

# Pengantar SWT (Standard Widget Toolkit)

**Mohammad Athar Januar**

attarinho@hotmail.com

http://attarinho.merantau.com

## ***Lisensi Dokumen:***

Copyright © 2003- 2006 IlmuKomputer.Com

Seluruh dokumen di **IlmuKomputer.Com** dapat digunakan, dimodifikasi dan disebarakan secara bebas untuk tujuan bukan komersial (*nonprofit*), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari **IlmuKomputer.Com**.

## **Abstrak:**

Tulisan ini berisikan tentang SWT (*Standard Widget Toolkit*), sebagai GUI (*Graphical User Interface*) di dalam Eclipse Platform. SWT merupakan produk IBM, begitu juga Eclipse sendiri dan sampai saat ini sangat diterima di kalangan pengembang Java. SWT dapat dikembangkan sebagai *stand-alone application*, *rich-client application*, pengembangan *plugin* untuk Eclipse sendiri, dan lain-lain.

**Keywords:** Eclipse, SWT, Standard Widget Toolkit, Java, Plug-in, JFace

## **1. LATAR BELAKANG**

Tutorial ini merupakan salah satu bab dari *study thesis* penulis yang mengembangkan sebuah *plugin* untuk sebuah projek ECN (*Event Channel Network*) di FZI (*Forschungszentrum Informatik*, Pusat Penelitian Informatika) di Karlsruhe, Jerman. Tutorial berikutnya dengan tema yang baru akan diterbitkan juga sejalan dengan penulisan *study thesis* penulis.

Pembaca diharapkan telah memiliki pengetahuan dasar Java, seperti Java GUI (AWT atau Swing) dan memiliki pengalaman dalam menggunakan Eclipse.

## **2. PENDAHULUAN**

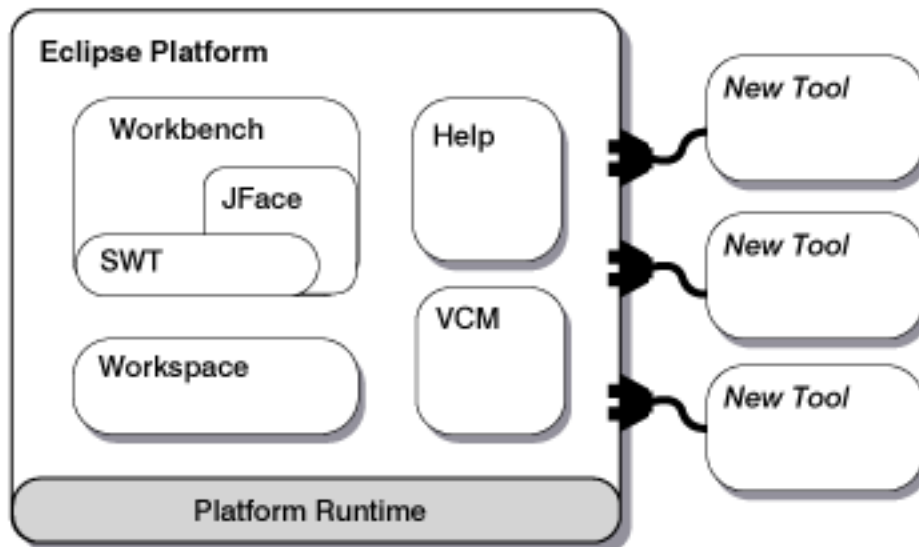
Pada tahun 2001 IBM meluncurkan Eclipse Platform sebagai satu kompetitor baru dalam persaingan IDE (*Integrated Development Environment*) Java. Kabarinya, IBM telah berinvestasi sebanyak 40 juta US\$ dalam pengembangan Eclipse Platform.

Eclipse dilengkapi dengan *GUI-library* baru bernama SWT (*Standard Widget Toolkit*), yang juga mulai

menggeser dominasi AWT (*Abstract Window Toolkit*) dan Swing yang diluncurkan oleh SUN Microsystem.

### 3. APAKAH SWT ITU?

SWT pada dasarnya adalah *GUI-library* yang terintegrasi di dalam Eclipse sebagai fondasi GUI. Dalam hirarki Platform [3], SWT berada di dalam *workbench* Eclipse (gambar 1), yang merupakan dasar tampilan Eclipse tersebut. SWT secara khusus memang dikembangkan untuk Eclipse.



**Gambar 1 : Arsitektur Eclipse**

Menurut situs resmi [Eclipse/SWT](#) [2] definisi SWT adalah:

*“SWT is an open source widget toolkit for Java designed to provide efficient, portable access to the user-interface facilities of the operating systems on which it is implemented.”*

Walau masih relatif baru, SWT mulai menggoyang dominasi AWT dan Swing, karena beberapa kelebihanannya, seperti:

- Komponen SWT didesain untuk efisiensi, akses yang portable terhadap fasilitas UI (*user interface*) di mana SWT diimplementasi dan dikembangkan.
- SWT lebih cepat dibandingkan Swing.
- *Native look & feel*, di platform mana pun SWT berada, tampilannya telah disesuaikan dengan platform tersebut, seperti Linux, OS X dan Windows.
- Desain yang bersih, sederhana, dan simpel.
- *Open source*, siapa pun boleh melihat *source code* dari SWT dan boleh mengembangkan sendiri.

Beberapa kelemahan SWT seperti kurangnya portabilitas, dalam arti pengembang mesti terlebih dahulu men-*download* SWT. Umurnya yang masih sangat muda, mengakibatkan dokumentasi, buku-buku terhitung sedikit. Akan tetapi SWT merupakan alternatif yang sangat menarik untuk dikembangkan, terutama dalam pengembangan Java-GUI menggunakan Eclipse.

Saat ini banyak didiskusikan dan diperdebatkan tentang kelebihan masing-masing Java GUI. Tutorial ini tidak akan membahas perdebatan Java GUI, akan tetapi untuk pembaca yang tertarik dapat

membaca referensi :

- [1] Ozgur Akan, *java.net* : [Why I choose SWT against Swing](#)
- [5] Barry Feigenbaum, *IBM* : [SWT, Swing or AWT : Which is right for you?](#)

#### 4. STRUKTUR SWT

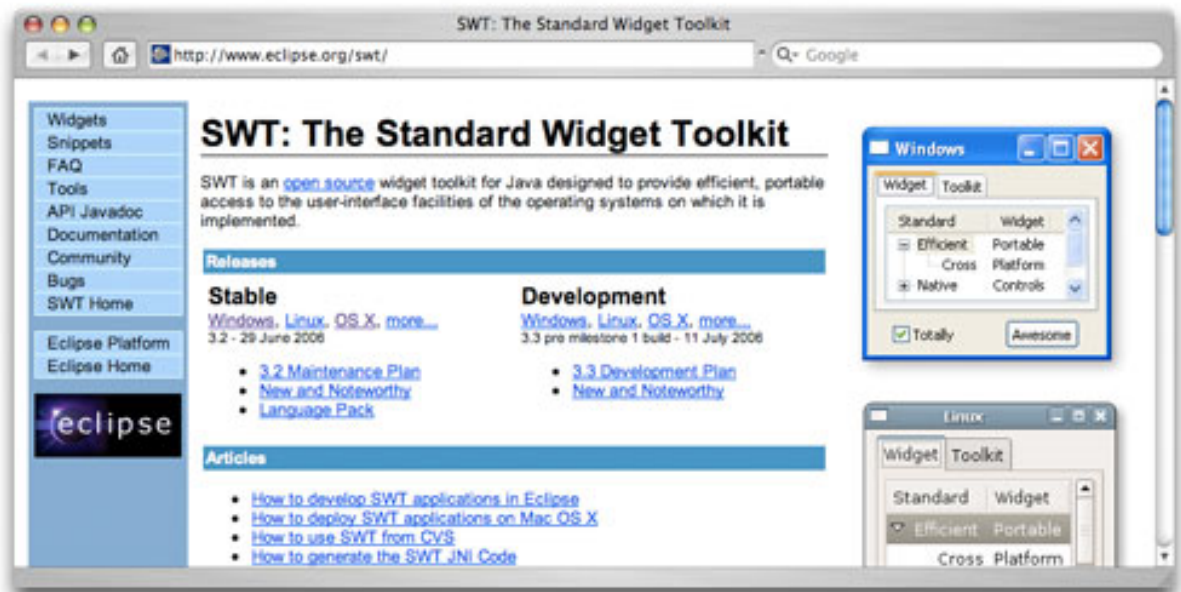
SWT memiliki struktur [4] yang lebih kompleks ketimbang AWT dan Swing. Berikut ini istilah-istilah penting SWT :

- *Widget* : Komponen dasar dari SWT, abstrak.
- *Control* : Widget yang memiliki basis dari system operasi, seperti *Button*, *Label*, *Text*, dan lain-lain.
- *Composite* : Control yang dapat mempunyai control lainnya.
- *Item* : Widget yang berada di dalam Control, seperti *List*, *Table*, dan lain-lain.

#### 5. "HELLO WORLD" DENGAN SWT

Penulis mengembangkan SWT dengan menggunakan Eclipse Platform 3.1.2, di Macintosh dengan OS X v. 10.4 ("Tiger"). Beberapa hal perlu diset terlebih dahulu sebelum mengembangkan aplikasi SWT.

- Instal Eclipse dari situs resmi [Eclipse](#).
- *Download SWT-library* dari situs resmi [Eclipse/SWT](#).  
Pilih platform (OS X, Linux, Windows) yang digunakan di bagian **Releases > Stable** (gambar 2).



**Gambar 2 : Situs Eclipse/SWT untuk download SWT-library**

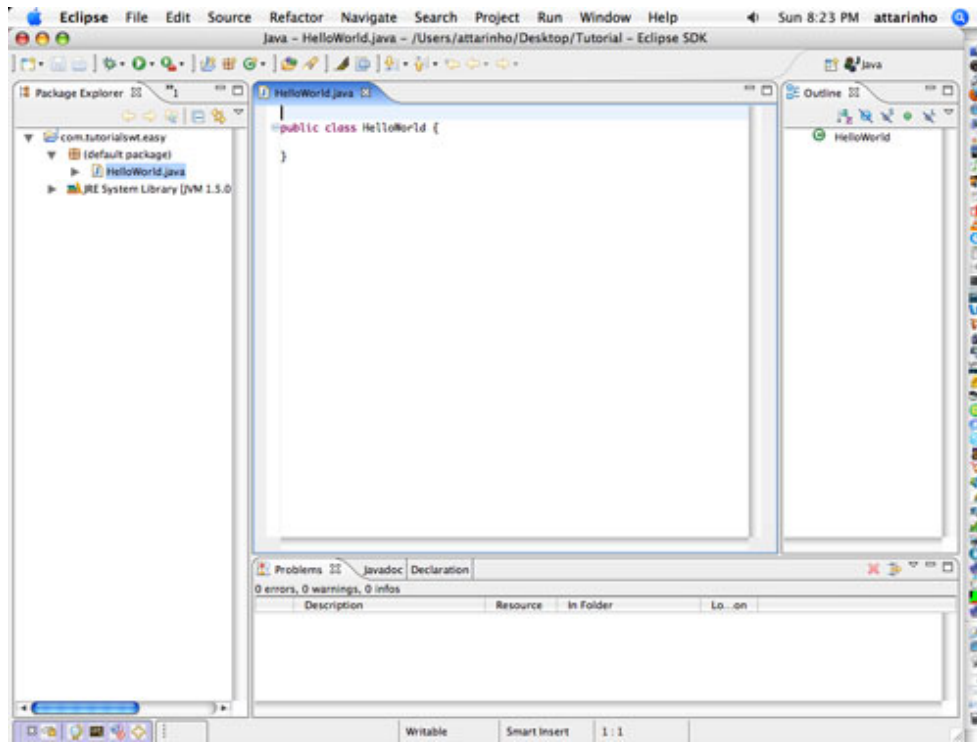
- Setelah download, *unpack* file tersebut di dalam direktori plugins dari Eclipse.

Sekarang Eclipse bisa dijalankan. Untuk mengembangkan SWT di dalam Eclipse dimulai dengan :

- Membuat **Java Project**  
**File > New > Project...**
- Kemudian akan muncul *project wizard*, pilih **Java Project**, kemudian tekan tombol **Next**.
- Eclipse akan meminta nama projek, ketik di bagian *Project name* : **com.tutorialswt.easy**, kemudian tekan tombol **Finish**.

Berikutnya, membuat satu kelas Java di dalam projek tersebut :

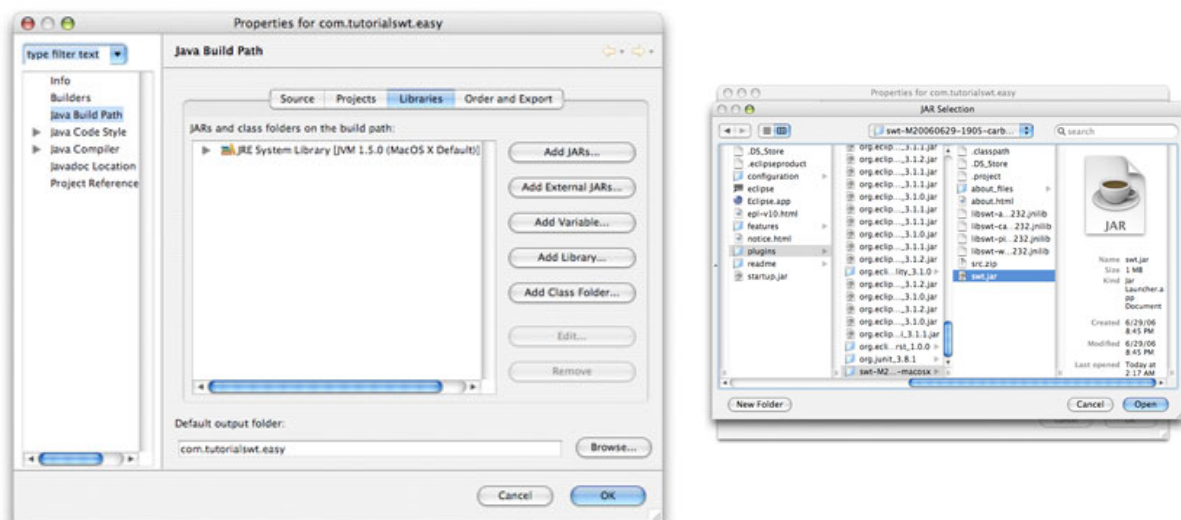
- Membuat **Java Class**  
**File > New > Class**
- Akan muncul juga sebuah *wizard*, masukkan nama kelas : **HelloWorld**, kemudian tekan tombol **Finish**. Eclipse akan terlihat seperti gambar 3 :



Gambar 3 : Projek *com.tutorialswt.easy* dengan kelas *HelloWorld.java*

Sebelum mulai menulis kelas ini, harus ada yang diset terlebih dahulu (gambar 4 dan 5):

- Mengimpor **swt.jar** ke dalam proyek  
**Project > Properties > Java Build Path > Libraries > Add External JARs...**
- Cari **swt.jar** di direktori *plugins*, seperti gambar di bawah ini, kemudian tekan tombol **Finish**.



Gambar 4 & 5 : Menambahkan *swt.jar*

Di dalam kelas **HelloWorld**, implementasikan kode :

```
import org.eclipse.swt.*;
import org.eclipse.swt.widgets.*;

public class HelloWorld {
    public static void main (String[] args) {

        Display display = new Display();
        Shell shell = new Shell(display);
        shell.setText("My First SWT Application");
        shell.setSize(300, 100);

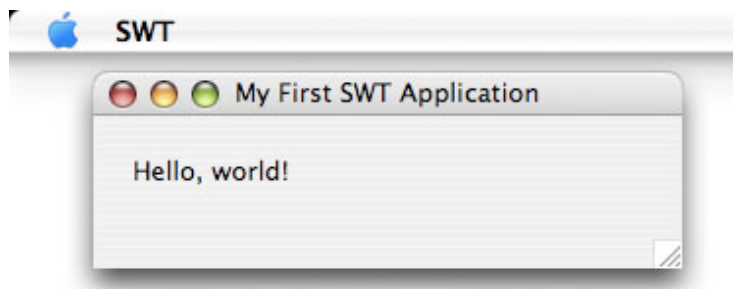
        Label label = new Label(shell, SWT.NONE);
        label.setText("Hello, world! ");
        label.setSize(100, 50);
        label.setLocation(20, 20);

        shell.open();
        while (!shell.isDisposed()) {
            if (!display.readAndDispatch())
                display.sleep();
        }
        shell.dispose();
    }
}
```

Untuk mengeksekusi kelas ini :

- **Run > Run... > Java Application > Arguments**
- Di dalam **VM arguments** ketik:  
-Djava.library.path=/Application/eclipse/plugins/swt-M20060629-1905-carbon-macosx/
- Untuk platform lainnya :  
**Windows** : <INSTALLDIR>\eclipse\plugins\org.eclipse.swt.win32\_2.1.0\os\win32\x86  
**Linux GTK** : <INSTALLDIR>/eclipse/plugins/org.eclipse.swt.gtk\_2.1.0/os/linux/x86
- Kemudian tekan tombol **Run**.

Kemudian window SWT akan tampil seperti gambar 6 di bawah ini :



**Gambar 6 :**  
*HelloWorld.java*, SWT yang paling sederhana

## 6. MEMBEDAH “HELLO WORLD”

Untuk dapat menggunakan komponen SWT, *library* harus diimport terlebih dahulu :

```
import org.eclipse.swt.*;  
import org.eclipse.swt.widgets.*;
```

Implementasi berada di dalam *main*-metode :

```
Display display = new Display();  
Shell shell = new Shell(display);
```

Sebuah `display` adalah objek yang mengandung semua komponen GUI. `display` pada umumnya tidak terlihat (*invisible*), hanya komponen di dalamnya yang terlihat. Biasanya sebuah aplikasi hanya memiliki satu `display`.

Sebuah `shell` adalah jendela (window) di dalam aplikasi.

```
shell.setText("My First SWT Application");  
shell.setSize(300, 100);
```

Untuk menambahkan label di atas `shell` dan mendefinisikan *style* dari label tersebut.

```
Label label = new Label(shell, SWT.NONE);  
label.setText("Hello, world! ");
```

Kode di bawah untuk menjalankan `shell` dan *event loop*. *Event loop* memiliki fungsi untuk membuat `shell` *visible*. Apabila `shell` ditutup, `display` harus dihapus. Kode ini selalu ada di setiap kelas yang mengembangkan SWT.

```
shell.open();  
while (!shell.isDisposed()) {  
    if (!display.readAndDispatch())  
        display.sleep();  
}  
shell.dispose();
```

Begitu secara singkat cara mengimplementasikan SWT yang sederhana.

## 7. KESIMPULAN

Eclipse Platform sebagai IDE yang sangat flexibel dan kompak mulai menjadi IDE favorit di kalangan pengembang Java. Selain itu, Eclipse menawarkan alternatif GUI dengan *SWT-library*-nya yang sangat terintegrasi di dalam Eclipse, akan tetapi Eclipse harus diset terlebih dahulu untuk dapat mulai mengembangkan aplikasi SWT. Untuk kedepannya, dengan Eclipse diharapkan SWT lebih sering digunakan ketimbang AWT atau Swing, akan tetapi masing-masing tetap memiliki kelebihan dan kekurangan.



## 8. BIBLIOGRAPHI

- [1] Ozgur Akan. *Why I choose SWT against Swing*. java.net 2004  
[http://weblogs.java.net/blog/aiga/archive/2004/11/why\\_i\\_choose\\_sw.html](http://weblogs.java.net/blog/aiga/archive/2004/11/why_i_choose_sw.html)
- [2] Eclipse Foundation  
<http://www.eclipse.org/swt/>
- [3] Marc Erickson. *Working the Eclipse Platform*. IBM 2001.  
<http://www-128.ibm.com/developerworks/opensource/library/os-plat/>
- [4] Barry Feigenbaum. *A gentle introduction to SWT and JFace*. IBM 2005  
<http://www-128.ibm.com/developerworks/opensource/library/os-jface1/>
- [5] Barry Feigenbaum. *SWT, Swing or AWT: Which is right for you*. IBM 2006  
<http://www-128.ibm.com/developerworks/opensource/library/os-swingswt/>

## 9. REFERENSI

- Department of Computer Science, University of Manitoba, Canada. *Installing Eclipse*. 2003
- Holzner, Steve. *Eclipse*. O'Reilly 2004.

## 10. BIOGRAPHI PENULIS



**Mohammad Athar Januar.** Angkatan 1999 SMUN 78 Jakarta. Sekarang sedang kuliah computer science program diplom di universitas Karlsruhe, Jerman. Pernah magang di sebuah institut kampus, sebagai asisten peneliti, selain itu juga aktif menjadi tutor Java. Hobby olahraga, dengar musik, nonton film dan pokoknya yang asik-asik. Mendalami bidang *software engineering*, sedang memulai *study thesis* dalam rangkaian kuliah menjelang *diplom thesis*, dengan tema “*Eclipse Plugin für ECN*” di pusat penelitian informatika (FZI, *Forschungszentrum Informatik*) di Karlsruhe, Jerman. Tertarik di bidang *web-development* menggunakan J2EE, dan juga pengembangan software menggunakan teknologi Java. Senang berbagi pengalaman *programming* kepada yang sedang mulai belajar, menjadi alasan untuk rajin menulis dan tutor di kampus. Di waktu senggang, ikut mengasuh portal informasi kuliah ke luar negeri [merantau.com](http://merantau.com), dan juga mengaktualisasikan situs pribadi [merantau.com/attarinho](http://merantau.com/attarinho).