

**IMPLEMENTASI PERSONAL EXTREME PROGRAMMING
DALAM PENGEMBANGAN INTEGRASI
SISTEM INFORMASI AKADEMIK DAN E-LEARNING**

SKRIPSI

Oleh:
ARDISCA EVANANDY
NIM. 17650083



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2022**

**IMPLEMENTASI PERSONAL EXTREME PROGRAMMING
DALAM PENGEMBANGAN INTEGRASI
SISTEM INFORMASI AKADEMIK DAN E-LEARNING**

SKRIPSI

Oleh:
ARDISCA EVANANDY
NIM. 17650083

**Diajukan kepada:
Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan Dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)**

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2022**

HALAMAN PERSETUJUAN

IMPLEMENTASI PERSONAL EXTREME PROGRAMMING DALAM PENGEMBANGAN INTEGRASI SISTEM INFORMASI AKADEMIK DAN E-LEARNING

SKRIPSI

Oleh:
ARDISCA EVANANDY
NIM. 17650083

Telah Diperiksa dan Disetujui untuk Diuji
Tanggal: 1 Desember 2022

Dosen Pembimbing I



Prof. Dr. Suhartono, M.Kom
NIP.19680519 200312 1 001

Dosen Pembimbing II



Dr. M. Faisal, M.T
NIP. 19740510 200501 1 007

Mengetahui,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang



Dr. Fachrol Kurniawan, M.MT, IPM
NIP. 19771020 200912 1 001

HALAMAN PENGESAHAN

IMPLEMENTASI PERSONAL EXTREME PROGRAMMING DALAM PENGEMBANGAN INTEGRASI SISTEM INFORMASI AKADEMIK DAN E-LEARNING

SKRIPSI

Oleh:
ARDISCA EVANANDY
NIM. 17650083

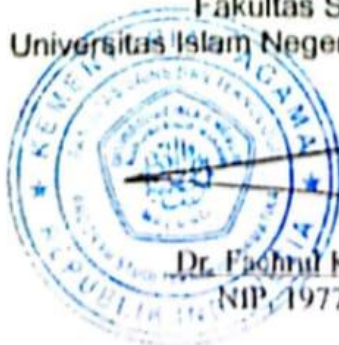
Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S. Kom)
Pada Tanggal: 12 Desember 2022


Susunan Dewan Penguji

1. Ketua Penguji : Fatchurrohman, M.Kom
NIP. 19700731 200501 1 002
2. Anggota Penguji I : Supriyono, M. Kom
NIP. 19841010 201903 1 012
3. Anggota Penguji II : Prof. Dr. Suhartono, M.Kom
NIP.19680519 200312 1 001
4. Anggota Penguji III: Dr. M. Faisal, M.T
NIP. 19740510 200501 1 007

()
()
()
()

Mengetahui,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang




Dr. Fachrul Kurniawan, M.MT, IPM
NIP. 19771020 200912 1 001

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Ardisca Evanandy

NIM : 17650083

Program Studi : Teknik Informatika

Fakultas : Sains dan Teknologi

Judul Skripsi : Implementasi *Personal Extreme Programming*

Dalam Pengembangan Integrasi Sistem Informasi Akademik dan
E-learning

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar – benar merupakan hasil karya saya sendiri, bukan merupakan pengambilalihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka. Apabila dikemudian hari terbukti atau dapat dibuktikan Skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 12 Desember 2022

Yang membuat pernyataan,



Ardisca Evanandy

NIM. 17650083

MOTTO

“Kalau capek istirahat, bukan berhenti”.

HALAMAN PERSEMBAHAN

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Puji Syukur kehadiran Allah SWT, shalawat dan salam kepada Rasul-Nya

Penulis persembahkan sebuah karya ini kepada:

Kepada orang tua penulis, Bapak Suliono dan Ibu Samusri yang selalu memberikan segala bentuk dukungan dan motivasi agar penulis mampu menyelesaikan skripsi dan lulus dengan cukup memuaskan. Kepada keluarga besar penulis dari pihak Ayah maupun pihak Ibu yang senantiasa bangga dan menyemangati penulis dalam melaksanakan kuliah.

Kepada bapak Prof. Dr Suhartono S.Si M.Kom selaku dosen pembimbing I dan bapak Dr. M. Faisal, M.T selaku dosen pembimbing II yang telah dengan sabar membimbing penulis agar dapat menyelesaikan skripsi. Serta seluruh dosen di prodi Teknik Informatika Universitas Islam Maulana Malik Ibrahim Malang yang telah mengajarkan ilmu-ilmu baru bagi penulis.

Teman-teman Teknik Informatika angkatan 2017 Universitas Islam Maulana Malik Ibrahim Malang yang selalu memberikan semangat dan dukungan kepada penulis. Serta seluruh orang yang pernah terlibat dan membantu baik secara langsung maupun secara tidak langsung, penulis mengucapkan banyak terima kasih.

KATA PENGANTAR

Assalamu 'alaikum Wr.Wb.

Puji syukur penulis panjatkan kepada Allah SWT yang senantiasa memberikan rahmat dan kesehatan, sehingga penulis mampu menyelesaikan skripsi ini dengan tepat waktu. Dalam menyelesaikan skripsi ini, bukan hanya karena usaha keras dari penulis sendiri, akan tetapi karena adanya dukungan dari berbagai pihak. Oleh karena itu penulis berterima kasih kepada:

1. Prof. Dr. M. Zainuddin, M.A., selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Dr. Sri Hariani, M.Si., selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Fachrul Kurniawan M.MT., IPM selaku Ketua Prodi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang
4. Prof. Dr Suhartono S.Si M.Kom selaku Dosen Pembimbing I yang dengan sabar membimbing penulisan skripsi dari awal hingga akhir.
5. Dr. M. Faisal, M.T selaku Dosen Pembimbing II yang telah membimbing, memberikan arahan penulisan skripsi dari awal hingga akhir.
6. Fatchurrohman, M.Kom selaku peguji I dan Supriyono, M. Kom selaku peguji II yang telah dengan sabar memberi arahan dan saran dalam menyelesaikan skripsi ini.
7. Dr. Cahyo Crysdiyan selaku dosen wali yang dengan sabar memberikan saran dan arahan dalam menempuh perkuliahan.
8. Segenap civitas akademik Program Studi Teknik Informatika, dan seluruh dosen yang telah memberikan ilmu dan arahan semasa kuliah.
9. Orang Tua yang telah memberikan banyak dukungan dan doa sehingga penulis mampu menyelesaikan skripsi.
10. Sahabat dan sahabati rayon "Pencerahan" Galileo serta teman-teman Teknik Informatika angkatan 2017 yang telah mendukung penulis dalam menyelesaikan skripsi khususnya Rafika Syahrani, Ayu Dian, Jayanti

Galuh, Ramadhana Fardian Perkasa, Hamdan Mursyida Arifin, dan Mochammad Fahim Fikri, Alvinna yang telah berjalan bersama menyelesaikan skripsi. Donny Kurniawan yang memberikan tumpangan ketika di Malang.

11. Serta semua pihak yang secara tidak langsung membantu penulis menyelesaikan skripsi yang tidak bisa disebutkan satu per satu.
12. *Last but not least, I wanna thank me .I wanna thank me for believing in me. I wanna thank me for doing all this hard work. I wanna thank me for having no days off. I wanna thank me for never quitting.*

Penulis menyadari bahwa dalam penulisan skripsi ini masih terdapat kekurangan. Maka dari itu penulis meminta maaf jikalau seandainya ada beberapa hal dalam penelitian ini yang kurang ataupun terlewat. Serta dengan senang hati penulis menerima segala saran dan kritik yang membangun. Disamping itu penulis juga berharap skripsi ini bermanfaat bagi semua orang yang membacanya. *Wassalamualaikum Wr.Wb.*

Malang,

2022

Penulis

DAFTAR ISI

HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
PERNYATAAN KEASLIAN TULISAN	v
MOTTO	vi
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR TABEL	xiii
DAFTAR GAMBAR	xiv
ABSTRAK	xvi
ABSTRACT	xvii
مستخلص البحث	xviii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	7
1.3 Tujuan Penelitian	7
1.4 Batasan Masalah	7
1.5 Manfaat Penelitian	8
1.6 Sistematika Penulisan	8
BAB II TUJUAN PUSTAKA	10
2.1 Penelitian Terdahulu	10
2.2 Pengembangan Aplikasi.....	15
2.3 Integrasi Aplikasi	15
2.4 Sistem Informasi Akademik	16
2.5 E-Learning	17
2.6 Web Service REST	18
2.7 <i>Agile Develepemt Method</i>	19
2.8 <i>Extreme Programming</i>	21

2.9 <i>Personal Extreme Programming</i>	23
2.10 Automated Testing	24
BAB III DESAIN DAN IMPLEMENTASI	26
3.1 Desain Penelitian	26
3.2 Metode Pengumpulan Data	27
3.2.1 Metode Observasi	27
3.2.1 Metode Studi Literatur	27
3.3 Analisis Sistem	27
3.3.1 Evaluasi Sistem Saat Ini	28
3.4 Usulan Perencanaan Sistem	28
3.4.1 Tujuan Perencanaan Sistem	29
3.4.2 Perencanaan Yang Diusulkan	29
3.4.3 Evaluasi Sistem Yang Diusulkan	29
3.5 Metode Pengembangan Sistem	30
3.5.1 Requirements	32
3.5.2 Planning	33
3.5.3 Iteration Initialization	36
3.5.4 Design	36
3.5.5 Implementation	39
3.5.6 Testing	40
3.5.7 Retrospective	41
BAB IV IMPLEMENTASI SISTEM	42
4.1 Iterasi Pertama	42
4.1.1 Iteration Initialization	42
4.1.2 Design	43
4.1.3 Implementation	45
4.1.4 System Testing	50
4.1.5 Restrospective	53
4.2 Iterasi Kedua	53
4.2.1 Iteration Initialization	53
4.2.2 Design	53

4.2.3 Implementation	56
4.2.4 System Testing	61
4.2.5 Restrospective	64
4.3 Iterasi Ketiga	64
4.3.1 Iteration Initialization.....	64
4.3.2 Design	65
4.3.3 Implementation	68
4.3.4 System Testing.....	73
4.3.5 Restrospective	76
4.4 Analisis Integration System	76
4.5 Analisis Penggunaan <i>Personal Extreme Programming</i>	82
4.6 Pengembangan Integrasi Sistem Informasi Akdemik dengan <i>E-learning</i> dalam Pandangan Islam.....	83
BAB V PENUTUP	87
5.1 Kesimpulan	87
5.2 Saran	89
DAFTAR PUSTAKA	90

DAFTAR TABEL

Tabel 2.1 Perbandingan Methode.....	13
Tabel 3.1 Evaluasi Sistem	28
Tabel 3.2 Perbandingan Sistem Lama dan Sistem Baru.....	30
Tabel 3.3 User Story.....	33
Tabel 3.4 Story Point dan Estimasi	34
Tabel 3.5 User Story Value	35
Tabel 3.6 Risk User Story	36
Tabel 4.1 Hasil Pengujian Jurnal Perkuliahan.....	52
Tabel 4.2 Restrospective Jurnal Perkuliahan.....	53
Tabel 4.3 Hasil Pengujian Absensi.....	64
Tabel 4.4 Restrospective Absensi.....	64
Tabel 4.5 Hasil Pengujian Nilai	76
Tabel 4.6 Restrospective Nilai	76
Tabel 4.7 Hasil Data Integrasi	79
Tabel 4.8 Tahapan Integrasi	79
Tabel 4.9 Perbandingan Jurnal Perkuliahan Sebelum dan Sesudah Integrasi	79
Tabel 4.10 Perbandingan Absensi Sebelum dan Sesudah Integrasi	81
Tabel 4.11 Perbandingan Sebelum dan Sesudah Integrasi	81

DAFTAR GAMBAR

Gambar 2.1 Alur Extreme Programming	21
Gambar 3.1 Alur Penelitian	26
Gambar 3.2 Alur Personal Extreme Programming.....	31
Gambar 3.3 DFD Level 0	37
Gambar 3.4 DFD Level 1	38
Gambar 3.5 Arsitektur Sistem	39
Gambar 4.1 Activity Diagram Jurnal Perkuliahan sebelum Integrasi	43
Gambar 4.2 Activity Diagram Jurnal Perkuliahan sesudah Integrasi.....	44
Gambar 4.3 Test Cases Sistem Jurnal Perkuliahan sebelum Integrasi	45
Gambar 4.4 Test Cases Sistem Jurnal Perkuliahan sesudah Integrasi.....	46
Gambar 4.5 Test Result Jurnal Perkuliahan setelah Integrasi	47
Gambar 4.6 Tampilan Jurnal Perkuliahan sebelum Integrasi.....	47
Gambar 4.7 API Jurnal Perkuliahan	48
Gambar 4.8 API Jurnal Perkuliahan	49
Gambar 4.9 Tampilan Jurnal Perkuliahan sesudah Integrasi	50
Gambar 4.10 Test Execution Jurnal Perkuliahan sebelum Integrasi	51
Gambar 4.11 Test Execution Jurnal Perkuliahan sesudah Integrasi.....	52
Gambar 4.12 Activity Diagram Absensi sebelum Integrasi	54
Gambar 4.13 Activity Diagram Absensi sesudah Integrasi.....	55
Gambar 4.14 Test Cases Sistem Absensi sebelum Integrasi Export	56
Gambar 4.15 Test Cases Sistem Absensi sebelum Integrasi Import	57
Gambar 4.16 Test Cases Sistem Absensi sesudah Integrasi.....	57
Gambar 4.17 Test Result Absensi setelah Integrasi	58
Gambar 4.18 Tampilan Absensi sebelum Integrasi.....	59
Gambar 4.19 API Absensi	60
Gambar 4.20 Tampilan Absensi sesudah Integrasi	61
Gambar 4.21 Test Execution Absensi sebelum Integrasi Export	62
Gambar 4.22 Test Execution Absensi sebelum Integrasi Import	62
Gambar 4.23 Test Execution Absensi sesudah Integrasi.....	63
Gambar 4.24 Activity Diagram Nilai sebelum Integrasi	66
Gambar 4.25 Activity Diagram Nilai sesudah Integrasi.....	67
Gambar 4.26 Test Cases Sistem Nilai sebelum Integrasi Import	69
Gambar 4.27 Test Cases Sistem Nilai sebelum Integrasi Export	69
Gambar 4.28 Test Cases Sistem Nilai sesudah Integrasi	70
Gambar 4.29 Test Result Nilai setelah Integrasi	70
Gambar 4.30 Tampilan Nilai sebelum Integrasi.....	71
Gambar 4.31 API Nilai	72

Gambar 4.32 Tampilan Nilai sesudah Integrasi	72
Gambar 4.33 Test Execution Nilai sebelum Integrasi Import	73
Gambar 4.34 Test Execution Nilai sebelum Integrasi Export	74
Gambar 4.35 Test Execution Nilai sesudah Integrasi.....	75

ABSTRAK

Evanandy, Ardisca. 2022. **Implementasi Personal Extreme Programming Dalam Pengembangan Integrasi Sistem Informasi Akademik dan E-learning.** Skripsi. Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Prof, Dr, Suhartono, M.kom (II) Dr. M. Faisal, M.T

Kata kunci: *Personal Extreme Programming*, Integrasi, Sistem Informasi Akademik, *E-learning*

Dosen adalah salah satu tenaga pengajar pada perguruan tinggi. Dalam melakukan kegiatan mengajar, dosen menggunakan platform Sistem Informasi akademik dan E-learning. Sistem Informasi Akademik (SIKAD) adalah suatu sistem yang dirancang untuk mengolah semua data mata pelajaran melalui perangkat lunak dan perangkat keras sehingga kinerja kegiatan pendidikan dapat disederhanakan menjadi informasi yang berguna dalam pengelolaan dan pemilihan perguruan tinggi. E-Learning atau pembelajaran elektronik merupakan salah satu media, dengan tujuan agar memudahkan kegiatan belajar mengajar. E-learning merupakan media pembelajaran elektronik yang muncul pada awal mula teknologi menjadi tumpuan perguruan tinggi di Indonesia. Pada UIN Malang, dua platform ini sudah digunakan oleh para dosen, namun belum terintegrasi dengan baik. Menggunakan teknologi pengintegrasian sistem dengan Representational state transfer (REST) dan juga memanfaatkan Personal Extreme Programming (PXP) sebagai pengembangan perangkat lunak, sistem sangat mungkin dapat dikembangkan agar lebih terintegrasi. Dalam pengintegrasian sistem, terdapat 3 fitur yang diintegrasikan, yaitu Jurnal Perkuliahan, Absensi, Nilai. Hasil dalam pengintegrasian sistem didapat pengintegrasian system Jurnal perkuliahan, hanya memerlukan 8 tahapan dan juga memerlukan waktu 36,684 detik sedangkan untuk system sebelumnya memerlukan 13 tahapan dan juga memerlukan waktu 54,717 detik. Pada pengintegrasian system Absensi, hanya memerlukan 7 tahapan dan juga memerlukan waktu 32,977 detik sedangkan untuk system sebelumnya memerlukan 23 tahapan dan juga memerlukan waktu 87,521 detik. Pada pengintegrasian system Nilai, hanya memerlukan 13 tahapan dan juga memerlukan waktu 41,576 detik sedangkan untuk system sebelumnya memerlukan 22 tahapan dan juga memerlukan waktu 115.32 detik. Sedangkan dalam pengembangan menggunakan PXP diperoleh hasil efisiensi berdasarkan perbandingan waktu estimasi dengan waktu sebenarnya mencapai 97%, hal ini terjadi karena ada hal tidak terduga pada iterasi pertama.

ABSTRACT

Evanandy, Ardisca. 2022. **Implementation of Personal Extreme Programming in the Development of Academic Information System Integration and E-learning..** Undergraduate Thesis. Department of Informatics Engineering, Faculty of Science and Technology, State Islamic University of Maulana Malik Ibrahim Malang. Supervisor: (I) Prof, Dr, Suhartono, M.Kom (II) Dr. M. Faisal, M.T

Keywords : *Keywords: Personal Extreme Programming, Integration, Academic Information Systems, E-learning*

The lecturer is one of the teaching staff at the university. In carrying out teaching activities, lecturers use the Academic Information System and E-learning platforms. The Academic Information System (SIKAD) is a system designed to process all subject data through software and hardware so that the performance of educational activities can be simplified into useful information in the management and selection of tertiary institutions. E-Learning or electronic learning is one of the media, with the aim of facilitating teaching and learning activities. E-learning is an electronic learning medium that emerged at the beginning of technology to become the foundation of higher education in Indonesia. At UIN Malang, these two platforms have been used by lecturers, but have not been well integrated. Using system integration technology with Representational State Transfer (REST) and also utilizing Personal Extreme Programming (XP) as software development, it is possible for the system to be developed to be more integrated. In system integration, there are 3 features that are integrated, namely Lecture Journal, Attendance, Grades. The results in system integration were obtained by integrating the lecture journal system, only requiring 8 stages and also requiring 36.684 seconds while for the previous system requiring 13 stages and also requiring 54.717 seconds. In integrating the attendance system, it only requires 7 stages and also takes 32.977 seconds, while the previous system requires 23 stages and also takes 87.521 seconds. In the integration of the Value system, it only requires 13 stages and also takes 41.576 seconds, while the previous system requires 22 stages and also takes 115.32 seconds. Whereas in the development using XP, efficiency results based on a comparison of estimated time with actual time reached 97%, this happened because there was something unexpected in the first iteration.

مستخلص البحث

تنفيذ البرمجة الشخصية المتطرفة في تطوير تكامل نظام المعلومات الأكاديمية والتعليم . إيفاناندي ، أريديسكا . 2022. الإلكتروني . مقال . برنامج دراسة هندسة المعلوماتية ، كلية العلوم والتكنولوجيا ، جامعة الولاية الإسلامية مولانا مالك

إبراهيم مالانج . المشرفون : (1) الأستاذ الدكتور سوهارتون محمد كوم (2) د . فيصل ، مت

الكلمات المفتاحية : البرمجة الشخصية المتطرفة ، التكامل ، نظم المعلومات الأكاديمية ، التعلم الإلكتروني

المحاضر هو أحد أعضاء هيئة التدريس بالجامعة . في القيام بأنشطة التدريس ، يستخدم المحاضرون نظام المعلومات هو نظام مصمم لمعالجة جميع بيانات الموضوع من (SIKAD) الأكاديمية ومنصات التعلم الإلكتروني . نظام المعلومات الأكاديمية خلال البرامج والأجهزة بحيث يمكن تبسيط أداء الأنشطة التعليمية إلى معلومات مفيدة في إدارة واختيار مؤسسات التعليم العالي . التعلم الإلكتروني أو التعلم الإلكتروني من الوسائط التي تهدف إلى تسهيل أنشطة التدريس والتعلم . التعلم الإلكتروني هو وسيلة تعلم تم استخدام هاتين المنصتين ، UIN Malang إلكترونية ظهرت في بداية التكنولوجيا لتصبح أساس التعليم العالي في إندونيسيا . في وأيضاً (REST) من قبل المحاضرين ، لكن لم يتم دمجها بشكل جيد . باستخدام تقنية تكامل النظام مع نقل الحالة التمثيلية كتطوير للبرامج ، من الممكن تطوير النظام ليكون أكثر تكاملاً . في تكامل النظام ، (PXP) استخدام البرمجة الشخصية المتطرفة هناك 3 ميزات مدمجة ، وهي مجلة المحاضرات ، الحضور ، الدرجات . تم الحصول على النتائج في تكامل النظام من خلال دمج نظام دفتر المحاضرات ، والذي يتطلب 8 مراحل فقط ويتطلب أيضاً 36.684 ثانية بينما يتطلب النظام السابق 13 مرحلة ويتطلب أيضاً ثانية . في تكامل نظام الحضور ، لا يتطلب سوى 7 مراحل ويستغرق أيضاً 32.977 ثانية ، بينما يتطلب النظام السابق 54.717 مرحلة ويستغرق أيضاً 87.521 ثانية . في تكامل نظام القيم ، لا يتطلب سوى 13 مرحلة ويستغرق أيضاً 41.576 ثانية ، 23 وصلت نتائج الكفاءة ، PXP بينما يتطلب النظام السابق 22 مرحلة ويستغرق أيضاً 115.32 ثانية . بينما في التطوير باستخدام . بناءً على مقارنة الوقت المقدر بالوقت الفعلي إلى 97٪ ، حدث هذا بسبب وجود شيء غير متوقع في التكرار الأول

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pendidikan adalah sesuatu yang penting bagi kehidupan sehari-hari. Dalam era teknologi yang sangat cepat sekali perkembangannya, masyarakat diwajibkan belajar terus menerus seiring berkembangnya teknologi. Dalam hadis yang di riwayatkan oleh Ibnu Majah,

طَلَبُ الْعِلْمِ فَرِيضَةٌ عَلَى كُلِّ مُسْلِمٍ

“Menuntut ilmu itu wajib atas setiap Muslim” (HR. Ibnu Majah no. 224, dari sahabat Anas bin Malik radhiyallahu ‘anhu, dishahihkan Al Albani dalam Shahiih al-Jaami’ish Shaghiir no. 3913). Itu berarti menuntut ilmu atau mengeyam pendidikan sangatlah penting bagia seluruh umat islam. Bagi orang-orang beriman yang menuntut ilmu, memilik derajat yang sangat tinggi di hadapan Allah *subhanahu wa ta'ala*, sebagaimana yang tercantum dalam Al-Quran, pada surat Al-Mujadalah ayat 11,

يَا أَيُّهَا الَّذِينَ آمَنُوا إِذَا قِيلَ لَكُمْ تَفَسَّحُوا فِي الْمَجَالِسِ فَافْسَحُوا يَفْسَحِ اللَّهُ لَكُمْ ۗ وَإِذَا قِيلَ انشُرُوا فَانشُرُوا يَرْفَعِ

اللَّهُ الَّذِينَ آمَنُوا مِنْكُمْ ۖ وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ ۗ وَاللَّهُ بِمَا تَعْمَلُونَ خَبِيرٌ

"Wahai orang-orang yang beriman! Apabila dikatakan kepadamu, berilah kelapangan di dalam majelis-majelis, maka lapangkanlah, niscaya Allah akan memberi kelapangan untukmu. Dan apabila dikatakan, Berdirilah kamu, maka berdirilah, niscaya Allah akan mengangkat (derajat) orang-orang yang beriman di

antaramu dan orang-orang yang diberi ilmu beberapa derajat. Dan Allah Mahateliti apa yang kamu kerjakan."

Dalam tafsir terbitan Kementrian Agama di Republik Indonesia, dijelaskan bahwa "Akhir ayat ini menerangkan bahwa Allah akan mengangkat derajat orang yang beriman, taat dan patuh kepada-Nya, melaksanakan perintah-Nya, menjauhi larangan-Nya, berusaha menciptakan suasana damai, aman, dan tenteram dalam masyarakat, demikian pula orang-orang berilmu yang menggunakan ilmunya untuk menegakkan kalimat Allah. Dari ayat ini dipahami bahwa orang-orang yang mempunyai derajat yang paling tinggi di sisi Allah ialah orang yang beriman dan berilmu. Ilmu yang diamalkan sesuai dengan yang diperintahkan Allah dan rasul-Nya.". Derajat orang yang menuntut ilmu dan mengamalkannya sangatlah tinggi di hadapan Allah *subhanahu wa ta'ala*. Maka dari itu menuntut ilmu menjadi sangat penting dimanapun berada.

Pada Era digital sekarang ini, menuntut ilmu menjadi sangat mudah, akses informasi yang sangat banyak dapat dijangkau hanya dengan ketikan tangan. Pada beberapa kampus negeri maupun swasta, sudah banyak yang menerapkan pembelajaran secara daring, melalui platform edukasi mereka masing masing, salah satunya adalah kampus Universitas Islam Negeri Maulana Malik Ibrahim Malang. Kampus ini sudah menerapkan platform edukasi *e-learning* guna kegiatan pembelajaran secara jarak jauh ataupun daring.

E-Learning atau pembelajaran elektronik merupakan salah satu media, dengan tujuan agar memudahkan kegiatan belajar mengajar. *E-learning* merupakan proses pembelajaran yang muncul pada awal mula teknologi menjadi

tumpuan perguruan tinggi di Indonesia. Layanan, kecepatan, kemudahan berbagi informasi dan pengetahuan, efisiensi dan efektivitas adalah alasan mengapa e-learning adalah jawaban untuk perbaikan pendidikan atau metode pembelajaran tradisional (Maturbongs, 2019). Dengan *e-Learning*, interaksi Mahasiswa dan Dosen akan terjalin dengan baik. Hal ini karena, *e-Learning* memudahkan dosen untuk melakukan pembelajaran jarak jauh sehingga Mahasiswa dengan mudah mengakses materi dari dosen. Selain menjadi media pembelajaran, *e-Learning* bisa digunakan mahasiswa untuk mengetahui kinerja belajar mereka pada setiap kegiatan belajar mengajar. Tetapi untuk mengetahui kinerja belajar keseluruhan, mahasiswa harus menunggu hingga akhir semester untuk melihatnya di Sistem Informasi Akademik, karena dosen harus melakukan rekap penilaian pada *e-learning*.

Sistem Informasi Akademik (SIKAD) adalah suatu sistem yang dirancang untuk mengolah semua data mata pelajaran melalui perangkat lunak dan perangkat keras sehingga kinerja kegiatan pendidikan dapat disederhanakan menjadi informasi yang berguna dalam pengelolaan dan pemilihan perguruan tinggi (Homaidi, 2016). SIKAD juga sering digunakan oleh mahasiswa untuk, mengetahui kinerja belajar mereka selama satu semester, pada media ini mahasiswa juga bisa memilih matakuliah yang ingin diambil pada semester selanjutnya. Oleh karena itu, SIKAD dan *e-Learning* menjadi suatu hal yang penting dan seharusnya ada pada perguruan tinggi. Sayangnya Sistem Informasi Akademik (SIKAD) dan *e-Learning* pada Universitas Islam Negeri Maulana

Malik Ibrahim Malang (UIN Malang) belum terintegrasi dengan baik, karena data *e-learning* dan SIAKAD masih dikelola secara terpisah.

Pada UIN Malang saat ini, pengelolaan data kedua sistem tersebut masih dibangun dan dikelola secara terpisah dan belum terintegrasi dengan baik. Dosen diharuskan memasukkan data nilai berulang kali pada *e-learning* dan juga siakad ketika ingin memberikan nilai pada Mahasiswa. Selain itu, pendataan dalam hal penilaian mahasiswa, dosen masih harus melakukan rekap absensi dan juga jurnal perkuliahan. Hal ini tentunya kurang efektif, karena kedua data masih dikelola secara terpisah dan juga belum terintegrasi dengan baik. Integrasi data antara SIAKAD dan E-Learning memang perlu dilakukan guna mengatasi hal tersebut.

SOAP (Simple Object Access Protocol) adalah bahasa markup berdasarkan XML untuk pengiriman pesan antar aplikasi. SOAP berguna sebagai pembungkus yang digunakan untuk bertukar objek data secara online. REST atau Representational State Transfer merupakan salah satu teknik arsitektur perangkat lunak untuk sebuah sistem distribusi seperti World Wide Web. REST tidak memerlukan penguraian XML dan tidak memerlukan header pesan untuk dan dari penyedia layanan. Ini pada akhirnya mendapat manfaat dari pengurangan penggunaan bandwidth. REST memiliki kinerja yang lebih baik daripada SOAP untuk memeriksa permintaan dan tanggapan untuk layanan web (Putra & Putera, 2019). Layanan jaringan REST adalah salah satu metode koneksi yang tepat antar suatu sistem maupun dengan antar jaringan, layanan jaringan REST dapat dijalankan secara sistematis menggunakan terminal cron ke *linux* (Hanani, 2020).

Dalam mengintegrasikan SIAKAD dan *E-learning* tentunya dibutuhkan sebuah metode pengembangan perangkat lunak yang dapat menyederhanakan langkah dalam proses rancang bangun sistem pada UIN Malang. Sudah banyak pada penelitian-penelitian terdahulu yang menjelaskan tentang bagaimana membangun atau mengembangkan suatu perangkat lunak, antara lain metode *Spiral*, *Waterfall*, *Rapid Application Development*, *Agile*, dan masih banyak yang lainnya. Extreme programming merupakan contoh pengembangan perangkat lunak yang menyederhanakan berbagai tahapan pengembangan sistem menjadi lebih efisien, fleksibel dan fleksibel (Fatoni & Dwi, 2016). Metode ini adalah metode Agile yang dipopulerkan oleh Kent Beck. Extreme programming digunakan untuk mengatasi kebutuhan yang tidak jelas dan sering berubah (*vague and volatile requirements*). Extreme programming (XP) digunakan untuk mengatasi kebutuhan yang berubah dengan cepat bahkan ketika proses pengembangan hampir selesai. Selain itu XP juga dimunculkan untuk mengatasi *requirements* yang kurang jelas dari user (Andry, 2019). Metode dalam pengerjaan sistem ini menggunakan Personal Extreme Programming (PXP). PXP digunakan untuk membuat sistem pada programmer tunggal supaya lebih cepat dan mudah dalam membuat sistem sesuai kebutuhan yang ada. Metode PXP ini cocok karena pada penelitian ini hanya dilakukan oleh satu orang programmer dan tidak bergantung pada sebuah tim. Perubahan-perubahan yang terjadi secara cepat mengenai kebutuhan klien akan sistem yang dibangun juga menambah nilai tambah untuk menggunakan metode PXP dikarenakan PXP selalu siap akan perubahan yang terjadi selama dalam masa pengembangan. Berbeda dengan

metode waterfall yang harus dilakukan secara urut dan tidak bisa kembali ke tahap-tahap sebelumnya. Selain itu, penelitian ini memiliki skala proyek menengah yang cocok untuk diimplementasikan dengan menggunakan PXP. Selain mengimplementasikan XP dalam perancangan perangkat lunak, perlu adanya metode pengujian dalam penelitian pengembangan perangkat lunak ini, yang dimana mampu menganalisis efektivitas serta fungsionalitas aplikasi yang sudah dibangun. Pengujian manual adalah metode pengujian di mana pemeriksa menyiapkan kasus uji dan melakukan uji kasus untuk mendeteksi kesalahan dalam perangkat lunak. Pengujian otomatis adalah skrip pengujian yang mengikuti langkah-langkah skrip pengujian dalam bahasa pemrograman apa pun untuk bekerja secara independen. *Automated testing* juga dapat menemukan bug lebih banyak daripada manual testing, serta lebih cepat dan efisien (Min et al., 2020). Untuk mengevaluasi kualitas sistem pada penelitian ini akan digunakan metode *Automated Testing* yaitu suatu metode yang dapat digunakan untuk mengevaluasi efektivitas dan kinerja dari aplikasi yang akan dibangun secara cepat. Sehingga hasil dari percobaan tersebut dapat dijadikan sebagai indikator kualitas dari sistem yang dikembangkan dan dapat dijadikan sebagai alat untuk perbaikan secara terus menerus.

Dengan mengacu pada latar belakang permasalahan yang telah dijelaskan dan berdasarkan hadits di atas, maka penulis mengajukan penelitian yang berjudul “Implementasi *Personal Extreme Programming* dalam Pengembangan Integrasi Sistem Informasi Akademik dan *E-learning*” dengan studi kasus Universitas Islam Negeri Maulana Malik Ibrahim Malang.

1.2 Rumusan Masalah

Berdasarkan uraian masalah pada latar belakang di atas, maka terdapat rumusan masalah dalam penelitian ini yaitu bagaimana “Implementasi *Personal Extreme Programming* dalam Pengembangan Integrasi Sistem Informasi Akademik dan *E-learning*” ?

1.3 Tujuan Penelitian

Berdasarkan penjelasan sebelumnya mengenai latar belakang dan rumusan masalah, maka tujuan dari penelitian ini adalah “Implementasi *Personal Extreme Programming* dalam Pengintegrasian Sistem Informasi Akademik dan *E-learning*”.

1.4 Batasan Masalah

Pada penelitian ini terdapat beberapa batasan masalah, adapun batasan-batasan pada penelitian ini adalah, sebagai berikut :

1. Pengembangan sistem ini berfokus pada penggunaan *Personal Extreme Programming* dalam pengintegrasian. Integrasi antara Sistem Informasi Akademik atau data akademik dengan data media pembelajaran *e-Learning*. Pada bagian integrasi pengelolaan jurnal perkuliahan, absensi dan nilai, dengan asumsi dosen menggunakan media pembelajaran *e-Learning* untuk pemberian materi, absensi, dan ujian.

1.5 Manfaat Penelitian

1. Penelitian ini bisa menjadi referensi untuk penelitian selanjutnya dalam hal implementasi *Personal Extreme Programming* pada pengembangan integrasi Sistem Informasi Akademik atau data akademik dan data media pembelajaran *E-learning*.
2. Pada rancangan program ini, bertujuan agar terciptanya sebuah sistem integrasi pada Sistem Informasi Akademik dengan *e-learning* agar memudahkan dosen dalam melakukan rekap data dari *e-learning*.

1.6 Sistematika Penulisan

Penelitian ini ditulis dengan sistem penulisan yang disusun sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisi beberapa subbagian yang menjelaskan tentang latar belakang masalah, pemecahan masalah, penelitian, manfaat penelitian, keterbatasan masalah, dan pengetahuan tentang sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini menjelaskan penelitian sebelumnya yang bertentangan dengan penelitian yang didefinisikan oleh penulis. Selain itu juga memuat teori dan konsep yang relevan dari berbagai sumber literatur yang berkaitan dengan pokok bahasan penelitian.

BAB III DESAIN DAN IMPLEMENTASI

Bab ini berisi tentang penjabaran dari pembuatan sistem, serta perancangan dari integrasi sistem informasi akademik dengan media pembelajaran elektronik atau *E-learning* pada UIN Malang menggunakan metode pengembangan sistem *Personal Extreme Programming*.

BAB IV IMPLEMENTASI SISTEM

Bab ini berisi tentang hasil dan implementasi dari sistem yang telah dibuat. Serta tes yang dilakukan dari mana kesimpulan dapat ditarik.

BAB V PENUTUP

Bab ini berisi tentang kesimpulan yang didapat dari hasil implementasi sistem serta beberapa saran untuk pengembangan penelitian di masa yang akan datang.

BAB II

TUJUAN PUSTAKA

2.1 Penelitian Terdahulu

Pada penelitian yang berjudul "Integrasi sistem informasi akademik dan e-learning UIN Maulana Malik Ibrahim Malang berbasis web service REST". Penulis berusaha mengintegrasikan antara data E-learning dan SIAKAD. Data E-Learning masih dimasukkan secara manual dan belum terjadwal secara otomatis. Web service REST (REpresentational State Transfer) merupakan metode yang dipilih penulis untuk mengintegrasikan antar sistem dan antar database. Hasil penelitian menunjukkan bahwa integrasi data antara SIAKAD dan E-Learning di UIN Malang dapat dilakukan menggunakan web service REST. Data di SIAKAD akan sama dengan data di E-Learning karena data sudah tersinkronisasi. Data tersebut adalah data mahasiswa, dosen, jurusan, fakultas dan matakuliah. Web service REST dapat dijalankan secara terjadwal dan otomatis melalui sistem dengan menggunakan crontab yang ada pada sistem operasi Linux atau FreeBSD, sehingga data tidak dimasukkan secara manual lagi. Dalam penelitian ini masih terdapat kekurangan, yaitu belum adanya integrasi data nilai. Penelitian selanjutnya disarankan untuk menambahkan integrasi data nilai tersebut, dimana data nilai yang ada di E-Learning yang menggunakan Moodle dapat dimasukkan ke dalam data nilai SIAKAD (Hanani, 2020).

Dalam penelitian yang berjudul "Perancangan dan Pembuatan Mobile Learning Interaktif Berbasis Android Dengan Metode Personal Extreme Programming", mengangkat sebuah masalah tentang kurang interaktifnya sebuah media

pembelajaran M-Learning. M-Learning kurang interaktif karena stagnan dan koneksi internet menjadi kendala saat menggunakan internet nirkabel. *Smart client* dapat diimplementasikan sebagai alat alami. Native tools dapat melakukan penyimpanan suatu data pada perangkat agar aktivitas sistem bisa bekerja meski tidak terhubung dengan koneksi data atau internet. Penggunaan pengembangan sistem Extreme Programming memudahkan pengembang yang bekerja sebagai user software atau aplikasi perangkat lunak untuk memperkirakan dan mengevaluasi semua tujuan utama sistem yang sedang dikembangkan. Maka dari itu, semua rilis diproduksi dalam waktu yang relatif singkat dari perencanaan awal (H. Rizal et al., 2014).

Penelitian yang berjudul “Perancangan Aplikasi *E-Learning* Berbasis Web Dengan Metode Extreme Programming Bagi Siswa (Studi Kasus: SDN Situ Gede 05 Bogor)”. Penulis berusaha memudahkan pembelajaran jarak jauh pada SDN Situ Gede 05 Bogor melalui penelitian ini. Aplikasi *e-learning* ini berbasis web dan menggunakan bahasa pemrograman *PHP* dan *MySQL* serta menggunakan metode program lanjutan. Pada aplikasi *e-learning* ini terdapat tiga user administrator, guru, siswa. Langkah-langkah yang dilakukan untuk mengimplementasikan kegiatan *E-Learning* dengan menggunakan pendekatan *Extreme Programming* adalah merancang, memodelkan sistem dan desain menggunakan *Unified Modeling Language* (UML) yang meliputi *Application case*, *Groups*, *Images 'o Operation* dan *Design Pictures*, hingga membuat storage plan menggunakan Citra Hubungan Perusahaan dan uji (test). Hasil ini dikarenakan adanya aplikasi yang mendukung kegiatan belajar mengajar yang

dapat ditemukan secara online. Aplikasi e-learning ini membantu guru dan siswa untuk mendemonstrasikan pengajaran (Dewi, 2021).

Penelitian yang berjudul “Perancangan Integrasi Sistem Informasi Akademik dengan E-Learning (Studi Kasus : Program Studi XYZ)”. Penulis mengeksplorasi alternatif untuk mengintegrasikan Sistem Informasi Akademik atau manajemen informasi akademiak dengan media pembelajaran *E-learning*. Ada tiga metode dalam percobaan ini, yaitu Sistem Informasi Akademik atau data akademik dengan data media pembelajaran *E-learning* dalam satu proses baru, Sistem Informasi Akademik dan *E-learning* dalam satu aplikasi sekaligus, Sistem Informasi Akademik dan *E-learning* dengan aplikasi tambahan. Penelitian ini menghasilkan sebuah alternatif dimana Sistem Informasi Akademik atau data akademik akan diintegrasikan dengan data media pembelajaran *e-learning* dengan aplikasi tambahan merupakan pilihan yang tepat, karena program integrasi berada di antara kedua sistem, sehingga lebih aman ketika digunakan. sistem. Pengguna juga tidak akan merasakan dampak perubahan karena sudah familiar dengan kedua sistem sehingga tidak perlu melakukan sosialisasi kembali (Bernadhi & Saptadi, 2016).

Penelitian terdahulu yang berjudul “*Comparative Analysis of Software Engineering Models from Traditional to Modern Methodologies*”, peneliti meneliti tentang perbandingan metode pengembangan perangkat lunak tradisional dengan modern. Penelitian ini berkaitan dengan metodologi yang mengkaji siklus hidup perangkat lunak melalui model pengembangan, yang dikenal sebagai siklus hidup pengembangan perangkat lunak. Mewakili model

tradisional yaitu *waterfall*, *Iteration*, *Spiral models* dan pengembangan modern seperti metodologi *Extreme programming*, *Scrum*, *Feature Driven Development*, *Component based software development methodologies*, dll. Semua model ini juga memiliki kelebihan dan kekurangan. Oleh karena itu, tujuan utama dari penelitian ini adalah untuk merepresentasikan model pengembangan perangkat lunak yang berbeda dengan menunjukkan praktik baik dan buruk dari masing-masing model.

Tabel 2. 1 Perbandingan Metode

<i>Parameters/Methodologies</i>	<i>Traditional</i>	<i>Agile</i>
<i>Requirements</i>	<i>Defined before implementation</i>	<i>Acquired iteratively</i>
<i>Rework cost</i>	<i>High</i>	<i>Low</i>
<i>Flexibility of Design</i>	<i>Hard to achieve</i>	<i>Easily achievable</i>
<i>Development directio</i>	<i>Fixed</i>	<i>Flexible</i>
<i>Fault detection</i>	<i>Problematic</i>	<i>Easy</i>
<i>Testing</i>	<i>After coding</i>	<i>On every iteration</i>
<i>Customer involvement</i>	<i>Low</i>	<i>High</i>
<i>Developers skill</i>	<i>Nothing in particular</i>	<i>Interpersonal skills & basic business knowledge</i>
<i>Project type</i>	<i>Large-scaled</i>	<i>Low to Medium scaled</i>
<i>Reusability</i>	<i>Hard to achieve</i>	<i>Easily achievable</i>
<i>Customer Satisfaction</i>	<i>Low</i>	<i>High</i>
<i>Performance</i>	<i>Now High</i>	<i>High</i>

Kelebihan dan kekurangan dari semua model ini telah dijelaskan. Metode tradisional digunakan untuk proyek yang sangat kritis di mana persyaratan tidak sering berubah, persyaratan terbatas dengan fitur terbatas, sejumlah besar

pengembang. Metodologi modern digunakan untuk proyek kritis yang agak rendah di mana persyaratan sering berubah, Fleksibilitas desain, peningkatan kualitas, Pengiriman berulang dan bertahap, Peningkatan Kinerja, Kemampuan deteksi kesalahan yang mudah. (Kumar & Bhatia, 2014)

Pada penelitian yang berjudul “RESTful Web Service untuk Integrasi Sistem Akademik dan Perpustakaan Universitas Perjuangan”. Pemanfaatan teknologi web service untuk menerapkan rencana REST pada sistem informasi dan sistem informasi perpustakaan. Desain REST mampu mengintegrasikan kedua sistem tersebut. Agar proses entri data dan verifikasi hanya dilakukan satu kali, mengatasi terjadinya duplikasi data dan mengurangi fungsi entri data. Sistem pertukaran data antar sistem menggunakan format JSON non-fungsional, yang memungkinkan akses ke bahasa pemrograman, arsitektur, atau sistem operasi yang berbeda (Rizal & Rahmatulloh, 2019).

Dari penelitian diatas, sudah banyak penggunaan *XP* dalam pengembangan suatu pengelolaan data akademik menggunakan sistem informasi akademik untuk memudahkan manajemen informasi akademik dan juga media pembelajaran elektronik *e-learning*. Pada penerapan integrasi sistem sendiri, sudah banyak juga penerapan integrasi sistem menggunakan arsitektur *REST*. Hal inilah yang mendorong peneliti untuk melakukan penelitian pengembangan sistem menggunakan *Personal Extreme Programming* dengan saling terintegrasi agar sistem informasi akademik bisa saling terjembatani dengan *e-learning* menggunakan *web service REST*.

2.2 Pengembangan Aplikasi

Pengembangan aplikasi kerap disalahartikan sebagai kegiatan seorang programmer melakukan sebuah aktifitas yaitu coding, proses menulis kode padahal sebenarnya pengembangan aplikasi lebih dari hanya melakukan proses menulis kode. Pengembangan Aplikasi adalah serangkaian proses yang dilakukan seorang atau sekelompok programmer untuk membuat rangkaian proses dari rancangan awal aplikasi, membuat prototype aplikasi, implementasi, pengujian akhir aplikasi hingga aplikasi selesai dan siap untuk digunakan. Dalam pengembangan aplikasi ada beberapa tahap yang telah dirancang sebagai syarat agar sebuah aplikasi dapat dibuat secara terstruktur dan terancang dengan baik. Tahap tersebut adalah pengumpulan informasi, perencanaan, pengembangan, dan perawatan. Tahap ini merupakan tahap yang penting untuk programmer dalam membangun aplikasi. Dalam pelaksanaannya ada beberapa metode pengembangan aplikasi yang kerap digunakan oleh para programmer yaitu seperti metode waterfall, metode spiral, metode XP (Extreme Programming), metode Kanban, agile development, metode prototype dan metode scrum. (Raharjana, 2017)

2.3 Integrasi Aplikasi

Sistem integrasi (integrated system) merupakan sebuah rangkaian proses untuk menghubungkan beberapa sistem komputerisasi dan software aplikasi, baik secara fisik maupun secara fungsional. Sistem terintegrasi akan menggabungkan komponen sub-sub sistem ke dalam satu sistem dan menjamin fungsi-fungsi dari sub sistem tersebut sebagai satu kesatuan sistem. Konsep Integrasi sistem adalah yaitu suatu konsep sistem yang dapat saling berhubungan satu dengan yang lain

dengan berbagai cara yang sesuai dengan keperluan. Hal ini sangat bermanfaat bila suatu data dalam file suatu sistem diperlukan juga oleh sistem yang lainnya atau output suatu sistem menjadi Input sistem lainnya. Pengintegrasian sistem informasi merupakan salah satu konsep kunci dari sistem Informasi Manajemen. Berbagai sistem dapat saling berhubungan satu dengan yang lain dengan berbagai cara yang sesuai dengan keperluannya. Aliran informasi diantara sistem sangat bermanfaat bila data dalam file suatu sistem diperlukan juga oleh sistem yang lainnya, atau output suatu sistem menjadi input bagi sistem lainnya. (Nurherliza, 2019)

2.4 Sistem Informasi Akademik

Sistem merupakan suatu kesatuan dengan komponen atau komponen-komponen yang dihubungkan bersama untuk mengendalikan aliran informasi, peralatan atau tenaga untuk mencapai suatu tujuan (Eriyatno, 2012). Informasi merupakan sebuah pesan maupun kumpulan pesan dengan urutan simbol, atau makna yang dapat dijelaskan dalam pesan atau dalam kumpulan pesan. Data dapat direkam atau ditransfer. Ini dapat dilaporkan sebagai sinyal, atau sebagai gelombang berbasis sinyal (Agustin, 2019). Sistem informasi merupakan komponen dari satu komponen yang saling berhubungan dengan komponen lainnya, hal ini bertujuan untuk menghasilkan informasi dalam bidang tertentu. Saat ini terdapat sejumlah sistem informasi di hampir semua bidang, dan salah satunya ada pada bidang pendidikan.

Bidang studi adalah tempat diselenggarakannya silabus atau kajian untuk memberikan dan mengembangkan pengetahuan tentang pendidikan yang dapat

dikelola oleh suatu lembaga pendidikan (Liatmaja & Wardati, 2013). Sistem Informasi Akademik adalah suatu sistem yang dirancang untuk mengolah data pendidikan, dengan perangkat lunak dan perangkat keras agar seluruh sistem kegiatan pendidikan dapat berjalan dengan lancar, agar dapat menjadi informasi yang berguna dalam pengelolaan pendidikan dan pengambilan keputusan. Sistem tersebut bertujuan untuk mempermudah penyelenggaraan pendidikan, sehingga lembaga pendidikan dapat memberikan informasi yang bermanfaat dan praktis (Homaidi, 2016).

2.5 E-Learning

Selain pada kegiatan pengelolaan administrasi akademik, proses kegiatan belajar mengajar juga sudah mulai menuju ke arah yang lebih moderen. Pada masa pandemi covid 19, para mahasiswa dan dosen tidak diperbolehkan untuk datang maupun melakukan kegiatan belajar mengajar di kampus secara tatap muka. Ini membuat beberapa Universitas atau lembaga pendidikan di Indonesia mulai menerapkan pembelajaran secara daring. Pada umumnya mahasiswa dan dosen akan memanfaatkan fasilitas digital yang berupa platform *E-learning* yang sudah disediakan oleh kampus, untuk kegiatan belajar mengajar. *E-learning* adalah pembelajaran pada semua jenjang pembelajaran formal dan nonformal yang menggunakan jaringan komputer (baik web maupun intranet) untuk kegiatan belajar dan mengajar (Qowaid et al., 2020).

Pembelajaran elektronik mempersingkat waktu belajar dan menghemat biaya pembelajaran. *E-learning* memfasilitasi interaksi antara siswa dan materi belajar, siswa dan guru/guru/instruktur dan teman sekelas. Siswa dapat berbagi informasi

dengan orang lain dan dapat mengakses materi pembelajaran setiap saat dan sering, dalam hal ini siswa dapat meningkatkan sumber belajar mereka (Siahaan, 2003). Dalam media pembelajaran elektronik *e-learning*, kehadiran seorang pengajar atau instruktur secara bertahap akan berkurang atau hilang karena peran guru akan digantikan oleh komputer. Imbas baik dengan munculnya media belajar elektronik *e-learning* akan memudahkan banyak hal, antara lain:

- a. Melakukan penyempurnaan-penyempurnaan terbaru dalam materi pembelajaran yang menjadi tanggung jawab utamanya sejalan dengan kemajuan ilmu pengetahuan terkini.
- b. Fokus melakukan pengembangan diri dan memaksimalkan penelitian pengajar guna meningkatkan wawasan.
- c. Mengelola atau memantau kegiatan parasiswa dalam belajar.

2.6 Web Service REST

Web services adalah sebuah teknologi yang dapat menggabungkan program satu dengan yang lainnya. Pada umumnya web service ini digunakan untuk mengkomunikasikan dua sistem aplikasi tanpa melihat dari sistem arsitektur maupun suatu sistem teknologi yang digunakan. Web Service memberikan fleksibilitas untuk membangun komunikasi antara sistem atau perangkat yang terpisah melalui internet. sudah banyak Web Service diterbitkan di seluruh internet yang dapat digunakan sesuai dengan kebutuhan konsumen (Bora & Bezboruah, 2015).

Representational State Transfer (REST) adalah kerangka kerja perangkat lunak untuk aplikasi terdistribusi seperti jaringan. REST telah berkembang menjadi

contoh utama paket layanan website periode ini. REST diciptakan oleh Roy Fielding pada tahun 2000 dalam jurnal medisnya (Triawan & Alipudin, 2021) REST dikabarkan lebih efisien, karena tidak menggunakan simbol tambahan seperti XML. Formulir respons di REST biasanya menggunakan XML, JSON, atau bentuk eksplisit lainnya. Ini mudah bagi penerima jawaban untuk membaca dan memahami.

Keuntungan menggunakan REST adalah mudah, karena transaksi berbasis REST menggunakan Internet Hypertext Transfer Protocol (HTTP) yang banyak digunakan dan standar, contoh konfigurasi REST adalah kolaborasi berbasis REST dengan melakukan semua komunikasi pemerintah menggunakan kode status HTTP standar. Untuk penjelasannya, 404 menyatakan bahwa perangkat yang diminta tidak ditemukan, kode 401 menyatakan bahwa permintaan tidak disetujui, kode 200 menyatakan bahwa semuanya OK, dan 500 menyatakan bahwa permintaan kesalahan tidak dapat ditemukan di server (R. Rizal & Rahmatulloh, 2019).

2.7 Agile Development Method

Agile development method adalah pengembangan perangkat lunak yang mempunyai rentang waktu pendek serta bisa beradaptasi dengan adanya perubahan yang mungkin akan terjadi selama proses dari pembangunan sistem tersebut berlangsung. Salah satu metode yang bisa dikatakan sebagai agile development method apabila dia bersifat interactive, incremental, flexibility, responding to change, dan lightweight. Ada beberapa hal yang penting dari Agile development method tersebut adalah pendeskripsian dari sistem yang bisa

dimengerti dengan mudah dan dapat digunakan oleh pihak client. Mengenai penjelasan yang dimaksud agile development method diatas yakni:

a. Iterative

Iterative ini bisa diimplementasikan dalam proses pembangunan sebagai penyelesaian perangkat lunak dengan cara pendekatan secara periodik guna menemukan solusi dari permasalahan yang diperoleh selama proses pembangunan sistem.

b. Incremental

Incremental ini pada pengembangan perangkat lunak tentu kebutuhan fungsionalitas pada sistem dibagi menjadi beberapa bagian yang lebih kecil. Pada setiap pengembangan fungsionalitas pada perangkat lunak akan semakin bertambah sehingga akan menjadi suatu perangkat lunak yang utuh.

c. Flexibility

Flexibility ini apabila sebuah perangkat lunak dikembangkan, sistem tersebut bisa beradaptasi dengan kondisi tersebut dikarenakan pengembang sudah melibatkan pihak client secara langsung terhadap proses pembangunan perangkat lunak. Oleh karena itu, pengembang harus siap ketika adanya perubahan yang akan terjadi.

d. Responding to change

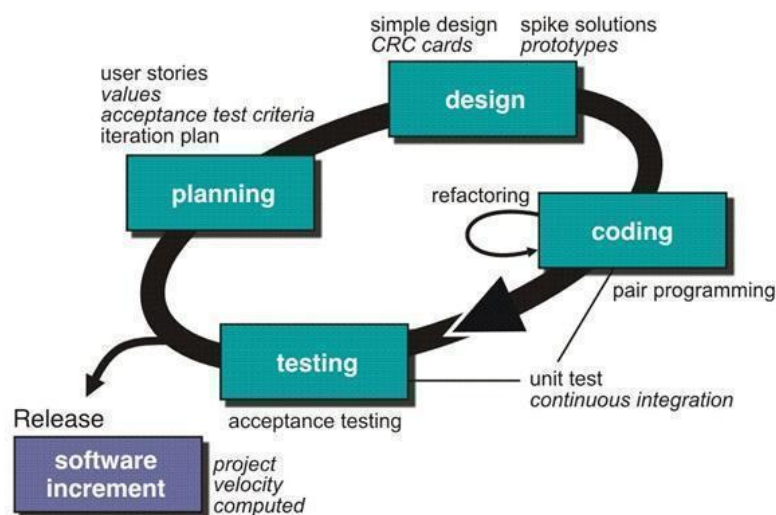
Responding to change ini memperbolehkan perubahan terkait requirement, code, and design dari perangkat lunak yang sudah dibuat.

e. Lightweight

Lightweight ini pengembang dapat meminimalkan dokumentasi requirement dan segmen-segmen yang tidak perlu selama pembuatan perangkat lunak sedang berlangsung dikarenakan situasi ini dapat memperlambat kecepatan dan waktu dalam pembuatan perangkat lunak.

2.8 Extreme Programming

Sistem XP merupakan salah satu cara untuk membuat perangkat lunak yang gesit untuk meningkatkan sistem. XP (Extreme Programming) menerapkan sejumlah aturan dalam praktiknya, yang meliputi perencanaan, produksi, pencatatan, pengujian (A. Fatoni & Dwi, 2016).



Gambar 2.1 Alur Extreme Programming

1. Planning

Pada tingkat desain, desain perangkat lunak akan dilakukan berdasarkan cerita pengguna. Cerita pengguna menggambarkan karakteristik dan fungsionalitas yang dibutuhkan pada program sistem perangkat lunak. Pembeli dan tim XP akan bekerja sama untuk memutuskan bagaimana mengatur cerita untuk komponen yang akan disertakan dalam aplikasi berikutnya atau dalam rilis pemutakhiran untuk dibangun oleh tim XP.

2. Design

XP menggunakan *Class Responsibility Collaboration card*, untuk mengenali dan mengelola object oriented class yang sesuai dengan software increment atau perancangan. Desain akan berisi semua penerapan dari user stories tanpa adanya perubahan sama sekali. Jika ada kendala dalam pembuatan cerita, XP akan menyarankan untuk membuat desain prototype. Ini adalah jawaban spike, prototipe yang diturunkan akan dilakukan dan dievaluasi. XP kemudian akan merekomendasikan refactoring, sebuah proses perbaikan yang secara teknis lebih maju. Refactoring adalah metode mengubah perangkat lunak sistem dengan cara yang tidak mengubah kode eksternal dari kode, yang meningkatkan struktur internalnya.

3. Coding

Sebelum melakukan pemrograman, unit test dilakukan untuk setiap kasus yang akan dibuat. XP merekomendasikan agar dua orang bekerja sama dalam pemrograman berpasangan untuk memastikan pengeditan waktu-

nyata waktu-nyata dan jaminan kualitas waktu-nyata. Setelah tes unit selesai, pengembang fokus pada apa yang akan dilakukan untuk lulus tes unit. Bagian ini akan mengacu pada desain sebelumnya. Karena tes unit dilakukan terlebih dahulu, perancangan harus dilakukan untuk lulus tes unit yang dilakukan.

4. Testing

Langkah ini akan berlaku untuk unit test yang telah diuji sebelumnya. Unit test yang telah dibuat harus diimplementasikan menggunakan suatu framework dan diatur ke dalam universal testing suite, integrasi dan validasi sistem dapat dilakukan setiap hari. Customer test (acceptance test) dilakukan oleh customer dan fokus pada keseluruhan fitur dan fungsional sistem. Acceptance test diperoleh dari customer stories yang telah diimplementasikan sebagai bagian dari software release.

2.9 Personal Extreme Programming

Metode Personal Extreme Programming (PXP) merupakan pengembangan sebutan bagi metode Extreme Programming (XP) yang masih memiliki banyak kesamaan diantara keduanya namun perbedaan sebutan tersebut dikarenakan metode PXP hanya dikembangkan oleh satu orang saja, sementara pada metode XP dapat dikembangkan oleh lebih dari dua orang pengembang. Meskipun secara garis besar kedua metode ini sama namun tetap saja metode ini tidak dapat digunakan untuk pengembangan sistem berskala besar.

Walaupun metode PXP termasuk dalam pengembangan perangkat lunak secara personal namun harus memiliki iterative, fleksibel ,dan responsif terhadap

pengembangan perangkat lunak seperti halnya dengan metode Extreme Programming atau bisa disebut dengan XP, metode tidak dapat digunakan untuk pengembangan perangkat lunak berskala besar.

Metode PXP ini awal mulanya diterapkan pada tahun 2009 oleh Dzhyrov dengan penelitian yang bertema “Personal Extreme Programming An Agile Process For Autonomus Development”. Pada keadaan ini metode Extreme Programming tidak bisa dikatakan sama ataupun mirip secara keseluruhan dengan Personal Extreme Programming dikarenakan pada tahap implementasi kedua metode ini memiliki 6 hal yang bisa diterapkan pada metode PXP dan XP yakni :

1. Continous Integration
2. Simple Design
3. Small Release
5. Refactoring
6. Test Driven Development
7. Spike Solution

Sedangkan pada tahap pelaksanaan yang lain yang terdapat pada Extreme programming apabila ingin diterapkan pada metode Personal Extreme Programming harus dimodifikasi terlebih dahulu agar pada tahap pelaksanaan tersebut bisa berjalan pada metode PXP.

2.10 Automated Testing

Tes komputer adalah desain program (skrip pengujian) yang mensimulasikan langkah-langkah pengujian manual dalam bahasa pemrograman apa pun dengan bantuan beberapa alat otomatis eksternal (Min et al., 2020). Berbeda dengan

Pengujian Manual Black Box, dalam penelitian ini alat pengujian otomatisasi, Katalon. Katalon digunakan dalam aplikasi eksperimental dalam kasus eksperimental. Uji kasus ini dilakukan dengan perekaman ulang dan pemutaran, yang berarti perlu untuk merekam setiap respons dalam satu peristiwa dalam aplikasi yang digunakan (Herlinda et al., 2019). Tujuannya untuk menganalisis tentang efektivitas aplikasi yang sudah dibangun.

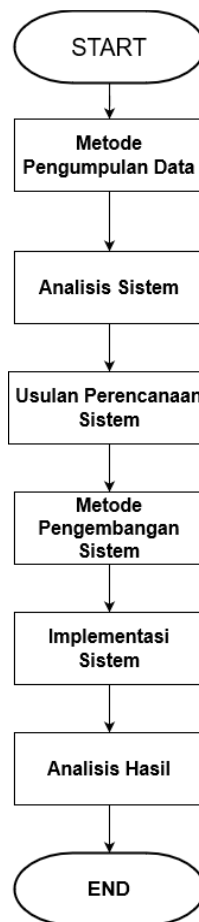
Proses pengujian dilakukan dengan menggunakan alat otomatisasi yaitu Katalon Studio. Alat ini adalah alat pengujian sumber terbuka yang dikembangkan oleh Katalon LLC dan dapat bekerja di semua sistem operasi. Penggunaan IDE khusus juga diterapkan pada Katalon Studio. Katalon Studio mempunyai tiga komponen inti guna bertujuan untuk mendukung pengujian pada berbagai platform seperti pengujian web, pengujian API, dan pengujian seluler (Kosasih & Cahyono, 2021).

BAB III

DESAIN DAN IMPLEMENTASI

3.1 Desain Penelitian

Dalam penelitian ini terdapat desain penelitian yang akan dilaksanakan, agar penelitian lebih terencana dan sistematis. Penelitian ini menggunakan jenis penelitian kualitatif, dimana data yang diperoleh bukan berupa angka. Penelitian ini menggunakan metode *Personal Extreme Programming* dengan metode pengujian *Automated Testing*. Dengan tahapan penelitian dalam bentuk flowchart sebagai berikut,



Gambar 3.1 Alur Penelitian

3.2 Metode Pengumpulan Data

3.2.1 Metode Observasi

Observasi adalah suatu metode pengumpulan data dengan cara memfokuskan langsung pada objek yang akan dipelajari. Observasi dilakukan dengan pengamatan atau peninjauan secara langsung di universitas yang akan ditinjau untuk beberapa kegiatan yang berhubungan dengan penelitian. Observasi dilakukan di UIN Malang. Observasi dilakukan untuk mengetahui dengan pasti kegiatan pembelajaran secara daring dan juga pemrosesan data akademik yang berjalan pada UIN Malang.

3.2.1 Metode Studi Literatur

Studi literatur merupakan cara mengumpulkan dan mempelajari informasi yang terkait dengan penelitian ini. Studi literatur dilakukan dengan mempelajari penelitian terdahulu yang berhubungan dengan penelitian ini agar dapat membantu menyelesaikan masalah dan dapat digunakan sebagai tinjauan pustaka. Selain itu penulis juga mendapatkan tinjauan pustaka dari website resmi di internet.

3.3 Analisis Sistem

Pengamatan pada sistem atau melakukan analisis adalah gambaran dari suatu program pada sistem yang bekerja pada UIN Malang. Analisis berfokus pada integrasi Sistem Informasi Akademik guna pengelolaan data akademik dengan media pembelajaran elektronik E-learning. Sistem yang digunakan saat ini masih ada beberapa fitur yang belum terintegrasi dengan baik antar sistem. Analisis

sistem ini bertujuan untuk mengetahui lebih jelas bagaimana cara kerja sistem saat ini untuk pengembangan sistem baru agar efisien.

3.3.1 Evaluasi Sistem Saat Ini

Berdasarkan hasil analisa penulis terhadap sistem yang sedang berjalan, dalam Sistem Informasi Akademik dan *E-Learning* pada UIN Malang belum terintegrasi dengan baik. Mengetahui sistem yang ada, penulis sampai pada kesimpulan bahwa:

Tabel 3.1 Evaluasi Sistem

Permasalahan	User
Dosen harus melakukan penginputan jurnal perkuliahan pada sistem secara satu persatu.	Dosen
Dosen harus melakukan penginputan absensi pada kedua sistem secara satu persatu.	Dosen
Dosen harus melakukan penginputan nilai pada kedua sistem secara satu persatu.	Dosen

3.4 Usulan Perencanaan Sistem

Perancangan sistem adalah tahapan analisis siklus pengembangan sistem yang ditetapkan dengan kebutuhan fungsional dan penyusunan rencana implementasi yang menggambarkan bagaimana merancang suatu sistem yang dapat diintegrasikan ke dalam satu kesatuan berupa gambar, rencana, dan sketsa atau

layout, dan meliputi perangkat keras dan konfigurasi. Komponen sistem perangkat lunak.

3.4.1 Tujuan Perencanaan Sistem

Perencanaan sistem ini menggambarkan prosedur sistem yang diusulkan dari sistem yang sudah ada sebelumnya. Adapun tujuan perencanaan sistem yang diusulkan yaitu, merancang integrasi Sistem Informasi Akademik atau data akademik dengan data media pembelajaran *e-Learning* Universitas Islam Negeri Maulana Malik Ibrahim Malang.

3.4.2 Perencanaan Yang Diusulkan

Berdasarkan analisis dari sistem yang sudah ada, perencanaan yang diusulkan dari sistem yang akan dibuat adalah lebih mengintegrasikan atau menjembatani antara data SIAKAD dan juga data E-learning. Data sistem yang nantinya akan dijembatani yaitu data User yang berkaitan dengan absensi, nilai dan jurnal perkuliahan pada kedua belah sistem.

3.4.3 Evaluasi Sistem Yang Diusulkan

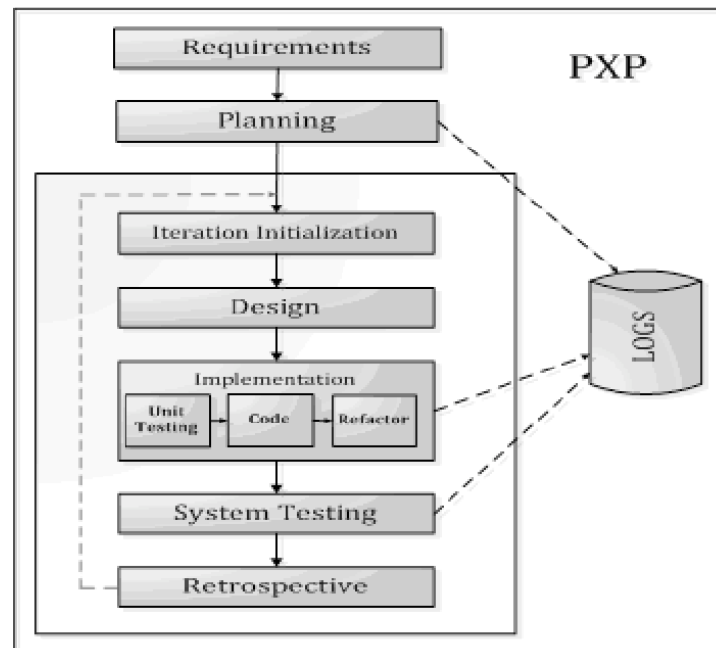
Berdasarkan analisa oleh penulis terhadap sistem yang diusulkan dengan sistem saat ini, setidaknya sudah lebih efektif dari sistem yang sebelumnya.

Tabel 3.2 Perbandingan Sistem Lama dan Sistem Baru

Sistem Lama	Sistem Baru
Dosen harus melakukan penginputan jurnal perkuliahan dengan satu persatu pada SIAKAD.	Dosen bisa melakukan penginputan jurnal kuliah tanpa harus menginput satu persatu pada sistem.
Dosen harus melakukan rekap dan juga penginputan absensi pada SIAKAD.	Dosen bisa melakukan penginputan nilai secara bersamaan tanpa harus melakukan rekap terlebih dahulu.
Dosen harus melakukan rekap dan juga penginputan nilai pada SIAKAD.	Dosen bisa melakukan penginputan absensi secara bersamaan tanpa harus melakukan rekap terlebih dahulu.

3.5 Metode Pengembangan Sistem

Sub bab ini akan menjelaskan mengenai langkah langkah yang akan dicapai selama penelitian ini dilakukan. Pengembangan sistem menggunakan metode Personal Extreme Programming (PXP). PXP terbagi dalam beberapa tahapan pengembangan diantaranya yaitu: Requirement, Planning, Iteration Initialization, Design, Implementation, Testing, dan Retrospective. Fase pengembangan PXP dapat dilihat pada gambar di bawah Pengembangan pengintegrasian pada penelitian ini menggunakan *API* dengan bahasa pemrograman *PHP*.



Gambar 3.2 Alur Personal Extreme Programming

Tahap dimulai dari Requirements. Requirements adalah tahap mengidentifikasi segala kebutuhan user mengenai sistem. Pada tahap awal, semua kebutuhan fungsional dan nonfungsional dari klien akan ditampung di dalam sebuah dokumen yang disebut dengan User story. Tahap dilanjutkan dengan fase Planning. Tahap ini merupakan tahap menentukan fungsi fungsionalitas keseluruhan yang akan dikembangkan. User story berisikan tentang kebutuhan fungsional beserta deskripsi dari fungsi tersebut. Dalam menentukan prioritas user story. Pengembang memperkirakan waktu yang dibutuhkan dalam membuat setiap user storynya, beserta dengan tingkat resiko dari story tersebut. User story yang memiliki tingkat resiko yang paling tinggi akan diimplementasikan terlebih dahulu. Selanjutnya adalah Iteration Initialization. Pada tahap ini merupakan tahap awal pada setiap iterasi. Iterasi yang dilakukan akan berfokus pada tugas yang telah ditentukan. Fungsionalitas yang sudah dibentuk pada tahap sebelumnya lalu

dijabarkan dalam bentuk diagram dengan menggunakan UML. Tahap selanjutnya adalah Design, ini merupakan pembuatan sebuah model dari kebutuhan user yang sudah didapatkan pada tahap sebelumnya. Selanjutnya adalah tahap Implementation yang merupakan tahap dimana pengembang akan mengimplementasi desain dari tahap sebelumnya menjadi sebuah kode yang dapat dijalankan di sebuah platform rancangan. Tahap implementation terbagi menjadi tiga tahapan yaitu: Unit testing, Code Generate, dan Code Refactoring. Dilanjutkan dengan Testing. Jenis test yang digunakan dalam penelitian ini adalah menggunakan metode *Automated Testing* dengan *Katalon*.

Terakhir adalah Retrospective. Retrospective adalah aktivitas terakhir dimana kegiatan ini melakukan evaluasi mengenai durasi pengerjaan. Aktivitas ini melihat kesesuaian pengerjaan dengan rencana pengerjaan. Jika terdapat ketidaksesuaian dalam pengerjaan, alasan yang terjadi selama pengembangan akan dipaparkan atau dituliskan di dalam laporan.

3.5.1 Requirements

Pada tahap requirements, pengembang mengumpulkan kebutuhan kebutuhan yang diperlukan akan sistem yang dibangun. Pengumpulan kebutuhan dilakukan dengan cara melakukan wawancara kepada Pusat Teknologi Informasi dan Angkalan Data (PTIPD) UIN Malang. PTIPD menjelaskan bahwa aktivitas Absensi, Input Nilai Perkuliahan, Serta Jurnal Perkuliahan masih dilakukan secara manual. Permasalahan tersebutlah yang menimbulkan kebutuhan akan suatu sistem nantinya. Kebutuhan tersebut dapat dituliskan dalam bentuk user story yang memiliki format seperti “Sebagai <jenis user> saya ingin <melakukan

sesuatu> agar dapat <manfaat dari tindakan tersebut>”. Setelah dilakukan fase kebutuhan, maka dihasilkan 3 user stories yang dapat dilihat pada di bawah.

Tabel 3.3 User Story

Kode	Sebagai Dosen saya...	Sehingga...
Story-01	Ingin melakukan penginputan jurnal perkuliahan tanpa harus menginputnya satu persatu setiap saat	Saya dapat menghemat waktu tanpa harus menginputkan setiap saat.
Story-02	Ingin melakukan penginputan absensi tanpa harus melakukan rekap terlebih dahulu.	Saya dapat menghemat waktu tanpa harus melakukan rekap
Story-03	Ingin melakukan penginputan nilai tanpa harus melakukan rekap terlebih dahulu.	Saya dapat menghemat waktu tanpa harus melakukan rekap

3.5.2 Planning

Pada tahap ini pengembang akan melakukan serangkaian modul yang akan dikerjakan per iterasi yang ada berdasarkan user stories yang telah didapatkan pada tahap sebelumnya. Setelah user story didapatkan, Langkah selanjutnya adalah menentukan perkiraan waktu yang akan dilakukan untuk melakukan task. Perkiraan waktu ini disebut dengan story point. Langkah selanjutnya adalah memberikan prioritas pada setiap user story yang akan dilakukan. Penentuan prioritas ini menerapkan metode prioritas ranking. Prioritas ranking dilakukan dengan cara memberikan kebebasan kepada PTIPD untuk menentukan user story mana yang memiliki tingkat kepentingan yang paling tinggi. Hasil dari metode ini akan mengurutkan user story dari yang tertinggi sampai dengan yang terendah. User story yang memiliki nilai tinggi akan diimplementasikan terlebih dahulu.

Setelah perkiraan waktu pengerjaan dan prioritas dari setiap user story didapatkan, klien dan pengembang akan menentukan user story yang dikerjakan per iterasinya dengan menggunakan velocity. User story akan dilakukan dengan

nilai maksimal velocity tiap iterasinya. Hal tersebut akan terus dilakukan sampai tidak ada iterasi yang tersisa.

3.5.2.1 Value Story

Sebelum memberikan value stories pada fase ini menentukan estimasi waktu yang dibutuhkan pada tiap bagian user stories. Estimasi waktu akan dipilah menjadi sebagian poin tergantung lama pembuatan semakin besar maka poin dari estimasi waktu yang dibutuhkan akan semakin besar, hasil dari estimasi waktu yang didapat bisa dilihat pada tabel berikut:

Tabel 3.4 Story Point dan Estimasi

Kode User Stories	Story Points	Estimasi
Story-01	2	2
Story-02	4	4
Story-03	3	3

Setelah menentukan story point pada setiap user stories, pengembang dan klien akan berdiskusi untuk menentukan setiap prioritas pada setiap user stories yang ada. Kemudian penentuan prioritas akan dilakukan penyortiran terhadap user stories sesuai dengan value dan risk. Value dari user stories ini dapat ditentukan dari nilai bisnis dari user stories. Ada 3 nilai value diantaranya, critical, significant business value, dan nice to have.

1. Critical, adalah adalah user stories Ketika tidak adanya user stories tersebut di dalam sistem yang akan dibangun, maka sistemnya tidak ada artinya. Terdapat 3 user stories yang termasuk dalam kategori ini.

- a. Story-01. User stories ini bertujuan agar sistem dapat terintegrasi dengan data Jurnal Perkuliahan.
 - b. Story-02. User stories ini bertujuan agar sistem dapat terintegrasi dengan data Absensi.
 - c. Story-03. User stories ini bertujuan agar sistem dapat terintegrasi dengan data Nilai.
2. Significant business value, adalah user stories yang sama sekali tidak masuk kedalam critical, tetapi mempunyai nilai bisnis yang signifikan. Dari rencananya yang telah dibuat, tidak terdapat user stories yang termasuk dalam kriteria ini.
 3. Nice to have, adalah user stories yang tidak mempunyai pengaruh terhadap proses bisnis didalam sistem. Dari rencana yang telah dibuat, tidak terdapat user stories yang termasuk dalam kriteria ini.

Tabel 3.5 User Story Value

Kode User Stories	Value
Story-01	Critical
Story-02	Critical
Story-03	Critical

Hasil dari value pada tabel diatas akan dilanjutkan dengan penentuan dari risk ditentukan dari risk index berdasarkan completeness, volatility, dan complexity. Mengenai penentuan completeness ini ditentukan oleh pengembang yakni berlandaskan dari rincian user stories tersebut. Kemudian Volatility ini ditentukan dapat di tentukan apakah ada kemungkinan terjadinya perubahan pada

user stories. Terakhir adalah complexity yakni ditentukan dari tingkat kesulitan pembuatan pada user stories yang ada.

Dari beberapa indeks tersebut menentukan adanya tingkatan dari risk tiap-tiap user stories yang terdiri dari low dengan nilai 0 sampai 1, medium dengan nilai 2 sampai 4, dan high dengan nilai 5 sampai 6. Hasil dari perencanaan iterasi tersebut dapat dilihat pada tabel di bawah.

Tabel 3.6 Risk User Story

Kode User Stories	Risk			
	Completeness	Volatility	Complexity	Total
Story-01	2	1	1	Medium
Story-02	2	1	3	High
Story-03	2	1	3	High

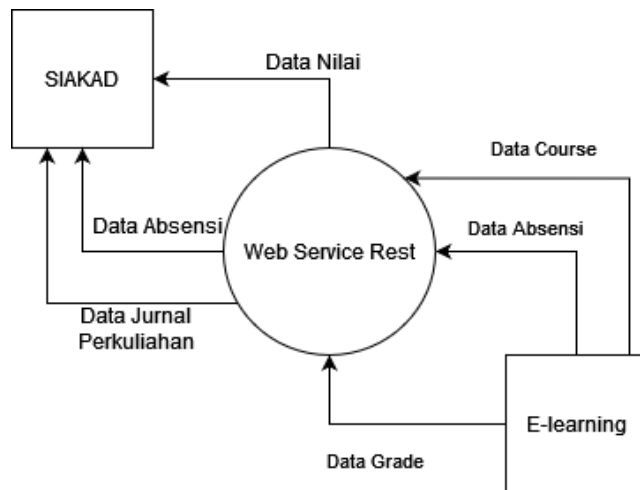
3.5.3 Iteration Initialization

Iteration Initialization merupakan awal pengerjaan pengembangan dari setiap iterasinya. User story yang akan dikerjakan sudah didapatkan pada tahap sebelumnya yaitu tahap planning.

3.5.4 Design

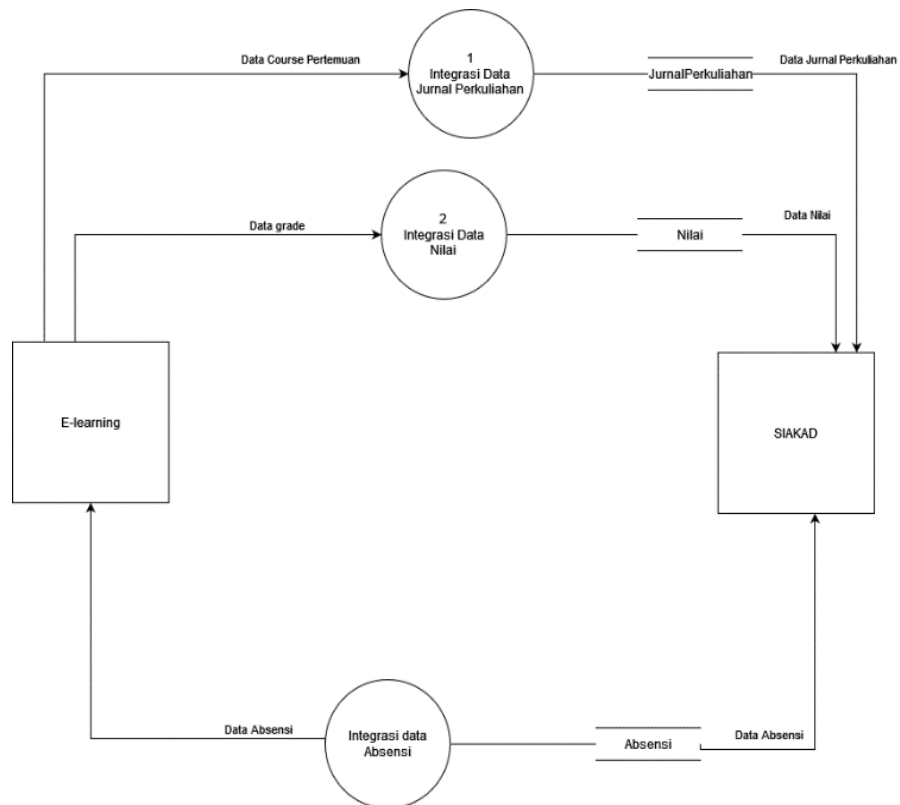
Tahap Design dilakukan dengan cara membuat sebuah model mengenai sistem yang akan dikembangkan selama iterasi. Desain sistem berdasarkan hasil dari proses requirements tanpa adanya design tambahan ketika adanya perubahan yang terjadi selama pengembangan. Desain yang digunakan dalam penelitian ini adalah Activity Diagram, Data Flow Diagram dan juga Arsitektur Sistem. Pada tahapan ini data flow dan Arsitektur system yang akan digunakan nantinya bisa dilihat pada keterangan bawah ini

a. Data Flow Diagram



Gambar 3. 3 DFD Level 0

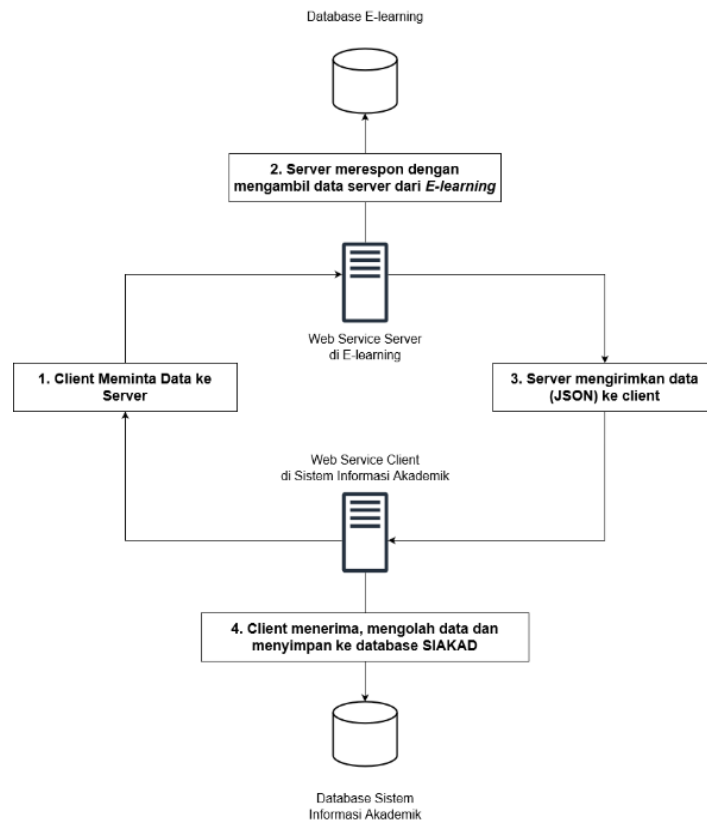
Context Diagram adalah diagram yang menggambarkan bagaimana proses dokumentasi data. Gambar di atas menjelaskan dokumentasi data pada system yang sudah terintegrasi. Pada integrasi sitem yang diusulkan, mengintegrasikan data jurnal perkuliahan, data nilai dan juga data absensi pada *e-learning*.



Gambar 3.4 DFD Level 1

DFD level 1 adalah lanjutan dari context diagram pada gambar sebelumnya. Pada system ini nantinya data dari *e-learning* yang berupa data course akan diintegrasikan dengan data jurnal perkuliahan di SIAKAD. Kemudian setelah itu data grade pada *e-learning* akan diintegrasikan dengan data penilaian di SIAKAD. Lalu data pada absensi di *e-learning* akan diintegrasikan dengan data absensi di SIAKAD.

b. Arsitektur Sistem



Gambar 3.5 Arsitektur Sistem

Pada Arsitektur integrasi data diatas, *client* atau user akan meminta data dari server *e-learning*, kemudian data akan diteruskan ke web service server untuk mengambil data pada database *e-learning*. Data yang sudah diambil akan diubah menjadi *JSON* kemudian data dikirim ke *client* dan dimasukkan di database SIAKAD.

3.5.5 Implementation

Pada tahap Implementasi ini, pengintegrasian akan menggunakan *Api*, Bahasa pemrograman yang akan digunakan adalah *PHP*. Data dari *e-learning*

akan diambil kemudian dirapikan dan dipilah lalu data akan dimasukkan kedalam data base SIAKAD. User dosen nantinya akan tinggal mengakses fitur yang diinginkan maka data akan otomatis terintegrasi tanpa harus merekap terlebih dahulu data.

Kemudian mengenai tahap ini yang terdiri dari unit testing, code, dan refactor. Unit testing adalah pengujian terhadap user story yang terdapat pada masing-masing iterasi kemudian menganalisa hasil dari keluaran yang telah dihasilkan. Selanjutnya ada code yakni perbaikan seluruh code apabila semua unit testing telah diuji sesuai dengan user story yang sudah ada dan berhasil. Dan pada tahap terakhir ada refactor merupakan proses memperbaiki struktur dalam sebuah sistem perangkat lunak tanpa harus mengubah fungsionalitas yang ada pada sistem. Artinya, pada proses refactoring dilakukan perubahan program untuk memperbaiki struktur, meminimalisasi program, atau untuk membuatnya mudah dimengerti. Pada tahap implementation dilakukan sampai tidak ada kekeliruan atau kesalahan, setelah itu akan berlanjut ke tahap system testing.

3.5.6 Testing

Tahapan dalam pengujian nantinya akan dimulai dengan Requirement analysis dimana pengujian siklus dimulai dengan tinjauan umum persyaratan user. Tujuan dari requirement analysis adalah untuk memahami persyaratan yang penting untuk pengujian sistem. Kemudian dilanjutkan dengan Case Design and Development, langkah ini digunakan untuk menemukan apa saja persyaratan komponen yang diperlukan. Lalu tahapan selanjutnya adalah Test Execution, tahapan ini akan melakukan peninjauan pada kode atau program yang ada.

Pengujian akan mengukur kualitas program dari segi kinerja (performance). Tahapan setelah itu adalah Test Closure. Tahap ini menghasilkan ringkasan dari testing yang telah dilakukan. Lalu ditahap terakhir adalah Test Process Analysis yang berisi ringkasan yang akan dianalisis untuk meningkatkan kinerja dari suatu aplikasi dengan menggunakan teknologi baru atau menambahkan fitur-fitur tambahan yang diberi.

3.5.7 Retrospective

Fase terakhir dari setiap iterasi adalah retrospective. Fase ini akan memaparkan kendala kendala apa saja yang terjadi selama pengembangan iterasi berlangsung. Di fase ini juga akan dijelaskan mengenai kesesuaian antara jadwal estimasi pengembangan dan waktu realisasi dari pengembangan. Jika terdapat ketidaksesuaian waktu pengerjaan, hal tersebut dapat menjadi bahan introspeksi pengembang agar hal tersebut dapat dihindari pada iterasi selanjutnya

BAB IV

IMPLEMENTASI SISTEM

Pada bab ini nantinya menjelaskan hasil dan pembahasan terkait penelitian yang telah dilakukan. Pada bab ini juga terdiri dari tiga iterasi yang menjelaskan perihal yang telah terjadi pada perubahan sistem yang telah terjadi selama proses pengembangan yang dikerjakan. Bab ini juga akan menjelaskan apa saja yang terjadi selama proses iteration initialization, design, implementation, sistem testing , dan juga retrospective

4.1 Iterasi Pertama

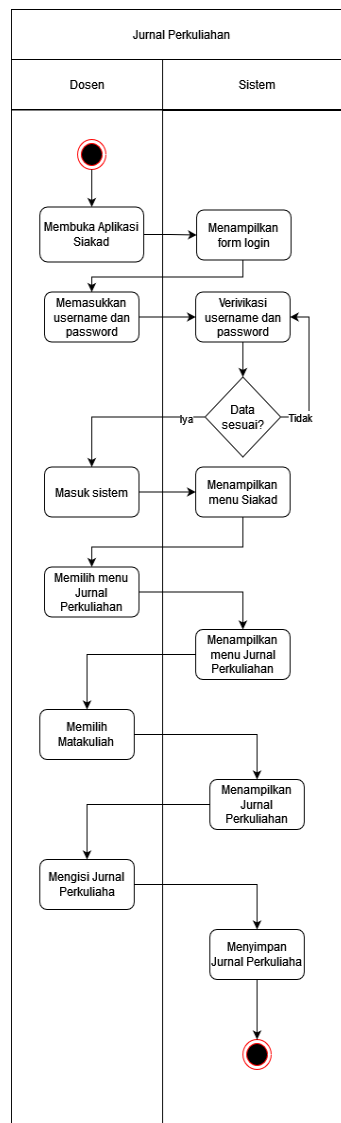
Pada iterasi pertama ini pengembang telah mengerjakan user stories 01 yang sudah dikerjakan pada bab sebelumnya. Story dikerjakan mulai dari risk dan value yang paling tinggi kemudian berlanjut sampai yang paling rendah. Dari semua story tersebut dibuatkan gambaran design dari tiap-tiap story yang nantinya ditunjukkan kepada client sehingga dapat melihat gambaran awal dari sistem yang dikerjakan nantinya. Kemudian dilanjutkan dengan implementasi dimana proses penulisan program atau code. Pada kasus ini setiap user story akan diuji nantinya pada tahap sistem testing dan pada tahap tersebut nantinya apakah sistem ini dapat dijalankan atau tidak.

4.1.1 Iteration Initialization

Pada iterasi 1 pengembangan integrasi akan mengerjakan user story 01.

4.1.2 Design

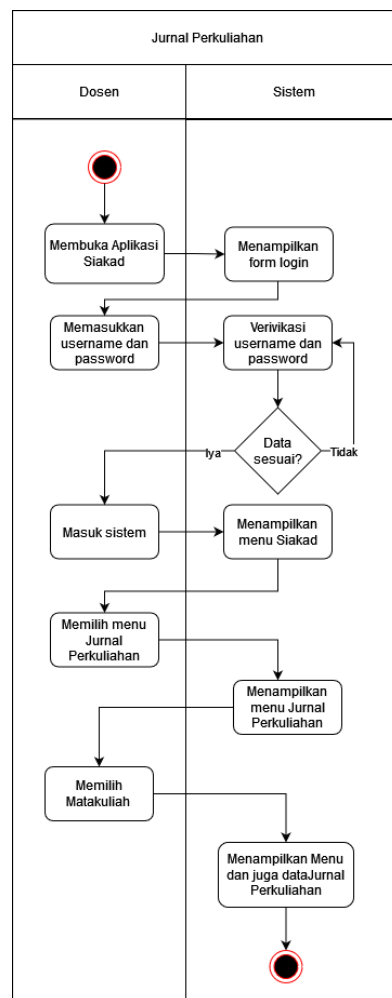
Design yang digunakan dalam iterasi ini adalah dengan menggunakan activity diagram. Pada activity diagram aplikasi yang belum terintegrasi akan tetap ditampilkan dan akan menjadi pembandingan dengan activity diagram setelah terintegrasi, agar terlihat perbedaannya setelah dan sebelum diintegrasikan.



Gambar 4.1 Activity Diagram Jurnal Perkuliahan sebelum Integrasi

Activity diagram dalam hal Jurnal Perkuliahan sebelum diintegrasikan, user dosen akan melakukan penginputan secara berulang kali. Setelah melakukan

kegiatan belajar, dosen akan langsung melakukan input pada SIAKAD pada pertemuan hari itu. Dosen juga bisa melakukan rekap dengan hari-hari sebelumnya kemudian data akan dimasukkan dalam satu hari.



Gambar 4.2 Activity Diagram Jurnal Perkuliahan sesudah Integrasi

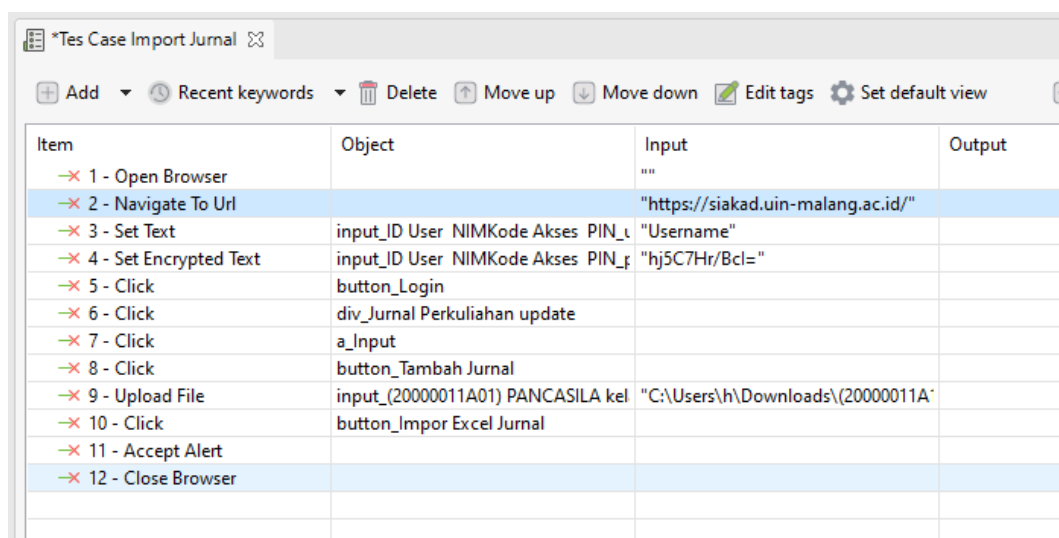
Pada gambar di atas, terlihat bahwa *activity diagram* sesudah jurnal perkuliahan diintegrasikan. Terlihat tahapan yang dilalui dosen lebih sedikit, itu karena dosen tidak perlu melakukan rekap dan penginputan data secara berulang. Sistem akan langsung merekap data dari *e-learning* dan kemudian data akan langsung dimasukkan dalam database SIAKAD.

4.1.3 Implementation

Tahap selanjutnya adalah tahap implementation dimana tahap ini dilakukan untuk mengembangkan desain menjadi sebuah kode yang dapat dijalankan. Tahap ini terbagi menjadi 3 tahapan yaitu unit testing, code generate, code refactoring.

4.1.2.1 Unit Testing

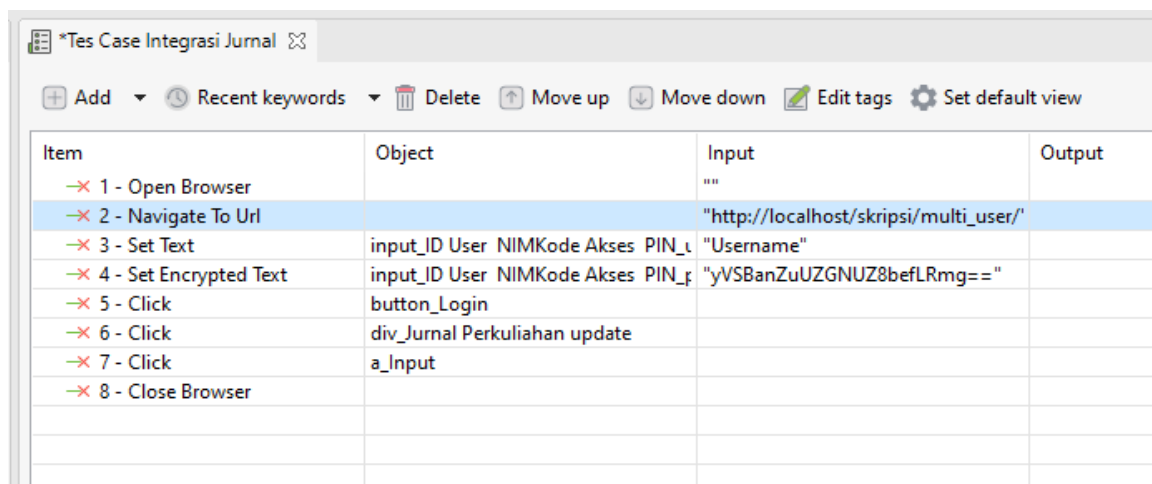
Unit testing yang dilakukan pada iterasi 1 menggunakan metode automated testing dengan katalon. Pada tahap pengujian test cases sistem jurnal perkuliahan, akan menguji dua sistem, yaitu sistem sebelum dan sesudah integrasi. Pada tahapan sistem sebelum diintegrasikan data akan direkap terlebih dahulu oleh dosen kemudian data diubah menjadi file excel, lalu dosen akan melakukan *import* data pada SIAKAD. Untuk hasil pengujian test cases dapat dilihat pada gambar di bawah.



Item	Object	Input	Output
→ 1 - Open Browser		""	
→ 2 - Navigate To Url		"https://siakad.uin-malang.ac.id/"	
→ 3 - Set Text	input_ID User NIMKode Akses PIN_t	"Username"	
→ 4 - Set Encrypted Text	input_ID User NIMKode Akses PIN_g	"hj5C7Hr/Bcl="	
→ 5 - Click	button_Login		
→ 6 - Click	div_Jurnal Perkuliahan update		
→ 7 - Click	a_Input		
→ 8 - Click	button_Tambah Jurnal		
→ 9 - Upload File	input_(20000011A01) PANCASILA kel	"C:\Users\h\Downloads\{20000011A"	
→ 10 - Click	button_Import Excel Jurnal		
→ 11 - Accept Alert			
→ 12 - Close Browser			

Gambar 4.3 Test Cases Sistem Jurnal Perkuliahan sebelum Integrasi

Pengujian test cases pada jurnal perkuliahan didapat 12 test step, dimulai dari membuka browser hingga menutup browser, tapi tahapan ini tidak termasuk perekapan data jurnal perkuliahan oleh dosen. Untuk sistem yang sudah diintegrasikan, dosen hanya tinggal membuka SIAKAD, data akan otomatis tertambahkan pada jurnal perkuliahan. Data yang diambil adalah data pertemuan course di *e-learning* yang kemudian diolah oleh sistem integrasi lalu ditambahkan kedalam data SIAKAD. Untuk hasil pengujian test cases dapat dilihat pada gambar di bawah.



Item	Object	Input	Output
1 - Open Browser		""	
2 - Navigate To Url		"http://localhost/skripsi/multi_user/"	
3 - Set Text	input_ID User NIMKode Akses PIN_u	"Username"	
4 - Set Encrypted Text	input_ID User NIMKode Akses PIN_f	"yVSBanZuUZGNUZ8befLRmg=="	
5 - Click	button_Login		
6 - Click	div_Jurnal Perkuliahan update		
7 - Click	a_Input		
8 - Close Browser			

Gambar 4.4 Test Cases Sistem Jurnal Perkuliahan setelah Integrasi

Pengujian test cases pada jurnal perkuliahan didapat 8 test step, dimulai dari membuka browser hingga menutup browser. Dosen tidak perlu melakukan rekap jurnal perkuliahan terlebih dahulu, karena data akan otomatis terintegrasikan dari pertemuan *e-learning*. Setelah melakukan test cases dan mendapatkan test step system akan di uji. Pada system pengintegrasian ini didapat system sudah sesuai dengan keperluan dan juga tidak ditemukan error. Setiap tes dilakukan

secara simultan sejak pengintegrasian dibuat seperti yang ditunjukkan pada gambar di bawah ini.

Test Suite ID	Test Suites/Ts Jurnal/TS Integrasi Jurnal	Local OS	Windows 10 64bit
Host name	h - DESKTOP-531EV0V	Platform	Firefox 106.0
Katalon version	8.5.2.208	End	2022-11-01 04:11:29
Start	2022-11-01 04:10:51		
Elapsed	37.160s		
Total TC	1		
Passed	1	Failed	0
Error	0	Skip	0
Incomplete	0		

Gambar 4.5 Test Result Jurnal Perkuliahan setelah Integrasi

4.1.2.2 Code

Setelah melakukan pengujian testing pembuatan sistem akan dilanjutkan pada fase code dimana nantinya code ditulis untuk membangun setiap fitur pada story itersai ini jadi satu. Pada pengintegrasian jurnal perkuliahan, data yang nantinya diambil adalah data course pada *e-learning*. Data Course pada *e-learning* nantinya akan diolah menjadi data yang dapat diintegrasikan pada SIAKAD. Untuk halaman pada SIAKAD sebelum data dari *e-learning* berhasil di Integrasikan dapat dilihat pada gambar di bawah.

Ke	Tanggal_Kuliah	Indikator	Materi_Kuliah	Keterangan	Edit	Hapus

Gambar 4.6 Tampilan Jurnal Perkuliahan sebelum Integrasi

Client akan memanggil data dari *web service server* dengan parameter *Mtk_Kode*, *cls*, *dosen*, *thn*, *bln*, *tgl*. Data dari *web service* akan dicek apakah data sudah ada atau belum pada data base SIAKAD. Jika data belum ada, *web service server* akan merespon dengan mengambil data *course* yang sudah diolah dari database *e-learning*, yang kemudian dikirim dalam format JSON tersajikan pada gambar 4.7 di bawah ini.

```
[
  {
    "MK": "BIOINFORMATIKA",
    "data": [
      {
        "Pertemuan": 0,
        "Nama Pertemuan": "Umum",
        "Tugas": "Pengumuman",
        "Jenis Tugas": "Kontrak Kuliah",
        "Tanggal": "2022-10-1"
      },
      {
        "Pertemuan": 1,
        "Nama Pertemuan": "Pertemuan 1",
        "Tugas": "Pertemuan Tugas Pertama MK B",
        "Jenis Tugas": "assign",
        "Tanggal": "2022-10-8"
      },
      {
        "Pertemuan": 2,
        "Nama Pertemuan": "Topik 2",
        "Tugas": "Tugas Pertemuan 2",
        "Jenis Tugas": "assign",
        "Tanggal": "2022-10-15"
      },
      {
        "Pertemuan": 3,
        "Nama Pertemuan": "Topik 3",
        "Tugas": "Ini Quiz",
        "Jenis Tugas": "quiz",
        "Tanggal": "2022-10-22"
      },
      {
        "Pertemuan": 4,
        "Nama Pertemuan": "Topik 4",
        "Tugas": "Tugas Pertemuan 3",
        "Jenis Tugas": "assign",
        "Tanggal": "2022-10-29"
      },
      {
        "Pertemuan": 5,
        "Nama Pertemuan": "Topik 5",
        "Tugas": "Tugas Pertemuan 4",
        "Jenis Tugas": "assign",
        "Tanggal": "2022-10-36"
      }
    ]
  }
]
```

Gambar 4.7 API Jurnal Perkuliahan

```

    },
    {
      "Pertemuan": 6,
      "Nama Pertemuan": "Topik 6",
      "Tugas": "Tugas Pertemuan 5",
      "Jenis Tugas": "assign",
      "Tanggal": "2022-10-43"
    },
    {
      "Pertemuan": 7,
      "Nama Pertemuan": "Topik 7",
      "Tugas": "Tugas Pertemuan 6",
      "Jenis Tugas": "assign",
      "Tanggal": "2022-10-50"
    },
    {
      "Pertemuan": 8,
      "Nama Pertemuan": "UTS",
      "Tugas": "Tugas Pertemuan 7",
      "Jenis Tugas": "UTS",
      "Tanggal": "2022-10-57"
    },
    {
      "Pertemuan": 9,
      "Nama Pertemuan": "Topik 9",
      "Tugas": "UTS",
      "Jenis Tugas": "assign",
      "Tanggal": "2022-10-64"
    },
    {
      "Pertemuan": 10,
      "Nama Pertemuan": "Topik 10",
      "Tugas": "Tugas Pertemuan 9",
      "Jenis Tugas": "assign",
      "Tanggal": "2022-10-71"
    },
    {
      "Pertemuan": 11,
      "Nama Pertemuan": "Topik 11",
      "Tugas": "Tugas Pertemuan 10",
      "Jenis Tugas": "assign",
      "Tanggal": "2022-10-78"
    },
    {
      "Pertemuan": 12,

      "Nama Pertemuan": "Topik 12",
      "Tugas": "Tugas Pertemuan 11",
      "Jenis Tugas": "assign",
      "Tanggal": "2022-10-85"
    },
    {
      "Pertemuan": 13,
      "Nama Pertemuan": "Topik 13",
      "Tugas": "Tugas Pertemuan 12",
      "Jenis Tugas": "assign",
      "Tanggal": "2022-10-92"
    },
    {
      "Pertemuan": 14,
      "Nama Pertemuan": "Topik 14",
      "Tugas": "Tugas Pertemuan 13",
      "Jenis Tugas": "assign",
      "Tanggal": "2022-10-99"
    },
    {
      "Pertemuan": 15,
      "Nama Pertemuan": "Topik 15",
      "Tugas": "Tugas Pertemuan 14",
      "Jenis Tugas": "assign",
      "Tanggal": "2022-10-106"
    },
    {
      "Pertemuan": 16,
      "Nama Pertemuan": "UAS",
      "Tugas": "Tugas Pertemuan 15",
      "Jenis Tugas": "UAS",
      "Tanggal": "2022-10-113"
    }
  ]
}

```

Gambar 4.8 API Jurnal Perkuliahan

Web service client akan menerima kemudian menyimpan data *course* format JSON tersebut ke dalam database SIAKAD. Untuk halaman pada SIAKAD sesudah data dari *e-learning* berhasil di Integrasikan dapat dilihat pada gambar di bawah.

Tahun 2022/2023 Semester Ganjil MataKuliah BIOINFORMATIKA [Tambah Jurnal](#)

Ke	Tanggal_Kuliah	Indikator	Materi_Kuliah	Keterangan	Edit	<input type="checkbox"/> Semua <input type="checkbox"/> Hapus
1	2022-10-01	Pengumuman	Umum	Kontrak Kuliah	Edit	<input type="checkbox"/> Hapus
2	2022-10-08	Pertemuan Tugas Pertama MK B	Pertemuan 1	assign	Edit	<input type="checkbox"/> Hapus
3	2022-10-15	Tugas Pertemuan 2, Inl Quiz	Topik 2	assign, quit	Edit	<input type="checkbox"/> Hapus
4	2022-10-22	Tugas Pertemuan 3	Topik 3	assign	Edit	<input type="checkbox"/> Hapus
5	2022-10-29	Tugas Pertemuan 4	Topik 4	assign	Edit	<input type="checkbox"/> Hapus
6	2022-11-05	Tugas Pertemuan 5	Topik 5	assign	Edit	<input type="checkbox"/> Hapus
7	2022-11-12	Tugas Pertemuan 6	Topik 6	assign	Edit	<input type="checkbox"/> Hapus
8	2022-11-19	Tugas Pertemuan 7	Topik 7	assign	Edit	<input type="checkbox"/> Hapus
9	2022-11-26	UTS	UTS	UTS	Edit	<input type="checkbox"/> Hapus
10	2022-12-03	Tugas Pertemuan 9	Topik 9	assign	Edit	<input type="checkbox"/> Hapus
11	2022-12-10	Tugas Pertemuan 10	Topik 10	assign	Edit	<input type="checkbox"/> Hapus
12	2022-12-17	Tugas Pertemuan 11	Topik 11	assign	Edit	<input type="checkbox"/> Hapus
13	2022-12-24	Tugas Pertemuan 12	Topik 12	assign	Edit	<input type="checkbox"/> Hapus
14	2022-12-31	Tugas Pertemuan 13	Topik 13	assign	Edit	<input type="checkbox"/> Hapus
15	2023-01-07	Tugas Pertemuan 14	Topik 14	assign	Edit	<input type="checkbox"/> Hapus
16	2023-01-14	Tugas Pertemuan 15	Topik 15	assign	Edit	<input type="checkbox"/> Hapus
17	2023-01-21	UAS	UAS	UAS	Edit	<input type="checkbox"/> Hapus

© 2022 code by PTPD UN Malang | MIT © Accusabul-med

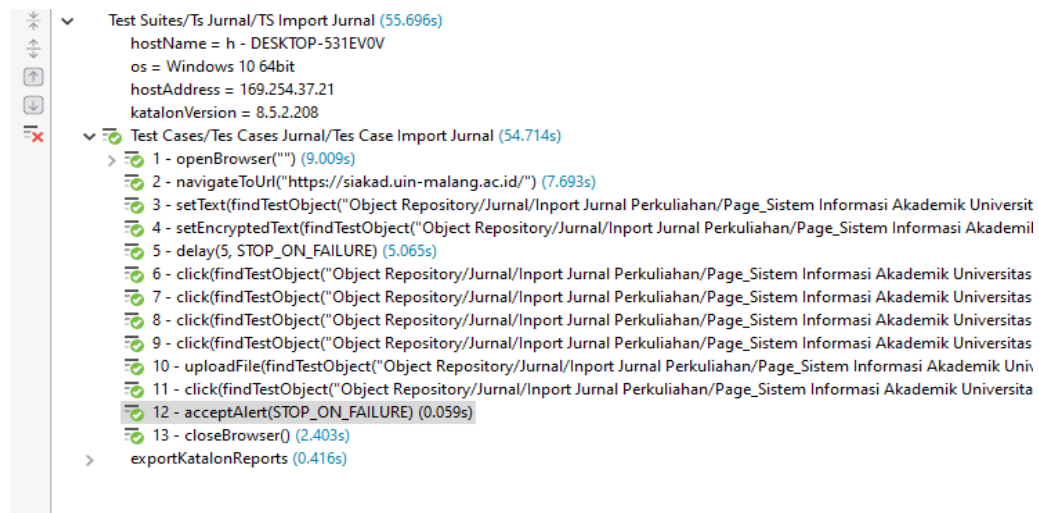
Gambar 4.9 Tampilan Jurnal Perkuliahan sesudah Integrasi

4.1.2.3 Refactor

Berdasarkan pada tahapan ini akan dikerjakan pengoptimalan code apabila dibutuhkan. Pada hasil implementasi dari setiap story tidak ditemukan code program yang harus dioptimalisasi lagi. Sehingga tidak dilakukan pengoptimalan atau refactor pada code program untuk iterasi ke-1.

4.1.4 System Testing

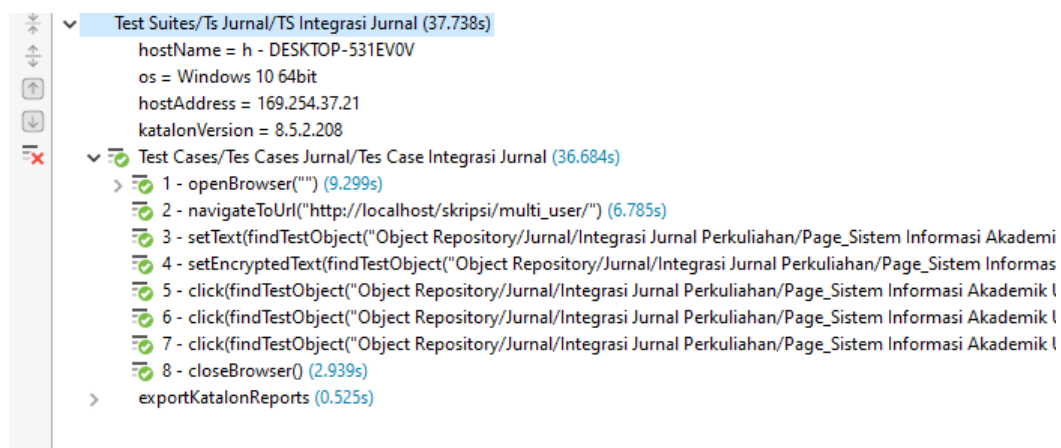
Setelah dilakukannya implementasi berupa coding, maka selanjutnya akan dilakukan pengujian unit testing. Setelah melakukan tahapan test cases jurnal perkuliahan, didapat 12 test step, namun ada tambahan test step pada jurnal perkuliahan ketika ingin melakukan import data, yaitu penambahan *delay* untuk konfirmasi *autentika* login. Tahapan selanjutnya adalah menguji setiap test step tersebut. Pengujian test step pada jurnal perkuliahan dapat dilihat pada gambar di bawah.



Gambar 4.10 Test Execution Jurnal Perkuliahan sebelum Integrasi

Pada pengujian test *execution* jurnal perkuliahan didapat waktu total 54,714 detik. Pengujian dilakukan dari membuka browser hingga menutup browser. Tahapan selanjutnya akan dilakukan test execution pada sistem yang sudah diintegrasikan. Pada dasarnya integration testing akan menguji masing-masing modul terlebih dahulu, diintegrasikan, lalu program akan diuji secara keseluruhan. Langkah pertama untuk melakukan pengujian pada penelitian ini adalah dengan melihat apakah Api dari e-learning sudah bisa dijalankan sesuai kebutuhan atau belum, lalu sistem akan diintegrasikan dengan SIAKAD. API Server pada iterasi ini sudah berhasil dijalankan sesuai kebutuhan, dapat dilihat pada gambar 4.7 dan gambar 4.8. Data yang akan diintegrasikan pada iterasi ini adalah data *course* pada e-learning dengan data jurnal perkuliahan pada SIAKAD, data berupa data tanggal kuliah yang akan disesuaikan dengan jadwal mengajar, indikator dengan hal apa saja yang dibahas pada hari itu di e-learning, materi kuliah dengan topik pembahasan pada e-learning . Setelah itu tahapan pengujian

akan langsung menguji system setelah diintegrasikan. Untuk hasil pengujian test *execution* dapat dilihat pada gambar di bawah.



Gambar 4.11 Test Execution Jurnal Perkuliahan sesudah Integrasi

Setelah sistem diintegrasikan, dilakukan test *execution* pada setiap step, kemudian didapat waktu pengujian 36,684 detik. Pada system pengintegrasian ini didapat system sudah sesuai dengan keperluan dan juga tidak ditemukan error. Setiap tes dilakukan secara simultan sejak pengintegrasian dibuat seperti yang ditunjukkan pada tabel di bawah ini.

Tabel 4.1 Hasil Pengujian Jurnal Perkuliahan

Kode User	Fungsi	Result
Stories		
Story-01	Mengintegrasikan data course pada <i>e-learning</i> dengan data jurnal perkuliahan di SIAKAD	<i>Passed</i>

4.1.5 Restrospective

Tabel 4.2 Restrospective Jurnal Perkuliahan

Iterasi 1			
User Story	Story Point	Estimasi	Pengerjaan
Story-01	2	2	2.5

Pada table diatas terlihat estimasi dan waktu yang digunakan tidak sama, hal ini dikarenakan adanya evaluasi test step dalam pengujian.

4.2 Iterasi Kedua

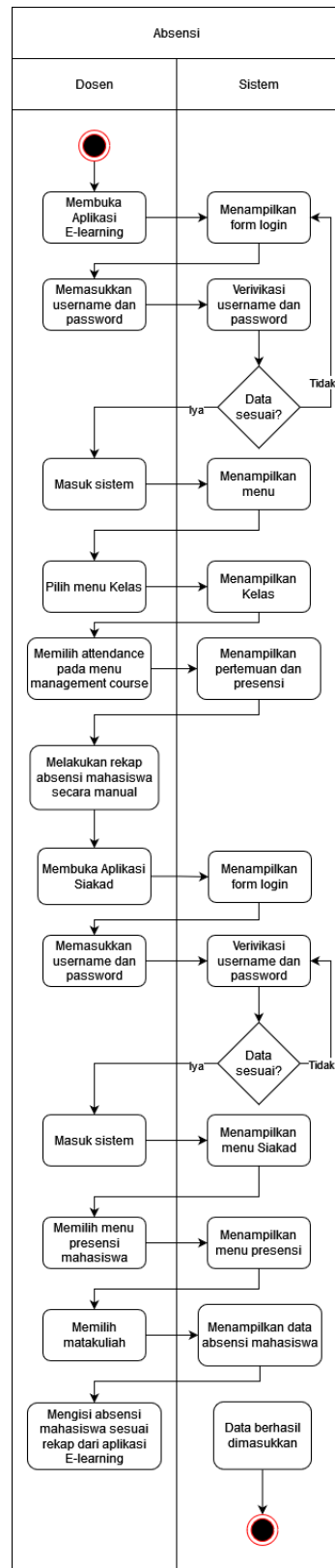
Pada iterasi kedua ini pengembang telah mengerjakan user stories 02 yang sudah dikerjakan pada bab sebelumnya. Story dikerjakan mulai dari risk dan value yang paling tinggi kemudian berlanjut sampai yang paling rendah. Dari semua story tersebut dibuatkan gambaran design dari tiap-tiap story. Kemudian dilanjutkan dengan implementasi dimana proses penulisan program atau code. Pada kasus ini setiap user story akan diuji nantinya pada tahap sistem testing dan pada tahap tersebut nantinya apakah sistem ini dapat dijalankan atau tidak.

4.2.1 Iteration Initialization

Pada iterasi 2 pengembangan integrasi akan mengerjakan user story 02.

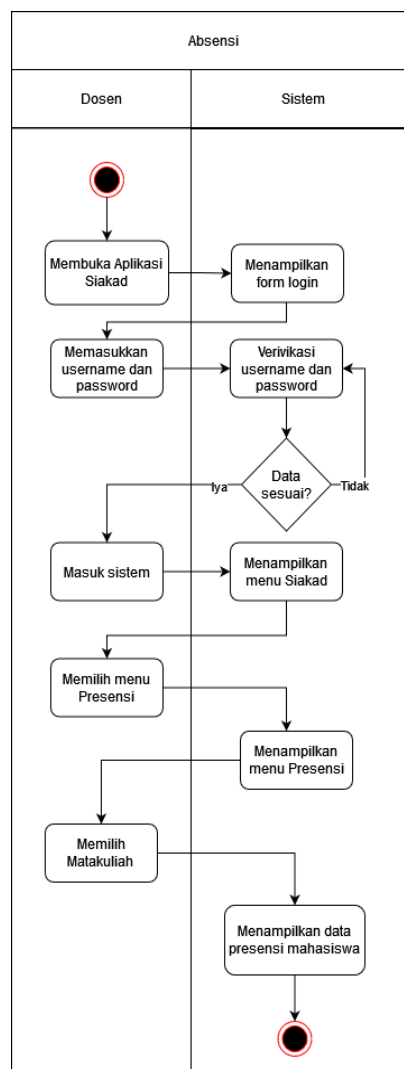
4.2.2 Design

Design yang digunakan dalam iterasi ini adalah dengan menggunakan activity diagram. Pada activity diagram aplikasi yang belum terintegrasi akan tetap ditampilkan dan akan menjadi pembanding dengan activity diagram setelah terintegrasi, agar terlihat perbedaannya setelah dan sebelum diintegrasikan.



Gambar 4.12 Activity Diagram Absensi sebelum Integrasi

Pada gambar 4.12 , dosen harus memasukkan data absensi secara persatu di SIAKAD. Dosen akan melakukan rekap absensi pada *e-learning* dan kemudian data akan dimasukkan ulang pada presensi mahasiswa di SIAKAD. Dosen harus melakukan rekap dan juga memasukkan data pada system SIAKAD agar data bisa tampil pada kedua system.



Gambar 4.13 Activity Diagram Absensi setelah Integrasi

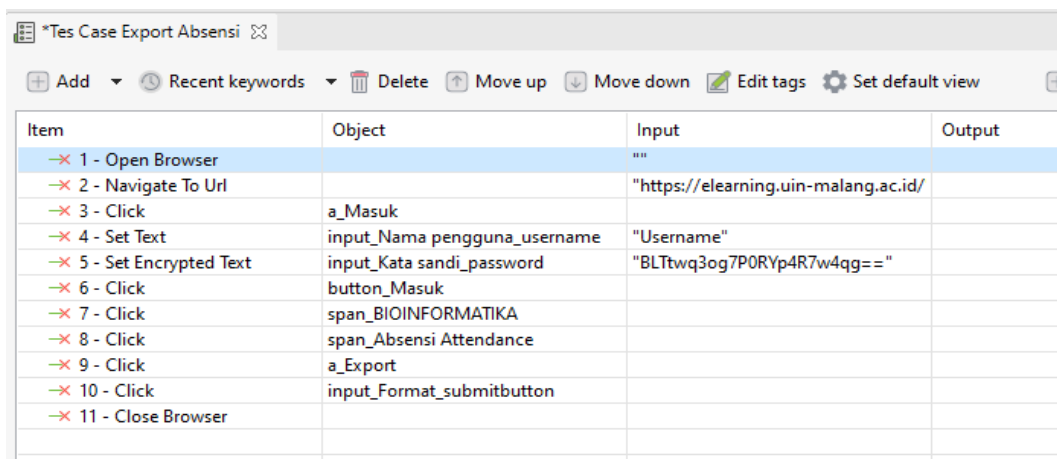
Pada gambar di atas, user dosen tidak perlu melakukan rekap absensi, karena data sudah terintegrasi antara *e-learning* dan juga SIAKAD. User Dosen hanya tinggal mengakses pada menu presensi untuk melihat absensi mahasiswa.

4.2.3 Implementation

Tahap selanjutnya adalah tahap implementation dimana tahap ini dilakukan untuk mengembangkan desain menjadi sebuah kode yang dapat dijalankan. Tahap ini terbagi menjadi 3 tahapan yaitu unit testing, code generate, code refactoring.

4.2.2.1 Unit Testing

Unit testing yang dilakukan pada iterasi 1 menggunakan metode automated testing dengan katalon. Pada tahap pengujian test cases sistem Absensi, akan menguji dua sistem, yaitu sistem sebelum dan sesudah integrasi. Pada tahapan sistem sebelum diintegrasikan data akan direkap terlebih dahulu oleh dosen kemudian data diubah menjadi file excel. Dalam perekapan data dosen harus melakukan *export* data melalui *e-learning* lalu dosen akan melakukan *import* data pada SIAKAD. Dalam test cases sebelum diintegrasikan akan dibuat dua tahapan, tahapan *export* data dari *e-learning* kemudian *import* data ke SIAKAD. Untuk hasil pengujian test cases dapat dilihat pada gambar di bawah.



Item	Object	Input	Output
1 - Open Browser		""	
2 - Navigate To Url		"https://elearning.uin-malang.ac.id/"	
3 - Click	a_Masuk		
4 - Set Text	input_Nama pengguna_username	"Username"	
5 - Set Encrypted Text	input_Kata sandi_password	"BLTtwq3og7P0RYp4R7w4qg=="	
6 - Click	button_Masuk		
7 - Click	span_BIOINFORMATIKA		
8 - Click	span_Absensi Attendance		
9 - Click	a_Export		
10 - Click	input_Format_submitbutton		
11 - Close Browser			

Gambar 4.14 Test Cases Sistem Absensi sebelum Integrasi Export

Item	Object	Input	Output
1 - Open Browser		""	
2 - Navigate To Url		"https://siakad.uin-malang.ac.id/"	
3 - Set Text	input_ID User NIMKode Akses PIN_u	"Username"	
4 - Set Encrypted Text	input_ID User NIMKode Akses PIN_f	"hj5C7Hr/Bcl="	
5 - Set Text	input_ID User NIMKode Akses PIN_c	"8"	
6 - Click	button_Login		
7 - Click	div_Presensi Mahasiswa update		
8 - Select Option By Value	select_pilih-PANCASILA (Kelas A Se	"797aaf5f99ffd6f40bfc9c150c6f4228";	
9 - Upload File	input_Unduh template presensi dahu	"C:\Users\h\Downloads\20000011A'	
10 - Click	button_IMPOR EXCEL		
11 - Accept Alert			
12 - Close Browser			

Gambar 4.15 Test Cases Sistem Absensi sebelum Integrasi Import

Pada tahapan *export* dan *import* data, didapat 11 test step pada *export* data dan juga 12 test step pada *import* data pada SIAKAD dimulai dari membuka browser hingga menutup browser. Untuk sistem yang sudah diintegrasikan, dosen hanya tinggal membuka SIAKAD, data akan otomatis tertambahkan pada absensi. Data yang diambil adalah data absensi pada pertemuan course di *e-learning* yang kemudian diolah oleh sistem integrasi lalu ditambahkan kedalam data SIAKAD. Untuk hasil pengujian test cases dapat dilihat pada gambar di bawah.

Item	Object	Input	Output
1 - Open Browser		""	
2 - Navigate To Url		"http://localhost/skripsi/multi_user/"	
3 - Set Text	input_ID User NIMKode Akses PIN_u	"Username"	
4 - Set Encrypted Text	input_ID User NIMKode Akses PIN_f	"yVSBanZuUZGNUZ8befLRmg="	
5 - Click	button_Login		
6 - Click	div_Presensi Mahasiswa update		
7 - Close Browser			

Gambar 4.16 Test Cases Sistem Absensi sesudah Integrasi

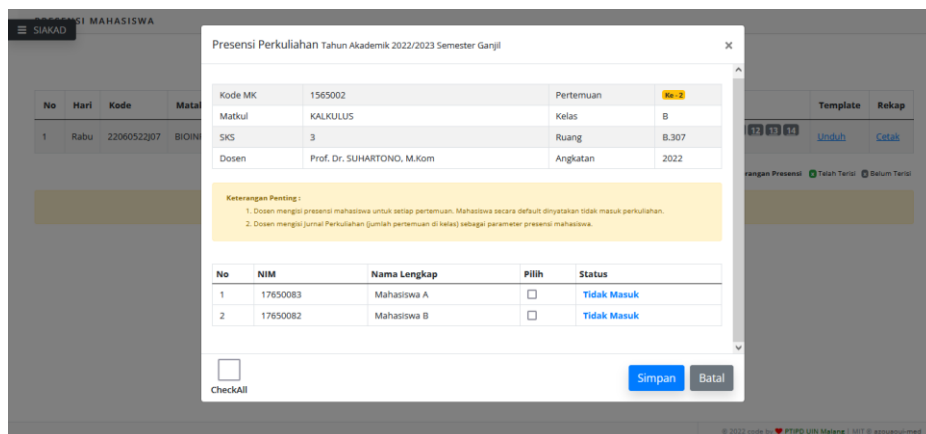
Sistem absensi yang terintegrasi, setelah dilakukan test cases terdapat 7 test step, dimulai dari membuka browser hingga menutup browser. Dosen tidak perlu melakukan rekap absensi terlebih dahulu, karena data akan otomatis terintegrasi dari pertemuan *e-learning*. Pada system pengintegrasian ini didapat system sudah sesuai dengan keperluan dan juga tidak ditemukan error. Setiap tes dilakukan secara simultan sejak pengintegrasian dibuat seperti yang ditunjukkan pada gambar di bawah ini.

Test Suite ID	Test Suites/Ts Absen/TS Integrasi Absensi		
Host name	h - DESKTOP-531EVOV	Local OS	Windows 10 64bit
Katalon version	8.5.2.208	Platform	Chrome 106.0.0.0
Start	2022-11-01 03:50:36	End	2022-11-01 03:51:10
Elapsed	33.423s		
Total TC	1		
Passed	1	Failed	0
Error	0	Skip	0
Incomplete	0		

Gambar 4.17 Test Result Absensi setelah Integrasi

4.2.2.2 Code

Code generate dilakukan dengan cara membuat sebuah source code yang dapat melakukan integrasi sesuai dengan user story 02. Pengintegrasian Absensi SIAKAD dengan *e-learning* nantinya akan mengambil data absensi pada *e-learning*. Data absensi pada *e-learning* nantinya akan diolah pada *server client* menjadi data yang dapat disimpan pada data base SIAKAD. Untuk halaman pada SIAKAD sebelum data dari *e-learning* berhasil di Integrasikan dapat dilihat pada gambar di bawah.



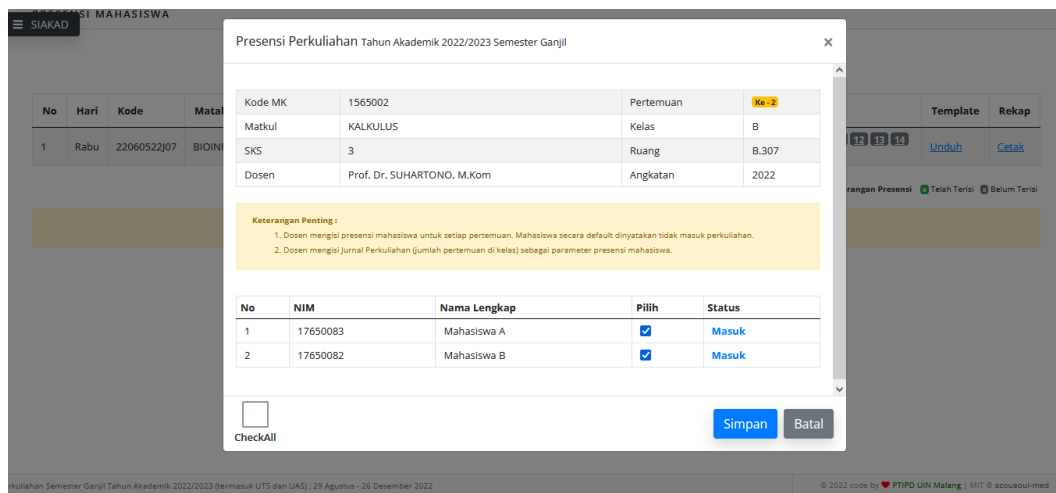
Gambar 4.18 Tampilan Absensi sebelum Integrasi

Pada *web service client* SIAKAD akan memanggil data *web service server* dengan parameter *Mtk_Kode*, *cls*, *dosen*. Data dari *web service*, akan dilakukan pengecekan, apakah data sudah ada atau belum pada data base SIAKAD. Jika data belum ada, *web service server* akan merespon dengan mengambil data absensi yang sudah diolah dari database *e-learning*, yang kemudian dikirim dalam format JSON tersajikan pada gambar 4.5 di bawah ini.

```
[
  {
    "17650083": [
      "1",
      "1",
      "0",
      "0",
      "0",
      "0",
      "0",
      "0",
      "0",
      "0",
      "0",
      "0",
      "0",
      "0",
      "0",
      "0",
      "0",
      "0"
    ]
  },
  {
    "17650082": [
      "0",
      "1",
      "0",
      "0",
      "0",
      "0",
      "0",
      "0",
      "0",
      "0",
      "0",
      "0",
      "0",
      "0",
      "0",
      "0",
      "0",
      "0"
    ]
  }
]
```

Gambar 4.19 API Absensi

Web service client akan menerima kemudian menyimpan data absensi format JSON tersebut ke dalam database SIAKAD. Untuk halaman pada SIAKAD sesudah data dari *e-learning* berhasil di Integrasikan dapat dilihat pada gambar di bawah.



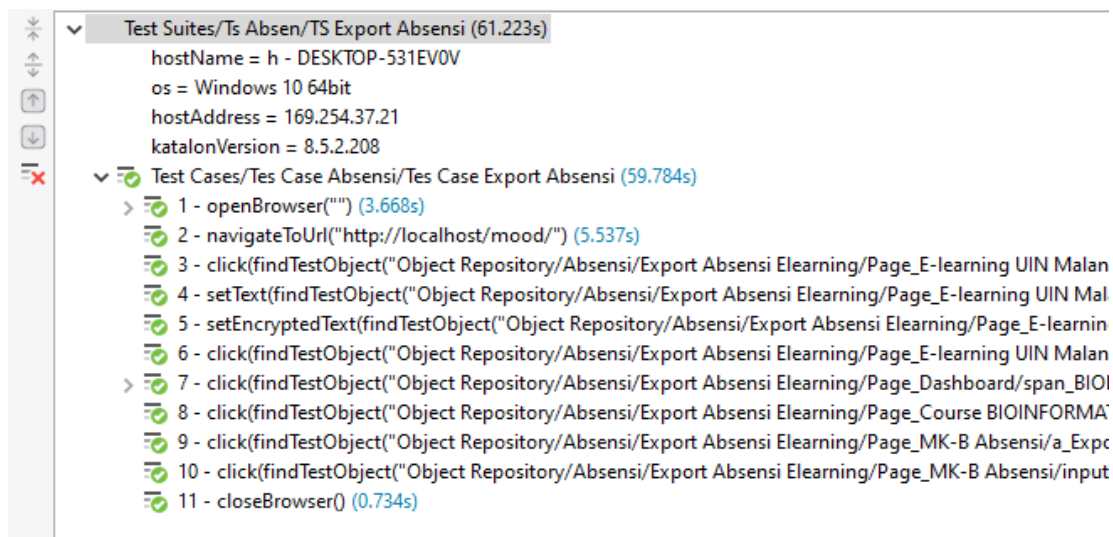
Gambar 4.20 Tampilan Absensi sesudah Integrasi

4.2.2.3 Refactor

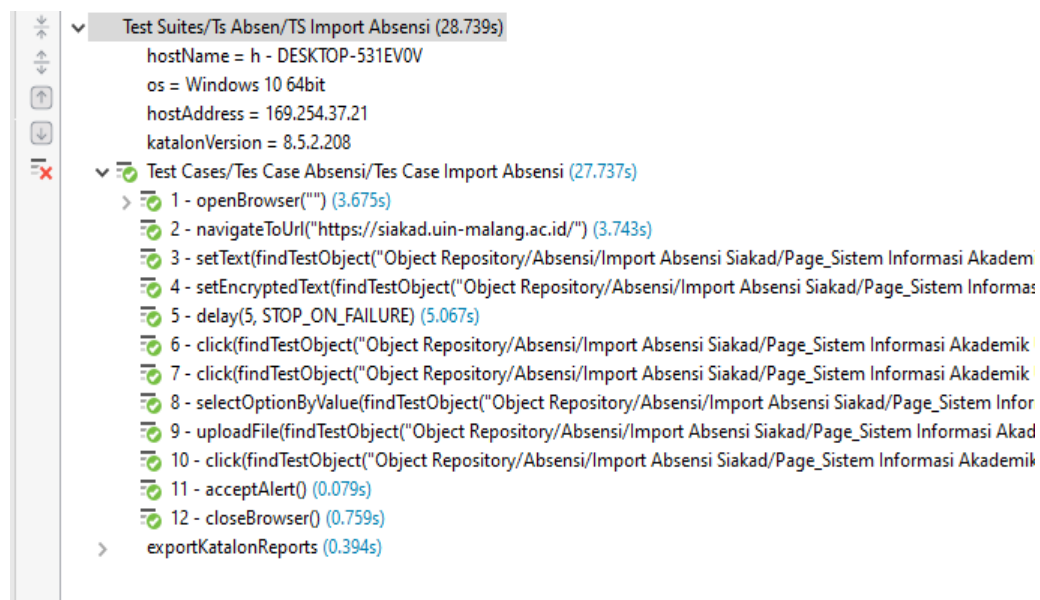
Berdasarkan pada tahapan ini akan dikerjakan pengoptimalan code apabila dibutuhkan. Pada hasil implementasi dari setiap story tidak ditemukan code program yang harus dioptimalisasi lagi. Sehingga tidak dilakukan pengoptimalan atau refactor pada code program untuk iterasi ke-2.

4.2.4 System Testing

Setelah dilakukannya implementasi berupa coding, maka selanjutnya akan dilakukan pengujian unit testing. Pada sistem sebelumnya di absensi SIAKAD, terdapat dua test cases yaitu export dan import. Pada tahapan export didapat 11 test step, sedangkan untuk import terdapat 12 test step. Tahapan selanjutnya adalah menguji setiap test step tersebut. Pengujian test step pada jurnal perkuliahan dapat dilihat pada gambar di bawah.



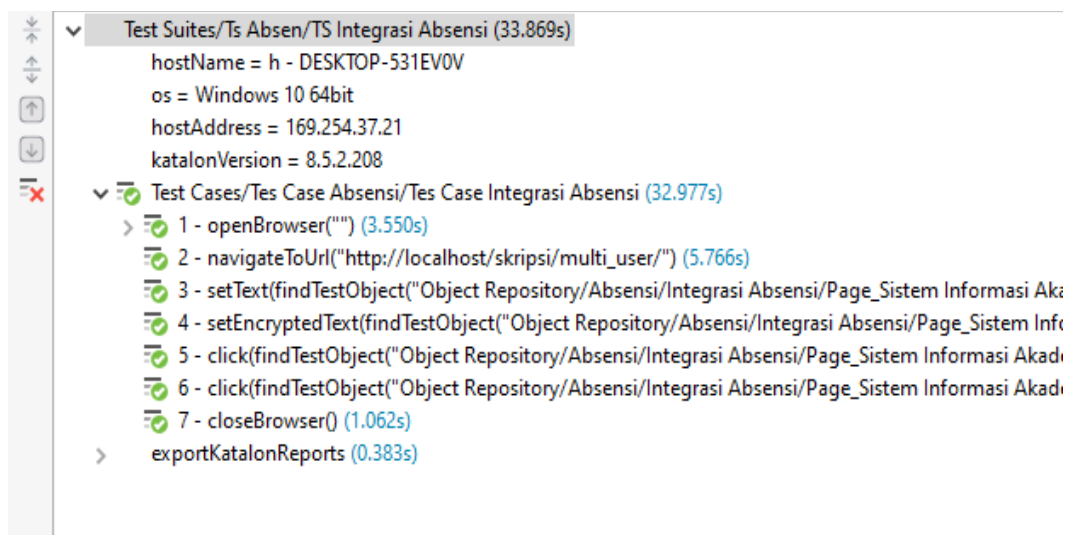
Gambar 4.21 Test Execution Absensi sebelum Integrasi Export



Gambar 4.22 Test Execution Absensi sebelum Integrasi Import

Pada pengujian test *execution* absensi perkuliahan didapat waktu total untuk *export* 59,784 detik dan untuk *import* 27,737 detik. Pengujian dilakukan dari membuka browser hingga menutup browser. Tahapan selanjutnya akan dilakukan test execution pada sistem yang sudah diintegrasikan. Pada dasarnya integration testing akan menguji masing-masing modul terlebih dahulu,

diintegrasikan, lalu program akan diuji secara keseluruhan. Langkah pertama untuk melakukan pengujian pada penelitian ini adalah dengan melihat apakah Api dari e-learning sudah bisa dijalankan sesuai kebutuhan atau belum, lalu sistem akan diintegrasikan dengan SIAKAD. API Server pada iterasi ini sudah berhasil dijalankan sesuai kebutuhan, dapat dilihat pada gambar 4.19. Data yang akan diintegrasikan pada iterasi ini adalah data absensi pada e-learning dengan data absensi pada SIAKAD, seperti nim dan juga absensi. Setelah itu tahapan pengujian akan langsung menguji system setelah diintegrasikan. Untuk hasil pengujian test execution dapat dilihat pada gambar di bawah.



Gambar 4.23 Test Execution Absensi sesudah Integrasi

Setelah sistem diintegrasikan, dilakukan test *execution* pada setiap step, kemudian didapat waktu pengujian 32,977 detik. Dalam menggunakan PXP model, proses tes dilakukan dengan menggunakan metode autometed testing dengan katalon. Pada system pengintegrasian ini didapat system sudah sesuai dengan keperluan dan juga tidak ditemukan error. Setiap tes dilakukan secara simultan sejak pengintegrasian dibuat seperti yang ditunjukkan pada tabel di bawah ini

Tabel 4.3 Hasil Pengujian Absensi

Kode User Stories	Fungsi	Result
Story-02	Mengintegrasikan data Absensi pada <i>e-learning</i> dengan data Absensi di SIAKAD	<i>Passed</i>

4.2.5 Restrospective

Tabel 4.4 Restrospective Absensi

Iterasi 2			
User Story	Story Point	Estimasi	Pengerjaan
Story-02	4	4	4

4.3 Iterasi Ketiga

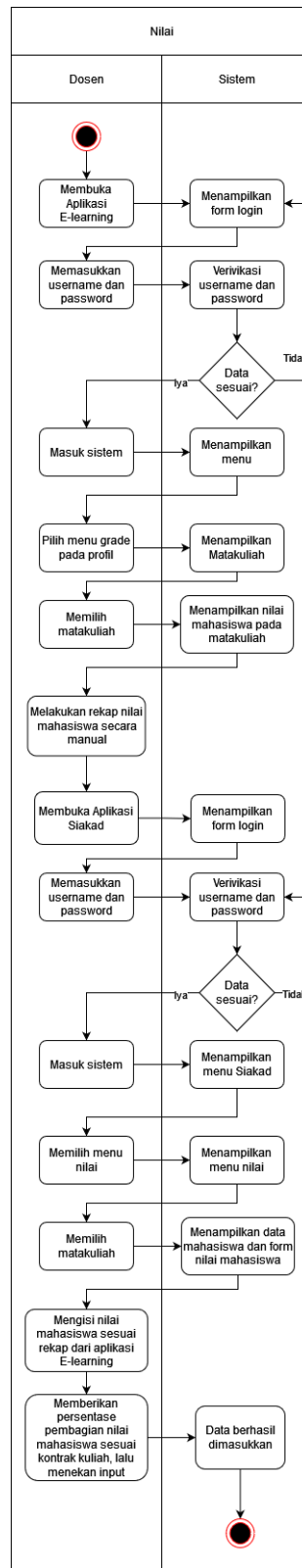
Pada iterasi ketiga ini pengembang telah mengerjakan user stories 03 yang sudah dikerjakan pada bab sebelumnya. Story dikerjakan mulai dari risk dan value yang paling tinggi kemudian berlanjut sampai yang paling rendah. Dari semua story tersebut dibuatkan gambaran design dari tiap-tiap story yang nantinya ditunjukkan kepada client sehingga dapat melihat gambaran awal dari sistem yang dikerjakan nantinya. Kemudian dilanjutkan dengan implementasi dimana proses penulisan program atau code. Pada kasus ini setiap user story akan diuji nantinya pada tahap sistem testing dan pada tahap tersebut nantinya apakah sistem ini dapat dijalankan atau tidak.

4.3.1 Iteration Initialization

Pada iterasi 3 pengembangan integrasi akan mengerjakan user story 03.

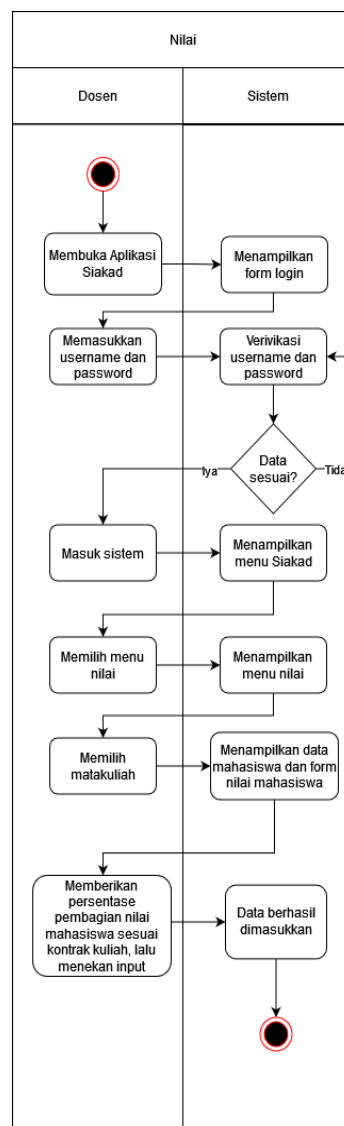
4.3.2 Design

Design yang digunakan dalam iterasi ini adalah dengan menggunakan activity diagram. Pada activity diagram aplikasi yang belum terintegrasi akan tetap ditampilkan dan akan menjadi pembanding dengan activity diagram setelah terintegrasi, agar terlihat perbedaannya setelah dan sebelum diintegrasikan.



Gambar 4.24 Activity Diagram Nilai sebelum Integrasi

Pada gambar di atas, dosen harus memberikan nilai secara persatu di *e-learning* terlebih dahulu dan kemudian dosen melakukan rekap secara manual pada setiap mahasiswa. Dosen harus membuka menu *grade* pada *e-learning* kemudian membuka menu *grade*, untuk melakukan rekap pada setiap mahasiswa. Setelah melakukan rekap, dosen akan membuka *e-learning* dan memasukkan nilai mahasiswa dengan satu persatu lalu memberikan persentase sesuai kontrak kuliah.



Gambar 4.25 Activity Diagram Nilai sesudah Integrasi

Gambar 4.25 adalah Activity Diagram User Dosen Ketika ingin memberikan nilai matakuliah pada mahasiswa. Dosen hanya tinggal melakukan input nilai sekali ketika dosen melakukan penilaian tugas mahasiswa. Ketika ingin memberikan nilai pada SIAKAD, dosen hanya tinggal memberikan persentase sesuai kesepakatan kontrak kuliah tanpa harus merekap satu persatu nilai mahasiswa.

4.3.3 Implementation

Tahap selanjutnya adalah tahap implementation dimana tahap ini dilakukan untuk mengembangkan desain menjadi sebuah kode yang dapat dijalankan. Tahap ini terbagi menjadi 3 tahapan yaitu unit testing, code generate, code refactoring.

4.3.2.1 Unit Testing

Unit testing yang dilakukan pada iterasi 1 menggunakan metode automated testing dengan katalon. Pengujian test cases sistem Nilai, akan menguji dua sistem, yaitu sistem sebelum dan sesudah integrasi. Pada tahapan sistem sebelum diintegrasikan data akan direkap terlebih dahulu oleh dosen kemudian data diubah menjadi file excel. Dalam perekapan data dosen harus melakukan *export* data melalui *e-learning* lalu dosen akan melakukan *import* data pada SIAKAD. Dalam test cases sebelum diintegrasikan akan dibuat dua tahapan, tahapan *export* data dari *e-learning* kemudian *import* data ke SIAKAD. Untuk hasil pengujian test cases dapat dilihat pada gambar di bawah.

Item	Object	Input	Output
→ 1 - Open Browser		""	
→ 2 - Navigate To Url		"https://elearning.uin-malang.ac.id/"	
→ 3 - Click	a_Masuk		
→ 4 - Set Text	input_Nama pengguna_username	"Username"	
→ 5 - Set Encrypted Text	input_Kata sandi_password	"BLTtwq3og7P0RYp4R7w4qg=="	
→ 6 - Click	button_Masuk		
→ 7 - Click	span_BIOINFORMATIKA		
→ 8 - Click	a_BIOINFORMATIKA_action-menu-tc		
→ 9 - Click	a_Gradebook setup		
→ 10 - Click	a_Export		
→ 11 - Click	input_Grade export decimal places_s		
→ 12 - Close Browser			

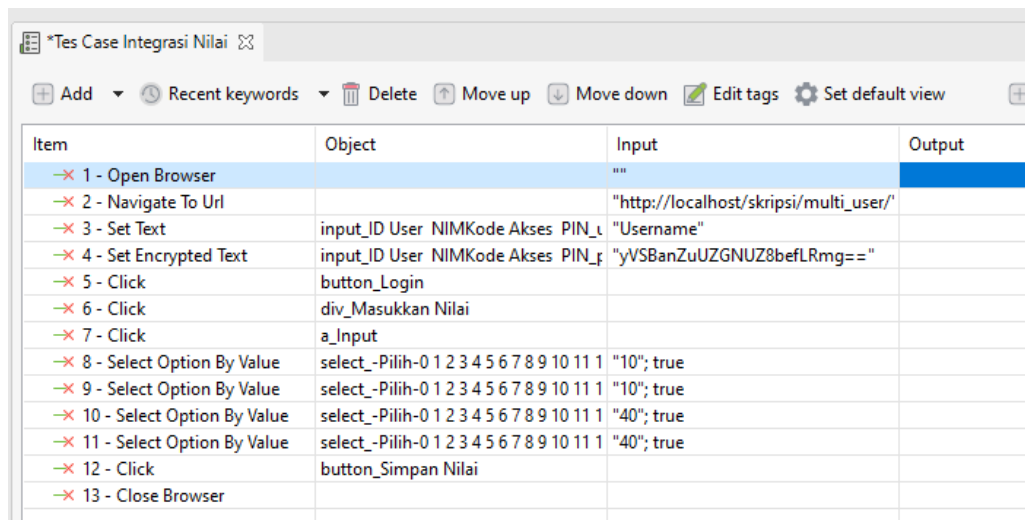
Gambar 4.26 Test Cases Sistem Nilai sebelum Integrasi Import

Item	Object	Input	Output
→ 1 - Open Browser		""	
→ 2 - Navigate To Url		"https://siakad.uin-malang.ac.id/"	
→ 3 - Set Text	input_ID User NIMKode Akses PIN_L	"Username"	
→ 4 - Set Encrypted Text	input_ID User NIMKode Akses PIN_L	"yVSBanZuUZGnuZ8befLRmg=="	
→ 5 - Click	button_Login		
→ 6 - Click	div_Masukkan Nilai		
→ 7 - Upload File	input_DOWNLOAD_excel_nilai	"C:\Users\h\Downloads\import_nilai	
→ 8 - Click	button_IMPORT EXCEL		
→ 9 - Accept Alert			
→ 10 - Close Browser			

Gambar 4.27 Test Cases Sistem Nilai sebelum Integrasi Export

Pada tahapan *export* dan *import* data, didapat 12 test step pada *export* data dan juga 10 test step pada *import* data pada SIAKAD dimulai dari membuka browser hingga menutup browser, total tes step yang dibutuhkan user ketika ingin melakukan penginputan nilai adalah 22 step. Untuk sistem yang sudah diintegrasikan, dosen hanya tinggal membuka SIAKAD, data akan otomatis tertambahkan pada data nilai. Data yang diambil adalah data *grade* pada *course* di *e-learning* yang kemudian diolah oleh sistem integrasi lalu ditambahkan kedalam

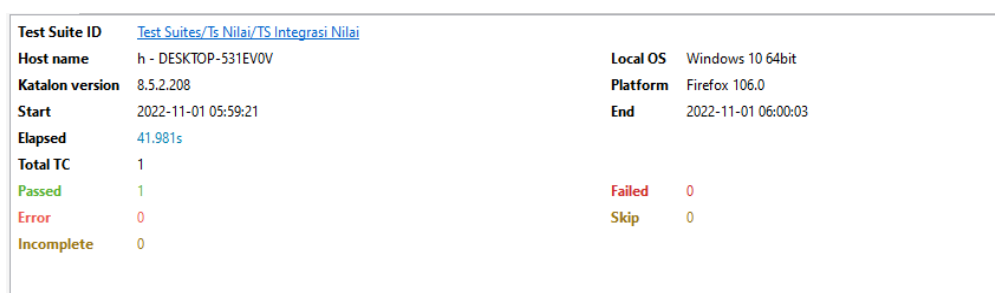
data SIAKAD. Untuk hasil pengujian test cases dapat dilihat pada gambar di bawah.



Item	Object	Input	Output
1 - Open Browser		""	
2 - Navigate To Url		"http://localhost/skripsi/multi_user/"	
3 - Set Text	input_ID User NIMKode Akses PIN_u	"Username"	
4 - Set Encrypted Text	input_ID User NIMKode Akses PIN_u	"yVSBanZuUZGNUZ8befLRmg=="	
5 - Click	button_Login		
6 - Click	div_Masukkan Nilai		
7 - Click	a_Input		
8 - Select Option By Value	select_-Pilih-0 1 2 3 4 5 6 7 8 9 10 11 1	"10"; true	
9 - Select Option By Value	select_-Pilih-0 1 2 3 4 5 6 7 8 9 10 11 1	"10"; true	
10 - Select Option By Value	select_-Pilih-0 1 2 3 4 5 6 7 8 9 10 11 1	"40"; true	
11 - Select Option By Value	select_-Pilih-0 1 2 3 4 5 6 7 8 9 10 11 1	"40"; true	
12 - Click	button_Simpan Nilai		
13 - Close Browser			

Gambar 4.28 Test Cases Sistem Nilai sesudah Integrasi

Sistem nilai yang terintegrasi, setelah dilakukan test cases terdapat 13 test step, dimulai dari membuka browser hingga menutup browser. Dosen tidak perlu melakukan rekap nilai terlebih dahulu, karena data akan otomatis terintegrasi dari pertemuan *e-learning*, tetapi dosen harus masih menginputkan data persentasi nilai sesuai dengan kontrak kuliah. Pada system pengintegrasian ini didapat system sudah sesuai dengan keperluan dan juga tidak ditemukan error. Setiap tes dilakukan secara simultan sejak pengintegrasian dibuat seperti yang ditunjukkan pada gambar di bawah ini.



Test Suite ID	Test Suites/Ts Nilai/TS Integrasi Nilai		
Host name	h - DESKTOP-531EV0V	Local OS	Windows 10 64bit
Katalon version	8.5.2.208	Platform	Firefox 106.0
Start	2022-11-01 05:59:21	End	2022-11-01 06:00:03
Elapsed	41.981s		
Total TC	1		
Passed	1	Failed	0
Error	0	Skip	0
Incomplete	0		

Gambar 4.29 Test Result Nilai setelah Integrasi

4.3.2.2 Code

Code generate dilakukan dengan cara membuat sebuah source code yang dapat melakukan integrasi sesuai dengan user story 03. Pada pengintegrasian Nilai, data yang nantinya diambil adalah data nilai pada *e-learning*. Data nilai pada *e-learning* nantinya akan diolah menjadi data yang dapat diintegrasikan pada SIAKAD. Untuk halaman pada SIAKAD sebelum data dari *e-learning* berhasil di Integrasikan dapat dilihat pada gambar di bawah.

Kode MK	22060522J07	Semester	Ganjil, 2022/2023
Matakuliah	BIOINFORMATIKA	Kelas / Ruang	A / MIT1
SKS	3	Status	Dosen Tunggal

1. Nilai yang dimasukkan berupa nilai angka dan bukan nilai huruf. Jika ada koma maka memakai tanda (,) bukan (.)
 2. Range penilaian adalah sebagai berikut : [KLIK DETAIL](#)

No	NIM	Nama Mahasiswa	_Q.U.I.Z_	Tugas_ / Tutorial_	Praktikum	UT I / Seminar_ Ilmiah	UT II / Ujian_ Blok	Nilai Akhir	
			-Pilih-	-Pilih-	-Pilih-	-Pilih-	-Pilih-	Angka	Huruf
Total persentase harus 100% untuk skoring akhir									
1	17650082	Mahasiswa B	0	0	0	0	0		
2	17650083	Mahasiswa A	0	0	0	0	0		

[Kembali](#) [Simpan Nilai](#) [Cetak Nilai](#)

© 2022 code by [PTIPD UIN Malang](#) | MIT @ azouabul-med

Gambar 4.30 Tampilan Nilai sebelum Integrasi

Data dari web service akan dilakukan pengecekan apakah data sudah ada atau belum pada data base SIAKAD. Jika data belum ada, *web service server* akan merespon dengan mengambil data absensi yang sudah diolah dari database *e-learning*, yang kemudian dikirim dalam format JSON tersajikan pada gambar 4.8 di bawah ini.

```
[
  {
    "Kelas": "214371",
    "MK": "BIOINFORMATIKA",
    "Dosen": "66245",
    "data": [
      {
        "Nim": "17650083",
        "Nama": "Mahasiswa A",
        "nilaiTugas": "89.857142857143",
        "nilaiQuiz": "66.66667",
        "nilaiPraktikum": "0",
        "nilaiUT1": "100",
        "nilaiUT2": "89"
      },
      {
        "Nim": "17650082",
        "Nama": "Mahasiswa B",
        "nilaiTugas": "5.6428571428571",
        "nilaiQuiz": "0",
        "nilaiPraktikum": "0",
        "nilaiUT1": "0",
        "nilaiUT2": "0"
      }
    ]
  }
]
```

Gambar 4.31 API Nilai

Web service client akan menerima kemudian menyimpan data nilai format JSON tersebut ke dalam database SIAKAD. Untuk halaman pada SIAKAD sesudah data dari *e-learning* berhasil di Integrasikan dapat dilihat pada gambar di bawah.

Kode MK	22060522j07	Semester	Ganjil, 2022/2023						
Matakuliah	BIOINFORMATIKA	Kelas / Ruang	A / MIT1						
SKS	3	Status	Dosen Tunggal						
1. Nilai yang dimasukkan berupa nilai angka dan bukan nilai huruf. Jika ada koma maka memakai tanda (,) bukan (.) 2. Range penilaian adalah sebagai berikut: KLIK DETAIL									
No	NIM	Nama Mahasiswa	_Q.U.I.Z_	Tugas / Tutorial	Praktikum	UT I / Seminar Ilmiah	UT II / Ujian Blok	Nilai Akhir	
Total persentase harus 100% untuk skoring akhir									
			-Pilih-	-Pilih-	-Pilih-	-Pilih-	-Pilih-	Angka	Huruf
1	17650083	Mahasiswa A	66.66667	89.857142857143	0	100	89		
2	17650082	Mahasiswa B	0	5.6428571428571	0	0	0		

Kembali

✓ Simpan Nilai
🖨️ Cetak Nilai

© 2022 code by ♥ PTIPD UIN Malang | MIT @ azousouli.med

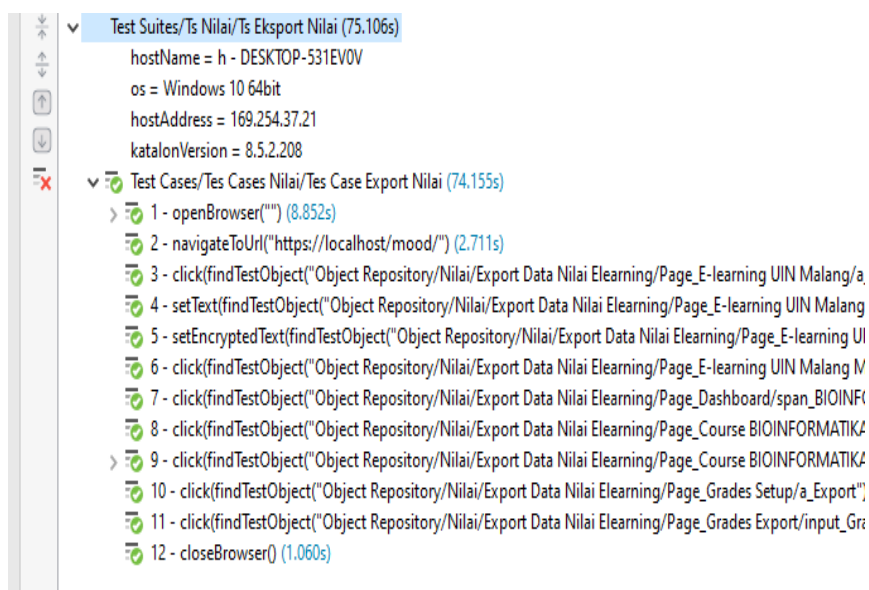
Gambar 4.32 Tampilan Nilai sesudah Integrasi

4.3.2.3 Refactor

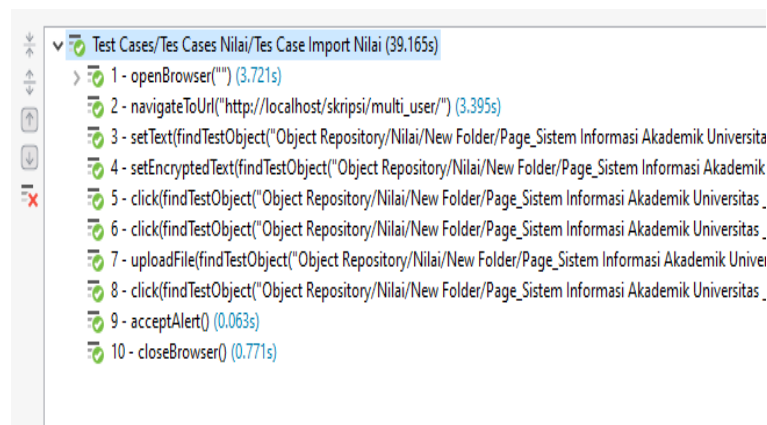
Berdasarkan pada tahapan ini akan dikerjakan pengoptimalan code apabila dibutuhkan. Pada hasil implementasi dari setiap story tidak ditemukan code program yang harus dioptimalisasi lagi. Sehingga tidak dilakukan pengoptimalan atau refactor pada code program untuk iterasi ke-3.

4.3.4 System Testing

Setelah dilakukannya implementasi berupa coding, maka selanjutnya akan dilakukan pengujian unit testing. Sistem sebelumnya pada absensi SIAKAD, terdapat dua test cases yaitu export dan import. Pada tahapan export didapat 12 test step, sedangkan untuk import terdapat 13 test step. Tahapan selanjutnya adalah menguji setiap test step tersebut. Pengujian test step pada jurnal perkuliahan dapat dilihat pada gambar di bawah.

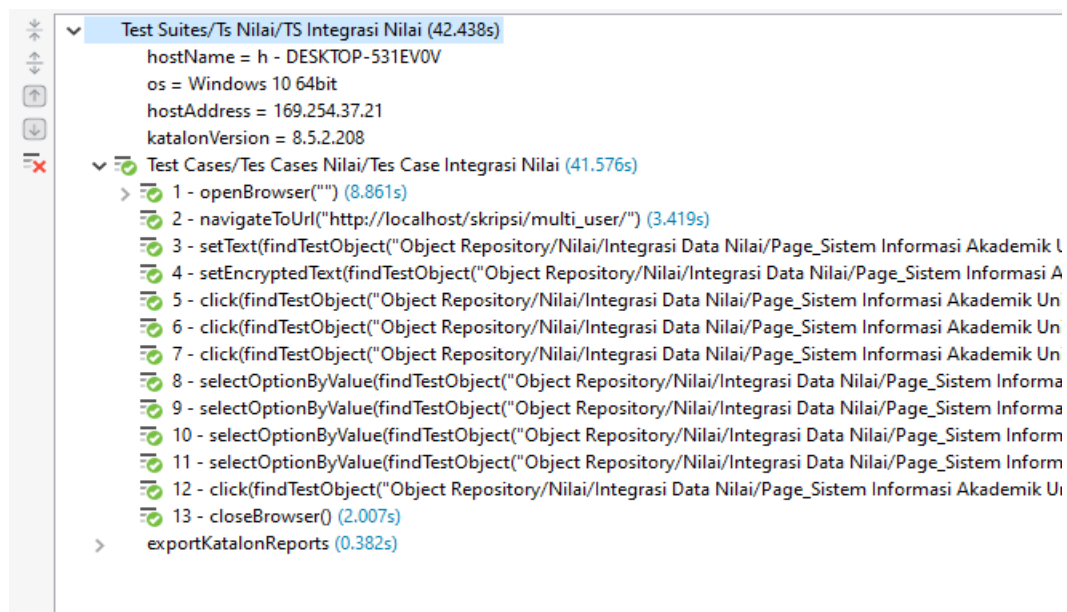


Gambar 4.33 Test Execution Nilai sebelum Integrasi Import



Gambar 4.34 Test Execution Nilai sebelum Integrasi Export

Pada pengujian test *execution* nilai perkuliahan didapat waktu total untuk *export* 71,155 detik dan untuk *import* 39,165 detik. Pengujian dilakukan dari membuka browser hingga menutup browser. Tahapan selanjutnya akan dilakukan test execution pada sistem yang sudah diintegrasikan. Pada dasarnya integration testing akan menguji masing-masing modul terlebih dahulu, diintegrasikan, lalu program akan diuji secara keseluruhan. Langkah pertama untuk melakukan pengujian pada penelitian ini adalah dengan melihat apakah Api dari e-learning sudah bisa dijalankan sesuai kebutuhan atau belum, lalu sistem akan diintegrasikan dengan SIAKAD. API Server pada iterasi ini sudah berhasil dijalankan sesuai kebutuhan, dapat dilihat pada gambar 4.31. Data yang akan diintegrasikan pada iterasi ini adalah data grade pada e-learning dengan data nilai pada SIAKAD, data berupa data nilai tugas, nilai quis, nilai uts, nilai uas yang sudah diolah dari e-learning. Setelah itu tahapan pengujian akan langsung menguji system setelah diintegrasikan. Untuk hasil pengujian test execution dapat dilihat pada gambar di bawah.



Gambar 4.35 Test Execution Nilai sesudah Integrasi

Setelah sistem diintegrasikan, dilakukan test *execution* pada setiap step, kemudian didapat waktu pengujian 41, 576 detik. Waktu terlama selain membuka browser dan navigate url adalah terjadi di *object button input*, sekitar 4,521 detik, hal itu bisa terjadi karena adanya proses pengecekan pada system apakah data dalam SIAKAd kosong atau tidak. Jika data kosong, sistem akan otomatis mengecek pada database *e-learning* dan jika data elearning tidak kosong, maka system akan langsung mengintegrasikan nilai ke database SIAKAD. Dalam menggunakan XP model, proses tes dilakukan dengan menggunakan metode autometed testing dengan katalon. Pada system pengintegrasian ini didapat system sudah sesuai dengan keperluan dan juga tidak ditemukan error. Setiap tes dilakukan secara simultan sejak pengintegrasian dibuat seperti yang ditunjukkan pada tabel 4.5

Tabel 4. 5 Hasil Pengujian Nilai

Kode User Stories	Fungsi	Result
Story-03	Mengintegrasikan data Nilai pada <i>e-learning</i> dengan data Nilai di SIAKAD	<i>Passed</i>

4.3.5 Restrospective

Tabel 4. 6 Restrospective Nilai

Iterasi 3			
User Story	Story Point	Estimasi	Pengerjaan
Story-03	3	3	3

4.4 Analisis Integration System

Pada dasarnya pengujian integrasi akan menguji masing-masing modul terlebih dahulu, diintegrasikan, lalu program akan diuji secara keseluruhan. Langkah pertama untuk melakukan pengujian pada penelitian ini adalah dengan melihat apakah Api dari e-learning sudah bisa dijalankan sesuai kebutuhan atau belum, lalu sistem akan diintegrasikan dengan SIAKAD. Data yang berhasil diintegrasikan serta tahapan penggunaan bisa dilihat pada table di bawah.

Tabel 4. 7 Hasil Data Integrasi

User Story	<i>e-learning</i>	SIAKAD	Status
Jurnal Perkuliahan	core_course name (Materi Kuliah) modules – nama (sap) modules - modname (Keterangan)	jurnal_perkuliahan materi_kuliah sap keterangan	passed
Abseni	mod_attendance nim grade	presensi Nim absensi	passed
Nilai	gradereport_user itemmodule gradeformatted	krs_nilai tugas_nilaiangka quiz_nilaiangka uts_nilaiangka uas_nilaiangka	passed

Tabel 4. 8 Tahapan Integrasi

User Story	Tahapan uji coba integrasi
Jurnal Perkuliahan	<ol style="list-style-type: none"> 1. Buka <i>e-learning</i> 2. Pilih Mtakuliah 3. Pilih Pertemuan 4. Tambahkan atau edit pertemuan 5. Buka Siakad 6. Buka fitur Jurnal Perkuliahan 7. Data Berhasil Terinput
Abseni	<ol style="list-style-type: none"> 1. Buka <i>e-learning</i> 2. Pilih Mtakuliah 3. Lakukan absensi pada menu absen 4. Buka SIAKAD 5. Buka fitur absensi 6. Data berhasil teinput
Nilai	<ol style="list-style-type: none"> 1. Buka <i>e-learning</i> 2. Pilih Mtakuliah 3. Lakukan penilaian pada tugas atau quiz 4. Buka SIAKAD 5. Pilih fitur input nilai 6. Data akan tampil 7. Masukkan persentase penilaian 8. Lalu klik input

Selanjutnya adalah melakukan *tes process analysiss*. *Test Process Analysis* adalah suatu tahapan testing dimana ringkasan dari setiap testing akan

dikumpulkan dan dianalisis dengan tujuan tertentu. Pada tahapan ini proses penganalisisan bertujuan untuk menentukan *test step* yang paling sedikit dan waktu terpendek dalam penggunaan antar sistem yang belum terintegrasi dengan sistem yang sudah terintegrasi. Proses analisis berdasarkan *tes cases* dan *test execute* yang sudah dijalankan pada tahapan sebelumnya.

a. Jurnal Perkuliahan

Penganalisisan sistem Jurnal perkuliahan berdasarkan hasil dari testing sebelumnya, yaitu sistem sebelum dan sesudah integrasi. Hasil rangkuman testing diatas dapat dilihat pada tabel di bawah :

Tabel 4. 9Perbandingan Jurnal Perkuliahan Sebelum dan Sesudah Integrasi

No	Sistem	Test Step	Waktu (detik)	Status
1	Sistem Jurnal Perkuliahan Sebelum Integrasi (Import data Jurnal Perkuliahan)	13	54,717	berhasil
2	Sistem Jurnal Perkuliahan Sesudah Integrasi (Integrasi data Jurnal Perkuliahan)	8	36,684	berhasil

Dari tabel diatas dapat terlihat bahwa test step dan juga waktu yang dibutuhkan dalam penggunaan sistem. Pengujian test cases pada sistem sebelum integrasi didapat 12 test step serta mendapat penambahan satu step dalam penginputan

autentikasi dan 8 test step setelah melakukan integrasi. Sedangkan untuk waktu yang diperoleh menunjukkan 54,717 detik untuk sistem sebelum integrasi dan juga 36,684 sesudah sistem integrasi. Kedua sistem berstatus *berhasil* yang berarti sistem berjalan sesuai keinginan dan tanpa error.

b. Absensi

Analisis sistem Absensi perkuliahan berdasarkan hasil dari testing sebelumnya, yaitu sitem sebelum dan sesudah integrasi, namun pada tahapan sistem sebelumnya terdapat dua tahap *export* dan *inport*. Pada tahapan ini sistem sebelumnya *export* data dari *e-learning* dan *inport* data ke SIAKAD akan langsung dihitung menjadi satu. Hasil rangkuman testing pada tahap sebelumnya dapat dilihat pada tabel di bawah :

Tabel 4. 10 Perbandingan Absensi Sebelum dan Sesudah Integrasi

No	Sistem	Test Step	Waktu (detik)	Status
1	Sistem Absensi Sebelum Integrasi (Export dan Import data Absensi)	23	87,521	berhasil
2	Sistem Absensi Sesudah Integrasi (Integrasi data Absensi)	7	32,977	berhasil

Pada rangkuman *testing* tabel diatas dapat terlihat bahwa *test step* dan juga waktu yang dibutuhkan dalam penggunaan sitem. Pengujian test cases pada sistem sebelum integrasi didapat 23 test step dan 7 test step setelah melakukan integrasi.

Sedangkan untuk waktu yang diperoleh menunjukkan 87,521 detik untuk sistem absensi sebelum integrasi dan juga 32,977 sesudah sistem integrasi. Kedua sistem berstatus *berhasil* yang berarti sistem berjalan sesuai keinginan dan tanpa error.

c. Nilai

Sistem Nilai perkuliahan pada analisis ini, berdasarkan hasil dari testing sebelumnya, yaitu sitem sebelum dan sesudah integrasi, namun pada tahapan sistem nilai sebelumnya terdapat dua tahap *export* dan *inport*. Pada tahapan ini sistem sebelumnya *export* data dari *e-learning* dan *inport* data ke SIAKAD akan langsung dihitung menjadi satu. Hasil rangkuman testing pada tahap sebelumnya dapat dilihat pada tabel di bawah :

Tabel 4.11Perbandingan Sebelum dan Sesudah Integrasi

No	Sistem	Test Step	Waktu (detik)	Status
1	Sistem Nilai Sebelum Integrasi (Export dan Import data Nilai)	22	115.32	berhasil
2	Sistem Nilai Sesudah Integrasi (Integrasi data Nilai)	13	41,576	berhasil

Rangkuman *testing* pada tabel diatas dapat terlihat bahwa *test step* dan juga waktu yang dibutuhkan dalam penggunaan sitem. Pengujian test cases pada sistem nilai sebelum integrasi didapat 22 test step dan 7 test step setelah melakukan integrasi. Sedangkan untuk waktu yang diperoleh menunjukkan 113,324 detik untuk sistem

absensi sebelum integrasi dan juga 32,977 sesudah sistem integrasi. Kedua sistem berstatus *berhasil* yang berarti sistem berjalan sesuai keinginan dan tanpa error.

4.5 Analisis Penggunaan *Personal Extreme Programming*

Setelah penggunaan *personal extreme programming* dalam pengembangan pengintegrasian SIAKAD dengan *e-learning*, metode pengembangan akan diamati seberapa efisien dan efektif metode ini diterapkan pada pengembangan pengintegrasian ini. Analisis ini dilandasi dari penelitian sebelumnya yang berjudul “Personal Extreme Programming – An Agile Process for Autonomous Developers” pada penelitian ini mengamati kinerja pengembang dalam menggunakan metode *agile*, dalam jurnal tersebut diantaranya mengamati selisih antara waktu aktual dan waktu yang direncanakan. Efisien adalah usaha yang mengharuskan penyelesaian pekerjaan dengan tepat waktu, cepat dan memuaskan. (ARYO, 2019)

Dalam menentukan efisiensi penggunaan *pxp*, ditentukan dengan membandingkan waktu pengerjaan secara nyata dengan waktu estimasi yang dibutuhkan. Dalam menentukan tingkat efisiensi berdasarkan waktu estimasi dengan waktu penyelesaian user story tanpa memperhatikan iterasi. (Dzhurov et al., 2009) Dengan demikian persamaan dapat direpresentasikan sebagai berikut:

$$Efficiency = \frac{Estimated\ time}{Task\ completion\ time} \times 100\%$$

Dalam perhitungannya didapat efisiensi penggunaan *pxp* adalah 97%.

$$Efficiency = \frac{(2 + 4 + 3)}{(2.5 + 4 + 3)} \times 100\% = 97\%$$

4.6 Pengembangan Integrasi Sistem Informasi Akademik dengan *E-learning* dalam Pandangan Islam

Pesan pertama agama Islam yang diajarkan Allah kepada Rasulullah Muhammad *shalallaahu alaihi wassalaam*, melalui malaikat Jibril merupakan model awal komunikasi pembelajaran dalam konteks pendidikan Islam. Komunikasi pembelajaran pada tahap ini berlangsung secara manual-tradisional, tanpa sentuhan teknologi kreatif, terutama teknologi komunikasi. Rasulullah Muhammad *shalallaahu alaihi wassalaam* berupaya menyimpan informasi yang diterimanya dengan cara menghafalnya, sehingga informasi itu dapat diberikan kembali saat diminta persis sebagaimana ia diterima. Melalui kemampuan daya hafal Muhammad *shalallaahu alaihi wassalaam* dan sahabat-sahabatnya terhadap ayat-ayat al-Qur'an dan hadis, sehingga tidak satu pun ayat al-Qur'an yang mengalami distorsi sebagai bukti jaminan Allah tetap memelihara kemurniannya. (Wajdi, 2010) Hal ini dapat menjadi sinyal betapa perlunya bantuan teknologi untuk menyamai kemampuan mereka, khusus bagi umat yang tidak memiliki daya kognitif seperti mereka.

Dalam agama islam, menuntut ilmu hukumnya wajib. Dalam hadis yang di riwayatkan oleh Ibnu Majah,

طَلَبُ الْعِلْمِ فَرِيضَةٌ عَلَى كُلِّ مُسْلِمٍ

“Menuntut ilmu itu wajib atas setiap Muslim” (HR. Ibnu Majah no. 224, dari sahabat Anas bin Malik radhiyallahu ‘anhu, dishahihkan Al Albani dalam Shahiih al-Jaami’ish Shaghiir no. 3913). Pada ranah pendidikan sering kita jumpai beberapa aplikasi untuk membantu guru atau dosen dalam mengajar siswa atau

mahasiswanya. Aplikasi ini sering digunakan untuk pembelajaran jarak jauh maupun hanya untuk pengumpulan tugas. Aplikasi dalam dunia pendidikan sudah sangat sering digunakan, bahkan tidak sedikit pelaku pendidikan yang sangat tergantung dalam penggunaannya. Dengan adanya bantuan teknologi, maka proses belajar bisa dipercepat, mudah dan sebagainya. Dalam kajian Psikologi Kognitif, kemampuan ingatan seseorang terbagi kepada tiga jenis yaitu kemampuan menerima, menyimpan dan memunculkan kembali informasi. Ingatan mempunyai pengaruh yang cukup besar terhadap keberhasilan pembelajaran. Semakin baik kemampuan ingatan seseorang, maka semakin banyak informasi yang dapat dia terima, simpan, dan munculkan kembali. (Fauzi, 2019) Dengan bantuan Teknologi Informasi dan Komunikasi, kemampuan menerima, menyimpan, memunculkan kembali ini dapat diakselerasi, sehingga lebih efektif dan efisien. Penyimpangan informasi dan pengrusakan selama dalam perjalanan dapat dihindari. Dengan demikian, keberhasilan pembelajaran dapat ditingkatkan. Maka, apa jadinya dalam penggunaan aplikasi terjadi kendala atau aplikasi masih belum terintegrasi pada aplikasi satu dengan yang lainnya. Hal ini dapat memperlambat proses belajar mengajar.

Di sinilah terlihat, betapa Teknologi Informasi dan Komunikasi memainkan peranan yang sangat penting dalam membantu manusia mengolah pesan, menyimpan dan memunculkannya kembali saat diminta guna menghindari distorsi dan reduksi semantik. Salah satu bentuk sinyal Teknologi Informasi dan Komunikasi terawal dalam Islam adalah proses saving data dengan cara menghafal ayat-ayat yang diturunkan secara periodik kepada Muhammad

shalallaahu alaihi wassalaam dengan metode manual. Proses ini sebenarnya merupakan sinyal akan kebutuhan teknologi yang lebih canggih di kemudian hari, sebagai pengembangan teknik menghafal menggunakan multimedia. Kewajiban tentang belajar dan pembelajaran Firman Allah QS. al-Nahl (16):125

أَدْخِ إِلَى سَبِيلِ رَبِّكَ بِالْحِكْمَةِ وَالْمَوْعِظَةِ الْحَسَنَةِ ۚ وَجِدْ لَهُم بِأَلَّتِي هِيَ أَحْسَنُ ۚ إِنَّ رَبَّكَ هُوَ أَعْلَمُ بِمَنْ

ضَلَّ عَنْ سَبِيلِهِ ۚ وَهُوَ أَعْلَمُ بِالْمُهْتَدِينَ

Artinya: “Serulah (manusia) kepada jalan Tuhan-mu dengan hikmah dan pelajaran yang baik dan bantahlah mereka dengan cara yang baik. Sesungguhnya Tuhanmu Dialah yang lebih mengetahui tentang siapa yang tersesat dari jalan-Nya dan Dialah yang lebih mengetahui orang-orang yang mendapat petunjuk.”

Menurut Tafsir Al-Muyassar adalah “Serulah (wahai rasul) oleh mu dan orang-orang yang mengikutimu kepada agama tuhanmu dan jalanNya yang lurus dengan cara bijaksana yang telah Allah wahyukan kepadamu di dalam al-qur’an dan -sunnah. Dan bicaralah kepada manusia dengan metode yang sesuai dengan mereka, dan nasihati mereka dengan baik-baik yang akan mendorong mereka menyukai kebaikan dan menjauhkan mereka dari keburukan. Dan debatlah mereka dengan cara perdebatan yang terbaik, dengan halus dan lemah lembut. sebab tidak ada kewajiban atas dirimu selain menyampaikan, Dan sungguh engkau telah menyampaikan, adapun hidayah bagi mereka terserah kepada Allah semata. Dia lebih tahu siapa saja yang sesat dari jalanNya dan Dia lebih tahu orang-orang yang akan mendapatkan hidayah.” Khusus untuk QS. al-Nahl (16):

125 di atas, adalah berkenaan dengan kewajiban belajar dan pembelajaran serta metodenya. Dalam ayat ini, Allah *subhanahu wa ta'ala* menyuruh dalam arti mewajibkan kepada Nabi Muhammad *shalallaahu alaihi wassalaam* dan umatnya untuk belajar dan mengajar dengan menggunakan metode pembelajaran yang baik. Dalam metode pembelajaran jarak jauh, sangat penting adanya system yang saling terintegrasi. (Wakka, 2020) Sesuai perjalanan waktu dan perkembangan kemajuan teknologi kreatif yang semakin canggih, dengan bantuan ilmu pengetahuan dan teknologi yang diilhamkan Allah kepada manusia, belakangan ini, ditemukan berbagai konsep yang tertuang dalam teknik informatika untuk membantu mempermudah manusia berkomunikasi dan mengirim informasi dengan lebih akurat, efektif dan efisien. Pada beberapa kampus negeri maupun swasta, sudah banyak yang menerapkan pembelajaran secara daring, melalui platform edukasi mereka masing masing. Kampus ini sudah menerapkan platform edukasi *e-learning* guna kegiatan pembelajaran secara jarak jauh ataupun daring, tetapi semakin banyaknya platform edukasi tanpa terintegrasi dari system satu dengan yang lain malah akan membuat pembelajaran tidak efektif. Dengan adanya system terintegrasi proses belajar mengajar akan semakin baik sesuai dengan perintah Allah *subhanahu wa ta'ala* dalam Al Quran.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan dengan mengimplementasikan *Personal Extreme Programming* dalam pengembangan integrasi Sistem Informasi Akademik dan *E-learning*, maka dapat diambil kesimpulan sebagai berikut:

1. Pada pengembangan perangkat lunak pengintegrasian Sistem Informasi Akademik dan *E-learning* menggunakan metode Extreme Programming berhasil dilakukan. Hasil tersebut didasarkan pada system yang sudah berjalan sesuai *recuirement*. Sedangkan tingkat efisiensi berdasarkan perbandingan waktu estimasi dengan waktu sebenarnya mencapai 97%, hal ini terjadi karena ada hal tidak teduga pada iterasi pertama.
2. Dalam pengintegrasian Sistem Informasi Akademik dan *E-learning* menggunakan methode *rest api* dikatakan system bisa berjalan lebih cepet dan memerlukan sedikit tahapan untuk menjalankannya daripada system sebelumnya. Pada pengintegrasian system Jurnal perkuliahan, hanya memerlukan 8 tahapan dan juga memerlukan waktu 36,684 detik

sedangkan untuk system sebelumnya memerlukan 13 tahapan dan juga memerlukan waktu 54,717 detik. Pada pengintegrasian system Absensi, hanya memerlukan 7 tahapan dan juga memerlukan waktu 32,977 detik sedangkan untuk system sebelumnya memerlukan 23 tahapan dan juga memerlukan waktu 87,521 detik. Pada pengintegrasian system Nilai, hanya memerlukan 13 tahapan dan juga memerlukan waktu 41,576 detik sedangkan untuk system sebelumnya memerlukan 22 tahapan dan juga memerlukan waktu 115.32 detik

5.2 Saran

Penelitian ini tentunya tidak terlepas dari kekurangan dan kelemahan. Pada penelitian ini sistem pengintegrasian diasumsikan ketika dosen menggunakan *e-learning* secara penuh dalam pembelajaran. Oleh karena itu, untuk kebaikan pengembangan sistem selanjutnya maka penulis menyarankan user dosen juga dapat menambahkan data atau mengintegrasikan data dari platform yang lain. Pada penelitian selanjutnya peneliti mengharapkan adanya perbandingan penggunaan metode agile dengan metode yang lainnya.

DAFTAR PUSTAKA

- Andry, J. F (2019). Perancangan Sistem Informasi Proyek Manajemen Menggunakan Metode Extreme Programming Berbasis Desktop (Studi Kasus: Perusahaan Kontraktor). *SIMADA (Jurnal Sistem Informasi Dan Manajemen Basis Data)*, 2(2), 104–113.
- Fatoni, A., & Dwi, D (2016). Rancang bangun sistem extreme programming sebagai metodologi pengembangan sistem. *PROSISKO: Jurnal Pengembangan Riset Dan Observasi Sistem Komputer*, 3(1).
- Homaidi, A (2016). Sistem Informasi Akademik AMIK Ibrahimy Berbasis Web. *Jurnal Ilmiah Informatika*, 1(1), 17–23.
- Maturbongs, Y. H (2019). *E-Learning Sebagai Pendidikan Tinggi*. 43–57.
- Agustin, H (2019). *Sistem Informasi Manajemen Dalam Perspektif Islam*. Rajawali Pers.
- Ariyanti, L., Satria, M. N. D., & Alita, D (2020). Sistem Informasi Akademik Dan Administrasi Dengan Metode Extreme Programming Pada Lembaga Kursus Dan Pelatihan. *Jurnal Teknologi Dan Sistem Informasi*, 1(1), 90–96.
- Bora, A., & Bezboruah, T (2015). A comparative investigation on implementation of RESTful versus SOAP based web services. *International Journal of Database Theory and Application*, 8(3), 297–312.
- Dewi, N. A (2021). PERANCANGAN APLIKASI E-LEARNING BERBASIS WEB DENGAN METODE EXTREME PROGRAMMING BAGI SISWA (STUDI KASUS: SDN SITU GEDE 05 BOGOR). *Jurnal ESIT (E-Bisnis)*,

Sistem Informasi, Teknologi Informasi), 16(1).

Eriyatno, L. L (2012). Ilmu Sistem: Meningkatkan Mutu dan Efektivitas Manajemen. *Bogor (ID): Guna Widya*.

Liatmaja, R., & Wardati, I. U (2013). Sistem Informasi Akademik Berbasis Web Pada Lembaga Bimbingan Belajar Be Excellent Pacitan. *IJNS-Indonesian Journal on Networking and Security*, 2(2).

Qowaid, Q., Junaedi, D., Romli, M., & Primarni, A (2020). Analisis Persepsi Civitas Akademika terhadap Implementasi Perkuliahan E-Learning Selama Pandemi Covid-19. *Reslaj: Religion Education Social Laa Roiba Journal*, 2(2), 114–141.

Rizal, H., Adhy, S., & Wirawan, P. W (2014). Perancangan Dan Pembuatan Mobile Learning Interaktif Berbasis Android Dengan Metode Personal Extreme Programming. *Journal of Informatics and Technology*, 2(3), 103–112..

Siahaan, S (2003). E-Learning (Pembelajaran Elektronik) sebagai salah satu alternatif kegiatan pembelajaran. *Jurnal Pendidikan Dan Kebudayaan*, 9(42), 303–321.

Pratiwi, P. M (2016). Analisis Peran Startup Corner Pada Bandung Techno Park Dalam Pengembangan Startup Berbasis Teknologi Melalui Pendekatan Model Timmons. *Bandung: Universitas Telkom*

Herlinda, H., Katarina, D., Ambarsari, E. W., & Kom, M (2019). Automation testing tool dalam pengujian aplikasi belajar tajwid pada platform Android. *STRING (Satuan Tulisan Riset Dan Inovasi Teknologi)*, 4(2), 205–212.

- Kosasih, Y., & Cahyono, A. B (2021). Automation Testing Tool Dalam Pengujian Aplikasi The Point Of Sale. *AUTOMATA*, 2(1).
- Bernadhi, B. D., & Saptadi, S (2016). Perancangan Integrasi Sistem Informasi Akademik dengan E-Learning (Studi Kasus: Program Studi XYZ). *J@ Ti Undip: Jurnal Teknik Industri*, 11(3), 177–186.
- Rizal, R., & Rahmatulloh, A (2019). Restful Web Service Untuk Integrasi Sistem Akademik Dan Perpustakaan Universitas Perjuangan. *Jurnal Ilmiah Informatika*, 7(01), 54–59.
- ARYO, K. A. J. I. (2019). *PENGGUNAAN PORT MANAGEMENT SYSTEM (PMS) DALAM PROSES PERENCANAAN WAKTU BONGKAR MUAT KAPAL UNTUK EFEKTIVITAS DAN EFISIENSI KERJA PADA PT. VARIA USAHA BAHARI GRESIK*. POLITEKNIK ILMU PELAYARAN SEMARANG.
- Dzhurov, Y., Krasteva, I., & Ilieva, S. (2009). *Personal Extreme Programming—An Agile Process for Autonomous Developers*.
- Fauzi, I. (2019). Pembelajaran Perspektif Psikologi Sufistik Imam Al-Ghazali Dan Psikologi Humanistik Abraham Maslow Dalam Pembentukan Kepribadian. *Journal of Teaching Dan Learning Research*, 1(2), 77–100.
- Hanani, A. (2020). Integrasi Sistem Informasi Akademik dan E-Learning UIN Maulana Malik Ibrahim Malang berbasis Web Service REST. *SMARTICS Journal*, 6(1), 17–24.
- Kumar, G., & Bhatia, P. K. (2014). Comparative analysis of software engineering models from traditional to modern methodologies. *2014 Fourth International*

Conference on Advanced Computing & Communication Technologies, 189–196.

Raharjana, I. K. (2017). *Pengembangan Sistem Informasi Menggunakan Metodologi Agile*. Deepublish.

Wajdi, F. (2010). *Tahfiz al-qur'an dalam kajian 'ulum al-qur'an (studi atas berbagai metode tahfiz)*. Jakarta: Sekolah Pascasarjana UIN Syarif Hidayatullah, 2008.

Wakka, A. (2020). Petunjuk Al-Qur'an Tentang Belajar Dan Pembelajaran. *Education and Learning Journal*, 1(1), 82–92.