

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Penelitian Terdahulu

Dalam mendukung penelitian kali ini, berikut adalah daftar dari jurnal dan buku yang penulis gunakan sebagai referensi penelitian. Selain membantu dalam membuat aplikasi, ilmu yang ada pada referensi berikut juga banyak memberikan informasi baru yang sangat bermanfaat. Penulis menggunakan referensi berikut untuk menjadi acuan dalam menyempurnakan aplikasi *e-commerce* yang penulis buat.

Table 2.1 Tabel Referensi Studi Relevan

No.	Nama	Judul	Metode Analisa
1.	Ibrahim, Muhammad Yusuf. (2016)	Penerapan Hukum Transaksi <i>Electronic Commerce</i> Terhadap Perlindungan Konsumen	Menjadikan referensi tampilan UI dalam membuat sistem aplikasi yang disukai masyarakat.
2.	A. Wulandari, Priyono. (2014)	<i>E-Commerce</i> Pakaian Pada Distro Bahana <i>Shop</i>	Menjadikan referensi dalam melakukan perancangan tampilan UI pada aplikasi yang baik.
3.	O. Muhammad Muslihudin. (2016).	Analisi Dan Perancangan Sistem Informasi Menggunakan Model Terstruktur Dan UML	Menjadikan referensi dalam melakukan perancangan sistem.
4.	K. Wiji Susanti. (2015).	Pengembangan <i>Electronic Commerce</i> Dalam Proses Meningkatkan UKM	Menjadikan referensi dalam membuat tampilan tampilan UI

		pada Dekranasda Kabupaten Piringsewu	pada aplikasi <i>e-commerce</i> .
5.	M. Muslihudin, W. Wulandari, and Mei Listiarini. (2017).	Perancangan Aplikasi <i>Business</i> Berbasis <i>Business to Consumer</i> .	Menjadikan referensi dalam membuat tampilan tampilan UI pada aplikasi <i>e-commerce</i> .
6.	Wulandari. (2013).	Analisi Dan Implementasi Sistem <i>E-Commerce</i> Furniture Pada Toko Jaya.	Menjadikan referensi dalam membuat tampilan tampilan UI pada aplikasi <i>e-commerce</i> .
7.	D. Irawan, Y. Rahsel, and T. Udin. (2017).	Perancangan Electronic Commerce Berbasis B2C Pada Toko Atk Sindoro.	Menjadikan referensi dalam membuat tampilan tampilan UI pada aplikasi <i>e-commerce</i> .

Yang membedakan penelitian penulis dengan penelitian yang sudah ada adalah penulis mengembangkan aplikasi ini murni dengan menggunakan bahasa swift. Penulis juga mengembangkan aplikasi ini dengan fitur Swift UI yang baru saja diluncurkan pada 2021. Metode Swift UI belum banyak digunakan karena menggunakan RAM yang sangat besar pada MacBook. Hal ini disebabkan adanya 2 *simulator* yang berjalan secara bersamaan.

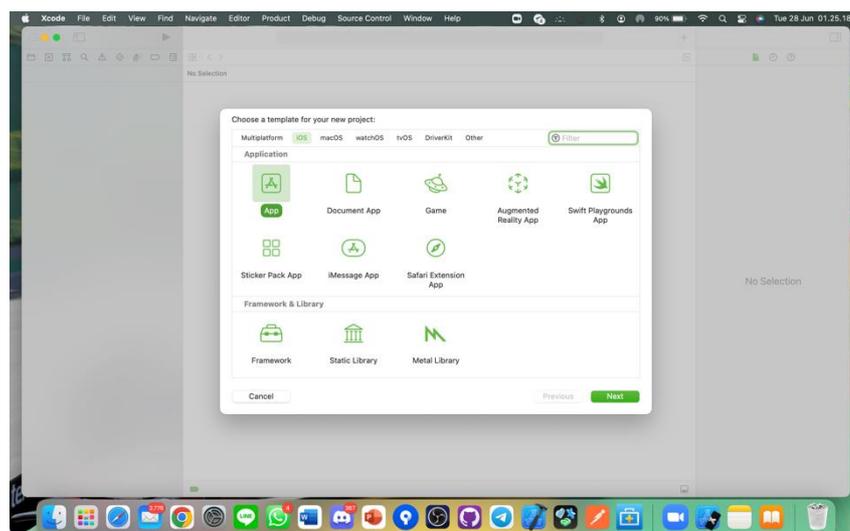
2.2 Swift

Swift merupakan bahasa pemrograman objek fungsional untuk pengembangan iOS dan OS X yang dibuat oleh Apple. Swift didesain untuk berdampingan dengan Objective-C dan menghindari program dari kode yang salah. Swift diperkenalkan oleh Apple pada acara tahunan Apple Worldwide Developer conference (WWDC) tahun 2014 [11]. Swift dibangun dengan compiler LLVM yang termasuk dalam Xcode 6 beta [2]. Xcode sendiri

merupakan aplikasi yang sudah disediakan oleh Apple untuk menjalankan bahasa pemrograman Swift dan melakukan *coding* untuk membuat aplikasi iOS pada Mac. 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 [2].

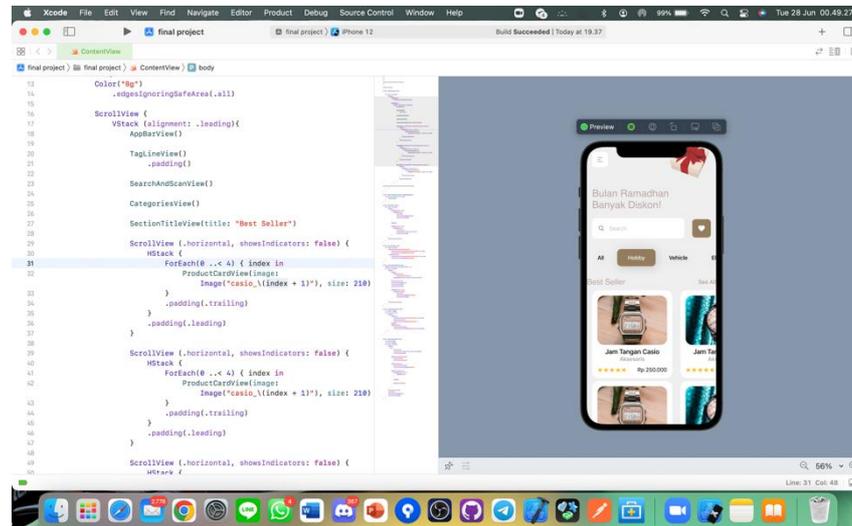
Swift bersifat gratis dan *open source*, dan tersedia untuk banyak pengembang, pendidik, dan siswa di bawah lisensi *open source* Apache 2.0 [1]. Swift menyediakan binari untuk macOS dan Linux yang dapat mengkompilasi kode untuk iOS, macOS, watchOS, tvOS, dan Linux. Dan untuk membantu Swift tumbuh menjadi bahasa yang lebih kuat lagi, Swift telah membuat komunitas tempat para pengguna untuk berkumpul dan dapat berkontribusi langsung ke kode sumber Swift. Karena Swift termasuk kedalam bahasa yang modern, maka bahasa yang digunakan sangatlah memudahkan penggunaannya [1][2].

Swift adalah bahasa pemrograman yang kokoh dan intuitif yang diciptakan Apple untuk mengembangkan aplikasi untuk iOS, Mac, Apple TV, dan Apple Watch. Bahasa ini didesain untuk memberikan lebih banyak kebebasan kepada pengembang. Swift mudah digunakan dan bersifat sumber terbuka, jadi siapa pun yang memiliki ide dapat menciptakan sesuatu yang luar biasa dan sesuai kemauan pengguna.



Gambar 2.1. Menu untuk membuat aplikasi pada iOS, Mac, Apple TV, dan Apple Watch.

Swift adalah bahasa yang cepat dan efisien yang menyediakan laporan secara real-time dengan menggunakan pilihan Swift UI dan dapat digabungkan dengan mulus ke dalam kode Objective-C yang ada. Jadi, pengembang dapat menulis kode yang lebih aman dan andal, hemat waktu, dan bahkan menciptakan pengalaman aplikasi yang lebih banyak.



Gambar 2.2. Contoh penggunaan Swift UI dalam membuat aplikasi.

Dapat dilihat pada gambar diatas, Swift mampu memberikan akses atau tampilan berupa *Live Coding*. Fitur ini Bernama Swift UI. *Live Coding* dapat memberikan kemudahan kepada pengguna agar dapat mengetahui apa yang sedang mereka lakukan tanpa ada *delay*. Dengan adanya fitur ini, pengguna dapat lebih mudah, cepat, dan dapat mengeluarkan kreatifitasnya tanpa perlu takut mendapatkan *coding error*.

Apple meng-klaim bahwa Swift dapat menghasilkan performa yang lebih baik. Dengan klaim tersebut maka swift menghasilkan aplikasi yang lebih baik. Apple juga mengklaim bahwa Swift akan 2,6x lebih cepat jika dibandingkan dengan Objective-C, dan hingga 8,4x lebih cepat jika dibandingkan dengan Python 2.7. Aplikasi Swift melakukan hal hebat seperti algoritme pencarian umum akan selesai jauh lebih cepat menggunakan Swift. Swift sendiri dijalankan dengan aplikasi yang sudah dibuat oleh Apple.inc yaitu Xcode [1].

2.3 Xcode

Setiap proses *coding* atau *programming* tentunya akan menggunakan software khusus agar proses pengembangan bisa berjalan dengan baik, stabil dan lebih cepat. Bahasa pemrograman Swift untuk iOS Apple pun mempunyai sebuah software khusus untuk membangun aplikasi dengan optimal. *Software* tersebut adalah Xcode.

Sederhananya, Xcode adalah aplikasi untuk menjalankan Swift untuk membuat aplikasi pada iOS, seperti AndroidStudio yang merupakan aplikasi untuk menjalankan system aplikasi android. Xcode sendiri memiliki banyak tools untuk proses development seperti manajemen perangkat, debugger, UI creator, profiling tools, sumber kontrol, dokumentasi dan lain sebagainya. Pada awalnya, Xcode hanya bisa dijalankan pada sistem macOS Apple saja. Walaupun saat ini mungkin tersedia Xcode tutorial alternatif untuk platform non-MacOS, biasanya kinerja Xcode tetap tidak akan optimal.

Saat pertama membuka aplikasi Xcode, maka akan langsung terlihat Welcome Screen Xcode, Anda akan disambut dengan beberapa fitur aktivitas yang bisa dilakukan dengan aplikasi ini seperti membangun project aplikasi iOS, membuka project yang pernah dikerjakan, coba belajar coding Swift, atau melakukan kloning projek dari repositori Git.



Gambar 2.3. Tampilan Welcome Screen pada aplikasi Xcode

UI Xcode dibuat sedemikian rupa agar bisa menampilkan banyak *tools* yang berguna untuk melakukan coding. Beberapa *tools* yang ada untuk mempermudah *coding* pada Xcode adalah:

1. *Debug*

Debug adalah fitur wajib ada dalam semua software IDE, termasuk Xcode. *Debug* digunakan untuk melakukan fixing / perbaikan *error bugs* pada aplikasi. Kita juga dapat melihat hasil *debug* untuk memastikan kembali bahwa aplikasi masih mempunyai *bugs* atau tidak.

2. Navigator

Peranan fitur navigator adalah sebagai penyedia sumber daya proyek, seperti penggunaan Swift, *breakpoints*, *problems*, dan lain sebagainya yang berkaitan dengan navigasi project.

3. Inspektur

Inspektur adalah fungsi untuk memeriksa dan menyesuaikan elemen-elemen antarmuka, atribut file, dan lain-lain. *Tools* ini paling banyak digunakan dalam proses *interface builder* (pembuatan *interface*).

4. Editor

Editor merupakan *tools* paling penting dari Xcode yakni tempat untuk menulis kode-kode Swift. Semua aktivitas *coding* berlangsung di editor Xcode, mulai dari awal menulis *coding* hingga uji testing terakhir.

5. Simulator

Xcode dilengkapi dengan tombol *play and stop* serta *item dropdown* yang bermanfaat sebagai *simulator*. Anda bisa memanfaatkannya sebagai *simulator* dari aplikasi yang sudah dibuat, jadi Anda bisa mengecek sendiri seperti apa kinerja dari aplikasi yang dibuat. Ada beragam jenis *platform simulator* yang bisa dipilih, seperti iPhone 11, iWatch, Mac, dan lain-lain. Hal ini tergantung pada jenis aplikasi apa yang akan dibuat dan dikembangkan.

6. Bilah Status

Bilah status berada di posisi tengah atas pada menu dari Xcode. Kegunaan bilah status adalah untuk menampilkan progress dari project serta error atau kesalahan yang terjadi di project Anda. Fitur ini akan sangat memudahkan Anda untuk bekerja sekaligus memantau proses kerja.

7. View

Xcode dilengkapi dengan berbagai tombol *view*. Ada tombol *view* “tinjauan kode” yang bisa dipakai untuk membandingkan dua file Swift secara berdampingan. Ada juga tombol “*hide*” untuk menyembunyikan panel sisi kiri dan kanan agar pandangan Anda bisa terfokus langsung ada bagian tengah editor Swift saja.

2.4 Perbandingan Swift dan Bahasa lainnya.

Swift hanya bisa dilakukan pada sistem operasi MacOS dengan menggunakan aplikasi Xcode. Berbeda dengan Tokopedia dan Bukalapak yang dibuat dengan menggunakan JavaScript, JavaScript merupakan bahasa pemrograman *Multiplatform*. *Multiplatform* berarti sebuah aplikasi yang menggunakan bahasa pemrograman Javascript bisa dibuat dengan basis sistem operasi apapun, termasuk diantaranya adalah sistem operasi populer seperti sistem operasi Windows, Linux, Android, dan berbagai macam sistem operasi lainnya.

Selain berbasis sistem operasi, aplikasi yang menggunakan bahasa pemrograman berbasis Javascript juga banyak yang bisa dijalankan pada perangkat komputer dan perangkat komunikasi, seperti handphone yang tidak menggunakan sistem operasi tertentu.

Meskipun Javascript bisa lakukan dibanyak platform dan mudah dikembangkan, namun Javascript dikenal dengan kelemahannya dalam melakukan enkripsi dari bahasa pemrograman Javascript. Hal ini menyebabkan aplikasi yang menggunakan Javascript sangat rentan untuk diretas. Maka dari itu, akan sangat berbahaya untuk membuat sebuah aplikasi yang membutuhkan password dan data pribadi. Berbeda dengan Swift yang sudah diberikan perlindungan lebih oleh Apple. Data yang diambil juga akan di enkripsi oleh Apple untuk menghindari

Kelemahan yang terdapat dari bahasa pemrograman Javascript ini adalah terbatasnya objek. Hal ini dipengaruhi oleh kondisi umum dari bahasa pemrograman Javascript yang sanga simple dan juga sederhana, sehingga tentu saja hal ini akan sanga mempengaruhi fitur-fitur yang dimiliki oleh bahasa

pemrograman Javascript ini. Berbeda dengan Xcode selaku aplikasi yang menjalankan swift. Xcode memiliki banyak variasi fitur yang dapat digunakan. Objek yang terdapat dalam Xcode juga sangat banyak dan bervariasi. Selain itu, sangat mudah untuk melakukan input asset pada Xcode, karena ukuran nya yang tidak akan berubah sehingga menjamin asset yang kita masukan tidak menurun kualitasnya.