

# BAB V

## BEKERJA DENGAN *FILECHOOSER*

---

### A. CAPAIAN PEMBELAJARAN

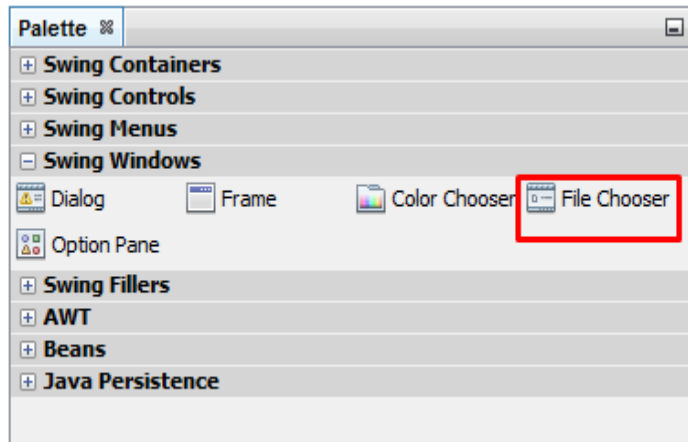
1. Mahasiswa dapat mengerti dan dapat menggunakan Open File Dialog
2. Mahasiswa dapat mengerti dan dapat menggunakan Save File Dialog
3. Mahasiswa dapat menggunakan jenis Open File dialog FileChooser
4. Mahasiswa dapat membuat program menggunakan Open file dialog dan Save File Dialog FileChooser

### B. MATERI

#### 1. FileChooser

*FileChooser* menyediakan GUI untuk menavigasi sistem file, dan kemudian memilih file dari daftar direktori, Untuk menampilkan *FileChooser*, Kita bisa menggunakan *JFileChooser* API untuk menampilkan dialog modal yang berisi *FileChooser*. Cara lain untuk menampilkan *FileChooser* adalah dengan

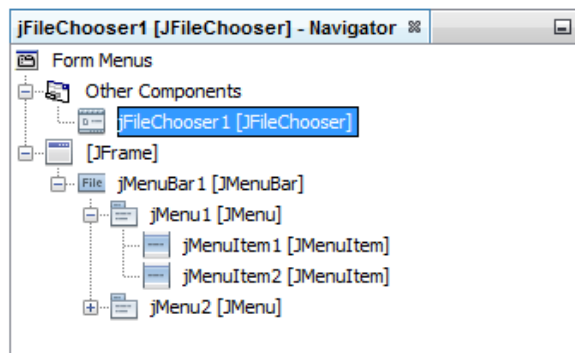
menambahkan sebuah instance JFileChooser dari Palette Swing windows kedalam form.



**Gambar 5. 1 kontrol File Chooser**

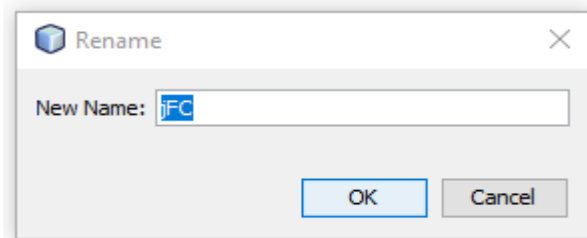
a. Bekerja dengan *File Chooser*

- 1) Drag *File Chooser* ke dalam form atau frame Anda, selanjutnya komponen FileChooser tidak terlihat muncul didalam frame . Itu tetapi kita dapat melihatnya di jendela Navigator:



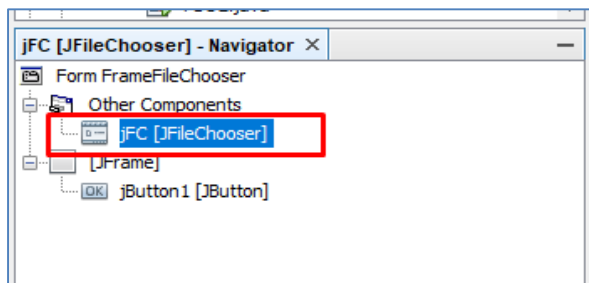
**Gambar 5. 2 Navigator *File Chooser***

- 2) Nama default untuk File Chooser adalah `jFileChooser1`. Untuk merubahnya Klik kanan pada `jFileChooser1` di jendela Navigator. Dari menu yang muncul, pilih `Change Variable Name`. Ketika kotak dialog muncul, contoh ketik `jFC` sebagai nama Variabel:



**Gambar 5. 3 Rubah Nama Variable FileChooser**

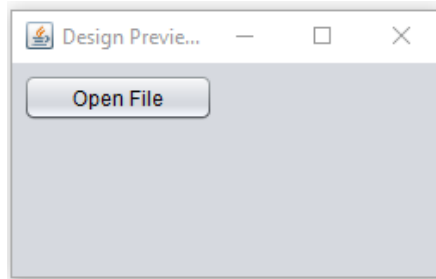
- 3) Klik `OK` untuk mengkonfirmasi perubahan. Jendela Navigator akan terlihat seperti ini:



**Gambar 5. 4 Navigator Hasil Perubahan FileChooser**

- 4) Sekarang kita telah memiliki File Chooser yang ditambahkan ke dalam project.
- 5) Untuk menampilkan kotak dialog File Chooser cukup sederhana.

- 6) Desain dan sisipkan komponen Button/Tombol kedalam Form seperti dibawah ini :

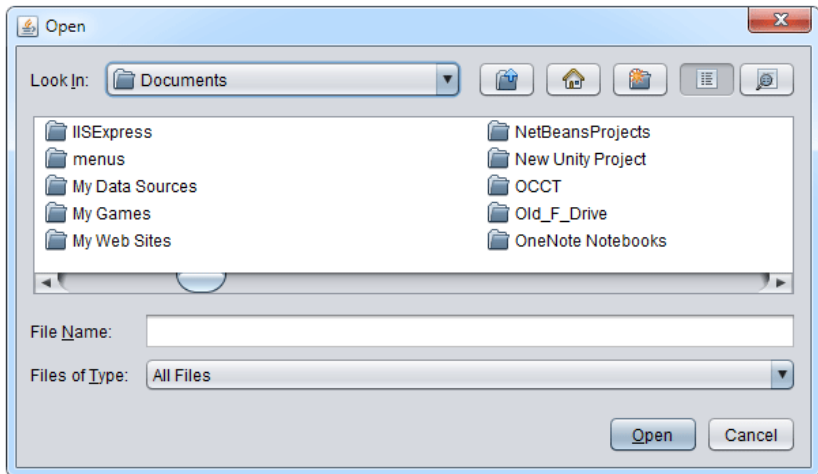


**Gambar 5. 5 Desain Tombol Open File**

- 7) Berikan code Tombol Open, pada events>action >actionPerformed

```
private void jButton1ActionPerformed  
(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    int returnVal = JFC.showOpenDialog( this  
);  
}
```

- 8) Jika berjalan baik maka hasil tampilan File Open Dialog menggunakan File Chooser seperti gambar dibawah ini



Metode `ShowOpenDialog` mengembalikan nilai bilangan bulat. Nilai ini memberitahu kita tombol mana yang diklik pada kotak dialog: `Open`, `Cancel`, dll. Penyimoanan nilai dalam variabel yang disebut `returnVal`.

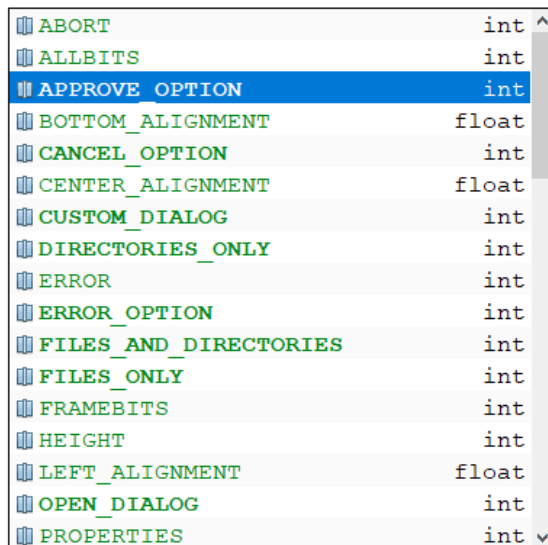
Hal yang dilakukan `Dialog Open` adalah memilih file - sebenarnya kotak itu tidak membuka apa pun. Jika kita ingin membuka file, kita harus menulis kodenya sendiri.

Pertama, tambahkan Pernyataan `IF` berikut pada kode , tepat di bawah dua baris lainnya ( `returnVal`):

```
if (returnVal ==  
    javax.swing.JFileChooser.APPROVE_OPTION) {  
}
```

Kita menggunakan Class `Swing` yang disebut `JFileChooser`. Dengan Class ini, kita dapat

memeriksa tombol mana yang diklik. Saat kita mengetik titik setelah JFileChooser, kita akan melihat daftar popup:



**Gambar 5. 7 Class Filechooser**

APPROVE\_OPTION berarti memilih tombol OK atau Yes. Kita menguji variabel `returnVal` untuk melihat apakah itu cocok dengan APPROVE\_OPTION (apakah pengguna mengklik OK?)

Untuk mendapatkan file yang dipilih oleh pengguna, bisa menggunakan metode yang disebut ***getSelectedFile***. Namun, ini mengembalikan objek File, bukan string. Objek File adalah bagian dari kelas IO di Java. Jadi tambahkan baris berikut ke Pernyataan IF kita :

```
java.io.File file = db.getSelectedFile( );
```

Jadi file yang dipilih oleh pengguna akan berakhir di objek File yang kita sebut file.

Untuk membuka file misalnya kita perlu mengkonversinya menjadi string:

```
String file_name = file.toString( );
```

Pada baris ini menggunakan metode toString dari objek File. Untuk menempatkan hasilnya dalam variabel baru bernama file\_name. Tambahkan baris ke Pernyataan IF Anda.

Untuk menampilkan nama file pada kotak dialog Pesan, tambahkan kode berikut ini

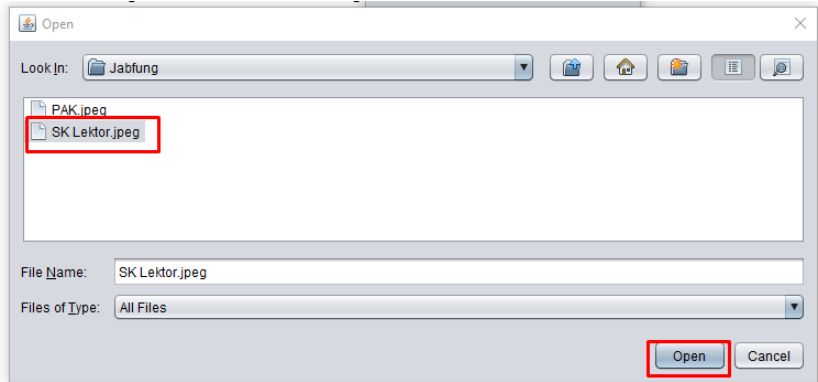
```
JOptionPane.showMessageDialog(this,  
file_name);
```

Secara lengkap code yang ditambahkan ketika file dibuka dan dipilih ditampilkan ke message dialog adalah sebagai berikut :

```
int returnVal = JFC.showOpenDialog( this );  
if(returnVal==javax.swing.JFileChooser.APPROVE_OPTION)  
{  
    java.io.File file = JFC.getSelectedFile( );  
    String file_name = file.toString();  
    JOptionPane.showMessageDialog(this, file_name);  
}
```

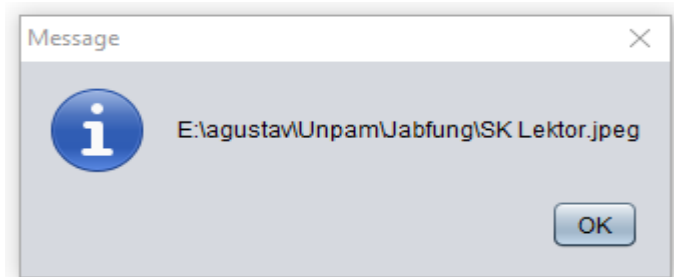
}

Jalankan program, jika berjalan baik maka hasilnya seperti gambar dibawah ini :



**Gambar 5. 8 Hasil Run File Chooser**

Pilih File yang akan ditampilkan lalu klik Open, maka hasil yang ditampilkan seperti gambar berikut ini :



**Gambar 5. 9 Nama File Chooser**



b. Menentukan File Of type (Jenis File ) pada Open File dialog File Chooser

Sebelum kami menambahkan kode untuk membuka file yang dipilih, kita mungkin telah memperhatikan bahwa Files of Type pada kotak dialog Open file Anda diatur ke "All File" (Semua file) . Kita dapat memfilter file pada daftar pilihan , sehingga pengguna hanya dapat membuka, katakanlah, file teks(Txt), atau hanya gambar dengan ekstensi tertentu (jpeg, gif, png).

Untuk memfilter daftar "Files of Type", kotak dialog memiliki metode `addChoosableFileFilter`. Tetapi Anda membutuhkan objek Filter di antara tanda kurung bulat.

Agar kode kita tidak terlalu panjang, tambahkan dua pernyataan impor berikut ke bagian atas kode, contoh seperti di bawah yang lain:

```
import javax.swing.filechooser.FileFilter;  
  
import  
javax.swing.filechooser.FileNameExtensionFilter;
```

Untuk menyiapkan filter ekstensi nama file, Anda perlu membuat objek `FileFilter` baru. Tambahkan baris berikut tepat sebelum baris pertama kode Anda (sebelum baris `int returnVal`):

```
FileFilter ft = new FileNameExtensionFilter("Text  
Files", "txt");
```

Di antara tanda kurung bulat `FileNameExtensionFilter` kita terlebih dahulu membutuhkan teks yang akan muncul pada daftar `Files of Type`. Setelah koma kita ketik nama file yang ingin ditampilkan. Ekstensi file yang tepat diperlukan, di sini, tetapi tanpa titik. Perhatikan tanda kutip ganda di atas.

Anda dapat menambahkan lebih dari satu ekstensi. Cukup ketik koma dan kemudian file yang ingin Anda tampilkan contoh :

```
FileFilter ft1 = new FileNameExtensionFilter("Docx  
Files", "docx");
```

```
FileFilter ft2 = new FileNameExtensionFilter("PDF  
Files", "pdf");
```

Setelah kita menyiapkan objek filter, kita dapat menggunakan metode `addChoosableFileFilter` dari kotak dialog :

```
jFC.addChoosableFileFilter( ft);  
  
jFC.addChoosableFileFilter( ft1);  
  
jFC.addChoosableFileFilter( ft2);
```

Baris kode lengkap untuk menentukan jenis file yang ditampilkan pada `Open File Dialog` Seperti dibawah ini :

```
FileFilter ft = new FileNameExtensionFilter("Text Files",  
"txt");
```

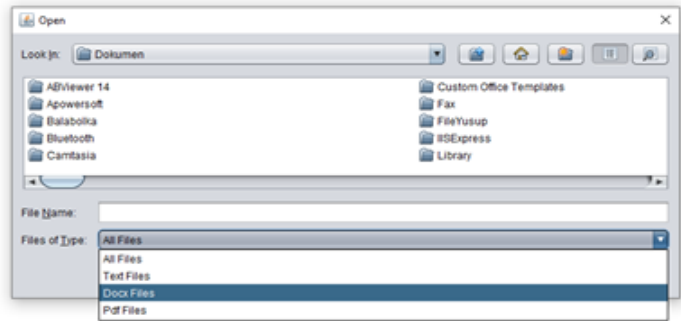
```
FileFilter ft1= new FileNameExtensionFilter("Docx Files", "docx");
```

```
FileFilter ft2 = new FileNameExtensionFilter("Pdf Files ", "pdf");
```

```
jFC.addChoosableFileFilter(ft);  
jFC.addChoosableFileFilter(ft1);  
jFC.addChoosableFileFilter(ft2);
```

```
int returnVal = jFC.showOpenDialog( this );  
  
if (returnVal ==  
    javax.swing.JFileChooser.APPROVE_OPTION) {  
    java.io.File file = jFC.getSelectedFile( );  
  
    String file_name = file.toString();  
  
    JOptionPane.showMessageDialog(this, file_name);  
}
```

Jalankan program dengan shift+F6, atau Run file , jika berjalan baik terlihat pada kotak *file of type* adalah jenis ekstensi file yang kita tulisi pada code , tampilan hasil seperti gambar dibawah ini :



**Gambar 5. 10 File Of Type**

## 2. Save File Dialog

Jika Kita ingin menampilkan kotak dialog Save File setelah kotak dialog Open File dilakukan, maka kita gunakan kembali File Chooser kali ini kita gunakan `SaveFileDialog`. Contoh penulisan kode nya :

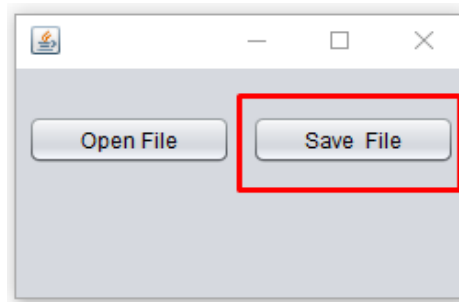
```
int returnVal = jFC.showSaveDialog( this );
```

Sekali lagi, ini cukup untuk menampilkan kotak dialog. Anda tidak perlu melakukan hal lain, kita juga dapat menambahkan filter file.

```
FileFilter ft = new FileNameExtensionFilter( "Text Files",
"txt"
);
jFC.addChoosableFileFilter( ft );
int returnVal = jFC.showSaveDialog(this);
```

Contoh Program Lengkap :

- 1) Modifikasi Form Open Dialog yang pernah di buat. Tambahkan Button untuk tombol Save seperti gambar dibawah ini



**Gambar 5. 11 Tombol Save File**

- 2) Berikan kode pada Tombol Save, klik kanan event>action>actionPerformed

- 3) Tulis program nya seperti code dibawah ini :

```
FileFilter ft = new FileNameExtensionFilter("Text Files",  
"txt");
```

```
FileFilter ft1 = new FileNameExtensionFilter("Docx Files  
", "docx");
```

```
FileFilter ft2 = new FileNameExtensionFilter("Pdf Files ",  
"pdf");
```

```
jFC.addChoosableFileFilter(ft);
```

```
jFC.addChoosableFileFilter(ft1);
```

```
jFC.addChoosableFileFilter(ft2);
```

```

        int returnVal = jFC.showSaveDialog(this );

        if (returnVal ==
javax.swing.JFileChooser.APPROVE_OPTION) {

            File fileToSave = jFC.getSelectedFile();

            String file_name = fileToSave.getAbsolutePath();

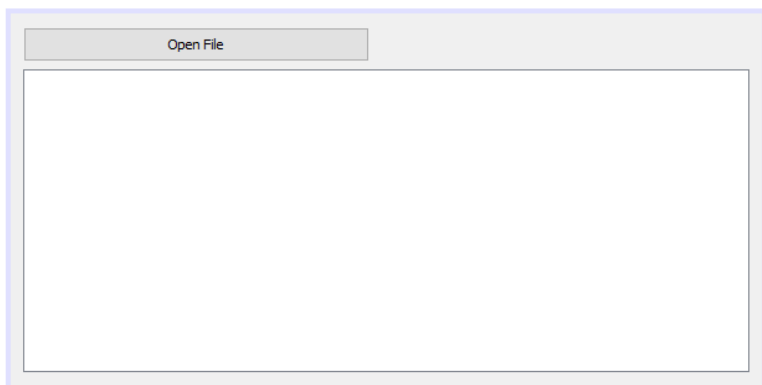
            JOptionPane.showMessageDialog(this, file_name);

        }

```

### C. LATIHAN

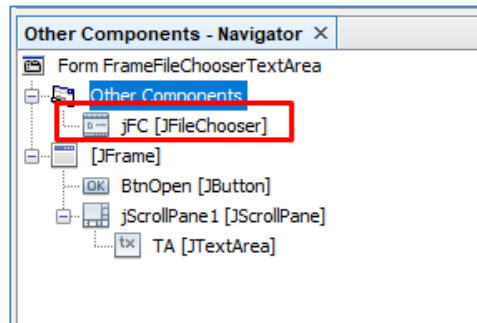
1. Siapkan Frame baru , lalu desain Form nya seperti gambar dibawah ini :



**Gambar 5. 12 Open File**

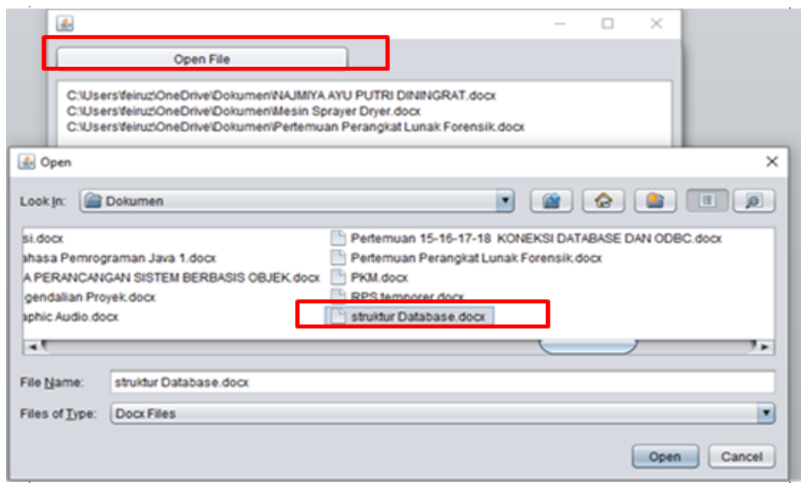
Komponen yang dibutuhkan :

- a. JButton untuk open File dialog
- b. JText Area untuk menampilkan File ketika dipilih
- c. Jchooser File untuk mengaktifkan file Open Dialog



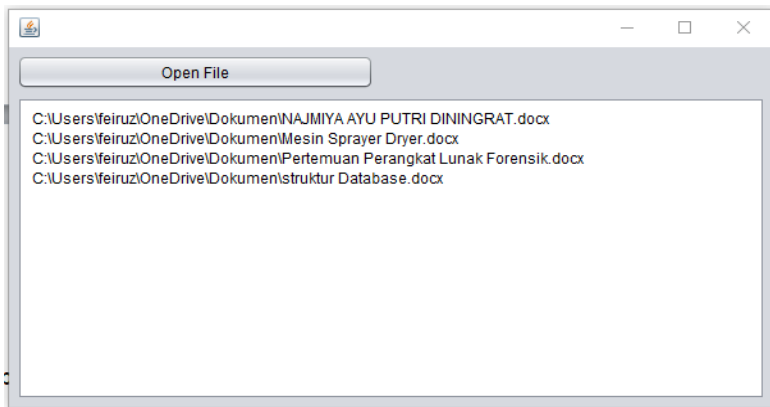
**Gambar 5. 13 Navigator Open File**

2. Keterangan program ketika di run dan kode berjalan baik :
  - a. Klik Open File maka dialog open akan menampilkan file sesuai dengan type data yang ditentukan
  - b. Pilih Filenya lalu klik Open Kembali



### Gambar 5. 14 Open File Terpilih

- c. Selanjutnya file terpilih akan muncul pada Text Area, begitu seterusnya hingga selesai contoh hasil



Gambar 5. 15 Open Dialog Hasil File Terpilih

3. Kode program untuk menampilkan File ke dalam Text Area ketika file open dialog dipilih .

```
private void
BtnOpenActionPerformed(java.awt.event.ActionEvent
evt) {
    // TODO add your handling code here:

    FileFilter ft = new FileNameExtensionFilter("Text
Files", "txt");
    FileFilter ft1 = new FileNameExtensionFilter("Docx
Files", "docx");
    FileFilter ft2 = new FileNameExtensionFilter("Pdf
Files", "pdf");
```



```

jFC.addChoosableFileFilter(ft);
jFC.addChoosableFileFilter(ft1);
jFC.addChoosableFileFilter(ft2);

int returnVal = jFC.showOpenDialog(this );
if (returnVal ==
javax.swing.JFileChooser.APPROVE_OPTION) {
    java.io.File file = jFC.getSelectedFile( );
    S1= S1+" "+jFC.getSelectedFile()+"\n" ;

    TA.setText(S1) ;
}
}

```

#### D. REFERENSI

Philip Conrod & Lou Tylee. Learn Java™ GUI Applications A JFC Swing NetBeans Tutorial 8th Edition , Kidware Software, LLC, 2015

Danny Poo Derek Kiong Swarnalatha Ashok, Object-Oriented Second edition Programming and Java , Springer 2008

[https://www.homeandlearn.co.uk/java/java\\_check\\_boxes.html](https://www.homeandlearn.co.uk/java/java_check_boxes.html), diakses pada   diakses pada tanggal 23 Juni 2022.

Romi satrio Wahono, Java Gui, 2016