

**PENGEMBANGAN DASHBOARD PEMBAGIAN KELOMPOK  
MAHASISWA BERBASIS *ARTIFICIAL INTELLIGENCE*  
DENGAN METODE *AGILE KANBAN***

**TUGAS AKHIR**

Diajukan sebagai syarat menyelesaikan jenjang strata Satu (S-1) di  
Program Studi Teknik Informatika, Fakultas Teknologi Industri, Institut  
Teknologi Sumatera

**Oleh:**

**Galin Nichola Gibran**

**121140050**



**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI SUMATERA  
LAMPUNG SELATAN  
2025**

## LEMBAR PENGESAHAN

Saya menyatakan bahwa Tugas Akhir berjudul “Pengembangan Dashboard Pembagian Kelompok Mahasiswa Berbasis *Artificial Intelligence* dengan Metode *Agile Kanban*” merupakan hasil karya saya sendiri dan belum pernah diajukan, baik sebagian maupun seluruhnya, di Institut Teknologi Sumatera atau institusi pendidikan lain oleh saya maupun pihak lain.

Lampung Selatan, 3 November 2025  
Penulis,

Galin Nichola Gibran  
NIM. 121140050

Foto 2x3

Diperiksa dan disetujui oleh,  
Pembimbing

1. Hafiz Budi Firmansyah  
NIP. 19910824 201903 1 014 .....  
.....
2. Hafiz Budi Firmansyah  
NIP. 19910824 201903 1 014 .....  
.....

Penguji

1. Dosen Penguji I  
NIP. 19900000 2000 00 0 000 .....  
.....
2. Dosen Penguji II  
NIP. 19900000 2000 00 0 000 .....  
.....

Disahkan oleh,  
Koordinator Program Studi Teknik Informatika  
Fakultas Teknologi Industri  
Institut Teknologi Sumatera

Andika Setiawan, S.Kom., M.Cs.  
NIP. 19911127 2022 03 1 007

## HALAMAN PERNYATAAN ORISINALITAS

Tugas Akhir dengan judul “Pengembangan Dashboard Pembagian Kelompok Mahasiswa Berbasis *Artificial Intelligence* dengan Metode *Agile Kanban*” adalah karya saya sendiri, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan benar.

**Nama : Galin Nichola Gibran**

**NIM : 121140050**

**Tanda Tangan : .....**

**Tanggal : .....**

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI  
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

Sebagai civitas akademik Institut Teknologi Sumatera, saya yang bertanda tangan di bawah ini:

Nama : Galin Nichola Gibran

NIM : 121140050

Program Studi : Teknik Informatika

Fakultas : Teknologi Industri

Jenis Karya : Tugas Akhir

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Institut Teknologi Sumatera **Hak Bebas Royalti Noneksklusif (Non-exclusive Royalty Free Right)** atas karya ilmiah saya yang berjudul:

**Pengembangan Dashboard Pembagian Kelompok Mahasiswa Berbasis**

***Artificial Intelligence dengan Metode Agile Kanban***

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Institut Teknologi Sumatera berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Lampung Selatan

Pada tanggal : 3 November 2025

Yang menyatakan

Galin Nichola Gibran

## KATA PENGANTAR

*Pada halaman ini mahasiswa berkesempatan untuk menyatakan terima kasih secara tertulis kepada pembimbing dan pihak lain yang telah memberi bimbingan, nasihat, saran dan kritik, kepada mereka yang telah membantu melakukan penelitian, kepada perorangan atau lembaga yang telah memberi bantuan keuangan, materi dan/atau sarana. Cara menulis kata pengantar beraneka ragam, tetapi hendaknya menggunakan kalimat yang baku. Ucapan terima kasih agar dibuat tidak berlebihan dan dibatasi pada pihak yang terkait secara ilmiah (berhubungan dengan subjek/materi penelitian).*

Puji syukur kehadirat Allah SWT/Tuhan Yang Maha Esa atas limpahan rahmat, karunia, serta petunjuk-Nya sehingga penyusunan tugas akhir ini telah terselesaikan dengan baik. Dalam penyusunan tugas akhir ini penulis telah banyak mendapatkan arahan, bantuan, serta dukungan dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. [Rektor ITERA] selaku Rektor Institut Teknologi Sumatera.
2. [Dekan FTI] selaku Dekan Fakultas Teknologi Industri.
3. [Koor Prodi IF] selaku Ketua Program Studi Teknik Informatika.
4. [Dosen Pembimbing] selaku Dosen Pembimbing atas ide, waktu, tenaga, perhatian, dan masukan yang telah disumbangsihkan kepada penulis.
5. [Isi nama lainnya]

Akhir kata penulis berharap semoga tugas akhir ini dapat memberikan manfaat bagi kita semua.

## RINGKASAN

Pengembangan Dashboard Pembagian Kelompok Mahasiswa Berbasis *Artificial*

*Intelligence dengan Metode Agile Kanban*

Galin Nichola Gibran

*Halaman Ringkasan berisi uraian singkat tentang latar belakang masalah, rumusan masalah, tujuan, metodologi penelitian, hasil dan analisis data, serta kesimpulan dan saran. Isi ringkasan tidak lebih dari 1000 kata (sekitar maksimal 2 halaman).*

  Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

  Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

## ABSTRAK

Pengembangan Dashboard Pembagian Kelompok Mahasiswa Berbasis *Artificial Intelligence* dengan Metode *Agile Kanban*

Galin Nichola Gibran

Halaman ABSTRAK berisi uraian tentang latar belakang, tujuan, metodologi penelitian, hasil / kesimpulan. Ditulis dalam BAHASA INDONESIA tidak lebih dari 250 kata, dengan jarak antar baris satu spasi. Pada akhir abstrak ditulis kata “Kata Kunci” yang dicetak tebal, diikuti tanda titik dua dan kata kunci yang tidak lebih dari 5 kata. Kata kunci terdiri dari kata-kata yang khusus menunjukkan dan berkaitan dengan bahan yang diteliti, metode/instrumen yang digunakan, topik penelitian. Kata kunci diketik pada jarak dua spasi dari baris akhir isi abstrak.

**Kata Kunci:** **kunci1, kunci2**

## **ABSTRACT**

Development of Student Group Division Dashboard Based on Artificial Intelligence with Agile Kanban Method

Galin Nichola Gibran

Halaman ABSTRACT berisi uraian tentang latar belakang, tujuan, metodologi penelitian, hasil / kesimpulan. Ditulis dalam BAHASA INGGRIS tidak lebih dari 250 kata, dengan jarak antar baris satu spasi. Secara khusus, kata dan kalimat pada halaman ini tidak perlu ditulis dengan huruf miring meskipun menggunakan Bahasa Inggris, kecuali terdapat huruf asing lain yang ditulis dengan huruf miring (misalnya huruf Latin atau Greek, dll). Pada akhir abstract ditulis kata “Keywords” yang dicetak tebal, diikuti tanda titik dua dan kata kunci yang tidak lebih dari 5 kata. Keywords terdiri dari kata-kata yang khusus menunjukkan dan berkaitan dengan bahan yang diteliti, metode/instrumen yang digunakan, topik penelitian. Keywords diketik pada jarak dua spasi dari baris akhir isi abstrak.

**Keywords:** **keywords1, keywords2**

## DAFTAR ISI

<b>LEMBAR PENGESAHAN .....</b>	<b>ii</b>
<b>HALAMAN PERNYATAAN ORISINALITAS .....</b>	<b>iii</b>
<b>HALAMAN PERSETUJUAN PUBLIKASI .....</b>	<b>iv</b>
<b>KATA PENGANTAR .....</b>	<b>v</b>
<b>RINGKASAN .....</b>	<b>vi</b>
<b>ABSTRAK .....</b>	<b>vii</b>
<b>ABSTRACT .....</b>	<b>viii</b>
<b>DAFTAR ISI .....</b>	<b>ix</b>
<b>DAFTAR TABEL .....</b>	<b>xi</b>
<b>DAFTAR GAMBAR .....</b>	<b>xii</b>
<b>DAFTAR RUMUS .....</b>	<b>xiii</b>
<b>DAFTAR KODE .....</b>	<b>xiv</b>
<b>BAB I Pendahuluan .....</b>	<b>1</b>
1.1 Latar Belakang Masalah .....	1
1.2 Rumusan Masalah .....	5
1.3 Tujuan Penelitian .....	5
1.4 Batasan Masalah .....	5
1.5 Manfaat Penelitian .....	6
1.6 Sistematika Penulisan .....	6
1.6.1 Bab I Pendahuluan .....	6
1.6.2 Bab II Tinjauan Pustaka .....	6
1.6.3 Bab III Metode Penelitian .....	7

<b>BAB II TINJAUAN PUSTAKA .....</b>	<b>8</b>
2.1 Tinjauan Pustaka .....	8
2.2 Dasar Teori .....	13
2.2.1 Sistem Informasi .....	13
2.2.2 Dashboard .....	13
2.2.3 Artificial Intelligence .....	14
2.2.4 EduTeams .....	14
2.2.5 Edu2Com .....	15
2.2.6 Agile Kanban .....	16
2.2.6.1 Work In Progress (WIP) .....	17
2.2.7 Unified Modeling Language (UML) .....	17
2.2.8 Entity Relationship Diagram .....	20
2.2.9 Grey Box Testing .....	21
<b>BAB III METODE PENELITIAN .....</b>	<b>23</b>
3.1 Alur Penelitian .....	23
3.2 Penjabaran Langkah Penelitian .....	24
3.2.1 Identifikasi Masalah .....	24
3.2.2 Studi Literatur .....	24
3.2.3 Implementasi Metode Agile Kanban .....	24
3.2.4 Pembahasan .....	25
3.2.5 Kesimpulan .....	25
3.3 Alat dan Bahan Tugas Akhir .....	25
3.3.1 Alat .....	25
3.3.2 Bahan .....	26
3.4 Metode Pengembangan .....	26
3.4.1 Tahap Perencanaan .....	26
3.4.1.1 Kebutuhan Fungsional .....	26
3.4.1.2 Kebutuhan Non-Fungsional .....	28

3.4.1.3	Backlog Awal .....	28
3.4.1.4	Use Case Diagram .....	31
3.4.1.5	Activity Diagram .....	31
3.4.1.6	<i>Entity Relationship Diagram (ERD)</i> .....	53
3.4.1.7	Tahap Pengembangan .....	57
3.4.1.8	Tahap Pengujian .....	57
<b>BAB IV HASIL DAN PEMBAHASAN .....</b>		<b>61</b>
4.1	Hasil Penelitian .....	61
4.1.1	Implementasi Fitur .....	61
4.2	Hasil Pengujian .....	66
4.3	Analisis Hasil Penelitian .....	67
<b>BAB V KESIMPULAN DAN SARAN .....</b>		<b>68</b>
5.1	Kesimpulan .....	68
5.2	Saran .....	68
<b>DAFTAR PUSTAKA .....</b>		<b>69</b>
<b>LAMPIRAN .....</b>		<b>73</b>
A	Dataset .....	73
B	Hasil Wawancara .....	73
C	Rincian Kasus Uji .....	73

## **DAFTAR TABEL**

Tabel 2.1	Tinjauan pustaka penelitian terdahulu .....	9
Tabel 2.2	Notasi <i>Use Case Diagram</i> .....	18
Tabel 2.3	Notasi <i>Activity Diagram</i> .....	19
Tabel 2.4	Notasi <i>Entity Relationship Diagram</i> .....	20
Tabel 2.5	Metode Pengujian <i>Grey Box Testing</i> .....	21
Tabel 3.1	Kebutuhan Fungsional .....	27
Tabel 3.2	Kebutuhan Non-Fungsional .....	28
Tabel 3.3	Tabel Backlog Awal .....	29
Tabel 3.4	Skenario Pengujian <i>Grey Box Testing</i> .....	58
Tabel 4.1	Implementasi Fitur Sistem EduTeams .....	61
Tabel 4.2	Data <i>dummy</i> Pengujian .....	66

## **DAFTAR GAMBAR**

Gambar 3.1 Alur Penelitian .....	23
Gambar 3.2 <i>Use Case Diagram</i> Sistem Pembagian Kelompok .....	31
Gambar 3.3 <i>Activity Diagram Login</i> .....	32
Gambar 3.4 <i>Activity Diagram Registrasi Akun</i> .....	34
Gambar 3.5 <i>Activity Diagram Logout</i> .....	35
Gambar 3.6 <i>Activity Diagram</i> Melihat Hasil Pembagian Kelompok ..	36
Gambar 3.7 <i>Activity Diagram</i> Melihat Persebaran MBTI, Keahlian, Preferensi, dan Jenis Kelamin Mahasiswa .....	37
Gambar 3.8 <i>Activity Diagram</i> Melihat Penjelasan MBTI .....	38
Gambar 3.9 <i>Activity Diagram</i> Mengganti Bahasa .....	39
Gambar 3.10 <i>Activity Diagram</i> Membuat Kelas .....	40
Gambar 3.11 <i>Activity Diagram</i> Membuat Tugas .....	41
Gambar 3.12 <i>Activity Diagram</i> Membuat Kuesioner Skill .....	42
Gambar 3.13 <i>Activity Diagram</i> Membuat Kuesioner Preferensi .....	43
Gambar 3.14 <i>Activity Diagram</i> Melihat Riwayat Pertanyaan Kuesioner	44
Gambar 3.15 <i>Activity Diagram</i> Membagikan Tautan Kuesioner .....	45
Gambar 3.16 <i>Activity Diagram</i> Melihat Daftar Mahasiswa .....	46
Gambar 3.17 <i>Activity Diagram</i> Melihat Jawaban Kuesioner Mahasiswa	47
Gambar 3.18 <i>Activity Diagram</i> Melakukan Pembagian Kelompok.....	48
Gambar 3.19 <i>Activity Diagram</i> Melihat Kualitas Kelompok .....	49
Gambar 3.20 <i>Activity Diagram Import</i> Hasil Pembagian Kelompok ...	50
Gambar 3.21 <i>Activity Diagram</i> Mengerjakan Kuesioner .....	51
Gambar 3.22 <i>Activity Diagram</i> Melihat Hasil Kuesioner Sendiri .....	52
Gambar 3.23 <i>Activity Diagram</i> Melihat Riwayat Pembagian Kelompok	53
Gambar 3.24 <i>Entity Relationship Diagram</i> (ERD) .....	56
Gambar 4.1 Contoh Graf Pengujian .....	67

## **DAFTAR RUMUS**

**DAFTAR KODE**

Kode 4.1 Akuisisi Gambar . . . . .	65
------------------------------------	----

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang Masalah**

Kemampuan manusia untuk berkolaborasi merupakan fondasi fundamental dalam membangun peradaban modern. Di era globalisasi dan transformasi digital, kolaborasi menjadi semakin kritis seiring dengan meningkatnya kompleksitas tantangan global seperti perubahan iklim, kesehatan publik, dan inovasi teknologi. Penelitian terbaru menunjukkan bahwa manusia memiliki dorongan kuat untuk bekerja sama dengan orang lain, yang menjadi penggerak utama transformasi sosial di era digital [1].

Dalam konteks pendidikan, khususnya pembelajaran kolaboratif, kolaborasi memainkan peran krusial dalam meningkatkan keterampilan akademik dan sosial siswa. Pembelajaran kolaboratif melibatkan siswa yang bekerja sama secara aktif untuk mencapai tujuan bersama, yang tidak hanya meningkatkan pemahaman konseptual tetapi juga mengembangkan keterampilan interpersonal dan pemecahan masalah [2]. Menurut penelitian tentang peran pembelajaran kolaboratif dalam pendidikan online, kolaborasi memungkinkan siswa mengembangkan keterampilan berpikir kritis dan komunikasi serta mengurangi perasaan isolasi yang sering dirasakan dalam pembelajaran individu [3]. Seiring dengan berkembangnya kompleksitas dunia modern, institusi pendidikan kini dihadapkan pada tantangan untuk mempersiapkan siswa menghadapi lingkungan yang semakin terkoneksi dan dinamis. Keberhasilan di masa depan tidak hanya bergantung pada penguasaan materi akademik, tetapi juga pada kemampuan untuk berkolaborasi secara efektif dalam tim yang beragam. Hal ini menjadikan pembelajaran kolaboratif bukan hanya sebagai metode pengajaran, tetapi sebagai komponen vital dalam membentuk generasi yang siap menghadapi tantangan masa depan.

Selain itu, kolaborasi merupakan salah satu keterampilan lulusan yang fundamental dalam pendidikan tinggi dan sering menjadi atribut utama dalam penilaian kualitas pendidikan. Kolaborasi yang efektif membantu siswa mengembangkan kemampuan bekerja dalam tim, berpikir kritis, dan keterampilan pemecahan masalah yang sangat relevan dengan dunia kerja. Ellis et al. (2021) menekankan bahwa kolaborasi dalam kelompok kecil dengan pendekatan pembelajaran mendalam, siswa dapat bekerja lebih efektif dan saling mendukung dalam memahami materi [4].

Meskipun memiliki banyak manfaat, pembelajaran kolaboratif juga menghadapi tantangan signifikan, terutama dalam pembentukan kelompok yang seimbang. Kesulitan ini sering muncul karena kurangnya informasi tentang keterampilan, preferensi, dan dinamika pribadi siswa, terutama di awal semester ketika pengajar belum familiar dengan profil siswa [5]. Pengajar biasanya harus membentuk kelompok secara manual dengan mengamati langsung interaksi siswa, menilai keterampilan dan preferensi mereka berdasarkan pengalaman pribadi, atau menggunakan metode pengelompokan sederhana seperti pembagian acak. Pendekatan ini tidak hanya memakan waktu tetapi juga sering gagal menghasilkan kelompok yang optimal. Akibatnya, terjadi berbagai masalah seperti ketidakseimbangan keterampilan antar anggota kelompok, konflik interpersonal, dan partisipasi yang tidak merata. Hal ini dapat menurunkan efektivitas kolaborasi, menurunkan motivasi siswa, dan pada akhirnya mempengaruhi kualitas hasil pembelajaran [6]. Fenomena ini menjadi semakin menantang dalam konteks pembelajaran hybrid dan daring, di mana interaksi tatap muka terbatas dan pengajar memiliki kesulitan lebih besar dalam memahami dinamika kelas. Oleh karena itu, dibutuhkan sistem pembentukan tim yang dapat mempertimbangkan berbagai faktor ini untuk meningkatkan kualitas pembelajaran kolaboratif.

Termotivasi dari permasalahan di atas, satu solusi inovatif yang menjanjikan datang dari Spanish National Research Council (CSIC). CSIC

adalah institusi penelitian publik terbesar di Spanyol yang berdidikasi untuk penelitian imiah dan teknologi di berbagai bidang termasuk bidang edukasi [7]. Mereka mengembangkan platform yang bernama EduTeams, dibuat sebagai sistem pembagian kelompok bernama dalam lingkungan kelas secara otomatis menggunakan bantuan kecerdasan buatan atau Artificial Intelligence (AI) berdasarkan keterampilan, kepribadian dan preferensi mahasiswa. Sistem ini bekerja dengan mengumpulkan data yang diperlukan dari setiap mahasiswa di kelas seperti tingkat keterampilan, kepribadian dan preferensi dari tugas yang diberikan. Setelah data selesai dikumpulkan, setiap mahasiswa akan terbagi secara merata berdasarkan data yang telah mereka isi [8].

Saat ini tim pengembang EduTeams menyediakan *Application Programming Interface* (API) [9] untuk algoritma EduTeams yang bernama Edu2Com yang akan digunakan oleh peneliti untuk mengembangkan sistem pembagian kelompok menggunakan kecerdasan buatan. Untuk mengimplementasikan sistem pembagian kelompok dengan kecerdasan buatan, diperlukan pengembangan dengan performa yang baik dan dapat dijalankan oleh semua pengguna. Oleh karena itu, peran pengembangan (*development*) sangat penting untuk berlangsungnya sistem yang berjalan dengan mulus [10]. Faktor lain yang harus dipertimbangkan untuk kelancaran proses pengembangan adalah metode *Software Development Life Cycle* (SDLC) yang akan digunakan.

Dalam proyek ini, kami mengadopsi *Agile Kanban* sebagai kerangka kerja *SDLC*, yang berfokus pada visualisasi alur kerja dan pembatasan pekerjaan-dalam-proses atau *work in progress* (WIP). Pendekatan ini sangat kontras dengan metode berbasis iterasi seperti *Scrum*, yang memaksakan progress yang telah ditentukan sebelumnya pada tim, berpotensi menyebabkan keterlambatan dalam beradaptasi dengan perubahan yang muncul dari hasil pengembangan dan umpan balik dari pengujian sistem. Alaidaros et al. (2021) mencatat bahwa kanban memungkinkan penyesuaian proses yang berkelanjutan, sesuai dengan kebutuhan siklus pengembangan iteratif dalam pengembangan sistem

dan optimasi pengembangan. Peneliti secara khusus akan melacak metrik kunci seperti waktu siklus rata-rata untuk menyelesaikan fitur, tingkat throughput fitur baru, dan jumlah WIP. Dengan demikian, ketahanan kami dalam menanggapi kebutuhan yang berkembang seiring dengan berjalannya kebutuhan dan desain dari sistem [11].

Dalam proses pengembangan, peneliti menggunakan teknik *grey box testing*, yang menggabungkan teknik *black box* (pengujian dari perspektif pengguna, tanpa pengetahuan tentang kode internal) dengan teknik *white box* (verifikasi struktur kode dan logika). Tidak seperti pengujian *black box murni*, yang mungkin gagal menemukan kesalahan yang berkaitan dengan pemrosesan data atau integrasi api, *grey box testing* memungkinkan pemeriksaan komponen dan alur data kunci tanpa memerlukan akses penuh ke kode sumber yang mendasarinya [12]. Khususnya, peneliti akan menggunakan skenario *grey box* untuk memvalidasi kebenaran data yang diterima dari API edu2com, memastikan bahwa struktur data JSON diproses dengan benar dan bahwa respons kesalahan ditangani dengan tepat. Hal ini dicapai melalui analisis model data API edu2com dan menghasilkan kasus pengujian yang menargetkan alur transformasi data yang kritis. Selain itu, kami akan menggunakan *grey box testing* untuk memverifikasi implementasi algoritma pembentukan tim yang dipandu oleh AI. Dengan berfokus pada perilaku sistem terhadap berbagai kombinasi data siswa, bertujuan untuk mengidentifikasi potensi bias, anomali, atau perilaku tak terduga yang mungkin tidak muncul melalui pengujian *black box* saja.

Penelitian ini sangat relevan dalam konteks tantangan pendidikan modern, yang menekankan inklusi dan kesetaraan. Meskipun sistem pembentukan tim berbasis AI memiliki potensi untuk meningkatkan keseimbangan dalam hal gender, latar belakang, dan keterampilan, kami menyadari bahwa teknologi saja tidak dapat menjamin hasil yang adil. Sebaliknya, kami bertujuan untuk mengeksplorasi bagaimana sistem ini dapat digunakan secara bertanggung jawab

untuk mendukung pengambilan keputusan manusia, memberikan wawasan tentang dinamika kelompok yang mungkin terlewatkan oleh pengajar, dan menciptakan lingkungan belajar yang lebih inklusif.

## 1.2 Rumusan Masalah

Dari permasalahan yang telah diidentifikasi pada latar belakang, maka rumusan masalah yang menjadi fokus penelitian adalah sebagai berikut:

1. Bagaimana merancang sistem pembagian kelompok berbasis kecerdasan buatan yang efektif menggunakan API Edu2Com?
2. Bagaimana hasil pengujian terhadap sistem pembagian kelompok berbasis kecerdasan buatan yang efektif menggunakan API Edu2Com?

## 1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Merancang dan mengembangkan sistem pembagian kelompok berbasis AI menggunakan API Edu2Com dari CSIC, yang dapat mengotomatisasi proses pembentukan tim berdasarkan keterampilan, kepribadian, dan preferensi siswa.
2. Mengimplementasikan metode Agile Kanban dalam proses pengembangan perangkat lunak untuk memastikan fleksibilitas dan kontinuitas perbaikan selama pengembangan.
3. Menggunakan metode *grey box testing* untuk menguji fungsionalitas sistem secara menyeluruh.

## 1.4 Batasan Masalah

Agar penelitian ini terfokus dan dapat diselesaikan dengan efektif dalam waktu yang ditentukan, maka ditetapkan beberapa batasan masalah sebagai berikut:

1. Penelitian hanya dilakukan pada platform EduTeams.
2. Pengembangan dashboard dibuat hanya berbasis website.

3. Fokus pengembangan hanya pada dashboard pembagian kelompok pada sisi mahasiswa dan dosen.

## **1.5 Manfaat Penelitian**

Adapun manfaat dari penelitian ini sebagai berikut:

1. Membantu dosen atau tenaga pendidik dalam membagi kelompok pada mahasiswa sesuai dengan jenis kelamin, keahlian, kepribadian, dan preferensi.
2. Mempermudah mahasiswa dalam menemukan kelompok

## **1.6 Sistematika Penulisan**

Dalam sistematika penulisan ini dijabarkan poin-poin dari isi setiap bab. Sistematika penulisan pada penelitian adalah sebagai berikut :

### **1.6.1 Bab I Pendahuluan**

Pada penelitian ini, Bab I Pendahuluan akan membahas pengenalan penelitian yang mencakup Latar Belakang Masalah, Rumusan Masalah, Tujuan Penelitian, Batasan Masalah, Manfaat Penelitian, serta Sistematika Penulisan. Bagian ini memberikan gambaran awal mengenai pentingnya pengembangan dashboard pembagian kelompok berbasis *Artificial Intelligence* menggunakan API Edu2Com dan metodologi Agile Kanban sebagai solusi dalam meningkatkan efektivitas kolaborasi mahasiswa.

### **1.6.2 Bab II Tinjauan Pustaka**

Pada penelitian ini, Bab II Tinjauan Pustaka akan mendiskusikan landasan teori dan kajian pustaka yang relevan, seperti konsep Sistem Informasi, Dashboard, *Artificial Intelligence*, serta algoritma dan parameter yang digunakan pada API Edu2Com. Selain itu, akan dibahas pula metode Agile Kanban, *Unified Modeling Language (UML)*, *Entity Relationship Diagram (ERD)*, dan metode *Grey Box Testing* yang mendukung pengembangan sistem ini. Kajian pustaka

bertujuan untuk memberikan dasar teoritis yang kuat dalam pengembangan dan pengujian sistem.

### **1.6.3 Bab III Metode Penelitian**

Pada penelitian ini, Bab III Metode Penelitian akan menjelaskan lebih rinci mengenai alur penelitian, langkah-langkah penelitian, alat dan bahan yang digunakan, serta metode pengembangan. Tahapan yang dijelaskan meliputi perencanaan fitur, implementasi metodologi Agile Kanban, dan pengujian sistem menggunakan *Grey Box Testing*. Selain itu, Bab III juga akan membahas pengelolaan data dan skenario pengujian

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Tinjauan Pustaka**

Tinjauan Pustaka ini disusun untuk memberikan gambaran komprehensif tentang penelitian terdahulu yang relevan dengan pembentukan kelompok berbasis kecerdasan buatan (AI) dan metode pengembangan perangkat lunak menggunakan Agile Kanban. Analisis dilakukan dengan mengelompokkan literatur berdasarkan dua pendekatan utama, yaitu:

1. Masalah yang diangkat dalam pembentukan kelompok kolaboratif.
2. Metode yang digunakan dalam pengembangan dan implementasi sistem.

Pembahasan setiap kategori dirangkum secara sistematis dalam tabel untuk mempermudah pembaca memahami kontribusi dan keterbatasan dari penelitian terdahulu. Detailnya dapat dilihat pada Tabel 2.1

Tabel 2.1 Tinjauan pustaka penelitian terdahulu

No.	Judul	Masalah	Metode	Hasil
1.	<i>An Apporach to Group Formation in Collaborative Learning Using Learning Paths in Learning Management System</i>	Fokus pada kelompok dengan kemampuan serupa, kurang membahsa kelompok dengan kemampuan berbeda atau data di luar dari <i>Learning Management System</i>	Algoritma k-means clustering dengan metrik Euclidean, Manhattan dan cosine	Framework meningkatkan pembentukan kelompok melalui analisis perilaku siswa, 75% siswa mengalami peningkatan nilai.
2.	<i>A Learner-Centered Technique for Collectively Configuring Inputs for an Algorithmic Team Formation Tool</i>	Berfokus pada konfigurasi manual berbasis preferensi siswa.	Workflow berbasis preferensi siswa melalui survei dan diskusi	Memberikan siswa kendali dalam menentukan kriteria pembentukan tim, meningkatkan kepuasan dan persepsi keadilan

No.	Judul	Masalah	Metode	Hasil
3	<i>FERN: Fair Team Formation for Mutually Beneficial Collaborative Learning</i>	Berfokus pada keadilan dan manfaat kolaboratif, lebih kompleks dibandingkan pendekatan berbasis API otomatis.	Multi-objektif optimisasi dengan algoritma heuristik (FERN).	Meningkatkan keadilan antar kelompok dengan optimisasi manfaat individu dan kelompok.
4	<i>Software Project Management Systems Using Kanban Method in CV. Primavisi Globalindo</i>	Fokus pada proyek perangkat lunak umum; memerlukan adaptasi untuk konteks pendidikan.	Agile Kanban	Sistem Kanban meningkatkan fleksibilitas dan respons terhadap perubahan proyek; validitas sistem diuji dengan hasil 100%

No.	Judul	Masalah	Metode	Hasil
5	<i>Implementation of Kanban Techniques in Software Development Process: An Empirical Study Based on Benefits and Challenges</i>	Fokus pada pengembangan perangkat lunak umum, membutuhkan adaptasi untuk konteks pendidikan.	Studi empiris tentang penerapan Kanban dalam pengembangan perangkat lunak, melibatkan 241 responden dari 67 perusahaan.	Mengidentifikasi peningkatan efisiensi tim, komunikasi, dan pengelolaan risiko dengan penerapan Kanban.

Penelitian pertama oleh Ramos et al. (2021) menggunakan algoritma k-means clustering untuk membentuk kelompok berdasarkan data perilaku siswa yang diperoleh dari LMS [13]. Hasilnya menunjukkan bahwa framework ini meningkatkan performa siswa dalam kolaborasi kelompok, dengan 75% siswa mencatat peningkatan nilai. Namun, pendekatan ini terbatas pada data LMS dan tidak membahas dinamika kelompok dan faktor non-akademik. Dalam konteks penelitian ini, metode Ramos et al. memberikan dasar untuk mengembangkan sistem pembentukan kelompok berbasis AI dengan mempertimbangkan faktor tambahan, seperti preferensi individu dan keterampilan interpersonal.

Penelitian kedua oleh Hastings et al. (2022) memperkenalkan pendekatan berbasis preferensi siswa dalam menentukan bobot kriteria untuk alat pembentukan tim algoritmik [14]. Studi ini menunjukkan bahwa siswa lebih puas ketika mereka dilibatkan dalam proses penentuan kriteria, dengan prioritas tertinggi pada komitmen kursus dan kesesuaian jadwal. Meskipun studi ini berfokus pada konfigurasi manual berbasis survei dan diskusi, pendekatan ini dapat mendukung sistem EduTeams dengan memungkinkan preferensi siswa untuk meningkatkan keadilan dan efektivitas dalam pembentukan tim.

Penelitian ketiga oleh Kalantzi et al. (2020) memperkenalkan FERN, sebuah kerangka kerja untuk pembentukan tim berbasis keadilan dalam pembelajaran kolaboratif [15]. Studi ini mengidentifikasi tantangan dalam memastikan keadilan dan manfaat tim, terutama terkait atribut yang dilindungi seperti gender dan ras. FERN menggunakan algoritma heuristik kompleks yang mempertimbangkan atribut yang dilindungi seperti gender dan ras, sambil mengoptimalkan manfaat individu dan kolektif dalam kelompok. Meskipun metode ini kompleks, hasil menunjukkan peningkatan signifikan dalam keadilan tim dan manfaat kolaboratif. Relevansi penelitian ini terletak pada kemampuan untuk mengatasi keterbatasan metode tradisional seperti pengelompokan acak.

Penelitian keempat oleh Ilmi et al. (2020) mengeksplorasi penerapan Kanban dalam sistem manajemen proyek berbasis web untuk perusahaan perangkat lunak [16]. Studi ini menyoroti keunggulan Kanban dalam visualisasi tugas, pengendalian *Work In Progress* (WIP), dan fleksibilitas terhadap perubahan. Meskipun konteksnya adalah industri perangkat lunak, metode ini dapat diadaptasi untuk mendukung pengembangan sistem EduTeams di penelitian ini, yang juga membutuhkan pengelolaan tugas yang dinamis.

Penelitian kelima oleh Riaz (2020) meneliti penerapan metode Kanban dalam proses pengembangan perangkat lunak melalui survei dan wawancara dengan profesional dari 67 perusahaan [17]. Studi ini menemukan bahwa Kanban meningkatkan efisiensi tim, mengurangi siklus pengembangan, dan menciptakan transparansi dalam pengelolaan proyek. Meskipun konteksnya adalah perangkat lunak umum, prinsip-prinsip Kanban seperti visualisasi alur kerja dan pembatasan WIP dapat diadaptasi untuk mendukung pengembangan sistem EduTeams dalam penelitian ini.

Tinjauan pustaka ini mengungkap bahwa pembentukan kelompok kolaboratif bukanlah sekadar proses mekanis, melainkan sistem kompleks yang memerlukan pendekatan multidimensional. Penelitian sebelumnya telah menunjukkan berbagai strategi: dari clustering berbasis algoritma hingga

pendekatan yang mempertimbangkan preferensi siswa dan isu keadilan.

Berbeda dari penelitian sebelumnya, penelitian ini bertujuan untuk mengembangkan sistem pembentukan kelompok kolaboratif yang efisien dan praktis. Fokus utama penelitian ini adalah memanfaatkan API Edu2Com untuk otomatisasi pembentukan kelompok berdasarkan keterampilan, preferensi, dan kebutuhan tugas, serta menggunakan metodologi Agile Kanban untuk memastikan pengembangan sistem berjalan secara adaptif. Dengan pendekatan ini, penelitian ini tidak hanya menyelesaikan masalah teknis dalam pembentukan kelompok, tetapi juga memastikan kelompok yang terbentuk mendukung pembelajaran kolaboratif yang optimal dan memperhatikan aspek manusiawi seperti keadilan dan dinamika interpersonal.

## 2.2 Dasar Teori

Berisi teori/konsep yang berkaitan/digunakan dalam tugas akhir yang dikerjakan. Gunakanlah data melalui buku/jurnal referensi, publikasi tugas akhir, penelitian, buku, dan informasi web yang dapat dipertanggungjawabkan, hindari penggunaan dasar teori melalui tautan Wikipedia, surat kabar, atau portal berita, yang dapat memiliki isi yang tidak bersifat fakta.

### 2.2.1 Sistem Informasi

Sistem informasi adalah salah satu teknologi yang diperlukan untuk memudahkan pencarian informasi yang dibutuhkan serta mengelola data dengan lebih efektif dan efisien [18]. Sistem informasi juga dapat diartikan sebagai sebuah sistem dalam organisasi yang mengintegrasikan manusia, teknologi, sistem, media, fasilitas, prosedur, serta pengendalian dengan tujuan untuk menciptakan alur informasi dan transaksi yang lebih mudah dan terstruktur.

### 2.2.2 Dashboard

*Dashboard* adalah alat visualisasi data yang dapat diintegrasikan untuk tujuan tertentu, memungkinkan pengguna memantau aktivitas yang sedang

berlangsung sekaligus mendukung proses pengambilan keputusan dan kebijakan [19]. Dalam dunia pendidikan, *dashboard* memiliki fungsi untuk memantau dan menganalisis kinerja mahasiswa maupun dosen. Umumnya, sebuah dashboard dilengkapi dengan komponen seperti grafik, tabel, indikator, filter, serta data yang disajikan secara real-time [20]. Dengan dashboard, pengguna seperti mahasiswa dan dosen dapat lebih mudah mengakses dan memahami data yang telah dimasukkan atau diolah, karena disajikan dalam bentuk visual yang informatif.

### **2.2.3 Artificial Intelligence**

*Artificial Intelligence* merupakan istilah dalam bahasa Inggris yang terdiri dari kalimat buatan dan kecerdasan. Kecerdasan Buatan melibatkan sejumlah teknik dan metode, termasuk pembelajaran mesin, pemrosesan bahasa alami, penglihatan komputer, dan pendekatan kecerdasan buatan lainnya. Tujuan utama *Artificial Intelligence* (AI) adalah menciptakan teknologi yang memungkinkan sebuah mesin untuk dapat menirukan perilaku manusia, memahami suatu pola, pemecahan masalah, dan pengambilan keputusan [21].

### **2.2.4 EduTeams**

EduTeams adalah sebuah platform inovatif yang dirancang untuk mendukung kegiatan kolaboratif dalam konteks pendidikan. Platform ini memanfaatkan prinsip-prinsip kecerdasan buatan untuk meningkatkan efektivitas pembentukan dan pengelolaan kelompok kerja. Dengan fitur-fitur seperti pembagian tugas otomatis, penilaian kolektif, dan komunikasi tim, EduTeams membantu institusi pendidikan menciptakan kelompok yang lebih produktif dan efisien. Selain itu, platform ini memungkinkan adaptasi terhadap gaya belajar dan tingkat keterampilan pengguna yang berbeda, sehingga dapat meningkatkan kolaborasi dan hasil belajar.

Pengembangan EduTeams dipimpin oleh *Artificial Intelligence Research*

*Institute* (IIIA), salah satu unit di bawah *Spanish National Research Council* (CSIC), lembaga penelitian terbesar di Spanyol yang berada di bawah Kementerian Sains dan Inovasi Spanyol. Didirikan pada tahun 1939, CSIC memiliki peran utama dalam memajukan ilmu pengetahuan melalui riset dasar dan terapan di berbagai bidang, termasuk sains, teknologi, dan humaniora. Sebagai salah satu lembaga riset paling bergengsi di Eropa, CSIC mendorong inovasi teknologi melalui kolaborasi dengan institusi akademik dan industri, baik di tingkat nasional maupun internasional [7].

### 2.2.5 Edu2Com

Edu2Com adalah algoritma *anytime heuristic* berbasis Feasible Team-For-Task Allocation Problem (FTAP) yang diimplementasikan melalui API REST untuk pembentukan tim mahasiswa secara optimal. Dirancang oleh tim pengembang EduTeams untuk menyelesaikan tantangan dalam dunia pendidikan, khususnya dalam hal penempatan mahasiswa pada pembagian kelompok. Algoritma ini bertujuan untuk membentuk tim-tim mahasiswa yang paling sesuai untuk melaksanakan tugas yang akan dikerjakan. Edu2Com bekerja dengan mempertimbangkan berbagai faktor penting, seperti keterampilan yang dibutuhkan untuk tugas tertentu, kemampuan yang dimiliki oleh mahasiswa, serta ukuran tim yang diperlukan [8]. Algoritma ini menggabungkan empat parameter utama ( $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $\delta$ ) untuk menyeimbangkan faktor:

1. Kesesuaian Keterampilan ( $\alpha$ ): Menghitung kesenjangan antara level keterampilan mahasiswa (`skills.level`) dan persyaratan tugas (`tasks.skills.level`).
2. Kompatibilitas Kepribadian ( $\beta$ ): Menganalisis dimensi MBTI (misalnya, *extraversion/introversion*) untuk meminimalkan konflik interpersonal.
3. Preferensi Tugas ( $\gamma$ ): Memprioritaskan mahasiswa berdasarkan preferensi tugas (`taskPreference`) yang diinput melalui formulir.
4. Kohesi Tim ( $\delta$ ): Mengoptimalkan kemiripan keterampilan antaranggota

(skillSimilarity.similarity) untuk kolaborasi efektif.

### 2.2.6 Agile Kanban

Kanban adalah metode manajemen alur kerja visual yang berfungsi sebagai sistem penjadwalan yang memberikan informasi tentang apa yang perlu dilakukan, kapan harus dilakukan, dan dalam jumlah berapa. Inti dari Kanban adalah visualisasi proses kerja melalui papan yang dibagi menjadi kolom-kolom yang mewakili tahapan pekerjaan (seperti Backlog, In Progress, Testing, dan Done). Setiap tugas atau pekerjaan diwakili oleh kartu yang bergerak melintasi kolom-kolom tersebut, mencerminkan status dan tahapan pekerjaan [11].

Berbeda dengan metodologi berbasis iterasi seperti Scrum yang memaksakan progress yang telah ditentukan sebelumnya pada tim, Kanban berfokus pada visualisasi alur kerja dan pembatasan pekerjaan-dalam-proses (Work-In-Progress atau WIP). Pendekatan ini memungkinkan penyesuaian proses yang berkelanjutan, sesuai dengan kebutuhan siklus pengembangan iteratif dalam pengembangan sistem dan optimasi pengembangan. Metrik kunci yang sering dilacak dalam Kanban meliputi waktu siklus rata-rata untuk menyelesaikan fitur, tingkat throughput fitur baru, dan jumlah antrian pekerjaan (WIP) [11].

Metode ini sangat fleksibel dan dapat disesuaikan dengan berbagai lingkungan kerja, menjadikannya alat yang sangat efektif untuk memperbaiki alur kerja secara berkelanjutan dan meningkatkan kualitas hasil akhir. Dalam pengembangan perangkat lunak, Kanban telah diadaptasi sebagai metodologi Agile, yang membantu tim dalam mengelola dan mengoptimalkan proses pengembangan secara lebih iteratif dan responsif, terutama dalam situasi dengan kebutuhan yang berubah-ubah dan prioritas yang bervariasi.

### 2.2.6.1 Work In Progress (WIP)

*Work In Progress* (WIP) adalah jumlah pekerjaan yang sedang berlangsung dalam sistem pada waktu tertentu, namun belum selesai. Dalam konteks Kanban, WIP merujuk pada semua kartu atau *work item* yang telah berada dalam kartu *Progress* dan belum mencapai kolom "Done" [22]. Konsep WIP menjadi fundamental dalam metodologi Kanban karena berkaitan erat dengan efisiensi aliran kerja dan kualitas hasil akhir.

Membatasi WIP (*WIP limits*) adalah salah satu prinsip inti dalam Kanban yang bertujuan untuk mencegah tim mengambil terlalu banyak pekerjaan secara bersamaan. Ketika tim memiliki terlalu banyak pekerjaan yang sedang berlangsung, hal ini dapat menyebabkan multitasking yang berlebihan, meningkatkan *lead time*, dan menurunkan kualitas hasil kerja. Dengan menetapkan batas WIP, tim dapat mempertahankan aliran kerja yang stabil dan fokus pada penyelesaian tugas yang ada sebelum memulai pekerjaan baru [23].

### 2.2.7 Unified Modeling Language (UML)

*Unified Modeling Language* (UML) merupakan model visual yang sangat berguna dalam proses pengembangan sistem. UML memungkinkan pengembang untuk menciptakan *blueprint* sistem yang akan dibuat. UML terdiri dari beberapa diagram yang membantu pengembang dalam mengomunikasikan sistem yang dirancang. Beberapa diagram tersebut meliputi flowchart, diagram konteks, use case, dan desain basis data [24]. Berikut ini adalah notasi yang akan digunakan untuk menyusun *Use Case Diagram* dan *Activity Diagram*.

Tabel 2.2 Notasi *Use Case Diagram*

Nama	Simbol	Keterangan
Actor		Menggambarkan pengguna atau entitas eksternal yang berinteraksi dengan sistem
Use Case		Menggambarkan urutan interaksi antar <i>actor</i> dengan sistem
System		Menggambarkan Lingkup spesifikasi dari fitur sistem
Association		Menggambarkan sebuah penghubung antara objek dengan objek lainnya
Include		Menggambarkan bahwa suatu <i>use case</i> secara keseluruhan adalah fungsionalitas <i>use case</i> lainnya
Extend		Menggambarkan bahwa suatu <i>use case</i> adalah tambahan fungsional dari <i>use case</i> lain jika terpenuhi kondisinya

Berikut ini adalah tabel 2.3 di bawah berisi notasi yang akan digunakan untuk menyusun *Activity Diagram*.

Tabel 2.3 Notasi *Activity Diagram*

Nama	Simbol	Keterangan
<i>Initial Node</i>		Menggambarkan status awal dari suatu aktivitas dalam sistem
<i>Final Node</i>		Menggambarkan status akhir dari suatu aktivitas dalam sistem
<i>Action</i>		Menggambarkan suatu aksi yang dilakukan pada suatu aktivitas dalam sistem
<i>Decision</i>		Menggambarkan percabangan dari aksi pada suatu aktivitas dalam sistem
<i>Join Node</i>		Menggambarkan gabungan antara dua atau lebih dari alur aktivitas menjadi satu aktivitas dalam satu waktu.
<i>Fork Node</i>		Menggambarkan pemisahan dari suatu alur aktivitas menjadi dua atau lebih aktivitas dalam satu waktu

Nama	Simbol	Keterangan
<i>Swimlane</i>		Menggambarkan ruang lingkup dari <i>actor</i> yang mengeksekusi atau melakukan serangkaian aktivitas dalam sistem.
<i>Connector</i>		Menggambarkan alur dari suatu aktivitas

### 2.2.8 Entity Relationship Diagram

*Entity relationship diagram* (ERD) merupakan suatu diagram yang dimanfaatkan merancang hubungan antar tabel dalam sebuah basis data. ERD juga dapat digunakan untuk menggambarkan kebutuhan data dari sebuah organisasi [25]. ERD paling sering digunakan untuk merepresentasikan apapun yang dibentuk dari basis data dikarenakan kesederhanaan dan diagram yang mudah dipahami [26]. Berikut adalah beberapa komponen yang terdapat pada ERD yang dapat dilihat pada Tabel 2.4.

Tabel 2.4 Notasi *Entity Relationship Diagram*

Nama	Simbol	Keterangan
<i>Entitas</i>		Entitas adalah suatu objek unik yang dapat diidentifikasi dalam lingkungan pemakai.
<i>Relasi</i>		Relasi adalah suatu objek untuk menunjukkan hubungan antara sejumlah entitas.

Nama	Simbol	Keterangan
Atribut		Atribut adalah suatu objek properti dari entitas atau relasi.
Garis	—	Garis adalah suatu objek penghubung antara relasi dengan entitas dan entitas dengan atribut.

### 2.2.9 Grey Box Testing

*Grey Box Testing* adalah metode pengujian perangkat lunak yang menggabungkan elemen-elemen dari *Black Box* dan *White Box Testing*, untuk memanfaatkan pengetahuan terbatas tentang struktur internal sistem dan merancang skenario pengujian yang lebih komprehensif. Pendekatan ini memungkinkan cakupan pengujian sistem yang lebih luas dengan menggabungkan akses terbatas ke algoritma dan struktur data internal dengan pengujian berbasis fungsionalitas eksternal [12], [27]. Pendekatan Grey Box Testing menggunakan teknik pengujian, seperti yang dijelaskan di Tabel 2.5 dibawah [28]:

Tabel 2.5 Metode Pengujian *Grey Box Testing*

Metode Pengujian	Interface Program	Status Program	Alur Program	Fungsional
<i>Matrix Test</i>	Tidak	Ya	Tidak	Ya
<i>Regression Test</i>	Tidak	Ya	Ya	Tidak
<i>Pattern Test</i>	Ya	Tidak	Ya	Ya
<i>Orthogonal Array Test</i>	Ya	Ya	Ya	Ya

*Matrix Test*: fokus pada status program, seperti validasi data atau hasil akhir tanpa memeriksa alur program atau antarmuka.

*Regression Test:* menguji stabilitas sistem terhadap perubahan kode, termasuk pengaruhnya terhadap alur program.

*Pattern Test:* menguji bagaimana antarmuka dan fungsionalitas program bekerja secara terintegrasi.

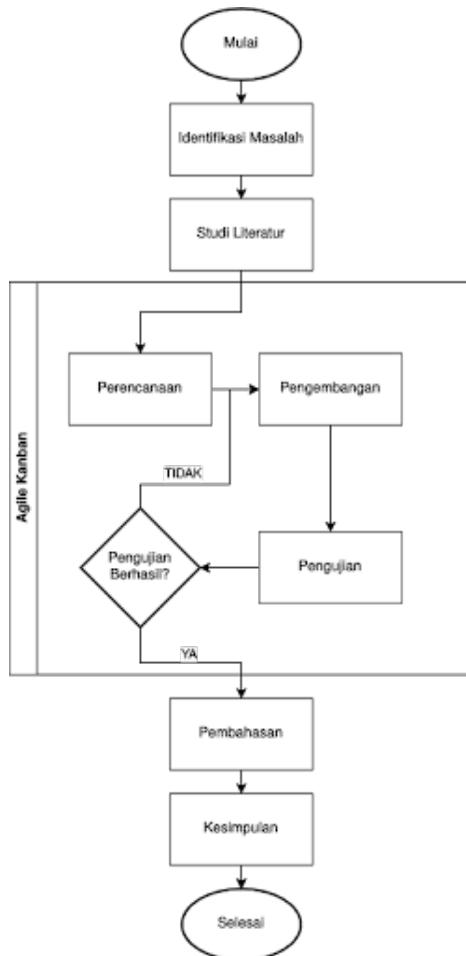
*Orthogonal Array Test:* mengombinasikan pengujian antarmuka, status, alur program, dan fungsionalitas secara menyeluruh.

## BAB III

### METODE PENELITIAN

#### 3.1 Alur Penelitian

Adapun alur penelitian pengembangan sistem pembagian kelompok yang dapat dilihat pada gambar 3.1.



Gambar 3.1 Alur Penelitian

### 3.2 Penjabaran Langkah Penelitian

Bagian ini akan menjelaskan secara detail tentang diagram alur penelitian yang telah dibuat.

#### 3.2.1 Identifikasi Masalah

Tahap ini mengidentifikasi permasalahan dalam proses pembagian kelompok mahasiswa yang dilakukan secara manual. Metode tradisional, seperti pembagian berdasarkan nomor urut, daftar absen, atau pertemanan, sering kali menghasilkan kelompok yang tidak seimbang. Masalah ini dapat menyebabkan ketidakseimbangan keterampilan antaranggota, konflik pribadi, dan motivasi belajar yang menurun. Sebagai solusi, sistem berbasis *Edu2Com API*, yang dikembangkan oleh *CSIC*, memungkinkan pembagian kelompok secara otomatis berdasarkan 4 parameter utama: kesesuaian keterampilan (*skill match*), kompatibilitas kepribadian, preferensi tugas, dan kohesi tim. Berdasarkan hasil observasi, sistem ini memerlukan *dashboard* yang dapat digunakan oleh dosen dan mahasiswa. Penelitian ini fokus pada pengembangan sistem berbasis metodologi *Agile Kanban*.

#### 3.2.2 Studi Literatur

Pada tahap ini, dilakukan kajian pustaka dengan menganalisis metode dan pendekatan yang mencakup studi-studi terdahulu yang berkaitan dengan pembentukan kelompok berbasis *AI* dan metode pengembangan perangkat lunak. Materi literatur ini dirangkum dalam tabel kajian pustaka pada bab 2 untuk memberikan gambaran terkait kontribusi penelitian sebelumnya serta relevansinya dengan penelitian yang dilakukan.

#### 3.2.3 Implementasi Metode Agile Kanban

Melakukan penelitian dan pengembangan dengan menggunakan metode *Agile Kanban* yang akan dijelaskan pada sub-bab 3.4. Dalam pengembangan menggunakan metode *Agile Kanban*, tahapan yang dilakukan meliputi 4 tahapan

mulai dari perencanaan, pengembangan, dan pengujian/evaluasi sistem.

### 3.2.4 Pembahasan

Hasil penelitian akan dibahas secara mendetail, mencakup analisis kinerja sistem, tingkat efektivitas pembagian kelompok, dan kemudahan penggunaan *dashboard*. Evaluasi dilakukan secara kuantitatif berdasarkan *Grey Box Testing*. Jika terdapat masalah atau kekurangan pada bagian pengujian, iterasi pengembangan dilakukan sebanyak 2 kali untuk memastikan semua komponen bekerja optimal.

### 3.2.5 Kesimpulan

Tahap akhir penelitian ini menyimpulkan hasil pengembangan sistem pembagian kelompok. Hasil kesimpulan ini menekankan pencapaian utama dari penelitian serta membuka peluang untuk pengembangan lebih lanjut.

## 3.3 Alat dan Bahan Tugas Akhir

Berisi alat-alat dan bahan-bahan yang digunakan dalam penelitian.

### 3.3.1 Alat

Alat yang digunakan untuk melakukan penelitian yaitu:

1. Laptop dengan spesifikasi:
  - (a) *Processor AMD Ryzen 5 5500U (12) @ 4.06 GHz*
  - (b) *RAM 16GB*
  - (c) *Sistem Operasi Fedora Linux 64-bit*
  - (d) *SSD 512GB*
2. *Zed Editor*
3. *Bun*
4. *Git*
5. *Helium Browser*
6. *LaTeX*

## 7. Draw.io

### 3.3.2 Bahan

Bahan yang digunakan/diperlukan untuk melakukan penelitian, dapat berupa:

1. Pertanyaan *MBTI* yang diperoleh dari *Openpsychometrics* yang bersifat *open source*.
2. Spesifikasi *API Edu2Com* untuk layanan pembentukan kelompok otomatis.

## 3.4 Metode Pengembangan

### 3.4.1 Tahap Perencanaan

Tahap perencanaan meliputi identifikasi kebutuhan pengguna, pembuatan daftar fitur, dan perumusan *backlog* awal. Diskusi dilakukan bersama calon pengguna (dosen), rekan proyek, serta *brainstorming* mandiri untuk menentukan fitur prioritas yang akan dikembangkan. *Backlog* disusun berdasarkan kebutuhan pengguna yang dikumpulkan dari proses wawancara *stakeholder* (Dosen dan Mahasiswa Informatika Institut Teknologi Sumatera) dan ditampilkan dalam papan *Kanban* dengan kolom-kolom: *To Do*, *In Progress*, *Testing*, dan *Done*. Pembatasan *WIP* diberlakukan untuk menjaga fokus pengembang dan menghindari *multitasking* berlebih, untuk penelitian ini *WIP* dibatasi menjadi 2 progres per tugas pengembangan.

#### 3.4.1.1 Kebutuhan Fungsional

Menganalisis kebutuhan fungsional sistem berdasarkan proses-proses yang akan dijalankan oleh sistem, disesuaikan dengan kebutuhan pengguna. Dari analisis tersebut, dapat dirumuskan beberapa poin pada tabel 3.1 berikut:

Tabel 3.1 Kebutuhan Fungsional

<b>Kode</b>	<b>Aktor Pengguna</b>	<b>Deskripsi</b>
F-01	Mahasiswa dan Dosen	Melakukan autentikasi
F-02	Mahasiswa dan Dosen	Melihat hasil pembagian kelompok
F-03	Mahasiswa dan Dosen	Melihat persebaran <i>MBTI</i> dan <i>skill</i> mahasiswa
F-04	Mahasiswa dan Dosen	Melihat penjelasan <i>MBTI</i>
F-05	Mahasiswa dan Dosen	Mengganti bahasa
F-06	Dosen	Membuat <i>class</i>
F-07	Dosen	Membuat <i>task</i>
F-08	Dosen	Membuat kuesioner
F-09	Dosen	Melihat riwayat pertanyaan kuesioner
F-10	Dosen	Membagikan tautan kuesioner
F-11	Dosen	Melihat daftar mahasiswa
F-12	Dosen	Melihat jawaban kuesioner mahasiswa
F-13	Dosen	Melakukan pembagian kelompok
F-14	Dosen	Melihat kualitas kelompok
F-16	Dosen	<i>Import</i> hasil pembagian kelompok
F-17	Mahasiswa	Mengerjakan kuesioner
F-18	Mahasiswa	Melihat hasil kuesioner sendiri
F-19	Mahasiswa	Melihat riwayat pembagian kelompok

### 3.4.1.2 Kebutuhan Non-Fungsional

Menentukan kebutuhan yang berfokus pada properti perilaku yang dimiliki oleh sistem. Analisis kebutuhan non-fungsional dilakukan untuk menentukan spesifikasi kebutuhan sistem. Berikut ini adalah tabel 3.2 kebutuhan non-fungsional:

Tabel 3.2 Kebutuhan Non-Fungsional

Kode	Parameter	Deskripsi
NF-01	<i>Reliability</i>	Sistem berjalan dengan semestinya dan sesuai dengan tujuan di rancangan awal
NF-02	<i>Portability</i>	Sistem dapat dijalankan diberbagai <i>web browser</i> , seperti Google Chrome, Microsoft Edge, dan lainnya
NF-03	<i>Responsiveness</i>	Sistem dapat memuat halaman dan hasil pembagian dengan cepat
NF-04	<i>Supportability</i>	Sistem dapat menggunakan Bahasa Indonesia dan Bahasa Inggris
NF-05	<i>Security</i>	Sistem dapat digunakan oleh pengguna yang telah terotentikasi

### 3.4.1.3 Backlog Awal

Pada tahap perencanaan, seluruh kebutuhan fungsional dan teknis yang telah diidentifikasi dirumuskan menjadi daftar *backlog* awal, yang berfungsi sebagai dasar proses pengembangan menggunakan metode *Agile Kanban*. *Backlog* ini memuat daftar fitur dan tugas utama yang akan direalisasikan selama siklus pengembangan, baik dari sisi pengguna (dosen dan mahasiswa) maupun sisi teknis (inisialisasi proyek, integrasi API, pengujian, dan *deployment*).

Setiap item *backlog* didefinisikan berdasarkan fungsi sistem, peran pengguna, serta tanggung jawab teknis pengembang, dan diurutkan berdasarkan

tingkat prioritas terhadap keberhasilan sistem. Fitur seperti *login*, pembentukan kelompok otomatis, dan pengerjaan kuesioner memiliki prioritas tinggi karena menjadi inti dari sistem, sedangkan fitur tambahan seperti visualisasi persebaran MBTI atau penggantian bahasa diletakkan pada prioritas rendah karena tidak berdampak langsung terhadap kelayakan fungsi utama.

*Backlog* ini juga mencakup aktivitas teknis non-fungsional, seperti inisialisasi proyek *backend* dan *frontend*, pengaturan struktur *database*, serta integrasi API Edu2Com. Hal ini mencerminkan kenyataan bahwa dalam penelitian ini, peneliti berperan sebagai pengembang tunggal yang bertanggung jawab terhadap keseluruhan aspek teknis sistem.

Setiap item *backlog* akan dipetakan ke dalam papan *Kanban* dengan status awal berada di kolom *To Do*, dan akan berpindah ke kolom *Doing*, *Testing*, dan *Done* seiring progres pengembangan. Dengan demikian, *backlog* awal ini berfungsi sebagai acuan visual sekaligus operasional dalam menjalankan tahapan *Agile Kanban* secara sistematis.

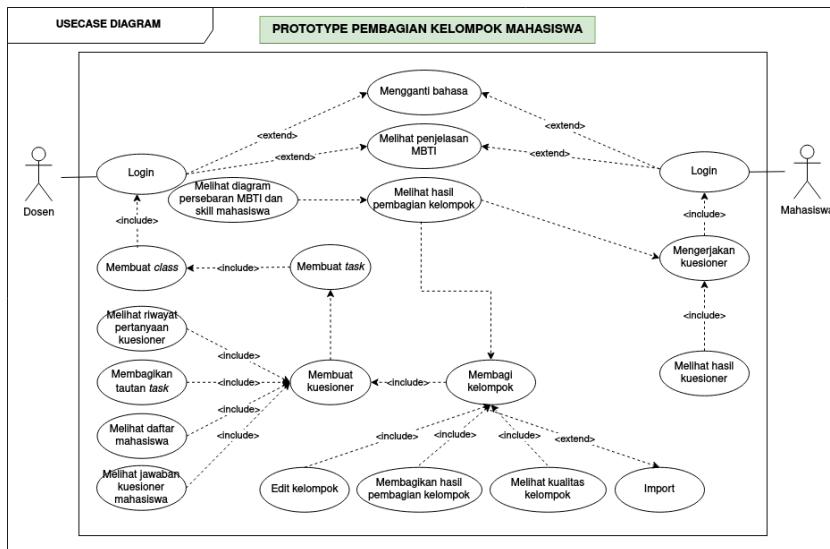
Tabel 3.3 Tabel Backlog Awal

No	Tugas / Fitur	Prioritas
1	Inisialisasi <i>frontend</i>	SANGAT TINGGI
2	Inisialisasi <i>backend</i>	SANGAT TINGGI
3	<i>Setup Database</i> dan <i>ERD</i>	SANGAT TINGGI
4	Integrasi API <i>Edu2Com</i>	SANGAT TINGGI
5	Autentikasi <i>login</i> dan Manajemen <i>Sesi</i>	TINGGI
6	<i>Logout</i> dari sistem	TINGGI

No	Tugas / Fitur	Prioritas
7	Pemilihan bahasa (Inggris dan Indonesia)	RENDAH
8	Membuat <i>class</i> dan Mengelola <i>classroom</i>	TINGGI
9	Membuat <i>task</i>	TINGGI
10	Melihat Daftar Mahasiswa	SEDANG
11	Membuat kuesioner kepribadian, <i>skill</i> , preferensi	TINGGI
12	Mengerjakan kuesioner oleh mahasiswa	TINGGI
13	Menyimpan hasil kuesioner ke <i>database</i>	TINGGI
14	Melihat hasil kuesioner	TINGGI
15	Membagikan tautan kuesioner	TINGGI
16	Proses pembentukan kelompok otomatis melalui <i>API</i>	SANGAT TINGGI
17	Validasi respon <i>API</i> & <i>mapping</i> data ke <i>Database</i>	TINGGI
18	Melihat hasil pembagian kelompok	TINGGI
19	Melihat riwayat pertanyaan kuesioner	TINGGI
20	Melihat kualitas kelompok	SEDANG
21	<i>Import</i> hasil pembagian kelompok	SEDANG
22	Dosen dapat <i>edit</i> anggota kelompok ( <i>manual override</i> )	SEDANG
23	Melihat persebaran <i>MBTI</i> dan <i>skill</i> mahasiswa ( <i>diagram</i> )	RENDAH
24	Melihat penjelasan <i>MBTI</i>	SEDANG
25	Pengujian <i>Grey Box</i> terhadap fitur utama	TINGGI
26	<i>Deployment</i> aplikasi ke <i>server</i>	SEDANG

### 3.4.1.4 Use Case Diagram

*Use case diagram* merupakan suatu diagram yang menggambarkan fungsi dasar pada sistem dan juga sebagai alat yang menggambarkan interaksi sistem dengan lingkungannya. Diagram ini menjelaskan aktivitas apa saja yang dapat dilakukan oleh setiap aktor pada sistem. Berikut *use case diagram* yang digunakan pada penelitian, yang dijelaskan pada gambar 3.2 berikut:



Gambar 3.2 Use Case Diagram Sistem Pembagian Kelompok

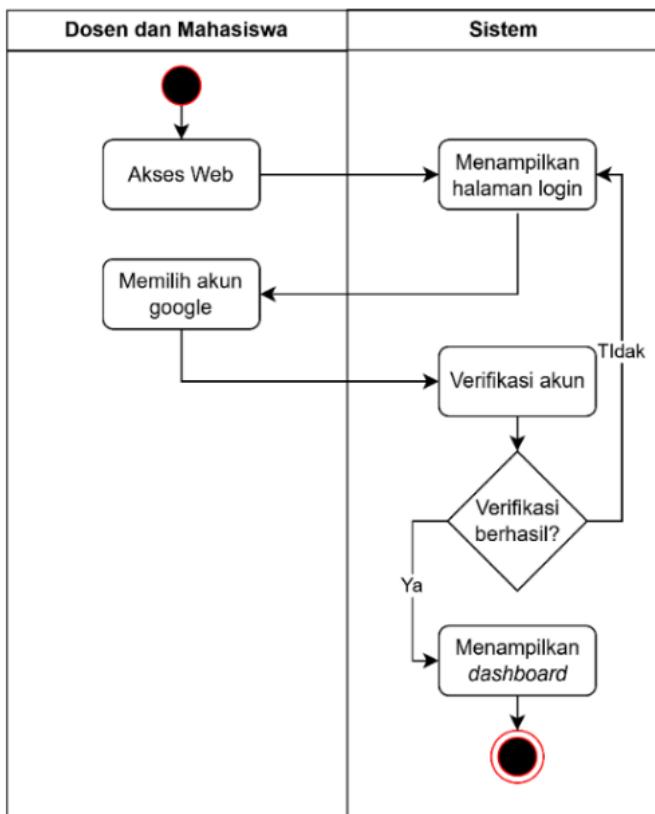
### 3.4.1.5 Activity Diagram

*Activity diagram* merupakan teknik permodelan sistem yang digunakan untuk mengilustrasikan alur ataupun kegiatan utama dan hubungan diantara kegiatan yang terjadi dengan sistem. Adapun *activity diagram* pada penelitian ini, yaitu sebagai berikut:

#### a. Activity Diagram Melakukan Autentifikasi

Gambar 3.3 menampilkan proses *login* dosen dan mahasiswa ke dalam sistem menggunakan akun Google. Proses dimulai ketika pengguna mengakses

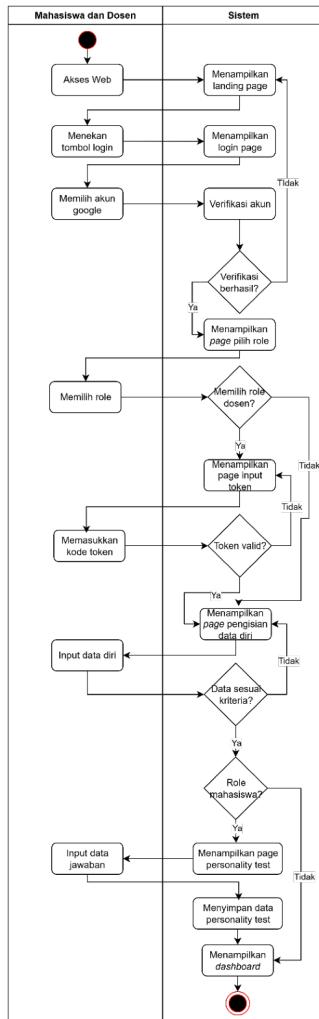
*web*, lalu sistem menampilkan halaman awal (*landing page*). Pengguna kemudian menekan tombol *login*, dan sistem akan menampilkan halaman *login*. Setelah pengguna memilih akun Google, sistem melakukan proses verifikasi akun. Jika verifikasi gagal, proses *login* dihentikan. Namun, jika verifikasi berhasil, sistem akan menampilkan *dashboard* sesuai peran pengguna.



Gambar 3.3 *Activity Diagram Login*

Gambar 3.4 menampilkan alur proses *login* dan pengisian data awal oleh pengguna yang terdiri dari mahasiswa dan dosen. Proses dimulai ketika pengguna mengakses *web* dan sistem menampilkan halaman awal, kemudian dilanjutkan dengan menekan tombol *login* dan memilih akun Google. Setelah sistem

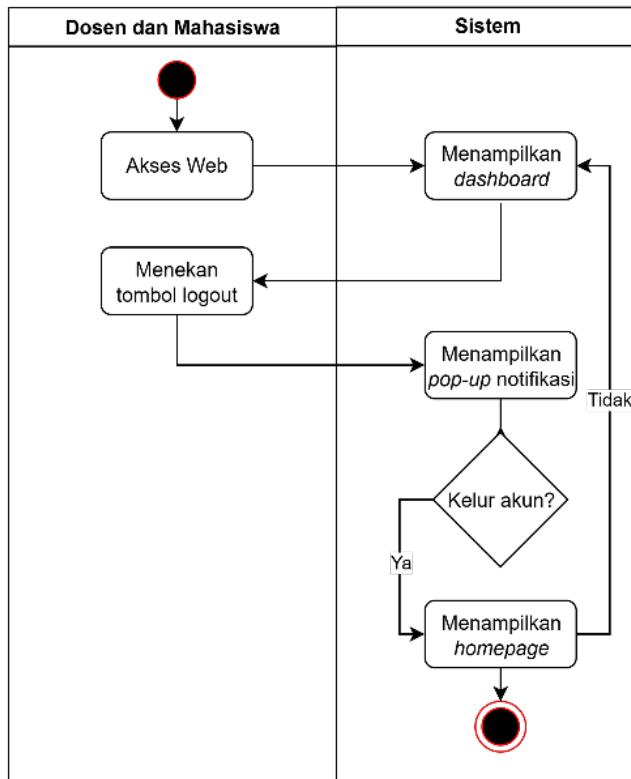
memverifikasi akun, pengguna akan diarahkan ke halaman pemilihan *role*. Jika pengguna memilih *role* sebagai dosen, maka sistem meminta *input* kode *token* untuk validasi. Jika *token* valid, dosen diminta mengisi data diri, dan apabila data sesuai kriteria, proses dilanjutkan ke halaman *dashboard*. Sebaliknya, jika pengguna memilih *role* sebagai mahasiswa, maka setelah mengisi data diri yang sesuai kriteria, sistem menampilkan halaman tes kepribadian untuk diisi. Setelah data tes kepribadian disimpan, sistem akan mengarahkan mahasiswa ke halaman *dashboard*.



Gambar 3.4 Activity Diagram Registrasi Akun

Gambar 3.5 menampilkan proses *logout* dari sistem yang dilakukan oleh dosen maupun mahasiswa. Proses dimulai ketika pengguna telah berada pada tampilan *dashboard* dan memilih untuk keluar dari akun dengan menekan tombol *logout*. Sistem kemudian menampilkan *pop-up* notifikasi untuk meminta konfirmasi apakah pengguna benar-benar ingin keluar. Jika pengguna

membatalkan, maka proses berhenti dan tetap berada di *dashboard*. Namun, jika pengguna menyetujui untuk keluar, maka sistem akan mengarahkan kembali ke halaman awal (*landing page*) sebagai akhir dari proses *logout*.

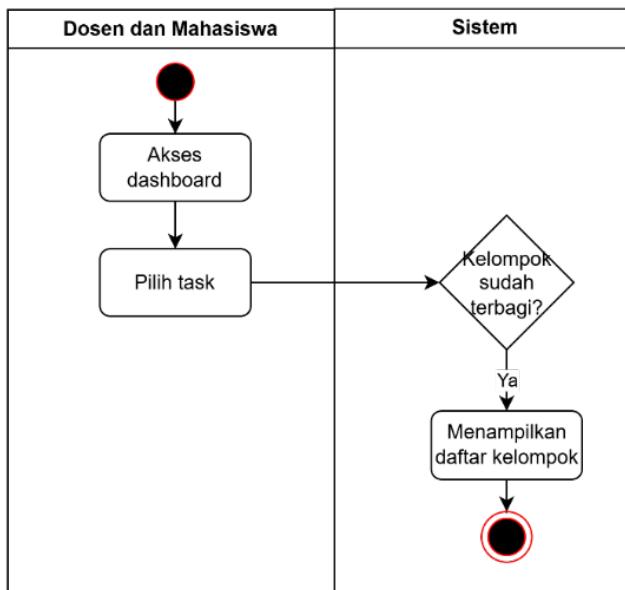


Gambar 3.5 Activity Diagram Logout

### b. Activity Diagram Melihat Hasil Pembagian Kelompok

Gambar 3.6 menampilkan proses yang dilakukan oleh dosen atau mahasiswa untuk melihat hasil pembagian kelompok. Proses dimulai ketika pengguna mengakses halaman *dashboard* dan memilih *task* atau menu yang berkaitan dengan informasi pembagian kelompok. Sistem kemudian akan memeriksa apakah proses pembagian kelompok telah dilakukan. Jika kelompok

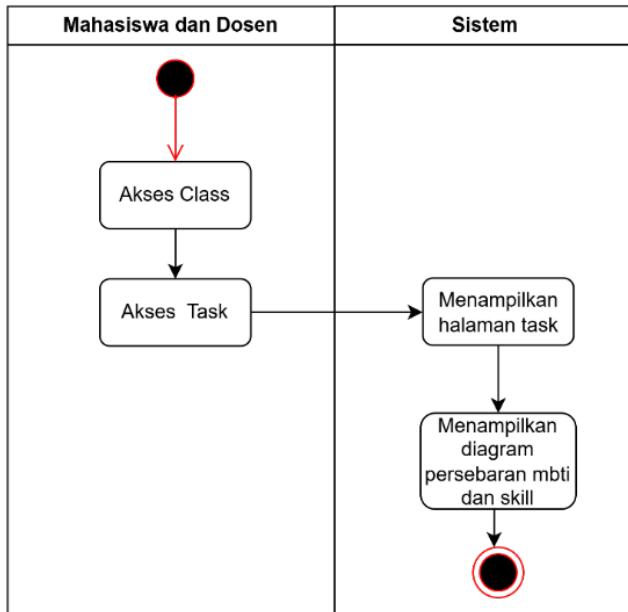
belum terbagi, maka sistem tidak menampilkan apa pun. Namun jika pembagian kelompok sudah dilakukan, maka sistem akan menampilkan daftar kelompok yang telah terbentuk kepada pengguna.



Gambar 3.6 *Activity Diagram* Melihat Hasil Pembagian Kelompok

### c. Activity Diagram Melihat Persebaran MBTI, Keahlian, Preferensi, dan Jenis Kelamin Mahasiswa

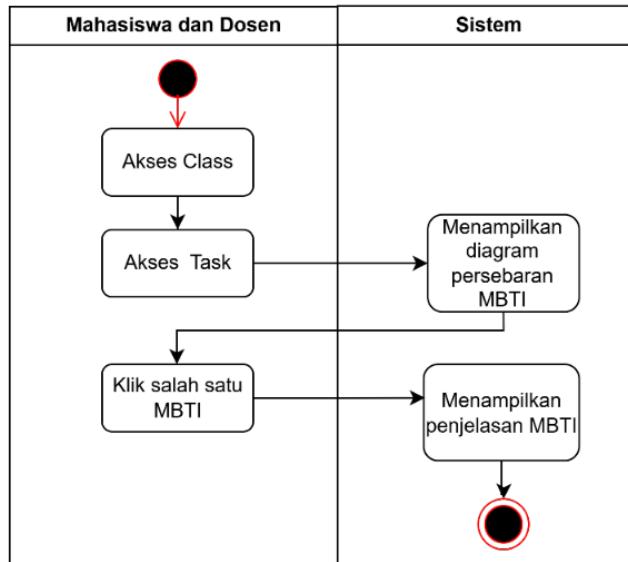
Gambar 3.7 menampilkan proses yang dilakukan oleh dosen dan mahasiswa untuk melihat persebaran MBTI dan keterampilan dalam suatu kelas. Proses dimulai saat pengguna mengakses kelas, kemudian memilih *task* atau tugas yang berkaitan. Setelah itu, sistem menampilkan halaman *task* yang memuat informasi terkait. Pada halaman tersebut, sistem secara otomatis menampilkan diagram visual yang menunjukkan persebaran tipe kepribadian MBTI, keterampilan, preferensi dan jenis kelamin dari seluruh anggota kelas, yang dapat digunakan sebagai dasar untuk analisis atau pembentukan kelompok.



Gambar 3.7 *Activity Diagram* Melihat Persebaran MBTI, Keahlian, Preferensi, dan Jenis Kelamin Mahasiswa

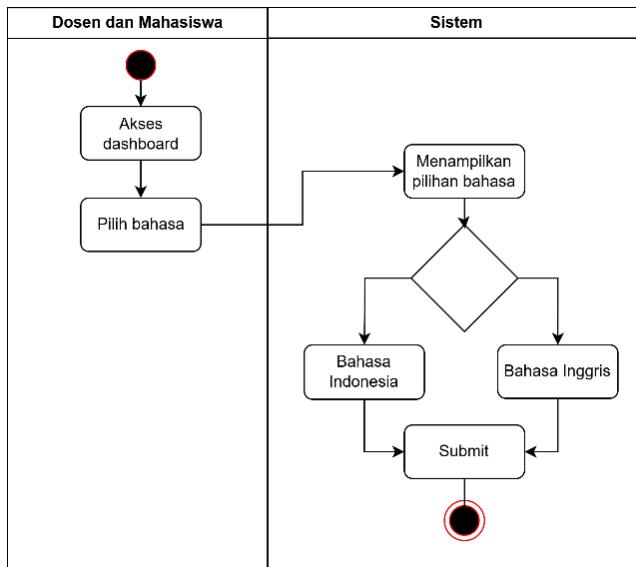
#### d. Activity Diagram Melihat Penjelasan MBTI

Gambar 3.8 menampilkan proses interaksi dosen dan mahasiswa untuk melihat penjelasan dari tipe kepribadian MBTI. Proses dimulai saat pengguna mengakses kelas, kemudian membuka halaman *task* yang berkaitan. Setelah sistem menampilkan diagram persebaran MBTI dari seluruh anggota kelas, pengguna dapat mengklik salah satu tipe MBTI yang ada pada diagram tersebut. Sistem kemudian akan menampilkan penjelasan lengkap mengenai tipe MBTI yang dipilih, sehingga pengguna dapat memahami karakteristik kepribadian tersebut secara lebih mendalam.

Gambar 3.8 *Activity Diagram* Melihat Penjelasan MBTI

#### e. Activity Diagram Mengganti Bahasa

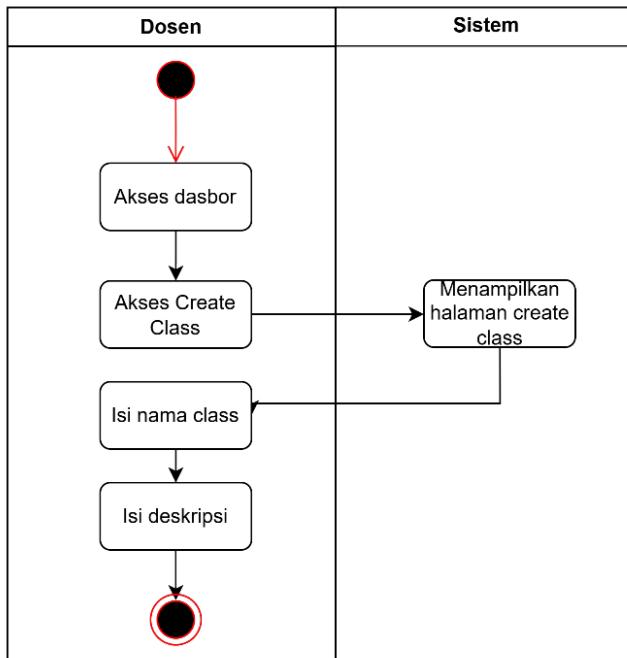
Gambar 3.9 menampilkan proses pemilihan bahasa tampilan sistem oleh dosen dan mahasiswa. Proses dimulai ketika pengguna mengakses *dashboard*, kemudian memilih opsi untuk mengganti bahasa. Sistem akan menampilkan pilihan bahasa yang tersedia, yaitu Bahasa Indonesia dan Bahasa Inggris. Pengguna kemudian memilih salah satu bahasa sesuai preferensinya, dan sistem akan menyimpan pilihan tersebut melalui proses *submit*, sehingga tampilan sistem akan disesuaikan dengan bahasa yang dipilih.



Gambar 3.9 *Activity Diagram* Mengganti Bahasa

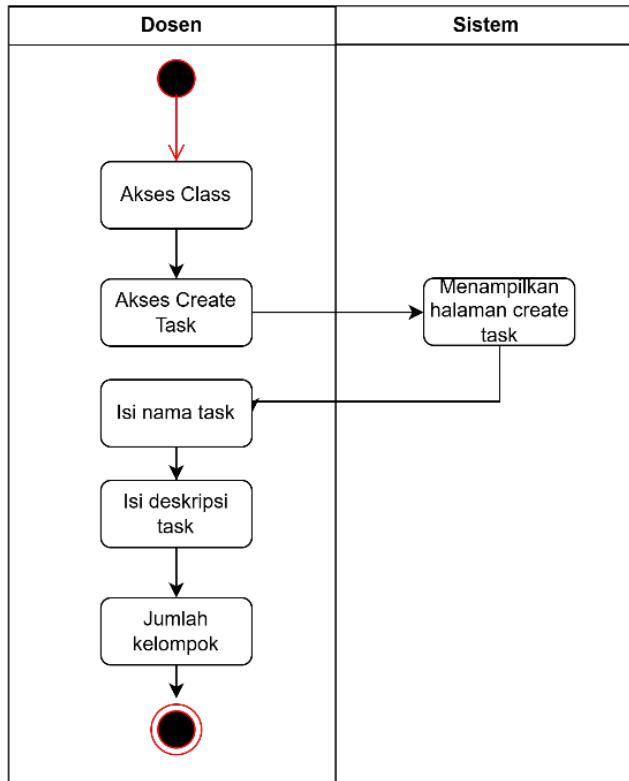
#### f. Activity Diagram Membuat Kelas

Gambar 3.10 menampilkan proses yang dilakukan oleh dosen untuk membuat kelas baru dalam sistem. Proses dimulai ketika dosen mengakses *dashboard*, lalu memilih menu *Create Class*. Setelah itu, sistem akan menampilkan halaman pembuatan kelas. Dosen kemudian mengisi nama kelas dan deskripsi kelas yang ingin dibuat. Proses ini bertujuan untuk mendefinisikan informasi dasar sebelum kelas dapat digunakan untuk aktivitas pembelajaran atau pembentukan kelompok.

Gambar 3.10 *Activity Diagram* Membuat Kelas

#### **g. Activity Diagram Membuat Tugas**

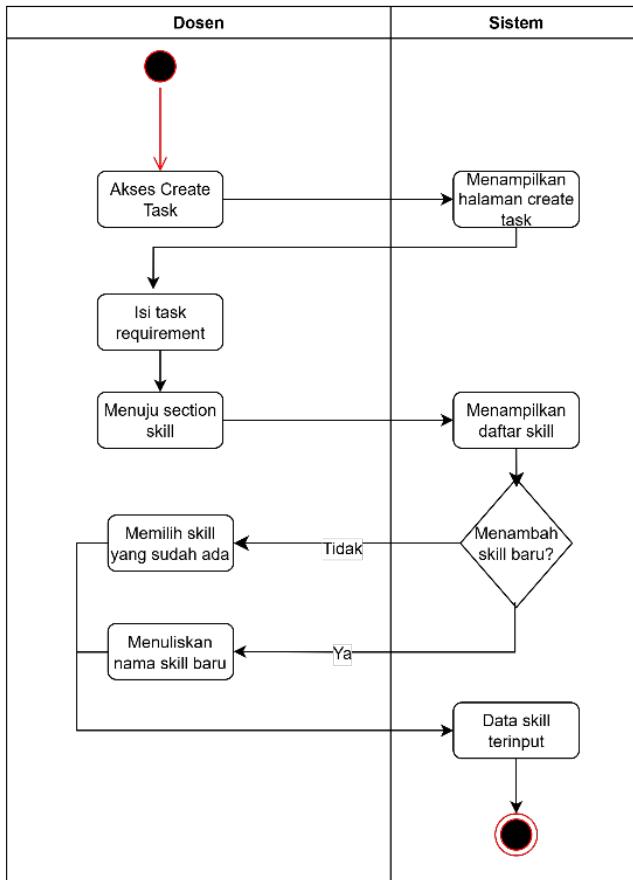
Gambar 3.11 menampilkan proses pembuatan *task* oleh dosen dalam sebuah kelas. Proses dimulai ketika dosen mengakses halaman kelas, lalu memilih opsi *Create Task*. Sistem kemudian menampilkan halaman untuk pembuatan *task*. Dosen mengisi nama *task*, deskripsi *task*, serta jumlah kelompok yang diinginkan. Proses ini bertujuan untuk mendefinisikan tugas yang akan digunakan sebagai dasar dalam pembentukan kelompok atau aktivitas pembelajaran mahasiswa dalam kelas tersebut.

Gambar 3.11 *Activity Diagram* Membuat Tugas

#### **h. Activity Diagram Membuat Kuesioner**

Gambar 3.12 menjelaskan alur kerja saat dosen menambahkan komponen *skill* ke dalam sebuah tugas. Proses ini diawali ketika dosen mengakses fitur pembuatan tugas dan mengisi persyaratan umumnya. Selanjutnya, dosen menuju ke bagian *skill*, di mana sistem akan menampilkan daftar *skill* yang sudah tersedia. Pada tahap ini, dosen dihadapkan pada pilihan untuk menambah *skill* baru: jika dosen memilih untuk tidak menambah *skill* baru, maka ia akan memilih salah satu *skill* dari daftar yang ada. Namun, jika dosen memilih untuk menambah *skill* baru, maka ia akan menuliskan nama untuk *skill* tersebut.

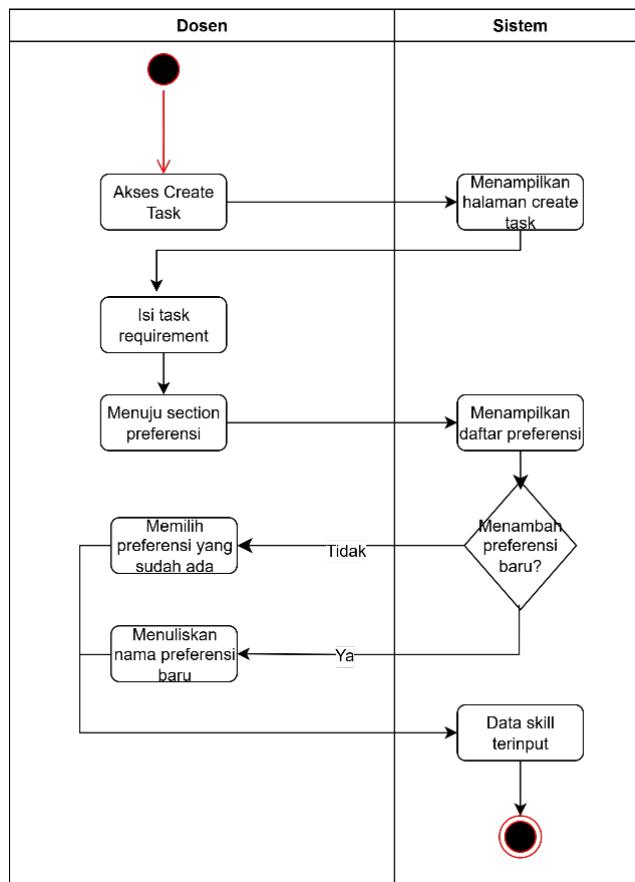
Setelah *skill* dipilih atau dibuat, sistem akan meng-*input* data *skill* tersebut ke dalam tugas, dan alur aktivitas pun selesai.



Gambar 3.12 Activity Diagram Membuat Kuesioner Skill

Gambar 3.13 menampilkan alur kerja saat dosen menambahkan komponen preferensi pada sebuah tugas. Proses ini diawali saat dosen mengakses fitur pembuatan tugas dan mengisi persyaratan dasarnya. Setelah itu, dosen menuju ke bagian (*section*) preferensi, dan sistem akan menampilkan daftar preferensi yang sudah ada. Dosen kemudian dihadapkan pada pilihan: jika tidak ingin

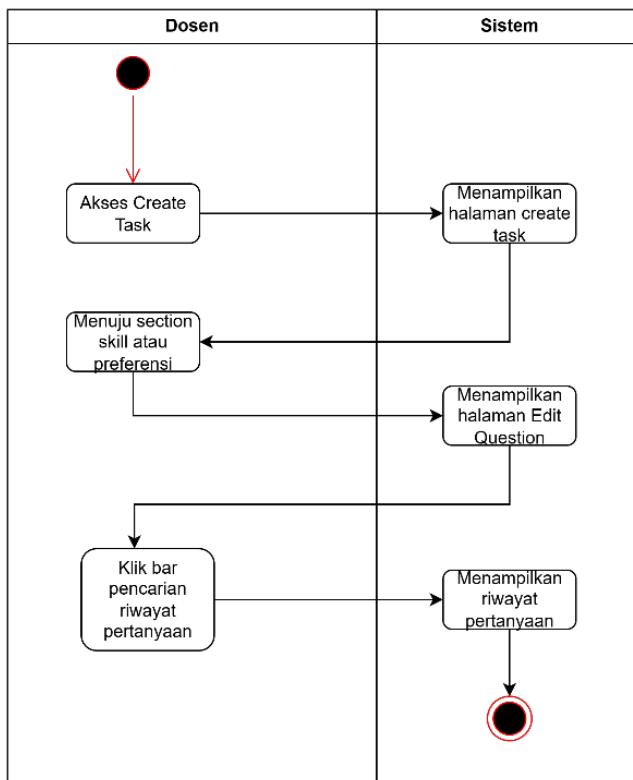
menambah preferensi baru, ia akan memilih salah satu preferensi yang tersedia dari daftar. Sebaliknya, jika ingin membuat preferensi baru, ia akan menuliskan nama untuk preferensi baru tersebut. Setelah preferensi dipilih atau dibuat, sistem akan menyimpan data yang *di-input* dan proses penambahan preferensi ini pun selesai.



Gambar 3.13 *Activity Diagram* Membuat Kuesioner Preferensi

### i. Activity Diagram Melihat Riwayat Pertanyaan Kuesioner

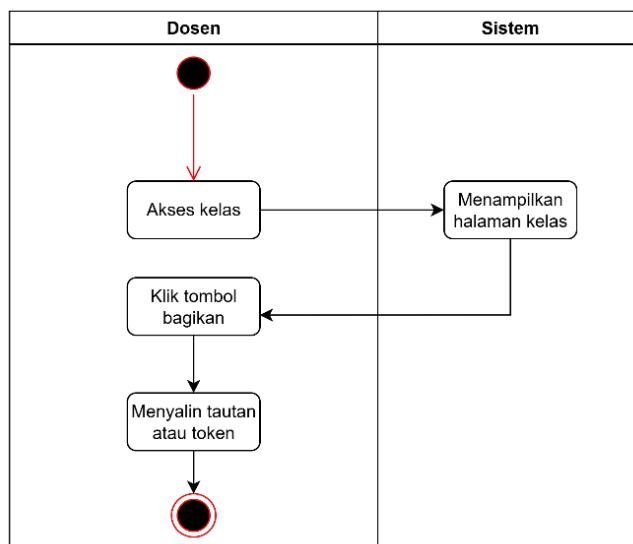
Gambar 3.14 menampilkan alur kerja dosen untuk melihat riwayat pertanyaan yang terkait dengan *skill* atau preferensi. Proses ini dimulai saat dosen mengakses halaman tugas, kemudian langsung menuju ke bagian (*section*) *skill* atau preferensi. Sistem akan merespons dengan menampilkan halaman *Edit Question*. Dari halaman tersebut, dosen mengklik bar pencarian untuk melihat riwayat pertanyaan. Setelah itu, sistem akan menampilkan riwayat pertanyaan yang tersimpan, dan proses untuk melihat riwayat ini pun berakhir.



Gambar 3.14 *Activity Diagram Melihat Riwayat Pertanyaan Kuesioner*

### j. Activity Diagram Membagikan Tautan Kuesioner

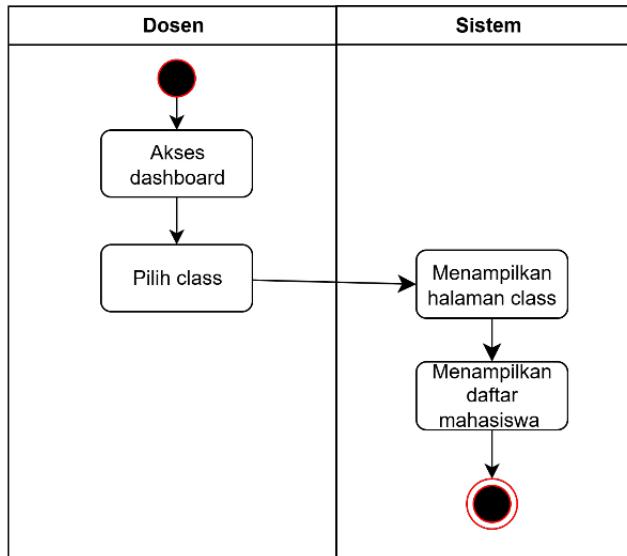
Gambar 3.15 menampilkan proses dosen saat akan membagikan kelas kepada mahasiswa. Alur dimulai ketika dosen mengakses halaman kelas yang ingin dibagikan, lalu sistem akan menampilkan detail dari halaman kelas tersebut. Pada halaman itu, dosen menekan tombol "bagikan". Setelahnya, dosen menyalin tautan atau *token* yang disediakan oleh sistem untuk dibagikan, dan proses untuk mendapatkan tautan kelas ini pun selesai.



Gambar 3.15 *Activity Diagram* Membagikan Tautan Kuesioner

### k. Activity Diagram Melihat Daftar Mahasiswa

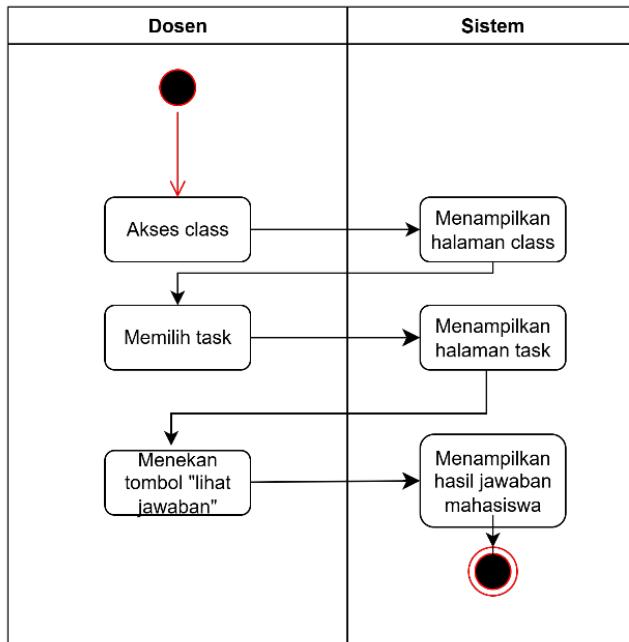
Gambar 3.16 menjelaskan alur kerja dosen untuk melihat daftar mahasiswa yang terkait dengan sebuah kelas. Proses dimulai saat dosen mengakses halaman *dashboard*, kemudian ia memilih salah satu kelas yang ingin dilihat. Setelah itu, tampilan akan beralih ke halaman kelas dan daftar mahasiswa juga akan ditampilkan di halaman tersebut.



Gambar 3.16 *Activity Diagram* Melihat Daftar Mahasiswa

### I. Activity Diagram Melihat Jawaban Kuesioner Mahasiswa

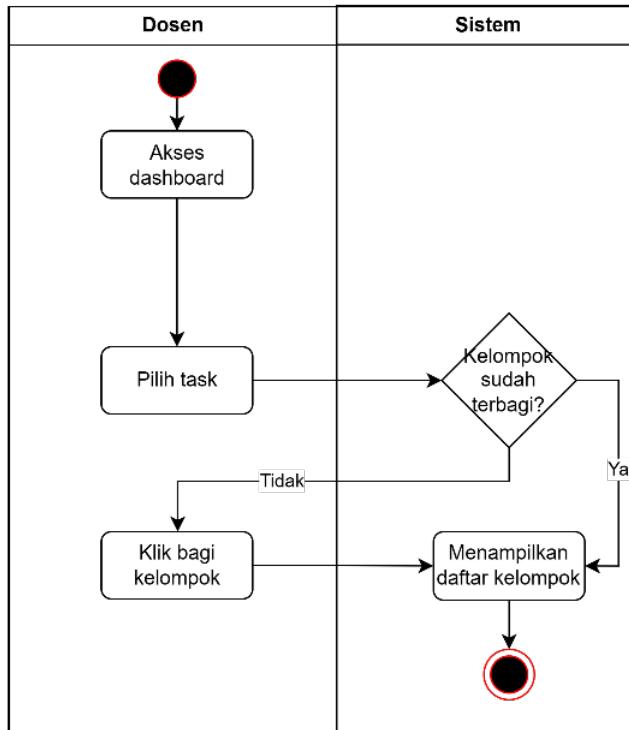
Gambar 3.17 menggambarkan alur kerja dosen ketika akan melihat jawaban mahasiswa pada suatu tugas. Proses diawali ketika dosen mengakses salah satu kelasnya, dan sistem menampilkan halaman kelas tersebut. Dari halaman kelas, dosen kemudian memilih salah satu tugas (*task*) yang ingin diperiksa, lalu sistem akan menampilkan halaman detail tugas itu. Selanjutnya, dosen menekan tombol "lihat jawaban", dan sistem akan merespons dengan menampilkan hasil jawaban dari mahasiswa untuk tugas yang dipilih, kemudian proses pun selesai.



Gambar 3.17 Activity Diagram Melihat Jawaban Kuesioner Mahasiswa

### **m. Activity Diagram Melakukan Pembagian Kelompok**

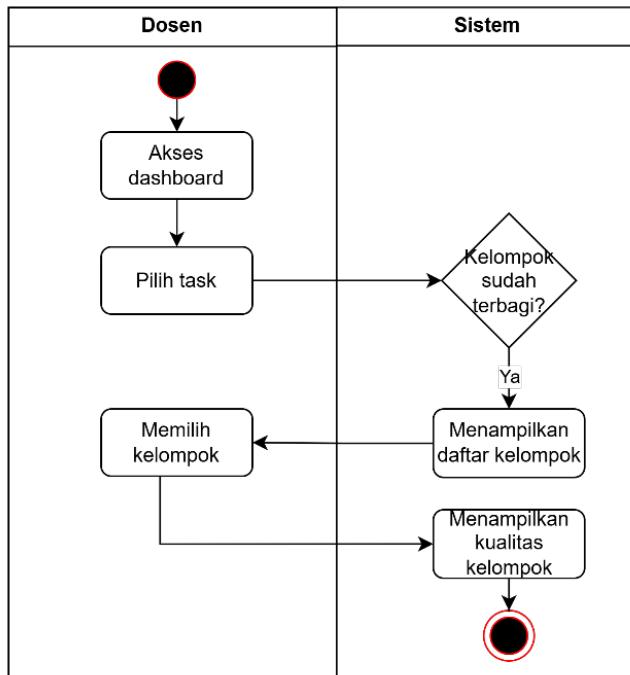
Gambar 3.18 menjelaskan alur kerja dosen dalam mengelola kelompok mahasiswa untuk suatu tugas. Proses dimulai saat dosen mengakses *dashboard* dan memilih tugas yang diinginkan. Setelah itu, sistem akan memeriksa apakah kelompok untuk tugas tersebut sudah terbagi atau belum. Jika kelompok sudah ada, sistem akan langsung menampilkan daftar kelompok mahasiswa. Namun, jika kelompok belum terbagi, dosen akan melakukan aksi dengan menekan tombol ”bagi kelompok” untuk memulai proses pembagian. Alur kerja ini berakhir setelah kelompok ditampilkan atau setelah dosen memulai aksi untuk membagi kelompok.



Gambar 3.18 *Activity Diagram* Melakukan Pembagian Kelompok

#### n. Activity Diagram Melihat Kualitas Kelompok

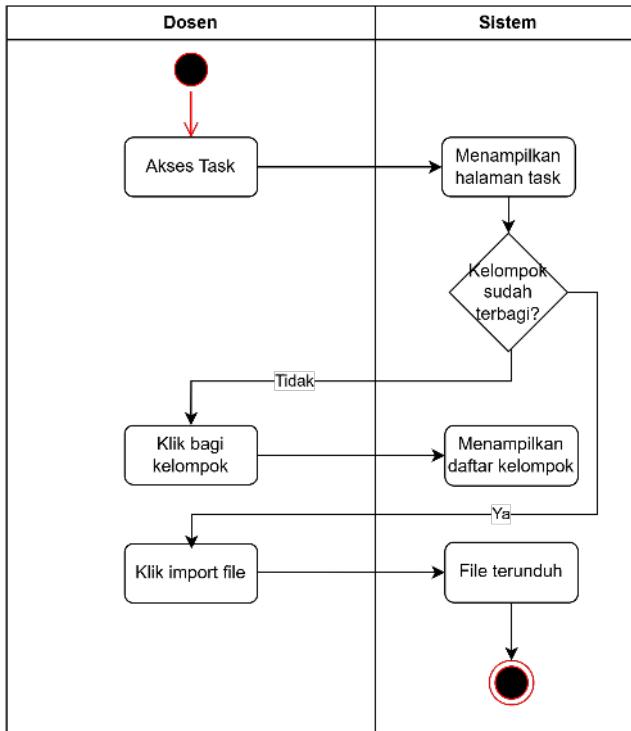
Gambar 3.19 menunjukkan alur kerja dosen untuk melihat kualitas sebuah kelompok pada tugas tertentu. Proses diawali saat dosen mengakses *dashboard* dan memilih salah satu tugas. Sistem kemudian akan memeriksa apakah kelompok pada tugas tersebut sudah terbagi; jika ya, sistem akan menampilkan daftar kelompok yang ada. Selanjutnya, dosen memilih salah satu kelompok dari daftar tersebut. Setelah kelompok dipilih, sistem akan merespons dengan menampilkan detail mengenai kualitas dari kelompok tersebut, dan alur kerja untuk melihat kualitas kelompok ini pun selesai.



Gambar 3.19 *Activity Diagram* Melihat Kualitas Kelompok

#### p. Activity Diagram Import Hasil Pembagian Kelompok

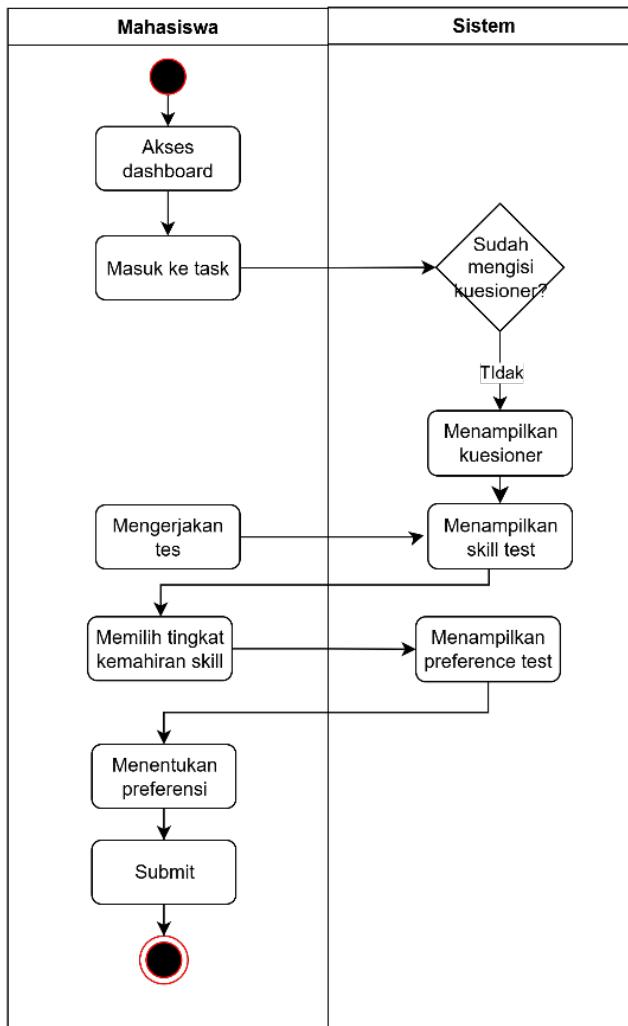
Gambar 3.20 menjelaskan dua alur kerja yang berbeda terkait pengelolaan kelompok dalam sebuah tugas, tergantung pada status pembagian kelompok. Proses diawali saat dosen mengakses halaman tugas, kemudian sistem akan memeriksa apakah kelompok sudah terbagi. Jika kelompok belum ada (alur "Tidak"), dosen dapat memulai proses pembagian dengan menekan tombol "bagi kelompok", lalu memilih untuk meng-*import* berkas guna membuat kelompok baru. Sebaliknya, jika kelompok sudah ada (alur "Ya"), sistem akan menampilkan daftar kelompok dan menyediakan berkas berisi data kelompok tersebut untuk diunduh. Kedua alur kerja tersebut berakhir setelah aksi masing-masing selesai.



Gambar 3.20 *Activity Diagram Import Hasil Pembagian Kelompok*

#### **q. Activity Diagram Mengerjakan Kuesioner**

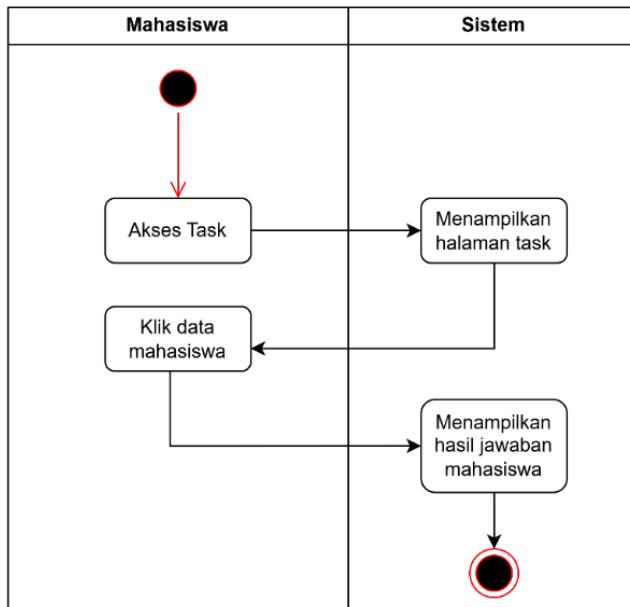
Gambar 3.21 menjelaskan alur kerja yang dialami mahasiswa saat perlu mengisi kuesioner awal pada sebuah tugas. Proses dimulai ketika mahasiswa mengakses *dashboard* dan masuk ke halaman tugas (*task*). Sistem akan memeriksa apakah kuesioner sudah diisi sebelumnya. Jika belum, sistem akan menampilkan kuesioner yang harus dikerjakan, dimulai dengan menampilkan tes *skill*. Mahasiswa kemudian mengerjakan tes tersebut dengan memilih tingkat kemahiran *skill*-nya. Setelah itu, sistem menampilkan tes preferensi, dan mahasiswa menentukan preferensinya. Setelah semua bagian diisi, mahasiswa menekan tombol *submit* untuk mengirimkan seluruh data, dan alur proses pun berakhir.

Gambar 3.21 *Activity Diagram* Mengerjakan Kuesioner

#### r. Activity Diagram Melihat Hasil Kuesioner Sendiri

Gambar 3.22 menjelaskan alur kerja seorang mahasiswa ketika akan melihat hasil jawabannya sendiri pada sebuah tugas. Proses ini dimulai saat mahasiswa mengakses halaman tugas yang telah dikerjakan, dan sistem akan

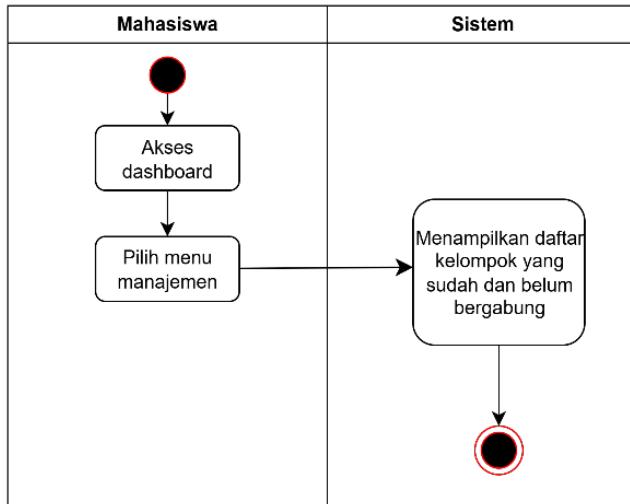
menampilkan halaman tersebut. Selanjutnya, mahasiswa menekan tombol atau menu "data mahasiswa" untuk melihat detail pengerjaannya. Sistem kemudian akan merespons dengan menampilkan hasil jawaban yang telah dikirim oleh mahasiswa tersebut, dan alur untuk melihat jawaban ini pun selesai.



Gambar 3.22 *Activity Diagram* Melihat Hasil Kuesioner Sendiri

#### **s. Activity Diagram Melihat Riwayat Pembagian Kelompok**

Gambar 3.23 menjelaskan alur kerja seorang mahasiswa untuk melihat status keanggotaan kelompoknya. Proses dimulai ketika mahasiswa mengakses halaman *dashboard* utama, kemudian memilih menu manajemen. Sebagai respons, sistem akan menampilkan sebuah daftar yang berisi informasi kelompok, yang dibedakan antara kelompok di mana mahasiswa tersebut sudah bergabung dan kelompok yang belum ia masuki. Alur untuk melihat daftar status kelompok ini pun selesai.



Gambar 3.23 *Activity Diagram* Melihat Riwayat Pembagian Kelompok

#### 3.4.1.6 Entity Relationship Diagram (ERD)

Gambar ERD di bawah ini memiliki beberapa entitas yang digunakan untuk mengelola anggota tim. Hubungan antar entitas dijelaskan melalui proses seperti pembagian tugas, pengumpulan data mahasiswa, dan pembentukan tim menggunakan Edu2Com API.

##### 1. Entitas User

Entitas ini menyimpan data pengguna, baik dosen maupun mahasiswa. Atribut yang dimiliki adalah:

- ***id***: Primary Key untuk mengidentifikasi pengguna.
- ***email***: Alamat *email* pengguna.
- ***password***: *Hash password* untuk otentikasi.
- ***gender***: *Gender* pengguna (Pria/Wanita).
- ***role\_id***: *foreign key* untuk entitas *Role*.

## 2. Entitas *Role*

Mendefinisikan peran: LECTURER atau STUDENT

## 3. *LecturerProfile* dan *StudentProfile*

Sistem menggunakan dua entitas profil terpisah:

- *LecturerProfile*: profil khusus untuk dosen
- *StudentProfile*: profil khusus untuk mahasiswa yang menyimpan data MBTI (ei, sn, tf, pj) dan status penyelesaian tes kepribadian

## 4. Entitas *Classroom*

Entitas ini merepresentasikan kelas virtual yang dikelola oleh dosen.

Atribut yang dimiliki adalah:

- ***id***: *primary key* untuk mengidentifikasi kelas
- ***name***: nama kelas
- ***code***: kode unik untuk mahasiswa bergabung
- ***lecturer\_id***: *foreign key* yang menghubungkan ke dosen pemilik
- ***is\_closed***: status apakah kelas masih aktif

## 5. Entitas *ClassroomTask*

Entitas ini menyimpan tugas-tugas yang dibuat dosen untuk pembentukan tim. Atribut yang dimiliki adalah:

- ***id***: *primary key* untuk mengidentifikasi tugas
- ***title***: judul tugas
- ***classroom\_id***: *foreign key* yang menghubungkan ke kelas tertentu
- ***team\_size***: jumlah anggota dalam satu tim (diperlukan oleh API Edu2Com)

## 6. Entitas *MBTI*

Kuisisioner MBTI untuk pengguna dengan *role Student*:

- ***question***: pertanyaan individual dalam kuesioner

- **options:** jawaban mahasiswa terhadap pertanyaan

### **7. Entitas *TaskPreference***

Entitas ini menyimpan preferensi mahasiswa terhadap tugas tertentu dengan skala 0-1.

### **8. Entitas *TeamFormationRequest***

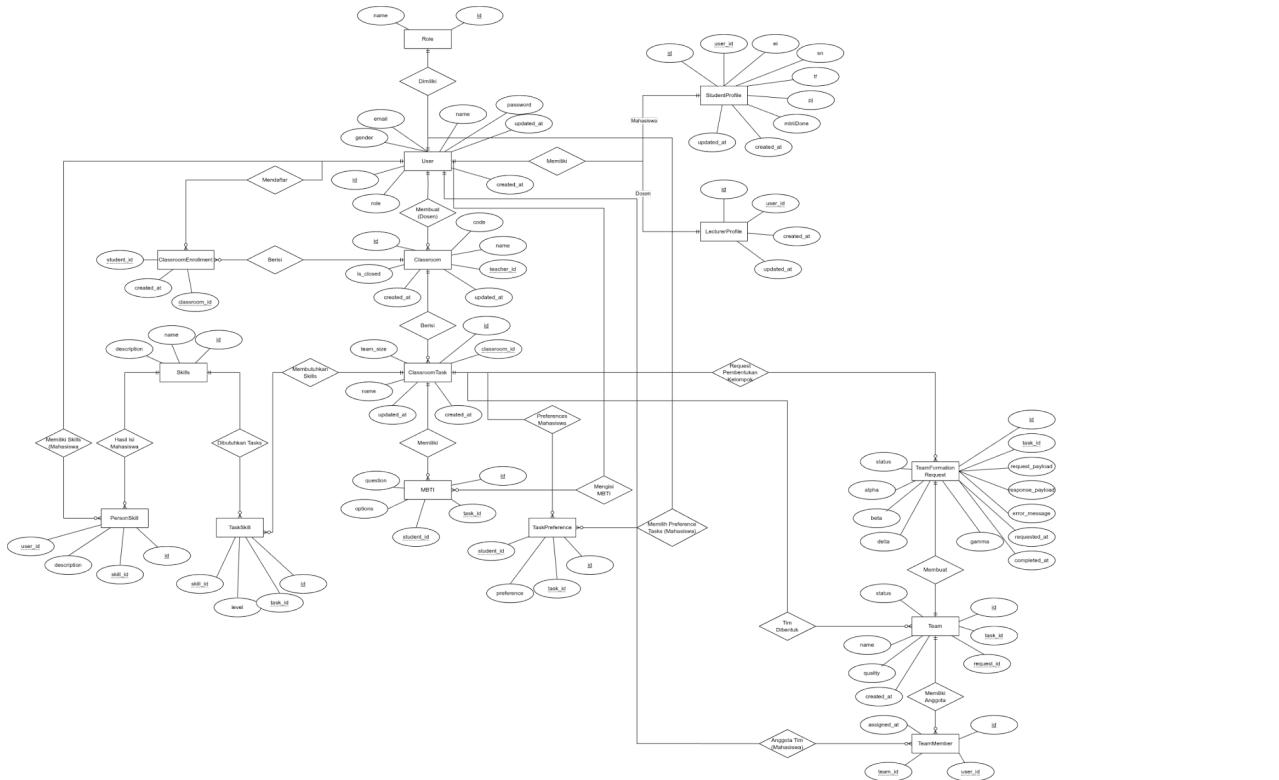
Entitas ini mencatat setiap permintaan pembentukan tim ke API Edu2Com. Atribut yang dimiliki adalah:

- **task\_id:** tugas yang akan dibentuk timnya
- **status:** status pembentukan (PENDING, PROCESSING, COMPLETED, FAILED)
- **alpha, beta, gamma, delta:** parameter pembentukan tim
- **request\_payload & response\_payload:** data JSON yang dikirim dan diterima dari API

### **9. Entitas *Team & TeamMember***

Entitas ini menyimpan hasil pembentukan tim:

- **team:** tim yang terbentuk dengan skor kualitas dari Edu2Com
- **team\_member:** anggota-anggota dalam sebuah tim



Gambar 3.24 Entity Relationship Diagram (ERD)

### 3.4.1.7 Tahap Pengembangan

Pada tahap pengembangan, pekerjaan mengikuti alur Kanban dengan empat kolom: *To Do*, *Doing*, *Testing* dan *Done*. Pengembang menarik (*pull*) tugas dari *To Do* ke *Doing* saat siap mengerjakannya, mengikuti prinsip *pull system* untuk menjaga ritme kerja sesuai kapasitas, bukan dorongan eksternal. Tugas mencakup pembuatan antarmuka web, logika aplikasi, integrasi API Edu2Com berbasis AI, serta pengujian. Bila ada hambatan (integrasi gagal, fitur tidak jelas), pengembang dapat mengalihkan fokus sementara ke *To Do* lain, atau menyelesaikan masalah lewat riset/peninjauan ulang.

Visualisasi tugas lewat papan Kanban memberi transparansi progres secara *real-time*. Bahkan dalam tim kecil atau solo, ini membantu pemantauan pencapaian dan sisa pekerjaan. Peneliti seperti Ilmi et al. (2020) menekankan bahwa visualisasi kanban dapat meningkatkan fleksibilitas terhadap perubahan dalam suatu proyek, sejalan dengan pengembangan ini yang setiap perubahan langsung tercermin di *backlog* dan papan kanban.

Pengujian dan *code review* dilakukan paralel dengan pengembangan. Setelah fitur (misal integrasi API) selesai, dibuat kartu uji di kolom *Testing*. Jika ditemukan *bug* (misal format JSON salah, *error* tak tertangani), kartu baru akan dibuat dan ditambahkan ke *backlog* dan diprioritaskan.

### 3.4.1.8 Tahap Pengujian

Tahap pengujian dilaksanakan segera setelah (dan bahkan bersamaan dengan) tahap pengembangan untuk menjamin kualitas setiap fitur yang dihasilkan. Peneliti menerapkan metode *grey box testing*, yaitu kombinasi pengujian *black box* dan *white box*, guna memastikan fungsionalitas sistem sesuai spesifikasi sekaligus memverifikasi integrasi internalnya. Pengujian dilakukan dengan skenario-skenario yang mencakup: validasi respons dan data dari API AI Edu2Com, pengecekan algoritma pengelompokan AI terhadap berbagai kombinasi data mahasiswa, serta uji antarmuka dan alur navigasi pada

*dashboard*. Setiap skenario pengujian dicatat sebagai tugas pada papan Kanban (di kolom *Testing*) agar dapat dimonitor progresnya. Jika suatu skenario uji menemukan kegagalan, langkah penanganannya adalah membuat kartu/item perbaikan di *backlog*. Misalnya, bila sistem gagal menangani kasus *error* dari API, maka diciptakan tugas baru untuk memperbaiki penanganan *error* tersebut. Siklus ini menunjukkan bagaimana *feedback loop* internal diatur dalam Kanban, hasil pengujian langsung menjadi masukan untuk *backlog*, yang kemudian ditangani dalam alur pengembangan selanjutnya.

Elemen dari *grey box testing* seperti *matrix test*, *regression test*, *pattern test* dan *orthogonal array test* akan kita gunakan untuk melakukan rangkaian skenario pengujian yang telah dikelompokkan dalam Tabel 3.4 di bawah ini.

Tabel 3.4 Skenario Pengujian *Grey Box Testing*

No	Komponen	Kode	Skenario	Teknik Pengujian	Hasil yang Diharapkan
1	<i>Login</i>	GB01	Login berhasil menggunakan <i>username/password</i> valid melalui Edu2Com API	<i>Matrix</i>	Pengguna login sukses, sesi aktif, <i>dashboard</i> muncul
		GB02	Menguji login dengan <i>username</i> valid tetapi <i>password</i> salah	<i>Pattern</i>	Login ditolak – sistem menampilkan pesan kesalahan “kredensial tidak valid”; akses tidak diberikan
2	Sesi Pengguna	GB03	Uji keamanan sesi, sesi aktif selama pengguna aktif, otomatis <i>expired</i> setelah <i>timeout</i>	<i>Regression</i>	Sesi dikelola aman, login ulang diperlukan setelah sesi berakhir
3	Input Kuesioner	GB04	Pengisian & penyimpanan kuesioner lengkap (uji simpan data ke <i>database</i> )	<i>Orthogonal Array</i>	Data disimpan lengkap & benar di <i>database</i>

No	Komponen	Kode	Skenario	Teknik Pengujian	Hasil yang Diharapkan
		GB05	Validasi <i>form</i> input kuesioner (kosong, format salah, data tidak valid)	<i>Regression</i>	Sistem menolak input invalid, pesan <i>error</i> jelas. Tidak ada data salah tersimpan
		GB06	Edit kuesioner yang sudah dikirim sebelumnya (uji <i>update database</i> )	<i>Matrix</i>	Data berhasil diperbarui secara konsisten tanpa <i>bug</i> baru
4	Pembentukan Kelompok (API)	GB07	Pengujian <i>end-to-end</i> pembentukan tim dengan data lengkap dan valid (login → kuesioner → formasi tim)	<i>Matrix</i>	API mengembalikan alokasi tim optimal sesuai parameter
		GB08	API menolak parameter permintaan tim yang invalid (ukuran tim tidak logis, data kosong)	<i>Orthogonal Array</i>	Mengembalikan pesan <i>error</i> yang jelas dan informatif
		GB09	Cek respons API dalam batas waktu tertentu dengan kombinasi jumlah permintaan berbeda ( $\leq 5$ detik, 10–100 pengguna bersamaan)	<i>Matrix / Performance Testing</i>	Respons API memenuhi batas waktu yang ditetapkan
5	Keamanan	GB10	Verifikasi <i>role-based access</i> (mahasiswa vs admin) terhadap fitur yang diizinkan & tidak diizinkan	<i>Regression Testing</i>	Akses dikontrol ketat, tidak ada pelanggaran <i>auth</i>
		GB11	Uji input berbahaya pada <i>form</i> ( <i>script injection</i> , CSRF, JSON invalid, <i>upload file</i> berbahaya)	<i>Orthogonal Array</i>	Sistem menolak semua input berbahaya tanpa terkecuali

No	Komponen	Kode	Skenario	Teknik Pengujian	Hasil yang Diharapkan
6	Ketahanan Sistem	GB12	Simulasi <i>error API eksternal</i> (Edu2Com API <i>down</i> , <i>timeout</i> , gagal respons)	<i>Regression</i>	Aplikasi tidak <i>crash</i> , menangani <i>error eksternal</i> dengan stabil, pesan <i>error</i> jelas
7	Kejelasan <i>Error</i>	GB13	Kejelasan pesan <i>error</i> di seluruh fitur (login, input <i>form</i> , API respons gagal)	<i>Pattern</i>	Pesan <i>error</i> informatif & mudah dibaca pengguna
8	Audit dan <i>Logging</i>	GB14	Pengujian pencatatan log untuk semua aksi penting (login/logout, <i>submit form</i> , pembentukan tim)	<i>Integration Regression</i>	Log audit tercatat lengkap, akurat, dan bisa ditelusuri
9	<i>Logout</i>	GB15	Verifikasi <i>logout</i> membersihkan sesi dan memastikan tidak ada akses lagi setelah <i>logout</i>	<i>Regression</i>	Sesi dihapus sepenuhnya, akses ke halaman dilindungi dicegah secara konsisten

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

#### **4.1 Hasil Penelitian**

Berisi hasil penelitian berdasarkan rancangan yang sudah dijelaskan pada Bab III, terutama dari Subbab 3.4. Bagi yang membuat alat, jelaskan alat yang jadi dalam bentuk apa. Bagi yang membuat aplikasi, jelaskan aplikasi yang jadi dalam bentuk seperti apa. Jabarkan dalam bentuk pseudocode dan dijelaskan per bagian kodennya. Gunakan gambar dan tabel sebagai alat bantu menjelaskan hasil.

##### **4.1.1 Implementasi Fitur**

Tabel 4.1 menunjukkan daftar fitur dan menu yang telah diimplementasikan dalam sistem EduTeams menggunakan metodologi Agile Kanban. Implementasi menggunakan pendekatan iteratif dengan fokus pada fitur-fitur inti terlebih dahulu, kemudian dikembangkan ke fitur-fitur lanjutan. Status implementasi mengikuti alur kerja Kanban dengan tiga kategori: *DONE* (fitur telah diimplementasikan dan diuji), *IN PROGRESS* (fitur sedang dalam tahap pengembangan atau perbaikan), dan *TODO* (fitur direncanakan namun belum diimplementasikan).

Tabel 4.1 Implementasi Fitur Sistem EduTeams

No	Fitur/Menu	Status
<b>A. AUTENTIKASI DAN OTORISASI</b>		
1	Login dengan Google OAuth	<i>DONE</i>
2	Role-Based Access Control (RBAC) - 3 role: Mahasiswa, Dosen, Admin	<i>DONE</i>

No	Fitur/Menu	Status
3	Session Management dengan tracking IP dan user agent	DONE
<b>B. ONBOARDING PENGGUNA</b>		
4	Pemilihan Role (Mahasiswa/Dosen) dengan auto-detection email institusi	DONE
5	Form Data Diri (NIM/NPM, nama, jenis kelamin)	DONE
6	Tes Kepribadian MBTI dengan 60 pertanyaan OJTS 1.2	DONE
7	Sistem scoring dan klasifikasi 16 tipe MBTI (EI, SN, TF, PJ)	DONE
8	Resume Onboarding untuk melanjutkan proses yang belum selesai	DONE
9	Welcome Splash Screen untuk pengguna baru	DONE
<b>C. DASHBOARD</b>		
10	Dashboard Mahasiswa dengan daftar kelas yang diikuti	DONE
11	Dashboard Dosen dengan statistik (total kelas, mahasiswa, tugas)	DONE
12	Navigasi Sidebar dan Top Navigation Bar	DONE
<b>D. MANAJEMEN KELAS</b>		
13	Pembuatan Kelas Baru (nama mata kuliah, kelas, tahun, periode)	DONE
14	Join Kelas via Share Token untuk mahasiswa	DONE
15	Daftar Mahasiswa per Kelas dengan data MBTI	DONE
16	Kelola Kelas (Edit, Hapus, Arsip)	DONE
17	Search dan Filter Mahasiswa	DONE
18	Hapus Mahasiswa dari Kelas	DONE

No	Fitur/Menu	Status
<b>E. MANAJEMEN TUGAS</b>		
19	Pembuatan Tugas (Assignment) dengan judul, deskripsi, tanggal mulai	<i>DONE</i>
20	Kelola Tugas dalam bentuk Table View dengan search dan filter	<i>DONE</i>
21	Edit Tugas dengan validation	<i>DONE</i>
22	Sistem Versioning Assignment untuk tracking perubahan	<i>DONE</i>
23	Deteksi Perubahan Assignment dengan flagging submission	<i>DONE</i>
24	Hapus dan Arsip Tugas	<i>DONE</i>
25	Sistem Topik per Assignment	<i>DONE</i>
<b>F. QUIZ DAN PREFERENSI</b>		
26	Quiz Penilaian Skill dengan Likert scale	<i>DONE</i>
27	Quiz Preferensi Topik untuk assignment	<i>DONE</i>
28	Submission Tracking dengan status (Belum Isi, Menunggu, Berhasil)	<i>DONE</i>
29	Resubmission untuk assignment yang berubah	<i>DONE</i>
<b>G. PEMBENTUKAN KELOMPOK</b>		
30	Integrasi Algoritma Pembentukan Kelompok (Edu2com API)	<i>DONE</i>
31	Algoritma Multi-Faktor (MBTI, Skill, Gender, Preferensi Topik)	<i>DONE</i>
32	Pembentukan kelompok berdasarkan jumlah kelompok	<i>DONE</i>
33	Pembentukan kelompok berdasarkan ukuran kelompok	<i>DONE</i>

No	Fitur/Menu	Status
34	Team Quality Scoring untuk setiap kelompok	DONE
35	Tampilan Hasil Kelompok untuk Dosen dan Mahasiswa	DONE
36	Reset dan Ulang Pembentukan Kelompok	DONE
<b>H. ANALITIK DAN VISUALISASI</b>		
37	Chart Distribusi 16 Tipe MBTI	DONE
38	Chart Distribusi Skill Mahasiswa	DONE
39	Chart Distribusi Gender	DONE
40	Radar Chart Personality (EI, SN, TF, PJ)	DONE
41	Statistik Submission Rate Assignment	DONE
<b>I. PROFIL DAN PENGATURAN</b>		
42	Halaman Profil dengan data lengkap MBTI	DONE
43	Deskripsi karakteristik 16 Tipe MBTI	DONE
44	Language Switcher (Bahasa Indonesia dan English)	DONE
45	CTA dan Navigasi pada Landing Page	DONE
<b>J. FITUR TAMBAHAN</b>		
46	Loading States dengan Suspense dan Skeleton Components	DONE
47	Error Handling dan Error Pages	DONE
48	Responsive Design (Mobile, Tablet, Desktop)	DONE
49	Fitur Aksesibilitas Dasar (ARIA Labels pada UI Components)	DONE
50	Multi-layer Caching Strategy (Memory + Redis)	DONE
<b>K. FITUR DALAM PENGERJAAN</b>		
51	Pre-fill Jawaban Quiz Skill dan Preferensi Sebelumnya	IN PROGRESS

No	Fitur/Menu	Status
52	Halaman Management untuk Mahasiswa dengan Tabs	<i>IN PROGRESS</i>
<b>L. FITUR YANG DIRENCANAKAN</b>		
53	Sistem Preferensi Antar Mahasiswa (Person-to-Person Preferences)	<i>TODO</i>
54	Notifikasi Real-time menggunakan WebSocket/SSE	<i>TODO</i>
55	Notifikasi Email untuk event penting menggunakan Resend	<i>TODO</i>
56	Export Data Kelompok ke PDF dan Spreadsheet	<i>TODO</i>
57	Notification Badge pada Management Page	<i>TODO</i>
58	Advanced Settings untuk Bobot Pembagian Kelompok	<i>TODO</i>
59	Legend dan Sort untuk MBTI Chart	<i>TODO</i>

Dari total 59 fitur yang diidentifikasi, 50 fitur (84.7%) telah selesai diimplementasikan dan diuji, 2 fitur (3.4%) sedang dalam tahap penggerjaan berupa implementasi fitur tambahan untuk mahasiswa, serta 7 fitur (11.9%) direncanakan untuk pengembangan selanjutnya. Tingkat kelengkapan implementasi yang tinggi ini menunjukkan bahwa sistem EduTeams telah mencapai tahap maturity yang baik untuk fitur-fitur inti dan siap untuk digunakan dalam lingkungan produksi dengan rencana pengembangan berkelanjutan untuk fitur-fitur tambahan.

Contoh implementasi kode dapat dituliskan menggunakan `\begin{lstlisting}`. Contoh kode dapat dilihat pada Kode 4.1.

Kode 4.1 Akuisisi Gambar

```

1 | def process_dataset(dataset_path):
2 |     image_files = glob(os.path.join(dataset_path, '*.*.png'))
3 |     image_files.sort()

```

```

4     for image_file in image_files:
5         frame = cv2.imread(image_file)
6         if frame is None:
7             continue
8         frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
9         cv2.imshow('Frame', frame)
10        if cv2.waitKey(1) & 0xFF == ord('q'):
11            break
12    cv2.destroyAllWindows()
13 def main():
14     datasets = get_all_dataset_folders(DATASET_ROOT)
15     for dataset in datasets:
16         process_dataset(dataset)
17         print("print string")

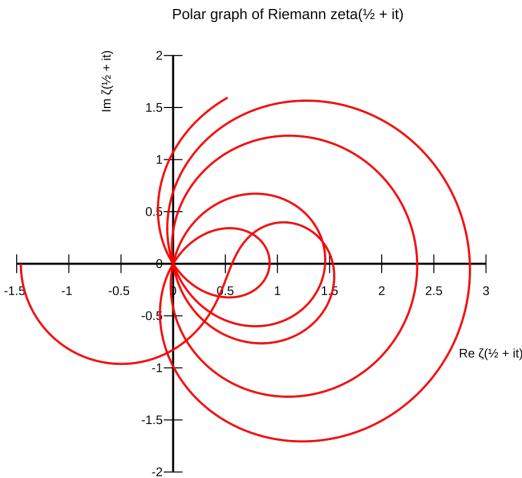
```

## 4.2 Hasil Pengujian

Berikan hasil pengujian berdasarkan rancangan & skenario yang sudah direncanakan sebelumnya pada Subbab ??.

Tabel 4.2 Data *dummy* Pengujian

Subjek	Hasil Prediksi (BPM)							GT
	F	NA	NO	RC	LC	M	C	
1	68	69	68	70	68	71	69	68
2	69	69	68	70	68	71	69	69
3	70	70	69	71	68	73	69	70
4	71	70	70	72	69	73	70	71
5	72	72	70	72	70	74	70	72



Gambar 4.1 Contoh Graf Pengujian

### 4.3 Analisis Hasil Penelitian

Berikan analisis hasil penelitian & pengujian, berupa data yang didapatkan dari penelitian & pengujian Tugas Akhir yang sudah anda kerjakan. Gunakan gambar dan tabel sebagai alat bantu menjelaskan analisis hasil. Data luaran penelitian yang dapat dianalisis berupa:

1. Hasil pengujian
2. Hasil kuesioner
3. Aplikasi yang dikembangkan

Analisis dapat membandingkan dengan hasil penelitian sebelumnya yang memiliki kemiripan topik.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Berisi kesimpulan dari hasil dan pembahasan terkait penelitian yang dilakukan, dapat juga berupa temuan yang Anda dapatkan setelah melakukan penelitian atau analisis terhadap tugas akhir Anda. Memberikan jawaban dari poin pada subbab Rumusan Masalah dan Tujuan Penelitian.

#### **5.2 Saran**

Berisi saran mengenai aspek tugas akhir atau temuan yang dapat dikembangkan dan diperkaya di tugas akhir selanjutnya. Saran dapat berkaitan erat pada subbab Analisis Hasil Penelitian.

## **DAFTAR PUSTAKA**

- [1] J. Zygmuntowski. “When Collaboration Becomes Ubiquitously Digital: A Review of Collaborative Society”. *Zagadnienia Nauk.* 55.3 (2022), pp. 93–97.
- [2] A. Kotsonis. “Educating for Collaboration: A Virtue Education Approach”. *Ethics Educ.* 17.3 (2022), pp. 311–323.
- [3] H. S. Lu and R. Smiles. “The Role of Collaborative Learning in the Online Education”. *Int. J. Econ. Bus. Manag. Res.* 6.6 (2022), pp. 125–137.
- [4] R. Ellis and F. Han. “Assessing university student collaboration in new ways”. *Assess. Eval. High. Educ.* 46.4 (2021), pp. 509–524.
- [5] D. Mburasek and O. Musimbi. “Development of a Tool for Team Formation in Engineering Education”. *Social Science Research Network* (2021).
- [6] F. L. Vinella, S. Koppelaar, and J. Masthoff. “Forming Teams of Learners Online in a User as Wizard Study with Openness, Conscientiousness, and cognitive Ability”. *Adjunct Proceedings of the 30th ACM Conference on User Modeling, Adaptation and Personalization.* UMAP ’22 Adjunct. 2022, pp. 283–292.
- [7] I. Bernal. “Digital.CSIC: Making the Case for Open Access at CSIC”. *Ser. Rev.* 37.1 (2011), pp. 3–8.
- [8] A. Georgara, C. Sierra, and J. A. Rodríguez-Aguilar. “Edu2Com: an anytime algorithm to form student teams in companies” (2023).
- [9] M. S. Nawaz, S. U. R. Khan, S. Hussain, et al. “A study on application programming interface recommendation: state-of-the-art techniques, challenges and future directions”. *Libr. Hi Tech* 41.2 (2022), pp. 355–385.

- [10] J. Govea, E. Ocampo Edye, S. Revelo-Tapia, et al. “Optimization and Scalability of Educational Platforms: Integration of Artificial Intelligence and Cloud Computing”. *Computers* 12.11 (2023).
- [11] H. Alaidaros, M. Omar, and R. Romli. “The state of the art of agile kanban method: challenges and opportunities”. *Indep. J. Manag. Prod.* (2021).
- [12] M. E. Khan and F. Khan. “A Comparative Study of White Box, Black Box and Grey Box Testing Techniques”. *Int. J. Adv. Comput. Sci. Appl. IJACSA* 3.6 (2012).
- [13] I. M. M. Ramos, D. B. Ramos, B. F. Gadelha, et al. “An Approach to Group Formation in Collaborative Learning Using Learning Paths in Learning Management Systems”. *IEEE Trans. Learn. Technol.* 14.5 (2021), pp. 555–567.
- [14] E. M. Hastings, S. R. Krishna Kumaran, K. Karahalios, et al. “A Learner-Centered Technique for Collectively Configuring Inputs for an Algorithmic Team Formation Tool”. *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education*. 2022, pp. 969–975.
- [15] M. Kalantzi, A. Polyzou, and G. Karypis. “FERN: Fair Team Formation for Mutually Beneficial Collaborative Learning”. *arXiv preprint arXiv:2011.11611* (2020).
- [16] M. A. Ilmi, F. Pradana, and W. H. N. Putra. “Software Project Management Systems Using Kanban Method in the CV. Primavisi Globalindo”. *INTENSIF J. Ilm. Penelit. Dan Penerapan Teknol. Sist. Inf.* 4.2 (2020).
- [17] “Implementation of Kanban Techniques in Software Development Process: An Empirical Study Based on Benefits and Challenges”. *Sukkur IBA J. Comput. Math. Sci.* 3.2 (2019), pp. 25–36.

- [18] S. Arief and Y. Sugiarti. “Literature Review: Analisis Metode Perancangan Sistem Informasi Akademik Berbasis Web”. *J. Ilm. ILMU Komput.* 8 (2022), pp. 87–93.
- [19] A. Sarikaya, M. Correll, L. Bartram, et al. “What Do We Talk About When We Talk About Dashboards?” *IEEE Trans. Vis. Comput. Graph.* 25.1 (2019), pp. 682–692.
- [20] G. Maulani, H. Komara, and S. Meiliana. “Rancang Bangun Sistem Informasi Monitoring Dashboard Traffic Work Order Berbasis Web”. *J. Cerita Creat. Educ. Res. Inf. Technol. Artif. Inform.* 6.2 (2020).
- [21] *What Is Artificial Intelligence (AI)?* <https://www.ibm.com/think/topics/artificial-intelligence>. Accessed: 23 January 2025.
- [22] David J. Anderson. *Kanban: Successful Evolutionary Change for Your Technology Business*. Blue Hole Press, 2010.
- [23] Daniel S. Vacanti. *Actionable Agile Metrics for Predictability: An Introduction*. Actionable Agile, 2015.
- [24] M. Ozkaya and F. Erata. “A survey on the practical use of UML for different software architecture viewpoints”. *Inf. Softw. Technol.* 121 (2020), p. 106275.
- [25] H. Kurniawan, W. Apriliah, I. Kurniawan, et al. “Penerapan Metode Waterfall Dalam Perancangan Sistem Informasi Penggajian Pada SMK Bina Karya Karawang”. *J. Interkom J. Publ. Ilm. Bid. Teknol. Inf. Dan Komun.* 14.4 (2020), pp. 159–169.
- [26] H. K. Al-Masree. “Extracting Entity Relationship Diagram (ERD) From Relational Database Schema”. *Int. J. Database Theory Appl.* 8.3 (2015), pp. 15–26.
- [27] A Comparative Study of Software Testing Techniques Viz. White Box Testing Black Box Testing and Grey Box Testing. [https://www.researchgate.net/publication/276028491\\_A\\_Comparative\\_Study\\_of\\_Software\\_Testing\\_Techniques\\_](https://www.researchgate.net/publication/276028491_A_Comparative_Study_of_Software_Testing_Techniques_)

[Viz\\_White\\_Box\\_Testing\\_Black\\_Box\\_Testing\\_and\\_Grey\\_Box\\_Testing.](#)

Accessed: 23 January 2025.

- [28] I. R. Dhaifullah, M. M. H, A. A. Salsabila, et al. “Survei Teknik Pengujian Software”. *J. Autom. Comput. Inf. Syst.* 2.1 (2022), pp. 31–38.

## **LAMPIRAN**

- A   Dataset**
- B   Hasil Wawancara**
- C   Rincian Kasus Uji**