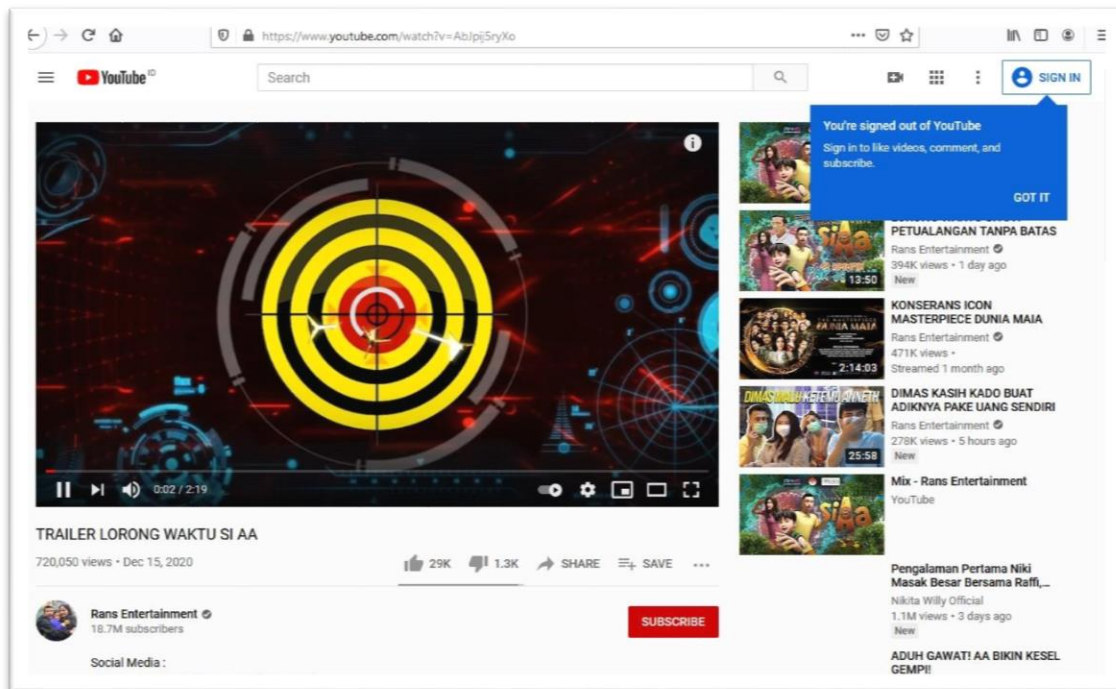


MODUL 12- XML



Gambar 1. Youtube pertama kali dirilis pada tahun 2005 dengan menggunakan Action Script dan Flash Player

Saat ini kita telah mengenal VARIABLE, FUNCTION, PERCABANGAN, dan LOGIKA PEMROGRAMAN dengan menggunakan Action Script 3.0. Sebenarnya, masih ada beberapa topik pemrograman yang harus kita pelajari lagi sebelum masuk ke topik OOP (Object Oriented Programming), di antaranya ada Operator Aritmatika, Looping, Array, XML, dan Component. Akan tetapi karena terbatasnya ruang dan waktu, maka kita akan langsung melompat ke pembahasan mengenai Pemrograman Berbasis Objek dengan menggunakan Action Script 3.0.

Beberapa di antara topik yang tertinggal tersebut dapat dipelajari pada modul Praktikum Teknologi Multimedia, sehingga diharapkan topik tentang Class dan Object ini akan memperkaya pengetahuan mahasiswa tentang Pemrograman Berbasis Objek dan bagaimana melakukan pemrograman berbasis objek dengan menggunakan Action Script 3.0.

CLASS DAN OBJECT

Secara konsep, **Class** adalah sebuah “cetak biru” dari **Object**. Seperti halnya sebuah cetak biru dari sebuah rumah, Class merupakan gambaran teknis dari rumah sebelum dibangun. Dengan kata lain, **Class** adalah gambaran skematis dari sebuah **Objek**, akan tetapi Class bukanlah Objek itu sendiri. Ketika kita membuat program dengan menggunakan konsep PBO (Pemrograman Berbasis Objek), pada prinsipnya kita membuat “cetak biru” atau Class. Berdasarkan Class tersebut, maka kemudian Objek akan dibuat pada saat program dijalankan.

Seperti halnya dengan cetak biru pada proses pembuatan rumah, seorang developer bisa dengan mudah menggunakan cetak biru tersebut untuk membangun banyak rumah sekaligus dengan pola yang sama. Hal ini menjadikan pekerjaannya sangat efisien karena sang

developer hanya perlu membayar satu kali kepada Arsitek yang merancang rumah tersebut. Namun, pada rumah-rumah tersebut masing-masing akan memiliki alamat yang berbeda, cat yang berbeda-beda, bentuk pintu yang berbeda, bahan pintu yang berbeda-beda, warna pintu yang berbeda-beda, dan sebagainya.

Sama halnya dengan analogi di atas, dengan adanya Class, maka programmer dapat berkali-kali membuat Object dengan pola yang sama. Namun, setiap Objek diberi nama yang berbeda, memiliki ukuran yang berbeda, warna yang berbeda, dan seterusnya. Pada prinsipnya Objek-objek ini memiliki pola yang sama karena menggunakan Class yang sama, akan tetapi memiliki nama yang unik dan property yang bisa diubah-ubah.

Pada aplikasi Flash, kita mengenal istilah “instance”. Kata “instance” sendiri menjelaskan tentang “sebuah Objek”. Ketika sebuah Objek diciptakan dari sebuah Class, proses ini dinamakan dengan “instantiated” dan dengan cara inilah sebuah Objek dilahirkan ke dunia. Dengan demikian, kata Objek tidak lain adalah “instance”.

Contoh proses pembuatan Objek pada aplikasi Flash terlihat pada saat kita membuat “Symbol” yang telah dilakukan sejak Flash versi 1. Symbol adalah sebuah Objek yang akan langsung terdaftar sebagai anggota unik di “Library”. Pada saat Symbol ini digandakan, dan kemudian diubah, maka semua Symbol akan diubah. Akan tetapi, posisi masing-masing Symbol tetap berada di tempatnya masing-masing, demikian halnya jika kita mengubah ukuran atau memutar salah satu Symbol tidak akan berpengaruh terhadap Symbol yang lain. Hal ini menjelaskan hubungan antara Class dan Objek. Symbol adalah sebuah Class, akan tetapi Symbol yang telah ditempatkan di Stage tidak lain adalah Objek. Setiap Objek yang dibuat dengan menggunakan Symbol yang sama akan memiliki kesamaan, akan tetapi dapat memiliki property yang berbeda-beda. Pada saat Symbol dibuat di Stage, maka setiap Symbol harus memiliki “nama instance” yang berbeda sehingga program akan melekat pada Symbol dengan nama instance tertentu.

Membuat Objek dari Class

Sebelum kita membahas lebih jauh tentang bagaimana cara membuat Class, terlebih dahulu kita pelajari mengenai cara membuat Objek dari Class tertentu dan bagaimana memanfaatkan method dan property Class tersebut.

Seperti yang telah kita lakukan berkali-kali pada Praktikum Teknologi Multimedia, untuk membuat Button kita bisa mulai dengan menggambar bentuk Button tersebut, contohnya dengan membuat tulisan HOME dengan Text Tool. Kemudian yang kita lakukan adalah dengan mengubah gambar Button tersebut menjadi Symbol: Button.

Setelah gambar tersebut diubah menjadi Symbol Button, maka proses berikutnya adalah memberi “nama instance”. Secara umum proses ini dinamakan dengan proses “instantiate”. Lalu bagaimana jika proses ini dilakukan dengan Action Script?

Contoh berikut ini adalah membuat Objek dari sebuah Class:

```
var waktu:Date = new Date();
```

```
var suara:Sound = new Sound();
```

Kedua contoh di atas menggunakan pola yang sama, yaitu menggunakan keyword “var” untuk deklarasi variable. Kemudian diikuti dengan nama variable (Objek) dan tipe data atau Class dari Objek tersebut. Perintah “new” digunakan untuk membuat Objek baru dari Class tertentu, misalnya variable “waktu” dibuat sebagai sebuah Objek dari Class Date, variable “suara” dibuat sebagai sebuah Objek dari Class Sound.

Script ini sama dengan proses sebelumnya yang dilakukan dengan membuat gambar Button, mengubahnya menjadi Symbol: Button, kemudian membuat “nama instance” Button tersebut. Mudah, bukan?

Method

Setelah kita membuat Objek dan memberi nama instance, maka kita bisa mengakses Method dan Property dari Objek atau Class. Contoh yang sering kita gunakan untuk Symbol Button adalah Method untuk menambahkan Event Listener, misalnya sebagai berikut:

```
home_btn.addEventListener();
```

Pada contoh ini kita menggunakan method yang dimiliki oleh Class. Contoh lain:

```
var jam    = waktu.getHours();
var menit = waktu.getMinutes();
var detik = waktu.getSeconds();
```

Sekarang kita bisa melihat bahwa setelah sebuah variable dideklarasikan dan menjadi sebuah Objek dari Class tertentu, maka method dari Class akan melekat pada Objek tersebut. Pada contoh di atas, sebuah Button dapat menggunakan method **addEventListener()**. Demikian halnya dengan Objek waktu dapat menggunakan method: **getHours()**, **getMinutes()**, dan **getSeconds()**.

Property

Seperti halnya method, property juga menjadi bagian yang melekat dari Class. Oleh karena itu, sebuah Objek dari Class tertentu akan memiliki property seperti halnya Class itu sendiri. Contoh berikut adalah bagaimana sebuah Objek menggunakan property dari Class:

```
var text1:TextField = new TextField();

text1.text = "Hello World";
text1.x    = 250;
text1.y    = 200;
```

Seperti halnya dengan method, setelah sebuah Objek dideklarasikan, maka Objek tersebut dapat menggunakan property yang dimiliki oleh Class. Contoh di atas memperlihatkan bagaimana Objek “text1” memiliki property: text, x, dan y. Ketiga property dapat diisi dengan sebuah nilai.

Membuat Objek di Stage

Berbeda dengan cara membuat Button atau Movie Clip yang langsung dibuat di Stage, Objek yang telah dibuat dengan menggunakan script tidak langsung ditempatkan di Stage. Untuk itu, dibutuhkan perintah untuk menempatkan Objek tersebut di layar (Stage). Contohnya adalah sebagai berikut:

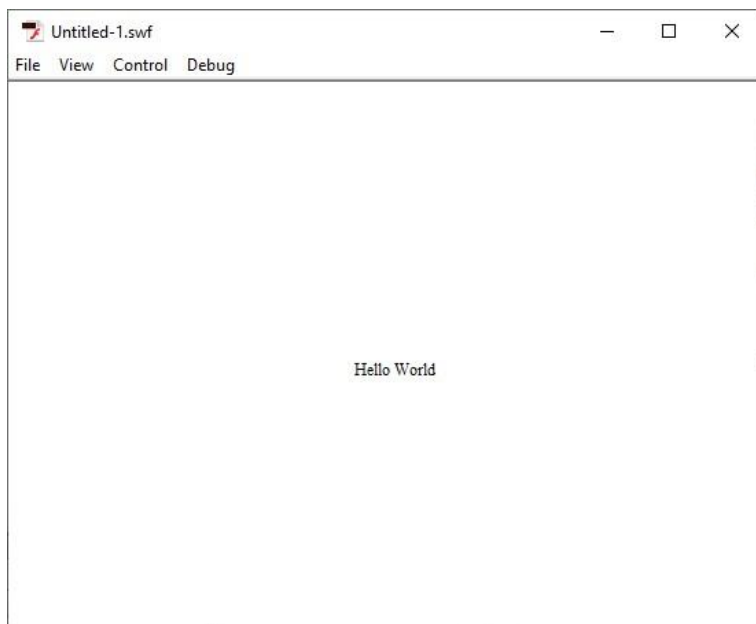
```
var text1:TextField = new TextField();

text1.text = "Hello World";
text1.x    = 250;
text1.y    = 200;

stage.addChild(text1);
```

Perintah “addChild” adalah method yang dimiliki oleh stage. Dengan menggunakan perintah addChild tersebut, maka Objek ditempatkan di layar (stage).

Jika script di atas dijalankan, maka kita akan mendapatkan tampilan berikut ini:



XML

Setelah kita mengetahui konsep dan penerapan Class dan Object, maka kita akan mencoba menggunakan konsep Class dan Object tersebut dalam membaca file XML. Seperti halnya Bahasa Pemrograman lainnya, Action Script 3.0 biasa dipergunakan untuk menulis dan membaca database. Akan tetapi, karena proses membaca dan menulis database membutuhkan “binding” yang kuat dengan system database, maka akan jauh lebih mudah apabila kita menggunakan script yang sudah teruji seperti PHP.

XML adalah kependekan dari **Extensible Markup Language**, yakni sebuah Bahasa “markup” yang dapat diperluas. Istilah “markup” merujuk pada proses menandai text sehingga akan

dibaca oleh Browser internet sebagai sebuah “perintah”. Sedangkan istilah “Extensible” merujuk pada “Tag” yang dapat dibuat sendiri oleh programmer tanpa mengikuti standar yang biasa dipergunakan pada HTML.

Jika Anda sudah mengenal HTML, maka istilah “markup” biasa dilakukan dengan menandai text dengan “<” dan “>”. Dengan cara tersebut, maka Browser Internet akan memperlakukannya berbeda dari text biasa tanpa tanda “markup” tersebut. Contohnya, ketika Browser Internet menemukan tanda “<h1>” dan “</h1>” maka kemudian teks yang diapit oleh kedua tanda tersebut akan ditampilkan berukuran besar dan tebal atau diberi style “Heading 1”. Istilah “markup” ini kemudian secara luas dikenal dengan istilah “Tag”.

Contoh lain dari “Tag” tersebut adalah untuk Bold, <i> untuk Italic, <u> untuk Underlined, <p> untuk paragraph, <a> untuk anchor (hyperlink), dan lain sebagainya. Pada prinsipnya HTML telah memiliki “Tag” baku yang akan dibaca dan diperlakukan berbeda oleh Browser Internet.

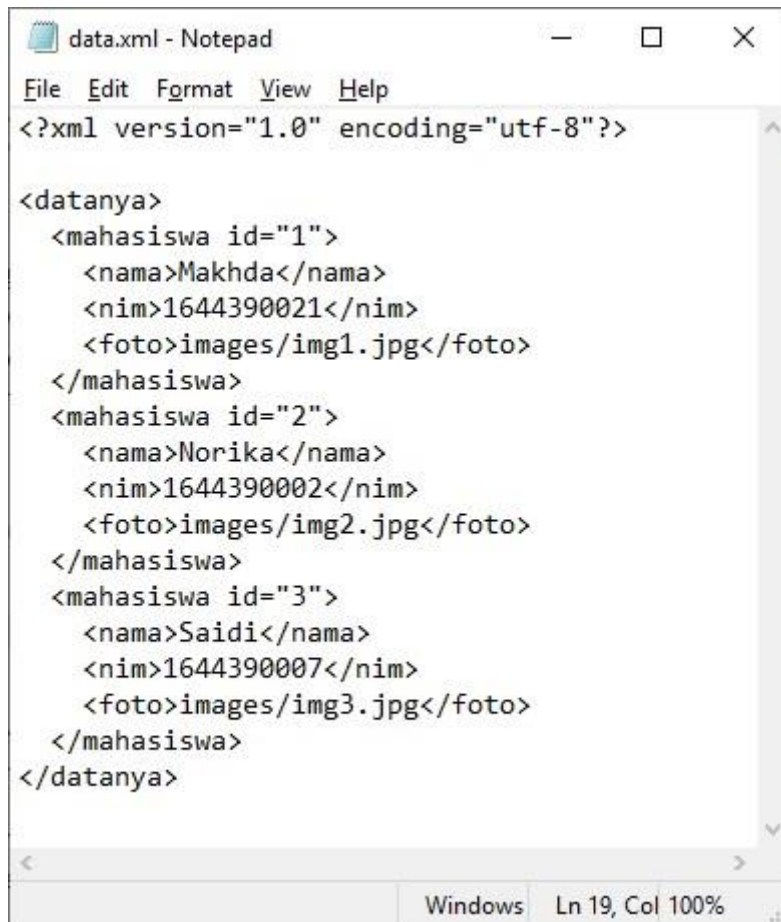
XML sedikit berbeda dari HTML, karena menggunakan “Tag” yang diperluas sehingga tidak lagi mengikuti standar baku tersebut. Tag yang digunakan dalam XML diatur oleh programmer sehingga mudah untuk dibaca dan dimengerti disesuaikan dengan jenis data yang ada di dalamnya. XML biasanya disandingkan dengan database, sehingga penulisan tag pada dokumen XML biasanya mirip dengan struktur data pada database. Contoh struktur data pada XML adalah sebagai berikut:

```
<mahasiswa id="1">
  <nama>Fulan bin Fulan</nama>
  <nim>20201234567890</nim>
  <jurusan>Teknik Informatika</jurusan>
  <foto>../images/20201234567890.jpg</foto>
</mahasiswa>
```

Format tag di atas tentu tidak dikenal pada HTML, akan tetapi dengan cara inilah XML dipergunakan secara luas untuk menyimpan data dengan struktur yang rapi. Secara sepintas kita dapat melihat bagaimana struktur data yang ada di dalamnya disusun dengan menggunakan tag yang sangat mudah dibaca.

Membuat Struktur XML

File XML dapat dibuat dengan menggunakan aplikasi pembuat text sederhana seperti Notepad pada Sistem Operasi Windows. Extension file XML adalah dengan menggunakan akhiran .xml sehingga akan menjadi file yang dibaca oleh Browser sebagai file xml. Format penulisan file XML harus mengikuti standar yang berlaku. Contohnya adalah sebagai berikut:



Pada contoh di atas, terlihat pada baris pertama dituliskan format standar XML sebagai berikut:

```
<?xml version="1.0" encoding="utf-8"?>
```

Baris pertama ini berfungsi untuk memberitahu Browser bahwa file ini adalah file XML, menggunakan format versi 1.0 dan menggunakan text dengan standar huruf Latin (UTF-8).

Baris berikutnya adalah Node pertama dengan tag “data”. Mirip sebuah folder pada computer, Node “data” ini merupakan sebuah kelompok Node dengan lingkup yang besar.

Node berikutnya di bawah Node “data” adalah Node “mahasiswa” yang menggunakan atribut “id”.

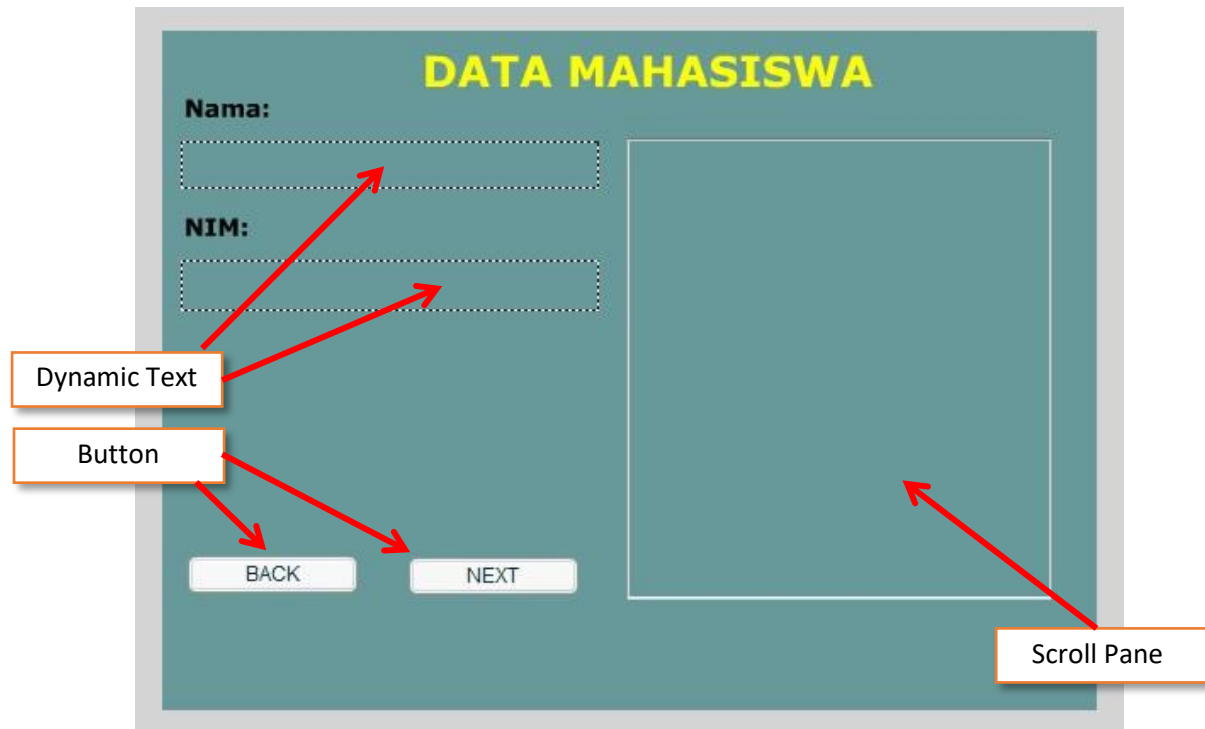
Di bawah Node “mahasiswa” terdapat 3 Node lagi yang terdiri dari: “nama”, “nim”, dan “foto”. Setiap Node mahasiswa dengan id yang berbeda-beda kemudian masing-masing diisi dengan 3 Node ini.

Membaca File XML

Setelah file XML dibuat dengan menggunakan aplikasi Notepad, selanjutnya kita membuat file movie Flash yang akan kita gunakan sebagai aplikasi interaktif yang dapat membaca file XML dan sekaligus menampilkan text dan media yang disertakan, misalnya foto, suara, dan video.

Untuk itu, kita dapat membuat file movie Flash dengan menggunakan Action Script 3.0. File movie Flash ini akan kita gunakan untuk menampilkan text dan foto seperti yang telah dituliskan dalam file data.xml.

Berikut ini adalah contoh antarmuka movie Flash yang akan kita gunakan untuk menampilkan text dan foto:



Berikutnya adalah menuliskan Action Script yang akan kita gunakan untuk membaca file XML yang telah dibuat sebelumnya:

```
//membuat loader untuk menyimpan alamat url var
loader:URLLoader = new ULLoader();

//membuat variable url untuk menyimpan alamat file var
url:URLRequest = new URLRequest("data.xml");

//load file xml loader.load(url);

//berhasil diload jalankan function
loader.addEventListener(Event.COMPLETE, loaderComplete);

function loaderComplete(e:Event):void{
```

```
//membuat objek myXML dari Class XML var
myXML:XML = new XML(loader.data);

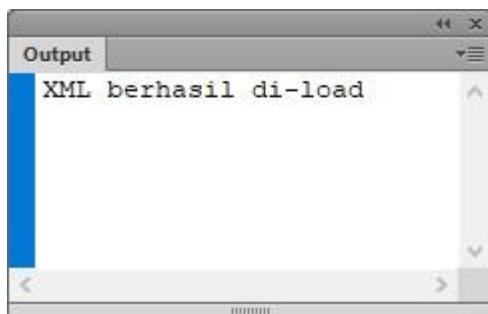
//abaikan penulisan spasi pada file xml
myXML.ignoreWhite = true;

//tampilkan pesan di panel output trace("XML
berhasil di-load");
}
```

Script di atas akan me-load file xml dengan beberapa langkah:

1. Membuat Objek Loader untuk menampung alamat url.
2. Membuat variable url untuk menyimpan alamat url file xml.
3. Me-load file xml ke Objek Loader.
4. Jika file xml selesai di-load, jalankan function.
5. Buat variable untuk menampung data dari file xml.
6. Abaikan spasi pada file xml.
7. Tampilkan pesan “XML berhasil di-load”.

Jika Movie Flash di atas dijalankan, maka ditampilkan pesan sebagai berikut:



Membaca Node XML

Setelah berhasil membaca file xml dan me-load data, maka tahap berikutnya adalah membaca Node xml yang diperlukan sesuai dengan variable yang disiapkan untuk menampung data tersebut.

Berikut ini adalah kelanjutan dari function tersebut:

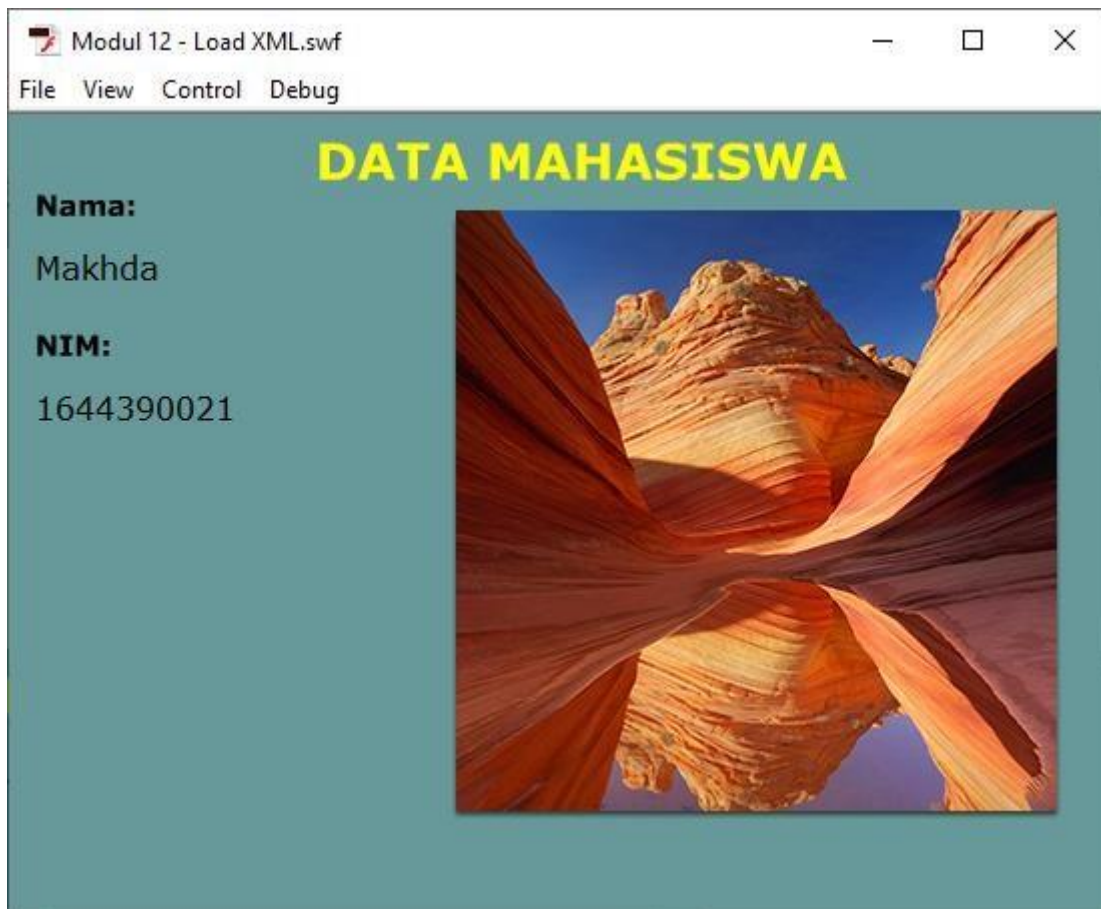
```
function loaderComplete(e:Event):void{

    //membuat objek myXML dari Class XML
    var myXML:XML = new XML(loader.data);

    //abaikan penulisan spasi pada file xml
    myXML.ignoreWhite = true;
```

```
//tempatkan data xml sebagai text dan konten foto
displaynama.text    = myXML.child(0).child(0);
displaynim.text     = myXML.child(0).child(1);
displayfoto.source  = myXML.child(0).child(2);
}
```

Script di atas akan menempatkan data pada XML menjadi data dalam bentuk yang sesuai dengan yang ditempatkan, yakni sebagai text dan sebagai source foto. Dengan cara tersebut maka akan ditampilkan halaman sebagai berikut:



Mengapa hal tersebut bisa terjadi? Perhatikan script untuk memindahkan data dari xml ke dalam text dan source foto sebagai berikut:

```
displaynama.text    = myXML.child(0).child(0);
displaynim.text     = myXML.child(0).child(1);
displayfoto.source  = myXML.child(0).child(2);
```

myXML akan merujuk pada tag **<data>** sebagai Node pertama.

Node kedua adalah **child(0)** yang merujuk pada tag **<mahasiswa>**. Index 0 adalah untuk urutan pertama sehingga data mahasiswa dengan id="1" yang akan dibaca oleh program.

Node ketiga adalah **child(0)** yang kedua yang merujuk pada tag **<nama>**, Index 0 adalah untuk urutan pertama pada Node ini yakni **<nama>**, sedangkan **child(1)** atau index 1 adalah untuk urutan kedua pada Node ini yakni **<nim>**, dan **child(2)** atau index 2 adalah untuk urutan ketiga pada Node ini yakni **<foto>**.

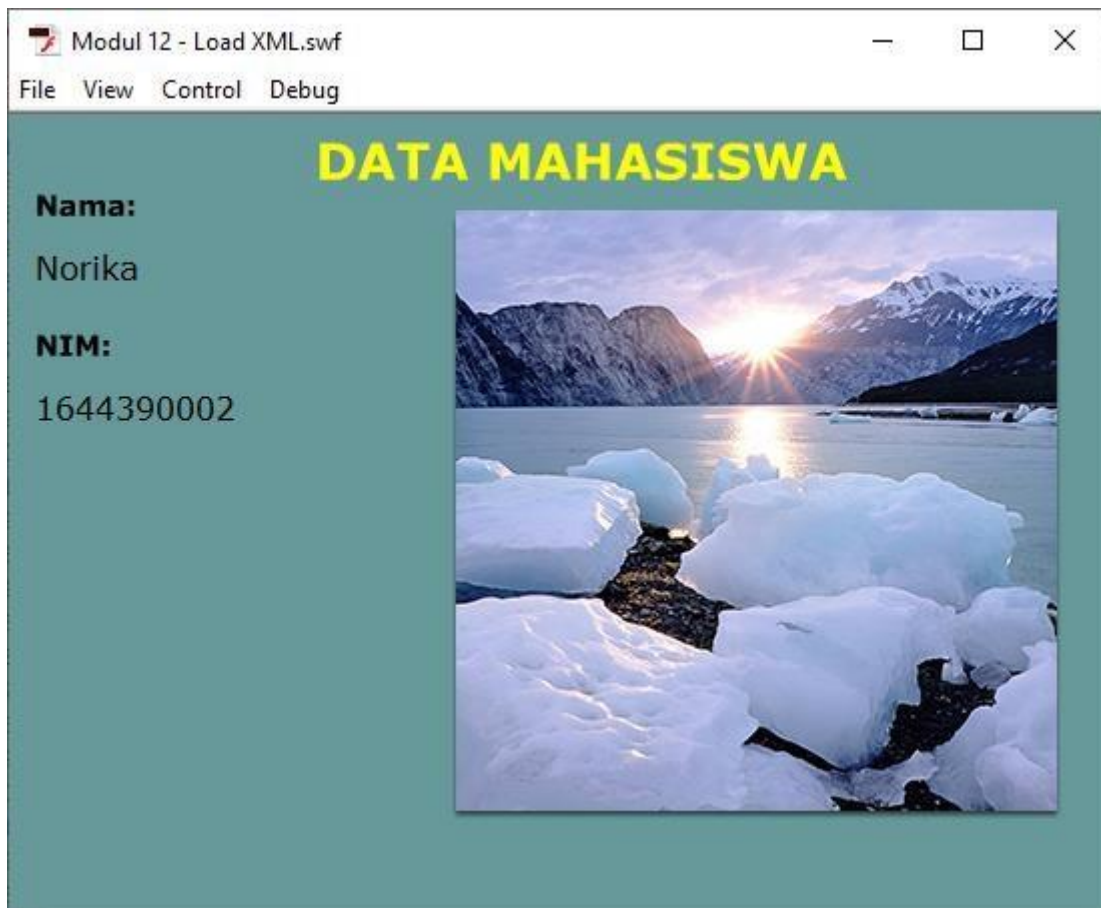
Bagaimana jika yang akan kita tampilkan adalah data mahasiswa dengan id="2", id="3", dan seterusnya?

Untuk mengubah pointer ke urutan data tertentu, maka kita bisa mengubah Node kedua, misalnya: **child(1)** untuk urutan kedua, **child(2)** untuk urutan ketiga, dan seterusnya.

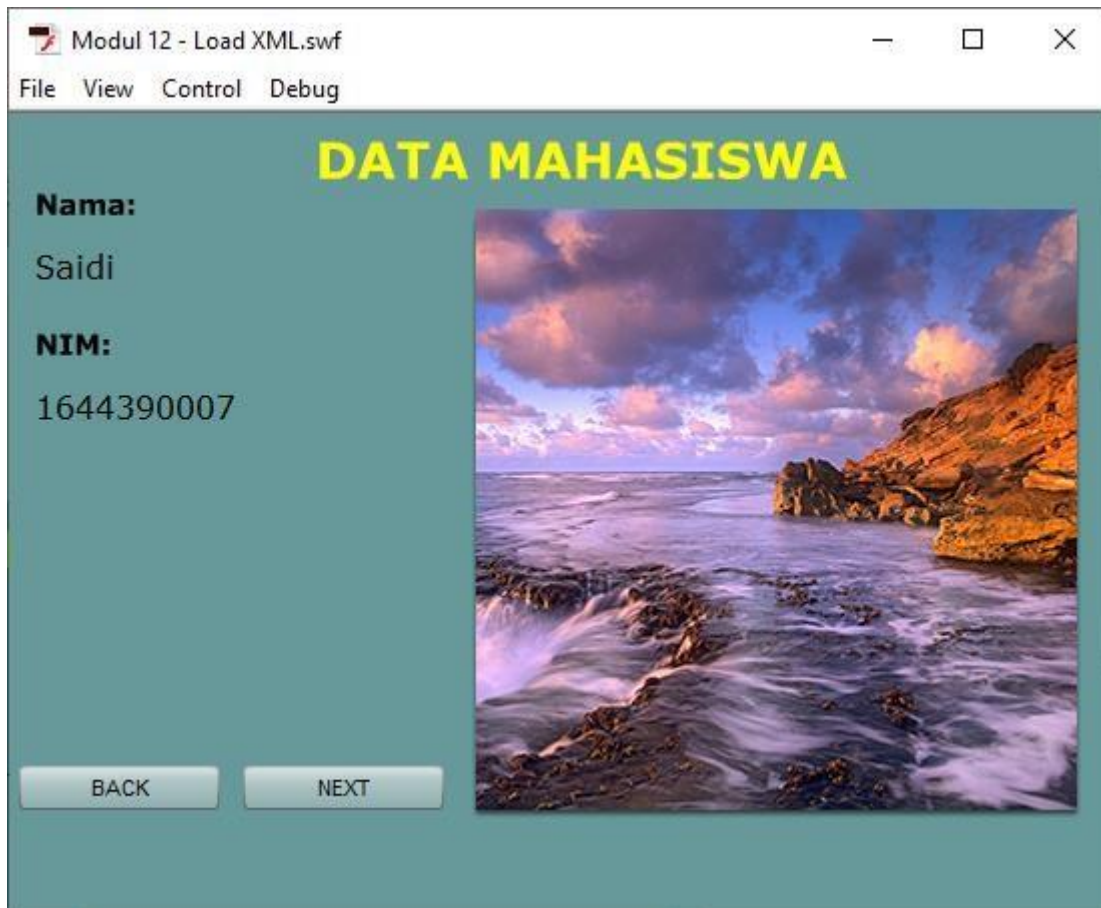
Contoh:

```
displaynama.text = myXML.child(1).child(0);
displaynim.text   = myXML.child(1).child(1);
displayfoto.source = myXML.child(1).child(2);
```

Akan menghasilkan tampilan sebagai berikut:



Nah, selanjutnya adalah membuat Button BACK dan NEXT untuk mengurangi atau menambah nomor indeks pada child pertama tadi.



Demikian penjelasan tentang XML, mudah-mudahan bermanfaat.