Berita Online Menggunakan Media RSS Pada MIDlet

Robertus Lilik Haryanto
lilik.haryanto@gmail.com
http://secangkirkopipanas.org

Berita bagi sebagian besar orang merupakan suatu hal yang sangat penting, karena bagi mereka berita dapat memperluas wawasan. Banyak sekali beredar berita di Internet, seperti Yahoo! News, Google! News, New York Times, dan masih banyak lagi.

Dengan menggunakan media RSS, pengguna tidak lagi perlu membuka Internet browser dan melakukan browsing pada situs berita yang dikehendaki, tetapi hanya dengan menggunakan RSS client yang akan mengambil informasi berita pada situs yang bersangkutan. Sebagai pengantar RSS, Anda dapat membacanya di Pengantar RSS.

Pendahuluan

Banyak aplikasi RSS Client yang sudah beredar, baik aplikasi web, aplikasi *mobile phone*, bahkan sampai berupa plugin-plugin dari Internet browser. Pada artikel ini, penulis akan mencoba menjelaskan cara pembuatan aplikasi *mobile phone* (MIDlet) yang berfungsi sebagai RSS client untuk mendapatkan informasi dari situs The New York Time.

Isi

Aplikasi yang akan dikembangkan ini menggunakan pustaka kXML2 sebagai *XML parser*. Sehingga pustaka ini perlu di ikutsertakan dalam project yang akan dibuat. Anda bisa mendapatkan pustaka kXML2 melalui situs http://kxml.sourceforge.net/.

Untuk mengambil data melalui koneksi Internet, Anda bisa menggunakan *source code* berikut:

```
try {
     conn = (HttpConnection)
Connector.open("http://www.nytimes.com/services/xml/rss/nyt/National.xml");
     conn.setRequestMethod(HttpConnection.GET);
     conn.setRequestProperty("Content-Type", "text/xml");
```

```
     dos = new DataOutputStream(conn.openDataOutputStream());
     parser = new RSSParser(50);
     parser.parse(new DataInputStream(conn.openDataInputStream()));
     getDisplay().setCurrent(get_listNews());
     int count = parser.getCount();
     for (int i = 0; i < count; i++) {
          get_listNews().append(
               parser.getRss().getItems()[i].getTitle() + "\n" +
               parser.getRss().getItems()[i].getPubDate(), null);
     }
} catch (Exception e) {
     get_alertError().setString(e.getMessage());
     getDisplay().setCurrent(get_alertError());
}
```

Setelah informasi diperoleh dari Internet, langkah berikutnya yaitu melakukan *parsing* data dengan langkah seperti kode berikut:

```
public void parse(DataInputStream dis) {
        parser = new KXmlParser();
        try {
            parser.setInput(new InputStreamReader(dis));
            parser.next();

            parser.require(XmlPullParser.START_TAG, null, "rss");
            parser.nextTag();

            parser.require(XmlPullParser.START_TAG, null, "channel");

            while (parser.nextTag() != XmlPullParser.END_TAG) {

                parser.require(XmlPullParser.START_TAG, null, "title");
                getRss().setTitle(parser.nextText());
                parser.require(XmlPullParser.END_TAG, null, "title");

                parser.nextTag();

                parser.require(XmlPullParser.START_TAG, null, "link");
                getRss().setLink(parser.nextText());
                parser.require(XmlPullParser.END_TAG, null, "link");

                parser.nextTag();

                parser.require(XmlPullParser.START_TAG, null, "description");
                getRss().setDescription(parser.nextText());
                parser.require(XmlPullParser.END_TAG, null, "description");

                parser.nextTag();

                parser.require(XmlPullParser.START_TAG, null, "language");
                getRss().setLanguage(parser.nextText());
                parser.require(XmlPullParser.END_TAG, null, "language");

                parser.nextTag();

                parser.require(XmlPullParser.START_TAG, null, "copyright");
                getRss().setCopyright(parser.nextText());
                parser.require(XmlPullParser.END_TAG, null, "copyright");

                parser.nextTag();

                parser.require(XmlPullParser.START_TAG, null,
"lastBuildDate");
                getRss().setLastBuildDate(parser.nextText());
                parser.require(XmlPullParser.END_TAG, null, "lastBuildDate");
                parser.nextTag();
```

```
                parser.require(XmlPullParser.START_TAG, null, "image");

                while (parser.nextTag() != XmlPullParser.END_TAG) {
                    RSSImage rssimage = new RSSImage();

                    parser.require(XmlPullParser.START_TAG, null, "title");
                    rssimage.setTitle(parser.nextText());
                    parser.require(XmlPullParser.END_TAG, null, "title");

                    parser.nextTag();

                    parser.require(XmlPullParser.START_TAG, null, "url");
                    rssimage.setUrl(parser.nextText());
                    parser.require(XmlPullParser.END_TAG, null, "url");

                    parser.nextTag();

                    parser.require(XmlPullParser.START_TAG, null, "link");
                    rssimage.setLink(parser.nextText());
                    parser.require(XmlPullParser.END_TAG, null, "link");

                    getRss().setImages(rssimage);
                }

                parser.require(XmlPullParser.END_TAG, null, "image");

                while (parser.nextTag() == XmlPullParser.START_TAG) {

                    parser.require(XmlPullParser.START_TAG, null, "item");

                    while (parser.nextTag() != XmlPullParser.END_TAG) {

                        rssitems[count] = new RSSItem();
                        parser.require(XmlPullParser.START_TAG, null,
"title");
                        rssitems[count].setTitle(parser.nextText());
                        parser.require(XmlPullParser.END_TAG, null, "title");

                        parser.nextTag();

                        parser.require(XmlPullParser.START_TAG, null, "link");
                        rssitems[count].setLink(parser.nextText());
                        parser.require(XmlPullParser.END_TAG, null, "link");

                        parser.nextTag();

                        parser.require(XmlPullParser.START_TAG, null,
"description");
                        rssitems[count].setDescription(parser.nextText());
                        parser.require(XmlPullParser.END_TAG, null,
"description");

                        parser.nextTag();

                        parser.require(XmlPullParser.START_TAG, null,
"author");
                        rssitems[count].setAuthor(parser.nextText());
                        parser.require(XmlPullParser.END_TAG, null, "author");

                        parser.nextTag();

                        parser.require(XmlPullParser.START_TAG, null, "guid");
                        rssitems[count].setGuid(parser.nextText());
                        parser.require(XmlPullParser.END_TAG, null, "guid");
```

```
                        parser.nextTag();

                        parser.require(XmlPullParser.START_TAG, null,
"pubDate");
                        rssitems[count].setPubDate(parser.nextText());
                        parser.require(XmlPullParser.END_TAG, null,
"pubDate");

                        count++;
                        getRss().setItems(rssitems);
                    }

                    parser.require(XmlPullParser.END_TAG, null, "item");
                }

                parser.require(XmlPullParser.END_TAG, null, "channel");

            }

            parser.require(XmlPullParser.END_TAG, null, "rss");

        } catch (Exception e) {
            e.printStackTrace();
        }
}
```

Berikut class-class yang digunakan untuk merepresentasikan data pada RSS:

1. RSS.java

```
package org.secangkirkopipanas.rss.common;

/**
 *
 * @author lilik.haryanto
 */
public class RSS {

    private String title;

    private String link;

    private String description;

    private String language;

    private String copyright;

    private String lastBuildDate;

    private RSSImage images;

    private RSSItem[] items;

    private static RSS instance;

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

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
```

```
    }

    public String getLink() {
        return link;
    }

    public void setLink(String link) {
        this.link = link;
    }

    public String getDescription() {
        return description;
    }

    public void setDescription(String description) {
        this.description = description;
    }

    public String getLanguage() {
        return language;
    }

    public void setLanguage(String language) {
        this.language = language;
    }

    public String getLastBuildDate() {
        return lastBuildDate;
    }

    public void setLastBuildDate(String lastBuildDate) {
        this.lastBuildDate = lastBuildDate;
    }

    public RSSImage getImages() {
        return images;
    }

    public void setImages(RSSImage images) {
        this.images = images;
    }

    public RSSItem[] getItems() {
        return items;
    }

    public void setItems(RSSItem[] items) {
        this.items = items;
    }

    public static synchronized RSS getInstance() {
        if (instance == null) instance = new RSS();
        return instance;
    }

    public String getCopyright() {
        return copyright;
    }

    public void setCopyright(String copyright) {
        this.copyright = copyright;
    }

}
```

2. RSSImage.java

```
package org.secangkirkopipanas.rss.common;

/**
 *
 * @author lilik.haryanto
 */
public class RSSImage {

    private String title;

    private String url;

    private String link;

    private String width;

    private String height;

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

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public String getUrl() {
        return url;
    }

    public void setUrl(String url) {
        this.url = url;
    }

    public String getLink() {
        return link;
    }

    public void setLink(String link) {
        this.link = link;
    }

    public String getWidth() {
        return width;
    }

    public void setWidth(String width) {
        this.width = width;
    }

    public String getHeight() {
        return height;
    }

    public void setHeight(String height) {
        this.height = height;
    }

}
```

3. RSSItem.java

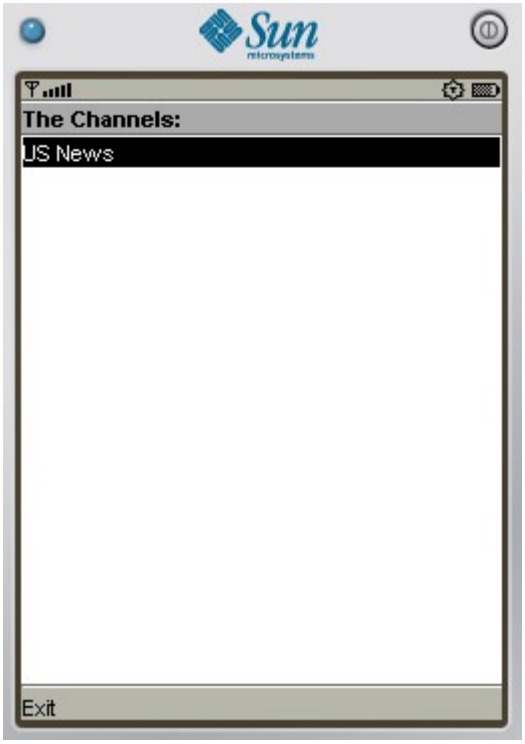
```
package org.secangkirkopipanas.rss.common;

/**
 *
 * @author lilik.haryanto
 */
public class RSSItem {

    private String title;

    private String link;

    private String description;

    private String author;

    private String guid;

    private String pubDate;

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

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public String getLink() {
        return link;
    }

    public void setLink(String link) {
        this.link = link;
    }

    public String getDescription() {
        return description;
    }

    public void setDescription(String description) {
        this.description = description;
    }

    public String getAuthor() {
        return author;
    }

    public void setAuthor(String author) {
        this.author = author;
    }

    public String getGuid() {
        return guid;
    }

    public void setGuid(String guid) {
        this.guid = guid;
    }
```

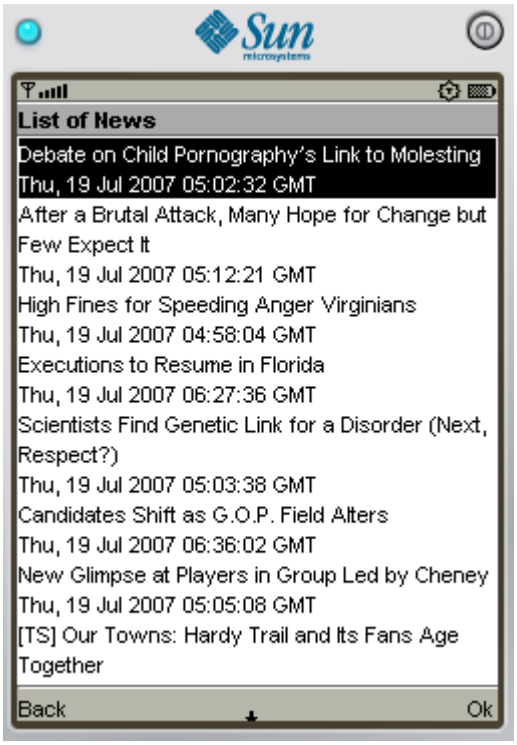
```
    public String getPubDate() {
        return pubDate;
    }

    public void setPubDate(String pubDate) {
        this.pubDate = pubDate;
    }

}
```

Dengan melakukan parsing data, maka data yang diperoleh dapat diinformasikan dengan format yang bisa diterima oleh pengguna, sebagai contoh yaitu dengan format list.

Pada bagian output di bawah ini, Anda dapat melihat hasil eksekusi dari aplikasi RSS client ini. **Selamat mencoba!**

Output





Penutup

Banyak ide yang dapat diimplementasikan selain menggunakan RSS, sebagai contoh yaitu menggunakan Web Service dalam penyajian berita. Dengan menggunakan RSS, informasi lain yang bisa Anda dapatkan antara lain daftar lagu terbaru, hasil pencarian dari mesin pencari (*search engine*), bahkan daftar email terbaru.

Pada artikel ini disertakan *source* program yang bisa Anda coba sendiri, sehingga dapat lebih mudah dalam proses pembelajaran.

Referensi

1. http://kxml.sourceforge.net/
2. kXML2 Documentation

Biografi Penulis

Robertus Lilik Haryanto. Lahir di Klaten, 2 Oktober 1983. Menyelesaikan pendidikan program S1 pada program studi Teknik Informatika Universitas Sanata Dharma (USD), Yogyakarta pada bulan Agustus 2005. Pernah bekerja di sebuah perusahaan penyedia layanan *web hosting*, PT. Inter Lintas Media Yogyakarta sebagai *web programmer*. Selain itu, pernah mengajar pelatihan Java dan Delphi di Gama Learning Center (GLC), Yogyakarta. Saat ini bekerja sebagai developer di PT. Jati Piranti Solusindo (eCom), Jakarta, sejak tahun 2005, dan juga aktif mengajar Java Programming di Binus Center Kelapa Gading sejak Desember 2006.

Menggeluti dunia pemrograman sejak tahun 1999 dan bahasa pemrograman yang paling digemari adalah Java (J2SE, J2ME, dan J2EE), C++, Microsoft C#.NET dan PHP. Selain melakukan beberapa penelitian, juga sedang mendalami beberapa bahasa pemrograman di atas.