

Extended Finite State Machine-Model Based Testing pada Aplikasi Mobile

Proposal Tugas Akhir

Kelas MK Penulisan Proposal (CCH4A3)

1301180352

Maulidito Dwinandana



**Program Studi Sarjana Informatika
Fakultas Informatika
Universitas Telkom
Bandung
2021**

Lembar Persetujuan

Extended Finite State Machine-Model Based Testing pada Aplikasi Mobile

Extended Finite State Machine-Model Based Testing for Mobile Application

NIM :1301180352

Maulidito Dwinandana

Proposal ini diajukan sebagai usulan pembuatan tugas akhir pada
Program Studi Sarjana Informatika
Fakultas Informatika Universitas Telkom

Bandung, 10 Desember 2021

Menyetujui

Calon Pembimbing 1

Calon Pembimbing 2

Dana Sulistiyo Kusumo, ST., MT., PhD
NIP:02780011

Rosa Reska Riskiana, S.T., M.T.I.
NIP:20930035

ABSTRAK

Tahapan testing merupakan tahapan penting dari proses pengembangan perangkat lunak. Tahapan testing dapat memberikan kualitas yang bagus dari sebuah perangkat lunak. Perkembangan teknologi yang cepat dapat membantu dalam pembelajaran antara guru dan siswa sebagai contoh nya aplikasi kahoot. Aplikasi kahoot merupakan aplikasi pembelajaran berbasis game paling populer yang digunakan dalam aktivitas kelas, Namun siswa memiliki beberapa masalah teknis dalam menggunakan aplikasi kahoot. Dalam tugas akhir ini kahoot akan dilakukan testing menggunakan model based testing dengan pendekatan extended finite state machine.

Kata Kunci: testing, aplikasi, Extended finite state machine, model based testing

1. PENDAHULUAN

1.1. Latar Belakang

Proses pengembangan perangkat lunak terdiri dari beberapa tahapan, antara lain Planning, Analysis, Design, Implementation, Testing & Integration, dan Maintenance. Tahap pengembangan perangkat lunak merupakan hal penting dalam pengembangan perangkat lunak karena untuk memastikan bahwa para stakeholder memahami desain, dan memahami kebutuhan mereka dan juga memastikan proyek berjalan sesuai harapan[1]. Tahapan testing menjadi hal penting, dikarenakan dalam survei dari Pemerintah US ada kerugian bisnis sekitar tahun 2000 sebab perangkat lunak yang telah di publish tidak dilakukan software testing mengakibatkan software memiliki kualitas yang buruk[2]. Untuk meningkatkan kualitas dari sebuah perangkat lunak diperlukan nya tahapan testing.

Testing itu sendiri adalah proses untuk memastikan kode dari perangkat lunak yang dirancang untuk melakukan sesuai yang diharapkan dan tidak melakukan sesuatu yang tidak diharapkan[3]. Testing akan menjadi pekerjaan yang repetitif atau berulang ulang dan memakan waktu jika sistem diperbarui dengan versi terbarunya, contohnya dalam pembuatan test case[4]. Sementara itu aplikasi berbasis mobile berkembang dengan cepat dari aspek teknologinya, fiturnya hingga desain perangkat lunak[4,12]. Dari berbagai macam metode testing, seperti manual testing, automated testing, hingga model based testing metode yang membuat test case secara otomatis ialah model based testing[4]. Metode model based testing telah menghasilkan kinerja yang baik untuk perangkat lunak non-mobile system, untuk penggunaan pada sistem mobile masih samar-samar apakah dapat bekerja dengan baik seperti penggunaan non-mobile system [4].

Perkembangan teknologi aplikasi berbasis mobile dapat diimplementasikan pada berbagai bidang contoh nya dalam ajar-mengajar antara guru dan siswa. Kahoot merupakan aplikasi pembelajaran berbasis game paling populer yang digunakan dalam aktivitas kelas[13]. Dalam penelitian yang dilakukan oleh Alf Inge Wang dan Rabail Tahir dengan jurnal berjudul “The effect of using Kahoot! for learning – A literature review” menyatakan bahwa aplikasi kahoot memberikan dampak positif dalam aktivitas dalam kelas, namun pada penelitian tersebut siswa mengalami masalah teknis pada aplikasi kahoot seperti tidak bisa mengubah

jawaban yang telah dikumpulkan[13]. maka dari pernyataan tersebut penulis akan menggunakan kahoot sebagai studi kasus tugas akhir.

Pendekatan model based testing yang akan digunakan untuk studi kasus ini menggunakan extended finite state machine, karena dalam aplikasi kahoot user memiliki operasi create ,read ,update ,delete pada quiz [5] sehingga membutuhkan pendekatan if statement, maka dari itu EFSM yang memiliki guard dan action dapat mengatasi masalah studi kasus. EFSM juga dapat memodelkan sistem yang memiliki behavioral / perilaku yang kompleks dan juga kombinasi antara perluasan dari statechart yaitu UML statechart dan case diagram dapat di transformasi menjadi EFSM model[6]. maka dari itu penulis akan menganalisis testing menggunakan MBT dengan pendekatan EFSM pada aplikasi kahoot berbasis mobile.

1.2. Perumusan Masalah

Berikut dibawah ini merupakan perumusan masalah dari tugas akhir ini :

1. Bagaimana hasil testing menggunakan MBT dengan pendekatan extended finite state machine pada aplikasi kahoot?

1.3. Batasan Masalah

Berikut dibawah ini merupakan Batasan masalah dari tugas akhir ini :

1. Dalam penelitian ini tidak menerapkan pada device OS ios/apple.
2. Tipe akun yang digunakan pada kahoot hanya student / siswa saja.

1.4. Tujuan

Berikut dibawah ini merupakan tujuan dari tugas akhir ini :

1. Mengetahui proses dan Mengimplementasi model based testing dengan menggunakan metode extended finite state machine pada aplikasi berbasis mobile

1.5. Rencana Kegiatan

Rencana Kegiatan yang akan dilakukan pada pelaksanaan tugas akhir ini adalah sebagai berikut :

1. Identifikasi Masalah

Penulis mencari permasalahan dalam ranah *information science and engineering* yang dapat dijadikan bahan penelitian.

2. Studi Literatur

Penulis mempelajari masalah yang telah didapatkan dengan melakukan pembacaan artikel, jurnal, buku, *paper*, dan literatur terkait permasalahan tersebut.

3. Pembuatan model program

Penulis melakukan pemodelan berdasarkan aplikasi studi kasus atau permasalahan yang didapatkan dari tahap identifikasi masalah

4. Pembuatan Program

Penulis membuat script untuk menyiapkan program otomatis pada aplikasi yang akan dilakukan test

5. Pengujian Kasus Menggunakan Program

Penulis melakukan Pengujian pada program dan model yang dibuat pada aplikasi studi kasus yang dipilih.

6. Penyusunan Laporan Akhir

Penyusunan laporan akhir yang berisi hasil akhir dari penelitian.

1.6. Jadwal Kegiatan

Kegiatan	Bulan					
	1	2	3	4	5	6
Identifikasi masalah						
Studi literatur						
Pembuatan model program						
Pembuatan program						
Pengujian Kasus Menggunakan Program						
Penyusunan laporan akhir						

*Keterangan: shading warna *grayscale*

2. KAJIAN PUSTAKA

2.1. Model Based Testing (MBT)

Model based Testing merupakan teknik testing pada perangkat lunak dimana test case dihasilkan dari sebuah model berdasarkan fungsi - fungsi dari sistem yang sedang di uji, sehingga model merupakan representasi dari sistem yang sedang di uji. ada beberapa 4 pendekatan utama dari model based testing[7] :

- pembuatan data test input dari model domain
- pembuatan test case dari model environment
- pembuatan test case dengan oracles dari perilaku model
- pembuatan test script dari tes abstrak

perbedaan utama model based testing dengan black-box testing dari pembuatan test, black box testing itu masih melakukan penulisan test secara manual berdasarkan dokumentasi software sedangkan model based testing membuat model dengan perilaku yang diharapkan dari sistem yang di uji dan alat dari model based testing akan otomatis membuat test dari model tersebut.

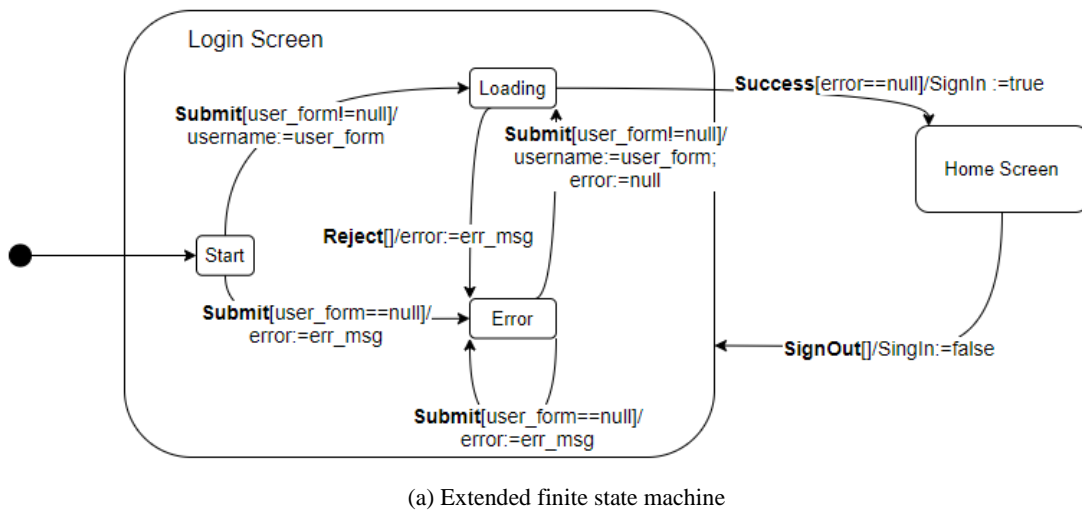
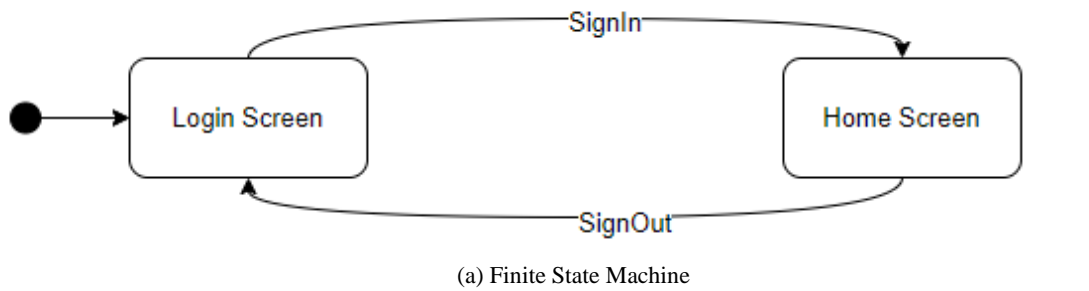
2.2. Extended Finite State Machine (EFSM)

Extended finite state machine merupakan perluasan dari finite state machine, finite state machine sendiri merupakan model komputasi dengan unsur dari state dan transitions. State merupakan keadaan atau posisi dari sebuah sistem contohnya posisi home screen atau posisi setting screen. Untuk transition merupakan perubahan state dengan cara digerakkan, dari state sebelum nya ke state selanjut nya atau bisa juga ke state yang sama. Pada transition menggunakan event sebagai label/penamaan dari transition, event ini yang akan memicu terjadi nya transition[8].

Sistem pada extended finite state machine menyerupai finite state machine, perbedaannya pada transition dimana transition di efsm memiliki syntax [guard condition] / { action } dimana guard / parameter sebagai kondisi if statement yang menentukan untuk melakukan transition jika if statement bernilai true, sedangkan action merupakan suatu output yang bersifat internal untuk model dan output yang dapat dibagi dengan model fsm lain. sehingga pada efsm akan memiliki beberapa

variabel internal.[6]

dicontohkan pada sistem login, terdapat 2 state yang mewakili kondisi pengguna berada di halaman login dan halaman home.



Gambar 2.1 Contoh perbedaan Finite state machine dengan EFSM pada sistem login

Pada gambar 2.1(a) menunjukkan contoh dari finite state machine pada sistem login. Pada state awal pengguna akan diarahkan ke Login Screen dan ketika ada event SignIn maka state akan berpindah dari Login Screen ke Home Screen. Pada Home Screen akan berpindah ke state Login Screen jika event SignOut dipanggil. Sedangkan pada EFSM pada gambar 2.1(b) menunjukkan Login Screen memiliki beberapa state baru yaitu Start, Loading, Error. Pada State awal di state Start pada Login Screen, variabel yang ada seperti username dan error bernilai null untuk SignIn bernilai false. Ketika pada state Start di panggil event Submit maka akan

melakukan pemeriksaan apakah user_form atau input dari user itu sama dengan null atau tidak, jika tidak null maka akan masuk ke state loading dan memasukan user_form ke variabel username yang akan dipakai pada sistem sedangkan jika user_form nya bernilai null maka state akan masuk ke state error dengan variabel error dimasukan err_msg dari sistem. Pada intinya EFSM akan melakukan pengecekan pada guard / parameter jika terdapat event yang sama dengan di dalam simbol tutup kurung , dan pengecekan yang benar akan melakukan action dengan syntax setelah garis miring.

2.3. Graphwalker

Graphwalker merupakan alat yang bersifat open-source / dapat digunakan oleh siapa saja untuk membuat model dalam bentuk grafik dan juga bisa membuat test case dari grafik yang berdasarkan finite state machine. Graphwalker memiliki 2 modul yaitu CLI tool dan GUI version. CLI tool sebagai pembuatan pada grafik pada model dan modul ini dapat bekerja dengan format graphml atau JSON, sedangkan untuk GUI version dapat membuat dan mengedit model hingga model dapat dilakukan verifikasi dengan menjalankan test path generation.[9]

Model yang digunakan terdiri dari 2 elemen yaitu vertices dan edge. Dari 2 elemen, model berdasarkan finite state machine dan elemen tersebut memungkinkan memiliki guard dan action yang mengarah ke extended finite state machine. model dapat menjalankan semua jalan yang berbeda dengan menggunakan modul studio.[9,10]

2.4. Appium

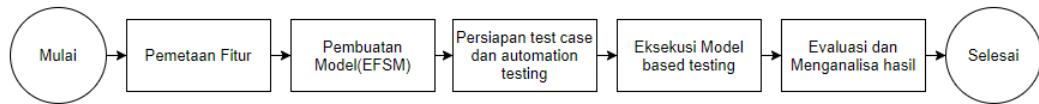
Appium merupakan perangkat lunak open source yang digunakan untuk melakukan otomasi pada website, mobile, dan hybrid application untuk platform Android, IOS, hingga windows. Appium bersifat cross-platform yang berarti appium dapat digunakan pada berbagai macam platform sehingga dapat memungkinkan menulis test untuk semua platform dengan menggunakan API.[11]

Konsep kerja pada perangkat lunak appium ialah menggunakan arsitektur client/server, dimana appium server menggunakan nodejs sebagai server nya. Nodejs akan menerima permintaan menggunakan request api yaitu POST atau GET dan setelah itu dikirimkan request tersebut menggunakan JSON wire protocol ke

perangkat yang digunakan di appium. Driver pada appium dapat dituliskan dengan berbagai macam bahasa pemrograman seperti JavaScript,Java,Python untuk melakukan otomasi pada device yang akan dilakukan test. Device yang dapat dilakukan untuk test menggunakan appium bisa menggunakan emulator atau Handphone asli.[12].

3. Alur Pemodelan

ada beberapa tahap dalam melakukan proses tugas akhir ini, tahap nya akan ditampilkan pada gambar 3.1.



Gambar 3.1 Tahap-tahap Proses penilitan

3.1. Pemetaan Fitur

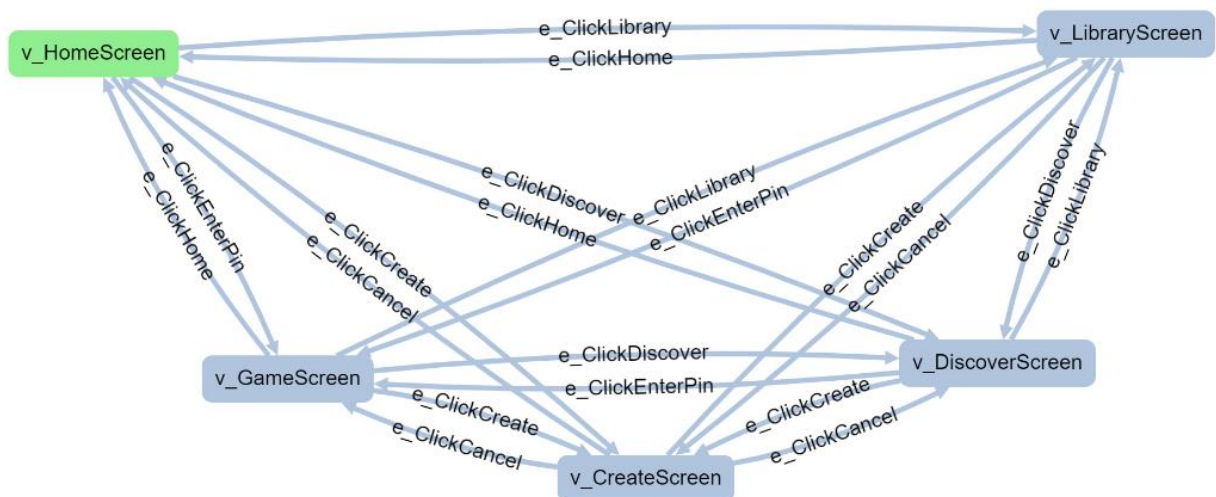
Fitur-Fitur Utama yang ada di aplikasi kahoot dibagi menjadi *screen*/layar, yaitu *home screen*, *discover screen*, *game screen*, *create screen*, dan *library screen*. Contoh fitur akan dijelaskan lebih detail pada tabel 3.1.

No	Feature	Input	Output
1	Home Screen	Menekan tombol Home pada bagian navigasi bawah	Menampilkan bagian home dengan konten yang berisi berbagai macam quiz, quiz milik user
2	Discover Screen	Menekan tombol Discover pada bagian navigasi bawah	Menampilkan semua quiz yang ada dengan dibagi beberapa topik
3	Game Screen	Menekan tombol Enter Pin pada bagian navigasi bawah	Menampilkan bagian game screen yang berisi masukan teks pin dan tombol enter
4	Create Screen	Menekan tombol Create pada bagian navigasi bawah	Menampilkan bagian create screen yang berisi formulir untuk pembuatan quiz
5	Library Screen	Menekan tombol Library pada bagian navigasi bawah	Menampilkan bagian library screen yang berisi laporan pada user

Tabel 3.1 Contoh tabel fitur-fitur pada aplikasi kahoot

3.2. Pembuatan Model (EFSM)

Dalam pembuatan model ada beberapa requirement yang bisa digunakan dengan fsm dan juga ada dengan pendekatan efsm dikarenakan ada requirement yang tidak begitu kompleks dalam proses nya, penulis menggunakan tool graph walker studio. Graph walker studio sudah memiliki fitur dalam pembuatan model dari vertex sebagai state aplikasi, edges sebagai transisi aplikasi, guard sebagai kondisi transisi untuk terpenuhi, dan action sebagai aksi yang dilakukan jika guard/kondisi terpenuhi.



Gambar 3.2 Contoh model layar Fitur Utama pada aplikasi kahoot

Pada gambar 3.1 merupakan contoh pembuatan model dengan pendekatan efsm. penamaan pada model dibedakan menjadi dua yaitu v dan e, v sebagai vertex yang menentukan state sekarang dan e sebagai edge yang melabeli event pada sebuah transitions. pada model tersebut menggunakan pendekatan efsm dengan menambahkan action e_ClickCreate dan guard pada e_ClickCancel, penggunaan guard dan action tersebut dikarenakan pada v_CreateScreen hanya dapat kembali pada state sebelumnya dan tidak bisa ke state lain.

3.3. Persiapan test case dan automation testing

Graphwalker memiliki tool untuk membuat test case abstrak dari sebuah model, yaitu Graphwalker cli. Dalam pembuatan test case abstrak menggunakan algoritma random-path untuk mencakup semua states dan transitions. test case abstrak akan digunakan untuk automation testing yang dilakukan pada appium.

Appium digunakan untuk komunikasi antara aplikasi kahoot. pada appium penulis membuat script yang dapat melacak state atau transition yang baru dan akan

di eksekusi. script dibuat berdasarkan pada requirement dengan nama fungsi-fungsinya mengikuti vertex dan edge pada model.

3.4. Eksekusi Model based Testing

Graphwalker cli memiliki tool eksekusi yang dapat digunakan pada appium. perangkat yang digunakan pada tahap eksekusi untuk aplikasi kahoot yaitu emulator genymotion. Eksekusi dilakukan dengan cara hasil dari test case abstrak yang telah dibuat akan dijalankan dan digunakan pada appium, dan appium akan memerintahkan emulator sesuai dengan perintah test case.

DAFTAR PUSTAKA

- [1] Scroggins, R. (2014). SDLC and development methodologies. *Global Journal of Computer Science and Technology*.
- [2] McLeod, R. (2007). *Software Testing: Testing Across the Entire Software Development Life Cycle*. Wiley Interscience.
- [3] Myers, G. J., Sandler, C., & Badgett, T. (2011). *The art of software testing*. John Wiley & Sons.
- [4] Gudmundsson, V., Lindvall, M., Aceto, L., Bergthorsson, J., & Ganesan, D. (2016). Model-based Testing of Mobile Systems--An Empirical Study on QuizUp Android App. *arXiv preprint arXiv:1606.00503*.
- [5] Licorish, S. A., Owen, H. E., Daniel, B., & George, J. L. (2018). Students' perception of Kahoot!'s influence on teaching and learning. *Research and Practice in Technology Enhanced Learning*, 13(1), 1-23.
- [6] Alagar, V. S., & Periyasamy, K. (2011). Extended finite state machine. In *Specification of software systems* (pp. 105-128). Springer, London.
- [7] Utting, M., & Legeard, B. (2010). *Practical model-based testing: a tools approach*. Elsevier.
- [8] Ulyantsev, V., & Tsarev, F. (2011). Extended finite-state machine induction using SAT-solver. *2011 10th International Conference on Machine Learning and Applications and Workshops*.
- [9] Korhonen, J. (2020). Automated Model Generation using Graphwalker Based On Given-When-Then Specifications.
- [10] Karl, K., & Olsson, N. (2014, April 27). Graphwalker/graphwalker-project. GitHub. Retrieved November 29, 2021, from <https://github.com/GraphWalker/graphwalker-project/wiki>.
- [11] appium.io, "Appium," [Online]. Available: <http://appium.io/>
- [12] Gabriel K. (2017) MODEL BASED TESTING MOBILE APPLICATIONS: A CASE STUDY OF MOODLE MOBILE APPLICATION.
- [13] Wang, A. I., & Tahir, R. (2020). The effect of using Kahoot! for learning – A literature review. *Computers & Education*, 149, 103818. <https://doi.org/10.1016/j.compedu.2020.103818>.