

**Application of Agile Methodology in the SIAK Mini
Project using Taiga and Github**

Name : Radingka Rocha Arfian
Student ID : 255150200111014
Study Program : Informatics Engineering
Class : TIF-B
Lecturer : Dr. Achmad Arwan, S.Kom., M.Kom.

TABLE OF CONTENTS

TABLE OF CONTENTS.....	2
CHAPTER I.....	4
INTRODUCTION.....	4
1.1 Background	4
1.2 Problem Statement.....	4
1.3 Objectives	4
1.4 Benefits	4
CHAPTER II	5
THEORETICAL FRAMEWORK	5
2.1 Agile Methodology.....	5
2.2 Scrum Framework.....	5
2.3 Taiga Project Management	5
2.4 GitHub Platform.....	5
2.5 Taiga and GitHub Integration	6
CHAPTER III	7
DISCUSSION	7
3.1 SIAK Mini Design Implementation in Taiga.....	7
3.1.1 Project Creation	7
3.1.2 Project Configuration.....	11
3.1.3 Project Execution	12
CHAPTER IV	15
RESULTS AND ANALYSIS.....	15
4.1 Sprint Implementation Result.....	15
CHAPTER V	16
CONCLUSION	16
5.1 Conclusions	16
APPENDICES	17
REFERENCES.....	19

Figure 3 1 New Project Interface	7
Figure 3 2 Add User Story Interface	8
Figure 3 3 New Sprint Interface and Added Sprint	9
Figure 3 4 New Task Interface	10
Figure 3 5 Example of Filled Task Interface	11
Figure 3 6 Example of Filled Task Interface	11
Figure 3 7 Programming Example	12
Figure 3 8 Commit File.....	13
Figure 3 9 Taiga and Github Webhook Configuration.....	14
Figure 6 1 Backlog Interface.....	17
Figure 6 2 Sprint 1 Interface	17
Figure 6 3 Github Repository Interface	18

CHAPTER I

INTRODUCTION

1.1 Background

Today, managing software development projects requires precise methodologies to ensure efficiency and quality. Higher education institutions with many academic aspects need integrated information systems to manage the complex flow of data and processes efficiently. SIAK (Sistem Informasi Akademik) in mini size is designed to simulate academic data management project on a focused and efficient scale.

Developing the project requires a structured and flexible project management approach. Agile methodology, precisely with scrum framework is one of proven approach in software development because of its iterative nature and adaptability when facing changes. Implementing this methodology requires project management tools that support effective support team collaboration and transparent progress tracking.

Taiga, an agile-based project management website, provides many framework, including scrum. For scrum framework, taiga has user story mapping and sprint planning that also comes with custom tag for efficient team organization. Integrating github to taiga enables synchronization and transfer log, enhancing development efficiency and project progress transparency.

1.2 Problem Statement

Based on the background, the identified problems are:

- How to utilize taiga project management to manage user stories and sprint planning after system design?
- How to integrate github to taiga to enhance team collaboration efficiency?

1.3 Objectives

The objectives of this project are:

- Utilize taiga project management to implement agile methodology with scrum framework
- Integrate github to taiga for commits and issues synchronizing related to user stories

1.4 Benefits

The expected benefits of this project are:

- Enhance ability to apply agile methodology and scrum framework
- Increase practical experience using modern project management tools (taiga)
- improved skill in team collaboration using github
- Increase understanding of a structured software development workflow

CHAPTER II

THEORETICAL FRAMEWORK

2.1 Agile Methodology

Agile is a methodology aims to deliver features into smaller tasks within short duration of time by iterations that consists pre-planning, planning, release planning, iteration planning, and manage product backlog. Agile comes into the software industry as an answer of fast-paced industry that requires rapid responses to ever-changing demands. Currently, agile is one of well-known software development process/strategy.

2.2 Scrum Framework

Scrum is one of the most popular agile framework for managing complex adaptive problems. It provides steps to control the software and product development process. It comprises three components, roles, ceremonies, and artifacts. The roles comprises product owner, scrum master, and the product developmental team. The ceremonies of the framework include the sprint planning as the goals and the sprint backlog, chosen from decided product backlog. The scrum master's goal is to teach and mentor the team and remove any complexities and problems.

2.3 Taiga Project Management

Taiga is an open-source project management tool for multi role that work in agile with scrum and kanban framework. It named after boreal forest which lies across the subarctic region that strikingly beautiful from distance but deadly if enter unprepared. Taiga provides intuitive and flexible interface for tasks, such as backlog management, sprint planning. It also can be self-hosted and integrates with other development tools like github.

2.4 GitHub Platform

Github is a web-based proprietary developer platform that allows developer to create, store, mange, and share their code. Github aims to build software-faster, smarter, and more securely. It commonly used to deliver scalable and secure solution for every team sizes. It also has github copilot to automate tasks, enhance code quality and productivity with intelligent and adaptive recommendations. Github also come with integrated tools that keep the team aligned and reduce manual tasks.

2.5 Taiga and GitHub Integration

Taiga has advantage of intuitive and flexible interface for tasks. Github has advantage of code repository of task automation, code quality enhancing, and secure solution for team. with integration of both of tools, It bridges the gap for student to support learning about teamwork and project management in near-to-real-world scenarios. Taiga holds the acceptance criteria check, pattern check, task with EE, closed tasks with AE, deviation effort estimation, unassigned tasks, tasks sd, commits task reference, and commits sd on project while tasks, closed tasks, commits, modified lines on github. One of the way to integrate both of them is with webhook.

Webhook is a feature that provide a way for notifications to be delivered whenever certain events occur to an external web server. Webhook let the user subscribe to events that happening in a software system to user's server by automatically receive a delivery of data. Webhook also used in various scenarios, such as triggering CI (continuous integration), sending notifications events from github, updating like jira about external issue, deploying to a production server, and audit purposes.

CHAPTER III

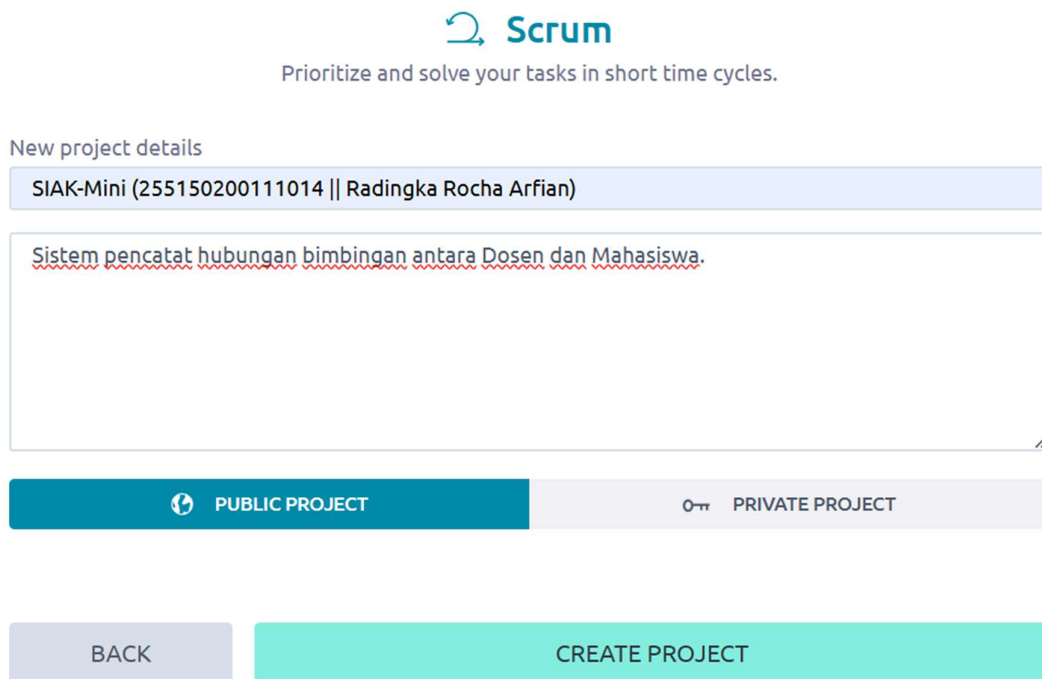
DISCUSSION

3.1 SIAK Mini Design Implementation in Taiga

3.1.1 Project Creation

1. Create New Project

After creating and login on an account, create new project in scrum template with the title and description based of the planned software design. In this case, the title is “SIAK Mini” and the description is “Sistem pencatat hubungan bimbingan antara Dosen dan Mahasiswa”.



The screenshot shows the 'New project details' form in the Scrum template. At the top, the Scrum logo and tagline 'Prioritize and solve your tasks in short time cycles.' are displayed. The form has two main input fields: a title field containing 'SIAK-Mini (255150200111014 || Radingka Rocha Arfian)' and a description field containing 'Sistem pencatat hubungan bimbingan antara Dosen dan Mahasiswa.'. Below these fields are two radio buttons for project visibility: 'PUBLIC PROJECT' (selected) and 'PRIVATE PROJECT'. At the bottom, there are two buttons: 'BACK' and 'CREATE PROJECT'.

Figure 3 1 New Project Interface

2. Create New User Story

On the backlog desktop, press “+ USER STORY” button and insert the description and the subject based of the designed user story. The description is the user story sentence and the subject is the short summary of the user story. In this case the subject is “Dosen dapat menambahkan mahasiswa” from user story in the description “Sebagai dosen, saya ingin menambahkan mahasiswa ke daftar bimbingan saya, agar data bimbingan selalu terupdate”.

New user story

0% 1 defined 1 closed 1 points /

Dosen dapat menambahkan mahasiswa

New

LOCATION

☒ at the bottom ☐ on top

Assign or Assign to me

POINTS

UX	?
Design	?
Front	?
Back	?
total points	?

0 Attachments +

Drop attachments here!

+ ADD A USER STORY

CREATE

Figure 3 2 Add User Story Interface

3. Create New Sprint

After recreating every user story in taiga's user story backlog, create new sprint with the “add+” button on the upper right corner based on designed sprint on taiga with the sprint title based on the features to be achieved in that sprint. In this case, the first sprint is “Pencatatan Hubungan Bimbingan” with one week as the time constrain. Then, drag the user story to the sprint based on designed sprint plan.

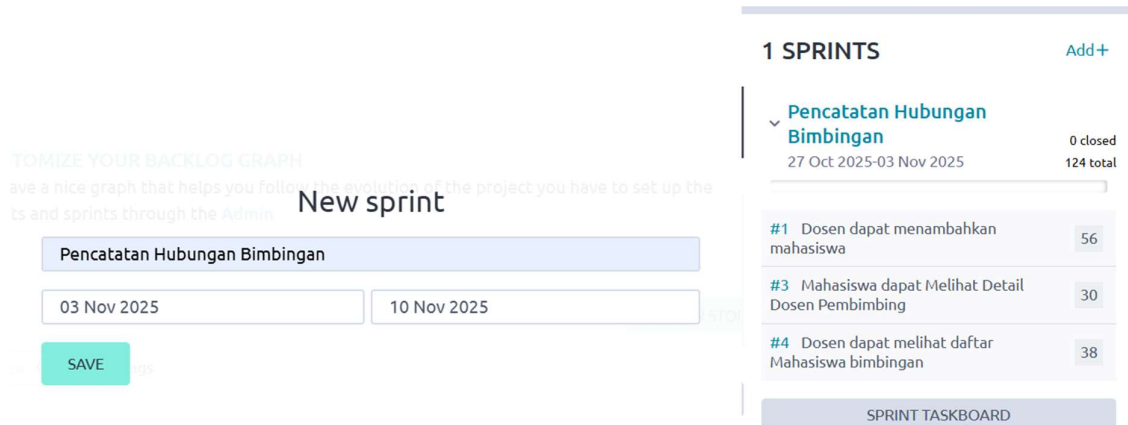


Figure 3 3 New Sprint Interface and Added Sprint

4. Create New Task

After filling the sprint with user stories, create new task on the “+” button on the user story block in sprint interface based on designed task plan. In this case, one of task on the first card is “Tambahkan atribut List mahasiswa bimbingan ke class Dosen.” as the description with the subject “Tambahkan atribut List mahasiswaBimbingan ke class Dosen” as the short summary of the description.

The screenshot displays the 'New task' interface. At the top, there's a header with 'New task' and a 'New' dropdown menu. Below this, a task title is entered: 'Add student guidance list attribute to Dosen class'. To the right of the title is an 'Assign or Assign to me' button. Below the title, there's a large text area for the description containing the text: 'Tambahkan atribut List<Mahasiswa> mahasiswaBimbingan ke class Dosen.'. Below the description area, there's a section for attachments showing '0 Attachments' and a '+'. At the bottom, there's a large green 'CREATE' button.

Figure 3 4 New Task Interface

3.1.2 Project Configuration

1. User Story Configuration

Adding priority point to user story would help the team to align user story from the most crucial to less important. Adding tag and organizing the role in the interface also help to minimizing team miscommunication and increase team proficiency with attachment if there is additional documentation.

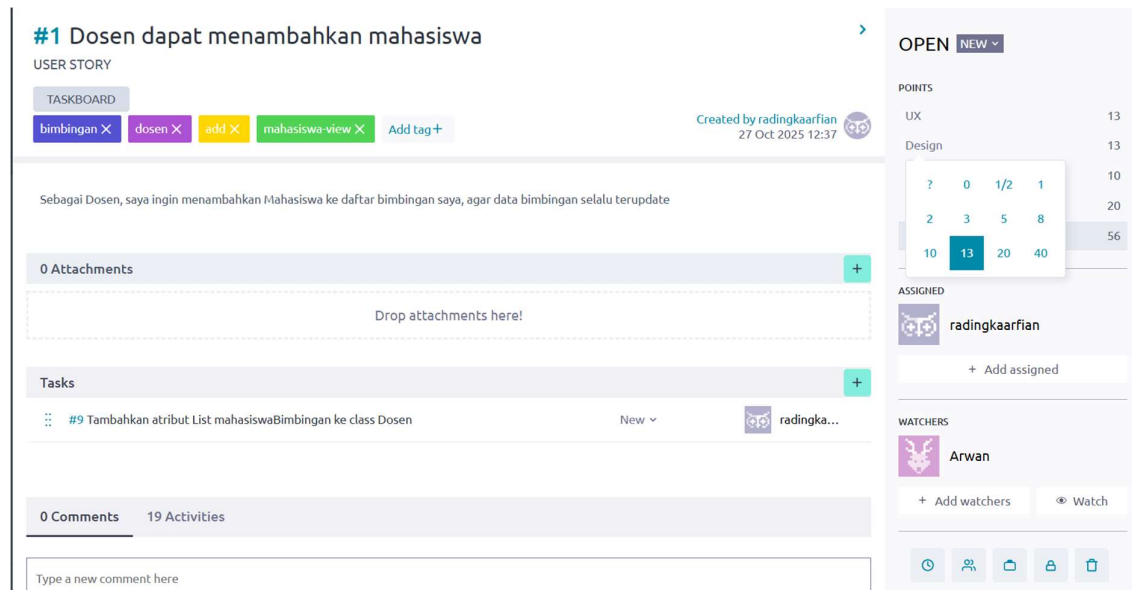


Figure 3 5 Example of Filled Task Interface

2. Task Configuration

Adding tag and organizing the role in the task interface would support team efficiency and coordination, with attachment as the reference for task details and additional documentation.

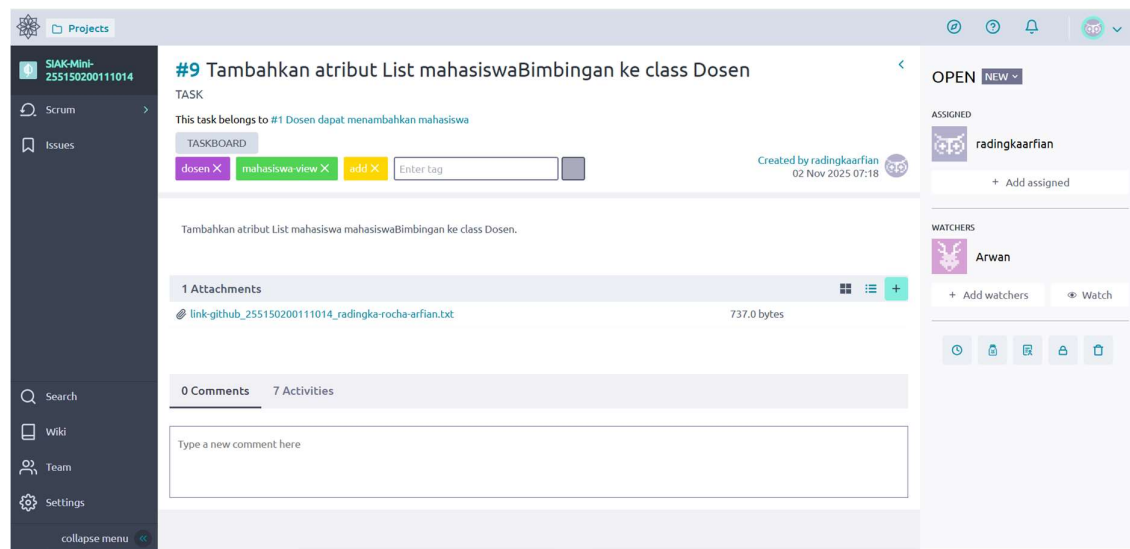
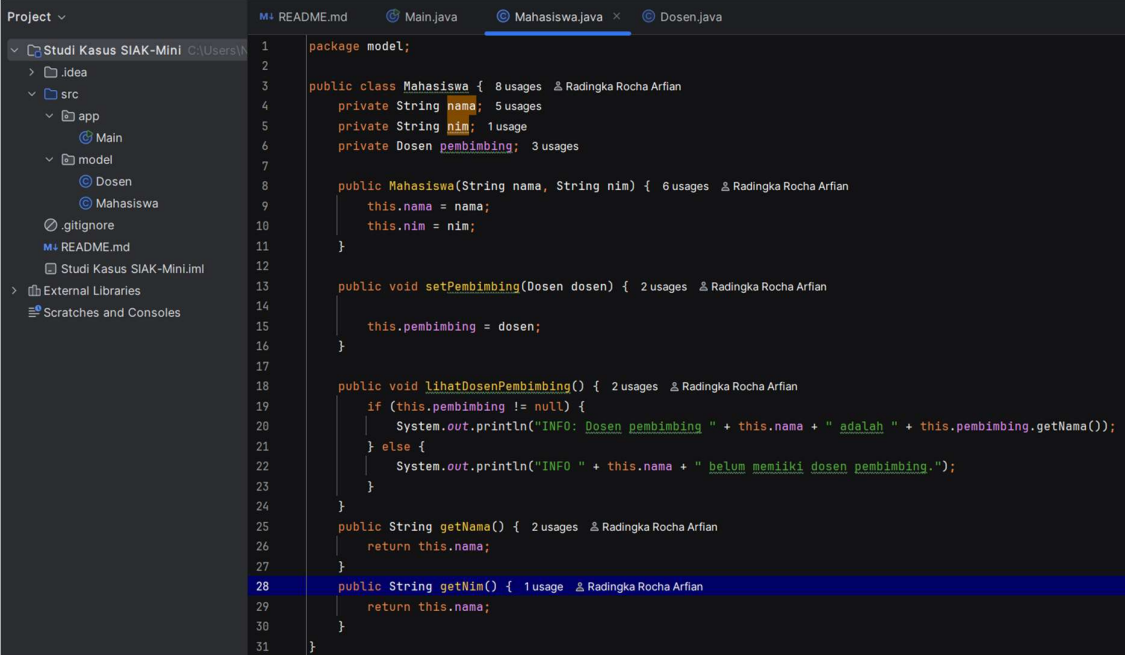


Figure 3 6 Example of Filled Task Interface

3.1.3 Project Execution

1. Programming Task

After the task and the design are clear, implement the assigned task. In this case, creating Dosen.java, Mahasiswa.java, and Main.java.



```

1 package model;
2
3 public class Mahasiswa { 8 usages  ⚠ Radingka Rocha Arfian
4     private String nama; 5 usages
5     private String nim; 1 usage
6     private Dosen pembimbing; 3 usages
7
8     public Mahasiswa(String nama, String nim) { 6 usages  ⚠ Radingka Rocha Arfian
9         this.nama = nama;
10        this.nim = nim;
11    }
12
13    public void setPembimbing(Dosen dosen) { 2 usages  ⚠ Radingka Rocha Arfian
14
15        this.pembimbing = dosen;
16    }
17
18    public void lihatDosenPembimbing() { 2 usages  ⚠ Radingka Rocha Arfian
19        if (this.pembimbing != null) {
20            System.out.println("INFO: Dosen pembimbing " + this.nama + " adalah " + this.pembimbing.getNama());
21        } else {
22            System.out.println("INFO " + this.nama + " belum memiliki dosen pembimbing.");
23        }
24    }
25
26    public String getNama() { 2 usages  ⚠ Radingka Rocha Arfian
27        return this.nama;
28    }
29
30    public String getNim() { 1 usage  ⚠ Radingka Rocha Arfian
31        return this.nim;
32    }
33 }
  
```

Figure 3 7 Programming Example

2. Github Repository

After the programming is ready to deploy, commit the local data to github repository using “share to github button” if using intelliJ IDE.

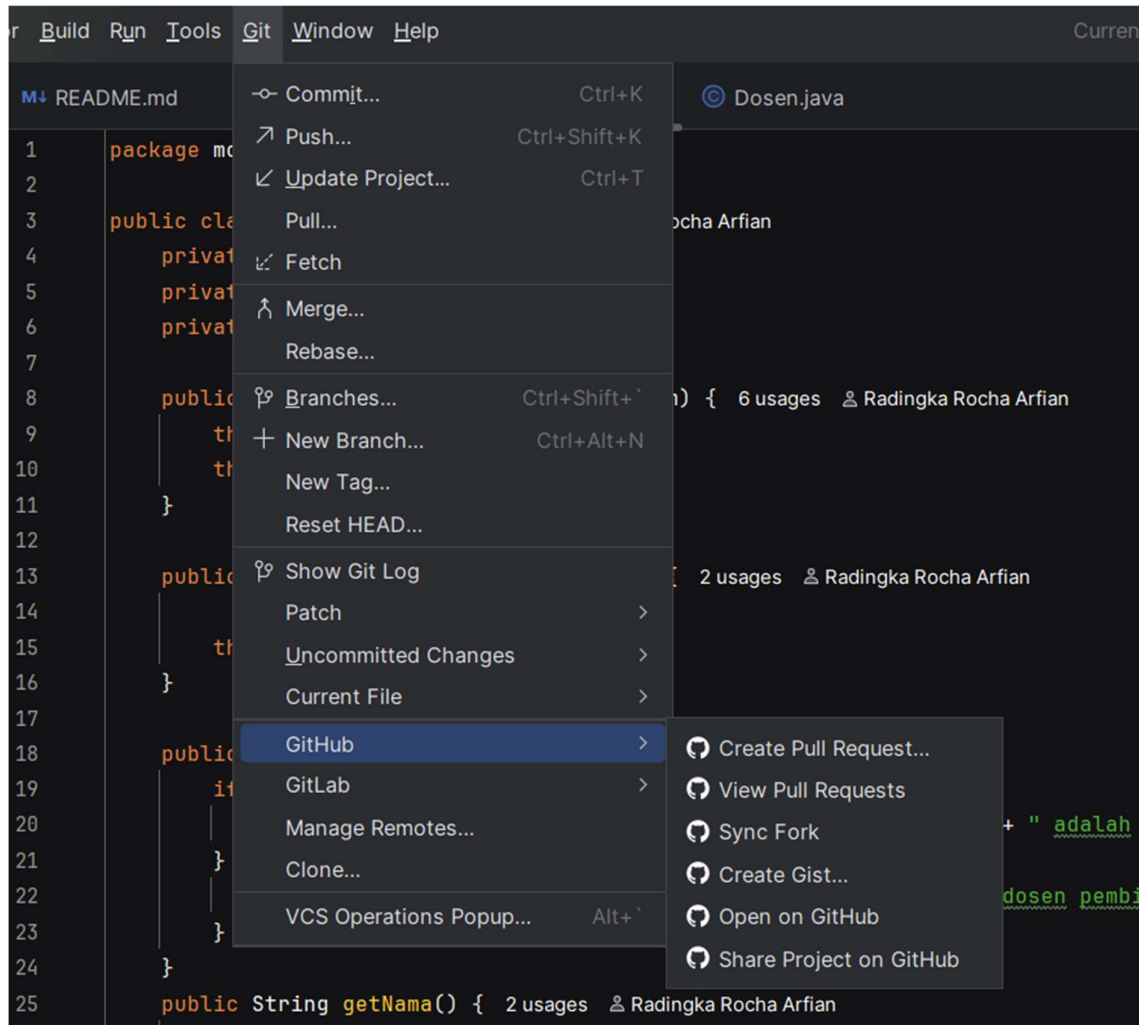


Figure 3 8 Commit File

3. Webhook Configuration

After the github repository is added, go to repository interface, then click setting > webhook > add webhook. Add the webhook using secret key and payload link from taiga with content type “application/json” that can be found on project interface, then click Settings > INTEGRATIONS > GITHUB.

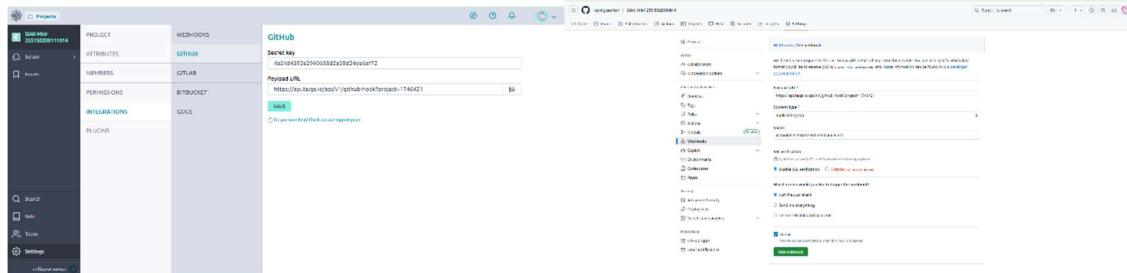


Figure 3 9 Taiga and Github Webhook Configuration

CHAPTER IV

RESULTS AND ANALYSIS

4.1 Sprint Implementation Result

Sprint 1 – Pencatatan Hubungan Bimbingan

After Implementing SIAK Mini design using taiga and github as tools, sprint 1 contains 86 total of points, with 23 for UX, 23 for Design, 15 for Front, 25 for Back as points per role. Programmed code committed in SIAK-Mini-255150200111014 github repository.

Planned User Stories

Sprint 1

US-001: "Sebagai Dosen, saya ingin menambahkan Mahasiswa ke daftar bimbingan."

Task 1: Tambahkan atribut List mahasiswaBimbingan ke class Dosen.

Task 2: Buat metode tambahMahasiswaBimbingan(Mahasiswa mhs) di class Dosen.

Task 3: Buat unit test untuk memverifikasi metode tambahMahasiswaBimbingan.

US-002: "Sebagai Mahasiswa, saya ingin melihat detail Dosen Pembimbing saya, agar saya tahu siapa yang harus dihubungi untuk konsultasi."

CHAPTER V

CONCLUSION

5.1 Conclusions

This project successfully demonstrated the practical implementation of agile methodology using scrum framework in developing SIAK Mini through integration of taiga project management and github. The Utilization of taiga proved effective in managing user stories, sprint planning, and task organization. The integration with github through webhook configuration creating unified workflow that bridged the gap between planning and implementation. This implementation demonstrated that agile approach can deliver functional increments while maintaining flexibility with proper tool support to adapt changing requirements on development process.

APPENDICES

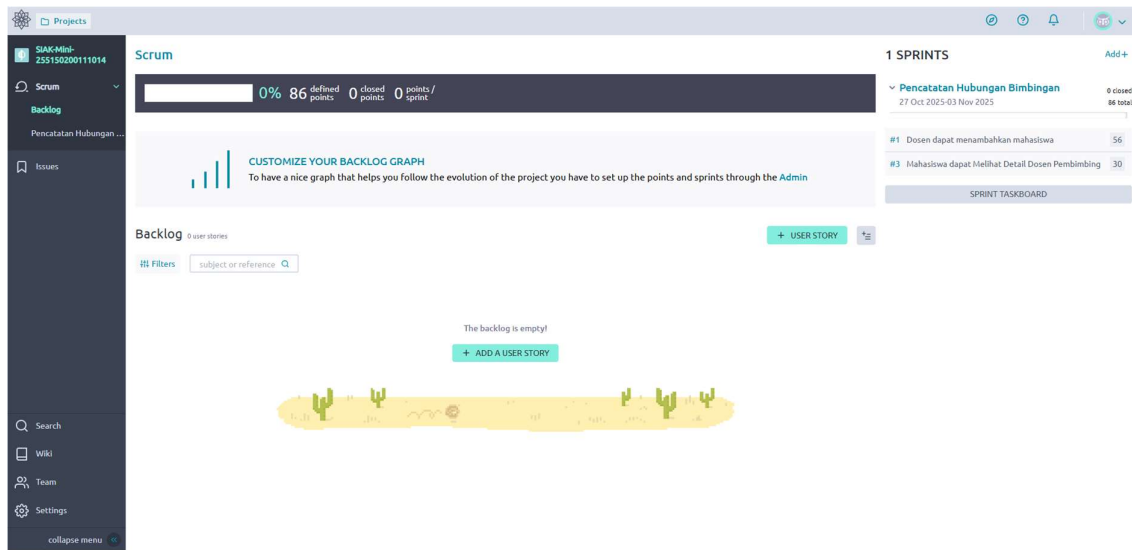


Figure 6 1 Backlog Interface

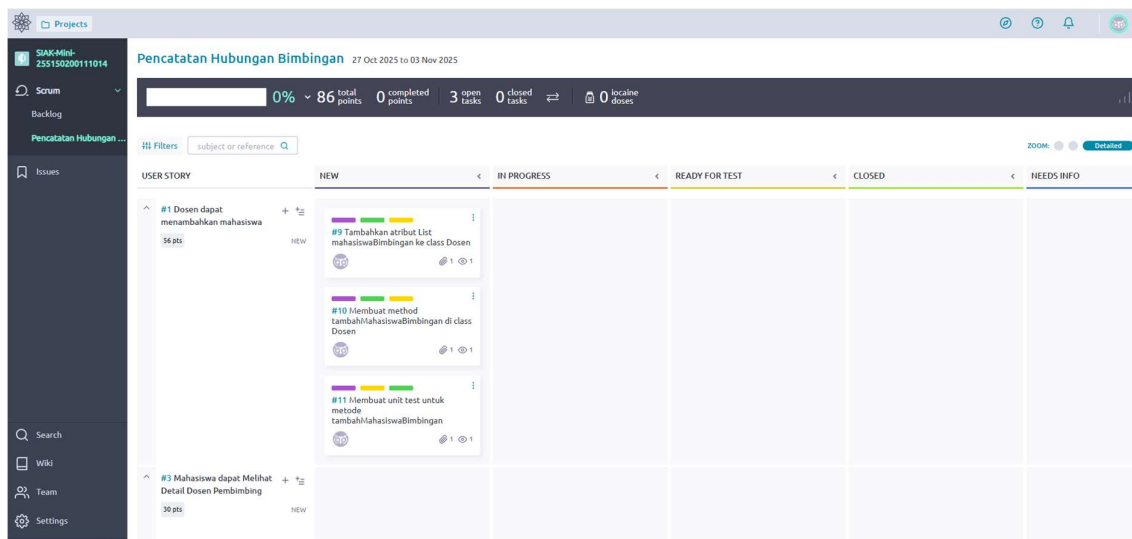


Figure 6 2 Sprint 1 Interface

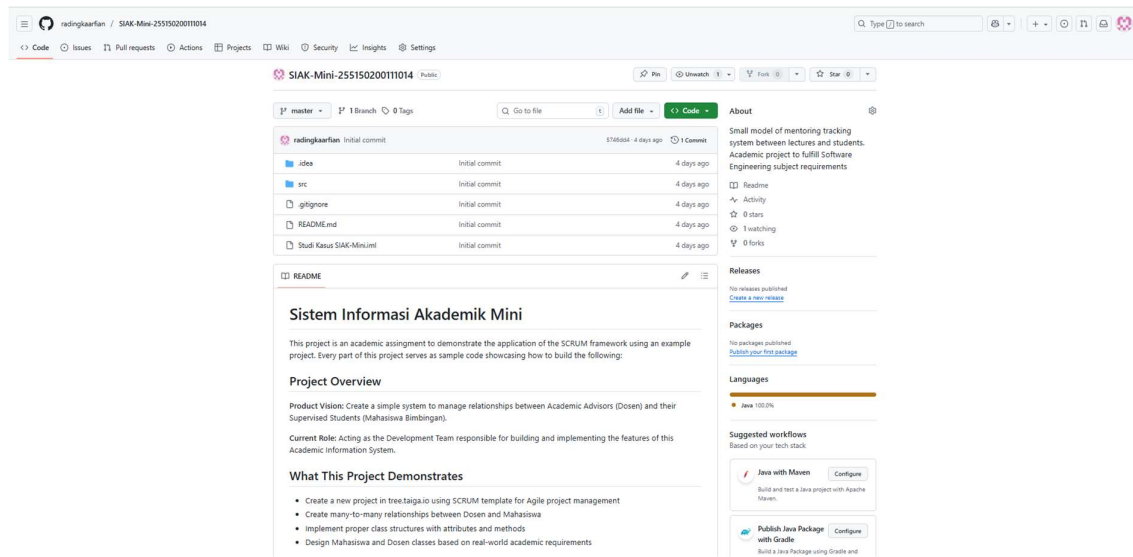


Figure 6 3 Github Repository Interface

Link Taiga:

<https://tree.taiga.io/project/radingkaarfian-siak-mini-255150200111014-radingka-rocha-arfian/backlog>

Link Github:

<https://github.com/radingkaarfian/SIAK-Mini-255150200111014.git>

REFERENCES

Farre, C., Lopez, L., Oriol, M., and Franch, X., 2025. LD@Taiga: An Embedded Learning Dashboard for Agile Project Management in Student Teams. Universitat Politècnica de Catalunya, [online] pp. 242-248. Available at: <<https://upcommons.upc.edu/server/api/core/bitstreams/7ab997e5-6daf-4ab1-af18-5c4face0fb3a/content>>

Waja, G., Shah, J., Nanavati, P., 2021. Agile Software Development. IT Department, KJSCE, [online] 5(12), pp 73-78. Available at: <https://www.researchgate.net/publication/354201110_AGILE_SOFTWARE_DEVELOPMENT>

Taiga.io. (n.d.). About us. [online] Available at: <https://taiga.io/about-us/> [Accessed 1 Nov. 2025].

GitHub. (n.d.). Why GitHub. [online] Available at: <https://github.com/why-github> [Accessed 1 Nov. 2025].