

# *Fair Assessment* pada Kolaborasi Pemodelan Persyaratan Perangkat Lunak UML

Danang Wahyu Utomo<sup>1</sup>, Egia Rosi Subhiyanto<sup>2</sup>, Defri Kurniawan<sup>3</sup>

**Abstract**—In software engineering education, students are expected to be able to collaborate with other student in team project. Based on the RPS and Software Engineering syllabus, student must be able to create SRS document in group assignment. In general, lecturers assess students based on team cohesiveness, student accuracy in answering questions, and final result of SRS documents. The weakness of the assessment model is that the lecturer cannot know student who are really active and contribute to the team. Student will get the same marks. There is no difference in marks between active and passive students. The impact, active student can become passive, because the result obtained in the presentation of final assignment is no different from the other student. It can make student contribution decrease. Student are reluctant to group in completing their final assignments. This paper aims to propose a fair assessment approach to provide an assessment that is appropriate to the composition of group assignment. Fair assessment consists of assignment component, diagram modeling, and final project. In addition, re-implementation was carried out on the previous collaborative tool by replacing the assessment model and adding some features such as friend search menu and real time discussion. The evaluation involved 55 students majority in Software Engineering to test fair assessment approach and new features of collaboration tool. The result shows that students agree that a fair assessment approach is appropriate to be used in supporting collaborative software engineering education, especially UML diagram modeling.

**Intisari**—Proses pengembangan perangkat lunak tidak dapat dilakukan secara individu. Masing-masing individu harus berkolaborasi untuk mencapai hasil maksimal yaitu perangkat lunak yang sesuai dengan kebutuhan. Dalam pendidikan rekayasa perangkat lunak, mahasiswa diharapkan mampu berkolaborasi dengan mahasiswa lainnya dalam kerja tim. Berdasarkan RPS dan silabus Rekayasa Perangkat Lunak, mahasiswa harus mampu membuat dokumen SKPL dalam tugas kelompok. Namun, di sesi presentasi tugas akhir, masih banyak mahasiswa yang melakukan kesalahan dalam penyusunan SKPL. Mahasiswa masih melakukan kesalahan dalam menggunakan komponen diagram. Permasalahan lainnya adalah penilaian yang dilakukan dosen umumnya hanya melihat garis besar hasil kerja kelompok. Mahasiswa dalam tim tersebut akan mendapatkan nilai yang sama. Tidak ada perbedaan nilai antara mahasiswa yang aktif dan pasif. Dampaknya, mahasiswa yang aktif dapat menjadi pasif, karena pada tugas akhir nilai yang diperoleh tidak berbeda dengan anggota satu tim. Makalah ini mengusulkan pendekatan penilaian adil (*fair assessment*) untuk memberikan penilaian yang sesuai dengan komposisi tugas anggota dalam tim. Penilaian adil terdiri atas komponen

penugasan, pemodelan *use case diagram*, dan proyek akhir. Selain itu, implementasi ulang dilakukan pada alat kolaborasi sebelumnya dengan mengganti model penilaian dan melakukan penambahan fitur seperti pencarian teman dan diskusi secara *real time*. Evaluasi melibatkan 55 mahasiswa jurusan Rekayasa Perangkat Lunak untuk menguji pendekatan penilaian dan fitur baru alat kolaborasi. Hasil menunjukkan bahwa mahasiswa setuju alat kolaborasi dengan penilaian adil layak digunakan dalam menunjang pembelajaran kolaborasi rekayasa perangkat lunak, khususnya pada pemodelan diagram UML.

**Kata Kunci**— *fair assessment*, rekayasa perangkat lunak, UML, pembelajaran kolaborasi, pendidikan.

## I. PENDAHULUAN

Pada level sarjana (Strata-1), Rekayasa Perangkat Lunak (RPL) dijadikan sebagai mata kuliah wajib yang digunakan sebagai acuan dalam pembangunan perangkat lunak. Mahasiswa diharapkan dapat memahami semua tahapan pengembangan, metode, dan alat yang digunakan dalam membangun sebuah perangkat lunak. Khususnya pada proyek pengembangan perangkat lunak, mahasiswa dilatih untuk fokus mempelajari pengetahuannya, aturannya, sistematis pengembangan, dan cara menerapkan metode dan alat pada kasus atau proyek yang berbeda [1], [2].

Pengembangan perangkat lunak perlu memperhatikan tahapan-tahapan di dalam rangkaian siklus pengembangan perangkat lunak, biasanya disebut *Software Development Life Cycle* (SDLC). SDLC memiliki tahapan sebagai berikut: analisis, desain, implementasi, pengujian, dan perbaikan (*maintenance*) [3]. Mata kuliah RPL menyediakan materi tentang analisis kebutuhan fungsional dan nonfungsional, pemodelan menggunakan diagram, dan metode pengujian perangkat lunak. Pada mata kuliah RPL Dasar, mahasiswa mempelajari pemodelan terstruktur, yaitu pemodelan menggunakan *Data Flow Diagram* (DFD), *Entity Relationship Diagram* (ERD), dan *State Transition Diagram* (STD). Mahasiswa mempelajari cara memodelkan perilaku pengguna melalui diagram-diagram tersebut, cara menghubungkan satu aktivitas dengan aktivitas lain, dan cara memberikan derajat kardinalitas pada masing-masing komponen diagram. Pada mata kuliah RPL Lanjut, mahasiswa baru belajar mengenai *Unified Modeling Language* (UML). UML digunakan karena pada saat ini mayoritas pemrograman lebih mengarah ke *object oriented programming* (OOP). UML merupakan model standar yang digunakan untuk menggambarkan artefak dari perangkat lunak. Selain itu, UML juga dapat digunakan untuk menentukan dan membangun artefak dari perangkat lunak tersebut [4]. Artefak terdiri atas objek, kelas, dan paket yang ditujukan dalam membuat aplikasi berbasis OOP. Pada mata kuliah RPL

<sup>1,2,3</sup>Dosen, Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro, Jl. Imam Bonjol 207 Semarang, 50131, Indonesia (tlp: 024-3517261; fax: 024-3569684; e-mail: danang.wu@dsn.dinus.ac.id)

Lanjut, mahasiswa harus mampu menggunakan komponen, *relationship*, dan struktur dari diagram UML. Permasalahan utamanya adalah mahasiswa tidak memahami sepenuhnya diagram UML seperti komponen diagram, *relationship*, dan cara penggunaannya dalam setiap kasus pemodelan. Hal ini dikarenakan pada semester sebelumnya, mahasiswa belajar pemodelan analisis terstruktur. Hal ini dapat menyebabkan mahasiswa tidak dapat menerapkan pengetahuannya pada proyek nyata pengembangan perangkat lunak di sebuah industri [5].

Berdasarkan Rencana Pembelajaran Semester (RPS) dan silabus mata kuliah RPL Lanjut, tugas proyek akhir adalah mengembangkan aplikasi mini seperti aplikasi berbasis web, *mobile*, maupun desktop dengan dilengkapi dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL). SKPL merupakan dokumen berisi analisis dari kebutuhan fungsional, pemodelan menggunakan UML, dan gambaran antarmuka (*interface*). Tujuan dari tugas akhir ini adalah melatih mahasiswa untuk menerapkan pengetahuan RPL pada kasus di dunia nyata saat ini. Mahasiswa juga diharapkan mampu merancang dokumen SKPL sebagai acuan dalam pengembangan proyek perangkat lunak. Selain itu, juga untuk melatih kemampuan kolaborasi mahasiswa dalam tim dengan latar belakang anggota yang berbeda-beda [6].

Seringkali implementasi program berbeda jauh dengan SPKL yang telah dirancang. Bahkan perilaku pengguna yang digambarkan pada diagram UML berbeda jauh dengan implementasinya. Permasalahannya adalah masih banyak mahasiswa yang bingung menggunakan komponen pada diagram yang digunakan. Sebagai contoh, mahasiswa tidak dapat membedakan komponen *include* dan *extend* pada *use case diagram*. Permasalahan lainnya, mahasiswa (umumnya senior) menyerahkan komponen tugasnya kepada anggota lain yang dirasa paham dan mampu menyelesaikannya. Alasannya agar nilai tugas kelompok mendapatkan hasil yang lebih baik jika dikerjakan oleh siswa yang rajin. Pada sisi pengajar, umumnya dosen hanya menilai tugas kelompok mahasiswa berdasarkan kekompakan menjawab dalam sesi presentasi, tingkat kebenaran jawaban dalam sesi tanya jawab dengan tim lain, dan kesesuaian antara SKPL dengan implementasi program. Kelemahannya, dosen tidak dapat mengetahui siswa yang benar-benar mengerjakan komposisi tugasnya dan siswa yang masih benar-benar bingung mengenai materi yang telah diajarkan. Model penilaian seperti ini dirasa tidak adil bagi mahasiswa yang mendapatkan semua komposisi tugas kelompok dari mahasiswa yang malas. Perlu adanya pendekatan penilaian yang disesuaikan dengan level mahasiswa, tingkat komposisi tugas, dan pembagian tugas yang merata terhadap tim [7].

Metode, model, dan pendekatan penilaian telah banyak diusulkan untuk memberikan penilaian yang sesuai dengan metode pembelajaran yang digunakan. Mengusulkan alat penilaian untuk melakukan pengukuran uji kompetensi siswa pada pendidikan tinggi [8], mengembangkan alat analitik pembelajaran guna menilai kompetensi kerja tim [9], *e-assessment* pada pembelajaran online [10], dan *formative assessment* pada *e-portfolio* [11] adalah beberapa contohnya.

Metode, model, maupun pendekatan yang diusulkan hanya fokus pada evaluasi pembelajaran yang digunakan, tetapi belum banyak diterapkan pada alat kolaborasi siswa. Alat kolaborasi yang diusulkan juga masih berfokus pada cara mengatur kolaborasi antar pengguna dalam sistem. Pada pendidikan RPL, alat kolaborasi hanya fokus pada cara antar individu dapat meningkatkan kemampuan komunikasi dan diskusi efektif. Mengusulkan alat kolaborasi guna menunjang aktivitas pengembangan perangkat lunak [12], kolaborasi siswa senior pada pengembangan proyek [2], mengusulkan alat kolaborasi membantu pembelajaran PBL dan menilai mahasiswa [13].

Pada makalah ini, desain ulang dilakukan pada alat kolaborasi yang sebelumnya telah diusulkan [6]. Pada rancangan sebelumnya, alat kolaborasi hanya fokus pada penilaian individu dan kerja tim. Perlu adanya pendekatan penilaian guna memberikan penilaian yang adil bagi siswa dalam menjalankan kerja tim pada proyek perangkat lunak. Rancangan saat ini berfokus pada pembelajaran dasar *use case diagram*. Mahasiswa dapat berbagi materi dengan mahasiswa lainnya, berdiskusi, dan membuat kasus baru yang berkaitan dengan *use case diagram*. Bagi pengajar, pendekatan penilaian yang diusulkan dapat memberikan nilai sesuai dengan tingkat kesulitan masing-masing tugas.

## II. PENDEKATAN PENILAIAN

*Learning outcome* adalah pencapaian pembelajaran siswa pada sebuah kursus atau pembelajaran formal di sebuah institusi. Capaian pembelajaran dalam bentuk aktivitas pembelajaran siswa selama proses belajar dan hasil akhir sebagai tolok ukur pemahaman dan evaluasi siswa terhadap materi. Penilaian *learning outcome* disesuaikan dengan integrasi antara *knowledge*, *skill*, dan *attitude* [8]. Pendekatan penilaian pada mata kuliah RPL diharapkan dapat meningkatkan kemampuan siswa dalam hal pengembangan sistem yang dimulai dari pemahaman analisis kebutuhan, pemodelan, implementasi, dan metode pengujian. Pendekatan penilaian harus disesuaikan dengan pembelajaran di kelas seperti keaktifan siswa di kelas dan pembelajaran kolaborasi [14].

Pada makalah ini diusulkan pendekatan penilaian baru bagi mahasiswa jurusan RPL yang meliputi komponen penugasan atau kuis, kontribusi pemodelan diagram, dan komponen proyek akhir yang melibatkan dokumen SKPL. Semua komponen penilaian ini diterapkan pada alat kolaborasi dengan fokus pembelajaran adalah *use case diagram*.

### A. Komponen Penugasan atau Kuis

Pada pembelajaran pemodelan menggunakan UML, mahasiswa belajar tentang komponen-komponen yang digunakan pada masing-masing diagram, cara menghubungkan antar komponen, dan cara menggambarkan kebutuhan fungsional pada diagram UML. Perlu adanya intensitas latihan agar mahasiswa dapat memahami dan menggunakan diagram UML sesuai dengan studi kasus. Dosen sebagai instruktur harus berperan aktif dalam memberikan variasi contoh kasus pada tiap pertemuan di kelas. Komponen penugasan digunakan

sebagai komponen penilaian pelatihan mahasiswa pada tiap minggunya. Dosen memberikan studi kasus untuk tiap minggunya kepada tim. Berikut adalah aturan umum penugasan.

1. Sifat penugasan adalah tugas kelompok.
2. Mahasiswa wajib membuat tim yang terdiri atas empat anggota.
3. Dosen menyediakan studi kasus untuk tiap tim, yang untuk penugasannya dibagi sesuai dengan jumlah tim.
4. Tugas dikirim ke ketua tim.
5. Ketua tim wajib mengirimkan deskripsi tugas ke masing-masing anggotanya.
6. Tiap anggota wajib mengirimkan bagian tugasnya ke ketua tim dan dosen.

Aturan-aturan tersebut bertujuan untuk meningkatkan tanggung jawab mahasiswa dalam tim, terutama mahasiswa malas yang sering menyerahkan komposisi tugasnya kepada mahasiswa lainnya. Dengan adanya alat kolaborasi ini, mahasiswa yang malas tidak dapat menyerahkan komposisi tugasnya kepada mahasiswa lain, karena masing-masing anggota memiliki *deadline* pengumpulan tugas. Selain itu, hal ini memudahkan dosen untuk memantau perkembangan pemahaman mahasiswa terkait materi yang diberikan tiap minggunya. Dosen dapat mengetahui kemajuan pengumpulan komposisi tugas dari masing-masing mahasiswa meskipun tanpa tatap muka. Model penilaian berasal dari bobot pada masing-masing subtask. Penilaiannya merupakan penjumlahan dari nilai yang diperoleh mahasiswa berdasarkan level tugas yang dipilih.

$$\text{nilai tugas} = \sum_{i=0}^n x_i * W \quad (1)$$

Pada (1),  $x_i$  adalah nilai mahasiswa dan  $W$  adalah pembobotan dari komposisi tugas yang telah dibagi oleh ketua tim. Alokasi pembobotan didasarkan pada tingkat kesulitan dari komposisi tugas.

TABEL I  
CONTOH PEMBOBOTAN PENUGASAN

No	Deskripsi Tugas	Bobot
1	Analisis studi kasus	30%
2	Kebutuhan fungsional	20%
3	Skenario <i>use case diagram</i>	25%
4	<i>Use case diagram</i>	25%

Tabel I menampilkan contoh pembobotan pada masing-masing deskripsi tugas. Misalnya, mahasiswa A memilih deskripsi tugas 1, maka nilai yang diperoleh mahasiswa A dikalikan dengan bobot 30%. Nilai ini menjadi bagian dari nilai individu mahasiswa.

### B. Pemodelan UML Diagram

Komponen pemodelan diagram bertujuan untuk memonitor keaktifan mahasiswa selama kerja tim, khususnya kontribusi dalam pemodelan diagram UML. Pada alat kolaborasi ini, mahasiswa dituntut untuk aktif dalam membuat pemodelan UML. Alat kolaborasi menyediakan program khusus pembelajaran *use case diagram*, yang dapat digunakan

sebagai media kontribusi bagi mahasiswa. Selain *use case diagram*, mahasiswa dapat berkontribusi dengan memberikan analisis, referensi penyelesaian masalah, dan memberikan skenario penyelesaian melalui diskusi interaktif.

$$\frac{\text{kontribusi anggota}}{\text{jumlah keseluruhan kontribusi}} \quad (2)$$

Pada (2), model penilaiannya menggunakan persentase dari kontribusi mahasiswa. Penilaian ini dihitung tiap minggunya. Segala jenis kontribusi mahasiswa dibandingkan dengan kontribusi anggota lainnya dalam tim. Adanya formula ini mendorong mahasiswa yang malas mengerjakan tugas dan mencoba berkontribusi lebih banyak untuk mendapatkan tambahan nilai yang tinggi. Berikut adalah aturan umum kontribusi pemodelan *UML diagram*.

1. Mahasiswa dilarang melakukan plagiarisme (kontribusi yang diberikan sama seperti anggota lainnya).
2. Kontribusi harus berkaitan dengan topik yang diberikan oleh dosen.
3. Segala jenis pertanyaan dalam forum tidak termasuk dalam kategori kontribusi.

Dengan adanya aturan tersebut, dosen memiliki peran penting dalam mengevaluasi semua jenis kontribusi mahasiswa. Dosen dapat menghapus kontribusi dari mahasiswanya jika tidak sesuai aturan yang berlaku, seperti kontribusi di luar topik, komponen diagram salah, dan referensi yang digunakan *out of date*.

### C. Komponen Proyek Akhir

Komponen proyek akhir memiliki bobot paling tinggi dari komponen penugasan dan pemodelan diagram. Pada komponen ini, komponen penilaian ini sering disebut sebagai “tugas akhir”. Tugas akhir ini berbasis proyek. Mahasiswa bekerja dalam tim membuat sebuah aplikasi sederhana. Mahasiswa dituntut untuk menerapkan metode pengembangan sistem, pemrograman, dan rencana evaluasi perangkat lunak dalam sebuah dokumen. Berikut adalah aturan umum penilaian proyek akhir.

1. Dosen menyediakan beberapa topik dan ketua tim memilih topik tersebut.
2. Tiap anggota tim wajib *login* pada alat kolaborasi. Jika salah satu anggota tidak hadir pada tugas proyek akhir, tim tersebut dianggap mengundurkan diri.
3. Tiap anggota mendapatkan satu deskripsi tugas.
4. Proyek akhir terdiri atas studi kasus, penyelesaian masalah, analisis kebutuhan fungsional, dan pemodelan *UML diagram* dengan skenario.
5. Waktu tempuh proyek akhir adalah 2 jam, tanpa toleransi keterlambatan.

Pada Tabel II, penilaian proyek akhir meliputi studi kasus, *problem solving*, kebutuhan fungsional dan nonfungsional, skenario *use case*, dan *use case diagram*. Model pengerjaannya adalah mahasiswa memilih topik yang telah disediakan oleh dosen, lalu ketua tim mulai menentukan deskripsi tugas bagi anggotanya. Pada komponen ini, mahasiswa dilatih untuk menentukan sendiri studi kasusnya, cara penyelesaiannya, macam-macam kebutuhan fungsionalnya,

dan cara menentukan perilaku pengguna melalui *use case diagram*.

TABEL II  
BOBOT PENILAIAN PROYEK AKHIR

Komponen	Bobot
Analisis Studi Kasus	24%
<i>Problem Solving</i>	16%
Kebutuhan Fungsional	22%
Skenario <i>use case</i>	15%
<i>Use case diagram</i>	23%

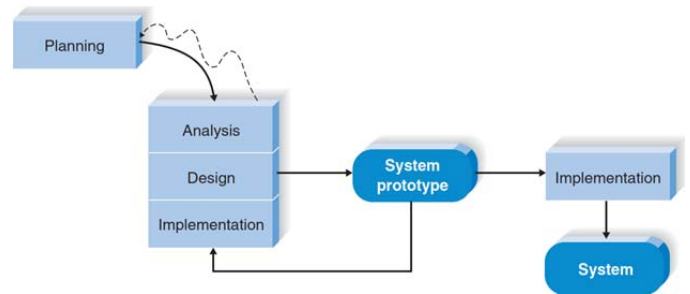
### III. METODOLOGI

Pada makalah ini, alat kolaborasi diterapkan pada kelas RPL Lanjut, yang materi pemodelannya telah menggunakan UML sebagai pemodelan dasar. Alat kolaborasi fokus pada pemahaman *use case diagram* yang meliputi komponen, *relationship*, dan pemecahan masalah pada contoh studi kasus. Mahasiswa diarahkan untuk menggunakan alat kolaborasi. Pada alat kolaborasi ini, mahasiswa diwajibkan membentuk tim berjumlah empat anggota yang komposisinya harus heterogen. Artinya, setidaknya ada satu anggota perempuan di dalam tim. Dosen menyiapkan tugas untuk sepuluh pertemuan dan dua *mini project*. Tugas diberikan tiap minggu dengan deskripsi dan bobot nilai untuk masing-masing deskripsi tugas. Tugas yang diberikan berbeda dengan tim lain, yang bertujuan melatih mahasiswa menghindari plagiarisme tugas. *Mini project* berupa pengembangan program atau aplikasi yang dilengkapi dengan dokumen SKPL. *Mini project* diberikan sebelum UTS dan sebelum UAS.

Pada pengembangan alat kolaborasi, SDLC yang digunakan adalah *prototyping model* sebagai acuan dalam membangun sistem. Gbr. 1 menunjukkan tahapan *prototyping model* melakukan tahap analisis, desain, implementasi secara bersamaan untuk mengembangkan sistem, dan mendapatkan evaluasi dan umpan balik dari pengguna secara cepat [15]. Ketika para analis dan pengguna kesulitan untuk menemukan kesepakatan terkait analisis kebutuhan fungsional dan rencana implementasi, prototipe menampilkan fitur-fitur yang ditawarkan kepada *stakeholder*. Meskipun fitur yang ditampilkan masih bersifat minimalis, tetapi dapat dijadikan sebagai evaluasi dan umpan balik secara cepat. Oleh karena itu, pengembang (*developer*) harus melakukan analisis, desain, dan implementasi ulang untuk melengkapi sistem prototipe sebelumnya, menambahkan fitur-fitur yang sebelumnya tidak ada.

Tahapan evaluasi dilakukan pada mahasiswa yang mengambil jurusan RPL dengan uji sampel 55 mahasiswa yang telah mempelajari materi pemodelan UML. Berbagai metode evaluasi banyak diusulkan untuk mendapatkan hasil kualitas yang sesuai, misalnya penggunaan angket, kuesioner dalam bentuk QUIZ, SUS, dan UEQ pada pengujian *usability e-learning* [16], dan mengusulkan metode pengujian menggunakan uji *paired sample t-test* [17]. Teknik evaluasi dilakukan dengan penyebaran kuesioner terhadap mahasiswa. Mahasiswa sebelumnya diberikan pemahaman tentang

penggunaan alat kolaborasi. Evaluasi dengan menggunakan kuesioner memudahkan para pengguna dalam mengevaluasi kualitas dari alat kolaborasi.



Gbr. 1 *System prototyping* [15].



Gbr. 2 *Use case diagram* alat kolaborasi.

### IV. ANALISIS DAN PERANCANGAN

Pada makalah ini dilakukan analisis, desain, dan implementasi ulang alat kolaborasi yaitu memperbaiki model penilaian dan melakukan penambahan fitur. Pada tahap analisis, ada penambahan kebutuhan fungsional seperti pencarian teman, *screenshot use case diagram*, dan diskusi interaktif secara *real time*. Selanjutnya, dilakukan pemodelan ulang seperti rancangan *use case diagram* dan perubahan tampilan pada manajemen tim. Pada tahap implementasi, alat kolaborasi berbasis web dibangun menggunakan bahasa pemrograman HTML, PHP, dan Javascript.

Gbr. 2 menunjukkan perilaku pengguna dalam sistem, yang menampilkan fungsi utama dalam alat kolaborasi. Secara garis besar, fungsi dalam *use case diagram* tersebut merupakan hasil dari analisis kebutuhan fungsional. Aktor utama dalam *use case diagram* adalah mahasiswa (*student*). Mahasiswa dapat membuat informasi dasar terkait nama dan email yang

digunakan sebagai data diri dalam membuat tim. Selanjutnya, mahasiswa diharapkan untuk membuat tim agar dapat mengerjakan tugas, membuat *use case diagram*, dan melihat hasil penilaian akhir. Fitur pembuatan *use case diagram* menggunakan konsep *drag and drop*. Hasil akhir dari pembuatan *use case diagram* berupa *screenshot*, yang nantinya dapat digunakan sebagai bahan kontribusi siswa pada forum diskusi.

V. HASIL DAN PEMBAHASAN

Pada tahap implementasi dilakukan pengembangan ulang alat kolaborasi, yaitu dengan mengganti model penilaian dan menambahkan fitur pencarian teman, *screenshot use case diagram*, dan diskusi interaktif secara *real time*.

Pada Tabel III diperlihatkan bahwa pada penelitian sebelumnya, pendekatan penelitian yang digunakan hanya fokus pada penilaian kerja tim pemula yang masih belajar dasar-dasar penggunaan UML. Mahasiswa pemula diberi tugas untuk mengevaluasi teman lainnya. Kelemahannya, tidak ada batasan atau tolok ukur dalam melakukan *peer evaluation*. Model penilaian yang diusulkan sekarang ditujukan kepada mahasiswa jurusan RPL dengan fokus pemodelan *UML diagram*, khususnya *use case diagram*. Meskipun fokus pembelajaran hanya diarahkan ke pemodelan *UML diagram*, tetapi penilaiannya diarahkan pada proyek perangkat lunak. Jadi, mahasiswa dilatih seperti menghadapi proyek perangkat lunak dengan kasus yang berbeda-beda.

TABEL III  
PERBANDINGAN FITUR ALAT KOLABORASI

Fitur	Alat Kolaborasi [18]	Usulan Alat Kolaborasi
Pembuatan Group	√	√
Pencarian Teman		√
Use case diagram	√	√
Screenshot diagram		√
Diskusi <i>real time</i>		√
Komponen Pendekatan Penilaian	Individu: - Tugas - <i>Peer Evaluation</i> Kelompok: - Tugas Tim - Diskusi - Kontribusi	1. Penugasan Kelompok 2. Pemodelan Diagram 3. Proyek Akhir

Pada tahap evaluasi, teknik pengujian yang digunakan adalah *black box*, *whitebox*, dan evaluasi dari sisi pengguna utama yaitu *User Acceptance Testing (UAT)*. Pada evaluasi UAT, sampel yang digunakan adalah 55 mahasiswa jurusan RPL yang berbeda angkatan.

A. Implementasi

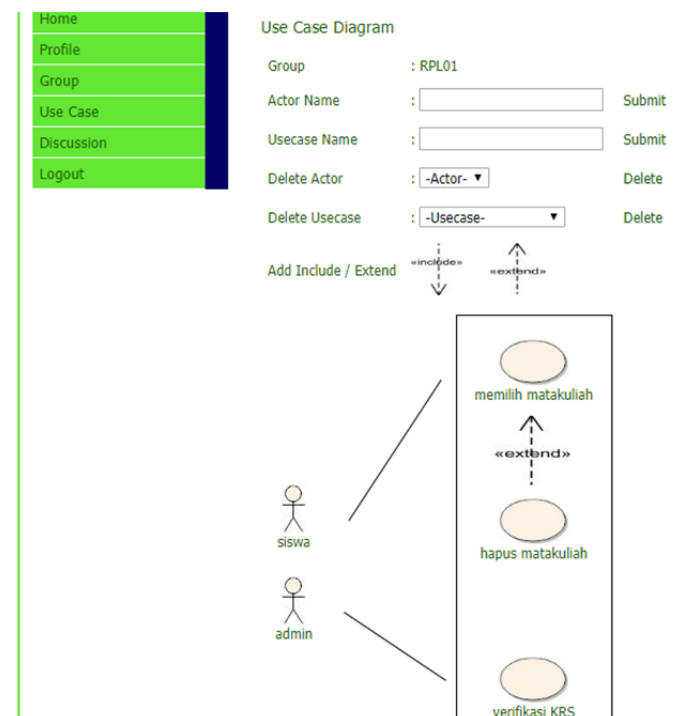
Tujuan implementasi ulang alat kolaborasi adalah mempromosikan penilaian adil bagi mahasiswa jurusan RPL dalam kolaborasi pendidikan perangkat lunak, khususnya pada pembelajaran UML. Mahasiswa dilatih untuk berkolaborasi dengan timnya, karena pada umumnya, mahasiswa menghindari tugas kelompok. RPL merupakan disiplin ilmu yang harus dikerjakan secara kerja tim. Alat kolaborasi juga

melatih komunikasi efektif antar mahasiswa meskipun berbeda angkatan.

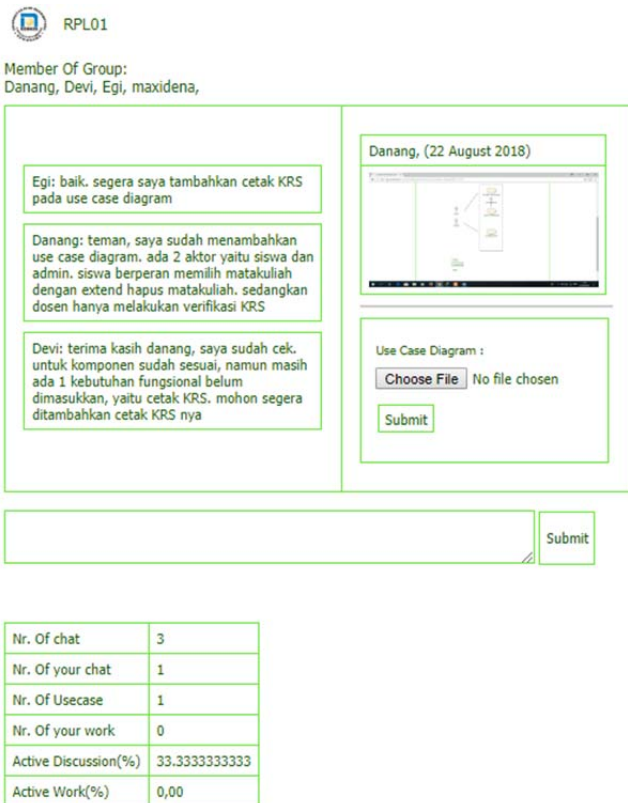
Gbr. 3 menampilkan cara menambahkan mahasiswa lain ke dalam tim. Jika mahasiswa sudah ditambahkan, akan keluar status “*Already Group*”. Maka, mahasiswa yang telah menjadi anggota tim dapat menggunakan fitur membuat *use case*, diskusi dengan tim, kontribusi, dan melihat hasil akhir penilaian. Adanya model seperti ini memudahkan dosen untuk memantau mahasiswa yang belum mendapatkan tim. Biasanya, mahasiswa senior enggan untuk menjadi satu tim dengan mahasiswa baru (di bawah satu level). Selain itu, dosen dapat mengetahui mahasiswa yang aktif dan pasif dalam kerja tim.



Gbr. 3 Tampilan menambahkan anggota.



Gbr. 4 Tampilan membuat use case diagram.



Gbr. 5. Tampilan forum diskusi.

Gbr. 4 menampilkan fasilitas pemodelan *use case diagram*. Tujuan penambahan fitur ini adalah untuk memudahkan para mahasiswa melakukan kontribusi langsung pada alat kolaborasi. Mahasiswa dapat berlatih sekaligus berdiskusi dengan teman-temannya. Hasil akhir dapat dikirimkan ke forum diskusi. Konsep pemodelan pada alat kolaborasi ini masih sederhana, karena tujuan utamanya untuk melatih mahasiswa benar-benar memahami komponen dan *relationship* dari *use case diagram*. Pertama, mahasiswa memasukkan *actor name* dan *usecase name*. Selanjutnya, mahasiswa dapat menambahkan *relationship* pada diagramnya. Pada *relationship association*, dapat langsung di-*drag* dari posisi aktor, sedangkan untuk *relationship include* dan *extends*, menggunakan model *drag and drop*.

Pada forum diskusi, mahasiswa dapat berkontribusi terhadap tugas timnya. Mahasiswa dapat menambahkan *use case scenario*, justifikasi pekerjaan mahasiswa lain, dan analisis mengenai studi kasus dan kebutuhan fungsional. Pada forum diskusi ini digunakan sebagai evaluasi keaktifan mahasiswa dalam kerja tim. Alat kolaborasi menampilkan waktu pengerjaan agar memudahkan dosen memantau mahasiswanya aktif dalam tim atau. Gbr. 5 menampilkan forum diskusi antar mahasiswa. Pada bagian kanan, terdapat tampilan gambar, yang berfungsi menampilkan hasil akhir dari pemodelan *use case diagram*. Konsep ini memudahkan mahasiswa untuk melakukan diskusi interaktif. Mahasiswa dapat langsung melakukan justifikasi, memberi kritik dan saran terhadap pemodelan yang dibuat oleh anggota lain. Pada bagian bawah, terdapat tabel yang menampilkan secara

keseluruhan jumlah kontribusi dari tiap mahasiswa dan total kontribusi tim. Hasilnya ditampilkan dalam bentuk persentase agar memudahkan dosen membaca keaktifan dari masing-masing tim. Adanya tampilan tabel ini dapat mendorong mahasiswa (malas) untuk berkontribusi lebih baik dari anggota lainnya. Mahasiswa akan merasa malu jika persentase yang ditampilkan masih 0 atau lebih rendah di antara anggota lainnya.

**B. Evaluasi**

Tahap evaluasi melakukan pengujian internal maupun eksternal sistem. Strategi pengujian yang digunakan adalah menggunakan metode pengujian *black box* dan evaluasi dari pengguna, biasa disebut UAT.

TABEL IV  
PENGUJIAN *BLACK BOX* PEMODELAN *USE CASE DIAGRAM*

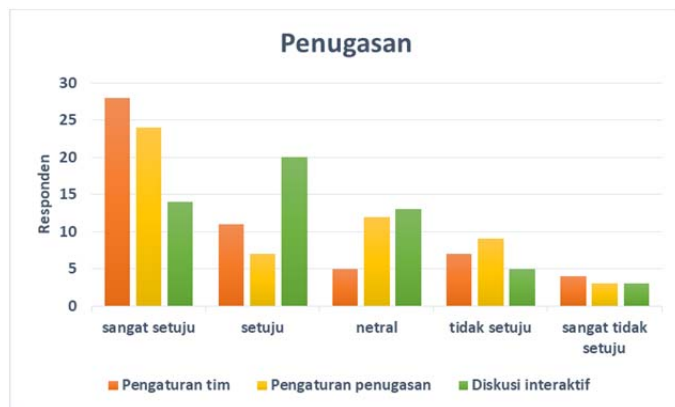
Data Masukan	Hasil yang Diharapkan	Observasi	Kesimpulan
<b>Kasus dan Hasil Uji: Data Benar</b>			
Nama <i>actor</i> , nama <i>use case</i> , penambahan <i>relationship association</i> , <i>include</i> , dan <i>extends</i>	Jika diisi dengan benar, proses akan menampilkan gambar diagram <i>use case</i>	Menampilkan <i>use case diagram</i>	Diterima
<b>Kasus dan Hasil Uji: Data Salah</b>			
Nama <i>actor</i> , nama <i>use case</i> , penambahan <i>relationship association</i> , <i>include</i> , dan <i>extends</i>	Jika form kosong, maka menampilkan notifikasi untuk mengisi <i>actor name</i> dan <i>usecase name</i>	Menampilkan notifikasi pengisian <i>actor name</i> dan <i>usecase name</i>	Diterima

TABEL V  
PENGUJIAN *BLACK BOX* MEMBUAT TIM

Data Masukan	Hasil yang Diharapkan	Observasi	Kesimpulan
<b>Kasus dan Hasil Uji: Data Benar</b>			
Nama grup dan ikon grup	Jika diisi dengan benar, proses akan menampilkan grup beserta ikonnya dan tautan tambah anggota, edit grup, dan hapus grup	Menampilkan nama grup, ikon, serta tautan tambah anggota, edit grup, dan hapus grup	Diterima
<b>Kasus dan Hasil Uji: Data Salah</b>			
Nama grup dan ikon grup	Jika form kosong, maka menampilkan notifikasi pengisian <i>form</i> nama grup dan <i>upload</i> ikon grup	Menampilkan notifikasi pengisian nama grup dan <i>upload</i> ikon grup	Diterima

TABEL VI  
TOPIK UAT

No	Pernyataan
1.	Pengaturan tim
2.	Pengaturan penugasan tim
3.	Peningkatan diskusi interaktif
4.	Kesesuaian <i>use case diagram</i> pada alat kolaborasi
5.	Kemudahan menggunakan komponen diagram
6.	Kesempatan kontribusi dalam tim
7.	Pengaturan proyek akhir
8.	Pendekatan penilaian adil ( <i>fair assessment</i> )
9.	Alat kolaborasi mudah digunakan dan dipahami
10.	<i>User interface</i>



Gbr. 6 Evaluasi parameter penugasan.

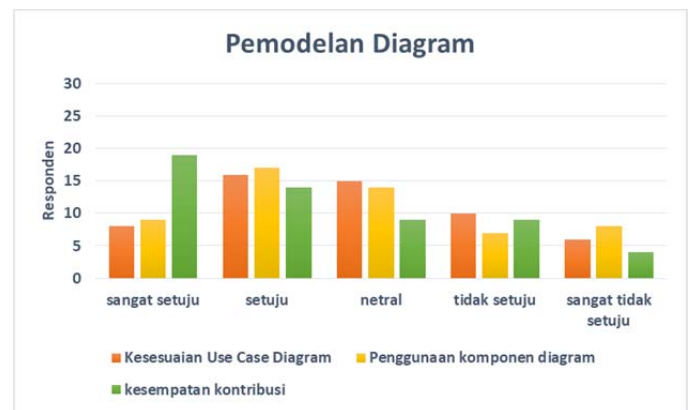
Pengujian *black box* mengevaluasi eksternal program seperti masukan data, hasil yang diharapkan, observasi program, dan kesimpulannya [19]. Makalah ini menguji fitur *pemodelan use case diagram* dan fitur membuat tim.

Tabel IV dan Tabel V menampilkan contoh pengujian *black box* dengan kondisi data benar dan data salah. Pengujian *black box* dilakukan untuk mengurangi kesalahan fungsi, tampilan, dan hasil yang tidak sesuai.

Pada pengujian UAT, evaluasi dilakukan menggunakan kuesioner untuk memudahkan evaluasi setiap persepsi pengguna. Sebelumnya, mahasiswa diberikan tutorial penggunaan sistem dan diberikan modul tentang penggunaan alat kolaborasi. Tabel VI menyajikan materi pengujian UAT.

Pada Tabel VI, materi pengujian tersebut berkaitan dengan materi pemodelan UML, parameter kolaborasi, kontribusi, penugasan, dan penilaian adil bagi kerja tim. Pengujian UAT dilaksanakan setelah mahasiswa melakukan percobaan (*trial*) selama satu minggu. Kuesioner yang digunakan menggunakan skala Likert dengan lima skala: Sangat Setuju, Setuju, Netral, Tidak Setuju, dan Sangat Tidak Setuju.

Pada evaluasi terdapat lima fokus utama pengujian, yaitu evaluasi pada penugasan, pemodelan diagram, pengaturan proyek akhir, pendekatan penilaian, dan kinerja sistem. Gbr. 6 menunjukkan hasil evaluasi pada fokus evaluasi penugasan. Grafik menunjukkan bahwa responden menjawab setuju di atas 60% komponen penugasan layak diterapkan pada alat kolaborasi. Hal ini menunjukkan bahwa mahasiswa tidak merasa kesulitan dalam mengatur timnya, mengatur pembagian tugas, dan dapat melakukan diskusi secara interaktif.



Gbr. 7 Evaluasi parameter pemodelan diagram.

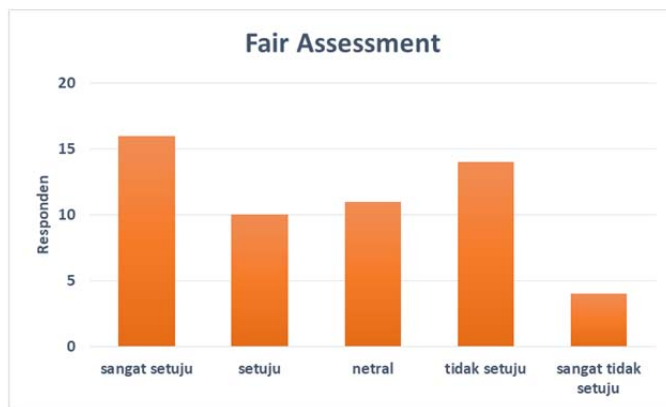


Gbr. 8 Evaluasi parameter pengaturan proyek akhir.

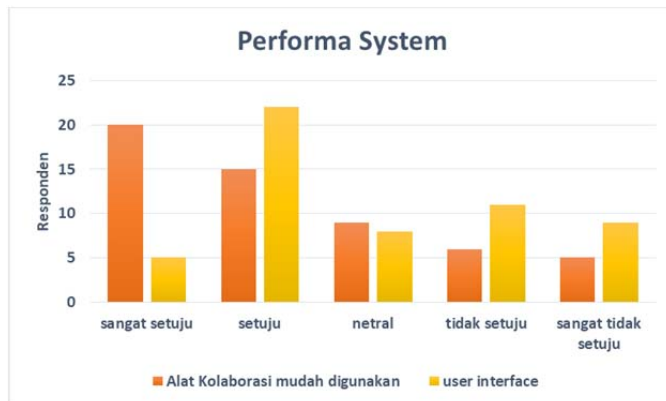
Grafik pada Gbr. 7 fokus pada evaluasi pemodelan diagram. Grafik menunjukkan bahwa rata-rata menjawab setuju di atas 50%. Jika dilihat pada grafik tersebut, persentase setuju dan sangat setuju lebih tinggi dibanding tidak setuju dan sangat tidak setuju. Dapat disimpulkan bahwa responden setuju pemodelan *use case diagram* pada alat kolaborasi dapat dijadikan acuan belajar mahasiswa, adanya kesesuaian penggunaan komponen diagram, dan mahasiswa mendapat kesempatan untuk berkontribusi lebih baik dibandingkan pembelajaran di kelas. Alat kolaborasi dapat membantu mahasiswa dalam pembelajaran pemodelan UML.

Gbr. 8 menunjukkan hasil evaluasi pada pengaturan proyek akhir. Grafik menunjukkan bahwa responden menjawab setuju di atas 41%, netral 29%, dan tidak sekitar 28%. Hal ini menunjukkan bahwa sebagian mahasiswa setuju pengaturan proyek akhir dapat digunakan sebagai komponen penilaian adil pada pembelajaran pemodelan perangkat lunak. Dapat disimpulkan bahwa pengaturan proyek akhir masih perlu perbaikan untuk mendapatkan peningkatan kepuasan pengguna.

Gbr. 9 menunjukkan hasil evaluasi penerapan *fair assessment* pada alat kolaborasi. Grafik menunjukkan bahwa responden menjawab setuju di atas 47%, netral 20%, dan tidak setuju 32%. Sebagian mahasiswa setuju bahwa pendekatan penilaian yang diusulkan sudah cukup adil untuk mengevaluasi kerja individu pada tim. Mahasiswa setuju bahwa penilaian disesuaikan dengan bobot tingkat kesulitan tugas, kontribusi, dan penilaian tugas akhir, yaitu proyek perangkat lunak.



Gbr. 9 Evaluasi parameter pendekatan penilaian.



Gbr. 10 Evaluasi kinerja sistem.

Gbr. 10 menunjukkan hasil evaluasi keseluruhan kinerja alat kolaborasi. Grafik menunjukkan bahwa responden menjawab setuju rata-rata di atas 55%, netral 15%, dan tidak setuju 27%. Hal ini membuktikan bahwa secara keseluruhan alat kolaborasi layak digunakan untuk menunjang pendidikan perangkat lunak pada level mahasiswa jurusan RPL. Mahasiswa setuju bahwa alat kolaborasi mudah dipahami dan mudah digunakan. Selain itu, dari sisi desain antarmuka, alat kolaborasi memiliki desain yang menarik. Artinya, desain tersebut tidak membuat para pengguna bingung dalam menggunakan alat kolaborasi.

Secara garis besar, mahasiswa menyatakan setuju bahwa alat kolaborasi dapat digunakan sebagai alat bantu pembelajaran mahasiswa RPL yang sedang belajar pengembangan proyek perangkat lunak. Diagram yang digunakan pada alat kolaborasi dapat dijadikan sebagai latihan mahasiswa dan pendekatan penilaian yang diusulkan cukup adil untuk mengevaluasi kerja tim mahasiswa meskipun berbeda angkatan. Pendekatan penilaian mampu mendorong tanggung jawab mahasiswa dan kontribusi mahasiswa yang awalnya malas untuk bekerja dalam tim.

## VI. KESIMPULAN

Pada makalah ini diusulkan pendekatan penilaian yang terdiri atas komponen penugasan, pemodelan diagram, dan proyek akhir. Evaluasi ditujukan kepada 55 mahasiswa jurusan RPL. Hasilnya, mahasiswa setuju di atas 60% bahwa penugasan proyek akhir layak digunakan sebagai bahan

penugasan mahasiswa dalam menghadapi proyek RPL. Mahasiswa setuju di atas 55% bahwa alat kolaborasi dengan pendekatan penilaian di dalamnya layak digunakan sebagai penunjang pembelajaran kelompok mahasiswa RPL, meskipun ada fitur yang harus diperbaiki untuk mendapatkan peningkatan kepuasan pengguna terhadap alat kolaborasi.

## UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada DP2M-DIKTI yang telah mendanai Penelitian Dosen Pemula dengan Daftar Isian Pelaksanaan Anggaran (DIPA) Nomor SP DIPA-042.06.1.401516/208 dan nomor kontrak 053/A38.04/UDN-09/II/201, mahasiswa jurusan RPL yang telah mendukung pelaksanaan pengujian *user acceptance testing*, dan terima kasih kepada JNTETI yang telah memberikan komentar dan saran.

## REFERENSI

- [1] T. Kilamo, I. Hammouda, dan M.A. Chatti, "Teaching Collaborative Software Development: A Case Study," *2012 34th Int. Conf. Softw. Eng.*, 2012, hal. 1165–1174.
- [2] C.Y. Chen dan P.P. Chong, "Software Engineering Education: A Study on Conducting Collaborative Senior Project Development," *J. Syst. Softw.*, Vol. 84, No. 3, hal. 479–491, Mar. 2011.
- [3] A. Dennis, B.H. Wixom, dan R.M. Roth, *System Analysis and Design*, 4th Edition, New Jersey, USA: Wiley, 2009.
- [4] G. Sedrakyan, S. Poelmans, dan M. Snoeck, "Assessing the Influence of Feedback-Inclusive Rapid Prototyping on Understanding the Semantics of Parallel UML Statecharts by Novice Modellers," *Inf. Softw. Technol.*, Vol. 82, hal. 159–172, 2017.
- [5] E.R. Subhiyacto dan D.W. Utomo, "RMTool; Sebuah Aplikasi Pemodelan Persyaratan Perangkat Lunak Menggunakan UML," *JNTETI*, Vol. 6, No. 3, hal. 268–274, 2017.
- [6] D.W. Utomo, E.R. Subhiyacto, dan S. Ahmad, "Tool Enhancement for Collaborative Software Engineering Education," *Semin. Nas. Teknol. Inf. dan Komun. 2015 (SENTIKA 2015)*, 2015, hal. 9–16.
- [7] D.W. Utomo dan E.R. Subhiyacto, "Assessing Novice Teams in Collaborative Software Engineering Education," *Semin. Nas. Apl. Teknol. Inf. 2016*, 2016, hal. 23–28.
- [8] M. Keinänen, J. Ursin, dan K. Nissinen, "How to Measure Students' Innovation Competences in Higher Education: Evaluation of an Assessment Tool in Authentic Learning Environments," *Stud. Educ. Eval.*, Vol. 58, hal. 30–36, 2018.
- [9] M.A. Conde, R. Colomo-Palacios, F.J. García-Peñalvo, dan X. Larrucea, "Teamwork Assessment in the Educational Web of Data: A Learning Analytics Approach Towards ISO 10018," *Telemat. Informatics*, Vol. 35, No. 3, hal. 551–563, 2018.
- [10] A.E. Guerrero-Roldán dan I. Noguera, "A Model for Aligning Assessment with Competences and Learning Activities in Online Courses," *Internet High. Educ.*, Vol. 38, hal. 36–46, 2018.
- [11] T. Hooker, "Transforming Teachers' Formative Assessment Practices Through Eportfolios," *Teach. Teach. Educ.*, Vol. 67, hal. 440–453, 2017.
- [12] J.M.R. Costa, R.M. Feitosa, dan C.R.B. de Souza, "Tool Support for Collaborative Software Development Based on Dependency Analysis," *2010 6th Int. Conf. on Collab. Comput. Networking, Appl. Work. (CollaborateCom)*, 2010, hal. 1–10.
- [13] M. Qomaruddin, A.A. Rahman, dan N.A. Iahad, "User Acceptance Test of Computer-Assisted Problem-Based Learning Assessment Tool (CAPBLAT)," *J. Educ. Learn.*, Vol. 8, Mo. 1, hal 71–77, 2014.
- [14] J. Chen, G. Qiu, L. Yuan, L. Zhang, dan G. Lu, "Assessing Teamwork Performance in Software Engineering Education: A Case in a Software Engineering Undergraduate Course," *2011 18th Asia-Pacific Softw. Eng. Conf.*, 2011, hal. 17–24.



- [15] A. Dennis, B.H. Wixom, dan R.M. Roth, *System Analysis and Design*, 5th Edition, New Jersey, USA: Wiley, 2012.
- [16] R. Yunis dan K. Telaumbanua, "Pengembangan E-Learning Berbasis LMS untuk Sekolah, Studi Kasus SMA/SMK di Sumatera Utara," *JNTETI*, Vol. 6, No. 1, hal. 32–36, 2017.
- [17] E. Susilo, F.D. Wijaya, dan R. Hartanto, "Perancangan dan Evaluasi User Interface Aplikasi Smart Grid Berbasis Mobile Application," *JNTETI*, Vol. 7, No. 2, hal. 150–157, 2018.
- [18] D.W. Utomo, E.R. Subhiyako, dan S. Ahmad, "Tool Enhancement for Collaborative Software Engineering Education," *Sentika*, 2015, hal. 9–16.
- [19] E.R. Subhiyako dan D.W. Utomo, "Software Testing Techniques and Strategies Use in Novice Software Teams," *J. Sist. Inf.*, Vol. 5, No. 5, hal. 556–562, 2016.