



**Riza Satria Perdana, S.T., M.T.**

Teknik Informatika - STEI ITB

**Reflection**

# Studi Kasus

Pemrograman Berorientasi Objek

# Menggunakan Reflection untuk Plugin

- Reflection dapat digunakan untuk membuat Plugin
- Apa itu plugin? Plugin adalah kelas atau komponen yang bisa ditambahkan ke aplikasi tanpa perlu rekompilasi aplikasi
  - Contoh yang terkenal adalah plugin browser yang memungkinkan browser bisa membuka aneka macam file di dalam browser (flash, PDF) dengan bantuan plugin

# Problem Description

- Kita memiliki sebuah aplikasi SpreadSheet
- Ada banyak format spreadsheet, misalnya:
  - Ms Excel, OpenOffice, CSV
- Kita ingin membuat plugin untuk membuka File pada sebuah Spreadsheet
  - Cukup dengan menambah kelas baru, maka aplikasi Spreadsheet yang kita miliki dapat membuka jenis file baru

# Contoh Sistem Plugin [1]: deskripsi plugin

- Plugin diimplementasikan sebagai kelas
  - Untuk memudahkan pembuatan program: plugin pasti mengimplementasikan interface FileLoader
- Plugin diletakkan di sebuah direktori tertentu
  - untuk menambah format file yang didukung, plugin cukup dikompilasi (tanpa mengkompilasi seluruh aplikasi spreadsheet) disalin ke direktori tersebut, tanpa mengkompilasi ulang aplikasi

# Contoh Sistem Plugin [2a]: Design Interface FileLoader

- Interface ini diperlukan agar Plugin pasti mengimplementasikan method untuk meload file
  - cara lain adalah dengan mewajibkan programmer plugin memakai nama method tertentu, tapi ini lebih rumit dan tidak bisa diperiksa compiler
- Isi interface (daftar method) boleh seperti apa saja (sesuai kebutuhan)

# Contoh Sistem Plugin [2b]: Interface FileLoader

```
public interface FileLoader {  
    /*load file */  
    public void loadFile(String filename);  
    /*cek apakah file didukung plugin ini*/  
    public boolean isSupported(String  
        filename);  
    /*dapatkan jumlah baris*/  
    public int getRowCount();  
    /*dapatkan jumlah kolom*/  
    public int getColCount();  
    /*dapatkan isi sel tertentu*/  
    public String getCell(int row, int  
        col);  
}
```

# Contoh Sistem Plugin [3]: Algoritma

- Algoritma untuk meload file kelas di sebuah direktori adalah:
  - [1] Dapatkan daftar file di direktori
  - [2] Panggil `Class.forName("nama kelas")` untuk mendapatkan objek Class untuk file tersebut
  - [3] Periksa apakah kelas mengimplementasikan `FileLoader`
  - [4] Instansiasi kelas
  - [5] Panggil method pada objek

# Contoh Sistem Plugin [3-1]: List file

- Contoh berikut untuk mendapatkan daftar file di sebuah direktori (untuk mempersingkat, semua exception handling tidak ditampilkan):
- Contoh nama direktori adalah “plugin” (direktori ini harus ada di class path)

```
File f = new File("plugin");  
String names[] = f.list();
```



# Contoh Sistem Plugin [3-2]: Meload Kelas

- Loading kelas dilakukan dengan `Class.forName()`,  
Class Loader Java akan otomatis mencari ke classpath

```
for (String name: names) {  
    Class c = Class.forName(name);  
    /*proses kelas */  
}
```

## Contoh Sistem Plugin [3-3a]: Memeriksa Interface

- Kelas yang ada di direktori plugin belum tentu mengimplementasikan interface FileLoader, jadi kita perlu memeriksanya dulu
- Dapatkan daftar interface yang diimplementasikan kelas (ingat sebuah kelas bisa mengimplementasikan banyak interface) dengan `getInterfaces()`, dan periksa namanya

## Contoh Sistem Plugin [3-3b]: Kode Memeriksa Interface

```
Class intfs[] = c.getInterfaces();  
for (Class itf : intfs) {  
    if  
        (itf.getName().equals("FileLoader"))  
    {  
        /*proses kelas*/  
    } else {  
        /*abaikan kelas ini*/  
    }  
}
```



# Contoh Sistem Plugin [3-4]: Instansiasi Kelas

- Jika sudah yakin bahwa kelas mengimplementasikan FileLoader, maka kelas bisa diinstansiasi

```
FileLoader fl = (FileLoader)c.newInstance();
```

- Umumnya untuk menyimpan daftar plugin digunakan Vector:

```
Vector<FileLoader> flplugins = new  
    Vector<FileLoader>();  
flplugins.add(fl);
```

# Contoh Sistem Plugin [3-5]: Memanggil method

- Kelas yang mengimplementasikan FileLoader yang sudah diinstansiasi bisa dipakai langsung:

```
fl.loadFile("hello.csv");
```

- Untuk plugin file loader, salah satu penggunaannya adalah menanyakan satu persatu plugin apakah mendukung file tertentu lalu meminta plugin yang mendukung untuk meloadnya

# Contoh Sistem Plugin [3-5]:

## Contoh method LoadFile

```
boolean loadFileFromPlugin(String name) {  
    for (FileLoader fl : flplugins) {  
        if (fl.isSupported(name)) {  
            fl.loadFile(name);  
            for(int i = 0; i<fl.getRowCount(); i++){  
                for(int j = 0; j<fl.getColCount(); j++) addCell(i,  
                    j, fl.getCell(i,j));  
            }  
            return true; /*ok, ada plugin yg sesuai*/  
        }  
    }  
    return false; /*gagal, file tidak didukung*/  
}
```



# Contoh penggunaan plugin

- Beberapa contoh plugin yang bisa dibuat:
  - plugin untuk menyimpan file
  - plugin untuk memproses data tertentu (misal plugin untuk mendefinisikan fungsi baru untuk spreadsheet)

# Terima Kasih