Model Formal dan Verifikasi Sistem Layanan Presensi RFID dengan Logika Temporal: Studi Kasus di Universitas Telkom, Indonesia

Formal Model and Verification for RFID Presence Service Using Temporal Logic: A Case Study in Telkom University, Indonesia

Damar Khoirul Huda[1]
[1]Prodi S1 Teknik Informatika, Fakultas Teknik, Universitas Telkom
[1]mie.yaminasin@gmail.com

Abstrak

Sistem presensi RFID (Radio Frequency Identification) di Universitas Telkom adalah suatu sistem layanan yang bertujuan untuk meningkatkan efisiensi dalam proses pencatatan kehadiran mahasiswa. Saat ini masalah masih banyak ditemukan sehingga keandalan sistem layanan presensi masih belum optimal. Salah satu masalahnya adalah presensi mahasiswa terkadang tidak terekam pada sistem tersebut.

Dalam tugas akhir ini, penulis terlebih dulu membuat dokumen tertulis (diagram aktivitas) dan spesifikasi sistem secara rinci dan jelas berdasarkan hasil observasi di lapangan. Hasil tersebut akan diformalisasikan ke dalam model formal dengan logika temporal tertentu. Selanjutnya penulis menggunakan metode formal untuk melakukan verifikasi spesifikasi keamanan (safety) dan keterjangkauan (liveness) berdasarkan model yang dibentuk.

Tugas akhir ini memberikan suatu contoh translasi dari diagram aktivitas ke dalam model formal untuk sistem layanan presensi RFID. Lebih jauh, penulis menunjukkan bahwa metode formal dapat mendukung verifikasi sistem layanan presensi RFID di Universitas Telkom.

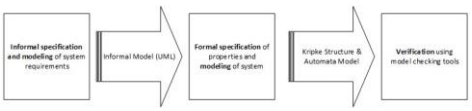
Kata Kunci: metode formal, RFID, diagram aktivitas, model formal

1 Pendahuluan

Kehadiran sistem layanan presensi RFID bagi Universitas Telkom adalah upaya untuk memberikan kemudahan dan meningkatkan efisiensi dalam proses presensi kehadiran mahasiswa. Meskipun layanan presensi RFID cukup menjanjikan, masalah masih dapat muncul ketika risiko tak terduga terjadi pada tahap implementasinya. Salah satu cara untuk mengurangi terjadinya risiko dalam tahap implementasi adalah menggunakan metode formal.

Metode Formal adalah suatu teknik dan alat yang berbasis logika matematika untuk spesifikasi, perancangan, dan verifikasi perangkat lunak atau perangkat keras. Dengan metode formal, spesifikasi dan model dari sebuah sistem akan tidak ambigu, konsisten dan lengkap [1]. Model checking adalah sebuah cara dalam metode formal untuk memverifikasi sistem dengan cara memodelkan FSM (Finite State Machine atau struktur kripke) dan mentranslasikan spesifikasi ke dalam formula logika tertentu. Model checking dapat menghasilkan counterexample untuk mengidentifikasi kesalahan yang mungkin tak terlihat pada spesifikasi maupun model tersebut [2].

Model checking untuk memverifikasi beberapa properti dari spesifikasi kebutuhan sebelumnya telah dijelaskan di [3]. Oleh karena itu pada tugas akhir ini, penulis akan memodelkan secara formal sistem layanan presensi RFID di Universitas Telkom dan melakukan verifikasi formal spesifikasi keamanan (safety) dan keterjangkauan (liveness/progress) pada model yang dibentuk. Untuk mencapai tujuan maka berikut alur penelitian yang dilakukan oleh penulis, Gambar 1 menunjukkan metode yang digunakan dalam menyelesaikan tugas akhir ini (diadaptasi dari[3]). Proses yang lebih jelas dapat dilihat pada Gambar 2.



Gambar 1: Tahapan pada Formalisasi dan Model Checking

Gambar 2: Detail dari Tahapan pada Formalisasi dan Model Checking

2 Kajian Pustaka

2.1 Metode Formal

Metode formal merupakan kumpulan notasi dan alat untuk menspesifikasikan persyaratan yang jelas dari sistem komputer dengan sebuah semantik formal yang digunakan untuk mendukung pembuktian dari properti yang ditinjau [4].

Setidaknya terdapat dua jenis metode verifikasi yang umum digunakan, yakni theorem proving dan model checking. Theorem proving dapat digunakan untuk melakukan verifikasi pada sistem dengan state tak berhingga banyaknya, namun harus dilakukan secara manual atau semi-otomatis oleh penguji dengan cara melakukan deduksi yang ketat (rigor). Model checking dapat dilakukan secara otomatis menggunakan model checker tools dan tidak memerlukan pengawasan, namun hanya dapat menangani sistem dengan state berhingga.

2.2 LTL (Logika Temporal Linear)

Logika temporal linier adalah suatu kerangka logika yang dapat dipakai untuk memodelkan suatu sistem dengan model waktu-linier dan bersifat diskrit. Salah satu sistem sederhana yang dapat dimodelkan dengan LTL adalah lampu lalu lintas jalan. Pada LTL, waktu dimodelkan secara diskrit dan linier (ilustrasi transisi ada pada Gambar 3).

Gambar 3: Ilustrasi Pemodelan LTL

2.2.1 Sintaks LTL

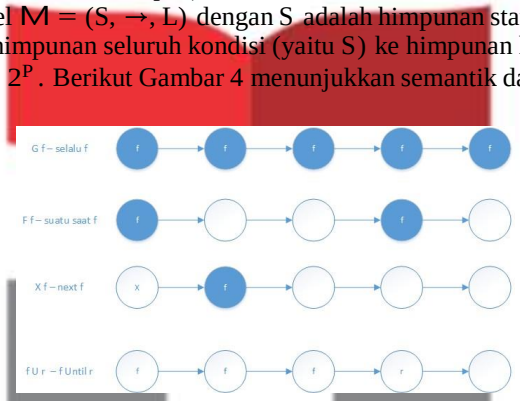
Berikut sintaks Backus Normal Form (BNF) untuk logika temporal linier, misalkan $P = \{p \mid p$ proposisi atom$\}$ menyatakan himpunan seluruh proposisi atom yang ditinjau dan $p \in P$. Formula φ didefinisikan dengan BNF berikut:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \rightarrow \varphi$$

$$| X\varphi | F\varphi | G\varphi | \varphi U \varphi | \varphi W \varphi | \varphi R \varphi$$

2.2.2 Semantik LTL

Semantik formula LTL ditinjau pada sebuah model formal yang juga disebut sebagai sistem transisi (beberapa literatur juga menyebutnya dengan struktur Kripke). Sebuah model atau sistem transisi untuk LTL dengan himpunan proposisi atom P adalah tripel $M = (S, \rightarrow, L)$ dengan S adalah himpunan state, $\rightarrow$ adalah relasi biner pada S, L adalah fungsi pelabelan dari himpunan seluruh kondisi (yaitu S) ke himpunan kuasa proposisi atom (yaitu 2^P), L dapat ditulis sebagai $L : S \rightarrow 2^P$. Berikut Gambar 4 menunjukkan semantik dari LTL.



Gambar 4: Ilustrasi Semantik LTL Pada Lintasan

2.3 PLTL (Logika Temporal Linear-Lampau)

Sejauh ini operator temporal pada LTL X, F, G, dan U menyatakan "masa depan". Terkadang kita ingin spesifikasi untuk menyatakan "masa lampau", seperti "dalam kondisi apapun ketika φ terjadi maka pasti ada beberapa ψ terjadi sebelumnya (di masa lampau)". Untuk menyatakan spesifikasi tersebut maka dibutuhkan operator O. Operator O merepresentasikan once dan analog dari operator temporal F (future). Tugas akhir ini hanya meninjau operator O saja. Pada spesifikasi "dalam kondisi apapun ketika φ terjadi maka pasti ada beberapa ψ terjadi sebelumnya (di masa lampau)" dapat kita translasikan menjadi $G(\varphi \rightarrow O\psi)$. Formula PLTL $G(\varphi \rightarrow O\psi)$ bisa direpresentasikan ke dalam formula LTL $\neg((\neg\psi)U(\varphi \wedge \neg\psi))$.

2.4 Model Checker NuSMV

NuSMV (New Symbolic Model Verifier) merupakan sebuah program sistem model checking yang bersifat open source [5, 6, 7]. Program NuSMV sendiri terdiri dari beberapa tipe deklarasi dari variabel sistem, pemberian nilai (assignments) yang mendefinisikan valid initial states (contoh $\mathbf{init}\ (b0) := 0$). Pemberian nilai yang mendefinisikan relasi perpindahan (contoh $\mathbf{next}\ (b0) := !b0$).

2.5 Diagram Aktivitas

Tujuan dari diagram aktivitas adalah untuk memodelkan arus prosedural dari tindakan (actions) yang merupakan bagian dari aktivitas yang lebih besar [8]. Diagram aktivitas dapat memodelkan spesifik use case pada tingkat yang lebih rinci. Diagram aktivitas juga dapat digunakan untuk memodelkan fungsi pada tingkatan sebuah sistem. Karena memodelkan arus prosedural, diagram aktivitas berfokus pada urutan tindakan (action sequence) dari eksekusi dan kondisi (guard) yang memicu pada tindakan (actions) tersebut.

2.6 Aturan Translasi dari Diagram Aktivitas ke NuSMV

Dalam translasi diagram aktivitas ke model formal dengan bahasa SMV, ada beberapa aturan yang harus dilakukan serta disebutkan pula batasan teknis pada translasi yang diberikan [9]. Batasan teknis terletak pada tidak bolehnya terdapat dua pseudo nodes (P N) yang saling terhubung secara langsung.

Beberapa aturan dalam mentranslasikan P N bergantung kepada successors maupun predecossors-nya untuk menentukan tindakan selanjutnya. Diagram aktivitas yang akan ditranslasikan ke dalam NuSMV tidak boleh mimiliki AN dengan lebih dari satu edge ke dalam dan lebih dari satu edge keluar (hyperedge). Jika terdapat hyperedge pada diagram aktivitas maka hyperedge tersebut harus dihilangkan dengan cara menggantinya ke dalam $P\ N^{fork,join}$.

Setelah memastikan tidak terdapat hyperedge pada diagram aktivitas dan tidak terdapat P N yang terhubung secara langsung maka translasi diagram aktivitas ke model formal dengan bahasa SMV dapat dilakukan dengan sembilan aturan yang diberikan oleh [9].

3 Konstruksi Model Formal

Pada bab ini akan dilakukan pembahasan mengenai konstruksi model formal yang berisikan data dan spesifikasi informal, diagram aktivitas yang dibentuk, serta translasi model formal dan spesifikasi formal dengan LTL dan PLTL.

3.1 Data dan Spesifikasi Informal

Model formal yang dibuat merupakan model dari sistem layanan presensi RFID di Universitas Telkom. Model formal tersebut berasal dari deskripsi oral (hasil wawancara) dan spesifikasi kebutuhan informal yang terdapat pada sistem layanan presensi RFID di Universitas Telkom. Data diambil dari hasil wawancara dengan narasumber, pencarian data secara langsung di lapangan, dan beberapa modifikasi didalamnya yang tidak mengubah langkah kerja sistem.

3.1.1 Data yang Didapatkan dari Wawancara dan Observasi

Berikut ini adalah beberapa contoh data yang penulis dapatkan dari hasil wawancara dan pencarian secara langsung.

1. Reader berfungsi untuk membaca data hexa dari kartu mahasiswa, mengirim data hexa tersebut ke middleware, dan menerima balasan dari middleware dengan waktu yang sudah ditentukan, yakni kurang dari 700ms.
2. Middleware berfungsi sebagai pengubah format hexa menjadi ASCII dan perantara antara reader dengan database.
3. Database RFID menyimpan presensi kehadiran mahasiswa seperti nim, nik dosen, kode mata kuliah, tanggal/waktu, dan juga database RFID menyimpan informasi milik reader itu sendiri seperti alamat IP, port, dll.

3.1.2 Spesifikasi Informal

Pada tahap pertama formalisasi dan model checking (selanjutnya kita sebut sebagai FMC) yang terdapat pada Gambar 1, spesifikasi kebutuhan informal dari sistem layanan presensi RFID di Universitas Telkom harus diidentifikasi terlebih dahulu. Spesifikasi kebutuhan informal tersebut pada tahap selanjutnya akan digunakan sebagai dasar untuk membangun model informal (diagram aktivitas) dan sifat-sifat yang harus dipenuhi olehnya. Berikut beberapa contoh kebutuhan spesifikasi informal yang penulis identifikasi berdasarkan dari data yang terdapat di lapangan.

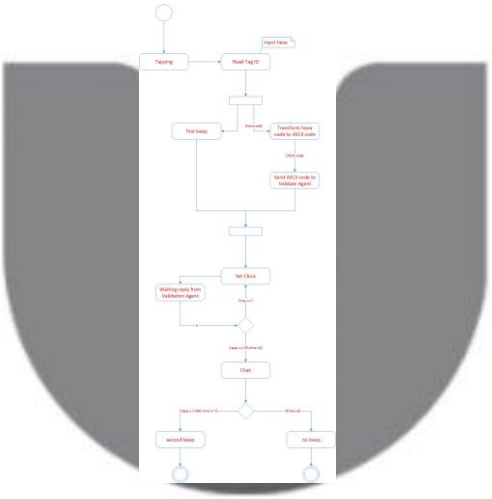
1. Dalam kondisi apapun ketika proses tap dilakukan maka aktivitas read_tag suatu ketika dapat memperoleh masukan berupa nomor unik (UID) dengan format hexa.
2. Dalam kondisi apapun reader tidak akan memberikan bunyi beep pertama (beep_1) jika aktivitas proses tapping tidak dilakukan.
3. Dalam kondisi apapun ketika aktivitas read_tag sukses maka reader akan mengeluarkan bunyi beep pertama dan data UID (ASCII) akan dikirim ke database RFID (agen validasi).

3.2 Diagram Aktivitas untuk Sistem Layanan Presensi RFID

Spesifikasi kebutuhan informal yang sudah teridentifikasi akan digunakan sebagai dasar dalam membangun model informal yang lebih rinci (UML diagram aktivitas). Berdasarkan [3] UML diagram aktivitas akan dipisah perbagian berdasarkan kegunaannya masing-masing dan [3] menyebutnya sebagai agen. Oleh karena itu penulis akan memisahkan diagram aktivitas menjadi dua agen, yakni diagram aktivitas agen reader dan agen validasi. Pada diagram aktivitas agen validasi akan dibagi menjadi tiga bagian, yakni agen validasi untuk pemeriksaan pengguna, agen validasi untuk dosen, dan agen validasi untuk mahasiswa[1].

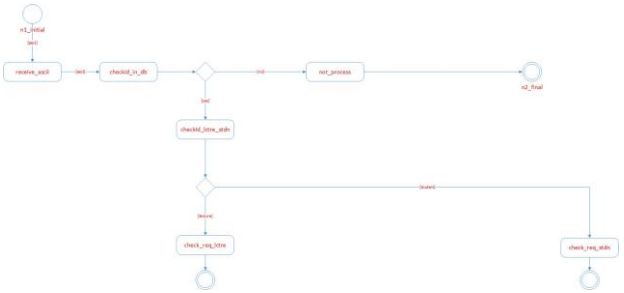
[1] Pemisahan diagram aktivitas pada agen validasi juga bertujuan untuk memperkecil model formal NuSMV yang terbentuk, sehingga proses model checking tidak terjadi state exploison.

3.2.1 Diagram Aktivitas Agen Reader



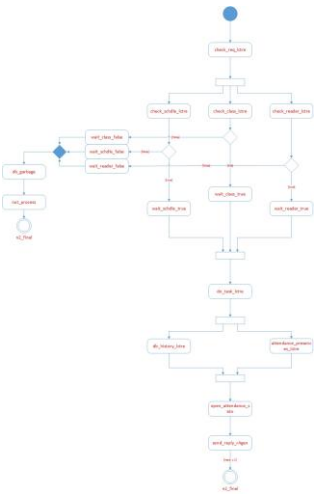
Gambar 5: Diagram Aktivitas Agen Reader

3.2.2 Diagram Aktivitas Agen Validasi untuk Pemeriksaan Pengguna



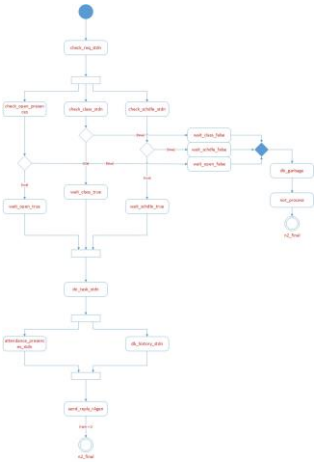
Gambar 6: Diagram Aktivitas Agen Validasi untuk Pemeriksaan Pengguna

3.2.3 Diagram Aktivitas Agen Validasi untuk Dosen



Gambar 7: Diagram Aktivitas Agen Validasi untuk Dosen

3.2.4 Diagram Aktivitas Agen Validasi untuk Mahasiswa



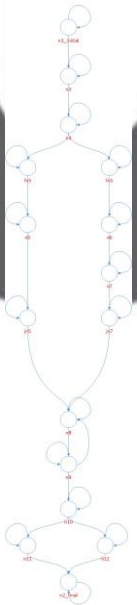
Gambar 8: Diagram Aktivitas Agen Validasi untuk Mahasiswa

3.3 Translasi dari Diagram Aktivitas ke NuSMV

Pada subbab ini penulis secara formal mentranslasikan diagram aktivitas agen reader, agen validasi untuk pemeriksaan pengguna, agen validasi untuk dosen, dan agen validasi untuk mahasiswa menggunakan bahasa SMV dengan merujuk pada penelitian yang dilakukan oleh Shahar, Oliver, dan Bernhard [9]. Pada penelitan [9] dijelaskan bagaimana langkah-langkah dalam mentranslasikan diagram aktivitas menjadi sebuah model yang formal dalam bahasa SMV dengan sembilan aturan yang ditetapkan. Sebagai contoh penulis hanya mengambil dari agen reader saja.

3.3.1 Kode NuSMV Agen Reader

Salah satu contoh hasil penerapan aturan 5 terhadap diagram aktivitas agen reader. Aturan 5 melakukan transisi TRANS terhadap semua edge yang terdapat pada setiap node dan transisi ke diri sendiri. Berikut Gambar 9 sebagai ilustrasi transisi aturan TRANS 5 terhadap diagram aktivitas agen reader.



Gambar 9: Ilustrasi TRANS Aturan 5 pada Diagram Aktivitas Agen Reader

3.4 Translasi Spesifikasi Informal ke Logika Temporal

Pada bagian ini, kita akan membahas spesifikasi yang dapat ditinjau pada agen reader dan agen validasi. Beberapa spesifikasi diperoleh dari penjelasan pada Subbab 3.1.2 dan beberapa spesifikasi diperoleh dari observasi visual terhadap diagram aktivitas yang terbentuk.

Spesifikasi keamanan (safety) menyatakan bahwa hal yang buruk tidak boleh pernah terjadi, sedangkan spesifikasi keterjangkauan (liveness) menyatakan bahwa sesuatu hal yang baik harus dapat terjadi. Secara default, spesifikasi akan ditranslasikan ke formula LTL yang bersesuaian [10]. Seandainya spesifikasi tidak dapat dinyatakan dalam LTL maka translasi spesifikasi ke PLTL atau CTL akan dilakukan.

3.4.1 Formula Logika Temporal

Berikut salah satu contoh translasi spesifikasi informal ke spesifikasi formal.

1. Dalam kondisi apapun ketika proses tap dilakukan maka aktivitas read tag suatu ketika dapat memperoleh masukan berupa nomor unik (UID) dengan format hexa.

$$G((ac = tapping) \rightarrow F((ac = read\,tag\ id) \wedge (uid = hexa)))$$

2. Dalam kondisi apapun reader tidak akan memberikan bunyi beep pertama (beep 1) jika aktivitas proses tapping tidak dilakukan.

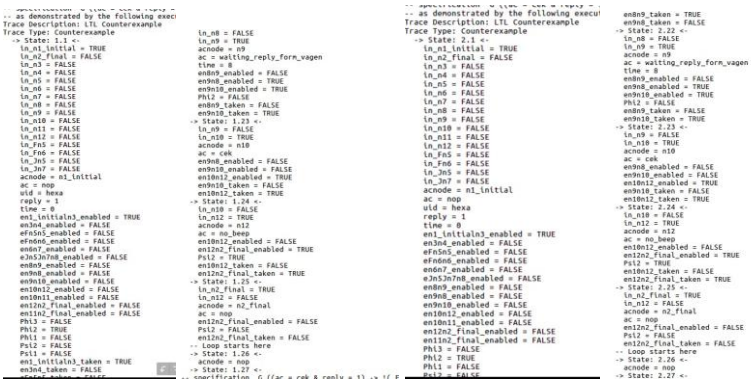
$$G((ac = beep) \rightarrow O(ac = tapping))$$

3. Dalam kondisi apapun ketika aktivitas read tag sukses maka reader akan mengeluarkan bunyi beep pertama dan data UID (ASCII) akan dikirim ke database RFID (agen validasi).

$$G((ac = read\,tag\ id) \rightarrow F(ac = beep) \wedge F((ac = send\ to\ vagen) \wedge (uid = ascii)))$$

4 Verifikasi Formal Model Sistem dan Analisis

Verifikasi dengan model checker NuSMV terhadap spesifikasi LTL, PLTL, CTL, dan model formal SMV agen reader dan agen validasi yang sudah dibentuk pada Subbab 3.3 hingga 3.4. Hasil yang didapatkan adalah semua spesifikasi bernilai **true**, kecuali spesifikasi 11 dan 12.



Gambar 10: Counterexample Agen Reader terhadap Spesifikasi 11 dan 12

5 Kesimpulan

Berikut adalah kesimpulan yang didapat dari kegiatan penelitian yang telah dilakukan:

1. Berdasarkan konstruksi model formal yang telah dilakukan pada pengerjaan tugas akhir ini, dapat disimpulkan bahwa diagram aktivitas pada sistem layanan presensi RFID di Universitas Telkom dapat ditranslasikan menjadi model formal.

2. Berdasarkan verifikasi dan analisis yang telah dilakukan pada pengerjaan tugas akhir ini, terdapat dua spesifikasi yang tidak dipenuhi dalam sistem layanan presensi RFID di Universitas Telkom, yakni Spesifikasi 11 dan 12. Ini berarti pada model yang dibentuk terdapat kondisi ketika reader mendapatkan balasan dari server namun reader tidak membunyikan beep kedua (beep_2) dan masuk ke kondisi (no_beep).

6 Saran

Berikut adalah saran-saran yang dapat disampaikan untuk penelitian selanjutnya:

1. Pada penelitian selanjutnya pendefinisian informal sistem layanan presensi RFID di Universitas Telkom tidak sebatas pada high level abstraction, melainkan juga pada low level abstraction, contohnya protokol RFID.
2. Pada penelitian selanjutnya, pembentukan model formal tidak hanya didasarkan pada diagram aktivitas saja, tetapi juga dapat menggunakan message sequence chart serta kerangka pemodelan formal lain (contohnya timed automaton).

Pustaka

[1] R. W. Butler, "What is formal methods," https://shemesh.larc.nasa.gov/fm/fm-what.html, Aug. 2001, online: accessed 20-May-2017.

[2] E. M. Clarke, O. Grumberg, and D. Peled, Model checking. MIT press, 1999.

[3] N. A. Bakar and A. Selamat, "Analyzing model checking approach for multi agent system verification," in Software Engineering (MySEC), 2011 5th Malaysian Conference in. IEEE, 2011, pp. 95–100.

[4] J. P. Bowen and M. G. Hinchey, Applications of formal methods. Prentice Hall PTR, 1995.

[5] M. Huth and M. Ryan, Logic in Computer Science: Modelling and reasoning about systems. Cambridge University Press, 2004.

[6] M. Fisher, An introduction to practical formal methods using temporal logic. John Wiley & Sons, 2011.

[7] M. Ben-Ari, Mathematical logic for computer science. Springer Science & Business Media, 2012.

[8] D. Bell, "Uml basics part ii: The activity diagram," IBM Global Services, Rational Software, 2003.

[9] S. Maoz, J. O. Ringert, and B. Rumpe, "An operational semantics for activity diagrams using smv," arXiv preprint arXiv:1409.2356, 2014.

[10] R. Eshuis, "Symbolic model checking of uml activity diagrams," ACM Transactions on Software Engineering and Methodology (TOSEM), vol. 15, no. 1, pp. 1–38, 2006.