

BAB VII

SWIFT FOR iOS

7.1 Tujuan

1. Praktikan mampu mengenal bahasa pemrograman *swift*
2. Praktikan mampu membuat program sederhana menggunakan Xcode
3. Praktikan dapat mengetahui fitur-fitur yang ada pada Xcode
4. Praktikan mampu membuat aplikasi yang dapat jalan di semua perangkat iPhone
5. Praktikan mampu menggunakan *segue*
6. Praktikan mampu menggunakan *constrains*

7.2 Dasar Teori

7.2.1 Pengenalan iOS

iOS adalah sistem operasi perangkat bergerak yang dikembangkan dan didistribusikan oleh Apple Inc. Sistem operasi ini pertama diluncurkan tahun 2007 untuk iPhone dan iPod Touch, dan telah dikembangkan untuk mendukung perangkat Apple lainnya seperti iPad dan Apple TV. Tidak seperti Windows Phone (Windows CE) Microsoft dan Android Google, Apple tidak melisensikan iOS untuk dapat diinstal di perangkat keras non-Apple. Pada 12 September 2012, App Store Apple berisi lebih dari 700.000 aplikasi iOS, yang secara kolektif telah diunduh lebih dari 30 miliar kali. SO ini memiliki pangsa pasar 14,9% untuk unit sistem operasi perangkat bergerak telepon cerdas yang dijual pada kuartal ketiga 2012, terbanyak setelah Android Google. Pada bulan Juni 2012, iOS mencakup 65% konsumsi data web perangkat bergerak (termasuk di iPod Touch dan iPad). Pada pertengahan 2012, terdapat 410 juta perangkat bergerak yang diaktifkan. Menurut Apple pada tanggal 12 September 2012, 400 juta perangkat bergerak iOS telah dijual sepanjang bulan Juni 2012.

Antarmuka pengguna iOS didasarkan pada konsep manipulasi langsung menggunakan gerakan multisentuh. Elemen kontrol antarmukanya meliputi slider, switch, dan tombol. Interaksi dengan SO ini mencakup gerakan seperti geser, sentuh, jepit, dan jepit buka, masing - masing memiliki arti tersendiri dalam konteks iOS dan antarmuka multi sentuhnya. Akselerometer internalnya dipakai oleh sejumlah aplikasi agar bisa merespon terhadap pengguncangan alat (misalnya membatalkan tindakan) atau memutarnya dalam tiga dimensi (misalnya beralih dari mode potret ke lanskap). iOS diturunkan dari OS X, yang memiliki fondasi Darwin dan karena itu iOS merupakan sistem operasi Unix.



Gambar 7.1 iOS 10

(Sumber : Wikipedia.org)

7.2.2 Swift

Swift adalah bahasa pemrograman objek fungsional untuk pengembangan iOS dan OS X yang dibuat oleh Apple. Didesain untuk berdampingan dengan Objective-C dan menghindari program dari kode yang salah. Swift diperkenalkan oleh Apple pada acara tahunan WWDC (*World Wide Developer Conference*) 2014. Bahasa ini dibangun dengan compiler LLVM yang termasuk dalam Xcode 6 beta. Sebuah buku berisi 500 halaman petunjuk, *The Swift Programming Language*, dirilis bersamaan dengan WWDC, dan tersedia secara gratis di *iBooks*. Pada tanggal 2 Juni 2014, aplikasi WWDC menjadi yang pertama dirilis ke publik yang ditulis dengan Swift. Pada tahun 2014 juga, Swift diupdate menjadi Swift 1.2 yang membawa beberapa perbaikan. Pada tahun 2015 diluncurkan Swift 3 sebagai pengembangan lebih lanjut dari Swift 1.2. Pada tahun tersebut pula Swift menjadi Bahasa pemrograman *open source* dan tersedia dibawah *Apache Liscence*. Pada tahun 2016 bersamaan dengan diluncurkannya MacOS Sierra, iOS 10, dan WatchOS 3, Swift kembali diupdate menjadi Swift 3 dengan IDE nya menjadi XCode



Gambar 7.2 Logo Swift

(Sumber : Wikipedia.org)

7.2.3 Xcode

Xcode adalah *IDE* yang lengkap untuk membangun aplikasi MacOS, iOS, WatchOS, dan tvOS, dengan adanya Xcode, sebuah aplikasi yang dapat membantu pengembang / developer dapat melakukan hal lebih dari sebelumnya yaitu lebih cepat, efisien, dan aplikasi yang dihasilkan berkualitas tinggi. Xcode dapat otomatis mengkonfigurasi aplikasi kita untuk menggunakan layanan terbaru Apple dan mengelola banyak gambar kita dalam katalog aset terpadu, dan membantu kita dalam merancang sebuah aplikasi yang terlihat cantik pada iOS, watchOS, tvOS atau macOS. Dengan adanya Xcode pengembang sangat mudah untuk membuat aplikasi dan menjalankan tes/simulasi, menganalisis kode, dan memantau kinerja yang dilakukan.

Saat ini versi Xcode yang terbaru adalah Xcode 8.0 yang sudah terintegrasi dengan SDK dan Simulator dari seluruh OS yang dimiliki oleh Apple. Xcode tersedia gratis bagi pengguna macOS dan dapat diunduh dari Mac App Store.



Gambar 7.3 Logo Xcode

(Sumber : Modul Praktikum MDP 2016)

7.3 Langkah Kerja

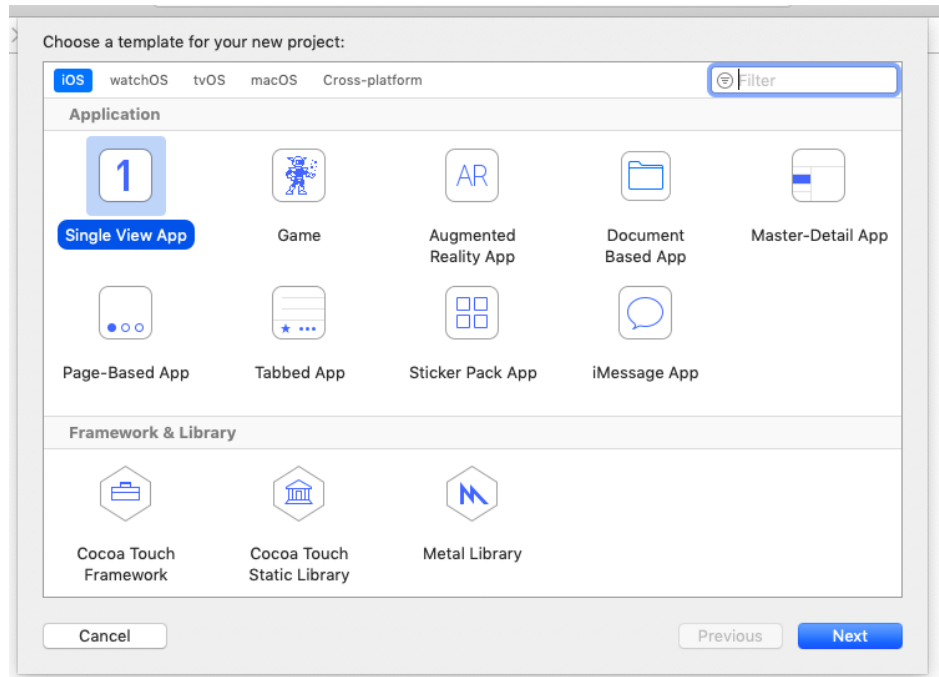
Pada percobaan ini akan dibuat aplikasi *Random Number Generator* dengan melakukan *Login* pada halaman pertama. Aplikasi ini menggunakan metode *random number generator*, memakai fungsi dalam Xcode.

1. Membuka aplikasi Xcode

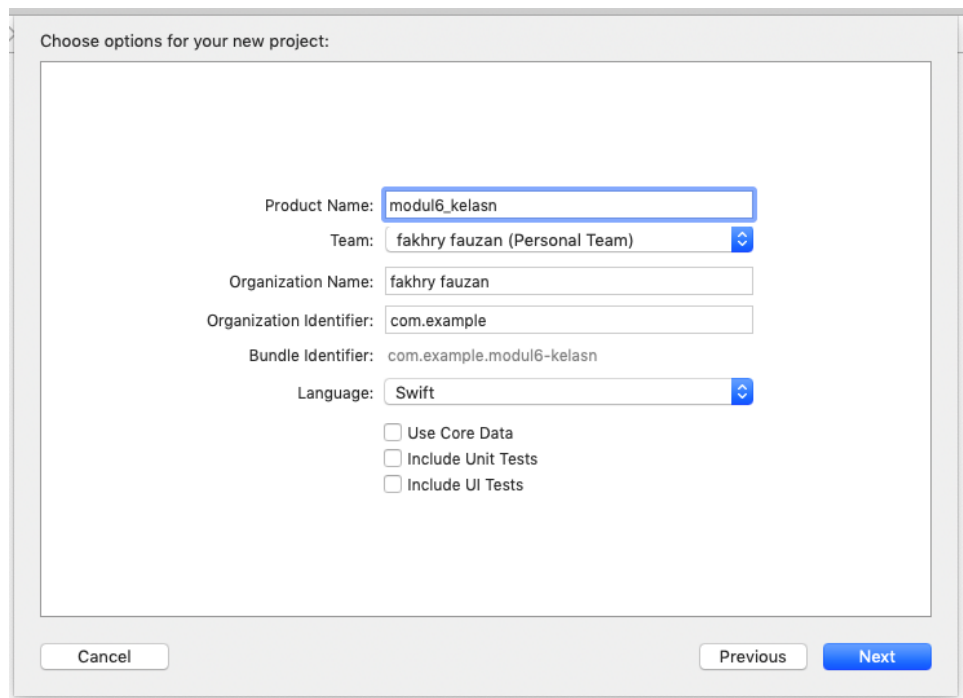


Gambar 7.4 Tampilan Awal Xcode

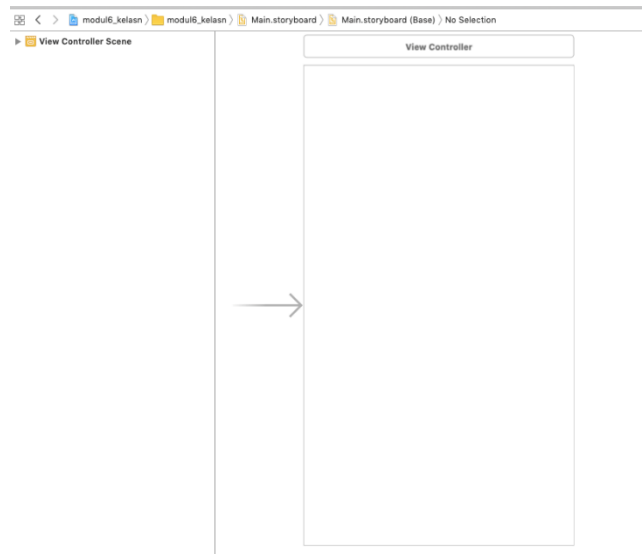
2. Pilih *Create a new Xcode project*
 - a. Memilih *template* untuk *project* baru, dalam hal ini pilih *Single View Application*

Gambar 7.5 Memilih *Single View App*

- b. Memberikan nama *project* yang akan dibuat, lalu pilih *device* iPhone

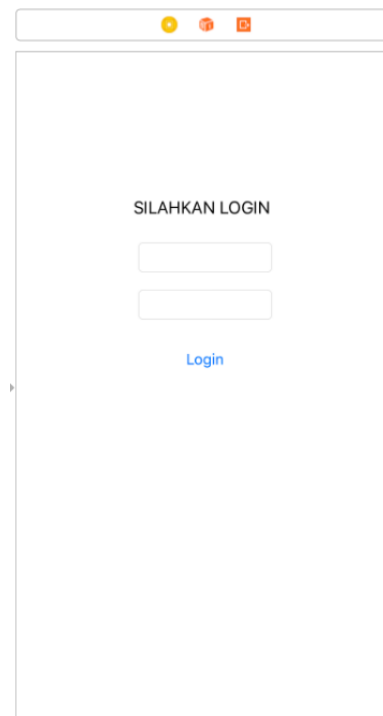
Gambar 7.6 Memberi Nama *Project*

c. Pilih Main.storyboard



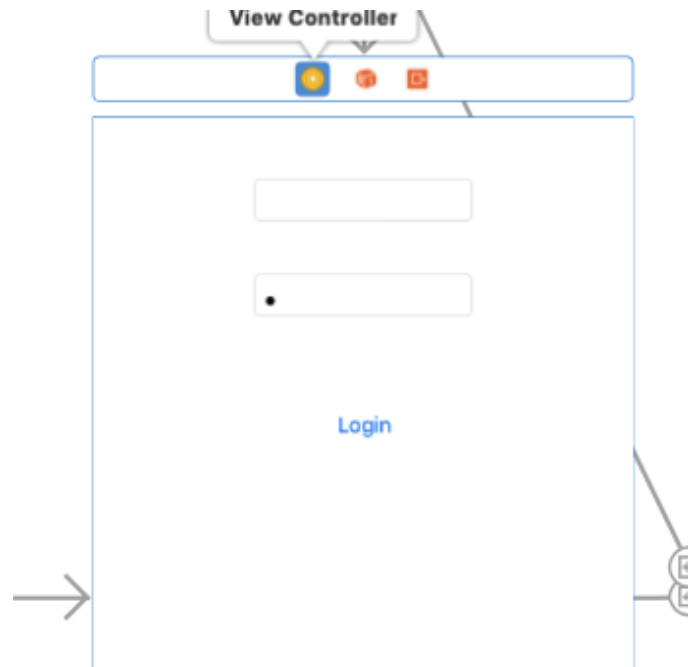
Gambar 7.7 Memilih *Main Storyboard*

3. *Drag and Drop Label, Text Field, dan Button* sesuai kebutuhan dan estetika



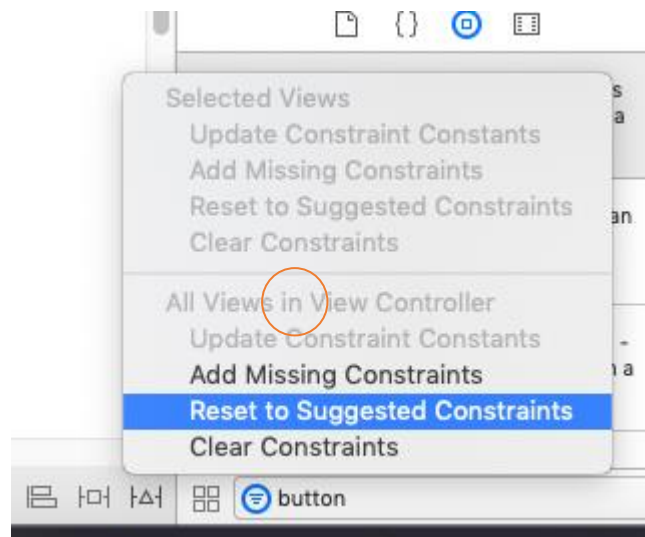
Gambar 7.8 Membuat Tampilan Aplikasi

4. Tekan *ViewController* pada *View*



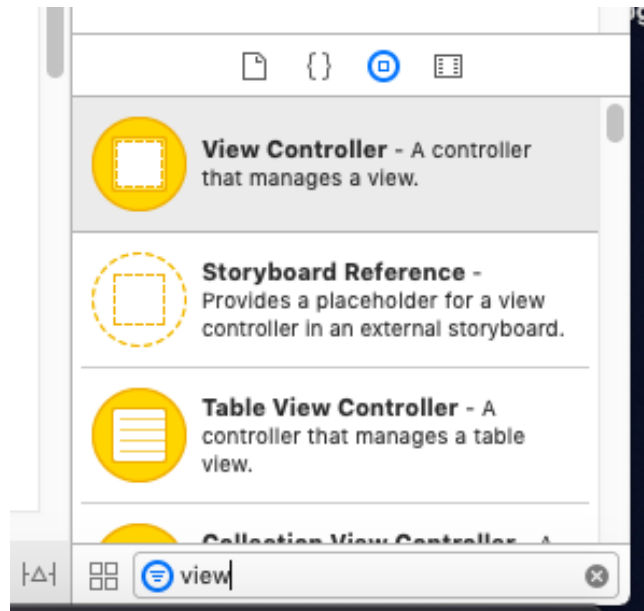
Gambar 7.9 Menekan ViewController pada View

5. Tekan *Resolve Auto Layout Issues*, dan tekan *Reset to Suggested Constraints* untuk melakukan *constraints* otomatis

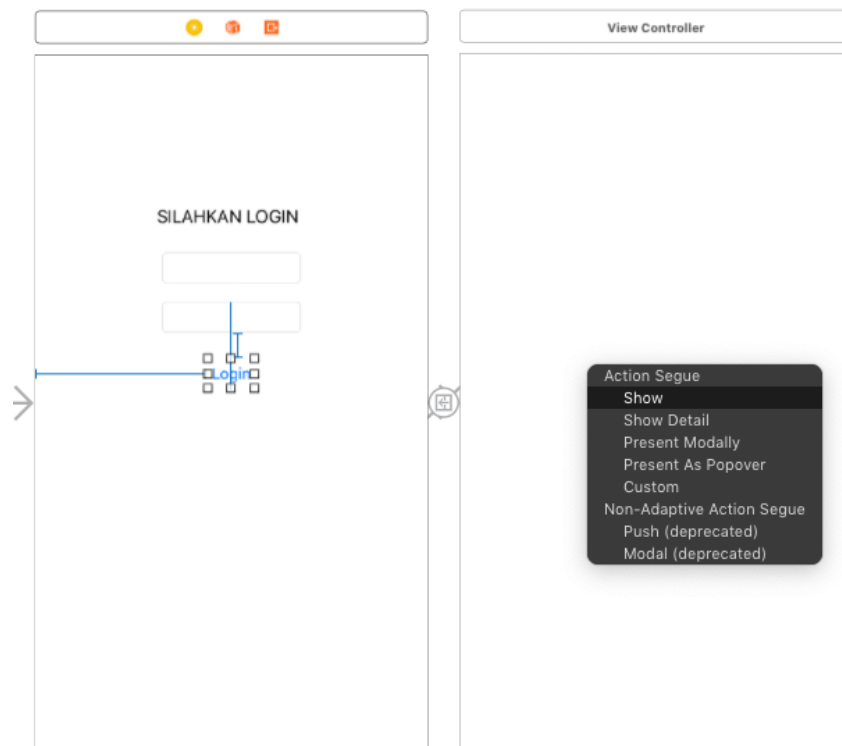


Gambar 7.10 Melakukan *Constraint* Otomatis

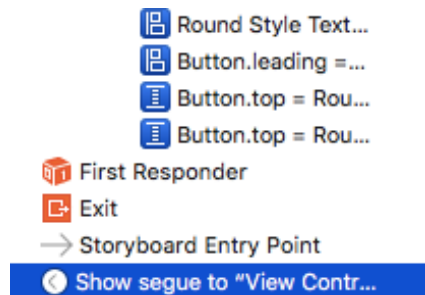
6. Tambahkan *view*

Gambar 7.11 Menambahkan *Segue*

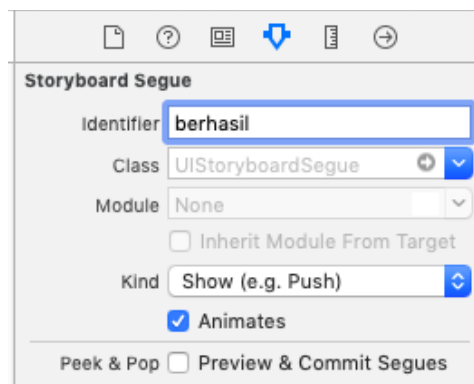
7. Buat *segue*, *segue* berfungsi sebagai arahan disaat pemindahan *view*.

Gambar 7.12 Membuat *Segue*

8. Berinama *segue* untuk *view* pertama

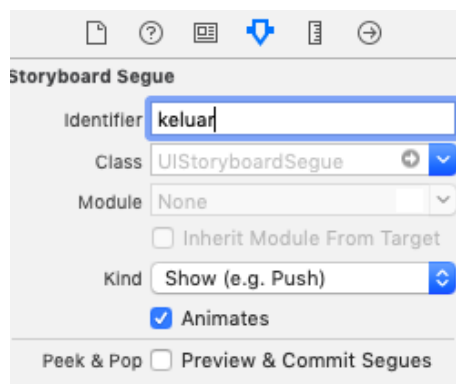


Gambar 7.13 *Show Segue*



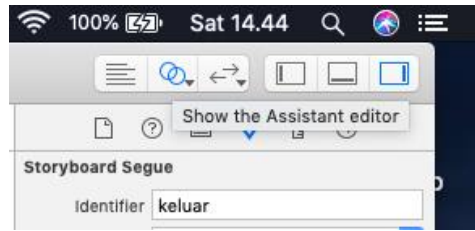
Gambar 7.14 Memberi nama *view* pertama

9. Begitu pula pada *view* kedua



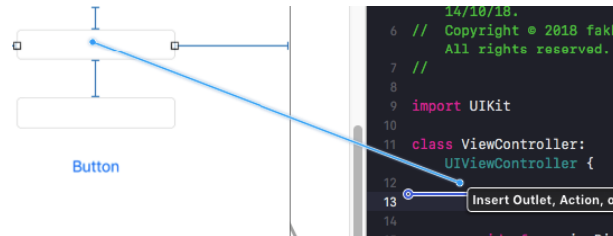
Gambar 7.15 Memberi nama *view* kedua

10. Tekan *show the assistant editor*

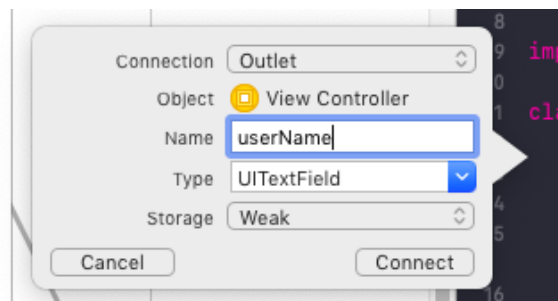


Gambar 7.16 Menekan *Show the Assistant Editor*

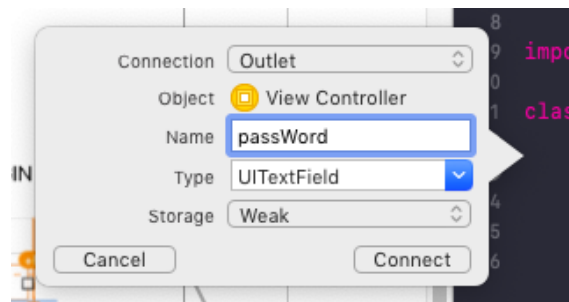
11. Koneksikan semua *widget* dengan halaman *viewController.swift*. Masukkan *button*, dan *textfield* ke *source code* dengan *drag* sambil menahan tombol *control* dari *main storyboard* ke halaman kode. Lalu beri nama *variable* bagi setiap objek.



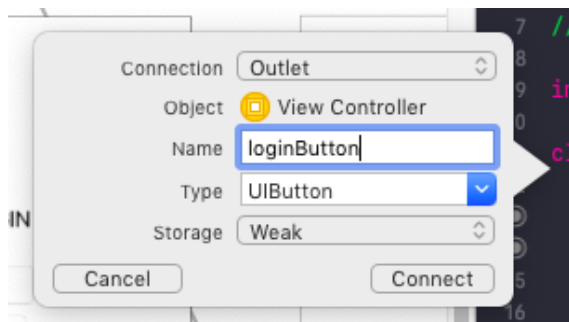
Gambar 7.17 Menghubungkan Widget dengan halaman *viewController.swift*



Gambar 7.18 Memberi nama *TextField1*

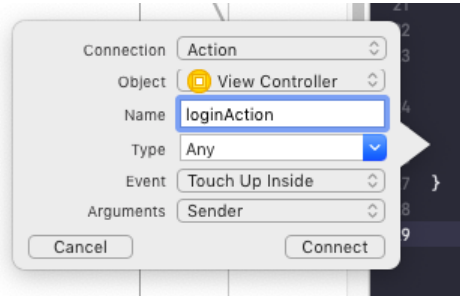


Gambar 7.19 Memberi nama *TextField2*



Gambar 7.20 Memberi nama *Button*

12. Pada *LoginAction* pilih *Action* pada *Connection*



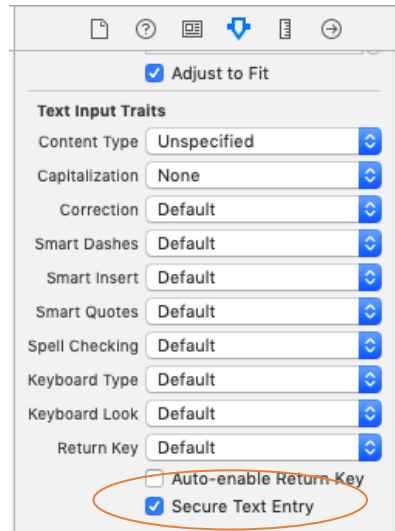
Gambar 7.21 Memberi nama *Action*

Tambahkan *code* berikut di dalam *loginAction()* ,

```
let user = "mdp"
let pass = "123456"

if (userName.text == user && passWord.text ==
pass) {
    performSegue(withIdentifier: "berhasil",
sender: self)
}

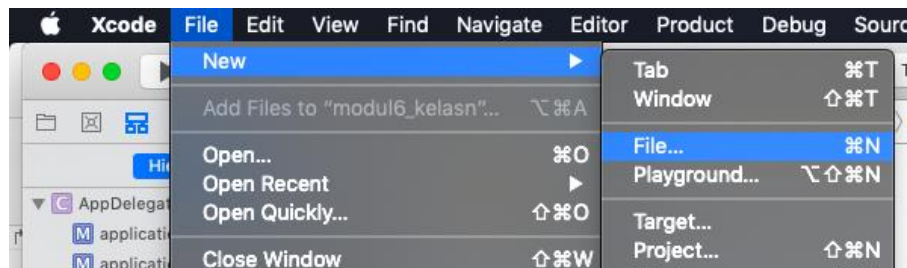
else if (userName.text == "" && passWord.text ==
"") {
    let alert = UIAlertController(title: "Login
Gagal", message: "Username/Password Kosong", preferredStyle:
UIAlertControllerStyle.alert)
    alert.addAction(UIAlertAction(title: "OK",
style: UIAlertActionStyle.cancel, handler: nil))
    self.present(alert, animated: true)
}
else {
    let alert = UIAlertController(title: "Login
Gagal", message: "Username/Password Salah", preferredStyle:
UIAlertControllerStyle.alert)
    alert.addAction(UIAlertAction(title: "OK",
style: UIAlertActionStyle.cancel, handler: nil))
    self.present(alert, animated: true)
}
}
```



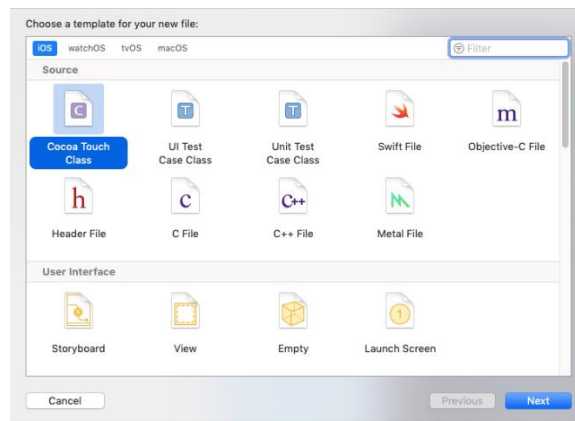
Gambar 7.22 Secure Text Entry

Klik *textfield* ke 2 klik *show the attributes inspector*, dan centang *Secure Text Entry*, guna men-dots *text* yang masuk.

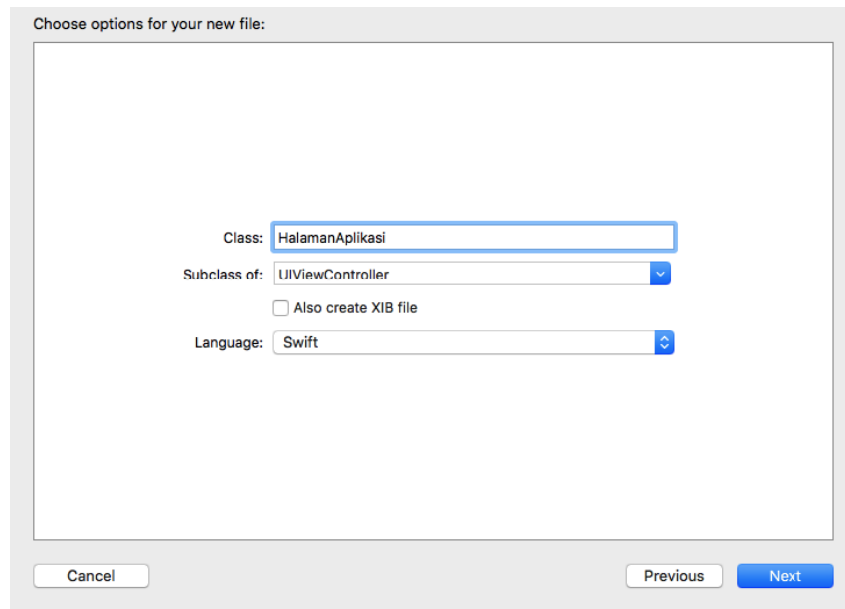
13. Buat file baru, Pilih *Swift File*, beri nama HalamanAplikasi, Ganti *Foundation* menjadi *UIKit* di HalamanAplikasi.swift



Gambar 7.23 New File



Gambar 7.24 Memilih Cocoa Touch Class

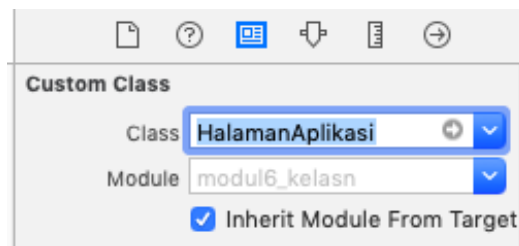


Gambar 7.25 Memberi Nama HalamanAplikasi

14. Pilih *view* ke dua dan pilih *class* HalamanAplikasi



Gambar 7.26 Memilih HalamanAplikasi (1)



Gambar 7.27 Memilih HalamanAplikasi (2)

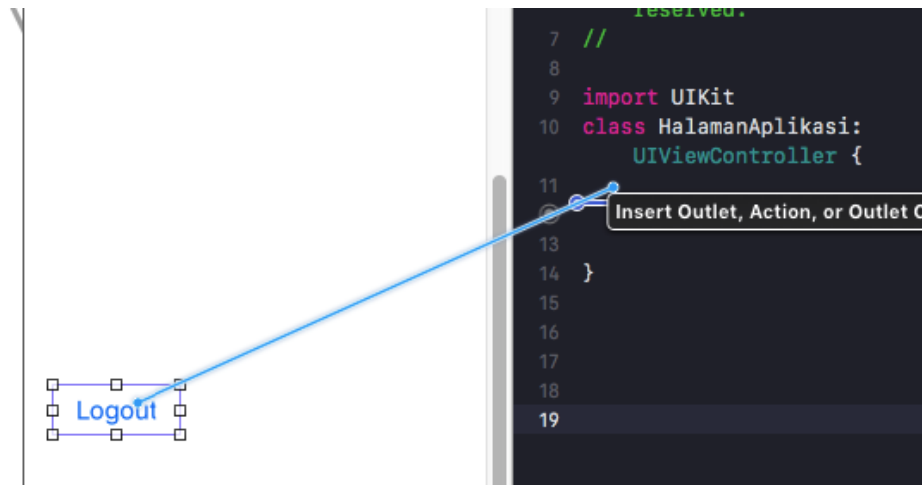
15. Pilih HalamanAplikasi.Swift



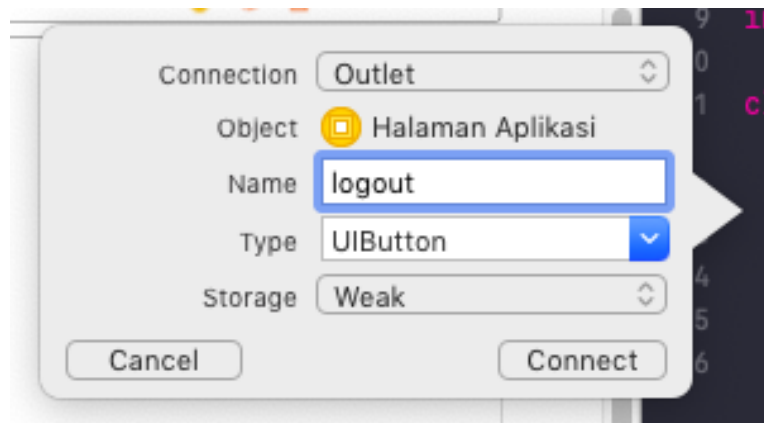
Gambar 7.28 Memilih HamalanAplikasi.swift

Dan pastikan yang terbuka adalah HalamanAplikasi.swift

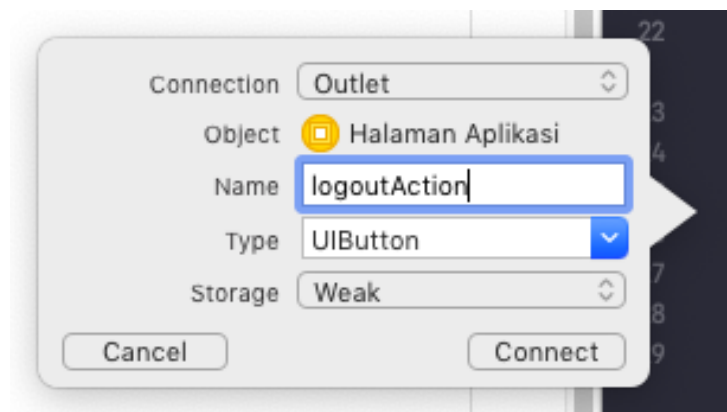
16. Tambahkan *Function* pada *Logout*



Gambar 7.29 Memasukkan Fungsi Tombol *Logout* ke *Code*



Gambar 7.30 Membuat *button logout*



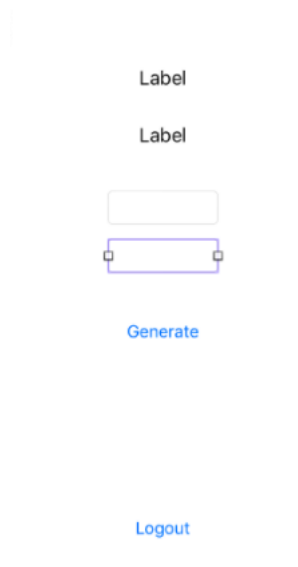
Gambar 7.31 Membuat *LogoutAction*

Di *function* *logoutAction* tambahkan *code* berikut

```
performSegue(withIdentifier: "keluar", sender:
self)
```

17. *Clean* dan *run* program

18. *Drag and Drop* *Label*, *Text Field*, dan *Button* sesuai kebutuhan dan estetika

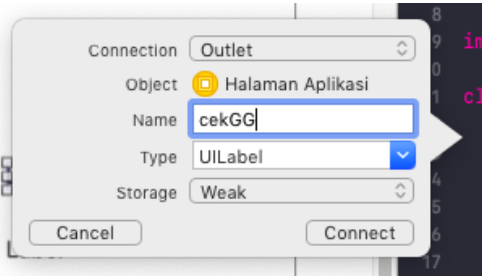


Gambar 7.32 Membuat Tampilan Halaman *Generate Number*

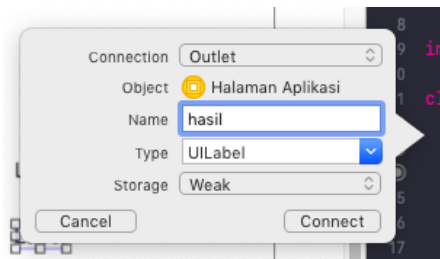
19. Masukkan *label*, *button*, dan *textfield* ke *source code* dengan *drag* sambil menahan tombol *control* dari *main storyboard* ke halaman kode. Lalu beri nama *variable* bagi setiap objek.



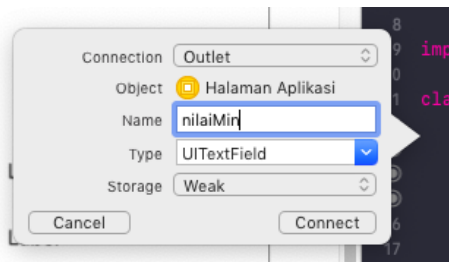
Gambar 7.33 Memasukkan komponen ke dalam *source code*



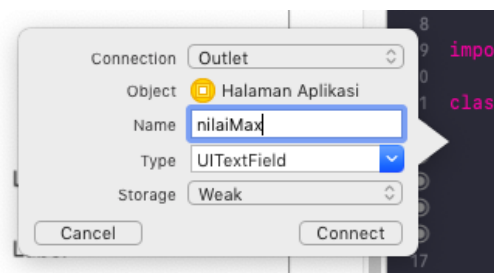
Gambar 7.34 memberi nama variabel cekGG



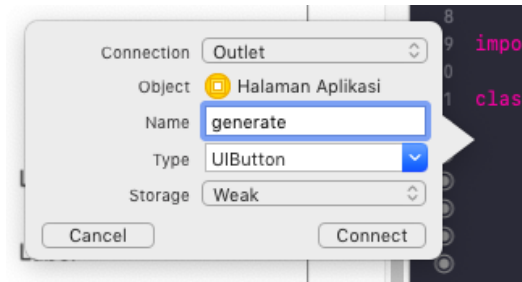
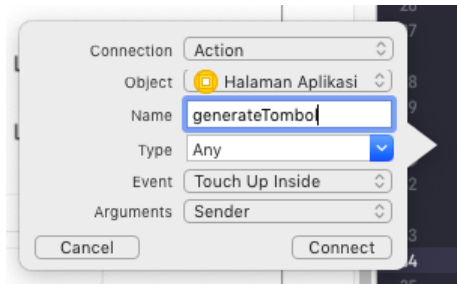
Gambar 7.35 memberi nama variabel hasil



Gambar 7.36 memberi nama variabel nilaiMin



Gambar 7.37 memberi nama variabel nilaiMax

Gambar 7.38 memberi nama variabel *generate*Gambar 7.39 memberi nama *action* generateTombol

Keterangan :

Label pertama = cekGG

Label kedua = hasil

Textfield pertama = nilaiMin

Textfield kedua = nilaiMax

Button = *generate*

Button Action = generateTombol

20. Masukkan *source code* seperti di bawah ini

```
var hasilakhir = Int()
override func viewDidLoad() {
    super.viewDidLoad()
    nilaiMin.text = "0"
    nilaiMax.text = "100"
    hasil.text = "Hasil"
    cekGG.text = "Genap / Ganjil ?"
}
```

21. Buat *function* berikut

```
func ganjilgenap(){
    if hasilakhir % 2 == 0 {
        cekGG.text = "Genap"
    }
}
```

```

        else{
            cekGG.text = "Ganjil"
        }
    }
}

```

22. Tambahkan *code* berikut di dalam `generateTombol()`

```

        let    nilai    =    UInt32((Int(nilaiMax.text!)!    -
Int(nilaiMin.text!)!) + 1)
        let random = arc4random_uniform(nilai)
        hasilakhir = Int(random)+Int(nilaiMin.text!)!
        hasil.text = String(hasilakhir);
        ganjilgenap()

```

23. *Run program* yang telah dibuat pada *Simulator*

SILAHKAN LOGIN

Login

Gambar 7.40 Tampilan *Login*



Gambar 7.41 Tampilan *Number Generator*

7.4 Hasil Percobaan dan Analisa

```

let user = "mdp"
let pass = "123456"

        if (userName.text == user && passWord.text == pass) {
            performSegue(withIdentifier: "berhasil", sender:
self)
        }

        else if (userName.text == "" && passWord.text == "") {
            let alert = UIAlertController(title: "Login Gagal",
message: "Username/Password Kosong", preferredStyle:
UIAlertControllerStyle.alert)
            alert.addAction(UIAlertAction(title: "OK", style:
UIAlertActionStyle.cancel, handler: nil))
            self.present(alert, animated: true)
        }
        else {
            let alert = UIAlertController(title: "Login Gagal",
message: "Username/Password Salah", preferredStyle:
UIAlertControllerStyle.alert)
            alert.addAction(UIAlertAction(title: "OK", style:
UIAlertActionStyle.cancel, handler: nil))
            self.present(alert, animated: true)
        }
    }
}

```

Source code diatas merupakan *source code Login*, dimana pada *source code* tersebut berisi *username* “mdp” dan *password* “123456” untuk melakukan proses *login*. Jika autentikasi berhasil maka akan dilakukan *segue* “berhasil” yang mana merupakan perpindahan halaman dari halaman *Login* ke HalamanAplikasi. Ketika *username* dan *password* yang dimasukkan salah maka akan muncul pesan “Username/Password Salah” dan ketika *username* dan *password* yang tidak dimasukkan maka akan muncul pesan “Username/Password Kosong”.

```
PerformSegueWithIdentifier("keluar", sender: self)
```

Source code diatas merupakan *source code Logout*. Pada *source code* ini jika *user* menekan tombol *Logout* maka akan dilakukan *segue* “keluar” yang mana halaman akan berpindah dari HalamanAplikasi ke halaman *Login*.

```

var hasilakhir = Int()
override func viewDidLoad() {
    super.viewDidLoad()
    nilaiMin.text = "0"
}

```

```

nilaiMax.text = "100"
hasil.text = "Hasil"
cekGG.text = "Genap / Ganjil ?"

```

Pada *source code* diatas, diberikan nilai awal pada *label* yang ada pada HalamanAplikasi. Nilai minimal (nilaiMin) diatur dengan nilai awal 0 dan untuk nilai maksimal (nilaiMax) diatur dengan nilai awal 100. Hasil memiliki nilai awal “Hasil” dan cekGG memiliki nilai awal “Genap / Ganjil?”

```

let nilai = UInt32((Int(nilaiMax.text!)! - Int(nilaiMin.text!)!)+1)
let random = arc4random_uniform(nilai)
hasilakhir = Int(random)+Int(nilaiMin.text!)!
hasil.text = String(hasilakhir);
ganjilgenap()

```

Untuk melakukan *generate* nomor menggunakan *source code* diatas. Ketika tombol *generate* ditekan maka akan diambil nilai dari nilaiMax – nilaiMin +1. Agar nilai yang dihasilkan menjadi *random* dengan menggunakan *arc4random_uniform* dari nilai yang telah didapat sebelumnya. Hasilakhir berupa nomor *random* dijumlahkan dengan *integer* dari nilaiMin tadi. Kemudian hasilakhir dimasukkan pada label hasil. Kemudian dilakukan *method* ganjilgenap().

```

func ganjilgenap(){
    if hasilakhir % 2 == 0 {
        cekGG.text = "Genap"
    }
    else{
        cekGG.text = "Ganjil"
    }
}

```

Method ganjilgenap() digunakan untuk melakukan *set* label dengan nilai “Genap” atau “Ganjil” sesuai dengan nilai dari hasil akhir di mod 2. Jika hasilakhir mod 2 == 0 maka cekGG akan bernilai “Genap” dan jika hasilakhir mod 2 == 1 maka cekGG akan bernilai “Ganjil”.

Berikut adalah hasil tangkapan layar mengenai aplikasi ini,

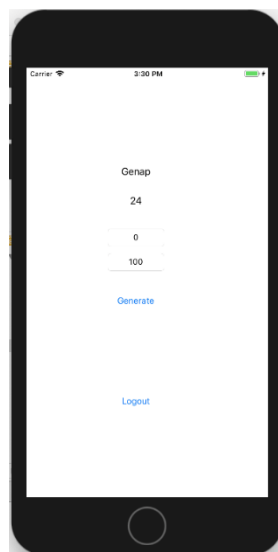


SILAHKAN LOGIN

Login

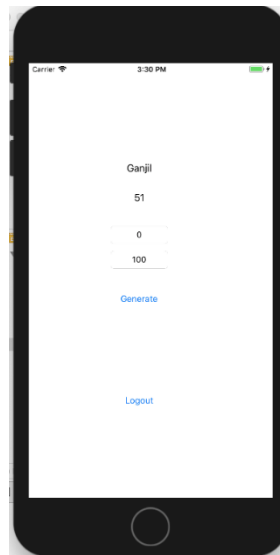
Gambar 7.42 Tampilan Program pada Halaman Login

Gambar diatas merupakan tampilan halaman *login*, dimana *Textfield* pertama (yang atas) digunakan untuk memasukkan *username* dan *Textfield* kedua (yang bawah) digunakan untuk memasukkan *password*. Setelah memasukkan *username* dan *password* kemudian menekan *button Login* untuk masuk ke Halaman Aplikasi.



Gambar 7.43 Tampilan Program pada Halaman Aplikasi saat genap

Gambar diatas adalah tampilan program saat melakukan *generate random* nomor dari nilai minimalnya 0 dan nilai maksimalnya 100. Nilai *random* yang keluar adalah 24. Karena $24 \bmod 2$ adalah 0 maka diatas angka 24 tersebut tertulis “Genap”.



Gambar 7.44 Tampilan Program pada Halaman Aplikasi saat ganjil

Gambar diatas adalah tampilan program saat melakukan *generate random* nomor dari nilai minimalnya 0 dan nilai maksimalnya 100. Nilai *random* yang keluar adalah 52. Karena $51 \bmod 2$ adalah 1 maka diatas angka 51 tersebut tertulis “Genap”.

7.5 Kesimpulan

1. Inisialisasi pada Xcode dapat dilakukan dengan melakukan *drag and drop*.
2. Bahasa pemrograman *swift* pada iMac dan Mac mini memiliki sedikit perbedaan.
3. Untuk menghubungkan anantara *segue* satu dengan *segue* lainnya hanya perlu dilakukan menarik *segue* tersebut ke halaman yang ingin dituju.
4. Untuk melakukan pengujian aplikasi menggunakan *simulator*.
5. Untuk menambahkan komponen seperti Label, *TextField*, maupun *Button* dapat dilakukan secara *drag and drop*.
6. Pemberian nama *segue* dapat dilakukan dengan membuka *View Controller* terlebih dahulu.
7. Sebelum dilakukan *run* program, terlebih dahulu dilakukan *clean*.
8. *Connection Action* digunakan memberikan perintah jika fitur diberi aksi seperti menekan *Button*.