

SnapTask: Web Aplikasi untuk Manajemen Proyek

SnapTask: Web Application for Project Management

ZAIMA FIROOS LIKAN^{1*}, RUSYDI BALFAS^{2*}, RIZKY RASYID WIRAKUSUMA³,
MUHAMMAD FIRZA GYANDRA SUKMA⁴

Abstrak

Keberhasilan sebuah proyek memiliki peran yang sangat signifikan di lingkungan. Keberhasilan proyek dapat diindikasikan oleh beberapa faktor utama, dimana terdapat 3 faktor yang perlu diperhatikan di dalam pelaksanaan pekerjaan proyek, yaitu biaya, waktu, dan mutu (Kerzner 2003). Untuk memastikan penyelesaian proyek yang sesuai dengan kebutuhan, dibutuhkan sebuah sistem untuk manajemen proyek untuk meminimalisir masalah yang mungkin timbul dalam pengembangan proyek (Kale *et al.* 2017). Sistem SnapTask dikembangkan dengan alur proses incremental dan metode agile menggunakan kerangka kerja Scrum dan Kanban. Pengembangan dilakukan dengan berbagai kebutuhan dalam tim dan dalam waktu yang sesuai. Pengembang berharap SnapTask dapat membantu Gen-Z dalam mengelola proyek mereka secara lebih efektif dan meningkatkan kolaborasi, meskipun pengguna tetap perlu mengembangkan keterampilan manajemen proyek secara mandiri.

Kata Kunci: agile, manajemen proyek, perangkat lunak, proyek, scrum

Abstract

The success of a project plays a very significant role in the environment. Project success can be indicated by several key factors, where there are three factors that need to be considered in the execution of project work, namely cost, time, and quality (Kerzner 2003). To ensure project completion that meets requirements, a project management system is needed to minimize potential problems in project development (Kale et al. 2017). The SnapTask system is developed with an incremental process flow and agile methods using the Scrum and Kanban frameworks. Development is carried out with various team needs and in a timely manner. The developers hope that SnapTask can help Gen-Z manage their projects more effectively and improve collaboration, although users still need to develop project management skills independently.

Keywords: agile, project, project management, scrum, software

PENDAHULUAN

Latar Belakang

Ketika sebuah proyek berhasil, yakni memenuhi kebutuhan orang-orang yang menggunakannya, berjalan dengan sempurna selama waktu tertentu, penyampaian produk tepat waktu maka jelas proyek dapat membuat berbagai hal menjadi lebih baik. Sebaliknya, ketika proyek gagal—ketika kebutuhan orang-orang tidak terpenuhi—hal-hal buruk dapat terjadi (Pressman 2015). Hal ini membuktikan bahwa keberhasilan sebuah proyek sangat signifikan di lingkup proyek tersebut berada, entah sosial, ekonomi, atau bisnis. Setiap keputusan dan langkah yang diambil dalam proyek adalah sangat penting dan dapat menyebabkan perubahan dalam proyek.

¹Zaima Firoos Likan, G6401221002, Paralel Praktikum 2, Departemen Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Bogor, Indonesia;

*Penulis Korespondensi: Tel/Faks: 6281385885812; Surel: zaimafirooslikan@apps.ipb.ac.id

²Rusydi Balfas, G6401221104, Paralel Praktikum 2, Departemen Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Bogor, Indonesia;

*Penulis Korespondensi: Tel/Faks: 62; Surel: rusydirusydi@apps.ipb.ac.id

³Rizky Rasyid Wirakusuma, G6401221127, Paralel Praktikum 2, Departemen Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Bogor, Indonesia;

⁴Muhammad Firza Gyandra Sukma, G6401221122, Paralel Praktikum 2, Departemen Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Bogor, Indonesia;

Keberhasilan pekerjaan pelaksanaan proyek dapat diindikasikan oleh beberapa faktor utama, dimana terdapat 3 faktor yang perlu diperhatikan di dalam pelaksanaan pekerjaan proyek, yaitu biaya, waktu, dan mutu (Kerzner 2003). Dalam proyek, satu atau lebih dari faktor tersebut dapat memiliki beban atau nilai yang lebih besar dari faktor lainnya, dan seorang manajer proyek harus dapat menyeimbangkan faktor-faktor tersebut dengan cara yang dapat mencapai semua tujuan dari proyek (Mahmoudi dan Feylizadeh 2017). Dalam menyeimbangkan faktor tersebut diperlukan sebuah cara untuk manajemen proyek yang dapat diterapkan pada seluruh jenis proyek secara luas. Manajemen proyek sendiri adalah proses yang dimulai dari perencanaan, penjadwalan, pengelolaan sumber daya, analisis kebutuhan, perancangan, dan pengujian untuk mencapai tujuan dan sasaran proyek (Vidianto, Haji 2020).

Tanpa manajemen proyek yang baik dan sesuai, akan sulit bagi sebuah proyek untuk selesai yang sesuai dengan kebutuhan dan persyaratan penyelesaian dari proyek yang ada. Untuk memastikan penyelesaian proyek yang sesuai dengan kebutuhan, dibutuhkan sebuah sistem untuk manajemen proyek untuk meminimalisir masalah yang mungkin timbul dalam pengembangan proyek (Kale *et al.* 2017).

Pada zaman digitalisasi ini, sistem harus efisien dan efektif saat digunakan sehingga teknologi informasi harus diintegrasikan ke dalam sistem untuk mempermudah manajemen proyek, terutama dalam otomatisasi. Semakin efisien sistem manajemen proyek, semakin efisien pula sebuah proyek dapat dijalankan. Untuk itu, dikembangkan aplikasi berbasis website untuk manajemen proyek dengan nama SnapTask. SnapTask akan digunakan hanya oleh manajer proyek, dimana SnapTask akan membantu manajer proyek dalam mengontrol proyek, membagi tugas selama proyek berjalan, dan pembagian waktu untuk setiap tugas.

Tujuan Aplikasi

SnapTask memiliki tujuan untuk:

1. Membantu manajer proyek dalam manajemen sumber daya dan faktor-faktor yang dimiliki sebuah proyek,
2. Menjadi sebuah sistem yang efektif dan efisien untuk manajemen proyek,
3. Membantu manajer proyek dalam peningkatan produktivitas yang dimiliki oleh sebuah proyek.

Manfaat Aplikasi

SnapTask diharapkan memberikan manfaat sebagai berikut:

1. Bagi pengembang: menyelesaikan proyek akhir sebagai syarat lulus mata kuliah Rekayasa Perangkat Lunak, menambah wawasan dan kemampuan dalam pengembangan perangkat lunak, dan membuat pengembang menjadi lebih terbiasa dalam pengembangan.
2. Bagi pengguna: dapat digunakan sebagai alternatif dalam sistem manajemen proyek.

Ruang Lingkup Aplikasi

Sebuah aplikasi harus dapat memenuhi kebutuhan-kebutuhan berbagai pengguna. Namun, dalam manajemen proyek, banyak kebutuhan yang diperlukan dan dapat berubah sehingga SnapTask akan dilimitasi fungsinya dalam beberapa hal. Ruang lingkup SnapTask terbatas pada:

1. SnapTask tidak mendukung untuk dilihat dan digunakan sebagai sistem kolaborasi oleh seluruh anggota proyek, hanya dapat dilihat oleh manajer proyek,

2. Aplikasi dapat membantu dalam pembagian kerja berdasarkan waktu, namun belum dapat membagi berdasarkan sumber daya yang dimiliki proyek,
3. SnapTask tidak dapat melihat histori dari kerja yang telah diselesaikan.

METODE

Arsitektur Pengembangan Aplikasi

| | |
|-------------------------------|--|
| Jenis aplikasi | : Web aplikasi |
| Alasan memilih jenis tersebut | : Mudah digunakan |
| Sistem operasi | : Windows 11 |
| <i>Text editor</i> | : Visual Studio Code |
| DBMS | : MySQL |
| <i>Web server</i> | : Xampp |
| Framework | : Native |
| Target dari aplikasi | : Gen-Z (anak-anak muda berumur 11-30 tahun) |
| Kriteria target | : Manajer proyek |
| Fitur yang tersedia | : |
| | 1. Quick tasks: Memungkinkan pengguna membuat, mengedit, dan menghapus tugas dengan cepat dan mudah |
| | 2. Project folder: Memungkinkan pengguna untuk membuat folder baru untuk jenis proyek tertentu dan mengklasifikasikan tugas mereka ke dalam folder tersebut. |

Alur Proses Incremental

Alur proses yang dipakai selama pengembangan SnapTask adalah alur incremental. Model proses incremental adalah model proses yang dipilih ketika kebutuhan awal didefinisikan dengan cukup baik, namun keseluruhan ruang lingkup upaya pengembangan menghalangi proses yang murni linier (Pressman 2015). SnapTask menggunakan klien palsu selama proses pengembangan, sehingga manajer proyek dapat merasakan proses komunikasi yang dibutuhkan selama pengembangan. Penggunaan klien palsu ini menyebabkan SnapTask memiliki *requirements* yang didefinisikan dengan cukup baik, namun tetap memiliki beberapa fungsi atau fitur-fitur yang ditambahkan atau dikurangkan selama proses pengembangan.

Tahapan yang dilakukan menggunakan proses incremental selanjutnya disempurnakan menggunakan metode agile yang dipakai dalam pengembangan SnapTask.

Metode Agile

Agile adalah metode yang digunakan untuk meratakan kurva biaya perubahan yang memungkinkan tim perangkat lunak untuk mengakomodasi perubahan di akhir proyek tanpa dampak biaya dan waktu yang dramatis. Metode ini memecah keseluruhan proyek menjadi bagian-bagian kecil yang akan dikerjakan secara berkala, yang selanjutnya disebut sebagai *sprint* (Shidqi dan Riqky 2021). Dalam metode agile ini, SnapTask menggunakan kerangka kerja Scrum dan Kanban. Hal ini dikarenakan Scrum dan Kanban dapat menjadi metode yang efektif untuk waktu pengembangan perangkat lunak yang singkat. Selain itu, fase *backlog* dapat membuat pengembang menjadi lebih fokus dalam proses pengembangan.

Tahapan yang dilakukan dalam pengembangan SnapTask menggunakan gabungan dari Scrum dan Kanban selanjutnya ditunjukkan dalam Tabel 1.

Tabel 1 Tahapan pengembangan SnapTask dengan Scrum

| No | Tahapan | Rincian |
|----|--------------------------------------|--|
| 1 | Product backlog dan <i>designing</i> | <p>Mengadakan komunikasi kepada klien</p> <p>Mendaftar semua fitur, fungsi, persyaratan, peningkatan, dan perbaikan yang diinginkan untuk SnapTask</p> <p>Mengurutkan seluruh daftar tersebut berdasarkan prioritas</p> <p>Menuliskan detail dan persyaratan spesifik untuk setiap fitur lalu menyusunnya, seperti ERD, diagram (UML, use case, class, dsbnya), UI/UX, dll</p> |
| 2 | Sprint planning meeting | <p>Memilih fitur dengan prioritas paling tinggi dari product backlog untuk dikerjakan pada sprint iterasi pertama</p> <p>Menentukan tujuan sprint</p> <p>Memecah pekerjaan yang diperlukan untuk setiap fitur menjadi lebih kecil sehingga dapat diselesaikan secara efektif dalam sprint</p> |
| 3 | Sprint backlog | <p>Membuat daftar tugas yang telah ditentukan dalam sprint planning</p> <p>Mengalokasikan pekerjaan pada tim</p> |
| 4 | Sprint | <p>Pembagian pekerjaan menggunakan Kanban Board</p> <p>Fase pengerjaan sprint sesuai dengan sprint backlog yang telah ditentukan sesuai fokus masing-masing anggota tim</p> |
| 5 | Sprint review meeting | <p>Meeting di akhir sprint untuk menunjukkan hasil dari sprint dan demonstrasi</p> |
| 6 | Sprint retrospective | <p>Meeting untuk membahas proses selama sprint, seperti bagian yang berjalan lancar, bagian yang bisa ditingkatkan, dan bagian yang mengalami hambatan dan diskusi untuk meningkatkan sprint selanjutnya</p> |
| 7 | Increment | <p>Hasil akhir dari seluruh sprint yang dilakukan, maksudnya gabungan dari seluruh sprint</p> |

HASIL DAN PEMBAHASAN

Aplikasi berbasis web yang bernama Snaptask ini dikembangkan untuk mempermudah manajer proyek dalam manajemen proyek yang mereka kembangkan. Snaptask dikembangkan menggunakan bahasa pemrograman HTML, CSS, JavaScript, PHP, dan menggunakan database MySQL. Kombinasi bahasa pemrograman ini memungkinkan pengembangan antarmuka pengguna yang interaktif dan pengelolaan data yang efisien. Seperti yang disebutkan dalam sebuah studi, "Penggunaan HTML, CSS, JavaScript, PHP, dan MySQL dalam pengembangan aplikasi web memungkinkan penciptaan sistem yang responsif dan dinamis, di mana komponen front-end dan back-end dapat berinteraksi secara efektif untuk mendukung fungsionalitas aplikasi." (Khoirunnisa *et al.* 2016).

User Stories

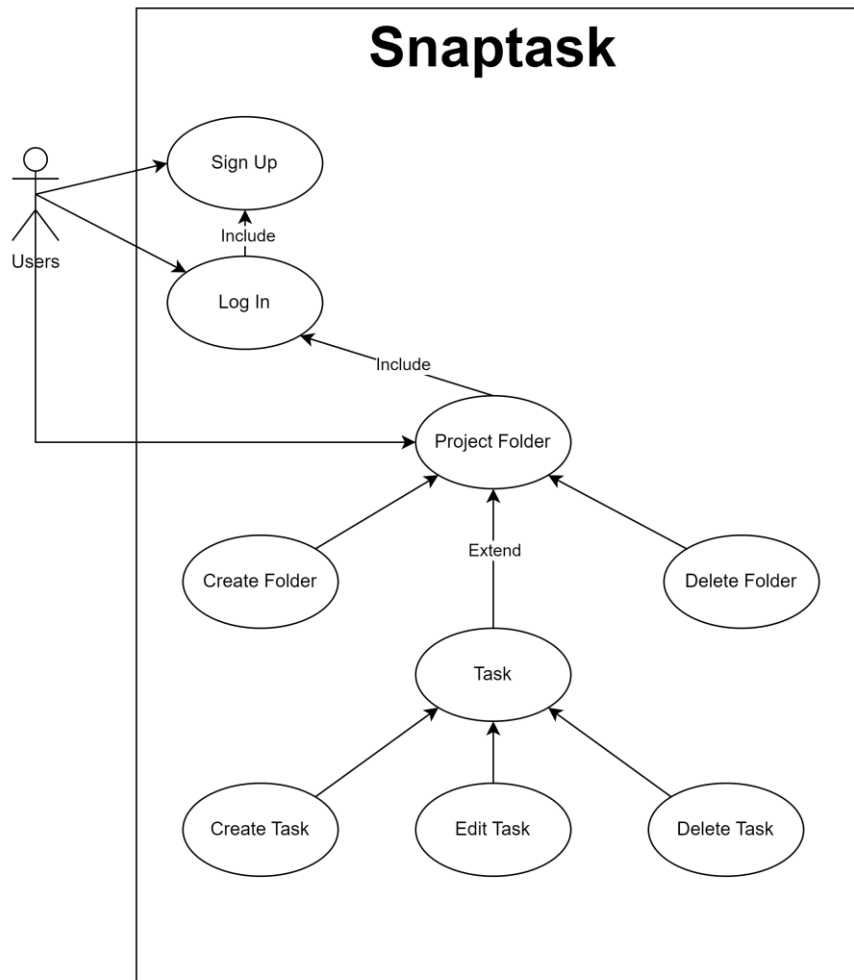
| | | |
|------------|-----------|-----------|
| Title: | Priority: | Estimate: |
| QuickTasks | High | 2 minggu |

| |
|---|
| <p>User Story:</p> <ul style="list-style-type: none"> • Sebagai pengguna yang sibuk, • Saya ingin dapat membuat tugas dengan cepat, • Sehingga saya bisa mengatur hari saya dengan efisien tanpa menghabiskan banyak waktu untuk perencanaan. |
| <p>Acceptance Criteria:</p> <ul style="list-style-type: none"> • Diberikan antarmuka yang intuitif, • Saat pengguna memilih untuk menambahkan tugas baru, • Lalu tugas tersebut dapat ditambahkan hanya dengan memasukkan judul dan tenggat waktu. |

| | | |
|---|------------------------------|----------------------------------|
| <p>Title:</p> <p>Project Folder</p> | <p>Priority:</p> <p>High</p> | <p>Estimate:</p> <p>2 minggu</p> |
| <p>User Story:</p> <ul style="list-style-type: none"> • Sebagai manajer proyek, • Saya ingin mengorganisir tugas berdasarkan proyek dalam folder yang terpisah, • Sehingga saya bisa dengan mudah melacak kemajuan proyek dan mengelola tugas dengan lebih terorganisir. | | |
| <p>Acceptance Criteria:</p> <ul style="list-style-type: none"> • Diberikan opsi untuk membuat folder baru, • Saat pengguna membuat folder dan menggolongkan tugas ke dalamnya, • Lalu sistem memungkinkan pengguna untuk melihat gambaran umum dari setiap folder, termasuk tugas yang sedang dikerjakan dan yang sudah selesai. | | |

UML Diagram

Use Case Diagram



Gambar 1 Use Case Diagram

Use Case: Sign Up

Aktor Utama: Pengguna

Tujuan dalam Konteks: Untuk mendaftar dan membuat akun baru agar dapat menggunakan aplikasi.

Prakondisi: Sistem siap menerima data pendaftaran baru, dan pengguna belum memiliki akun terdaftar.

Trigger: Pengguna memutuskan untuk mendaftar di aplikasi.

Skenario:

1. Pengguna membuka halaman pendaftaran.
2. Pengguna memasukkan data diri seperti username, email, dan password.
3. Pengguna mengirimkan formulir pendaftaran.
4. Sistem memverifikasi data dan membuat akun baru.
5. Pengguna menerima konfirmasi pendaftaran melalui email.

Eksepsi:

1. Username atau email sudah terdaftar: sistem memberikan pesan kesalahan dan meminta pengguna memasukkan data yang berbeda.

2. Data tidak valid: sistem memberikan pesan kesalahan dan meminta pengguna memperbaiki data yang dimasukkan.

Prioritas: Esensial, harus diimplementasikan pada iterasi pertama.

Frekuensi Penggunaan: Sebanyak satu kali per pengguna baru.

Saluran ke Aktor: Melalui antarmuka halaman pendaftaran.

Aktor Sekunder: Tidak ada.

Saluran ke Aktor Sekunder: Tidak ada.

Isu Terbuka:

1. Apakah perlu ada verifikasi tambahan seperti CAPTCHA untuk mencegah bot?
2. Berapa lama waktu yang diberikan untuk pengguna menyelesaikan pendaftaran sebelum sesi berakhir?

Use Case: Log In

Aktor Utama: Pengguna

Tujuan dalam Konteks: Untuk masuk ke aplikasi menggunakan kredensial yang sudah dibuat.

Prakondisi: Pengguna sudah memiliki akun yang terdaftar.

Trigger: Pengguna memutuskan untuk masuk ke aplikasi.

Skenario:

1. Pengguna membuka halaman login.
2. Pengguna memasukkan username dan password.
3. Pengguna mengirimkan formulir login.
4. Sistem memverifikasi kredensial dan mengizinkan akses jika valid.
5. Pengguna diarahkan ke beranda aplikasi.

Eksepsi:

1. Username atau password salah: sistem memberikan pesan kesalahan dan meminta pengguna mencoba lagi.
2. Akun tidak terdaftar: sistem memberikan pesan kesalahan dan mengarahkan pengguna untuk mendaftar.

Prioritas: Esensial, harus diimplementasikan pada iterasi pertama.

Frekuensi Penggunaan: Beberapa kali per hari per pengguna.

Saluran ke Aktor: Melalui antarmuka halaman login.

Aktor Sekunder: Tidak ada.

Saluran ke Aktor Sekunder: Tidak ada.

Isu Terbuka:

1. Apakah perlu ada pengamanan tambahan seperti autentikasi dua faktor?
2. Berapa kali pengguna bisa salah memasukkan password sebelum akun dikunci sementara?

Use Case: Create Project Folder

Aktor Utama: Pengguna

Tujuan dalam Konteks: Untuk membuat folder proyek baru sebagai wadah mengorganisir tugas-tugas terkait.

Prakondisi: Pengguna sudah masuk ke dalam aplikasi.

Trigger: Pengguna memutuskan untuk membuat proyek baru.

Skenario:

1. Pengguna membuka halaman pembuatan proyek.
2. Pengguna memasukkan nama proyek yang diinginkan.
3. Pengguna mengirimkan formulir pembuatan proyek.
4. Sistem membuat folder proyek baru dan menampilkan konfirmasi.

Eksepsi:

1. Nama proyek sudah ada: sistem memberikan pesan kesalahan dan meminta pengguna memasukkan nama yang berbeda.

Prioritas: Esensial, harus diimplementasikan pada iterasi pertama.

Frekuensi Penggunaan: Sesuai kebutuhan pengguna.

Saluran ke Aktor: Melalui antarmuka halaman pembuatan proyek.

Aktor Sekunder: Tidak ada.

Saluran ke Aktor Sekunder: Tidak ada.

Isu Terbuka:

1. Apakah perlu ada fitur template untuk pembuatan proyek?
2. Apakah pengguna bisa membuat proyek tanpa nama?

Use Case: Delete Project Folder

Aktor Utama: Pengguna

Tujuan dalam Konteks: Untuk menghapus folder proyek yang tidak diperlukan lagi.

Prakondisi: Pengguna sudah masuk ke dalam aplikasi dan memiliki proyek yang dapat dihapus.

Trigger: Pengguna memutuskan untuk menghapus proyek.

Skenario:

1. Pengguna membuka halaman proyek.
2. Pengguna memilih folder proyek yang akan dihapus.
3. Pengguna mengonfirmasi penghapusan.
4. Sistem menghapus folder proyek dan semua tugas terkait.

Eksepsi:

1. Pengguna membatalkan penghapusan: sistem tidak melakukan tindakan apa pun.

Prioritas: Esensial, harus diimplementasikan pada iterasi pertama.

Frekuensi Penggunaan: Sesuai kebutuhan pengguna.

Saluran ke Aktor: Melalui antarmuka halaman proyek.

Aktor Sekunder: Tidak ada.

Saluran ke Aktor Sekunder: Tidak ada.

Isu Terbuka:

1. Apakah perlu ada fitur pemulihan proyek yang terhapus?
2. Berapa lama data proyek yang dihapus disimpan sebelum dihapus permanen?

Use Case: Create Task

Aktor Utama: Pengguna

Tujuan dalam Konteks: Untuk membuat tugas baru dalam proyek yang sudah ada.

Prakondisi: Pengguna sudah masuk ke dalam aplikasi dan berada di dalam proyek tertentu.

Trigger: Pengguna memutuskan untuk menambahkan tugas baru.

Skenario:

1. Pengguna membuka halaman proyek.
2. Pengguna memilih opsi untuk menambahkan tugas baru.
3. Pengguna memasukkan informasi tugas seperti judul, deskripsi, tanggal jatuh tempo, dan kategori.
4. Pengguna mengirimkan formulir pembuatan tugas.
5. Sistem membuat tugas baru dan menampilkan konfirmasi.

Eksepsi:

1. Informasi tugas tidak lengkap: sistem memberikan pesan kesalahan dan meminta pengguna melengkapi informasi.

Prioritas: Esensial, harus diimplementasikan pada iterasi pertama.

Frekuensi Penggunaan: Sesuai kebutuhan pengguna.

Saluran ke Aktor: Melalui antarmuka halaman proyek.

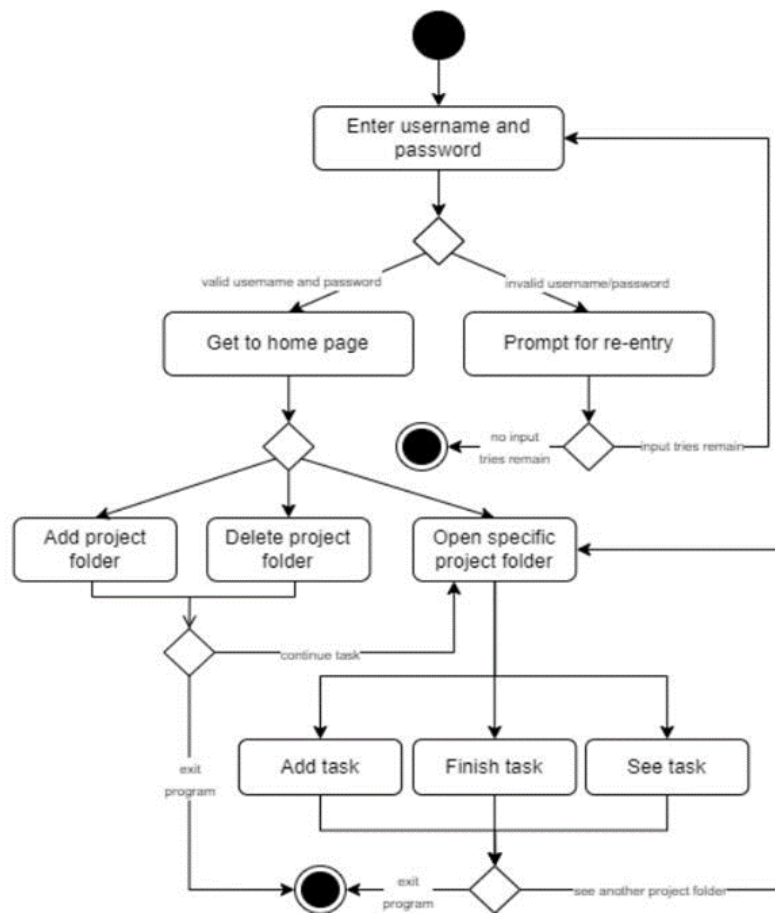
Aktor Sekunder: Tidak ada.

Saluran ke Aktor Sekunder: Tidak ada.

Isu Terbuka:

1. Apakah perlu ada pengingat otomatis untuk tugas yang mendekati tanggal jatuh tempo?
2. Apakah pengguna bisa menambahkan tugas tanpa memasukkan tanggal jatuh tempo?

Activity Diagram



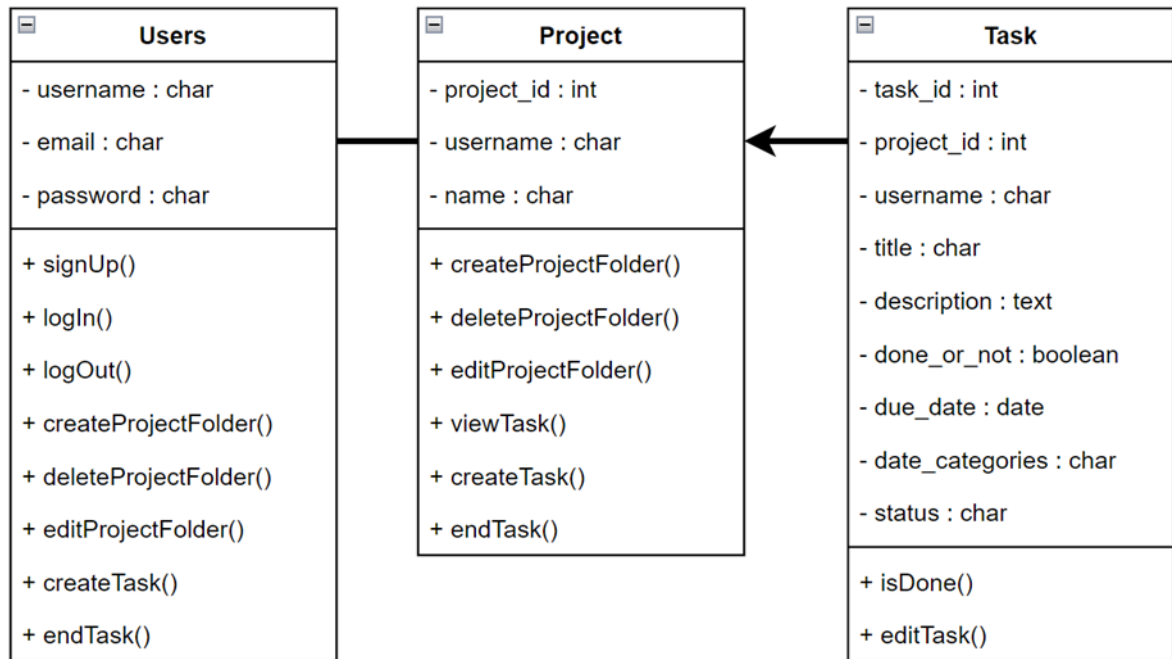
Gambar 2 Activity Diagram

Keterangan:

- Proses dimulai saat pengguna membuka web aplikasi Snaptask
- Di awal user akan diarahkan ke menu sign up untuk membuat akun, jika user sudah mempunyai akun bisa langsung klik tombol “Already Have an Account?”
- Setelah user login akan langsung diarahkan ke menu utama yang berisi daftar project si pemilik akun nya dan juga bisa menambahkan project
- Setelah user masuk ke halaman homepage, user menambahkan akun melalui tombol yang ada di home page
- Setelah user mengisi form untuk menambahkan Project, user bisa langsung ke halaman project detail

Di halaman project detail berisi daftar task yang ada untuk today, tomorrow dan juga upcoming, di halaman ini juga user bisa menambahkan task, mengedit task, menghapus task dan juga menandai suatu task selesai.

Class Diagram



Gambar 3 Class Diagram

Keterangan:

Users: Kelas yang merepresentasikan pengguna aplikasi, memiliki atribut:

- username: char:** Nama pengguna.
- email: char:** Alamat email pengguna.
- password: char:** Kata sandi pengguna.

Metode yang dimiliki:

- signUp():** Untuk mendaftarkan pengguna baru.
- login():** Untuk masuk ke dalam aplikasi.
- logout():** Untuk keluar dari aplikasi.
- createProjectFolder():** Untuk membuat folder proyek baru.
- deleteProjectFolder():** Untuk menghapus folder proyek yang ada.
- editProjectFolder():** Untuk mengedit folder proyek yang ada.
- createTask():** Untuk membuat tugas baru.
- endTask():** Untuk menandai tugas sebagai selesai.

Project: Kelas yang merepresentasikan proyek, memiliki atribut:

- project_id: int:** ID unik untuk setiap proyek.
- username: char:** Nama pengguna yang membuat proyek.
- name: char:** Nama proyek.

Metode yang dimiliki:

- createProjectFolder():** Untuk membuat folder proyek baru.
- deleteProjectFolder():** Untuk menghapus folder proyek yang ada.
- editProjectFolder():** Untuk mengedit folder proyek yang ada.
- viewTask():** Untuk melihat tugas yang ada dalam proyek.
- createTask():** Untuk membuat tugas baru dalam proyek.
- endTask():** Untuk menandai tugas dalam proyek sebagai selesai.

Task:

Kelas yang merepresentasikan tugas, memiliki atribut:

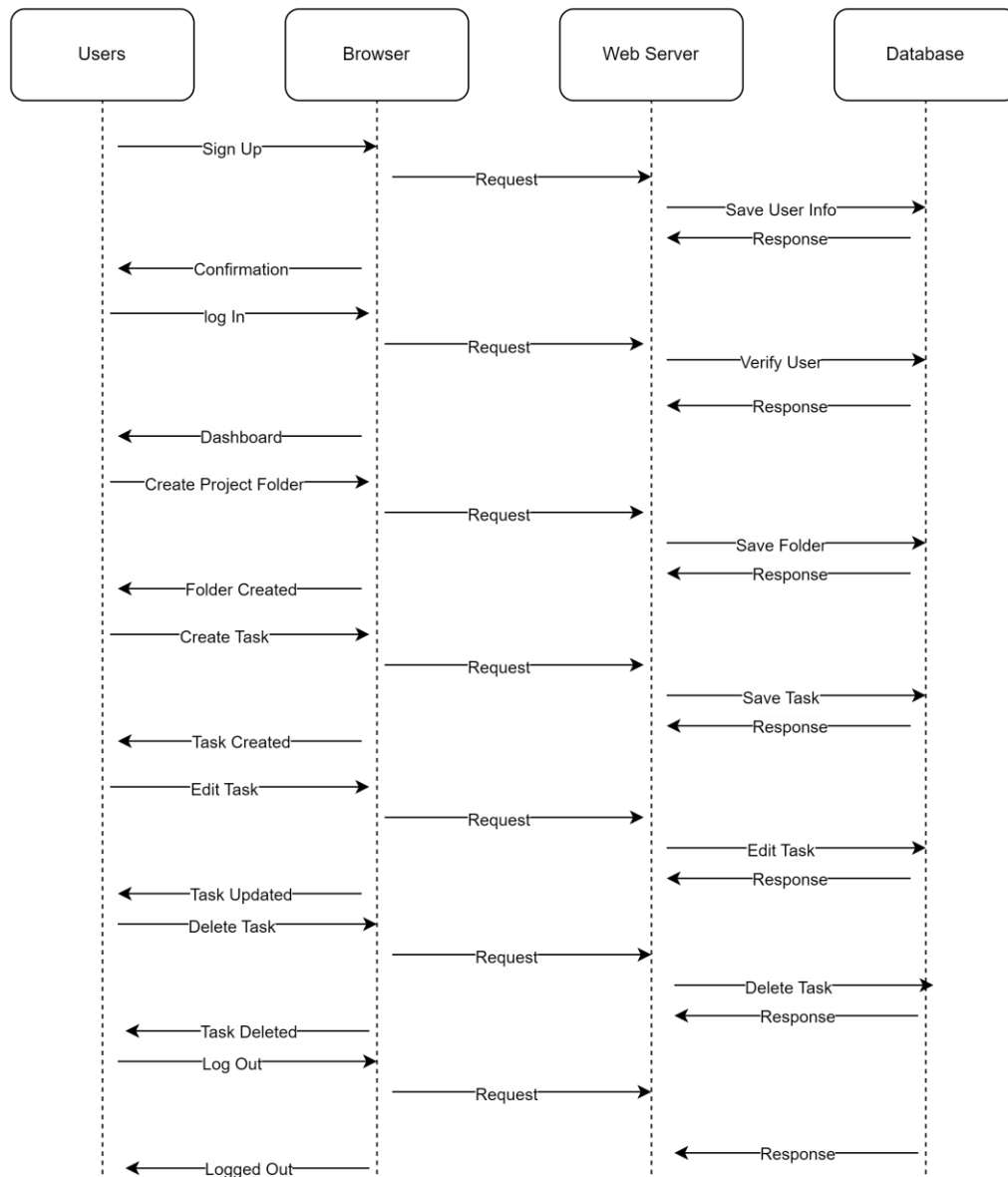
- task_id: int:** ID unik untuk setiap tugas.

- b. **project_id: int:** ID proyek yang terkait dengan tugas.
- c. **username: char:** Nama pengguna yang membuat tugas.
- d. **title: char:** Judul tugas.
- e. **description: text:** Deskripsi tugas.
- f. **done_or_not: boolean:** Status apakah tugas sudah selesai atau belum.
- g. **due_date: date:** Tanggal jatuh tempo tugas.
- h. **date_categories: char:** Kategori tanggal tugas (today, tomorrow, upcoming).
- i. **status: char:** Status tugas (misalnya, pending, in progress, completed).

Metode yang dimiliki:

- a. **isDone():** Untuk memeriksa apakah tugas sudah selesai.
- b. **editTask():** Untuk mengedit tugas yang ada.

Sequence Diagram



Gambar 4 Sequence Diagram

Sign Up:

- Pengguna mengisi formulir pendaftaran melalui browser.
- Data pendaftaran dikirim ke server melalui request.
- Server menyimpan informasi pengguna baru ke database.
- Server mengirimkan konfirmasi pendaftaran kembali ke browser, dan pengguna melihat pesan konfirmasi di layar.

Log In:

- Pengguna memasukkan kredensial log in melalui browser.
- Kredensial dikirim ke server untuk verifikasi.
- Server memverifikasi informasi dengan data yang ada di database.
- Jika berhasil, pengguna diarahkan ke dashboard aplikasi.

Create Project Folder:

- Pengguna memilih untuk membuat folder proyek baru melalui browser.
- Browser mengirim permintaan ke server untuk membuat folder proyek.
- Server menyimpan informasi folder proyek baru ke database.
- Server mengirimkan respons ke browser bahwa folder proyek berhasil dibuat.

Create Task:

- Pengguna memilih untuk membuat tugas baru dalam folder proyek.
- Browser mengirim permintaan ke server untuk membuat tugas baru.
- Server menyimpan informasi tugas baru ke database.
- Server mengirimkan respons ke browser bahwa tugas baru berhasil dibuat.

Edit Task:

- Pengguna memilih untuk mengedit tugas yang ada.
- Browser mengirim permintaan ke server untuk memperbarui informasi tugas.
- Server memperbarui informasi tugas di database.
- Server mengirimkan respons ke browser bahwa tugas berhasil diperbarui.

Delete Task:

- Pengguna memilih untuk menghapus tugas yang ada.
- Browser mengirim permintaan ke server untuk menghapus tugas.
- Server menghapus informasi tugas dari database.
- Server mengirimkan respons ke browser bahwa tugas berhasil dihapus.

Log Out:

- Pengguna memilih untuk keluar dari aplikasi.
- Browser mengirim permintaan ke server untuk log out.
- Server mengirimkan respons bahwa pengguna telah berhasil log out.

Pembagian Kerja

Pembagian kerja dalam pengembangan SnapTask dilakukan sebagai berikut:

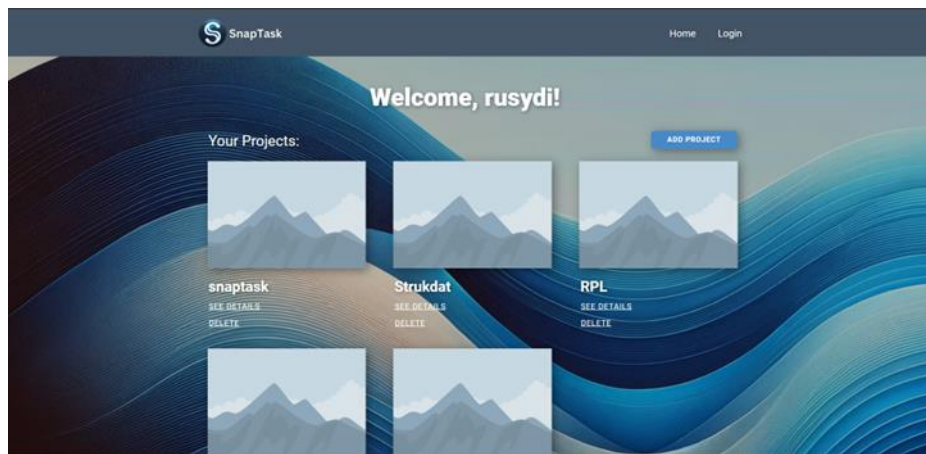
| NIM | Nama | Tugas |
|-------------|------------------------------|---|
| G6401221002 | Zaima Firoos Likan | <i>Project manager</i> , UI Designer, pembuatan demonstrasi, pembuatan laporan |
| G6401221104 | Rusydi Balfas | <i>Backend Developer</i> , pembuatan demonstrasi, dan pembuatan laporan |
| G6401221122 | Muhammad Firza Gyandra Sukma | <i>Product testing</i> |
| G6401221127 | Rizky Rasyid Wirakusuma | <i>Front-end developer</i> , pembuatan dan pengeditan video demonstrasi, pembuatan <i>deck</i> presentasi |

Timeline Pengerjaan Proyek

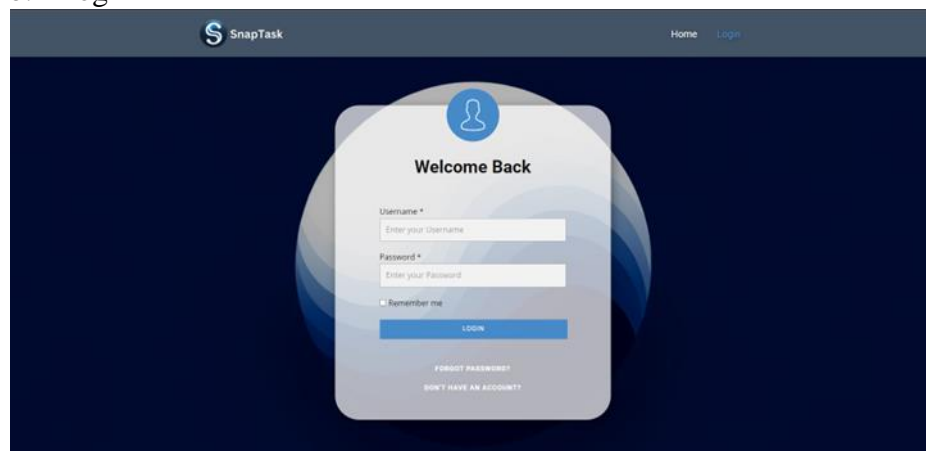
Keterangan: (1) FL: Firoos Likan, (2) RB: Rusydi Balfas, (3) RR: Rizky Rasyid, (4) FG: Firza Gyandra

Tampilan Website

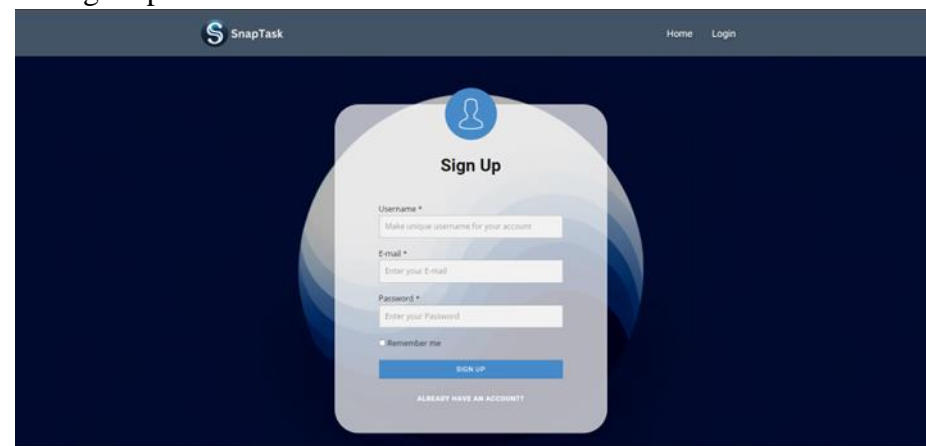
- a. Homepage



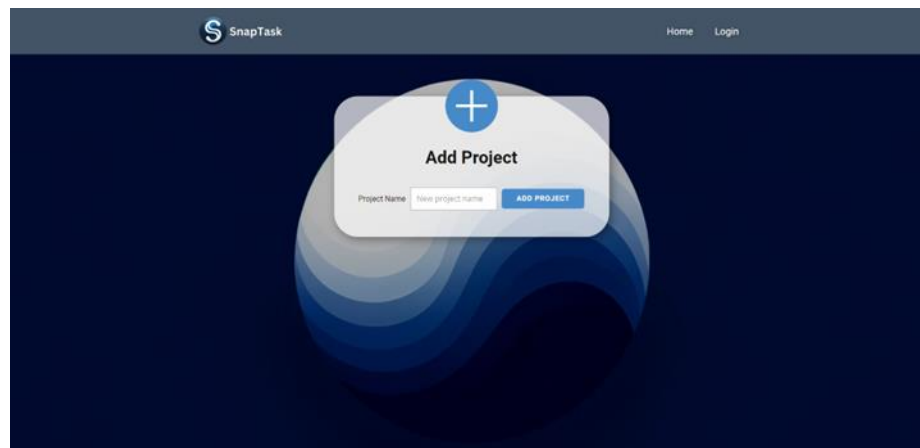
b. Log in



c. Sign Up



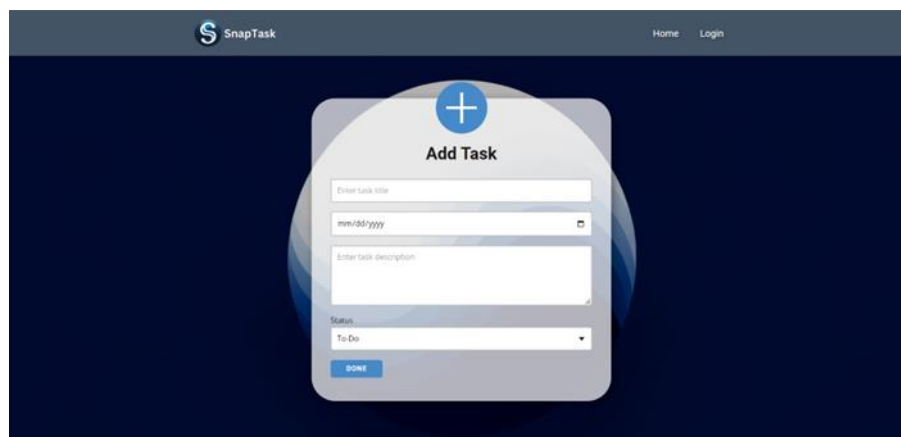
d. Form menambahkan Project



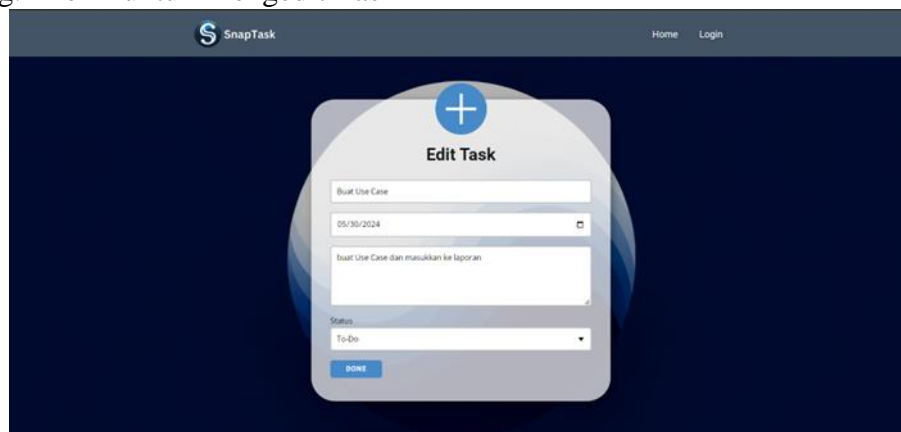
e. Detail Project dan Task Task nya



f. Form untuk Menambahkan Task



g. Form untuk Mengedit Task



Testing & QA

| Modul | Features | Test Steps | Expected Result | Status | Note |
|----------|--------------------------------|---|---|--------|------|
| SnapTask | Form untuk menambahkan project | Pada halaman Homepage, User klik add project | User diarahkan pada halaman Add Project | PASS | |
| | | | Pada halaman add project terdapat kolom pengisian nama project baru | PASS | |
| | | | Pada halaman add project terdapat button add project | PASS | |
| | | | User dapat mengisi kolom nama project baru | PASS | |
| | | | User dapat menambahkan project | PASS | |
| | | | User diarahkan pada halaman Project and Task | PASS | |
| | Form untuk menambahkan task | Pada halaman Project and Task, User klik add task | User diarahkan pada halaman add task | PASS | |
| | | | Pada halaman add task terdapat kolom pengisian judul task, set tanggal, deskripsi task, dan status task | PASS | |
| | | | Pada halaman add task terdapat button done | PASS | |
| | | | User dapat mengisi kolom judul task, set tanggal, deskripsi task, dan memilih status task | PASS | |
| | | | User dapat menambahkan task | PASS | |
| | | | User diarahkan pada halaman Project and Task | PASS | |
| | Form untuk mengedit task | Pada halaman Project and Task, User klik button pencil yang terdapat di sebelah kanan task yang telah dibuat sebelumnya | User diarahkan pada halaman edit task | PASS | |
| | | | Pada halaman edit task terdapat button done | PASS | |
| | | | Pada halaman edit task terdapat kolom buat use case, set tanggal, kolom deskripsi use case, dan memilih status task | PASS | |
| | | | User dapat mengedit task | PASS | |
| | | | User diarahkan pada halaman Project and Task | PASS | |

SIMPULAN

Setelah melalui serangkaian proses pengembangan yang cukup panjang, pengembang berhasil mengembangkan Snaptask, sebuah aplikasi web yang memudahkan manajer proyek dalam manajemen proyek. Dengan fitur seperti QuickTasks untuk pengelolaan tugas dan Project Folder untuk pengorganisasian proyek, SnapTask menawarkan solusi efisien dan inovatif. Aplikasi ini dirancang agar mudah digunakan dan mendukung produktivitas generasi muda yang aktif dalam berbagai kegiatan. Pengembang berharap Snaptask dapat membantu manajer proyek dalam mengelola proyek mereka secara lebih efektif dan meningkatkan kolaborasi, meskipun pengguna tetap perlu mengembangkan keterampilan manajemen proyek secara mandiri.

UCAPAN TERIMA KASIH

Pengembang ingin menyampaikan rasa terima kasih yang mendalam kepada semua pihak yang telah mendukung dan berkontribusi dalam pengembangan website ini. Apresiasi khusus untuk seluruh anggota tim yang bekerja keras dalam merancang, mengimplementasikan, dan menguji fitur-fitur website, serta menghasilkan solusi inovatif dan relevan. Pengembang juga berterima kasih kepada asisten praktikum yang memberikan dukungan moral dan motivasi yang menjadi pendorong pengembang untuk menyelesaikan proyek ini. Terakhir, pengembang mengucapkan terima kasih kepada dosen-dosen pengampu Mata Kuliah RPL yang menyediakan fasilitas dan sumber daya yang sangat mendukung proses pengembangan website ini.

DAFTAR PUSTAKA

- Kerzner H. 2003. *System Approach to Planning, Scheduling, and Controlling*. Edisi ke-8. United States of America: John Wiley & Sons, Inc.
- Pressman R, Maxim B. 2014. *Software Engineering: A Practitioner's Approach*. Edisi ke-8. New York: McGraw-Hill Education.
- Mahmoudi A, Feylizadeh M. 2017. A mathematical model for crashing projects by considering time, cost, quality, and risk. *Journal of Project Management*. 2:27-36.
- Vidianto A, Haji W. 2020. Sistem informasi manajemen proyek berbasis kanban (studi kasus: PT. XYZ). *Jurnal Teknologi Informasi dan Ilmu Komputer*. 7(2):282-292. doi: 10.25126/jtiik.202071676.
- Kale S, Shewale A, Sarang P, Pawar P, Sadruddin S. 2017. Project Management System. 5(2).
- Khoirunnisa, Refika, Isnanto R, Martono KT. 2016. Pembuatan aplikasi web manajemen laundry dan integrasi data dengan web service. *Jurnal Teknologi dan Sistem Komputer*, 4(1):93-101. doi:10.14710/jtsiskom.4.1.2016.93-101.
- Shidqi M, Riqky MA. 2021. Pengembangan aplikasi dan website manajemen proyek PT. Santai Berkualitas Syberindo menggunakan metode agile. *SEMINASTIKA*. 3(1):8-15. <https://doi.org/10.47002/seminastika.v3i1.249>