Pemodelan Integration Use Case (IUC): Perancangan Use Case Diagram (UML) untuk Sistem-sistem yang Terintegrasi

Tita Ayu Rospricilia[1], Mochamad Nizar Palefi Ma'ady[2]
[1,2]Jurusan Sistem Informasi, Fakultas Rekayasa Industri, Telkom University
Email: [1]titaayu@telkomuniversity.ac.id, [2]mnizarpm@telkomnuniversity.ac.id

Abstract. *Use case modeling in developing software for a system has become an old literature. Digital business competition is becoming faster and more integrated. Based on the observation results, the problem that arises is that the development of a system or software not only analyzes the development of its own system but also considers how the system can be integrated with other systems or features in other company applications. Therefore, a solution is provided for designing use-case diagrams for integrated systems. However, based on the authors' observations, there is still very little literature discussing this. This article aims to identify and explain integration use case (IUC) modeling through discussions of case studies of school management information systems from various sources, as well as educating readers about the importance of IUC based on the use of good and correct UML notation. The methodology used in this study includes problem identification, data collection, literature study, analysis of existing systems, development of integrated use cases, and drawing conclusions. The results of this article are a detailed discussion of the basic notations used to describe use case diagrams, making integration use case diagrams (IUCD) for systems integrated with existing systems or new systems. The type of validation data collected were quantitative data in the form of a questionnaire that produced a percentage of 88.61% with a valid category.*
Keywords: *Object Oriented, Unified Modeling Language, System Integration, Integration Use Case, School Information System Management*

Abstrak. *Pemodelan use case dalam mengembangkan perangkat lunak untuk sebuah sistem sudah menjadi literatur lama. Dewasa ini, persaingan bisnis digital semakin cepat dan serba terintegrasi. Berdasarkan hasil observasi, permasalahan yang muncul ialah pembangunan sistem atau perangkat lunak kini tidak hanya menganalisis pengembangan sistemnya sendiri, namun juga mempertimbangkan bagaimana sistem nanti dapat terintegrasi dengan sistem lain atau fitur pada aplikasi perusahaan lain. Oleh karena itu, diberikan solusi perancangan use case diagram untuk sistem-sistem yang terintegrasi. Namun berdasarkan pengamatan penulis, literatur yang membahas hal teserbut masih sangat sedikit. Artikel ini bertujuan untuk mengidentifikasi dan menerangkan pemodelan integration use case (IUC) melalui pembahasan di dalam studi kasus sistem informasi manajemen sekolah dari berbagai sumber, sekaligus sebagai edukasi kepada pembaca akan pentingnya IUC berdasarkan penggunaan notasi UML yang baik dan benar. Metodologi yang digunakan dalam penelitian ini adalah identifikasi masalah, pengumpulan data, studi literatur, analisis system eksisting, pengembangan integrated use case, dan penarikan kesimpulan. Hasil dari artikel ini adalah pembahasan secara detail terkait dasar notasi-notasi dalam penggambaran use case diagram, pembuatan integration use case diagram (IUCD) untuk sistem yang terintegrasi dengan sistem eksisting atau sistem baru. Jenis data validasi yang dikumpulkan berupa data kuantitatif berupa angket yang menghasilkan presentase 88,61% dengan kategori valid.*
Kata Kunci: *Orientasi Objek, Unified Modeling Language, Integrasi Sistem, Use Case Integrasi, Sistem Informasi Manajemen Sekolah*

1. Pendahuluan

Persaingan bisnis digital yang masif menghasilkan produk-produk digital yang serba terintegrasi. Suatu sistem atau aplikasi diharapkan (*requirements*) memiliki kemampuan untuk terhubung dengan aplikasi lain (Idris & Delvika, 2014). Misalkan, kebutuhan fungsional sistem yang dibangun diharapkan mampu melakukan pembayaran melalui *e-wallet*. Maka, sistem tersebut cukup terhubung ke sistem dari perusahaan lain yang memang berfokus pada pengembangan pelayanan pembayaran *e-wallet*. Karena memang sistem pembayarannya sudah matang dan telah teruji

sebelumnya. Sedangkan, bila mengembangkan sistem pembayaran sendiri, maka prosesnya harus melalui fase yang lama (*design - development*) dan membutuhkan berbagai pengujian (*testing*) sebelum benar-benar siap *released* (Sugeng Pranoto, 2024).

Saat ini perkembangan teknologi memberikan kemudahan, istilah "sistem informasi terintegrasi" mengacu pada platform teknologi yang memungkinkan organisasi untuk mengintegrasikan dan mengoordinasikan proses bisnis mereka (Idris & Delvika, 2014). Dengan kata lain, ambang batas integrasi yang cukup tinggi untuk mengakomodasi kebutuhan data dan informasi yang tidak konsisten (Idris & Delvika, 2014). Salah satu diantaranya adalah bidang *financial technology, fintech* merupakan inovasi dalam teknologi keuangan, memiliki kemampuan untuk mengubah model bisnis layanan keuangan melalui berbagai fitur yang terintegrasi ke dalam sistem informasi (Mihuandayani & Utami, 2018). Contohnya adalah ShopeePay yang merupakan salah satu aplikasi E-wallet yang terintegrasi pada sistem pembayaran E di aplikasi *E-commerce* Shopee (Kinasih & Kussudyarsana, 2023).

Integrasi sistem informasi yang efektif memerlukan perencanaan yang teliti dan mengikuti aturan yang jelas, sehingga dapat menghasilkan sistem yang sesuai dengan kebutuhan dan teknologi yang digunakan oleh suatu organisasi (Warkim & Sensuse, 2017). Sehingga penting untuk memahami perancangan sistem yang mampu mendesain relasinya dengan sistem lain. Dalam pengembangannya, diperlukan dasar sebagai konsep pembangunan sistem atau *Software Development Life Cycle* (SDLC). Salah satu metode SDLC adalah *Use Case Driven Object Modeling with UML*, yang dikenal sebagai *Iconix process* (Wahyu Agung Pratama, 2023). UML dapat menggambarkan desain sistem informasi yang akan dikembangkan, sehingga menghasilkan sistem informasi yang siap untuk digunakan oleh pengguna (Putra & Andriani, 2019). Metode *Iconix Process* dipilih karena menawarkan penggunaan UML yang minimalis sehingga hanya memerlukan beberapa langkah saja untuk melakukan analisis system (Tarisa Kamilia, 2023). Tahapan yang diperlukan dalam metode ini adalah *requirements, analysis/preliminary design, preliminary design review, detailed design,* dan *implementation* (Wahyu Agung Pratama, 2023).

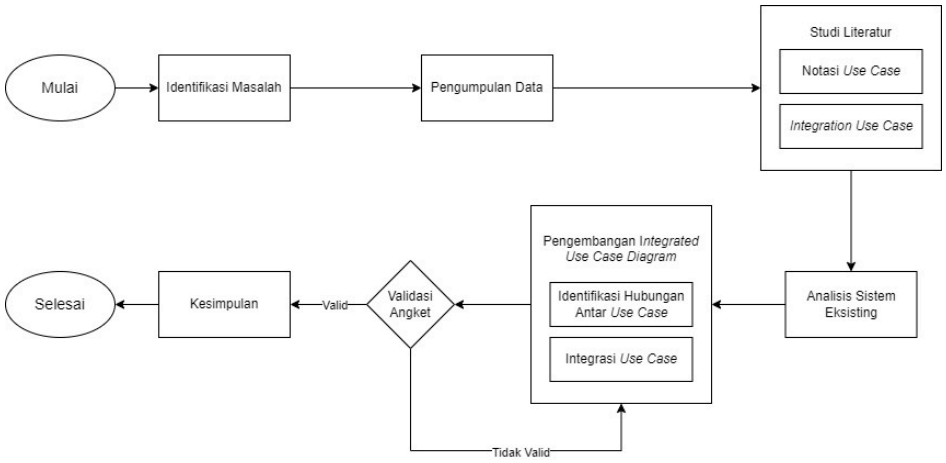
Pada tahap *requirements*, terdapat aktivitas *use case modelling*. Tahap ini merupakan bagian dari tahapan ICONIX Process, di mana dilakukan identifikasi aktor dan aktivitas proses bisnis yang sedang berlangsung. Hal ini menggambarkan kegiatan yang dilakukan pengguna serta bagaimana sistem meresponsnya (Wahyu Agung Pratama, 2023). *Use Case* (UC) adalah salah satu jenis diagram UML yang digunakan untuk memvisualisasikan interaksi antara pengguna dan sistem dalam lingkungan tertentu. (Sugeng Pranoto, 2024). Model UC berperan sebagai benang merah dalam seluruh proses pengembangan sistem. Ini berfungsi sebagai spesifikasi utama untuk persyaratan fungsional sistem, landasan analisis dan desain, masukan untuk perencanaan iteratif, dasar untuk definisi kasus uji, serta panduan untuk dokumentasi pengguna (Kurniawan, 2018). Dalam menggambarkan sebuah UC, terdapat 2 elemen penting yang harus digambarkan, yaitu aktor dan UC (Kurniawan, 2018). Setiap aktor dapat berinteraksi dengan sistem melalui berbagai Use Case (UC). Sebaliknya, setiap Use Case dapat dijalankan oleh lebih dari satu actor (Kurniawan, 2018). Antar aktor ataupun antar UC dapat memiliki relasi, masing-masing dengan spesifikasi yang berbeda (Kurniawan, 2018). Namun, pada beberapa penelitian, UC digunakan untuk merancang sistem tunggal. Belum banyak penelitian yang membangun UC untuk perancangan system terintegrasi dengan system yang lain, atau yang disebut dengan *integration use case* (IUC). *Integration Use Cases* (IUC) dapat digunakan untuk memodelkan layanan dari dua sudut pandang berbeda selama siklus hidupnya (Ville Alkkiomäki, 2007). IUC didasarkan pada standar UML, dan mereka tidak memasukkan elemen baru apa pun ke UML. (OMG: Unified Modeling Language version 2.0, 2005). IUC mengimplementasikan elemen UML standar dan mendefinisikan teknik untuk memodelkan kasus penggunaan dalam pendekatan *service oriented* (Ville Alkkiomäki, 2007).

Pada penelitian (Smolander, 2015) membangun sebuah IUC mewakili antarmuka layanan abstrak antara penyedia layanan dan konsumen layanan. Jika *enterprise service bus* (ESB), antrian pesan, atau sistem middleware lain digunakan dalam integrasi sistem, maka kasus penggunaan integrasi juga menjelaskan peran dan tindakan middleware antar sistem (Smolander, 2015). Penelitian lain oleh (Ville Alkkiomäki, 2007) yang mengambil contoh di mana pengguna menggunakan In-House System A untuk mengirim pesan ke subkontraktor perusahaan. IUC dibuat dengan mengintegrasikan antara tiga

system (Ville Alkkiomäki, 2007). Susunan artikel ini adalah; pendahuluan termasuk berisi *review* paper lain, metodologi, hasil dan pembahasan, serta kesimpulan sekaligus saran penelitian kedepan.

2. Metodologi

Dalam mengembangkan perangkat lunak, proses perancangan sistem termasuk dalam tahap desain (*design phase*) di kaidah *software development life cycle* (SDLC). Bagian yang krusial dari tahap desain adalah pembuatan *use case,* karena dari sini akan menurun ke pembuatan desain-desain berikutnya seperti *use case scenario*, *activity diagram*, *robustness diagram*, *sequence diagram*, dan *class diagram*.



Gambar 1. Metodologi Penelitian

Gambar 1 menampilkan metodologi pada penelitian ini. *Step by step* dimulai dari:

a. Identifikasi masalah: Tahap pertama pada penelitian ini yaitu mengidentifikasi permasalahan yang muncul terkait integrasi *use case* melalui observasi
b. Pengumpulan data: Metode pengumpulan data dilakukan dengan kualitatif melalui wawancara. Pada tahap ini dilakukan pengumpulan data terkait sistem eksisting dan sistem yang akan dibangun.
c. Studi literatur: Tahapan yang dilakukan mencakup mencari literatur terkait notasi *use case diagram,* dan implementasi *integration use case*
d. Analisis sistem eksisting: Berdasarkan informasi yang didapatkan dari tahapan pengumpulan data, pada penelitian ini mengangkat sistem manajemen sekolah
e. Pengembangan *integrated use case diagram:* mencakup identifikasi hubungan antar *use case,* dan melakukan integrasi pada *use case* tersebut. Selanjutnya dilakukan Tarik Kesimpulan, dan tahapan selesai.

Penggambaran atau perancangan perangkat lunak yang digunakan sudah terstandarisasi dan menjadi bahasa pemodelan baku yakni *unified modeling language* (UML). Pada sub-bab selanjutnya, kami menjelaskan standar notasi *use case* dalam UML, lalu membahas konsep dan teknik penggambaran *integration use case*.

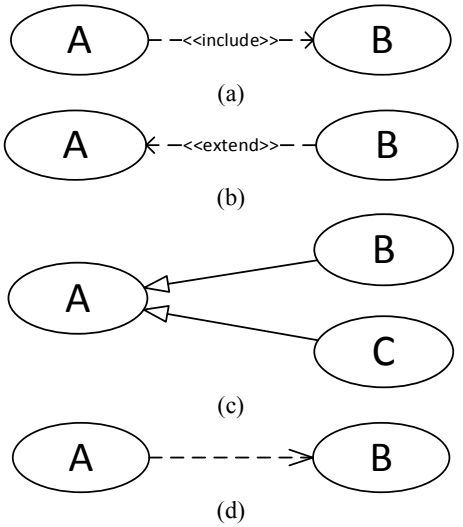
2.1 Notasi relasi *use case diagram* yang umum digunakan dalam UML

Use case adalah UML pada tahap pertama sebagai artifak yang menentukan keberhasilan pengembangan perangkat lunak. Use case diperoleh dari hasil penggalian *system requirement* utama pada tahapan analisis SDLC. Gambar 2 menunjukkan notasi-notasi relasi antar *use case* yang paling sering digunakan dalam menggambar *use case diagram*.

Notasi relasi *include* menggambarkan; setelah selesai memproses *use case* A, maka selanjutnya akan memproses *use case* B (*mandatory*) seperti ditunjukkan pada gambar 2 (a). Selanjutnya, gambar 1 (b) relasi *extend* antar *use case* yang artinya setelah proses *use case* A, bisa memproses *use case* B atau tidak (*optional*). Contoh mudahnya *use case* B adalah *use case* 'meminta bantuan' (*help menu*). Relasi *generalization* dari *use case* B dan C, seperti ditunjukkan gambar 1 (c), merupakan relasi spesifikasi

ISSN: 2579-566X (Online)
ISSN: 2477-5274 (Print)

dari *use case* A. Pemberian nama pada *use case* B dan C boleh tidak mengikuti kaidah pemberian nama *use case*; kata kerja + kata benda. Contohnya seperti bila *use case* A 'melakukan pembayaran', *use case* B dan C dapat berupa *use case* 'pembayaran melalui credit card' dan 'pembayaran melalui bank', berurutan. Gambar terakhir 1 (d) adalah relasi *dependency*, ini mirip seperti *include*, namun ada ketergantungan keberhasilan *use case* sebelumnya. Misalkan, *use case* A 'mengautorisasi pengguna', bila berhasil maka ke *use case* B 'memvalidasi kredensial'.

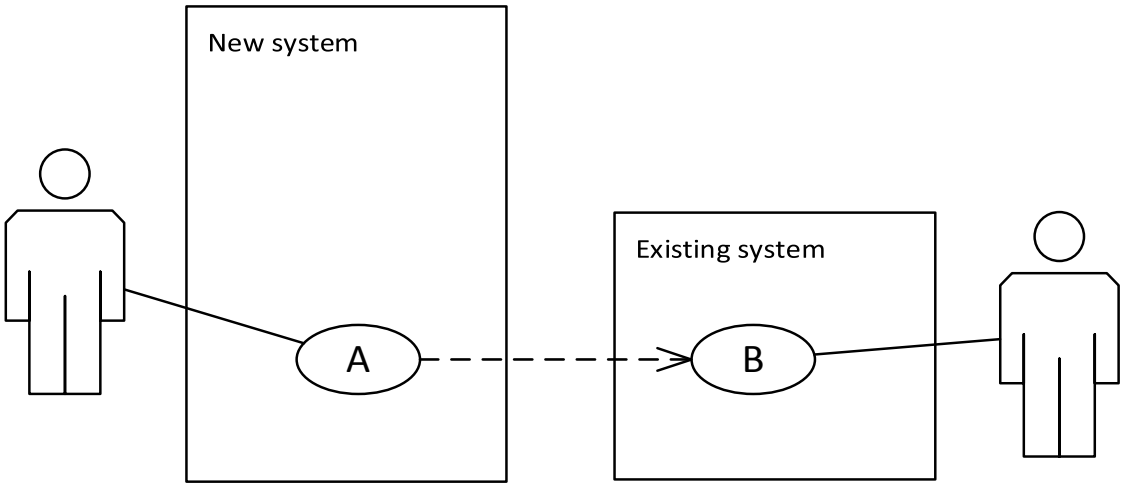


Gambar 2. Jenis-jenis relasi antar *use case* yang umum digunakan; (a) relasi *include*, (b) relasi *extend*, (c) relasi *generalization*, dan (d) relasi *dependency*

2.2 Pemodelan *integration use case* (IUC) untuk sistem-sistem terintegrasi

Pemahaman relasi antar use case pada sub-bab 2.1 menjadi acuan pada pembahasan *use case* diagram untuk sistem terintegrasi berikut. Pembuatan use case diagram tidak sama seperti membuat alur proses atau skenario, karena itu perlu adanya *use case scenario*. Sehingga, apa yang dimunculkan ke dalam sebuah *use case* merupakan yang memang menjadi requirement utama dari sistem yang hendak dikembangkan. *Use case diagram* untuk sistem terintegrasi dapat bedakan menjadi dua, yaitu terintegrasi dengan sistem eksisting (*existing*) atau sistem yang memang juga baru dirancang (*new*).

2.2.1 IUC sistem terintegrasi sistem eksisting



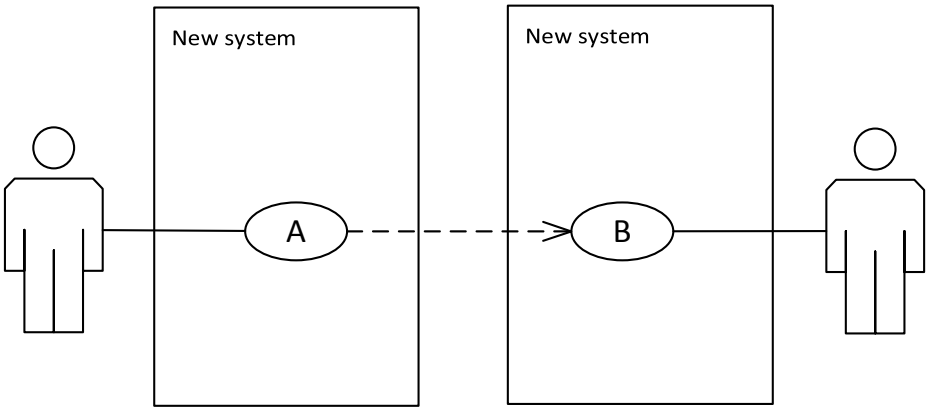
Gambar 3. *Integration use case* diagram untuk integrasi dengan sistem lama

Penggunaan korelasi antar use case pada sistem integrasi dapat menggunakan notasi relasi yang sama seperti dalam satu sistem, seperti *include relation, extend relation*, dst. Gambar 3 menunjukkan hubungan relasi antar use case yang terhubung lintas *boundary system* dengan relasi *dependency*. Relas lintas *boundary system* ini membutuhkan penanganan secara khusus misalkan pendekatan budaya *development and operations* atau penggunaan layanan API atau lainnya.

Use case A berada dalam *boundary system* yang memang hendak dibangun (*new system*), sedangkan use case B berada pada *boundary system* yang telah ada aplikasi atau sistemnya (*existing system*). Misalkan, *use case* A dapat berupa *use case* 'melakukan registrasi' yan itu terintegrasi dengan sistem autentikasi yang telah ada sebelumnya, yaitu *use case* 'mengautentikasi akun'. Hal itu dikarenakan, misalkan, proses autentikasi akun tersebut memiliki fokus penjaminan kualitas tersendiri, sehingga harus melalui verifikasi email atau nomor handphone.

2.2.2 IUC sistem terintegrasi sistem baru

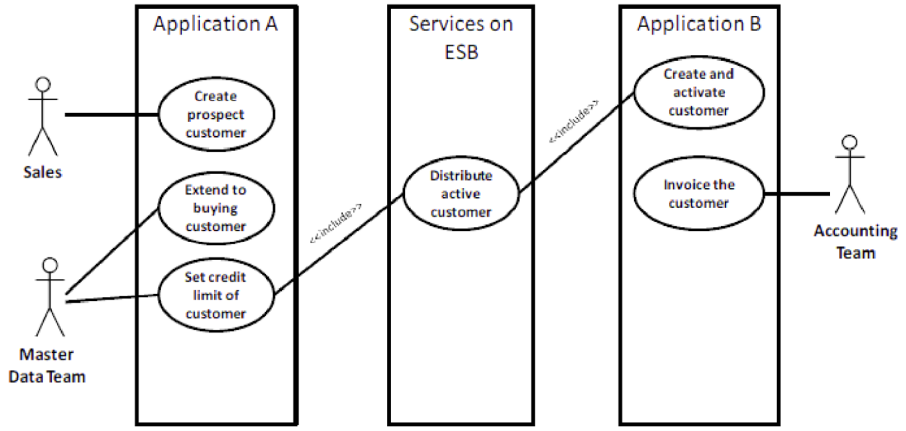
Gambar 4 menunjukkan relasi antar use case untuk kondisi kedua sistem sama-sama hendak dikembangkan, belum ada aplikasi eksisting.



Gambar 4. *Integration use case diagram* untuk integrasi dengan sistem baru

IUC diagram integrasi ke sistem baru artinya kedua sistem merupakan sistem yang dikembangkan. Dalam proses pengembangannya biasanya membutuhkan lebih banyak relasi antar use case atau saat diturunkan ke use case scenario daripada integrasi ke sistem lama. Contohnya adalah perusahaan yang bergerak di bidang peternakan digital mengembangkan sistem untuk calon pelanggan dapat memilih dan membeli hewan ternak, tapi satu sisi anak perusahaannya mengembangkan aplikasi sebagai perspektif pelanggan ketika telah membeli.

Lebih lanjut, kami juga menambahkan bagaimana penerapan *integration use case diagram* dengan perantara *services* seperti yang telah dilakukan oleh artikel publikasi lain yang ditampilkan pada Gambar 6.



Gambar 5. *Integration use case* (IUC) dengan *service-oriented architecture* di antara aplikasi A dan B oleh Alkkiomaki dan Smolander (2015).

 ISSN: 2579-566X (Online)
ISSN: 2477-5274 (Print)

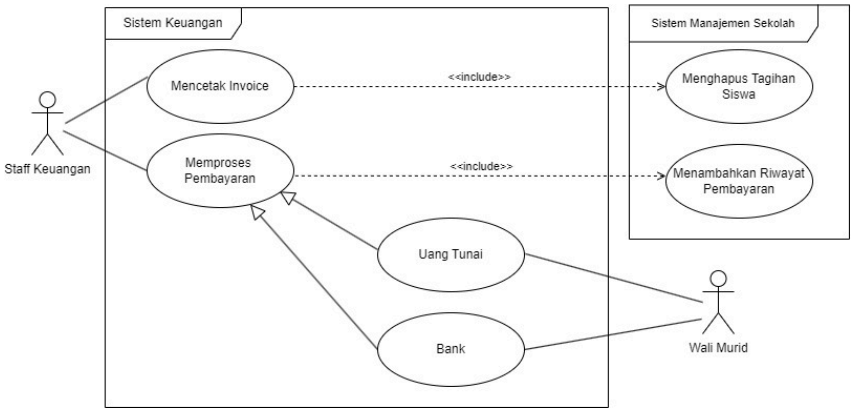
Dalam publikasinya berjudul '*Observations of service identification from two enterprises'* pada integrasi *use case* seperti Gambar 5. Dalam tulisannya, Alkkiomaki dan Smolander (2015) memanfaatkan enterprise service bus (ESB) sebelum aplikasi A terintegrasi dengan aplikasi B. Hal ini serupa seperti penggunaan *third-party* seperti pembuatan API services untuk memperoleh data tertentu pada aplikasi lain. Sehingga, *services boundary 'services on ESB'* sangat penting bagi developer untuk menunjukkan bahwa integrasi tidak langsung dihubungkan ke *use case* atau fitur dari aplikasi lain, melainkan melalui suatu layanan terlebih dahulu.

3. Hasil dan Pembahasan

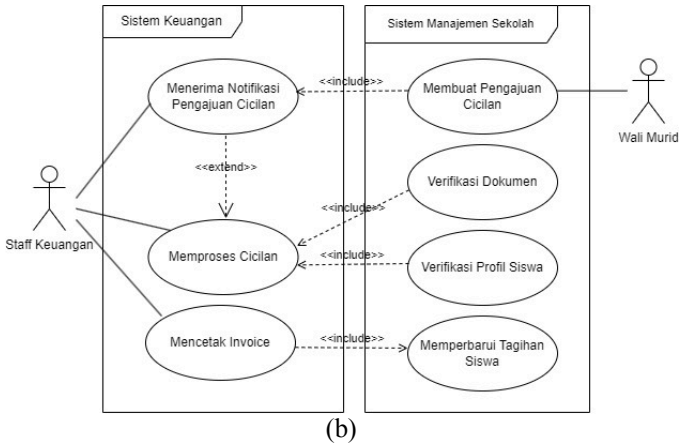
Pada bab ini kami membahas hasil implementasi metodologi pengembangan desain *integration use case* dengan studi kasus sistem integrasi keuangan terhadap sistem manajemen eksisting maupun baru SD Kumalasari Bhayangkari 1 Surabaya menggunakan standar SDLC model *waterfall*, dan bagaimana hasil validasi sistemnya menggunakan penyebaran kuesioner pengujian blackbox dengan pertimbangan perhitungan total skor empirik (TSE) dan skor maksimal (SMAX).

3.1 Hasil implementasi pada studi kasus sistem integrasi keuangan sekolah

Gambar 6 adalah desain IUC dengan studi kasus pada sistem informasi manajemen sekolah pada sistem keuangan. Gambar 6 (a) menunjukkan sistem manajemen sekolah sebagai sistem eksisting. *Use case* dari sistem keuangan dapat dihubungkan langsung ke *use case* 'menghapus tagihan siswa'. Use case 'memproses pembayaran' pun juga sama. Sehingga, relasinya lintas *system boundary*. Mirip dengan itu, pada sistem yang hendak dibangun sistem manajemen sekolah direlasikan dengan sistem keuangan seperti pada Gambar 6 (b). Sehingga, dalam konteks ini keduanya sama-sama sistem yang akan dikembangkan namun pada dua aplikasi yang berbeda.



(a)



(b)

Gambar 6. Contoh penerapan IUC; (a) sistem keuangan ke sistem eksisting manajemen sekolah, dan (b) sistem keuangan ke sistem baru manajemen sekolah.

Lebih lanjut, berdasarkan desain integrasi di atas, bahwa sistem keuangan merupakan sistem baru yang hendak dikembangkan pada gambar (a), sedangkan gambar (b) menunjukkan bahwa kedua sistem; sistem keuangan dan sistem manajemen sekolah, adalah sistem yang sama-sama hendak dibangun atau dikembangkan. Akan tetapi, walaupun sama-sama sistem yang hendak dikembangkan, gambar (b) memiliki spesifikasi fitur yang berbeda seperti proses cicilan dan proses tagihan. Oleh karena itu, kedua sistem tidak bisa dimasukkan ke dalam suatu *system boundary* yang sama karena dapat dimungkinkan agar dikembangkan oleh tim pengembang yang berbeda. Dalam praktiknya, tim pengembang sistem keuangan cukup berfokus pada use case yang hanya terlibat di sistem keuangan. Sebaliknya, tim pengembang hanya berfokus pada use case yang terdapat di sistem manajemen sekolah. Sehingga, di akhir dari pengembangan ada proses penyatuan sistem atau biasa disebut integrasi fitur.

Contoh lain biasa terjadi ketika kedua aktor berfokus pada pengembangan fitur yang berbeda. IUC dengan kondisi ini membutuhkan saling berkolaborasi antar tim pengembang dengan baik sejak dari awal pengembangan perangkat lunak, oleh karena itu, hingga munculnya konsep budaya development and operations. Relasi lintas sistem menunjukkan penggunaan fitur atau menu itu sudah masuk ke aplikasi lain. Misalkan, penggunaan suatu aplikasi pemesanan makanan dan minuman (*food and beverage*) dapat memilih menu makanan yang akan dipesan. Lalu ketika masuk ke tahap pembayaran, aplikasi akan mengarahkan ke aplikasi lain sejenis *e-wallet*. Aplikasi ini membutuhkan pengguna memasukkan pin, lalu akan kembali ke aplikasi restoran tadi.

3.2 Validasi sistem

Setelah itu, dilakukan validasi angket untuk staff dan guru sekolah. Responden pada uji coba desain *use case diagram* adalah guru dan staff SD Kemala Bhayangkari 1 Surabaya yang terdiri dari 14 orang. Angket dihitung menggunakan rumus perhitungan TSE dan SMAX dengan hasil 88,61%. Data hasil uji desain ditunjukkan pada Tabel 1.

Tabel 1. Hasil Validasi Angket

No	*Pernyataan*	*Penilaian*			
		TSE	***SMAX***	***V***	***Kategori***
1	Kemudahan implementasi desain use case	68	72	94,44%	Valid
2	Pemahaman integrasi antar sistem	52	72	72,22%	Valid
3	Kesesuaian dengan kebutuhan fungsional	64	72	88,89%	Valid
4	Pemahaman notasi use case	70	72	97,22%	Valid
5	Kemudahan actor dalam menjalankan sistem	65	72	90,28%	Valid
	Total Skor	319	360	88,61%	Valid

4. Kesimpulan

Artikel ini memaparkan pentingnya pemahaman perancangan *use case* diagram untuk sistem terintegrasi, yaitu dengan sistem baru dan sistem eksisting, mengingat semakin pesatnya kolaborasi antar perusahaan dan menuntut aplikasi agar mampu terintegrasi aplikasi-aplikasi lain. Kemudian, teori dasar seputar relasi dipaparkan di awal pembahasan, terutama pada perbedaan relasi *dependency*, relasi *include, relasi extend*, dan relasi *generalization*. Berdasarkan percobaan pada implementasi di SD Kumalasari Bhayangkari 1 Surabaya, sistem keuangan dapat diintegrasikan baik dengan sistem eksisting maupun sistem yang juga baru melalui IUC dengan tingkat skor pengujian blackbox sebesar 88,61%. Selanjutnya penerapan dengan studi kasus sistem informasi manajemen (SIM) sekolah dengan aplikasi yang dirancang yaitu sistem keuangan. Penelitian selanjutnya adalah menarik untuk dikaji yaitu turunan dari *integration use case diagram* pada UML berikutnya seperti *activitiy diagram, robustness diagram, dan sequence diagram* dengan tetap membawa esensi dari *use case* integrasi.

Referensi

Idris, I., & Delvika, Y. (2014). ANALISIS PERANCANGAN SISTEM INFORMASI TERINTEGRASI DI LINGKUNGAN PERGURUAN TINGGI SWASTA DI MEDAN. *Teknovasi*, 15-26.

Kinasih, D. P., & Kussudyarsana. (2023). The Influence Of E-Service Quality, E-Payment, And Web Design Quality On Customer Satisfaction In Purchasing Gacoan Noodles Through The Shopeefood Application. *Jurnal Ilmiah Ekonomi dan Bisnis*, 159-178.

Kurniawan, T. A. (2018). PEMODELAN USE CASE (UML): EVALUASI TERHADAP BEBERAPA KESALAHAN DALAM PRAKTIK. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, 77-86.

Mihuandayani, & Utami, E. (2018). Design Concept Integration Tax Payment System with Implementing Financial Technology. *I.J. Information Engineering and Electronic Business*, 15-22.

OMG: Unified Modeling Language version 2.0. (2005, July). Retrieved from OMG: Unified Modeling Language version 2.0. Online, http://www.omg.org: https://www.omg.org/spec/UML/2.0/

Putra, D. W., & Andriani, R. (2019). Unified Modelling Language (UML) dalam Perancangan Sistem Informasi Permohonan Pembayaran Restitusi SPPD. *TEKNOIF*, 32-39.

Smolander, V. A. (2015). OBSERVATIONS OF SERVICE IDENTIFICATION FROM TWO ENTERPRISES. *International Journal of Software Engineering & Applications (IJSEA)*, 33-43.

Sugeng Pranoto, S. S. (2024). Penerapan UML Dalam Perancangan Sistem Informasi Pelaporan Dan Evaluasi Pembangunan Pada Bagian Administrasi Pembangunan Sekretariat Daerah Kota Tebing Tinggi. *SURPLUS : JURNAL EKONOMI DAN BISNIS*, 384 - 401.

Tarisa Kamilia, E. M. (2023). Designing Information System Inventory and Transaction Reports Web-Based Using ICONIX Process Method. *Inform : Jurnal Ilmiah Bidang Teknologi Informasi dan Komunikasi*, 14-26.

Verbeke, W., Dejaeger, K., Martens, D., Hur, J., & Baesens, B. (2012). New Insights into Churn Prediction in the Telecommunication Sector: A Profit Driven Data Mining Approach. *European Journal of Operational Research, 218*(1), 211-229. doi:10.1016/j.ejor.2011.09.031

Ville Alkkiomäki, K. S. (2007). Integration Use Cases – An Applied UML Technique for Modeling Functional Requirements in Service Oriented Architecture . *Springer-Verlag Berlin Heidelberg*, 190–202.

Wahyu Agung Pratama, Z. S. (2023). Rancang Bangun Pendataan Stok Barang Operasional Dengan Metode Iconix Process Berbasis Web. *Bulletin of Information Technology (BIT)*, 105 - 111.

Warkim, & Sensuse, D. I. (2017). Model Integrasi Sistem dengan Pendekatan Metode Service Oriented Architecture dan Model View Controller pada Pusat Penelitian Perkembangan Iptek Lembaga Ilmu Pengetahuan Indonesia. *Jurnal Teknik Informatika dan Sistem Informasi*, 84-103.

Witten, I. H., Frank, E., & Hall, M. A. (2011). *Data Mining: Practical Machine Learning Tools and Techniques* (3rd ed.). Burlington: Morgan Kaufmann.

Yap, B. W., Rani, K. A., Rahman, H. A., Fong, S., Khairudin, Z., & Abdullah, N. N. (2014). An Application of Oversampling, Undersampling, Bagging and Boosting in Handling Imbalanced Datasets. *Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013). 285*, pp. 13-22. Singapore: Springer. doi:10.1007/978-981-4585-18-7_2