Mobile (Programming) Technology

Modul Praktikum

# Topik

Pengenalan SL4A, Class Android, Pengenalan awal Script dengan SL4A.

# Tujuan

Setelah menyelesaikan praktikum ini, mahasiswa diharapkan mampu:

1. Menggunakan SL4A untuk membuat aplikasi android.
2. Membuat aplikasi hello world dengan method ttsSpeak() android.
3. Membuat aplikasi masukan (*input*) dengan menggunakan method dialogGetInput().
4. Membuat aplikasi keluaran (*output*) dengan menggunakan method makeToast().
5. Membuat aplikasi masukan/keluaran (*input/*output) dengan memanfaatkan dialog Alert.

# Alat dan Bahan

* Gadget android dengan aplikasi **QPython3**, Informasi unduh dan instalasi QPython3 dapat diperoleh di android Playstore.
* Dokumentasi **SL4A** **API** bisa lihat di: <https://github.com/damonkohler/sl4a/blob/wiki/ApiReference.md>
* Untuk pemrograman melalui emulator android melalui komputer, bisa diunduh di: <http://developer.android.com/sdk/index.html>

# Rangkuman/Teori

## [Tentang SL4A]

**SL4A** (*Scripting Layer for Android*), merupakan aplikasi pembuat *script open source* untuk membuat aplikasi android. **SL4A** selain tersedia untuk *script* pemrograman ***python***, juga tersedia untuk beberapa bahasa pemrograman lainnya (seperti: Perl, Ruby, Lua, BeanShell, JavaScript and Tcl).

Dengan SL4A program dapat beriteraksi pada perangkat android dengan menggunakan *high level API*, sehingga memungkinkan bagi kita untuk membuat program yang dapat mengakses data sensor, SMS, melakukan *render* *user interface* dan sebagainya.

## [Class Android]

Untuk membuat aplikasi android dengan **SL4A** (*Scripting Layer for Android*), kita harus melakukan import class Android terlebih dahulu, baru kemudian kita membuat object (misal nama *object*: droid) dari class Android tersebut.

Instruksi untuk Import Android:

**from android import Android**

Instruksi untuk membuat *object* : (misal nama *object*: **droid**)

**droid = Android()**

Setelah kita membuat object droid (misal nama *object*: droid) tersebut, selanjutnya kita dapat menggunakan API atau UIfaçade (façade sama dengan visual user interface di android) dengan memanggil method object tersebut.

## [Method ttsSpeak()]

Method **ttsSpeak**() ini merupakan method yang dapat digunakan agar gadget android mengucapkan (dengan suara) string yang kita minta.

Instruksi untuk menggunakan method ttsSpeak() :

*<nama\_object>***.ttsSpeak(***<string\_pesan>***)**

Contoh: (misal nama *object*: **droid**)

from android import Android

droid = Android()

droid.ttsSpeak(“Hello World”)

## [Method makeToast()]

Method **makeToast**() ini merupakan method yang dapat digunakan agar gadget android menampilkan splash label string yang kita minta.

Instruksi untuk menggunakan method ttsSpeak() :

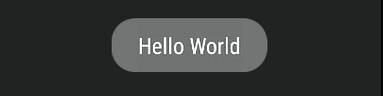
*<nama\_object>***.makeToast(***<string\_pesan>***)**

Contoh: (misal nama *object*: **droid**)

from android import Android

droid = Android()

droid. makeToast (“Hello World”)



Gambar 1.1. Tampilan dari method makeToast.

## [Method dialogGetInput()]

Method **dialogGetInput**() ini merupakan method yang dapat digunakan agar gadget android menampilkan dialog masukan yang. Dialog masukan terdiri dari judul (title dialog), teks pesan masukan, dan input berupa text box.

Penggunaan method dialogGetInput() ini biasanya disertai dengan penggunaan method result untuk membaca hasil pembacaan dari masukan (*input*) pengguna.

Instruksi untuk menggunakan method **dialogGetInput**() dengan **result**:

<*nama\_variabel\_penampung*> = <*nama\_object*>**.dialogGetInput(**

<*string\_title\_dialog*>,

<*string\_pesan\_input>*

*<string\_teks\_default*>**).result**

Contoh: (misal nama *object*: **droid**)

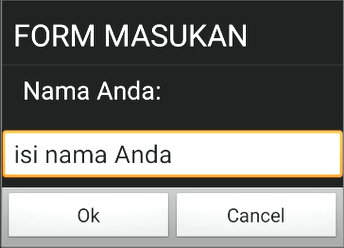
from android import Android

droid = Android()

nama = droid.dialogGetInput(“FORMULIR MASUKAN”, “Nama Anda: “, “isi nama anda di sini”).result

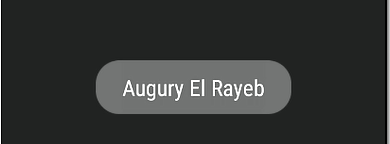
droid. makeToast (nama)

hasil dari contoh di atas adalah; akan tampil dialog masukan (*input*) yang menyediakan *textbox* untuk diisi oleh pengguna. Selanjutnya method makeToast() akan menampilkan teks yang diisi oleh pengguna.



Gambar 1.2. Dialog alert

jika pengguna mengisi ***Augury El Rayeb*** pada input box dialog, makan tampilan hasilnya adalah seperti gambar berikut:



Gambar 1.3. Hasil dari dialog

## [Membuat Dialog Alert]

Method **dialogCreateAlert**() ini merupakan method yang dapat digunakan agar gadget android menampilkan dialog *alert* (dialog waspada). Dialog masukan terdiri dari judul (title dialog), teks pesan *alert* (pesan waspada), dan beberapa tombol yang bisa ditampilkan sesuai keinginan kita.

Penggunaan method dialogGetInput() ini biasanya disertai dengan penggunaan method **dialogSetPositiveButtonText**() dan **dialogSetNegativeButtonText**() sebagai tombol yang akan tampil pada dialog, serta **dialogShow**() untuk menampilkan dialog alert yang sudah kita buat.

## [Method dialogCreateAlert()]

Instruksi untuk menggunakan method **dialogCreateAlert** ():

<*nama\_object*>**.dialogCreateAlert(**<*string\_title\_dialog*>,

<*string\_pesan\_alert>***)**

Untuk menambahkan tombol pada dialog alert yang kita buat, maka kita bisa menggunakan method **dialogSetPositiveButtonText**() dan **dialogSetNegativeButtonText**()

## [Method dialogSetPositiveButtonText() dan dialogSetNegativeButtonText()]

Method **dialogSetPositiveButtonText**() dan **dialogSetNegativeButtonText**()merupakanmethod yang digunakan untuk menambahkan tombol pada dialog.

Instruksi untuk menggunakan method **dialogSetPositiveButtonText**():

<*nama\_object*>**.dialogSetPositiveButtonText(**<*string\_tombol*>**)**

Instruksi untuk menggunakan method **dialogSetNegativeButtonText**():

<*nama\_object*>**.** **dialogSetNegativeButtonText(**<*string\_tombol*>**)**

Selanjutnya setelah kita menambahkan tombol ke dialog alert yang kita buat, maka untuk menampilkan dialog alert yang sudah lengkap dengan tombolnya digunakan method **dialogShow**().

## [Method dialogShow()]

Method **dialogShow**() merupakanmethod yang digunakan untuk menampilkan dialog yang sudah kita buat.

Instruksi untuk menggunakan method **dialogShow**():

<*nama\_object*>**.dialogShow()**

Contoh: (misal nama *object*: **droid**)

from android import Android

droid = Android()

nama = droid.dialogCreateAlert(“PERHATIAN”, “Harap pilih tombol yang anda suka“)

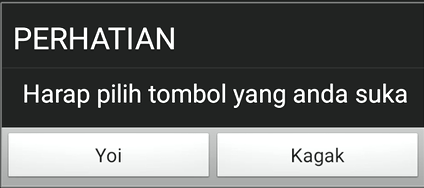
droid.dialogSetPositiveButtonText(“Yoi”)

droid.dialogSetNegativeButtonText(“Kagak”)

droid.dialogShow()

Hasil dari contoh di atas adalah; akan tampil dialog alert yang memiliki title (judul) **PERHATIAN**, serta pesan alertnya adalah **teks Harap pilih tombol yang anda suka**, dan tombol dengan tulisan **Yoi** dan tombol dengan tulisan **kagak**.

Gambar Contoh dialog alert dengan positive button dan negative button:



Gambar 1.1. Dialog alert

Dalam membuat dialog alert seperti di atas (menggunakan method **dialogSetPositiveButtonText**() dan **dialogSetNegativeButtonText**() sebagai tombol), kita juga bisa menggunakan method **dialogSetItems**() dalam membuat dialog alert yang dapat berfungsi sebagai dialog pilihan.

## [Method dialogSetItems()]

Method **dialogSetItems**() merupakanmethod yang digunakan untuk menampilkan pilihan pada dialog yang kita buat.

Instruksi untuk menggunakan method **dialogSetItems**():

<*nama\_object*>**.** **dialogSetItems(**[<*JSON\_Array\_items*>]**)**

Contoh: (misal nama *object*: **droid**)

from android import Android

droid = Android()

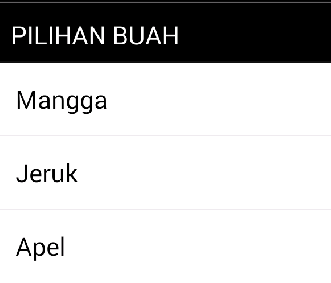
nama = droid.dialogCreateAlert(“PILIHAN BUAH”)

droid.dialogSetItems( [ “Mangga”, “Jeruk”, “Apel” ] )

droid.dialogShow()

Hasil dari contoh di atas adalah; akan tampil dialog alert yang memiliki title (judul) **PILIHAN BUAH**, serta teks tombol pilihan adalah tomboldengan tulisan **Mangga**, tomboldengan tulisan **Jeruk**, dan tomboldengan tulisan **Apel**.

Gambar Contoh dialog alert yang berfungsi sebagai dialog pilihan:



Gambar 1.2. Dialog pilihan

# Praktek

Pada bagian ini, akan dibahas mengenai praktek yang akan dilakukan. Untuk mempermudah pencatatan hasil praktek dan penarikan kesimpulan terhadap hasil praktek tersebut, Anda dapat menggunakan form pada **Appendix 1.**

## Praktek 1: Hello World dengan MakeToast

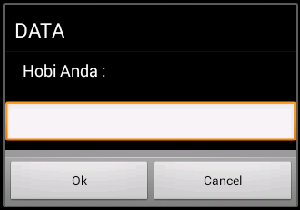
1. Buat *Script* dengan nama: **lab\_1\_HelloWorld\_makeToast.py**
2. Tambahkan instruksi agar jika *Script* dijalankan maka akan tampil splash label dengan tulisan: **Hello World.**

## Praktek 2: Masukan dengan dialogGetInput

1. Buat *Script* dengan nama: **lab\_2\_dialogGetInput.py**
2. Tambahkan instruksi dengan menggunakan method **dialogGetInput()** , untuk membaca nama user melalui masukan dialog.
3. Tambahkan instruksi untuk membuat *splash label* yang menampilkan tulisan: **Selamat datang di Android**
4. Tambahkan instruksi untuk membuat *splash label* yang menampilkan tulisan: **nama** yang di-*input* *user* melalui dialog.
5. Tambahkan instruksi dengan menggunakan method **dialogGetInput()** , untuk membaca tulisan yang ditulis user melalui dialog.

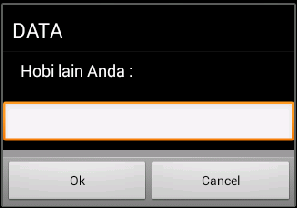
## Praktek 3: Menggunakan Alert

1. Buat *Script* dengan nama: **lab\_3\_alert.py**.
2. Tambahkan instruksi untuk membuat dialog alert yang berjudul: **PERHATIAN**, dan berisi pesan: **Anda akan diminta untuk mengisi data Anda**
3. Tambahkan instruksi dengan menggunakan method **dialogGetInput()** , untuk membuat dialog masukan seperti berikut:

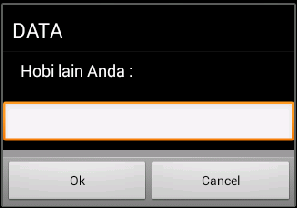


Masukan user tersebut akan tersimpan dalam variabel **pil1**

1. Tambahkan instruksi dengan menggunakan method **dialogGetInput()** , untuk membuat dialog masukan seperti berikut:

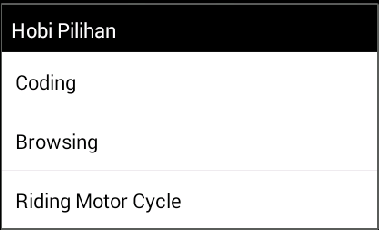


1. Tambahkan instruksi dengan menggunakan method **dialogGetInput()** , untuk membuat dialog masukan seperti berikut:



1. Tambahkan instruksi untuk membuat dialog alert yang berjudul: **Hobi Pilihan**, dan berisi item pilihan yang tadi sudah diisi oleh user

Berikut adalah contoh tampilan dialog alert:



# Analisa Hasil Praktek

Berdasarkan praktek yang telah Anda lakukan, buat kesimpulan:

# Tugas

# Appendix 1: Form Data Praktek

Nama : ……………………….

NIM : ……………………….

Tgl : ……………………….

|  |  |
| --- | --- |
| **Penggalan *Source code*** | **Keluaran / *Output*** |
|  |  |
|  |  |
|  |  |