

PENGEMBANGAN MEDIA PEMBELAJARAN BERBASIS NATIVE IOS UNTUK MELACAK AKTIVITAS SELF-DIRECTED LEARNING (SDL) PADA PEMBELAJARAN BAHASA PEMROGRAMAN SWIFT

Rizal Hidayat

S1 Pendidikan Teknologi Informasi, Fakultas Teknik, Universitas Negeri Surabaya
rizalhidayat@mhs.unesa.ac.id

Bambang Sujatmiko

S1 Pendidikan Teknologi Informasi, Fakultas Teknik, Universitas Negeri Surabaya
bambang Sujatmiko@unesa.ac.id

Abstrak

Saat ini, teknologi informasi sudah berkembang pesat. Salah satu perkembangan di bidang teknologi ini ditandai dengan munculnya berbagai macam platform yang memiliki bahasa pemrograman tersendiri. Hal ini membuat peserta didik harus belajar lebih banyak dari yang sudah didapatkan dari Lembaga Pendidikan. Alasannya agar mampu berkompetisi di dunia teknologi yang semakin bersaing. Salah satu cara yaitu dengan melakukan pembelajaran mandiri atau Self-Directed Learning (SDL). Namun, kekurangannya yaitu peserta didik tidak tahu kemajuan belajarnya. Oleh karena itu, peneliti tertarik untuk mengembangkan media pembelajaran untuk melacak aktivitas pembelajaran peserta didik. Metode pengembangan yang digunakan yaitu metode waterfall dan menggunakan IDE Xcode 12 serta Swift 5.3 sebagai bahasa pemrogramannya. Dan menggunakan arsitektur MVC sebagai algoritma pemrograman. Sedangkan metode pengumpulan data menggunakan metode eksperimen. Data yang diperoleh dianalisis satu per satu untuk menentukan keberhasilan pengembangan media pembelajaran. Aktivitas *Self-Directed Learning* sendiri terdiri dari empat tahap yaitu persiapan, penerapan, pemantauan dan evaluasi. Hasilnya, media pembelajaran dapat dikembangkan menggunakan algoritma arsitektur MVC dan mampu melacak aktivitas-aktivitas tersebut dari peserta didik namun tidak bisa mengetahui penyebab keterlambatan atau ketidaksesuaian hasil peserta didik dengan tujuan pembuat materi.

Kata Kunci: *Pelacakan Aktivitas, Self-Directed Learning, Swift.*

Abstract

Currently, information technology has developed rapidly. One of the developments in the field of technology is marked by the emergence of various platforms that have their own programming languages. This makes students have to learn more than what has been obtained from educational institutions. The reason is to be able to compete in an increasingly competitive world of technology. One way is to do self-directed learning (SDL). However, the drawback is that students do not know their learning progress. Therefore, researchers are interested in developing learning media to track students' learning activities. The development method used is the waterfall method and uses the Xcode 12 IDE and Swift 5.3 as the programming language. And use MVC architecture as programming algorithm. While the method of data collection using the experimental method. The data obtained were analyzed one by one to determine the success of learning media development. Self-Directed Learning activity itself consists of four stages, namely preparation, implementing, monitoring and evaluation. As a result, learning media can be developed using the MVC architectural algorithm and are able to track these activities from students but cannot find out the causes of delays or discrepancies in student results with the objectives of the material maker.

Keywords: *Activity Tracking, Self-Directed Learning, Swift.*

PENDAHULUAN

Teknologi informasi sudah berkembang pesat. Perkembangan ini ditandai dengan munculnya berbagai macam platform seperti aplikasi web, aplikasi android, aplikasi iOS, aplikasi desktop, dan masih banyak lagi. Masalah utama yang ditimbulkan oleh perbedaan platform tersebut adalah bahasa pemrograman untuk mengembangkan aplikasi pada setiap platform juga

berbeda dan akan selalu muncul bahasa pemrograman baru. (Barlow-Jones, 2012)

Lembaga pendidikan seperti Sekolah Menengah Kejuruan (SMK) atau universitas merupakan tempat utama untuk belajar pemrograman. Lembaga ini memberikan ilmu tentang konsep pemrograman secara umum dan tidak secara khusus mempelajari bahasa pemrograman itu sendiri. Karena alasan inilah para peserta didik tersebut melakukan pembelajaran mandiri atau Self-Directed

Learning (SDL) untuk mempelajari secara khusus bahasa pemrograman yang mereka butuhkan.

Self-directed learning adalah kondisi dimana pembelajar memiliki control sepenuhnya dalam proses tanggung jawab tuhan atasnya, meskipun nantinya mereka membutuhkan nasihat dari seorang guru (Huda, 2013).

Berdasarkan uraian tersebut, peneliti tertarik untuk mengembangkan media pembelajaran untuk melacak aktivitas Self-Directed Learning (SDL) pada pembelajaran bahasa pemrograman Swift. Hasil pengembangan media pembelajaran ini diharapkan dapat menjadi acuan untuk penelitian lain tentang Self-Directed Learning (SDL) lebih mendalam.

Swift dipilih peneliti sebagai materi pembelajaran karena Swift merupakan bahasa pemrograman baru yang dirilis secara resmi pada 2 Juni 2014 dan merupakan satu dari dua bahasa pemrograman yang didukung dalam proses pengembangan aplikasi iOS.

Rumusan masalah yang akan diselesaikan dalam penelitian ini yaitu:

1. Bagaimana mengembangkan media pembelajaran berbasis native iOS untuk melacak aktivitas Self-Directed Learning (SDL) ?
2. Bagaimana cara mengetahui aktivitas Self-Directed Learning (SDL) selama pembelajaran pemrograman?
3. Bagaimana cara mengetahui aktivitas Self-Directed Learning (SDL) dengan menggunakan media pembelajaran yang dikembangkan?

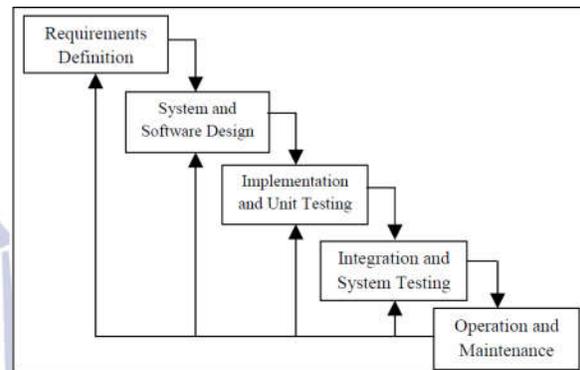
Pelacakan aktivitas yang dilacak meliputi 4 tahap (Huda, 2013) yaitu perencanaan, penerapan, pemantauan dan evaluasi.

Berdasarkan rumusan masalah tersebut penelitian ini bertujuan untuk mengembangkan media pembelajaran berbasis *native* iOS untuk melacak aktivitas Self-Directed Learning, mengetahui aktivitas Self-Directed Learning (SDL) selama pembelajaran pemrograman dan mengetahui aktivitas Self-Directed Learning dengan menggunakan media pembelajaran yang dikembangkan. Penelitian juga mempunyai batasan sebagai berikut :

1. Model pembelajaran yang digunakan yaitu model Self-Directed Learning.
2. Media pembelajaran yang dikembangkan menggunakan bahasa pemrograman Swift.
3. Subjek penelitian merupakan mahasiswa.
4. Materi yang digunakan pada media pembelajaran ini mengacu pada kelas online Dicoding yang dapat diakses secara gratis (<https://www.dicoding.com/academies/145>) yang meliputi materi : a. *The Basic*, b. *Operators* c. *Strings and Character*, d. *Collection Type*, e. *Control Flow*

METODE

Pengembangan media pembelajaran ini menggunakan model waterfall. Gambar 1 merupakan siklus hidup (*life cycle*) dalam model waterfall.



Gambar 1 Siklus hidup model waterfall

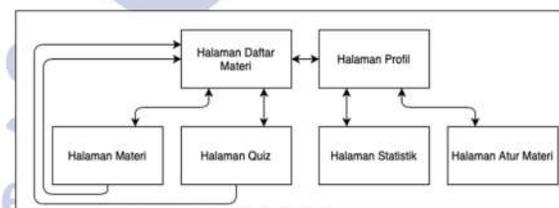
Tahap Pengembangan Model Waterfall

A. Analisis Kebutuhan (*Requirements Definition*)

Analisis yang dilakukan menggunakan metode studi literatur. Literatur yang dibaca dan dianalisis merupakan literatur yang berupa buku dan jurnal internasional tentang Self-Directed Learning. Poin penting yang ditemukan dan mempengaruhi dalam tahap pengembangan berikutnya yaitu peserta didik menentukan tujuan atau materi belajarnya sendiri.

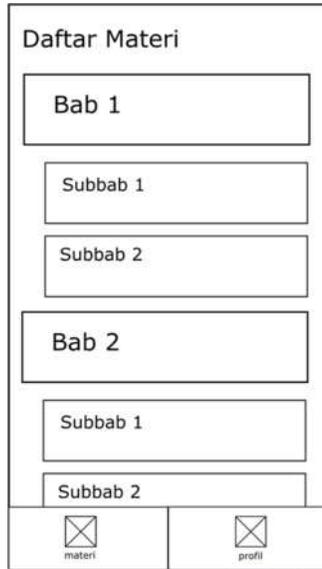
B. Desain (*System and Software Design*)

Desain media pembelajaran ini dibagi menjadi dua bagian, yang pertama yaitu rancangan navigasi yang dapat dilihat pada Gambar 2.



Gambar 2 Rancangan Navigasi

Dan yang kedua adalah rancangan UI dan UX yang terdiri dari rancangan halaman daftar materi beserta algoritma dari masing-masing halaman.



Gambar 3 Rancangan Halaman Daftar Materi

Algoritma yang digunakan pada Gambar 3 yaitu :

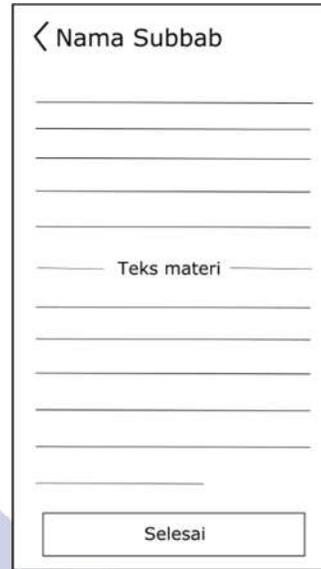
1. Mulai
2. Ambil *list* data bab dari *database*
3. Ambil *list* data subbab dari *database*
4. Tampilkan *list* data ke *UITableView*
5. Selesai



Gambar 4 Rancangan Halaman Profil

Algoritma yang digunakan pada Gambar 4 yaitu :

1. Mulai
2. Tampilkan *UIButton* "Lihat Statistik Belajar"
3. Tampilkan *UIButton* "Atur Materi Belajar"
4. Selesai



Gambar 5 Rancangan Halaman Materi

Algoritma yang digunakan pada Gambar 5 yaitu :

1. Mulai
2. Tentukkan subbab
3. Ambil data teks materi dari *database* berdasarkan subbab
4. Tampilkan teks ke *UILabel*
5. Hitung durasi waktu
6. Simpan durasi waktu ke *database*
7. Buka kunci subbab berikutnya
8. Selesai

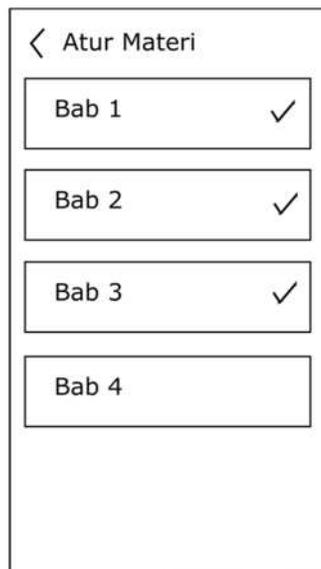


Gambar 6 Rancangan Halaman Kuis

Algoritma yang digunakan pada Gambar 6 yaitu :

1. Mulai
2. Tentukkan subbab

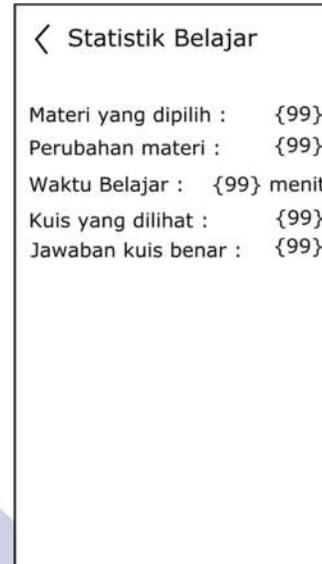
3. Ambil data soal dan pilihan jawaban dari *database* berdasarkan subbab
4. Tampilkan teks ke *UILabel*, *UIButton* A,B,C,D
5. Hitung durasi waktu
6. Hitung jumlah kunjungan
7. Bandingkan jawaban
8. Simpan durasi waktu ke *database*
9. Simpan jumlah kunjungan ke *database*
10. Buka kunci subbab berikutnya
11. Selesai



Gambar 7 Rancangan Halaman Atur Materi

Algoritma yang digunakan pada Gambar 7 yaitu :

1. Mulai
2. Ambil *list* data dari *database*
3. Tampilkan *list* ke *UITableView*
4. Hitung jumlah ubah materi
5. Simpan jumlah ubah materi ke *database*
6. Simpan *list* data ke *database*
7. Selesai

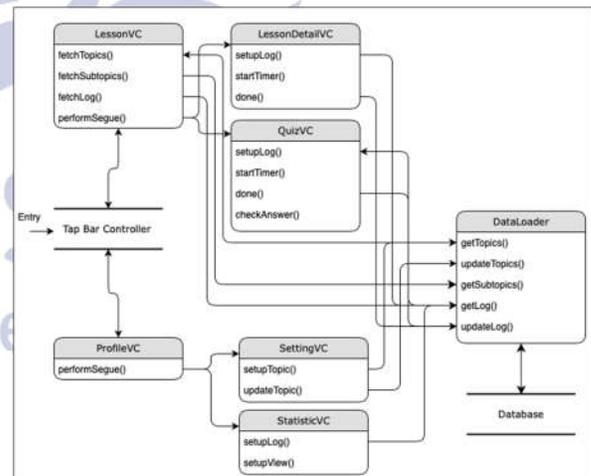


Gambar 8 Rancangan Halaman Statistik

Algoritma yang digunakan pada Gambar 8 yaitu :

1. Mulai
2. Ambil *log* data dari *database*
3. Ubah data waktu belajar dari detik ke menit
4. Tampilkan semua data ke *UILabel*
5. Selesai

Keseluruhan alur algoritma yang diterapkan dapat dilihat pada Gambar 9.



Gambar 9 Alur algoritma yang diterapkan

C. Implementasi (Implementaion and Unit Testing)

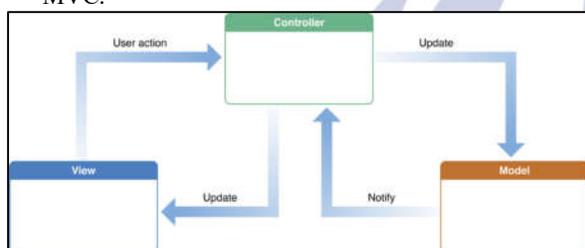
Media pembelajarann ini dikembangkan dengan tools Xcode 12. Bahasa pemrograman yang digunakan dalam pengembangan yaitu Swift 5.3.

Xcode merupakan IDE (*Integrated Development Environment*) yang dikembangkan oleh Apple untuk mengembangkan perangkat lunak untuk macOS, iOS,

iPadOS, WatchOS dan tvOS. Xcode 12 merupakan versi terbaru dan stabil (per 2020).

Sedangkan Swift merupakan bahasa pemrograman yang dikembangkan oleh Apple yang bersifat memiliki tujuan umum, multi-paradigma dan terkompilasi. Pertama kali dirilis pada tahun 2014, Swift dikembangkan sebagai bahasa pemrograman sebelumnya yaitu Objective-C, karena Objective-C Sebagian besar tidak berubah sejak awal 1980-an dan tidak memiliki fitur bahasa modern. Baik Objective-C atau Swift merupakan bahasa utama untuk mengembangkan perangkat lunak di lingkungan Apple. Swift 5.3 sendiri merupakan versi terbaru dan stabil (per 2020).

Setiap halaman yang dibuat pada media pembelajaran yang dikembangkan ini menggunakan algoritma *architecture pattern* MVC (Model-View-Controller). Gambar 10 merupakan alur arsitektur MVC.



Gambar 10 Alur arsitektur MVC

Model merupakan bagian yang mewakili data dari sebuah aplikasi. Dalam media pembelajaran ini rencananya akan menggunakan JSON dan menggunakan *singleton* untuk fungsi pemanggilan dari *Controller*.

View merupakan bagian dari tampilan aplikasi. Sebuah kelas yang dapat digunakan kembali karena ia tidak mengandung domain-specific logic apa pun. Contohnya adalah UILabel, sebuah kelas View yang menyajikan teks di layar iOS dan dapat digunakan kembali atau di-extend. Ketika peserta didik berinteraksi dengan media pembelajaran, mereka akan berinteraksi dengan *view layer*. Sebuah *object view/layer view* tidak mengandung *business logic* sama sekali.

Controller merupakan perantara antara *model* dan *view layer*. Disini semua *business logic* dilakukan mulai dari pengambilan data dari database, menghitung waktu yang dihabiskan dalam melihat layer yang menggunakan *class Timer*.

D. Integrasi (*Integration and System Testing*)

Untuk menyimpan hasil penggunaan media pembelajaran, data diolah secara *asynchronous* sehingga tidak mengganggu proses kegiatan belajar peserta didik.

E. Pemeliharaan (*Operation and Maintenance*)

Pada tahap ini media yang dikembangkan akan diperbaiki ketika ada kendala atau kerusakan.

Selanjutnya media pembelajaran akan dipublikasikan melalui *TestFlight* yang disediakan oleh Apple. Kemudian link media pembelajaran disebarakan secara tertutup kepada subyek penelitian yaitu mahasiswa.

Pengumpulan Data

TestFlight merupakan platform yang disediakan oleh Apple untuk melakukan *beta test* suatu aplikasi. Di dalam *TestFlight* terdapat fitur *feedback* yang mampu menerima umpan balik dari pengguna berupa teks singkat atau *screenshot* dari aplikasi. Melalui fitur ini, aktivitas peserta didik yang ada pada Gambar 8 dikirimkan ke peneliti secara manual kemudian dianalisa.

Data yang terdapat pada tampilan Gambar 8 terdiri dari 4 tahap kategori yaitu perencanaan, penerapan, pemantauan dan evaluasi.

Tahap perencanaan dihitung dari jumlah materi yang dipilih oleh peserta didik dan jumlah pengubahan materi yang sudah dipilih. Tahap penerapan dihitung dari jumlah menjawab pertanyaan dengan benar. Tahap pemantauan dihitung dari waktu peserta membaca materi yang disajikan. Satuan waktu yang digunakan yaitu menit. Sedangkan tahap evaluasi dapat dilihat dari perbandingan waktu maksimal pengerjaan yang dibuat oleh pembuat materi dengan waktu pengerjaan peserta didik.

Materi yang disajikan merupakan materi belajar online Dicoding (<https://www.dicoding.com/academies/145>). Materi ini mempunyai batas pengerjaan waktu maksimal 18 jam 40 menit. Nantinya waktu belajar peserta didik akan dibandingkan dengan waktu target maksimal materi yang disediakan.

HASIL DAN PEMBAHASAN

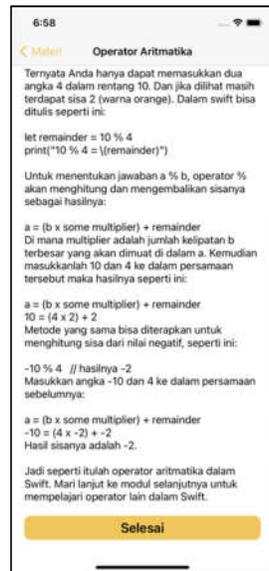
Setelah tiga minggu, media pembelajaran yang dikembangkan sesuai dengan desain yang diharapkan.

Kemudian berikut ini merupakan hasil setiap halaman dari media pembelajaran yang telah dikembangkan :



Gambar 11 Tampilan halaman daftar

Gambar 11 merupakan halaman daftar materi yang memuat daftar bab yang ingin dipelajari dan subbab dari masing-masing materi. *View* pada halaman ini menggunakan *class UITableView* untuk menampilkan subbab yang terdiri dari dua kategori yaitu 'materi' dan 'kuis'. *Controller* pada halaman ini bertugas untuk melakukan pengecekan ke *database* apakah subbab sudah terselamatkan atau belum kemudian *View* akan diberitahu hasil pengecekan tersebut. Subbab yang belum terbuka akan berwarna gelap dan mempunyai ikon gembok.



Gambar 12 Tampilan Halaman Materi

Halaman materi yang dapat dilihat pada Gambar 12 menerapkan *UIScrollView* dan *UIButton* 'Selesai' dibagian paling bawah sehingga peserta didik harus *scrolling* halaman sebelum melanjutkan ke subbab berikutnya. Selain menampilkan data materi dari *database*, *Controller* pada halaman ini juga bertugas melakukan perhitungan waktu belajar menggunakan *class Timer* dari Swift. *Timer* diatur dengan interval 1 detik. Kemudian data disimpan ke *database log* Ketika peserta didik berpindah halaman. Berikut ini merupakan *source code* dari Gambar 12 :

```

1. import UIKit
2.
3. class LessonDetailVC: UIViewController {
4.
5.     var subtopic: Subtopic?
6.     var duration = 0
7.     var viewed = 0
8.     var timer = Timer()
9.
10.    @IBOutlet weak var detailLbl: UILabel!
11.    override func viewDidLoad() {
12.        super.viewDidLoad()
13.
14.        // Do any additional setup after
        loading the view.
15.        setupLog()
16.        setupView()
17.        startTimer()
18.    }

```

```

19.     func setupView() {
20.         detailLbl.text = subtopic?.detail
21.         navigationItem.title =
            subtopic?.name
22.     }
23.
24.     func setupLog(){
25.         let log : Log =
            DataLoader.getLog(subtopicID: subtopic?.id
            ?? 1)
26.         duration = log.duration
27.         viewed = log.viewed ?? 0
28.         viewed += 1
29.     }
30.
31.     func startTimer(){
32.         timer =
            Timer.scheduledTimer(withTimeInterval: 1.0,
            repeats: true) { (Timer) in
33.             self.duration += 1
34.             print(self.duration)
35.         }
36.     }
37.
38.    @IBAction func doneButton(_ sender:
            UIButton) {
39.        DataLoader.updateLog(subtopicId:
            subtopic?.id ?? 1, duration: duration,
            status: SubtopicStatus.cleared, viewed:
            viewed)
40.        navigationController?.popViewController(animated: true)
41.        timer.invalidate()
42.    }
43.
44.
45.    override func viewWillDisappear(_
            animated: Bool) {
46.        let log : Log =
            DataLoader.getLog(subtopicID: subtopic?.id
            ?? 1)
47.        DataLoader.updateLog(subtopicId:
            subtopic?.id ?? 1, duration: duration,
            status: log.status, viewed: viewed)
48.        timer.invalidate()
49.    }
50.
51. }

```

Keterangan :

Baris 5-8 : deklarasi variabel. Variabel *subtopic* berfungsi untuk menampung data subbab yang diterima dari halaman sebelumnya, *duration* digunakan untuk menampung nilai waktu saat melihat halaman, *viewed* digunakan untuk menampung nilai waktu kunjungan dan *timer* untuk menampung obyek dari *class Timer*.

Baris 10 : merupakan variabel referensi *view*.

Baris 11-18 : fungsi *viewDidLoad()* merupakan fungsi dari *UIViewController*. *Code* yang ada didalam fungsi ini akan dijalankan ketika *view* sudah termuat. Disinilah tempat inisialisasi semua fungsi.

Baris 19-22: fungsi *setupView()* bertugas untuk mengubah teks pada *detailLbl* dan *navigation title* (judul halaman) dengan teks yang ada pada variabel *subtopic*.

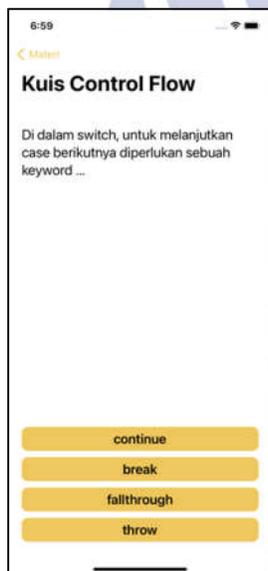
Baris 24-29 : fungsi *setupLog()* bertugas untuk mengambil data *duration* dan *viewed* dari *database*

berdasarkan *subtopic id*. Kemudian nilai *viewed* ditambah 1 karena halaman ini sudah terbuka.

Baris 31-36 : fungsi *startTimer()* bertugas untuk menghitung waktu saat melihat halaman. Caranya dengan memanggil *scheduledTimer* dengan interval 1.0 detik. Kemudian di dalam *closure scheduledTimer*, nilai *duration* ditambah 1.

Baris 38-42 : fungsi *doneButton()* merupakan referensi *action* dari *UIButton*. Fungsi ini bertugas untuk memperbarui data *duration*, *viewed* dan *status* dari *subtopic*. *Status* yang diubah berupa *enum* bernilai *unlocked* menjadi *cleared*. Selain itu fungsi ini bertugas untuk menutup halaman dan menghentikan *timer* dengan memanggil fungsi *invalidate()*.

Baris 45-49 : fungsi *viewWillDisappear* merupakan fungsi dari *UIViewController*. *Code* yang ada didalam fungsi ini akan dijalankan ketika tombol *back* ditekan. Fungsi ini bertugas sama dengan fungsi *doneButton()* namun tidak mengubah *status* dari *subtopic*.



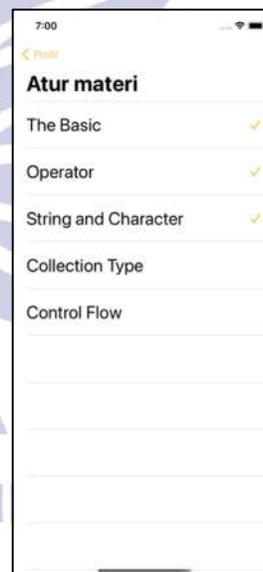
Gambar 13 Tampilan Halaman Kuis

Controller pada halaman materi juga masih menerapkan *Timer* yang dilihat pada Gambar 13. Selain *Timer*, *Controller* pada halaman kuis juga diterapkan perhitungan kunjungan halaman dan disimpan ke *database log*. Hal ini berfungsi untuk melacak pemahaman peserta didik pada materi yang sudah disajikan. Apabila semakin banyak jumlah kunjungan kuis, maka peserta didik semakin tidak paham dengan materi.



Gambar 14 Tampilan halaman profil

Pada Gambar 14 atau halaman profil merupakan halaman paling sederhana, *Controller* disini hanya bertugas untuk navigasi menuju kepada halaman atur materi dan statistik belajar. Tidak ada logika yang signifikan pada halaman ini.



Gambar 15 Tampilan halaman atur materi

Kemudian pada Gambar 15 merupakan halaman atur materi yang berfungsi untuk mengubah materi yang ingin dipelajari oleh peserta didik. *View* pada halaman ini menggunakan *UITableView* dengan *accessoryType checkmark*. *Controller* bertanggungjawab untuk menyimpan pengaturan ke *database*. Setelah mengubah materi dan kembali ke halaman daftar materi pada Gambar 11, *Controller* pada halaman tersebut akan terupdate secara otomatis berdasarkan *database* terbaru. Jumlah pergantian materi ini juga dihitung oleh *Controller*. Perhitungan ini

berfungsi untuk menilai kesiapan dari peserta didik. Apabila terlalu banyak mengubah pilihan materi, menandakan peserta didik tidak siap untuk melakukan pembelajaran mandiri.



Gambar 16 Halaman statistik belajar

Semua perhitungan jumlah materi, jumlah perubahan materi, waktu belajar, jumlah kuis yang dikunjungi dan jumlah kuis yang benar dicatat pada *database* dan kemudian ditampilkan pada Gambar 16 menggunakan *UITextView*. Data-data ini bersifat *persistent offline* yang artinya apabila media pembelajaran dicopot dari perangkat maka data juga menghilang.

Untuk menguji kesesuaian media pembelajaran dengan yang diharapkan uji fungsionalitas dilakukan dan hasilnya dapat dilihat pada Tabel 1 berikut :

Tabel 1 Hasil uji coba fungsionalitas

Fitur	Hasil	Status
Menampilkan daftar materi berdasarkan pilihan	Dapat menampilkan materi kemudian mengelompokkan subbabnya berdasarkan bab masing-masing dan menghilangkan materi yang tidak dipilih dari daftar	Sesuai
Menghitung jumlah perubahan materi	Dapat menghitung jumlah perubahan	Sesuai

	materi pada halaman atur materi	
Menghitung waktu tampilan	Dapat menghitung waktu tampilan dalam satuan detik	Sesuai
Menghitung jumlah kunjungan kuis	Dapat menghitung jumlah kunjungan kuis	Sesuai
Menghitung jumlah kuis yang benar	Dapat menghitung jumlah kuis yang benar	Sesuai
Menampilkan seluruh data aktivitas	Dapat menampilkan seluruh data aktivitas pada halaman statistik belajar	Sesuai

Berdasarkan Tabel 1, algoritma arsitektur MVC dapat digunakan untuk mengembangkan media pembelajaran.

Kemudian dalam kurun waktu dua minggu, peserta didik mengirimkan *screenshot* hasil kegiatan mereka melalui *TestFlight*. Data-data tersebut disajikan pada Tabel 2.

Tabel 2 Data hasil kegiatan belajar

Peserta didik	(a)	(b)	(c)	(d)	(e)
1	5	0	1027	9	5
2	5	0	1087	8	5
3	5	0	948	9	5
4	5	0	873	8	5
5	4	1	943	13	4
6	5	2	965	12	5
7	5	0	844	12	5
8	3	3	654	6	3
9	5	0	811	12	5
10	5	0	1576	15	5
11	5	0	1325	14	5
12	5	2	1444	13	5

Keterangan :

- (a) jumlah materi yang dipilih
- (b) jumlah perubahan materi
- (c) waktu belajar (dalam menit)
- (d) jumlah kuis yang dikunjungi
- (e) jumlah kuis yang benar

Analisis data dapat dilihat dari rata-rata perbandingan tiap variabel. Waktu belajar (c) berbanding lurus dengan jumlah materi yang dipilih (a) dan jumlah kuis yang dikunjungi (d) berbanding lurus dengan jumlah kuis benar (e). Perbandingan ini dapat dilihat pada Tabel 3.

Tabel 3 Tabel perbandingan data terhadap data lain

Peserta didik	Perbandingan (c) terhadap (a)	Perbandingan (d) terhadap (e)
1	205,4	1,8
2	217,4	1,6
3	189,6	1,8
4	174,6	1,6
5	235,75	3,25
6	193	2,4
7	168,8	2,4
8	218	2
9	162,2	2,4
10	315,2	3
11	265	2,8
12	288,8	2,6

Kemudian berikut ini merupakan perhitungan rata-rata tiap data :

$$\text{Rata rata c terhadap a} = \frac{(c/a)/\Sigma(c/a)}{\text{Jumlah data}}$$

$$\text{Rata rata d terhadap e} = \frac{(d/e)/\Sigma(d/e)}{\text{Jumlah data}}$$

Sehingga dapat diperoleh tabel perhitungan pada Tabel 4 seperti berikut :

Tabel 4 Data hasil perhitungan rata-rata

Rata-rata (c) terhadap (a)	Rata-rata (d) terhadap (e)
219,48	2,3

Dalam tahap persiapan, Tabel 2 menunjukkan bahwa peserta 1,2,3,4,7,9,10 dan 11 tidak melakukan perubahan materi. Peserta didik 5 melakukan 1 kali perubahan materi namun hal ini sesuai dengan jumlah materi yang dia ingin pelajari yaitu 4 materi. Sedangkan peserta 6 dan 12 melakukan perubahan materi sebanyak 2 kali sedangkan materi yang mereka pelajari tetap sama yaitu 5 materi hal ini menandakan mereka ragu-ragu untuk menentukan materi yang ingin mereka pelajari. Peserta 8 melakukan perubahan materi sebanyak 3 kali dan materi yang dipilih 3 materi, perubahan materi ini terlalu banyak mengingat fitur yang dikembangkan dapat dilakukan dengan 1 kali perubahan saja. Hal ini menandakan peserta 8 tidak siap menentukan materi belajarnya.

Pada tahap penerapan, Tabel 2 menunjukkan bahwa semua peserta berhasil menyelesaikan kuis yang disediakan. Namun Tabel 3 dan 4 menunjukkan bahwa peserta 5,6,7,9,10,11,12 melebihi rata-rata dari seluruh siswa. Hal ini menunjukkan bahwa peserta tersebut melakukan kunjungan materi kembali lebih banyak daripada peserta lain.

Kemudian dalam tahap pemantauan, waktu membaca materi dapat dilihat pada Tabel 2. Media pembelajaran ini tidak dapat melacak aktivitas diluar aplikasi, seperti membuka internet, email, dll.

Selanjutnya dalam tahap evaluasi, batas waktu maksimal materi yang disediakan adalah 18 jam 40 menit (1220 menit). Dari Tabel 2 dapat dilihat bahwa peserta didik 10, 11 dan 12 melebihi waktu tersebut. Namun jika dilihat dari Tabel 3 dan 4, peserta didik 5, 10, 11 dan 12 yang melebihi rata-rata dari seluruh peserta. Hal ini menunjukkan kemungkinan bahwa batas waktu maksimal yang ditentukan oleh pembuat materi tidak sesuai. Namun untuk membuktikannya diperlukan penelitian lebih lanjut dan data yang lebih banyak.

Uraian data diatas menunjukkan bahwa media pembelajaran yang dikembangkan mampu melacak aktivitas dari peserta didik namun tidak bisa mengetahui penyebab keterlambatan atau ketidaksesuaian hasil peserta didik dengan tujuan pembuat materi. Keterlambatan dan ketidaksesuaian ini bisa terjadi oleh 2 faktor, faktor luar dan dalam. Faktor luar seperti kesibukan peserta, bantuan teman, melihat internet dan lain-lain. Dan juga faktor dari dalam yaitu materi itu sendiri kurang menarik bagi peserta didik sehingga diperlukan penelitian lebih lanjut untuk mengetahui penyebab keterlambatan dan ketidaksesuaian ini. Parameter tambahan juga mungkin diperlukan untuk menambah tingkat kecocokan hasil kegiatan peserta didik dan tujuan dari pembuat materi.

PENUTUP

Kesimpulan

1. Media pembelajaran ini dapat dikembangkan menggunakan metode waterfall dan algoritma arsitektur MVC (Model-View-Controller) yang membagi tugas dan bagian masing-masing. Sehingga pengembangan media sesuai dengan yang diharapkan.
2. Aktivitas *Self-Directed Learning* terdiri dari empat tahap yaitu persiapan, penerapan, pemantauan dan evaluasi. Selama belajar pemrograman peserta didik melakukan persiapan yaitu dengan memilih materi yang ingin dipelajari, melakukan penerapan dengan cara menjawab kuis yang sudah disediakan, melakukan pemantauan dan evaluasi dengan membandingkan waktu belajar dengan batas waktu maksimal belajar.
3. Media pembelajaran yang dikembangkan mampu melacak aktivitas-aktivitas tersebut dari peserta didik namun tidak bisa mengetahui penyebab keterlambatan atau ketidaksesuaian hasil peserta didik dengan tujuan pembuat materi. Keterlambatan dan ketidaksesuaian ini bisa terjadi oleh 2 faktor, faktor luar dan dalam. Faktor luar seperti kesibukan peserta, bantuan teman, melihat internet dan lain-lain. Dan juga faktor dari dalam yaitu materi itu sendiri kurang menarik bagi peserta didik sehingga diperlukan penelitian lebih lanjut untuk mengetahui penyebab keterlambatan dan ketidaksesuaian ini. Parameter tambahan juga mungkin

diperlukan untuk menambah tingkat kecocokan hasil kegiatan peserta didik dan tujuan dari pembuat materi.

Saran

Media pembelajaran yang dikembangkan merupakan sebuah langkah awal dari sebuah pelacakan aktivitas Self-Directed Learning. Media ini bisa digunakan meskipun menggunakan materi yang berbeda meskipun parameter tambahan mungkin dibutuhkan.

DAFTAR PUSTAKA

- Allen, V. J., Bradford, S., Burson, B., Resmer, M. (2005). System and Method for On-line Educational Course Gradebook with Tracking of Student Activity. United States Patent Application Publication. <https://patents.google.com/patent/US20060046239A1/en>
- Badan Pusat Statistik. (2018). Penggunaan dan Pemanfaatan Teknologi Informasi dan Komunikasi (P2TIK) Sektor Pendidikan 2018. <https://www.bps.go.id/publication/2018/12/24/27971845a9d616341333d103/penggunaan-dan-pemanfaatan-teknologi-informasi-dan-komunikasi--p2tik--sektor-pendidikan-2018.html>
- Barlow-Jones, Glenda & Chetty, Jacqui. (2012). Bridging the Gap: the Role of Mediated Transfer for Computer Programming. Conference: 2012 4th International Conference on Education Technology and Computer (ICETC 2012) Volume: IPCSIT vol. 43 (2012) © (2012) IACSIT Press, Singapore.
- Huda, M. (2013). Model-model Pengajaran dan Pembelajaran. Yogyakarta: Pustaka Pelajar.
- Levy, Y. (2007). Comparing dropouts and persistence in e-learning courses. *Computers & Education* Volume 48, Issue 2, February 2007, Pages 185-204. <https://doi.org/10.1016/j.compedu.2004.12.004>
- Michinov, N., Brunot, S., Bohec, O. L., Juhel, J., Delaval, M. (2011). Procrastination, participation, and performance in online learning environments. *Computers & Education* Volume 56, Issue 1, January 2011, Pages 243-252. <https://doi.org/10.1016/j.compedu.2010.07.025>
- Morris, Libby V., Finnegan, C., Wu, S. (2005). Tracking student behavior, persistence, and achievement in online courses. *The Internet and Higher Education* Volume 8, Issue 3, 3rd Quarter 2005, Pages 221-231. <http://doi.org/10.1016/j.iheduc.2005.06.009>
- Rovai, A. P. (2002). Sense of community, perceived cognitive learning, and persistence in asynchronous learning networks, *The Internet and Higher Education* Volume 5, 2002, Pages 313-332. [https://doi.org/10.1016/S1096-7516\(02\)00130-6](https://doi.org/10.1016/S1096-7516(02)00130-6)
- Santos, L. J., Govaerts, S., Verbert, K., Duval, E. (2012). Goal-oriented visualizations of activity tracking: a case study with engineering students. *Proceedings of*

the 2d Internatio Conference on Learning Analytics and Knowledge, April 2012, Pages 143-152. <https://doi.org/10.1145/2330601.2330639>

Yang,, J. C. Quadir, B. Chen, N., Miao, Q. (2016). Effects of online presence on learning performance in a blog-based online course. *The Internet and Higher Education* Volume 30, July 2016, Pages 11-20. <https://doi.org/10.1016/j.iheduc.2016.04.002>

Zhang, H., Almeroth, K., Knight, A., Bulger, M. & Mayer, R. (2007). Moodog: Tracking students' Online Learning Activities. In C. Montgomerie & J. Seale (Eds.), *Proceedings of ED-MEDIA 2007--World Conference on Educational Multimedia, Hypermedia & Telecommunications* (pp. 4415-4422). Vancouver, Canada: Association for the Advancement of Computing in Education (AACE). <https://www.learntechlib.org/primary/p/26016/>