String Nama;
    private String Nim;
    private float Ipk;

    /** Creates a new instance of Mahasiswa */
    public Mahasiswa(String Nama, String Nim, float Ipk) {
        this.Nama=Nama;
        this.Nim = Nim;
        this.Ipk = Ipk;
    }
    public String getNama()
    {
        return Nama;
    }
    public String getNim()
    {
        return Nim;
    }
    public float getIpk()
    {
        return Ipk;
    }
}
```

```

    public void setIpk(float ipk)
    {
        this.Ipk = ipk;
    }
    public String toString()
    {
        return Nama+"/"+Nim+"/"+Ipk;
    }
}

```

Membuka Database

Untuk mengakses atau membuat baru suatu file database db4o, digunakan method **Db4o.openFile()** , yang menyediakan path file database sebagai parameternya untuk memperoleh instance **ObjectContainer**.

ObjectContainer merepresentasikan database, dan akan menjadi interface utama pada db4o. untuk menutup file database digunakan method **#.close()**.

```

ObjectContainer db=Db4o.openFile(path_file_object_database);
try {
    // do something with db4o

    ---Menambah object---
    ---Menghapus object---
    ---Mengupdate object---
    ---Menampilkan semua object---
    ---Me-retrieve object berdasar criteria---

}
finally {
    db.close();
}

```

Menyimpan Object

Untuk menyimpan suatu object ke dalam database db4o, digunakan perintah **set()** dengan object yang tersimpan sebagai parameternya.

```

// Simpan data mahasiswa pertama
Mahasiswa mahasiswa1=new Mahasiswa("Andi","383008", Float.valueOf("3.25"));
db.set(mahasiswa1);
System.out.println("Stored "+mahasiswa1);

```

Output:

Stored Andi/383008/3.25

```

// Simpan data mahasiswa kedua
Mahasiswa mahasiswa1=new Mahasiswa("Budi","379008", Float.valueOf("3.67"));
db.set(mahasiswa1);
System.out.println("Stored "+mahasiswa1);

```

Output:

Stored Budi/379008/3.67

Me-retrieve Object

Db4o menyediakan tiga jenis Query yang dipakai untuk melakukan proses retrieving (...**baca mengambil**) data object dari database. Ketiga jenis object query itu antara lain:

- 1
- 2
- 3

Karena tujuan membuat aplikasi sederhana, maka saat ini jenis query yang digunakan adalah **Query by Example (QBE)**. QBE merupakan suatu bentuk query yang disediakan oleh db4o yang ditujukan bagi pemula yang baru melakukan penyesuaian dengan database for object salah satunya dengan **db4o**.

Ketika menggunakan QBE, kita akan membuat suatu **prototype object**. Kemudian kita akan meminta db4o untuk mengembalikan object dari database yang kriterianya bersesuaian dengan field dari prototype object yang kita berikan.

Hasil yang diinginkan dikembalikan sebagai instance **ObjectSet**. Dalam hal ini, kita menggunakan method **listResult** untuk menampilkan isi dari hasil retrieve.

```
public static void listResult(ObjectSet result) {
    System.out.println(result.size());
    while(result.hasNext()) {
        System.out.println(result.next());
    }
}
```

Untuk meretrieve **semua data** mahasiswa dari database, kita sediakan sebuah prototype yang kosong seperti berikut ini:

```
Mahasiswa prototype=new Mahasiswa(null, null, 0);
ObjectSet result=db.get(prototype);
listResult(result);
```

Output:

2
Budi/379008/3.67
Andi/383008/3.25

Db4o juga menyediakan suatu shortcut untuk untuk meretrieve semua instance dari suatu class:

```
ObjectSet result=db.get(Mahasiswa.class);
listResult(result);
```

Output:

2

Budi/379008/3.67
Andi/383008/3.25

Menampilkan object dengan nama mahasiswa bernama “Andi”

```
Mahasiswa prototype=new Mahasiswa("Andi", null, 0);
ObjectSet result=db.get(prototype);
listResult(result);
```

Output:

1
Andi/383008/3.25

Menampilkan object dengan NIM mahasiswa “379008”

```
Mahasiswa prototype=new Mahasiswa(null, "379008", 0);
ObjectSet result=db.get(prototype);
listResult(result);
```

Output:

1
Budi/379008/3.67

Meng-Update Object

Untuk meng-update object hampir mirip dengan cara menyimpan object ke-database. Pada kenyataannya, kita menggunakan perintah **set()** yang sama untuk meng-update object. Cukup memanggil kembali perintah **set()** setelah memodifikasi setiap object.

```
ObjectSet result=db.get(new Mahasiswa("Andi", null, 0));
Mahasiswa found=(Mahasiswa)result.next();
Mahasiswa.setIpk(3.45);
db.set(found);
System.out.println("Mengupdate Ipk mahasiswa untuk "+found);
```

Output:

Mengupdate Ipk mahasiswa untuk Andi/383008/3.45
2
Budi/379008/3.67
Andi/383008/3.45

Menghapus Object

Untuk menghapus object dari database digunakan method **delete()**. Sebagai contoh menghapus object pertama dengan nama “Andi”:

```
ObjectSet result=db.get(new Mahasiswa("Andi", null, 0));
Mahasiswa found=(Mahasiswa)result.next();
db.delete(found);
System.out.println("Deleted "+found);
```

Output:

Deleted Andi/383008/3.45
1
Budi/379008/3.67

Full Source Code

```

/*
 * Main.java
 *
 * Created on 4 Desember 2010, 22:25
 *
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
 */

package prjmahasiswa;
import java.io.File;
import com.db4o.Db4o;
import com.db4o.ObjectContainer;
import com.db4o.ObjectSet;
import prjmahasiswa.Mahasiswa;
/**
 *
 * @author kamid
 */
public class Main {

    /** Creates a new instance of Main */
    public Main() {
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        ObjectContainer db=Db4o.openFile("Mahasiswa.yap");
        try
        {
            storeFirstMahasiswa(db);
            storeSecondMahasiswa(db);
            retrieveAllMahasiswa(db);
            //retrieveMahasiswaByName(db);
            //updateIpkMahasiswa(db);
            //deleteMahasiswa(db);
        }
        finally
        {
            db.close();
        }
    }
    public static void storeFirstMahasiswa(ObjectContainer db) {
        Mahasiswa mahasiswa1=new Mahasiswa("Andi", "383008", Float.valueOf("3.25"));
        db.set(mahasiswa1);
        System.out.println("Stored "+mahasiswa1);
    }
    public static void storeSecondMahasiswa(ObjectContainer db) {
        Mahasiswa mahasiswa2=new Mahasiswa("Budi", "379008", Float.valueOf("3.67"));
        db.set(mahasiswa2);
        System.out.println("Stored "+mahasiswa2);
    }
    public static void retrieveAllMahasiswa(ObjectContainer db) {
        ObjectSet result=db.get(Mahasiswa.class);
        listResult(result);
    }
    public static void retrieveMahasiswaByName(ObjectContainer db) {
        Mahasiswa prototype=new Mahasiswa("Andi", null, 0);
        ObjectSet result=db.get(prototype);
        listResult(result);
    }
    public static void updateIpkMahasiswa(ObjectContainer db) {
        ObjectSet result=db.get(new Mahasiswa("Andi", null, 0));
    }

```

```

        Mahasiswa found=(Mahasiswa)result.next();
        found.setIpk(Float.valueOf("3.45"));
        db.set(found);
        System.out.println("Mengupdate Ipk mahasiswa untuk "+found);
        retrieveAllMahasiswa(db);
    }
    public static void deleteMahasiswa(ObjectContainer db) {
        ObjectSet result=db.get(new Mahasiswa("Andi", null, 0));
        Mahasiswa found=(Mahasiswa)result.next();
        db.delete(found);
        System.out.println("Deleted "+found);
        retrieveAllMahasiswa(db);
    }
    public static void listResult(ObjectSet result) {
        System.out.println(result.size());
        while(result.hasNext()) {
            System.out.println(result.next());
        }
    }
}

```

Output:

```

Stored Andi/383008/3.25
Stored Budi/379008/3.67
2
Budi/379008/3.67
Andi/383008/3.25
1
Andi/383008/3.25
Mengupdate Ipk mahasiswa untuk Andi/383008/3.45
2
Budi/379008/3.67
Andi/383008/3.45
Deleted Andi/383008/3.45
1
Budi/379008/3.67

```

Referensi

Db4o-5.2 Tutorial, db4objects Inc., USA, available: <http://www.db4o.com>

Irwanto, Djon., 2007, *Membangun Object Oriented Software dengan Java dan Object Database*, PT Elex Media Komputindo, Jakarta