



# Otodidak Behind The Scene

Buku Satu

Panduan Lengkap Membuat Aplikasi GUI dengan Netbeans  
Khusus untuk Pemula dan Orang Awam

**Judul buku:**

*Otodidak Behind The Scene Seri I: Panduan Lengkap Membuat Aplikasi GUI dengan Netbeans Khusus untuk Pemula dan Orang Awam*

**Tujuan buku:**

*mengajari semua orang Indonesia terutama yang berusia muda untuk membuat software aplikasi Linux*

**Platform:**

*Linux, Windows, dan Mac OS X*

**Bahasa pemrograman:**

*Java*

**Banyak seri:**

*tidak diketahui*

**Harga buku:**

*gratis*

**Lisensi buku:**

*halal disebarluaskan asal tidak untuk komersial*

**Persenjataan layout buku:**

*Libreoffice di atas mesin Ubuntu 12.04*

**Software desain kover:**

*Inkscape*

**Penerbit:**

<http://malsasa.wordpress.com>

# Isi Buku Ini

- Semangat!
- Pembukaan
- Bagaimana Cara membuat Software?
- Apa Yang Harus Dimiliki Untuk Membuat Software?
- Penjelasan Mengenai Peralatan Kita
- Mulai Mengenal Netbeans
- Membuat Proyek Baru
- Membuat Frame
- Penjelasan Dari Perkenalan Dengan Netbeans
- Membongkar Tampilan Utama Otodidak
- Membuat Aplikasi Pertama Anda
- Pra-Deploy
- Deploy!
- Mengenal Netbeans (Lagi)
- Masalah 1
- Trik Netbeans 1
- Rangkuman 1
- Latihan 1
- Penutup Buku Satu

## *Jangan Minder!*



Hal paling awal yang mesti Anda miliki untuk memulai pemrograman adalah mental. Jangan minder dengan para programmer yang mendahului kita dengan segala kemampuan pemrogramannya. Jangan minder pula jika Anda termasuk orang yang tidak memiliki komputer, Anda masih punya kesempatan! Saya sendiri membuat Otodidak dengan memulai desainnya dalam buku tulis sekolah biasa 58 lembar. Otodidak dibuat dari desain-desain, dan itu saya tulis manual di dalam buku ini. Dan foto halaman berikut adalah halaman pertama catatan Otodidak Behind The Scene yang asli.

Progres Java C# Chaido , Malas 6 November  
7 November 2012

- Bisa JFrame
- Label Gasing bisa teks bisa gambar
- TextArea bisa teks baris-baris
- Button bisa diklik
- Button bisa diberi logo
- Button bisa membuka JFrame lain
- ```
if (btAnu.isEnabled() == true) {  
    new JFrameAnu().setVisible(true);  
}
```

  
// pasti bisa membuka JFrameAnu
- ~~checkbox~~ bisa di-cetak
- ~~checkbox~~ bisa menguasai textfield
- ```
if (cbAnu.isSelected() == true) {  
    tfAnu.setEnabled(true);  
    tfAnu.setText("");  
}
```
- Event > ~~setEnabled~~ actionPerformed
- String bisa diubah jadi int      Integer Anu = Integer.valueOf(VarString);
- Int bisa diubah jadi string      .setText("" + AnuInt);
- TextField bisa mengirim data ke variabel
- Kode pengiriman data malah ditulis di Button
- ```
String JumlahSoto; ← deklarasi luar  
JumlahSoto = tfSoto.getText();
```
- Belum bisa membuang border dari TextArea
- Belum bisa multiple line text di label
- Belum bisa klik tombol, JFrame berakhal, tanpa buka baru
- Belum bisa progress bar sebelum aplikasi nyala

Better try than never

Buku ini hanyalah catatan setiap saya berhasil atau gagal membuat kode Java

# Pembukaan

Dengan mengharap rahmat Allah, saya membuka buku seri pertama Otodidak Behind The Scene ini. Buku ini saya susun sebagai panduan lengkap membuat *software* aplikasi sendiri untuk pemula dan orang awam, berdasarkan proses pembuatan aplikasi Otodidak versi 1. Ringkasnya, buku ini membocorkan bagaimana developer membuat aplikasinya sendiri. Tujuan buku ini adalah mengajak pemuda Indonesia dan teman-teman saya di SMK/SMP untuk membuat *software* sendiri, dengan **cara yang paling mudah**. Buku ini hanyalah *proof of concept* bahwa membuat *software* aplikasi GUI itu sangat mudah. Buku ini akan diterbitkan berseri dan pada akhir seri buku, diharapkan semua pembaca dapat membuat *software* aplikasinya masing-masing. Semoga buku ini mengawali lahirnya buku-buku lain atau paling tidak aplikasi-aplikasi lain buatan teman-teman sekalian di seluruh Indonesia.

Sebagai peringatan paling mendasar, saya ingin mengatakan:

***Yang ditekankan dalam Otodidak Behind The Scene adalah membuat software secara drag and drop, bukan menulis kode-kode.***

Mengapa saya memilih jalan ini? Karena inilah jalan termudah bagi saya sendiri untuk membuat Otodidak dan bagi Anda juga untuk membuatnya.

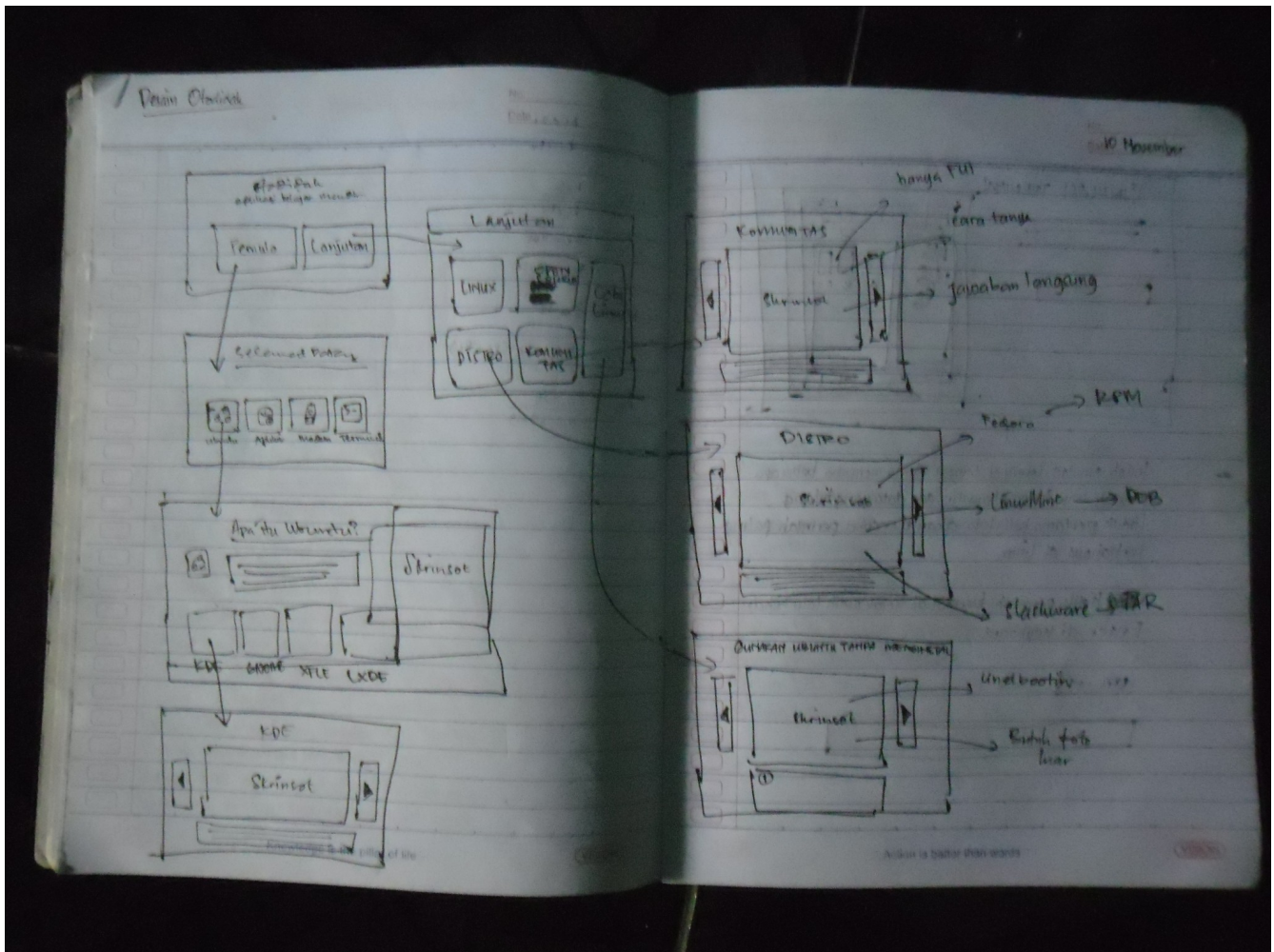


# Bagaimana Cara Membuat Sebuah Software?

Ya, bagaimana caranya? Banyak sekali pengguna awam yang tidak tahu. Saya akan bocorkan rahasianya.

- 1) Desain
- 2) Melakukan pemrograman
- 3) Mengetes *software*
- 4) Memaketkan *software*

Dengan 4 langkah inilah saya membuat Otodidak. Dan dengan 4 langkah global inilah semua *software* termasuk game diciptakan. Dan segalanya diawali dengan desain, begitu pula Otodidak. Saya menuliskan desain Otodidak di sebuah buku tulis SIDU 58 lembar untuk sekolah saya. Apa sih desain itu? Apa saja isinya? Untuk sementara, silakan lihat foto halaman desain saya:



Tidak ada aturan yang baku untuk membuat desain. Maka itu pembuatan *software* seperti Otodidak sangatlah menyenangkan, karena kebebasan mutlak milik kita. Anda bisa melakukan desain *software* sendiri dengan mendasarkan pada foto ini atau dengan menulis. Misal:



- 1) Nama *software*
- 2) Tanggal awal desain
- 3) Tujuan *software*, misalnya untuk belajar Linux
- 4) Bahasa pemrograman yang mau dipakai, misalnya Java
- 5) Kira-kira bagaimana tampilannya, gambar sendiri
- 6) Bagaimana tampilan *software* Anda akan bekerja, beri panah seperti gambar di atas.

Hanya saja, satu pegangan dalam membuat desain adalah:

**tuliskan semua ide tetapi nanti pakailah yang paling bagus saja.** Dengan pegangan inilah saya membuat desain sebarang mungkin di buku itu.

# Apa Yang Harus Dimiliki Untuk Membuat *Software*?

Pertanyaan sederhana yang bikin penasaran pengguna pemula. Apa saja yang saya siapkan untuk membuat Otodidak?

- Sistem operasi: Ubuntu 12.04
- Mesin: laptop ASUS X44C
- Hardware inti: mouse dan keyboard
- Aplikasi pembuat *software*: Netbeans versi 7.1.2
- Aplikasi bantu: GIMP, Inkscape, Gedit, Chromium Browser,

Itulah persenjataan saya. Sampai sini, pasti ada pembaca yang bingung. Mungkin karena belum kenal Netbeans, Inkscape, dan teman-temannya. Tidak mengapa. Perlahan akan saya jelaskan. Sebelumnya, Anda harus mengenal ini: **aplikasi pembuat *software***. Anda bisa menyebutnya: ***software* pembuat *software***. Apa ini? Inilah inti dari segala proses Anda nanti dalam membuat *software*. Anda akan:

- memrogram
- mengetik kode-kode pemrograman
- membuat antarmuka GUI dengan drag and drop
- memasukkan gambar ke dalam *software* Anda (seperti dalam Otodidak)
- menjalankan *software* Anda
- mengetes *software*

di dalam aplikasi pembuat *software* ini. Jika Anda senang membaca, maka Anda akan menyadari yang saya maksud dengan ini adalah Integrated Development Environment atau yang biasa disebut IDE. Ya, kita membutuhkan *software* untuk membuat *software* lain. *Software* macam inilah yang disebut IDE, kita memasukkan kode-kode pemrograman di sana. Mengapa saya jelaskan ini dulu? Karena semenjak awal saya tahu adanya pembuatan *software*, saya tidak tahu kode-kodenya harus diketik di mana. Ketidaktahuan ini berlangsung selama beberapa tahun sementara tidak ada yang mengajari saya. Cukup buruk, bukan? Oleh karenanya, demi memangkas waktu belajar Anda, ketahuilah apa itu IDE.

Bagaimana tampilan sebuah IDE? Apakah IDE hanya satu? Atau ada banyak? Tampilan IDE seperti gambar setelah paragraf ini. IDE ada banyak. Dan biasanya IDE itu ada berdasarkan bahasa pemrograman yang digunakan. Misalnya, pemrograman dalam Bahasa Java, maka IDE yang dipakai yang untuk Java. Pemrograman dalam Bahasa Pascal, maka IDE yang dipakai yang untuk Pascal. Tidak bisa IDE khusus Java dipakai untuk membuat *software* dalam Bahasa Pascal. Nah, IDE itu mengikuti bahasa pemrograman. Dan saya tunjukkan nama-nama IDE yang paling terkenal di dunia: Netbeans untuk Bahasa Java, DevC++ untuk Bahasa C atau C++, Microsoft Visual Basic Studio untuk Bahasa Visual Basic, dan FreePascal untuk Bahasa Pascal. Sebenarnya ada ratusan lagi IDE lain yang bagus pada setiap bahasa pemrograman tetapi tidak mungkin saya sebut semua di sini.

# Penjelasan Mengenai Persenjataan Kita

- 1) Apakah saya wajib memiliki semuanya?
- 2) Di mana saya bisa memperolehnya?
- 3) Apa yang harus dilakukan dengan itu semua?

Tentu saja tidak. Yang harus dimiliki hanya Netbeans. Ingat, dalam membuat Otodidak saya memakai versi 7.1.2. Anda tidak harus menggunakan Ubuntu. Windows pun bisa. Namun, untuk membuat Otodidak, paling nyaman menggunakan Ubuntu. Mengapa? Karena tumpuan Otodidak hanyalah gambar dan ikon-ikon. Semua itu saya dapatkan dari Ubuntu. Terutama untuk ikon, saya tidaklah membuatnya dari nol. Saya cukup mengambil ikon yang disediakan Ubuntu di direktori `/usr/share/icons/`. Proyek Otodidak ini juga bertujuan mendayagunakan ikon-ikon di situ biar bermanfaat untuk orang lain.

Netbeans → <http://netbeans.org>

Inkscape → <http://inkscape.org>

GIMP → <http://gimp.org>

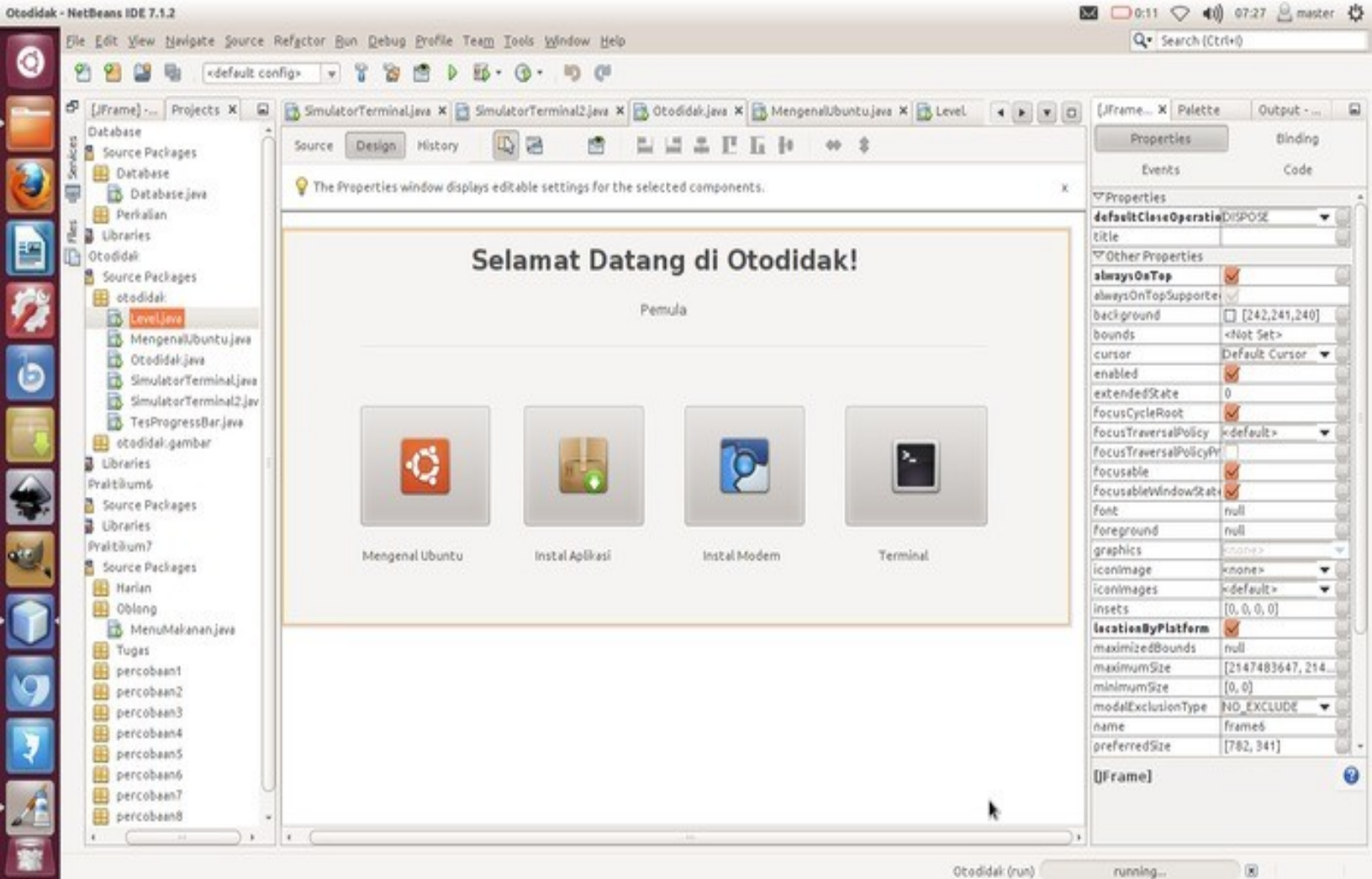
Pinta → <http://pinta-project.com/>

Gedit → sudah ada di Ubuntu dan bisa digantikan dengan Notepad di Windows

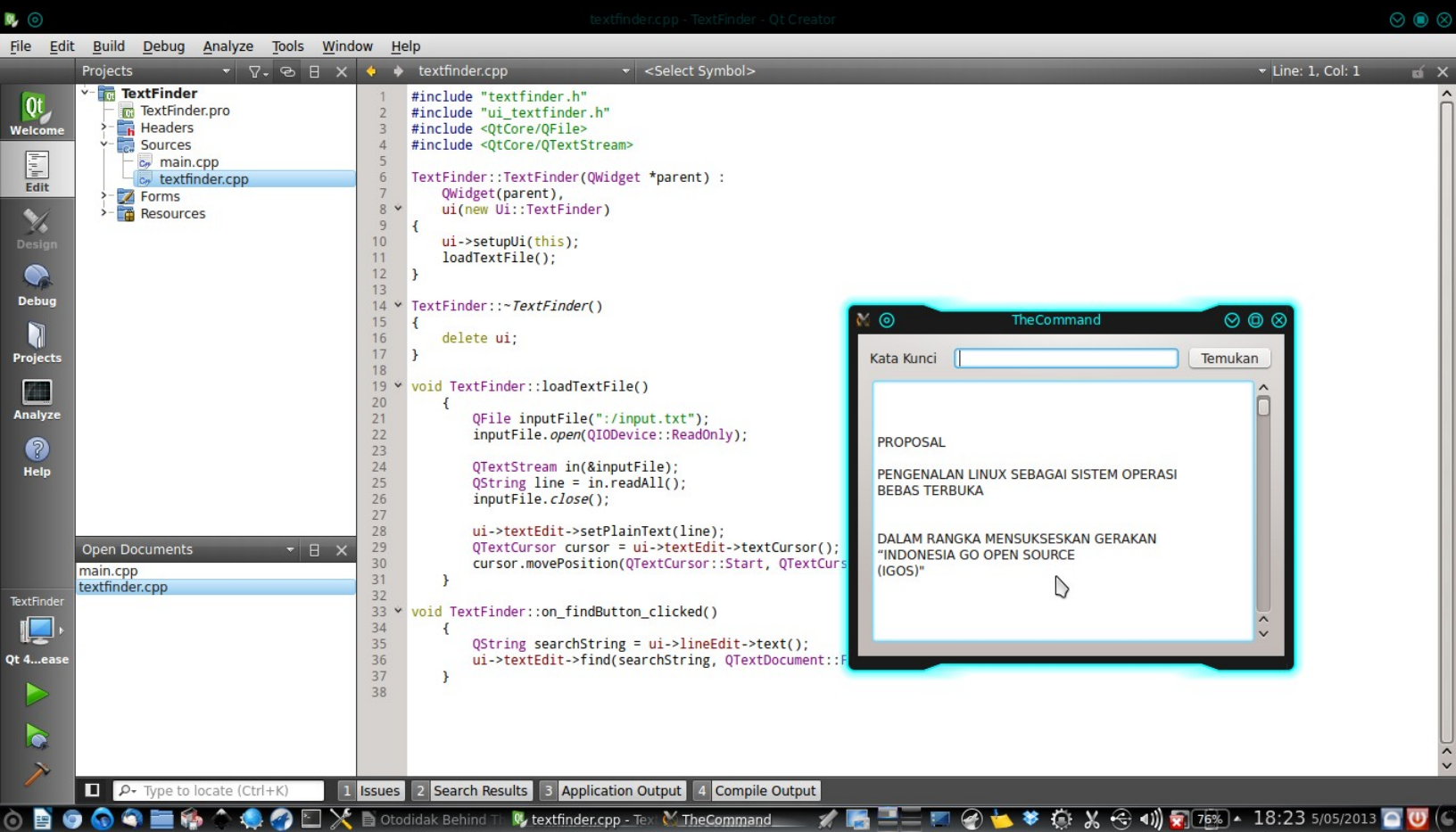
Chromium Browser → <http://www.chromium.org/> bisa diganti dengan Firefox/Chrome

Silakan kunjungi web tersebut untuk memperoleh programnya. Jika Anda menggunakan Linux, sebaiknya unduh Netbeans 7.1.2 yang untuk Linux. Unduh versi yang sama untuk Anda yang memakai Windows. Saya instal Netbeans bukan dari repo resmi Ubuntu tetapi dari berkas `.sh` yang saya unduh dari web resmi Netbeans. Ingat, ukurannya sekitar 70 MB.

**Catatan:** untuk pengguna Ubuntu, saya rekomendasikan Netbeans 7.1.2 dari web resminya. Sebaiknya tidak instal dari repo resmi karena hanya tersedia versi 7.0.1 (saat tulisan ini dibuat) dan Matisse mengalami hang saat saya coba. Hematlah bandwidth Anda.

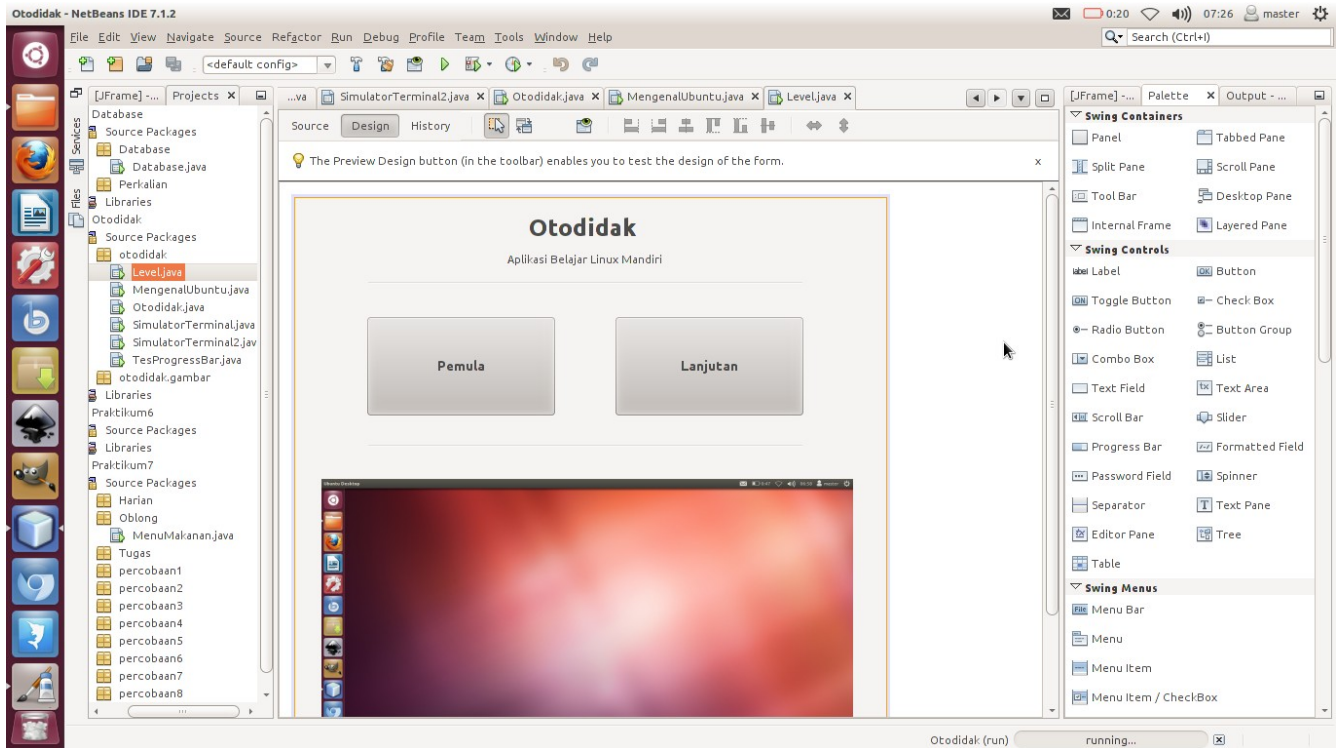


Yang Anda lihat di atas adalah Netbeans, IDE untuk Java. IDE inilah yang saya pilih untuk membuat Otodidak. Yang di bawah ini namanya QtCreator, IDE untuk Bahasa C++.



# Mulai Mengetahui Netbeans

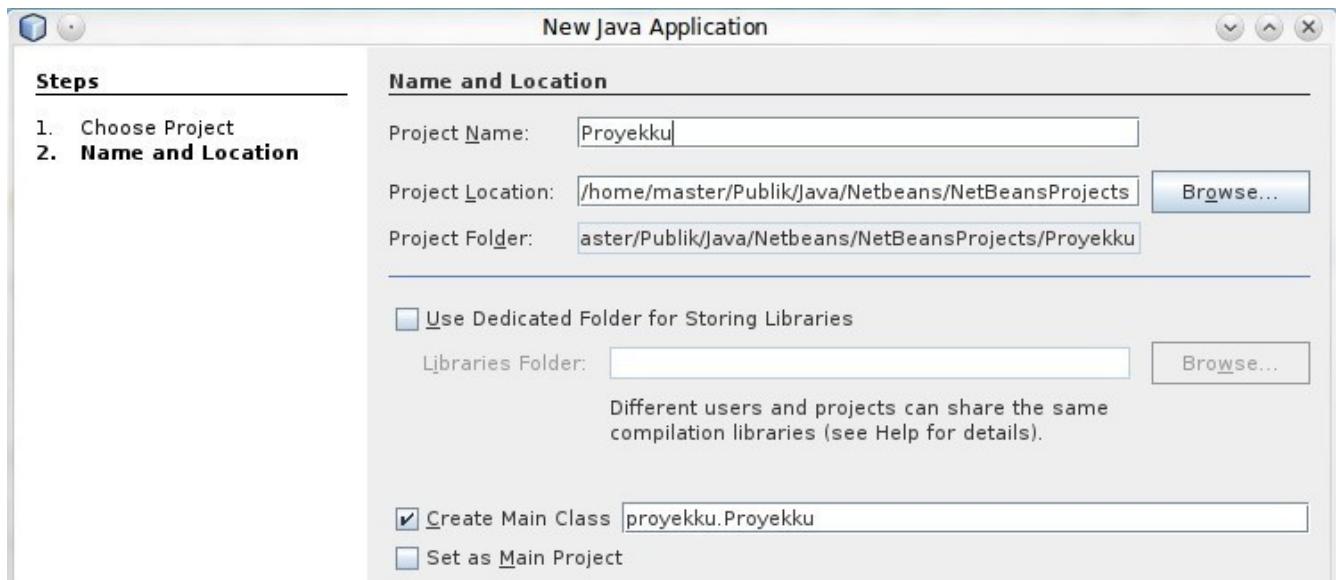
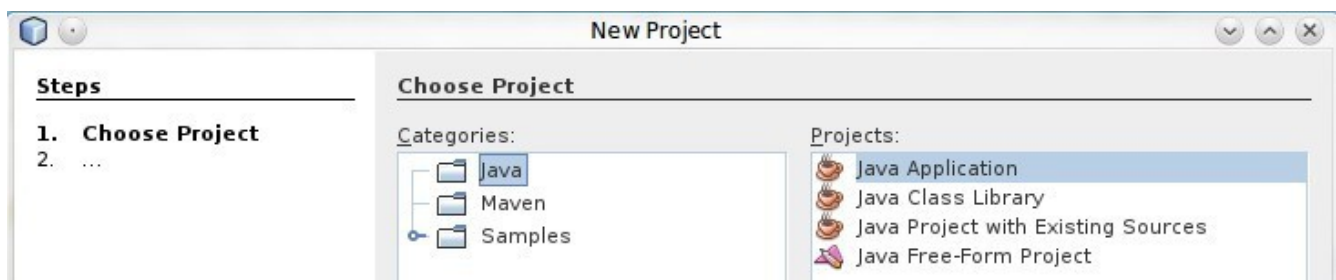
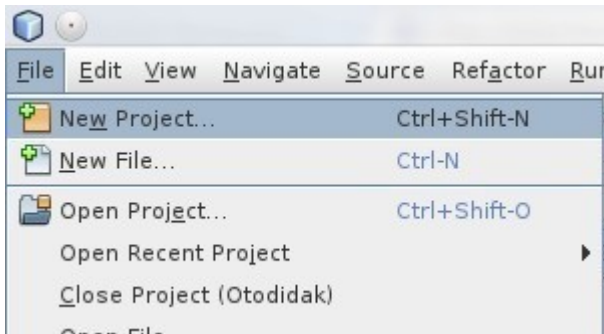
Buku ini ditujukan hanya untuk dasar pengenalan, sehingga dominan pengenalan. Anda harus mengenal aplikasi Netbeans dahulu untuk tahu di mana Anda harus bekerja dalam Netbeans. Mengapa? Karena fitur Netbeans sangat banyak, sedangkan untuk membuat *software* macam Otodidak hanya dibutuhkan 1/100-nya mungkin. Janganlah Anda merabakan sendiri karena itu menghabiskan waktu. Biarkan saya bocorkan informasinya untuk Anda.



Saya tidak akan menjelaskan satu per satu antarmuka Netbeans karena itu tidak efektif. Saya hanya jelaskan globalnya dan merinci pada apa yang dilakukan untuk membuat aplikasi GUI.

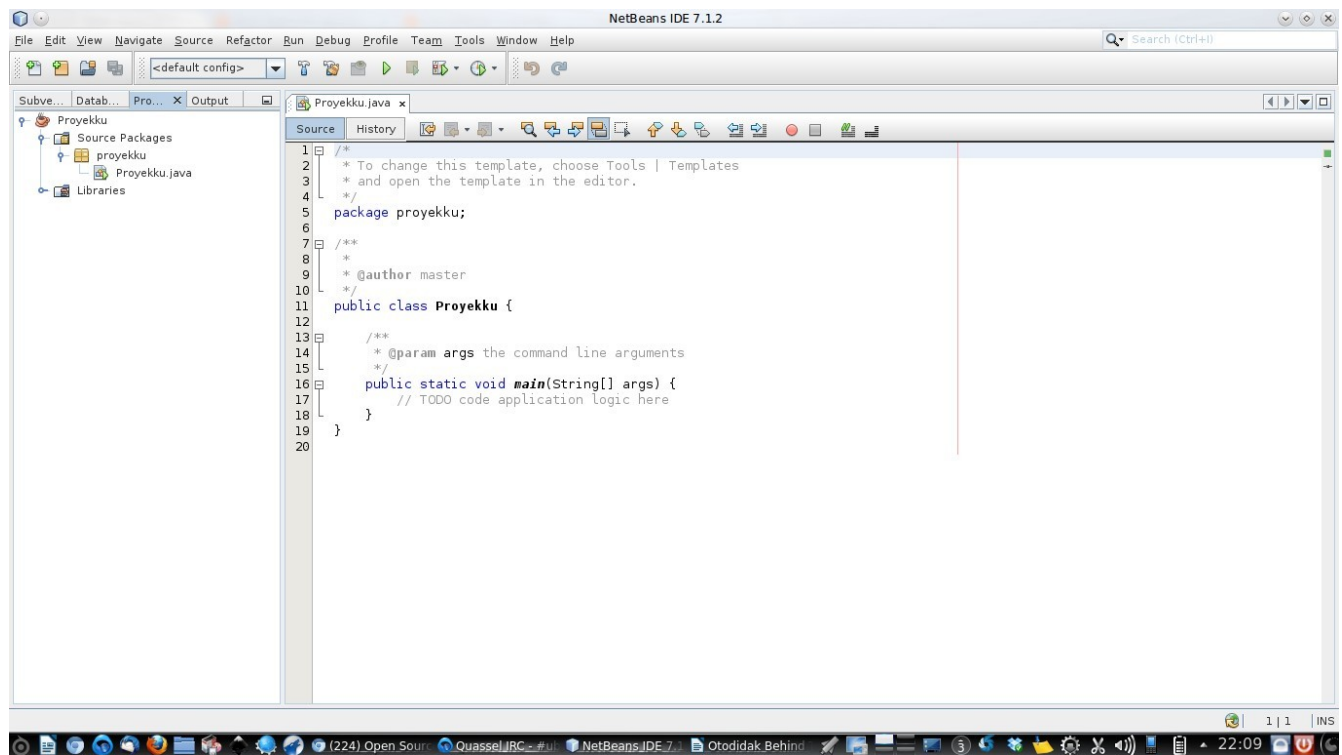
# Membuat Proyek Baru

Ringkas waktu Anda dengan mengikuti ini.



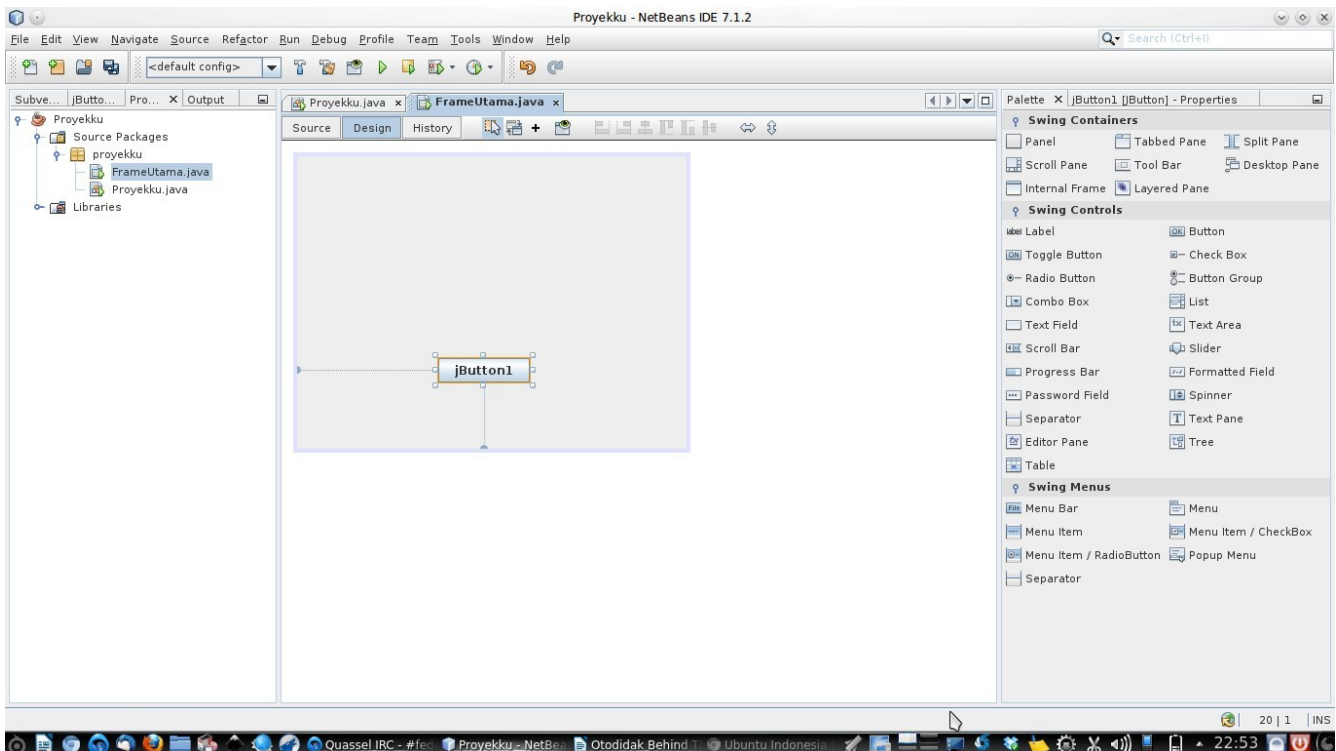
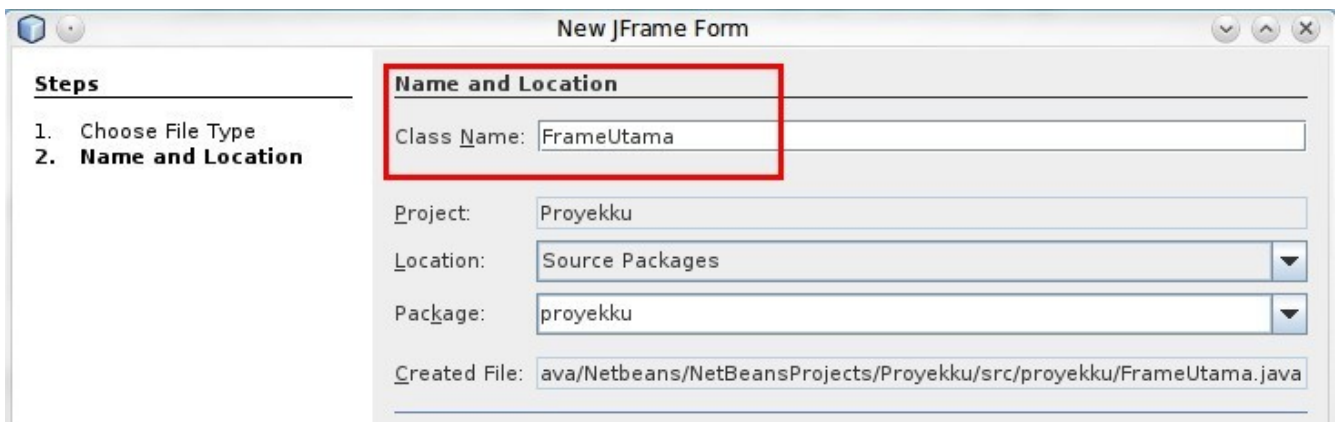
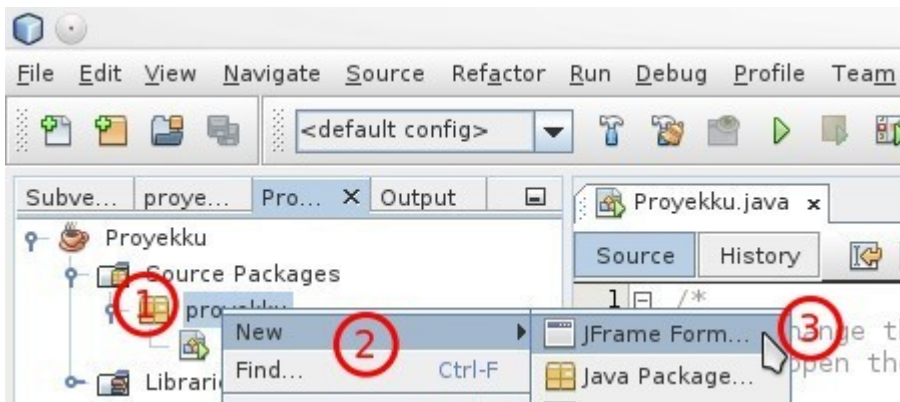


## Hasil pembuatan proyek baru Netbeans



Anda jika baru kali ini mengenal pemrograman, selamat datang. Inilah bahasa pemrograman Java, inilah Netbeans. Saya janji tidak akan jelaskan tombol Netbeans satu-satu. Itu tidak efektif. Sekarang mulai masuk ke bagian pertama pembuatan aplikasi, bagian terpenting.

# Membuat Frame



Tujuan ini semua hanyalah skrinshot terakhir ini. Apa ini? Akan saya jelaskan.

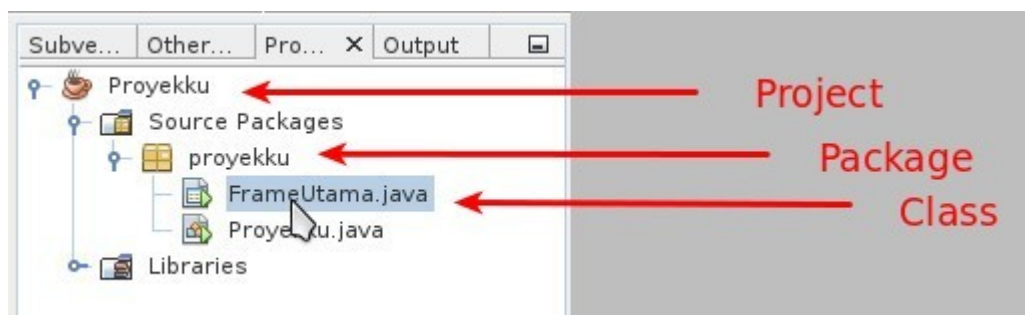
# Penjelasan Dari Perkenalan Dengan Netbeans

Anda harus mengenal beberapa kata penting ini:

1. Project
2. Class
3. Package
4. JFrame
5. Matisse

Apa itu? Mengapa penting? Ini adalah 5 hal yang akan saya pakai terus secara intens dalam buku ini. Dan memang hanya dengan 5 hal inilah Anda akan membuat aplikasi. Saya harus melakukan eksplorasi dahulu saat pertama membuat proyek Otodidak. Sekarang karena saya sudah tahu hanya ini yang dibutuhkan, maka Anda cukup fokus pada 5 hal ini dengan mengabaikan sebegitu banyaknya tombol dan menu di Netbeans. Ini lebih hemat waktu.

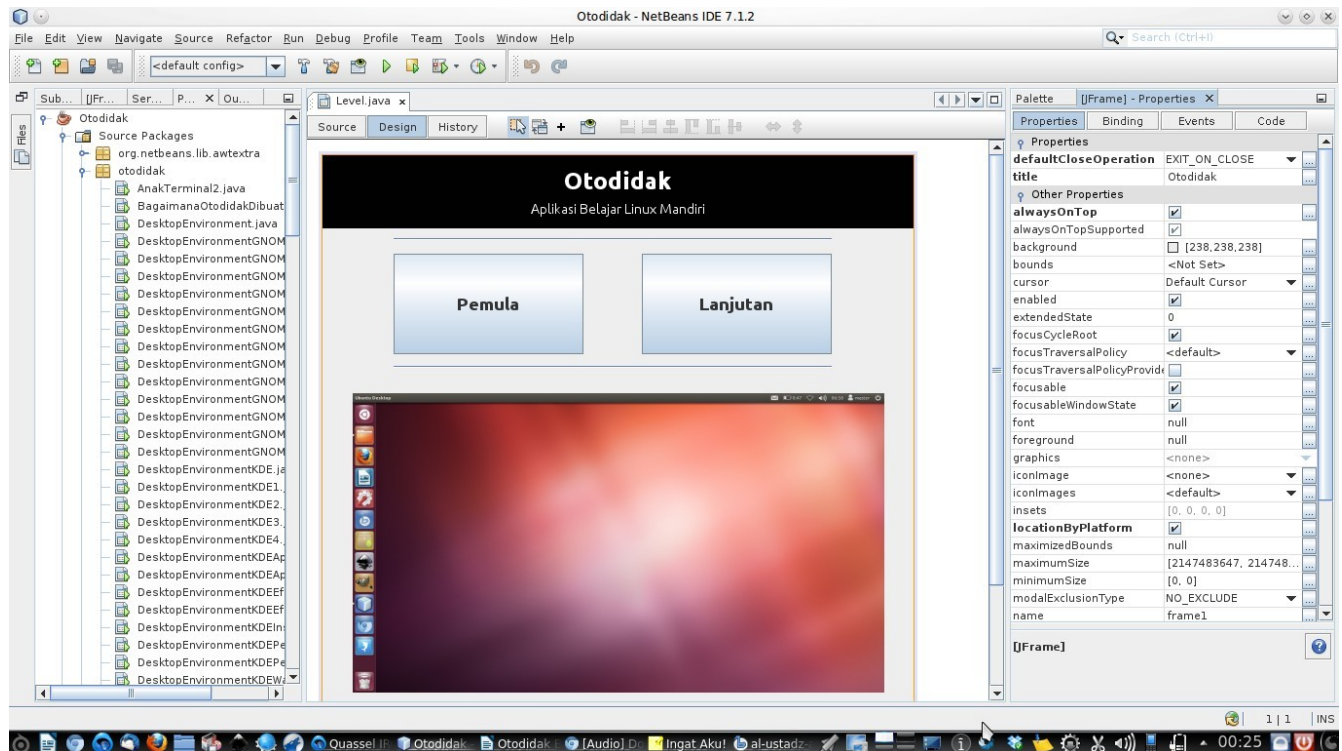
- **Project:** satu folder yang menampung seluruh kode program Anda. Ini ditampilkan sebagai folder tertinggi dalam panel kanan di Netbeans.
- **Class:** nama suatu kelompok fungsi dalam bahasa pemrograman Java. Nanti jika Anda sudah mencoba, akan paham sendiri. Semua berkas yang bernama \*.java adalah kode program Anda, dan nama berkas pasti sama persis dengan nama class. Misalnya Anda membuat class baru, maka nama class ini akan menjadi nama berkas .java juga. Ini sudah aturan dalam bahasa pemrograman Java. Bisa dibuat dengan klik kanan di **Package > New > Java Class**.
- **Package:** folder yang menampung berkas .java dan berkas pendukung proyek Anda. Kalau di Otodidak, saya memiliki 3 package. Satu untuk berkas .java, satu untuk gambar-gambar, dan satu untuk pustaka Java yang harus disertakan. Bisa dibuat dengan klik kanan di **Source Packages > New > Java Package**.



- **JFrame:** sebenarnya termasuk salah satu class dalam Java, yakni class Jendela. Ya JFrame inilah yang sebenarnya membuat Otodidak memiliki GUI. JFrame ini bisa diibaratkan kanvas, kita meletakkan tombol, gambar, teks, dan segalanya di atasnya. Inilah jendela aplikasi Anda.
- **Matisse:** GUI builder di dalam Netbeans. Developer Netbeans memberi nama untuk pembuat GUI di dalam Netbeans, suatu hal yang sangat unik. Jika Anda dan saya membuat aplikasi dengan drag and drop di Netbeans, itu artinya kita sedang menggunakan Matisse. Beberapa orang yang lama belajar Linux sendiri tidak tahu

Matisse saat bertanya mengenai Otodidak. Mungkin Anda sekarang lebih beruntung.

Sekarang, ini saya tunjukkan bagaimana *Project* asli Otodidak. Ini untuk membuat kesan nyata di benak kita.

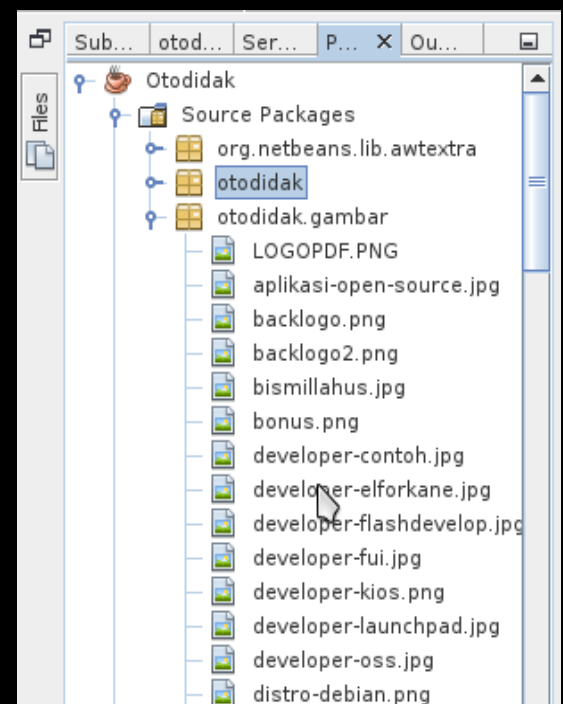


Sedikit penjelasan, yang Anda lihat di panel kiri itu kelas-kelas JFrame milik Otodidak. Setiap kelas JFrame itu mewakili 1 jendela dalam Otodidak. Anda akan membuat yang seperti ini. Anda perlu juga melihat isi dari Package **otodidak.gambar**, yang isinya hanya gambar-gambar dari Otodidak. Lihat gambar berikut.

Sebagai informasi, Otodidak berisi sekitar 306 kelas Java JFrame. Anda tidak usah sebanyak itu. Cukup tahu dasarnya saja. Namun buku satu ini hanya berisi pengenalan. Anda hanya akan membuat satu aplikasi paling dasar dengan buku satu ini.

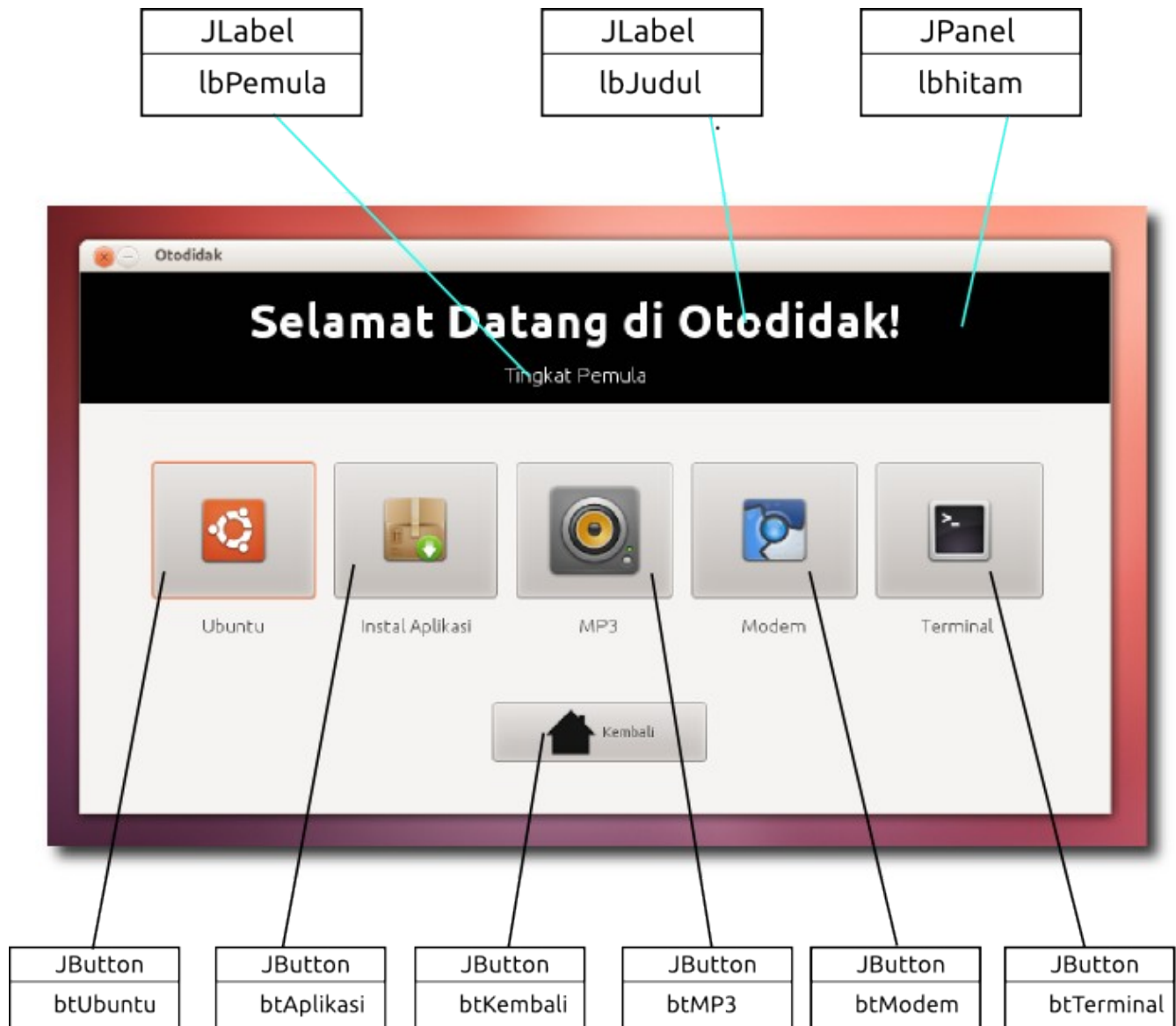
Untuk memantapkan, saya tunjukkan ada 3 buah Package dalam Otodidak:

1. **org.netbeans.lib.awtextra**
2. **otodidak**
3. **otodidak.gambar**



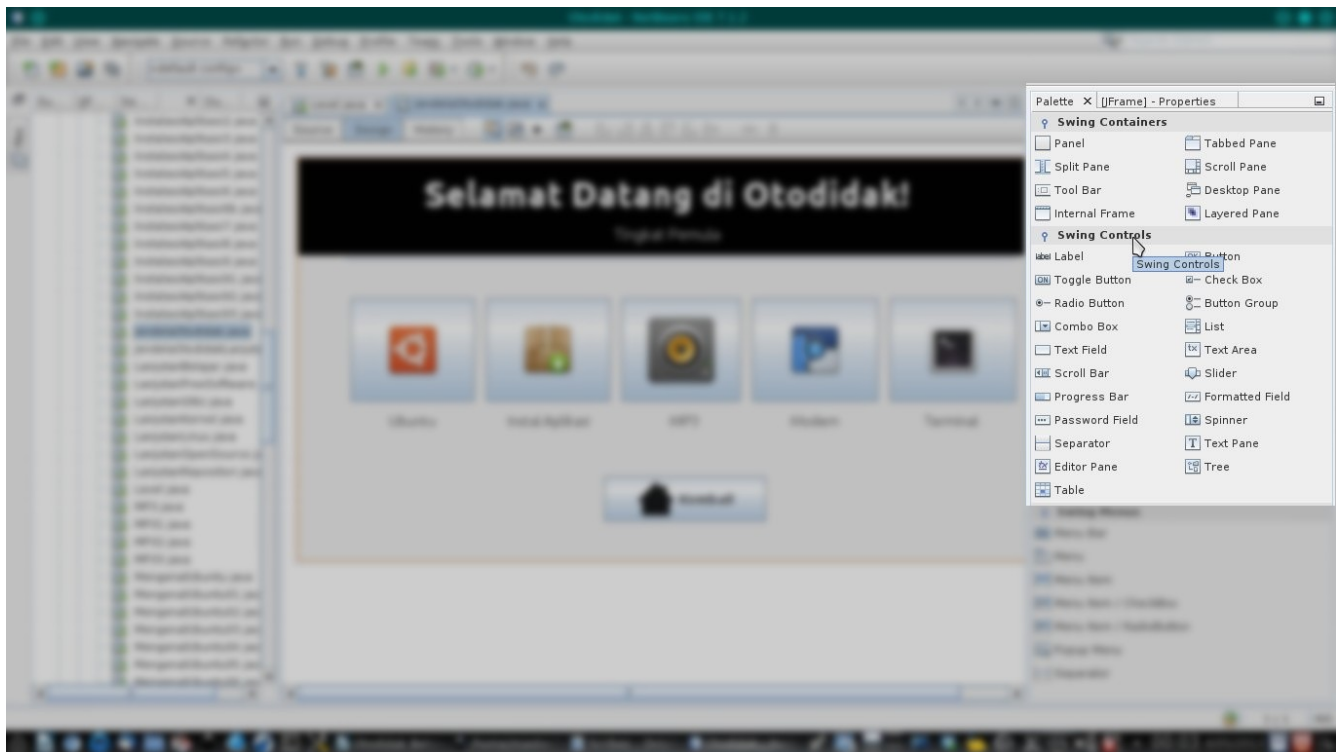
# Membongkar Tampilan Utama Otodidak

Untuk memudahkan belajar, saya akan memberi contoh yang termudah. Cukup dari aplikasi yang sudah ada kemudian dijelaskan per bagiannya. Anda harus tahu ini dahulu sebelum melangkah ke bagian pembuatan aplikasi. Perhatikan gambar berikut.

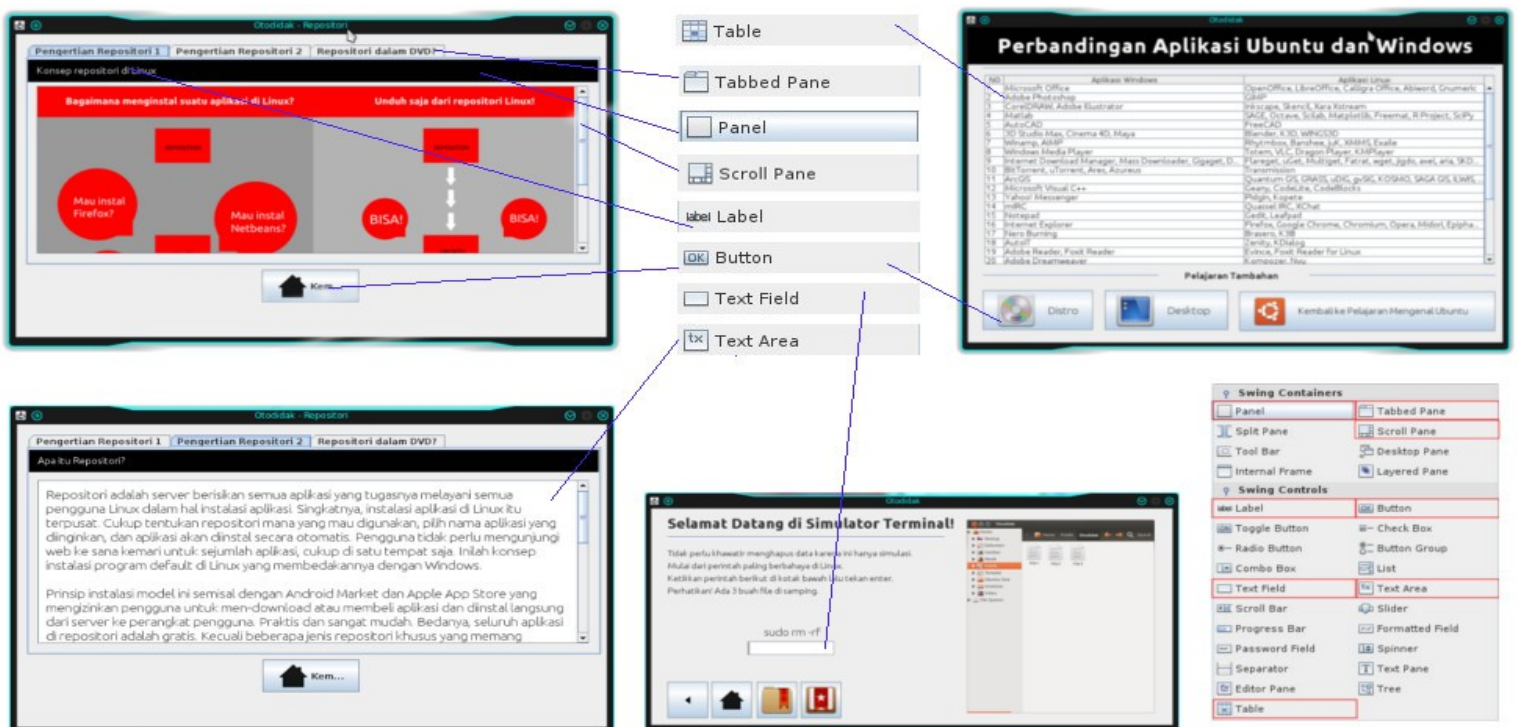


Ini adalah penjelasan komponen GUI apa saja yang dipakai untuk membangun tampilan utama Otodidak. Ternyata hanya JButton, JLabel, dan JPanel. Komponen apa ini? Ini adalah komponen GUI yang sudah tersedia di Matisse. Ini adalah bagian dari Java SWING. Dengan Matisse, kita cukup drag and drop komponen untuk membangun aplikasi kita. Bagaimana? Mudah, bukan? Sekarang, tentu saja Anda perlu mengetahui apa yang di-drag and drop.





Bagian skreensot yang saya fokuskan itu adalah komponen-komponen GUI yang akan kita pakai dalam pembuatan aplikasi. Otodidak sendiri dibuat hanya dari 7 komponen dari sekian banyak komponen yang disediakan dalam Matisse. Dan yang paling banyak dipakai dalam Otodidak adalah Button. Artinya, Anda akan fokus pada beberapa komponen saja untuk membuat aplikasi. Tidak perlu menghajar semuanya. Sekarang saya kenalkan ketujuh komponen GUI itu.



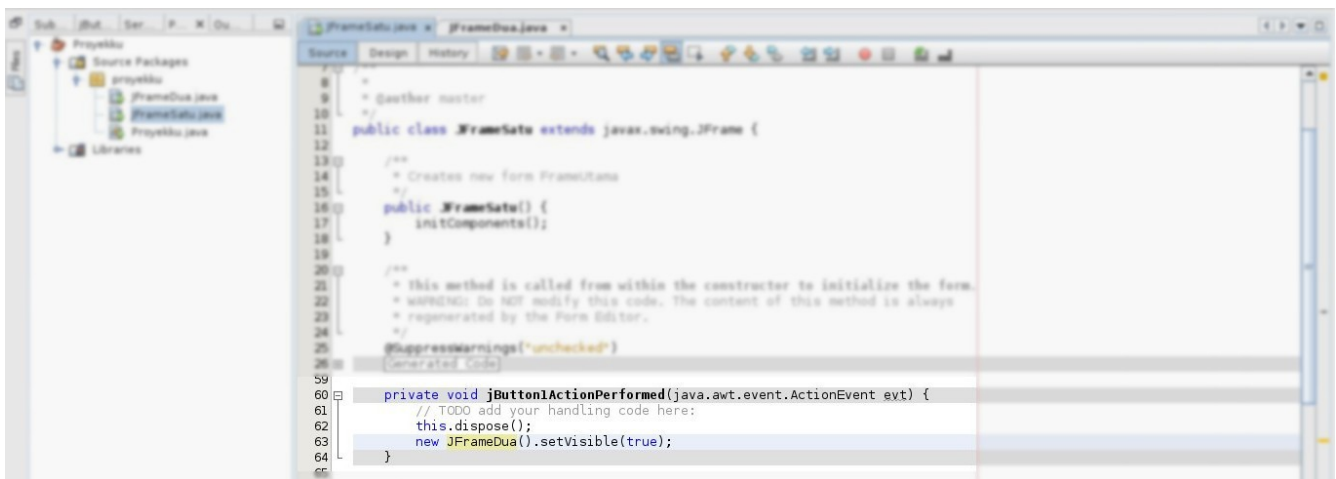


# Membuat Aplikasi Pertama Anda

Saatnya masuk ke kelas. Siap untuk membuat aplikasi pertama Anda? Saya berusaha menuliskan caranya hanya dalam 1 halaman ini. Buka Netbeans, buka *Project* Anda. Kalau saya, nama *Project* adalah Proyekku. Samakan saja dengan saya. Mari memulai.

1. Buat sebuah JFrame bernama **JFrameSatu**.
2. Letakkan satu JButton di situ. Biarkan namanya jButton1.
3. Buat sebuah JFrame lagi di dalam Package yang sama, beri nama **JFrameDua**.
4. Letakkan satu JButton juga di situ. Ubah namanya (tekan F2) menjadi jButton2.
5. Kembali ke JFrame pertama.
6. **Klik kanan JButton > Events > Action > ActionPerformed**. Ini akan membuka editor kode Netbeans secara otomatis, langsung pada kode fungsi event pada JButton.
7. Salin kode berikut pada isi { }:  

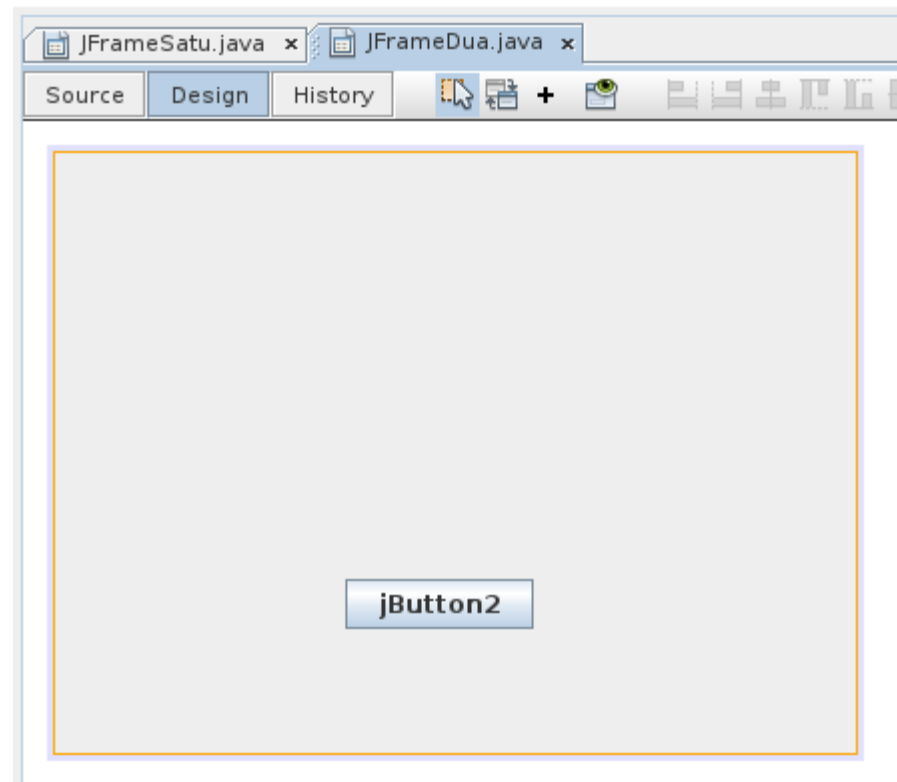
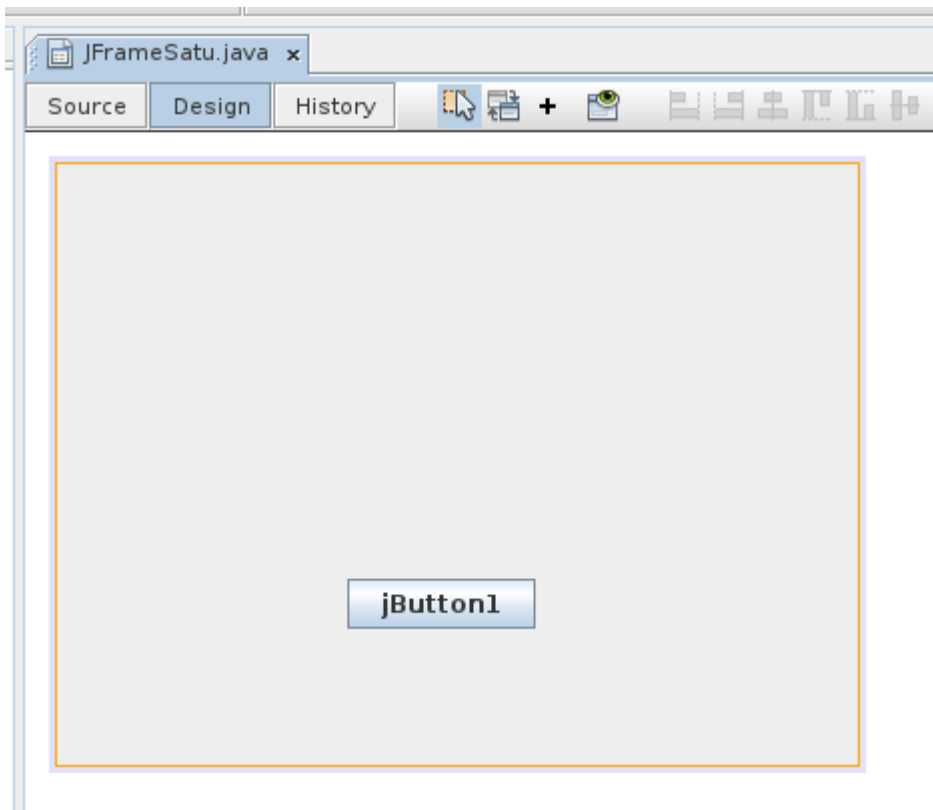
```
this.dispose();  
JframeDua.setVisible(true);
```
8. Sehingga sama dengan skrinsot ini:



9. Sudah? Tidak ada kode salah? Sekarang klik tombol **Design** di bawah Tab JFrameSatu.
10. Anda akan kembali melihat JFrameSatu Anda. Sekarang Run aplikasi (JFrame Anda) dengan **menu Run > Run File**. Ingat menu ini, karena akan Anda pakai terus.
11. Coba klik tombol jButton1. Apakah ia otomatis membuka JframeDua? JframeDua itu yang ada jButton2-nya, bukan? Jika ya, selamat. Anda telah berhasil membuat aplikasi pertama Anda.



Otodidak sebenarnya cuma kumpulan tombol yang berisi 2 baris kode ini, untuk memanggil JFrame-JFrame lain. Cuma itu. Berikut ini skrinsot dari JFrame yang kita desain:

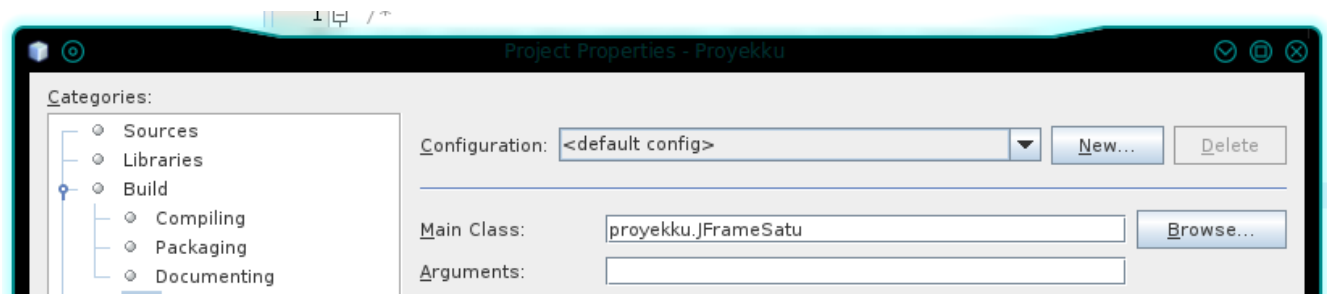


Apa maksud 2 baris kode tersebut? Baris pertama itu maksudnya untuk menutup jendela (dispose) dan baris kedua itu membuka jendela baru dari JFrameDua dengan kondisi setVisible(true) alias tampak. Jika false maka tidak tampak.

# Pra-Deploy

Mungkin ini sedikit mengganggu. Namun sekali Anda lakukan ini, nantinya *deploy* akan menjadi sangat lancar. Ikuti saja.

- 1) Lihat, dengan nama proyek **Proyekku**, maka di dalam Package **proyekku** pasti ada sebuah kelas bernama **Proyekku.java**. Benar, bukan?
- 2) Hapus kelas tersebut.
- 3) Buka menu Run > Set Project Configuration > Customize.
- 4) Ganti Main Class dari **proyekku.Proyekku** menjadi **proyekku.JFrameSatu**.
- 5) OK.




## Penjelasan Pra-Deploy

Sebenarnya, apa yang sedang kita lakukan ini? Jika Anda cerdas, akan terbaca bahwa kita sedang mengatur agar kelas utama – kelas yang pertama kali dijalankan nanti saat program dibuka – adalah JFrameSatu. Jika masih ada kelas Proyekku, maka nanti yang pertama kali dijalankan adalah kelas Proyekku sedangkan kelas ini tidak ada apa-apanya. Program Anda akan berjalan tidak semestinya. Yang kita inginkan justru JFrameSatu yang berjalan pertama kali. Maka dari itu kita hapus kelas Proyekku dan kita atur agar kelas utama adalah JFrameSatu. Jika Anda bingung, jangan pedulikan. Lanjut saja.

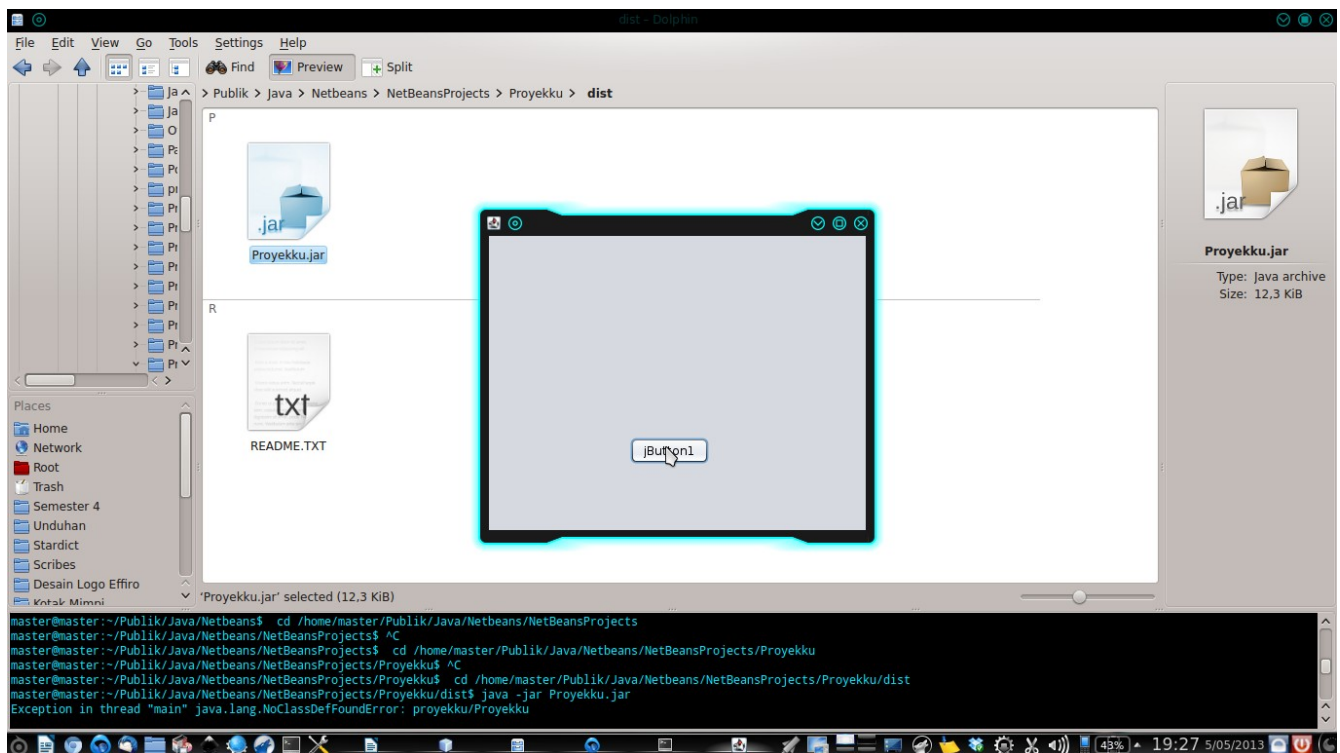
# Deploy!

Dalam kamus saya, *deploy* itu maknanya menyebarkan. Di dalam pemrograman Java, bersama Netbeans, maknanya adalah membuat berkas eksekutabel tunggal dari aplikasi kita. Jadi, kita akan membuat semacam EXE setelah kita memprogram. Nanti kita bisa kopi berkas semacam EXE ini ke komputer lain dan itu akan berjalan di sana. Format asli hasil *deploy* Java adalah .jar. Jika Anda dekat dengan lingkungan pemrograman atau *mobile smartphone*, pasti kenal format universal ini. Format ini sanggup berjalan di segala OS. Kita akan membuatnya dengan sekali klik.

- 1) Pastikan Anda sedang membuka proyek Anda.
- 2) Klik tombol  **Clean and Build Project**.
- 3) Sudah.

Lalu, di manakah letak berkas buatan kita itu? Kalau di komputer saya, ia ada di `/home/master/Publik/Java/Netbeans/NetbeansProject/Proyekku/dist`. Alamat ini bisa berbeda di komputer Anda tergantung di mana Anda mengatur pertama kali proyek Anda disimpan. Pokoknya, .jar senantiasa di-deploy ke dalam folder dist. Inilah eksekutabel kita. Aplikasi kita sendiri! Sekarang mari dijalankan.

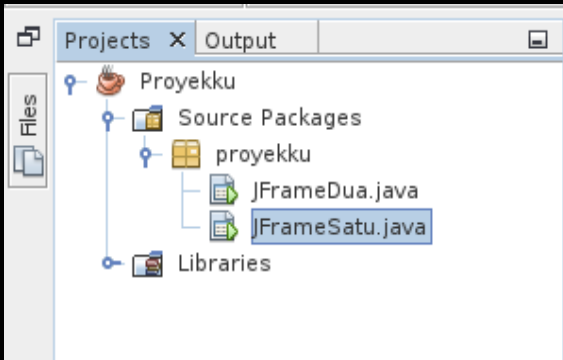
- 1) Klik kanan > Properties > TAB Permission > centang Executable.
- 2) OK.
- 3) Klik 2 kali pada .jar Anda.
- 4) Selamat, Anda telah menyelesaikan aplikasi pertama Anda!



# Mengenal Netbeans (Lagi)

Anda harus mengenal Netbeans. Mengapa? Untuk menghemat waktu. Jika Anda bisa bekerja secara cepat, aplikasi Anda cepat selesai. Bagaimana bisa? Dengan pemahaman terhadap “senjata” Anda ini. Perhatikan, pada dasarnya, Netbeans memiliki tampilan kanan, tengah dan kiri. Jika Anda memang belum familiar dengan yang namanya IDE, maka saya kenalkan dulu.

PANEL KIRI

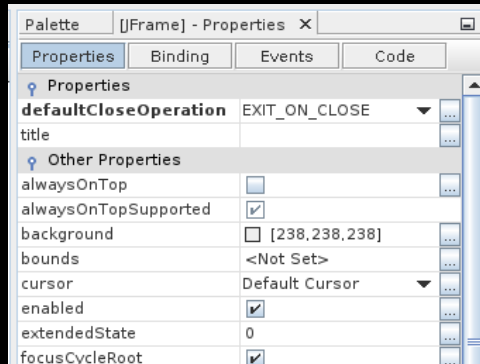


Panel kiri Netbeans adalah Project Explorer. Entahlah namanya apa intinya digunakan untuk melihat isi proyek. Apa yang akan Anda lakukan dengan ini?

- Menambah Package
- Menambah Class (Java atau JFrame Form)
- Mengurus Libraries (nanti pada akhir seri buku)

Selain itu, panel kanan ini berisi TAB Subversion, Services, Output, dan lain-lain. Apa Anda memerlukannya? Tidak untuk saat ini.

PANEL KANAN TAB PROPERTIES



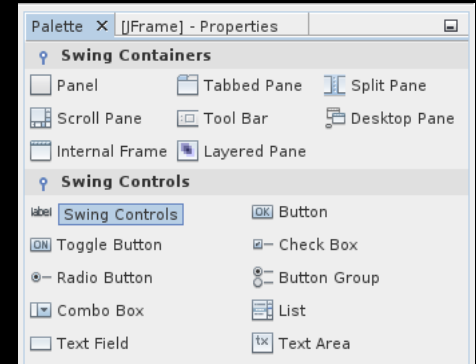
Panel kanan ini istimewa. Ada 2 TAB utama yakni **Palette** dan **Properties**. Pada TAB **Properties** ini, Anda akan melakukan hal-hal yang sangat penting:

- Memberi gambar pada JLabel
- Memberi nama JFrame
- Memberi judul JFrame
- Mengatur close operation pada JFrame
- Mengatur ukuran dan posisi JFrame
- Mengatur segala-galanya pada setiap komponen GUI Matisse

Mengapa ini ada? Karena setiap komponen memiliki ratusan properti. Apa itu properti? Itu adalah sifat yang bisa diubah pada setiap komponen. Pengubahan sifat inilah yang akan sering Anda lakukan dalam membuat aplikasi.

Semua komponen termasuk Button, Label, TextArea, Panel, dan lain-lain ada propertinya dan diatur di sini.

PANEL KANAN TAB PALLETE



Panel kanan pada TAB Pallette ini adalah kunci segala pembuatan GUI kita. Yang dimaksud drag and drop itu adalah penggunaan Pallette ini.

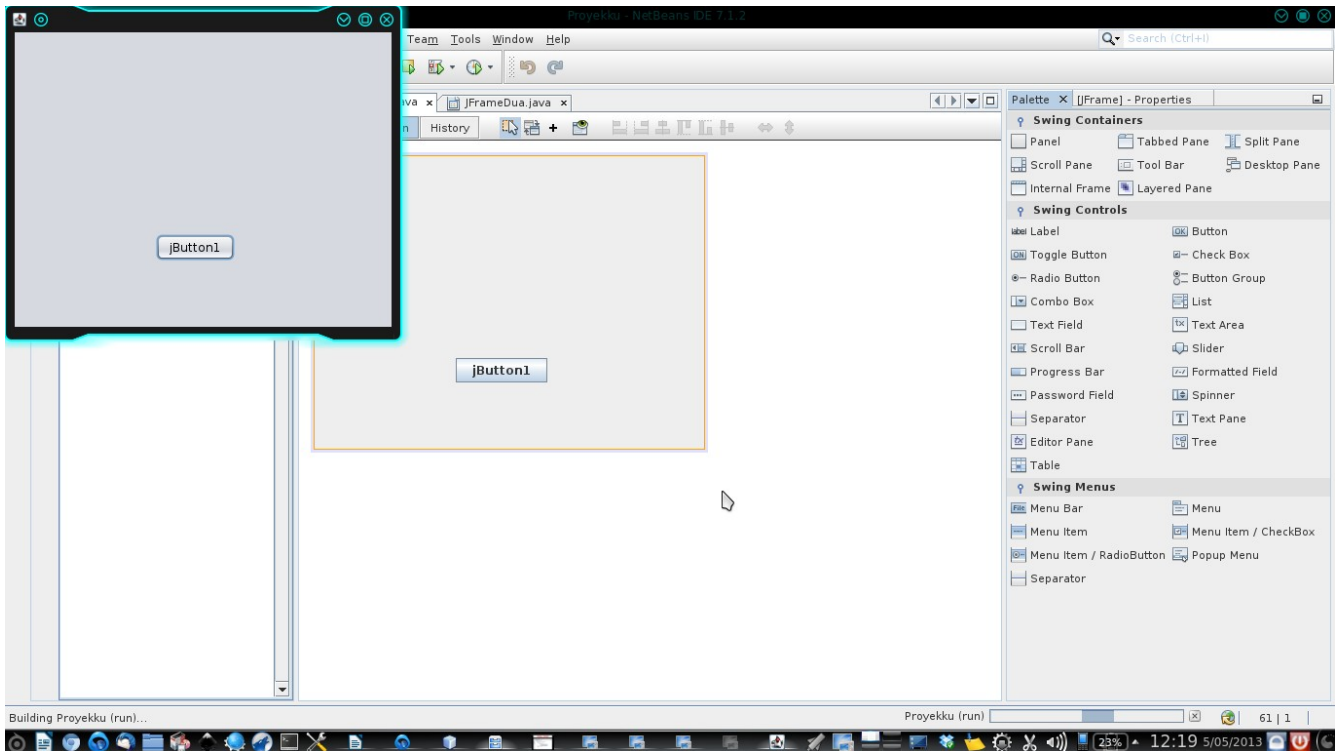
Matisse menyediakan sekian komponen GUI di dalam Pallette ini. Kita cukup drag salah satunya lalu drop di kanvas tengah. Jadilah sebuah GUI.

Matisse dengan Pallette ini sesungguhnya menyederhanakan pemrograman. Anda bisa membuat GUI tanpa Matisse. Namun Anda harus membuat kode programnya satu per satu yang itu untuk satu Button saja bisa 20 baris kode. Dengan Matisse semua itu menjadi simpel.

Jika sebelum ini Anda pernah menggunakan Microsoft Visual Basic, maka sesungguhnya Matisse sangat mirip dengannya.

# Masalah 1

Insya Allah, dalam setiap seri buku ini, saya akan kupas masalah yang terjadi dan saya beri trik Netbeans. Mengapa? Untuk menghemat waktu Anda juga. Karena ada sekian masalah yang saya hadapi saat membuat aplikasi yang itu harus saya selesaikan dengan mencari di internet selama sehari-hari. Saya akan persingkat dengan buku ini saja. Sekarang waktunya masalah pertama.



**Masalah:** JFrame selalu tampil di pojok kanan atas.

**Harapan:** JFrame selalu tampil di tengah.

**Solusi:** masukkan kode ini

```
setLocationRelativeTo(null);
```

pada bagian kode

```
public JFrameSatu() {  
    initComponents();  
}
```

di dalam JFrameSatu. Jika Anda ingin JFrameDua juga ada di tengah (dan senantiasa di tengah), lakukan hal yang sama. Kode yang benar seperti di samping.

```
16 public JFrameSatu() {  
17     initComponents();  
18     setLocationRelativeTo(null);  
19 }
```



# Trik Netbeans 1

Tekan **ESC** untuk membatalkan segala aksi yang sedang dilakukan.

## Rangkuman 1

Anda telah mendapat kode 2 baris:

```
this.dispose();  
new JFrameDua().setVisible(true);
```

Baris pertama untuk menutup JFrame. Baris kedua untuk memanggil JFrameDua dalam kondisi tampak. Inilah inti aplikasi Otodidak.

Anda telah mendapat kode tambahan:

```
setLocationRelativeTo(null);
```

Ini untuk memastikan letak jendela kita agar selalu di tengah layar.

Setiap usai deploy, kita perlu memberikan hak eksekutabel pada .jar. Jika tidak, maka .jar tidak akan bisa dijalankan.

Sebelum deploy, satu kali dalam pembuatan proyek, atur Main Class mengarah pada kelas yang kita inginkan.

**Menu Run > Set Project Configuration > Customize.**

## Latihan 1

Memang ketika saya di kelas, PR itu tidak menyenangkan. Namun kini setelah mencicip manisnya pemrograman, adanya tugas terasa kembali ke masa TK. Anda bisa melakukan ini untuk mengisi waktu belajar Anda:

- 1) Buatlah JFrame baru dengan TextField. Coba-coba saja.
- 2) Tambahkan TextArea.
- 3) Tambahkan Panel.
- 4) Gabungkan langkah 1, 2, 3 ke dalam 1 JFrame baru.
- 5) Buatlah 1 JFrame baru yang punya button, hubungkan button ke JFrame yang baru jadi.
- 6) Beri 1 tombol di JFrame satunya, hubungkan ke JFrame sebelumnya.
- 7) Tes program Anda. Dia harus bisa saling membuka antarframe.

Percayalah, tugas sebanyak ini akan kurang.

# Ingin Mengunduh Otodidak?

Anda bisa memperolehnya dari <http://malsasa.wordpress.com/2013/02/24/otodidak-versi-1-dirilis/>. Jika tidak mungkin mengunjunginya, Anda bisa memanfaatkan *mirror* berikut:

- <http://otodidak.linux.or.id>
- <http://sourceforge.net/projects/otodidak/?source=directory>

## Penutup Buku Satu

Alhamdulillah, Anda dan saya telah menyelesaikan buku satu ini. Dengan berbekal ini saja, Anda yang kreatif bisa membuat aplikasi sederhana sendiri. Sementara Anda belum mendapat cara pemaketan Debian, pemaketan ke EXE, bagaimana memasukkan gambar, membuat *link*, dan lain-lain. Sabar, belajar itu perlahan-lahan. Sampai jumpa di buku dua. Ingat, buku ini tidak lepas dari cacat. Silakan komentar di blog saya untuk mengungkap kritik dan saran. Semoga ini bermanfaat.