

PROSES ICONIX DALAM ANALISA RANCANGAN APLIKASI INFORMASI JADWAL DAN TUGAS BERBASIS ANDROID

Aidina Ristyawan

Fakultas Teknik, Program Studi Sistem Informasi
Universitas Nusantara PGRI Kediri
Email: ristykdir@gmail.com

Dwi Harini

Fakultas Teknik, Program Studi Sistem Informasi
Universitas Nusantara PGRI Kediri
Email address: dwiharini1970@yahoo.com

ABSTRAK

Dalam pembuatan suatu perangkat lunak yang efisien dan sesuai dengan kebutuhan fungsional pengguna, dibutuhkan suatu analisa perancangan yang mampu menjembatani antara kebutuhan pengguna dengan proses pembuatan kode perangkat lunak, dari analisa perancangan yang tepat maka akan memudahkan pengembang membuat kode perangkat lunak. Terdapat beberapa model yang dapat menggambarkan analisa perancangan untuk pembuatan suatu perangkat lunak. Model tersebut diantaranya yang sering digunakan adalah UML. Namun dengan banyaknya diagram notasi UML sering membuat kita beranggapan bahwa UML merupakan langkah – langkah apa saja yang harus dilakukan dalam merancang sistem atau perangkat lunak. Padahal UML sebenarnya hanya kumpulan diagram / notasi dan teknik visualisasi, tidak menentukan urutan – urutan langkah dalam membuat perangkat lunak. Dengan menggunakan *ICONIX Process* kita dapat menggunakan UML untuk menggambarkan model perancangan dan sekaligus menganalisa dari rancangan yang kita buat. Pada penelitian ini peneliti membuat diagram analisa dan perancangan yang nantinya dapat digunakan sebagai panduan atau gambaran umum dalam pembuatan perangkat lunak informasi jadwal dan tugas kuliah berbasis android. Pada penelitian ini hanya membahas tentang model analisa dan perancangannya saja.

Kata kunci: perancangan; *iconix*; UML; informasi.

ABSTRACT

In making software that is efficient and in accordance with the user requirement's, it needs an analysis that can bridge the needs of the user with the device manufacturing process, from the right design analysis that will allow users to create soft code devices. Some models can be used to process software. These models are UML. But with UML notation diagrams that often make us assume that UML is any steps that must be done in a device system. UML is actually just a collection of diagrams / notations and visualization techniques, does not specify sequences - makes steps in creating software. By using the ICONIX process we can use UML to describe the model and analyze the production we make. In this study, researchers make analysis and design diagrams that can be used as a guide or making general data in making android timetable information and assignment information software. In this study only discussed about the analysis and design models only.

Keywords: *design; iconix; UML; information.*

1. PENDAHULUAN

Di dalam pembuatan perangkat lunak yang baik pasti diperlukan suatu rancangan yang baik pula, dan tentu saja perangkat lunak yang akan dibuat tersebut harus memenuhi kebutuhan fungsional dari pengguna. Dalam proses perancangan perangkat lunak terdapat beberapa peranan [1], yaitu : memberikan panduan tentang aktivitas yang dilakukan tim, menentukan artefak apa saja yang harus dibuat, menentukan tugas pada pengembang baik secara anggota tim ataupun pribadi dan memberikan kriteria sebagai tolok ukur hasil dan aktivitas proyek.

Dalam melakukan perancangan kita bisa mengvisualisasikan dengan menggunakan notasi UML (*Unified Modeling Language*), UML merupakan bahasa pemodelan standar untuk merancang pengembangan perangkat lunak [2]. Namun dengan keberadaan notasi UML yang beraneka ragam terkadang membuat kita beranggapan bahwa UML merupakan langkah – langkah yang harus dilakukan

dalam merancang sistem atau perangkat lunak. Padahal UML sebenarnya hanya kumpulan diagram / notasi dan teknik visualisasi, tidak menentukan urutan – urutan langkah dalam membuat perangkat lunak. Hal ini sering terjadi dikarenakan bahwa UML dianggap merupakan langkah – langkah secara teori saja dalam pembuatan perangkat lunak, sehingga sering mengakibatkan sering dilupakannya proses perancangan terhadap pembuatan aplikasi, atau bahkan dilakukan secara terbalik, yaitu pembuatan aplikasi terlebih dahulu, baru kemudian dibuatlah diagram UML. Lalu bagaimanakah cara kita dapat membuat perangkat lunak yang sesuai dengan kebutuhan pengguna?. Tentunya harus kita rancang terlebih dahulu keperluan apa saja yang dibutuhkan dalam aktivitas perangkat lunak yang akan dibuat.

Mengingat pentingnya analisa perancangan suatu perangkat lunak, sehingga perangkat lunak yang sudah jadi kedepannya efisien dan dapat memenuhi kebutuhan pengguna, terutama pada Program Studi Sistem Informasi Universitas Nusantara PGRI Kediri, maka pada penelitian ini akan dilakukan perancangan yang bertujuan untuk dapat digunakan sebagai panduan dan atau gambaran kebutuhan fungsional dari aplikasi informasi jadwal dan tugas berbasis android yang akan dibuat pada penelitian selanjutnya. Rancangan perangkat lunak yang diteliti, akan divisualisasikan dengan menggunakan notasi atau diagram UML, dan dibantu dengan metode perancangan *ICONIX Process*.

Berdasar latar belakang tersebut, peneliti menentukan rumusan masalah berupa : 1) Bagaimana proses pembuatan rancangan aplikasi informasi jadwal dan tugas berbasis android ? dan 2) Diagram rancangan apa sajakah yang dibutuhkan untuk pembuatan aplikasi informasi jadwal dan tugas berbasis android nantinya ?.

1.1 UML

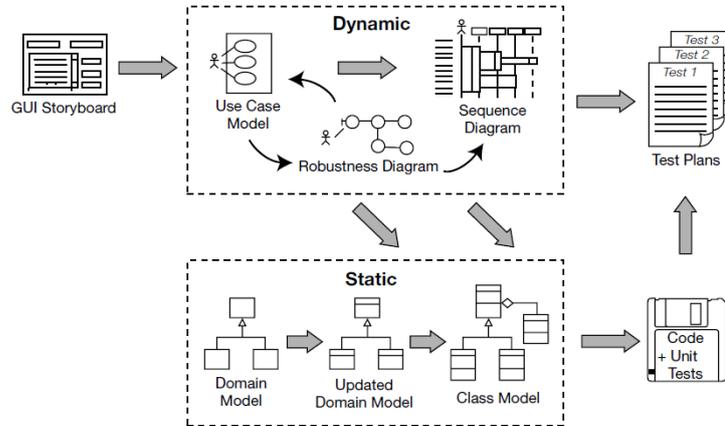
UML merupakan bahasa pemodelan baku yang digunakan untuk pengembangan rancangan perangkat lunak dan sistem [2]. UML dapat digunakan untuk membuat rancangan semua model jenis perangkat lunak, dimana perangkat lunak tersebut dapat dijalankan baik pada perangkat keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Dalam UML juga menggunakan *class* dan *operation*, sehingga UML cocok digunakan untuk merancang model pembuatan perangkat lunak berbasis objek [3]. Terdapat beberapa diagram pemodelan pada UML yang sering digunakan[2], diantaranya adalah :

- a) *Use Case* : Model diagram yang menggambarkan tentang kebutuhan fungsional dari suatu perangkat lunak.
- b) *Activity Diagram* : Model diagram yang menggambarkan tentang alur kerja dari perangkat lunak.
- c) *Class Diagram* : Diagram yang menggambarkan struktur, deskripsi class, package dan objek beserta hubungannya satu sama lain seperti containment, pewarisan, asosiasi, dan lain - lain.
- d) *Statechart Diagram* : Diagram yang menggambarkan transisi dan perubahan keadaan (dari satu state ke state lainnya) dari suatu objek pada sistem sebagai akibat dari pemicu yang diterima
- e) *Activity Diagram* : Diagram yang menggambarkan setiap aliran aktivitas dalam sistem yang dirancang, bagaimana masing - masing alur berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir.
- f) *Sequence Diagram* : Diagram yang menggambarkan interaksi antar objek baik di dalam maupun di sekitar sistem (termasuk pengguna, tampilan, dan sebagainya) berupa message yang digambarkan terhadap waktu.
- g) *Collaboration Diagram* : Diagram yang menggambarkan interaksi antar objek seperti sequence diagram, tetapi lebih menekankan pada peran setiap objek dan bukan pada waktu penyampaian message.
- h) *Component Diagram* : Diagram yang menggambarkan struktur dan hubungan antar komponen perangkat lunak, termasuk ketergantungan (*dependency*).
- i) *Deployment Diagram* : Diagram yang menggambarkan secara detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisik.

1.2 *ICONIX Process*

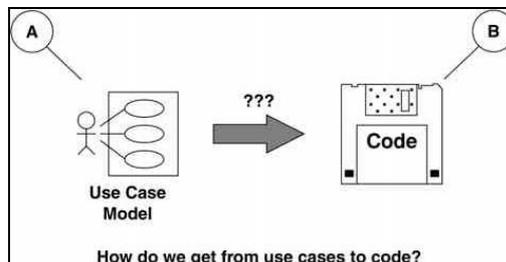
Pendekatan *ICONIX Process* berada ditengah antara pendekatan *Rational Unified Process* (RUP) yang luas dan pendekatan *eXtreme Programming* (XP) yang sangat sempit. Pendekatan *ICONIX Process* didasarkan pada notasi *use case* seperti RUP, tetapi tidak terlalu rumit seperti yang dihasilkan oleh RUP. Lingkup *ICONIX Process* juga sempit dan singkat seperti XP, namun mengutamakan analisa dan desain seperti yang dilakukan XP [4]

Secara teori, setiap aspek dari UML berpotensi berguna, tetapi dalam prakteknya, tampaknya tidak pernah ada cukup waktu untuk melakukan pemodelan, analisis, dan desain. Selalu ada tekanan dari manajemen untuk beralih ke pembuatan kode, untuk memulai pengodean lebih awal karena kemajuan pada proyek perangkat lunak cenderung diukur oleh seberapa banyak kode yang ada. Proses *ICONIX*, seperti yang diperlihatkan dalam gambar 1, adalah pendekatan minimalis yang disederhanakan yang berfokus pada area yang terletak di antara penggunaan kasus dan kode [5].



Gambar 1. *ICONIX* Process

Tujuan utama dari *ICONIX* Process adalah menjembatani bagaimanakah membuat kode program berdasarkan use case yang telah dibuat. Seperti terlihat pada gambar 2, Di mana titik A menggambarkan ide atau fungsi apa saja yang harus dilakukan sistem (digambarkan dalam bentuk use case). Sedangkan titik B menggambarkan kode - kode program yang komplit, telah diuji, dan telah bisa mengerjakan apa yang dijabarkan pada use case[4].



Gambar 2. Cara kerja *ICONIX* Process

ICONIX Process berusaha menjembatani antara pembuatan kode program (titik B) berdasar kebutuhan sistem yang tergambar pada *use case* (titik A). *ICONIX* Process dapat dikatakan panduan dalam pengembangan perangkat lunak yang mirip dengan buku masakan yang maksudnya adalah terdapat penjelasan setiap seri dari langkah – langkah yang spesifik untuk membuat model pengembangan perangkat lunak [6]. Dalam *ICONIX* Process terdapat empat langkah yang harus dikerjakan diantaranya [7] adalah :

- a) *Requirements*
 - 1) *Functional Requirements* : adalah kebutuhan fungsional yang diperlukan dalam pembuatan perangkat lunak.
 - 2) *Domain Modeling* : merupakan penggambaran objek nyata terhadap keadaan yang sebenarnya dari suatu lingkup area yang diwujudkan ke dalam bentuk abstraksi kelas. Kata kunci dalam menentukan Domain Model adalah objek – ubjek utama (kata benda) yang terlibat dalam proses bisnis suatu sistem [8]. tahap ini wajib dilakukan karena menggambarkan kerangka berpikir tentang informasi apa saja yang akan diolah oleh sistem [9].
 - 3) *GUI Storyboard* : merupakan gambaran awal dan sederhana dari tampilan sistem yang akan dibuat.

- 4) *Use Case Modeling* : merupakan tahapan yang mengidentifikasi aktor atau setiap pengguna yang terlibat dalam penggunaan sistem beserta masing – masing skenario dasar dan skenario alternatifnya
- b) *Analysis and Preliminary Design*
 - 1) *Robustness Analysis* : merupakan analisa kehandalan yang berfungsi untuk menemukan objek – objek baru yang belum teridentifikasi sebelumnya [10].
 - 2) *Update Domain Model* : perubahan yang dilakukan terhadap *Domain Modeling* dari tahap *Requirements* sebelumnya. Perubahan tersebut baik menghilangkan kelas – kelas yang sama atau ganda maupun penambahan kelas – kelas baru serta pemberian atribut pada setiap kelas. Perubahan ini menyesuaikan dari hasil analisa kehandalan.
- c) *Detailed Design*
 - 1) *Sequence Diagram* : merupakan gambaran interaksi antar objek baik di dalam maupun di sekitar sistem (termasuk pengguna, tampilan, dan sebagainya) berupa message yang digambarkan terhadap waktu.
 - 2) *Update Domain Model* : perubahan yang dilakukan kembali terhadap *Domain Modeling* dari tahap *Analysis and Preliminary Design* sebelumnya berdasarkan hasil perancangan *Sequence Diagram*.
- d) *Implementation*
 - 1) *Coding / Unit Testing* : merupakan tahap pembuatan kode program untuk mengembangkan atau membuat sistem beserta pengujian setiap algoritma kode program.
 - 2) *Integration and Scenario Testing* : penggabungan setiap unit / kode program menjadi satu kesatuan sistem beserta pengujian sistem secara keseluruhan.

2. METODOLOGI PENELITIAN

Metodologi penelitian adalah gambaran langkah – langkah yang di tempuh dalam menjalankan penelitian, adapun metode penelitian yang digunakan pada penelitian ini adalah metode penelitian studi kasus yang akan digabungkan dengan model *ICONIX Process*. penggunaan *ICONIX Process* cukup efisien karena tanpa menggunakan banyak pentabelan namun tanpa mengabaikan analisis dan perancangan yang singkat seperti halnya *eXtreme Programming* sehingga tetap fokus pada definisi kebutuhan sistem [11] dan kebutuhan pengguna [12]. Bahkan *ICONIX Process* telah berhasil digunakan dalam perancangan teknologi *Reality Augmented* [13]. Adapun urutan langkah metode penelitian studi kasus sebagai berikut :

- a) Pemilihan kasus
- b) Pengumpulan data
- c) Analisis data
- d) Perbaikan (refinement)
- e) Penulisan laporan

Pada penelitian ini peneliti memasukkan setiap langkah – langkah yang terdapat pada model *ICONIX Process*. Adapun langkah – langkahnya adalah :

- a) Pemilihan kasus
Pada penelitian ini peneliti memilih kasus tentang proses proses *ICONIX* yang dikerjakan dalam membuat rancangan aplikasi informasi jadwal dan tugas berbasis android
- b) Pengumpulan data
Data yang digunakan dalam penelitian ini menggunakan data KRS setiap mahasiswa dan jadwal mengajar dosen.
- c) Analisis data
Analisa yang digunakan dari *ICONIX Proses* adalah : pembuatan *functional requirement, domain model, use case behavioral requirement, menentukan Technical Architecture dan pembuatan sequence diagram*.
- d) Perbaikan (Refinement)
Requirement review, Robustness Analysis, Preliminary Design Review dan Critical Design Review.
- e) Penulisan laporan / dokumentasi
Merupakan hasil akhir yang berbentuk artikel penelitian.

3. HASIL DAN PEMBAHASAN

Untuk melakukan pembuatan perangkat lunak yang efektif dan efisien sesuai dengan kebutuhan proses bisnis dan kebutuhan fungsional, perlu dilakukan pembuatan rancangan perangkat lunak. Pembuatan rancangan inilah yang akan menjelaskan dan memandu bagaimanakah langkah - langkah

teknis pembuatan kode program perangkat lunak. Adapun rancangan yang telah dibuat berdasar *ICONIX Process* adalah :

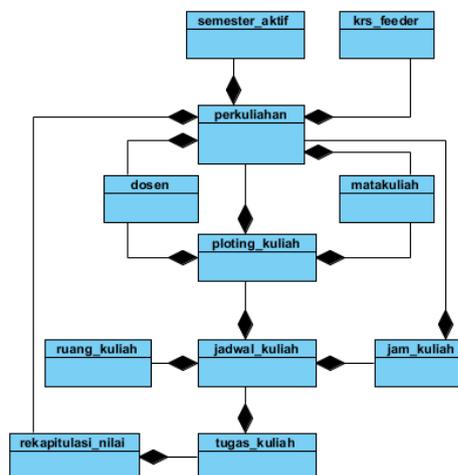
3.1 Penulisan *Functional Requirement Awal*

pada langkah ini mendefinisikan kebutuhan fungsional yang diinginkan pengguna agar bisa dikerjakan oleh perangkat lunak, adapun hasil *Functional Requirement* awal sebagai berikut :

- Sistem harus dapat melakukan proses pendataan kegiatan perkuliahan mulai dari input **semester aktif**, data “**krs-feeder**” dari SIAKAD Universitas Nusantara PGRI melalui aplikasi web yang dilakukan oleh petugas administrasi program studi. Dari hasil tersebut sistem harus bisa melakukan proses pendataan **matakuliah**, dan **ploting dosen** secara otomatis. Format data krs meliputi : npm, nama, program studi, kode matakuliah, dosen, kelas dan status.
- Sistem harus dapat melakukan manajemen **jadwal perkuliahan** mulai dari pendataan **jam kuliah** dan penjadwalan kuliah berdasar data krs yang telah dimasukkan, proses ini dilakukan oleh petugas administrasi program studi
- Sistem harus dapat menampilkan **informasi Jadwal perkuliahan** kepada mahasiswa sesuai dengan matakuliah yang telah diambil oleh mahasiswa melalui data krs. Informasi meliputi : Hari, jam perkuliahan, matakuliah, dosen pengajar, ruangan perkuliahan.
- Sistem harus dapat melakukan proses penugasan mahasiswa dan memasukan nilai mahasiswa dari tugas yang diberikan oleh dosen.
- Sistem harus dapat menampilkan **informasi penugasan perkuliahan** yang diberikan oleh dosen pengajar. Informasi meliputi Matuliah, tugas – tugas perkuliahan, informasi nilai dari masing – masing tugas tersebut.
- Sistem harus dapat melakukan proses **rekapitulasi nilai** tugas mahasiswa per masing – masing dosen pengajar.

3.2 Membuat *Domain Model*

Pada tahap ini berfungsi untuk menemukan objek yang digambarkan dalam bentuk class yang menjadi representasi dari benda dan konsep dalam kehidupan nyata, dan dibutuhkan dalam perangkat lunak. Dari ringkasan *functional requirement* sebelumnya dapat digambarkan melalui diagram *domain model* seperti tercantum pada gambar 3.

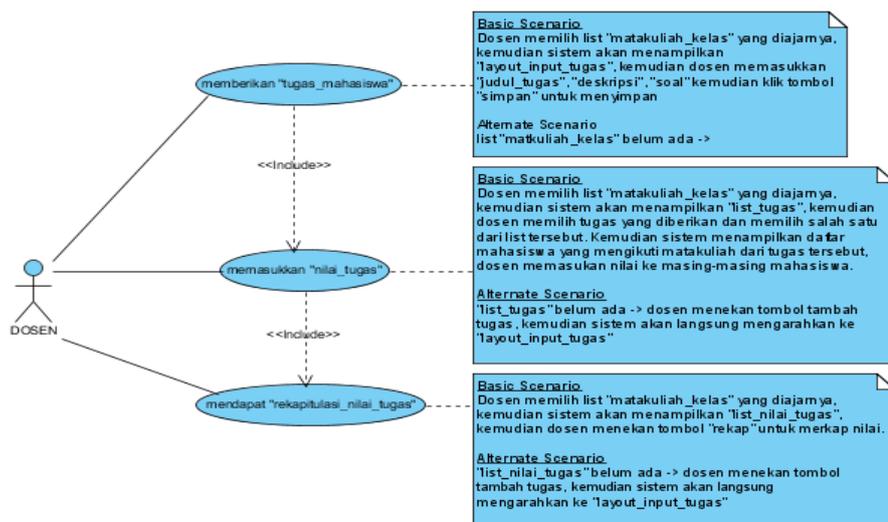


Gambar 3. *Domain Model* awal

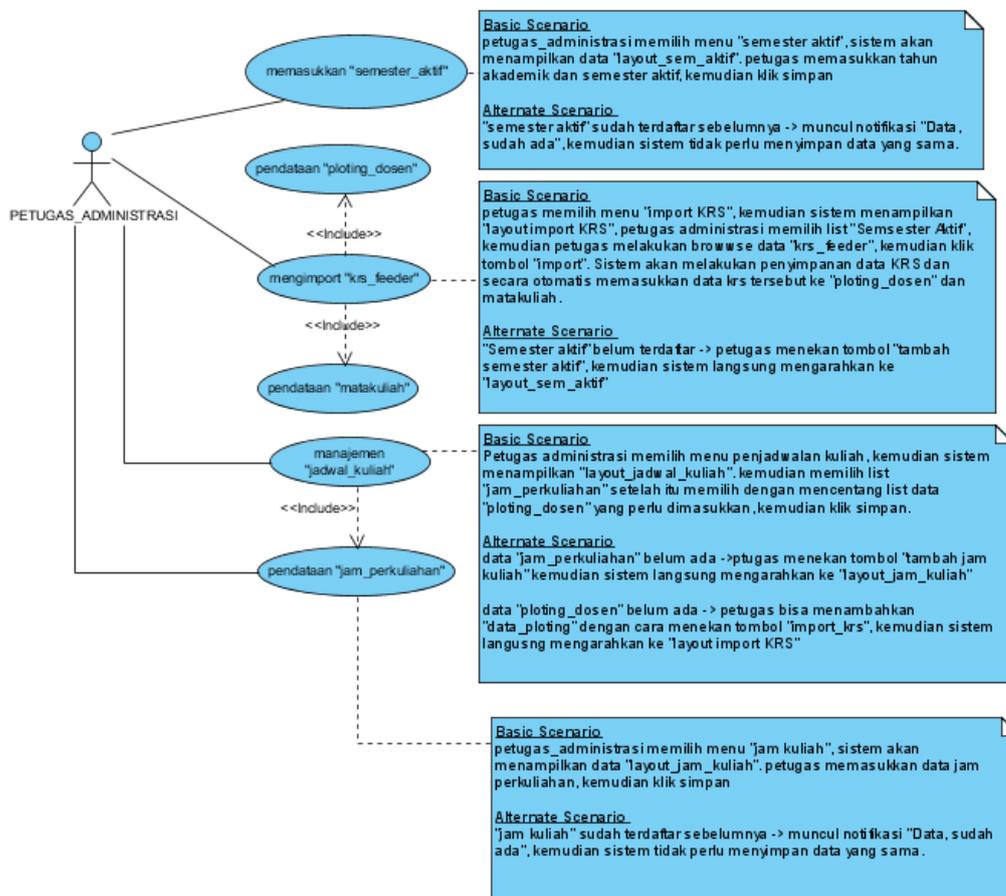
Dari gambar 3 tersebut menunjukkan hubungan antar masing objek maupun konsep yang dibutuhkan dalam aplikasi informasi jadwal dan tugas berbasis android.

3.3 Membuat *Use Case Behavioral Requirement*

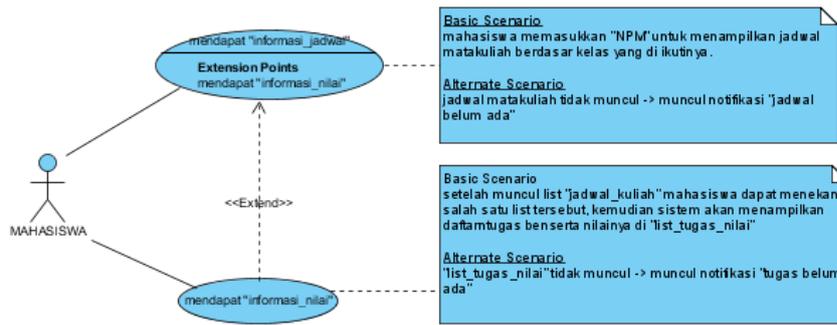
Diagram *Use case* di sini mendefinisikan *behavioral requirements* berdasarkan *functional requirement* (dan sumber lainnya) yang telah dibuat. Penggambaran *behavioral requirement* digambarkan per *actor* yang terlibat. Adapun diagram tersebut tercantum sebagai berikut :



Gambar 4. Use Case Aktor Dosen



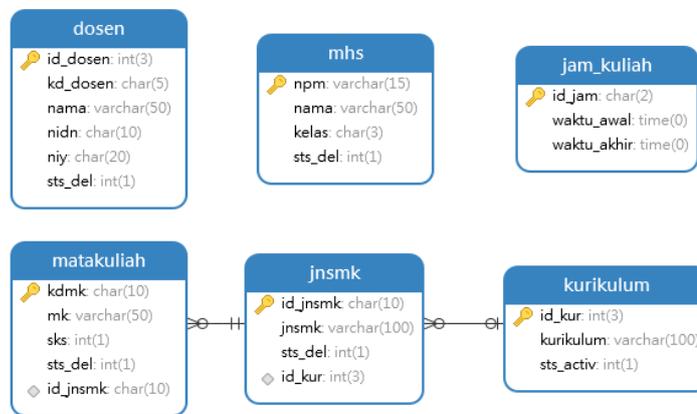
Gambar 5. Use Case Aktor Petugas_administrasi



Gambar 6. Use Case Aktor Mahasiswa

3.4 Perancangan Model Database Awal

Pada tahap ini dilakukan perancangan database berdasar *use case behavioral requirements* sebelumnya.



Gambar 7. Model Database Awal

3.5 Melakukan Requirement Review

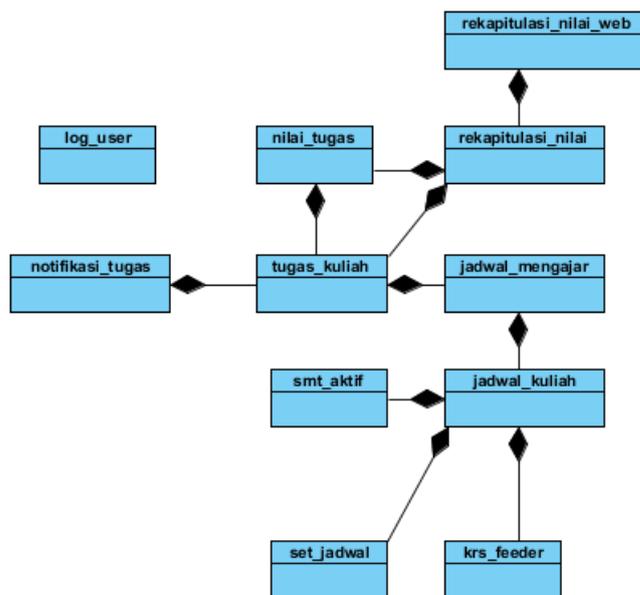
Requirement review merupakan kegiatan memastikan bahwa *use case & domain model* telah dibuat dengan baik. Pada kegiatan ini melibatkan semua aktor pengguna sistem untuk memastikan bahwa *use case (behavioral requirement)* dan *functional requirement* sesuai dengan yang diharapkan. Pada tahap ini menghasilkan keputusan untuk melakukan perubahan *functional requirement, domain model* dan *use case behavioral requirement*. Adapun hasil perubahan tersebut adalah :

a) *Functional Requirement*

- 1) Mahasiswa perlu mendapatkan informasi melalui aplikasi android tentang ruang, dosen, matakuliah yg diadakan dan jam (disebut “**jadwal kuliah**”) yang telah diprogramkan KRS dari SIAKAD [filter by kelas]
- 2) Mahasiswa perlu mendapat informasi tentang tugas dan status nilai (disebut “**tugas kuliah**”) pada masing – masing matakuliah yang dijadwalkan melalui aplikasi android
- 3) Mahasiswa perlu mendapatkan notifikasi (disebut “**notifikasi tugas**”) jika ada tugas baru setiap matakuliah melalui aplikasi android
- 4) Dosen perlu mendapatkan informasi “**jadwal mengajar**” yang akan diajarnya. Informasi terdiri dari : ruang, dosen, matakuliah yg diajarkan dan jam melalui aplikasi android [filter by dosen]
- 5) Dosen perlu dapat melakukan pencatatan penugasan pada mata kuliah yang diampunya pada masing – masing kelas (disebut “**tugas mhs**”) melalui aplikasi android
- 6) Dosen perlu dapat memasukkan nilai masing – masing tugas mahasiswa yang telah diberikan (disebut “**nilai tugas**”) melalui aplikasi android
- 7) Dosen perlu mendapat informasi / rekapitulasi dari semua nilai tugas yang diberikan (disebut “**rekapitulasi nilai**”) melalui aplikasi android
- 8) Untuk mendukung fitur – fitur tersebut di atas petugas administrasi prodi membutuhkan sistem berbasis web untuk dapat memasukkan data KRS mahasiswa (disebut “**krs feeder**”) dan seting

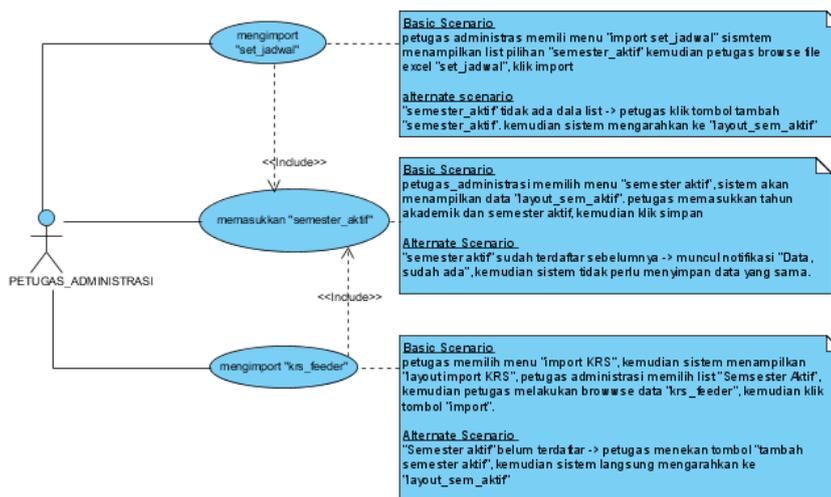
- ruang jadwal kuliah (disebut “set_jadwal”) secara serentak /import xls. (didapatkan data krs_feeder & Setting_ruang_dan_jadwal berupa file xls → diperoleh pada 3 September 2018)
- 9) Petugas administrasi prodi perlu mendapat informasi / rekapitulasi dari semua nilai tugas yang diberikan melalui aplikasi berbasis web (disebut “rekapitulasi_nilai_web”).
 - 10) Diperlukan fitur login pada aplikasi berbasis web maupun android dengan menggunakan data user (disebut “log_user”)
 - 11) Sistem harus dapat melakukan pengaturan semester aktif (disebut “smt_aktif”) dari data krs_feeder dan Setting_ruang_dan_jadwal

b) Domain Model

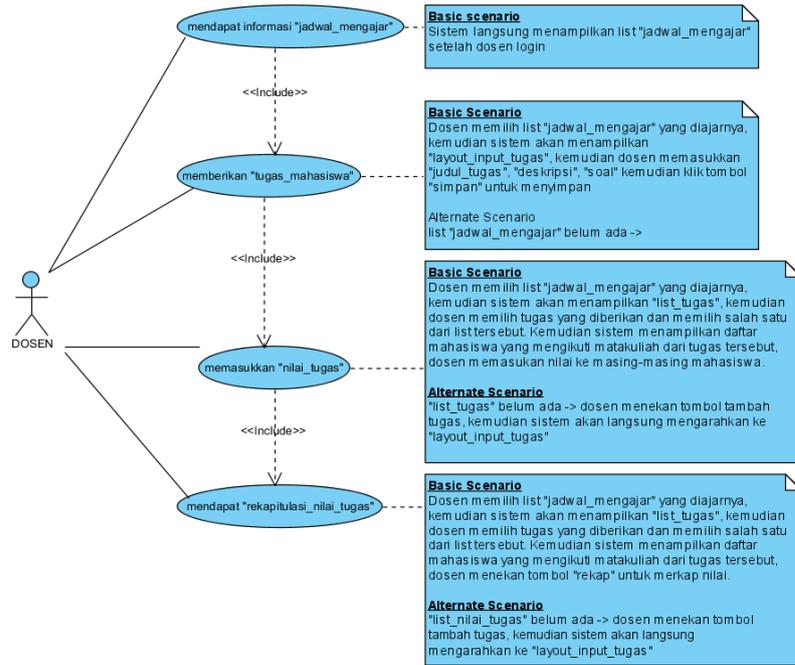


Gambar 8. Perubahan Domain Model

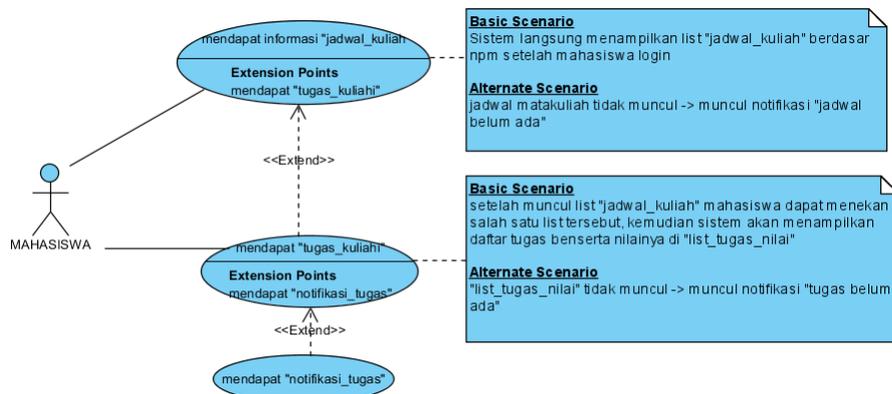
c) Use case behavioral requirement



Gambar 9. Perubahan Use case Aktor Petugas_administrasi



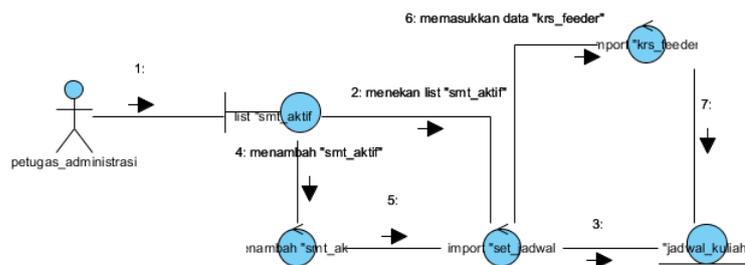
Gambar 10. Perubahan Use Case Aktor Dosen



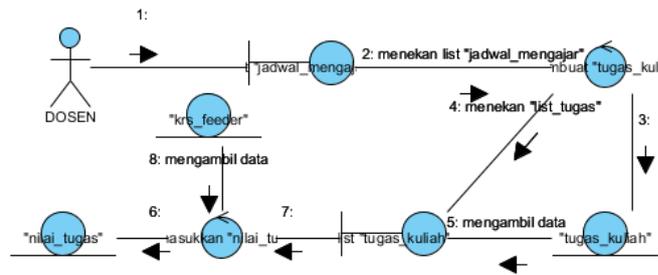
Gambar 11. Perubahan Use Case Aktor Mahasiswa

3.6 Melakukan Robustness Analysis

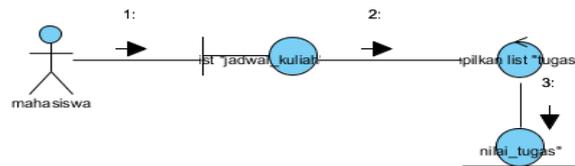
Setelah proses *requirement review* maka perlu didokumentasikan ke dalam diagram *Robustness Analysis* sebagai berikut :



Gambar 12. Robustness Analysis Aktor Petugas_administrasi



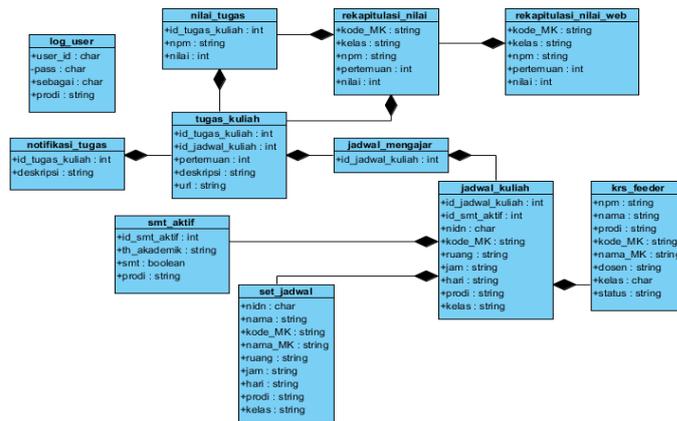
Gambar 13. Robustness Analysis Aktor Dosen



Gambar 14. Robustness Analysis Aktor Mahasiswa

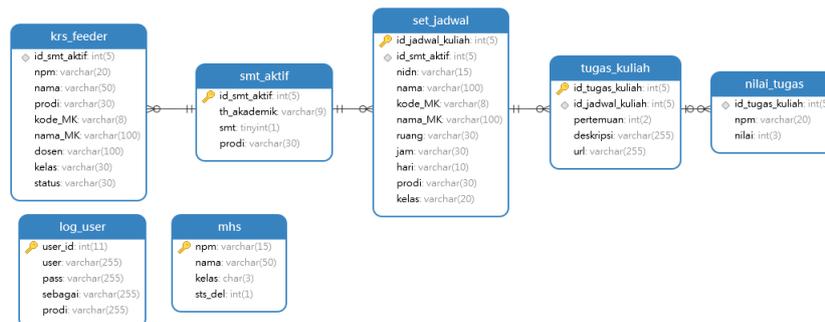
3.7 Melengkapi Domain Model

Pada tahap ini merupakan hasil dari *Robustness Analysis* sehingga perlu dilakukan perubahan dan mengisi *domain model* dengan atribut, seperti yang terlihat pada gambar 15.



Gambar 15. Domain Model Dengan Atribut

Dari domain model pada gambar 15 tersebut maka dapat digunakan sebagai dasar untuk melakukan perubahan model database yang digunakan, adapun perubahan model database yang telah dilakukan adalah sebagai berikut :



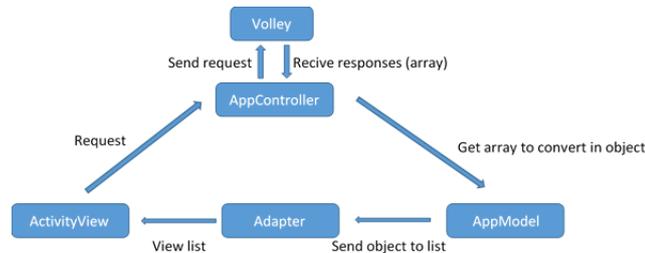
Gambar 16. Perubahan model database

3.8 Melakukan Preliminary Design Review

Kembali lagi seluruh tim melakukan *review* dan memastikan bahwa semua yang telah dibuat sesuai dengan requirement. Ini adalah langkah terakhir dimana pelanggan (*stackholder*) terlibat. Hal ini karena langkah berikutnya melibatkan proses technical.

3.9 Menentukan Technical Architecture (framework yang akan digunakan)

Adapun arsitektur *framework* secara teknis yang akan digunakan untuk membuat perangkat lunak aplikasi informasi jadwal dan tugas kuliah terdapat pada gambar 17.

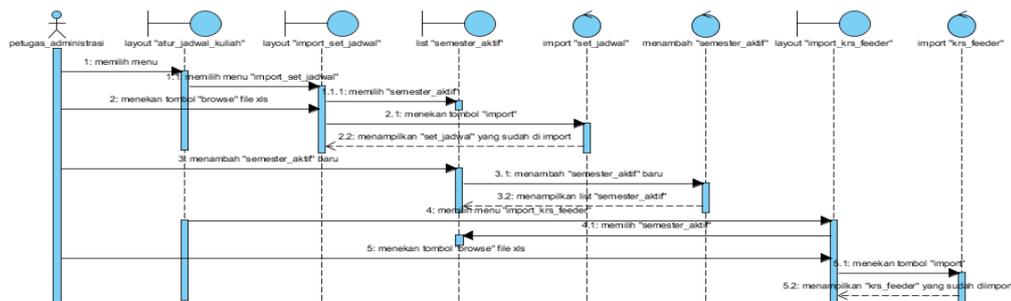


Gambar 17. Framework Technical Arschitecture

3.10 Membuat Sequence Diagram

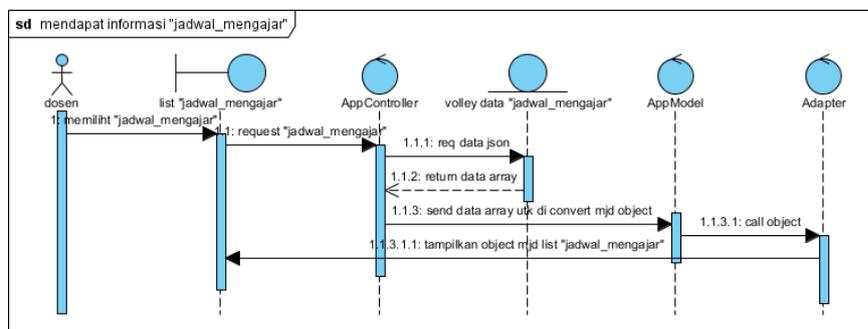
Pada tahap ini menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu yang tercantum pada gambar berikut :

a) Sequence diagram petugas_administrasi

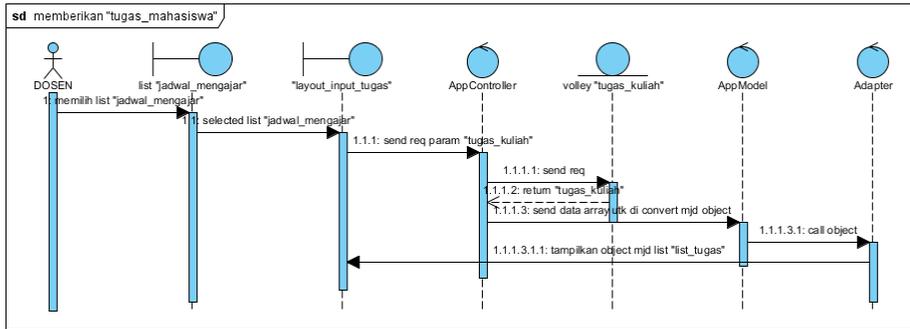


Gambar 18. Sequence Diagram Petugas_administrasi

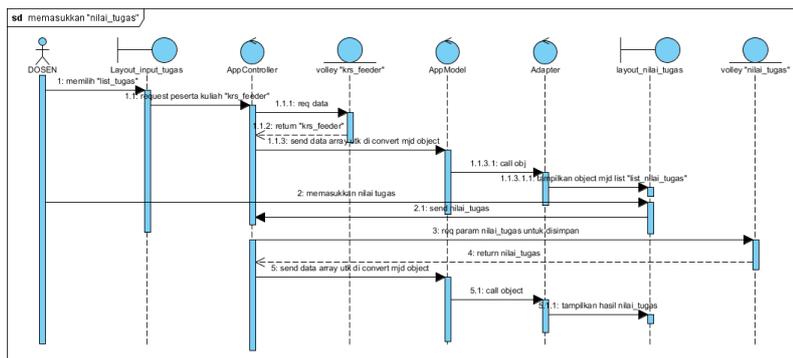
b) Sequence diagram dosen



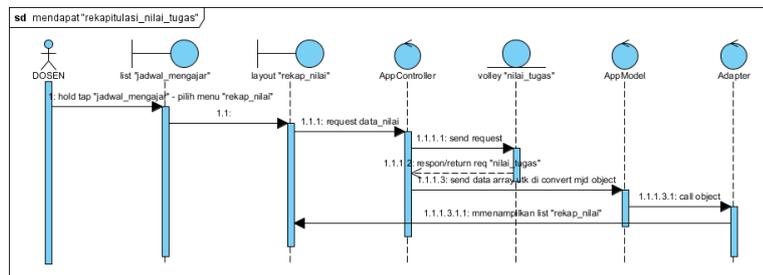
Gambar 19. Sequence Diagram Dosen Case Mendapat Informasi Jadwal Mengajar



Gambar 20. Sequence Diagram Dosen Case Memberikan Tugas Mahasiswa

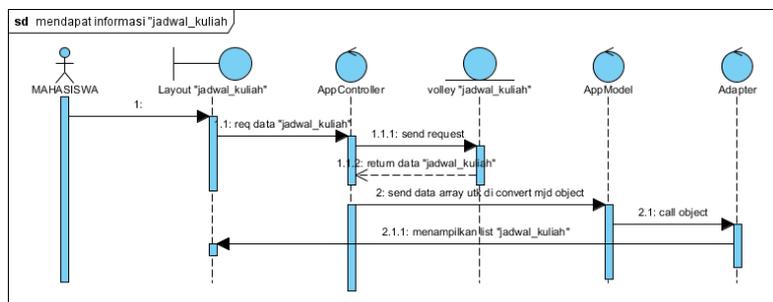


Gambar 21. Sequence Diagram Dosen Case Memasukkan Nilai Tugas

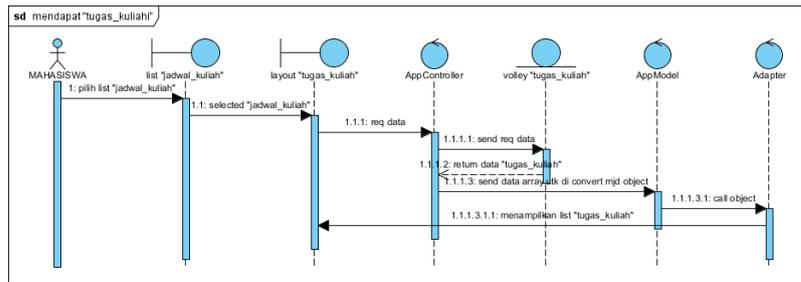


Gambar 22. Sequence Diagram Dosen Case Mendapat Rekapitulasi Nilai Tugas

c) Sequence diagram mahasiswa



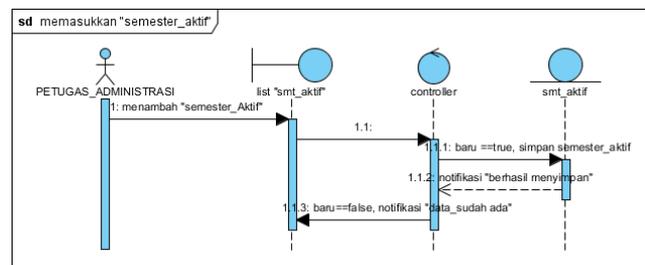
Gambar 23. Sequence Diagram Mahasiswa Case Mendapat Informasi Jadwal Kuliah



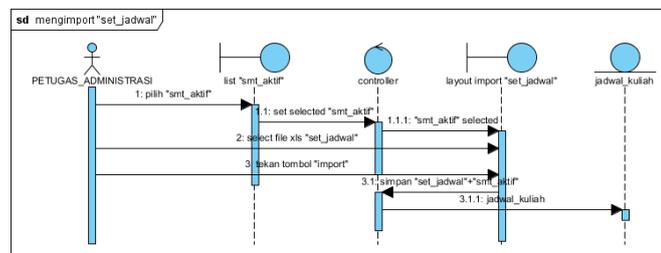
Gambar 24. Sequence Diagram Mahasiswa Case Mendapat Tugas Kuliah

3.11 Melakukan Critical Design Review

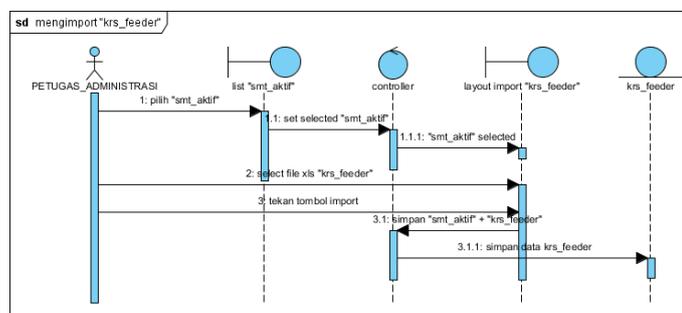
Tahap ini merupakan tahap akhir sebelum pembuatan perangkat lunak. Pada tahap ini kembali melakukan review untuk memastikan bahwa tidak ada yang kurang pada *sequence diagram*. Memastikan bahwa setiap *class* yang ada telah memiliki atribut dan operasi yang didefinisikan secara lengkap (memiliki nama, tipe data, parameter, dan sebagainya). Pada tahap ini ditemukan kesalahan dalam pembuatan *sequence diagram* untuk aktor petugas administrasi, dimana *sequence diagram* seharusnya menjabarkan setiap *case* yang dimiliki aktor petugas administrasi. Adapun hasil perbaikan dari *sequence diagram* tersebut sebagai berikut :



Gambar 25. Perubahan Sequence Diagram Petugas Administrasi Case Memasukkan Semester Aktif



Gambar 26. Perubahan Sequence Diagram Petugas Administrasi Case Mengimport Set_Jadwal



Gambar 27. Perubahan Sequence Diagram Petugas Administrasi Case Mengimport Krs_Aktif

4. KESIMPULAN

Dari hasil penelitian yang telah dilakukan maka dapat diambil kesimpulan guna menjawab permasalahan yang tercantum pada bagian pendahuluan, diantaranya adalah :

- a) Proses pembuatan rancangan aplikasi informasi jadwal dan tugas berbasis android dapat dilakukan dengan baik dengan menggunakan *ICONIX Process*, dimana dalam menggunakan *ICONIX Process* terdapat beberapa *review* untuk mengkoreksi kekurangan – kekurangan, sehingga peneliti dapat segera mengisi atau merevisi kekurangan tersebut dengan cepat.
- b) Dari hasil diagram perancangan yang telah dibuat, maka terdapat beberapa diagram yang dapat memberikan gambaran atau panduan teknis dalam proses pembuatan aplikasi informasi jadwal dan tugas berbasis android, beberapa rancangan tersebut adalah :
 - 1) *Functional Analysis* pada sub bab 3.5 poin 'a' dan gambar rancangan use case mulai gambar 9, 10 dan 11, yang dapat mewakili suatu kebutuhan fungsional / fitur / kemampuan perangkat lunak untuk bisa melakukan aktivitas – aktivitas yang disebutkan.
 - 2) *Domain model* (gambar 15) yang mewakili objek, kelas dan konsep yang dibutuhkan dalam proses pembuatan aplikasi informasi jadwal dan tugas berbasis android.
 - 3) Model database (gambar 16) yang dapat digunakan sebagai dasar basis data yang dibutuhkan dalam proses pembuatan aplikasi informasi jadwal dan tugas berbasis android.
 - 4) *Framework Technical Architecture* (gambar 17) yang dapat digunakan sebagai struktur dasar teknis kode dalam proses pembuatan aplikasi informasi jadwal dan tugas berbasis android.
 - 5) *Sequence Diagram* (gambar 19 – 27) yang dapat berguna sebagai dasar model interaksi pengguna dengan perangkat lunak yang akan dibuat, serta juga sebagai panduan dalam menyusun kode algoritma pemrograman dalam proses pembuatan aplikasi informasi jadwal dan tugas berbasis android.

DAFTAR PUSTAKA

- [1] G. Booch, *Object Solutions: Managing the Object-oriented Project*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 1995.
- [2] K. Hamilton and R. Miles, *Learning UML 2.0*. O'Reilly, 2006.
- [3] S. Dharwiyanti and R. S. Wahono, "Pengantar Unified Modeling Language (UML)," *IlmuKomputer.com*, pp. 1–13, 2003.
- [4] D. Rosenberg and K. Scott, *Applying Use Case Driven Object Modeling with UML: An Annotated e-Commerce Example*, First. Upper Saddle River: Addison Wesley, 2001.
- [5] D. Rosenberg and M. Stephens, *Use Case Driven Object Modeling with UML Theory and Practice*. New York: Springer, 2007.
- [6] Romeo; Hilmy Nur Ramdhani; Dewiyani Sunarto;, "Rancang Bangun Aplikasi Pelayanan Dan Rekam Medis Calon Tenaga Kerja Indonesia Pada AL-HUDA MEDICAL CENTER," *JSIKA*, vol. 05, no. 09, pp. 3–8, 2016.
- [7] K. D. Dharmawan and W. S. Sari, "Pembangunan Situs Web Menggunakan Metode Iconix Process Untuk Strategi Penjualan Komputer Pada Cv . Citra Mandiri Semarang," *J. JOINS Udinus (Journal Inf. Syst.*, vol. 1, no. 2, pp. 193–201, 2016.
- [8] A. Setiyanto, F. Samopa, and Alwi, "Pembuatan Sistem Informasi Cuti pada Kantor Pelayanan Perbendaharaan Negara dengan Menggunakan PHP dan MySQL," *Tek. Pomits*, vol. 2, no. 2, pp. 381–384, 2013.
- [9] B. Anindito and F. Samopa, "Perancangan dan analisis biaya manfaat perbaikan sistem penilaian kinerja guru dan karyawan smp mujahidin," in *Prosiding Seminar Nasional Manajemen Teknologi XXIV*, 2016, no. January, p. C-8-1.
- [10] M. R. A. S. A. Suwondo, "RANCANG BANGUN SISTEM INFORMASI UKM SEBAGAI SARANA PERDAGANGAN BATIK DENGAN E-MARKETPLACE DI PEKALONGAN," in *SENTRINOV*, 2016, vol. 01, pp. 561–577.
- [11] S. A. F. Istifani;, "Rancang Bangun Aplikasi Koperasi Simpan Pinjam dengan Metode Viewpoint Oriented Requirement Definition," *J. SISFO*, vol. 07, no. 02, pp. 165–179, 2018.
- [12] J. O. Wisono, D. Pramono, and M. C. Saputra, "Analisis dan Perancangan Sistem Informasi Pemesanan Customized Jersey pada Injers Malang Berbasis Web," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 3, no. 4, pp. 3677–3686, 2019.
- [13] I. Sufyan, A. L. Fernandes, A. Sumantri, P. Studi, T. Informatika, and I. Process, "Penerapan Teknologi Augmented Reality Untuk Pengenalan Produk Berbasis Android Di PT Pipa Mas Putih," *J. Responsive Tek. Inform.*, vol. 2, no. 2, p. 135, 2018.