**APLIKASI MANAJEMEN PROYEK BERBASIS *WEB* PADA PT MILLENNIA SOLUSI INFORMATIKA**

**LAPORAN KERJA PRAKTIK**

Diajukan Sebagai Salah Satu Syarat

Untuk Memenuhi Mata Kuliah Kerja Praktik

**Strata 1**

**Disusun Oleh:**

**Muhammad Faza Yujahid**

**9882405120121022**



**PROGRAM STUDI SISTEM INFORMASI**

**FAKULTAS TEKNOLOGI DAN INFORMATIKA**

**UNIVERSITAS INFORMATIKA DAN BISNIS INDONESIA**

**BANDUNG**

**2023**

# LEMBAR PENGESAHAAN

|  |  |  |
| --- | --- | --- |
| Nama | : | Muhammad Faza Yujahid |
| NPM | : | 9882405120121022 |
| Program Studi | : | Sistem Informasi |
| Judul | : | Aplikasi Manajemen Proyek Berbasis *Web* Pada PT Millennia Solusi Informatika |

Disetujui dan Disahkan oleh:

|  |  |  |
| --- | --- | --- |
|  |  | Bandung, 2023 |
| Pembimbing  (Chairul Habibi, M.Kom)  NIDN : 0420088705 |  | Mengetahui,  Ketua Program Studi  Sistem Informasi  (Titan Parama Yoga, S.Kom., M.Kom)  NIDN : 0422047703 |

# KATA PENGANTAR

Puji serta syukur kehadirat Allah SWT yang telah melimpahkan segala Rahmat dan karunia-nya, sehingga penulis menyelesaikan Laporan Kerja Praktik yang berjudul “Aplikasi Manajemen Proyek Berbasis *Web* Pada PT Millennia Solusi Informatika”, tujuan dari penulisan ini adalah untuk memenuhi persyaratan dalam menyelesaikan Kerja Praktik Program Studi Sistem Informasi Universitas Informatika dan Bisnis Indonesia.

Dengan segala keterbatasan penulis menyadari bahwa Laporan Kerja Praktik ini tidak akan terwujud tanpa bantuan, bimbingan dan dorongan dari berbagai pihak, untuk itu dengan segala kerendahan hati penulis menyampaikan ucapan terima kasih yang sebesar - besarnya kepada :

1. Orang tua tercinta yang selalu tiada henti memberikan doa dan dukungan baik moral maupun material kepada penulis untuk dapat mencapai hasil yang terbaik.
2. Chairul Habibi, M.Kom sebagai dosen pembimbing yang telah banyak membantu menyelesaikan laporan ini.
3. Budiman, S.T., M.Kom selaku Dekan Fakultas Teknologi dan Informatika.
4. Titan Parama Yoga, S.Kom., M.Kom selaku Ketua Program Studi Sistem Informasi.
5. Chandra Kurniawan Sebagai pembimbing lapangan Kerja Praktik di perusahaan PT Millennia Solusi Informatika.
6. Teman-teman yang membantu memberi masukan dan dukungan untuk menyelesaikan Laporan Kerja Praktik ini.
7. Dan kepada semua pihak yang telah membantu penulis dalam menyelesaikan Laporan Kerja Praktik ini yang tidak bisa disebutkan satu persatu.

Penulis menyadari bahwa penulisan Laporan Kerja Praktik ini masih jauh dari kata sempurna, untuk itu penulis menerima kritik dan saran yang sifatnya membangun akan senantiasa penulis terima dengan senang hati sebagai bahan evaluasi penulis dikemudian hari.

Akhir kata semoga Laporan Kerja Praktik ini dapat digunakan sebagaimana mestinya serta berguna bagi penulis khususnya dan bagi para pembaca yang berminat pada umumnya.

|  |
| --- |
| Bandung, 2023 |
| Penulis, |
| ( Muhammad Faza Yujahid ) |

# DAFTAR ISI

[LEMBAR PENGESAHAAN ii](#_Toc141704209)

[KATA PENGANTAR iii](#_Toc141704210)

[DAFTAR ISI v](#_Toc141704211)

[DAFTAR GAMBAR vii](#_Toc141704212)

[DAFTAR TABEL ix](#_Toc141704213)

BAB I [PENDAHULUAN 10](#_Toc141704215)

[1.1 Latar Belakang 10](#_Toc141704216)

[1.2 Ruang Lingkup Kerja Praktek 11](#_Toc141704217)

[1.3 Tujuan dan Manfaat 11](#_Toc141704218)

[A. Tujuan 11](#_Toc141704219)

[B. Manfaat 12](#_Toc141704220)

[1.4 Sistematika Penulisan 12](#_Toc141704221)

BAB II [GAMBARAN UMUM PERUSAHAAN 14](#_Toc141704223)

[2.1 Sejarah Perusahaan 14](#_Toc141704224)

[2.2 Struktur Organisasi 14](#_Toc141704225)

[2.1 Visi dan Misi Perusahaan 15](#_Toc141704226)

[A. Visi 15](#_Toc141704227)

[B. Misi 15](#_Toc141704228)

[2.3 Logo Perusahaan 15](#_Toc141704229)

[2.1 Budaya Kerja dan Filosofi Perusahaan 15](#_Toc141704230)

BAB III [HASIL KEGIATAN UJIAN PRAKTIK 17](#_Toc141704232)

[3.1 Waktu Pelaksanaan Kerja Praktik 17](#_Toc141704233)

[3.2 Bidang Kegiatan Kerja Praktik 17](#_Toc141704234)

[3.3 Sistem dan Prosedur Kerja 18](#_Toc141704235)

[3.4 Tugas dan Tanggung Jawab 18](#_Toc141704236)

[3.5 Landasan Teori 18](#_Toc141704237)

[3.5.1. Aplikasi 18](#_Toc141704238)

[3.5.2. Manajemen Proyek 19](#_Toc141704239)

[*3.5.3.* *Website* 19](#_Toc141704240)

[*3.5.4.* *Framework* 20](#_Toc141704241)

[*3.5.5.* *NextJS* 20](#_Toc141704242)

[*3.5.6.* *ExpressJS* 21](#_Toc141704243)

[3.5.7. Basis Data 21](#_Toc141704244)

[*3.5.8.* *PostgreSQL* 22](#_Toc141704245)

[*3.5.9.* Diagram Alur(*Flowchart)* 23](#_Toc141704246)

[*3.5.10.* *Application Programming Interface(API)* 24](#_Toc141704247)

[3.6 Implementasi Sistem 24](#_Toc141704248)

[*3.6.1.* Diagram Alur*(flowchart)* Sistem 24](#_Toc141704249)

[3.6.2. *Screenshoot* Aplikasi 35](#_Toc141704250)

[*3.6.3.* Pengujian Aplikasi dengan Metode *Black Box* 41](#_Toc141704251)

BAB IV [KESIMPULAN DAN SARAN 56](#_Toc141704253)

[4.1 Kesimpulan 56](#_Toc141704254)

[4.2 Saran 56](#_Toc141704255)

[DAFTAR PUSTAKA 57](#_Toc141704256)

# DAFTAR GAMBAR

[Gambar 2. 1 Struktur Organisasi Perusahaan 14](#_Toc141703759)

[Gambar 2. 2 Logo Perusahaan 15](#_Toc141703760)

[Gambar 3.5.9. Komponen Diagram Alur(*Flowchart*) 23](#_Toc141704039)

[Gambar 3.6.1. 1. Diagram Alur(*Flowchart*) *Login* 24](#_Toc141703738)

[Gambar 3.6.1. 2. Diagram Alur(*Flowchart*) *register* 25](#_Toc141703739)

[Gambar 3.6.1. 3. Diagram Alur(*Flowchart*) *Invite* *People* 25](#_Toc141703740)

[Gambar 3.6.1. 4. Diagram Alur(*Flowchart*) *Create* *Workspace* 26](#_Toc141703741)

[Gambar 3.6.1. 5. Diagram Alur(*Flowchart*) *Create* *Project* 26](#_Toc141703742)

[Gambar 3.6.1. 6. Diagram Alur(*Flowchart*) *Create* *Sprint* 27](#_Toc141703743)

[Gambar 3.6.1. 7. Diagram Alur(*Flowchart*) *Edit* *Sprint* 27](#_Toc141703744)

[Gambar 3.6.1. 8. Diagram Alur(*Flowchart*) *Start* *Sprint* 28](#_Toc141703745)

[Gambar 3.6.1. 9. Diagram Alur(*Flowchart*) *Create* *Task* 28](#_Toc141703746)

[Gambar 3.6.1. 10. Diagram Alur(*Flowchart*) *Change* *Task* *Sprint* 29](#_Toc141703747)

[Gambar 3.6.1. 11. Diagram Alur(*Flowchart*) *Attach* *File* 29](#_Toc141703748)

[Gambar 3.6.1. 12. Diagram Alur(*Flowchart*) *Add* *Issue* 30](#_Toc141703749)

[Gambar 3.6.1. 13. Diagram Alur(*Flowchart*) *Edit* *Task* *Description* 30](#_Toc141703750)

[Gambar 3.6.1. 14. Diagram Alur(*Flowchart*) tambah atau ubah *Task* *Parent* 31](#_Toc141703751)

[Gambar 3.6.1. 15. Diagram Alur(*Flowchart*) Ubah *Detail* *Task* 31](#_Toc141703752)

[Gambar 3.6.1. 16. Diagram Alur(*Flowchart*) *Create* *Task* *Status* 32](#_Toc141703753)

[Gambar 3.6.1. 17. Diagram Alur(*Flowchart*) ubah *Task* *Status* dengan *Kanban* 32](#_Toc141703754)

[Gambar 3.6.1. 18. Diagram Alur(*Flowchart*) *Complete* *Sprint* 33](#_Toc141703755)

[Gambar 3.6.1. 19. Diagram Alur(*Flowchart*) *Project* Delete 33](#_Toc141703756)

[Gambar 3.6.1. 20. Diagram Alur(*Flowchart*) ubah *Project* *Name* 34](#_Toc141703757)

[Gambar 3.6.1. 21. Diagram Alur(*Flowchart*) *Remove* *Project* *Handler* 34](#_Toc141703758)

[Gambar 3.6.2. 1. *Screenshoot* *Login* 35](#_Toc141704083)

[Gambar 3.6.2. 2. *Screenshoot* *Register* 35](#_Toc141704084)

[Gambar 3.6.2. 3. *Screenshoot* *Setup* *Account* 36](#_Toc141704085)

[Gambar 3.6.2. 4. *Screenshoot* *Dashboard* 36](#_Toc141704086)

[Gambar 3.6.2. 5. *Screenshoot* Halaman *Workspace* 37](#_Toc141704087)

[Gambar 3.6.2. 6. *Screenshoot* Halaman *Member* *Report* 37](#_Toc141704088)

[Gambar 3.6.2. 7. *Screenshoot* Halaman *Team* 38](#_Toc141704089)

[Gambar 3.6.2. 8. *Screenshoot* Halaman *Member* 38](#_Toc141704090)

[Gambar 3.6.2. 9. *Screenshoot* Halaman *Backlog* *Project* 39](#_Toc141704091)

[Gambar 3.6.2. 10. *Screenshoot* Halaman *Kanban* *Board* *Project* 39](#_Toc141704092)

[Gambar 3.6.2. 11. *Screenshoot* Halaman *Roadmap* *Project* 40](#_Toc141704093)

[Gambar 3.6.2. 12. *Screenshoot* Halaman *Task* *Detail* 40](#_Toc141704094)

# DAFTAR TABEL

[Tabel 3.6.3. 1. Pengujian autentikasi akun 41](#_Toc141704142)

[Tabel 3.6.3. 2 Pengujian *Register* *User* 41](#_Toc141704143)

[Tabel 3.6.3. 3 Pengujian *Dashboard* 42](#_Toc141704144)

[Tabel 3.6.3. 4 Pengujian *Workspace* 43](#_Toc141704145)

[Tabel 3.6.3. 5 Pengujian *Member* *Report* 45](#_Toc141704146)

[Tabel 3.6.3. 6 Pengujian *Teams* 45](#_Toc141704147)

[Tabel 3.6.3. 7 Pengujian *Project* 46](#_Toc141704148)

[Tabel 3.6.3. 8 Pengujian *Backlog* *Project* 47](#_Toc141704149)

[Tabel 3.6.3. 9 Pengujian *Board* *Project* 48](#_Toc141704150)

[Tabel 3.6.3. 10 Pengujian *Roadmap* *Project* 49](#_Toc141704151)

[Tabel 3.6.3. 11 Pengujian *Task* *Detail* 50](#_Toc141704152)

[Tabel 3.6.3. 12 Pengujian *Attachment* *File* 53](#_Toc141704153)

[Tabel 3.6.3. 13 Pengujian *Child* *Task* 53](#_Toc141704154)

[Tabel 3.6.3. 14 Pengujian *Task* *Issue* 54](#_Toc141704155)

[Tabel 3.6.3. 15 Pengujian *Task Activity* 54](#_Toc141704156)

# BAB I

# PENDAHULUAN

## **Latar Belakang**

Manajemen proyek adalah penerapan pengetahuan, keterampilan, alat, dan teknik untuk kegiatan proyek untuk memenuhi persyaratan proyek. Manajemen proyek dicapai melalui penerapan dan integrasi yang tepat dari proses manajemen proyek yang diidentifikasi untuk proyek tersebut. Manajemen proyek memungkinkan organisasi untuk melaksanakan proyek efektif dan efisien(*Project* Management Institute, 2017). Manajemen proyek dapat diterapkan pada jenis proyek apapun, dan dipakai secara luas untuk dalam menyelesaikan proyek yang besar dan kompleks. Fokus utama manajemen proyek adalah pencapaian semua tujuan akhir proyek dengan segala batasan yang ada, waktu dan dana yang tersedia.

Manajemen proyek yang efektif dan efisien harus dianggap sebagai kompetensi strategis dalam organisasi. Ini memungkinkan organisasi untuk mengaitkan hasil proyek dengan tujuan bisnis, Bersaing secara lebih efektif di pasar, Mempertahankan organisasi, dan, Menanggapi dampak perubahan lingkungan bisnis pada proyek(*Project* Management Institute, 2017).

Seiring dengan perkembangan teknologi informasi, manusia mulai menggunakan komputer dalam melakukan manajemen proyek, untuk membantu otomatisasi dan perhitungan. Aplikasi manajemen proyek menjadi salah satu alat yang memberikan keuntungan pada perusahaan, seperti meningkatkan keuntungan bisnis, meningkatkan efisiensi operasional dalam pengawasan status dari proyek yang berjalan (Christanto & Wibowo, 2020).

PT. Millennia Solusi Informatika merupakan Perusahaan yang berfokus pada pengembangan aplikasi untuk berbagai kebutuhan bisnis. PT Millennia Solusi Informatika telah berpengalaman dalam pengembangan aplikasi dengan tim pengembang yang terampil dan berdedikasi dalam menghasilkan produk yang berkualitas. Namun, seiring dengan meningkatnya kompleksitas proyek yang ditangani dan kebutuhan untuk berkolaborasi dengan tim yang terlibat dalam proyek, PT. Millennia Solusi Informatika menyadari diperlukannya suatu aplikasi manajemen proyek yang dapat membantu mereka dalam menghadapi tantangan tersebut dan meningkatkan efisiensi serta efektivitas dalam mengelola proyek – proyek mereka.

Aplikasi manajemen proyek berbasi *web* dapat menjadi solusi yang tepat bagi PT Millennia Solusi Informatika. Dengan mengimplementasikan aplikasian manajemen berbasis *web,* Perusahaan dapat memastikan kolaborasi dan aksesibilitas antar anggota tim dapat berjalan secara optimal.

Berdasarkan latar belakang masalah diatas maka penulis mengambil topik judul laporan “APLIKASI MANAJEMEN PROYEK BERBASIS WEB PADA PT MILLENNIA SOLUSI INFORMATIKA”

## **Ruang Lingkup Kerja Praktek**

Berdasarkan latar belakang masalah yang telah dijelaskan diatas, maka dapat ruang lingkup masalah pada penelitian ini adalah sebagai berikut:

1. Proses pembuatan aplikasi dilakukan sesuai dengan permintaan dan kebutuhan pihak Perusahaan.
2. Membangun aplikasi manajemen proyek berbasis *web* menggunakan *PostgreSQL* sebagai basis data.
3. Fitur yang diperlukan dalam aplikasi manajemen proyek berbasis *web*, untuk memenuhi kebutuhan PT Millennia Solusi Informatika dalam mengelola proyek-proyeknya secara efisien.

## **Tujuan dan Manfaat**

### **Tujuan**

Maksud dan tujuan dari Kerja Praktik (KP) ini adalah sebagai berikut:

1. Mengaplikasikan teori yang diperoleh di perkuliahan dan mengetahui kondisi kenyataan yang ada di lapangan.
2. Menambah pemahaman, memperluas pengetahuan, mengasah keterampilan, serta melatih diri untuk menjadi tenaga kerja yang profesional sehingga dapat mempersiapkan diri secara mental untuk bersaing di dunia kerja.
3. Merancang aplikasi manajemen proyek untuk perusahaan PT Millennia Solusi Informatika.
4. Membantu perusahaan PT Millennia Solusi Informatika dalam mengelola *project* yang diberikan *client* kepada perusahaan.

### **Manfaat**

Manfaat yang didapat Ketika melaksanakan kerja praktik ini antara lain:

1. Bagi mahasiswa, dapat mengetahui bagaimana menerapkan dan meningkatkan pengetahuan yang diperoleh di bangku perkuliahan dan dapat menambah pemahaman mengenai dunia industri.
2. Bagi universitas, Dapat menjadi sarana untuk menjalin kerja sama dengan pihak PT millennia Solusi Informatika.
3. Bagi Perusahaan, dapat membantu menyelesaikan permasalahan yang dihadapi melalui aplikasi yang telah dibangun.

## **Sistematika Penulisan**

Laporan ini disusun secara sistematis berdasarkan tata tulis laporan yang di tetapkan oleh Departemen Sistem Informasi Universitas Informatika dan Bisnis Indonesia dengan urutan sebagai berikut:

**BAB I PENDAHULUAN**

Bab ini menjelaskan tentang latar belakang masalah, dari permasalahan disebutkan pada perumusan masalah, pembatasan masalah yang menjelaskan batasan - batasan dari perancangan yang dibuat sehingga tidak menyimpang dari ketentuan yang telah diterapkan, dan tujuan praktik adalah merancang aplikasi program, kemudian dilanjutkan dengan sistematika penulisan kerja praktik.

**BAB II GAMBARAN UMUM PERUSAHAAN**

Bab ini menjelaskan tentang sejarah perusahaan, struktur organisasi, visi dan misi, budaya kerja dan filosofi, dan logo perusahaan.

**BAB III HASIL KEGIATAN KERJA PRAKTIK**

Bab ini membahas hasil kerja dari kerja praktik, mulai dari alur pada aplikasi yang dibangun sampai dengan pengujian aplikasi.

**BAB IV KESIMPULAN DAN SARAN**

Bab ini berisi kesimpulan dari rumusan masalah yang telah dibuat dan saran untuk perbaikan sistem kedepannya.

# BAB II

# GAMBARAN UMUM PERUSAHAAN

## **Sejarah Perusahaan**

PT Millennia Solusi Informatika, merupakan perusahaan IT *Services* dan IT *Consulting*, yang berspesialisasi di bidang *Software* *Development* *and* IT *Consultant*. Secara resmi PT Millennia Solusi Informatika berdiri sejak tahun 2019 yang dipimpin oleh bapak Umar Alhabsyi, beserta karyawan yang telah bekerja di perusahaan tersebut.

Untuk mendapatkan manfaat dari era milenial ini, sebuah organisasi akan dituntut untuk memilih solusi serta pendekatan tata kelola dan manajemen yang tepat, karena dengan hal tersebut organisasi dapat menyelesaikan akar permasalahan yang sedang dihadapi. Hal tersebut dapat terwujud dengan pendekatan tata kelola dan manajemen yang komprehensif, maka dari itu PT Millennia Solusi Informatika hadir sebagai penyedia layanan IT *Management* dan IT *Development* yang terpercaya.

## **Struktur Organisasi**

(sumber: Perusahaan)

Gambar 2. 1 Struktur Organisasi Perusahaan

## **Visi dan Misi Perusahaan**

### **Visi**

Menjadi perusahaan IT yang unggul, andal, dan kompetitif.

### **Misi**

1. Memberikan pelayanan terbaik dan solusi total dengan fokus pada kepuasan pelanggan.
2. Bersinergi dalam mengasah pengalaman dengan terus menerus meningkatkan kualitas pelayanan.
3. Meningkatkan kinerja dan profitabilitas melalui efisiensi dan profesionalisme guna menjaga komitmen kami sebagai perusahaan yang sehat dan terpercaya secara berkelanjutan.

## **Logo Perusahaan**

Logo adalah sebuah tulisan, sketsa, atau gambar yang mempunyai makna tertentu dan bisa mewakili identitas atas suatu bentuk entitas, seperti lembaga, organisasi, perusahaan, daerah, negara, atau produk. PT Millennia Solusi Informatika memiliki logo sebagai berikut :



(Sumber: Perusahaan)

Gambar 2. 2 Logo Perusahaan

## **Budaya Kerja dan Filosofi Perusahaan**

PT Millennia Solusi Informatika mengutamakan kualitas dalam setiap aspek pekerjaan. Kami berkomitmen untuk memberikan produk dan layanan yang berkualitas tinggi, yang ditopang oleh standar industri serta praktik terbaik. PT Millennia Solusi Informatika mendorong karyawan untuk berusaha mencapai keunggulan dalam pekerjaan mereka dan terus meningkatkan kualitas produk dan layanan kami melalui inovasi dan pembelajaran berkelanjutan.

PT Millennia Solusi Informatika menyediakan program pelatihan dan pengembangan yang komprehensif untuk membantu karyawannya dalam meningkatkan keterampilan, memperluas pengetahuan, dan mencapai potensi maksimal. Perusahaan juga mendorong karyawan untuk mengambil inisiatif dalam mengembangkan karier, serta memberikan kesempatan untuk pertumbuhan dan kemajuan profesional.

PT Millennia Solusi Informatika mendorong lingkungan kerja yang inklusif, di mana setiap individu dihormati, dihargai, dan diberdayakan. Kami menghargai keragaman latar belakang, ide, dan perspektif, dan kami percaya bahwa kolaborasi antara individu dengan pemikiran yang berbeda dapat menghasilkan hasil yang luar biasa. Kami mengupayakan keadilan, kesetaraan, dan non-diskriminasi dalam segala aspek pengelolaan sumber daya manusia.

PT Millennia Solusi Informatika menjunjung tinggi etika dan integritas dalam setiap interaksi bisnis. Kami berkomitmen untuk bertindak dengan jujur, transparan, dan adil dalam hubungan dengan karyawan, pelanggan, mitra bisnis, dan seluruh pemangku kepentingan. Kami menghargai kerjasama yang saling menguntungkan dan menjaga standar etika yang tinggi dalam semua keputusan dan tindakan yang kami ambil.

# BAB III

# HASIL KEGIATAN UJIAN PRAKTIK

## **Waktu Pelaksanaan Kerja Praktik**

Penulis melaksanakan kerja praktik dalam jangka waktu 3 (tiga) bulan, yang dimulai pada tanggal 01 Maret 2023 sampai dengan 31 Mei 2023. Kerja praktik dilaksanakan secara WFH(*Work From Home*), dimana penulis melaporkan progress setiap 2 kali dalam 1 bulan ke PT Millennia Solusi Informatika yang berlokasi di Metro Trade Center, Jl. Soekarno Hatta No.590, Sekejati, Kec. Buahbatu, Kota Bandung, Jawa Barat 40286.

## **Bidang Kegiatan Kerja Praktik**

PT Millennia Solusi Informatika memberikan tanggung jawab dalam beberapa posisi untuk peserta kerja praktik, yaitu perancangan, *frontend* *development* dan *backend* *development.* Pihak perusahaan memberikan tugas spesifik kepada peserta sesuai dengan peran dan fungsi dari masing - masing posisi yang diberikan.

Untuk posisi perancangan, diberikan tugas untuk membuat tampilan antarmuka (UI) di *platform* dengan menggunakan Figma, sebagai gambaran untuk nantinya diterapkan oleh tim *frontend* *development.* Selain itu, peserta kerja paraktik pada posisi perancangan juga bertugas merancang desain *database* dengan PostgreSQL sebagai *Relational Database Management System* (RDBMS).

Untuk posisi *frontend developer*, diberikan tugas untuk mengimplementasikan tampilan yang telah dirancang menggunakan Figma oleh tim perancangan. Dalam *frontend development*, *framework* *Next*.js digunakan untuk mengembangkan aplikasi dari sisi tampilan.

Untuk posisi *Backend developer* bertugas dalam pengembangan *backend* menggunakan metode penerapan API dengan *framework* *Express*.js.

## **Sistem dan Prosedur Kerja**

Pada PT Millennia Solusi Informatika, sistem dan prosedur kerja dilakukan secara *hybrid,* dimana peserta kerja praktik membuat tim yang beranggotakan dua orang, yang masing masing berperan dalam proses perancangan, *frontend development,* dan *backend development.* Kerja praktik ini dilaksanakan dalam kurun waktu 3 bulan, dengan sistem kerja *work from home*, dengan laporan progres pengerjaan yang dilakukan secara rutin setiap 2 kali dalam satu bulan secara langsung ke perusahaan.

## **Tugas dan Tanggung Jawab**

Selama kegiatan kerja praktik di perusahaan PT Millennia Solusi Informatika penulis diberikan tugas untuk membangun aplikasi manajemen proyek berbasis *web* berdasarkan perancangan yang telah dibuat sebelumnya. proses bangun aplikasi dibagi menjadi dua tahap, pertama yaitu pembuatan tampilan *frontend* dengan menggunakan *NextJS* dan pembuatan *backend* dengan menggunakan *ExpressJS.*

Hasil dari tampilan *frontend* dan *backend* tersebut selanjutnya dihubungkan dengan menerapkan metode API(*Application Programming Interface*).

## **Landasan Teori**

Salah satu dasar teori yang digunakan sebagai acuan dalam Aplikasi Manajemen Proyek Berbasis *Web* di PT Millennia Solusi Informatika mencakup teori dasar *Application Programming Interface*(API).

### **Aplikasi**

Secara istilah pengertian aplikasi adalah suatu program yang siap digunakan untuk melaksanakan suatu fungsi bagi pengguna yang dapat digunakan oleh suatu sasaran yang akan dituju (Juansyah, 2015).

Menurut Jogiyanto(2013) aplikasi merupakan penerapan, menyimpan sesuatu hal, data, permasalahan, pekerjaan kedalam suatu sarana atau media yang dapat digunakan untuk menerapkan atau mengmplementasikan hal atau permasalahan yang ada sehingga berubah menjadi suatu bentuk yang baru tanpa menghilangkan nilai-nilai dasar dari hal data, permasalahan, dan pekerjaan itu sendiri.

Berdasarkan dua penjelasan tersebut bisa disimpulkan bahwa aplikasi adalah suatu program yang digunakan untuk menyimpan sesuatu hal kedalam suatu sarana yang dapat digunakan untuk menerapkan suatu hal atau permasalahan.

### **Manajemen Proyek**

Manajemen proyek adalah suatu strategi yang perlu dilakukan untuk mencapai efisiensi dan efektivitas dalam suatu perusahaan. Perencanaan manajemen proyek dilakukan untuk menjalankan proyek sebaik mungkin. Tujuan dari manajemen proyek adalah meningkatkan keuntungan perusahaan, namun kebutuhan yang meningkat dalam proyek dapat menjadi masalah jika perusahaan tidak memiliki manajemen yang baik. Manajemen proyek diciptakan untuk menghindari atau mengurangi kegagalan dan risiko proyek. Manajemen yang baik melibatkan aktivitas seperti penjadwalan, pengelolaan sumber daya manusia, dan estimasi biaya proyek yang harus dianggarkan oleh perusahaan. (K. A. Wibowo & Sulistyono, 2017)

### ***Website***

Menurut Sebok, Vermat, dan tim (2018 : 70), *website* adalah sekumpulan halaman yang saling berhubungan yang di dalamnya terdapat beberapa item seperti dokumen dan gambar yang tersimpan di dalam *web server.*

Sedangkan menurut Sholechul Azis, (2013)” *Website* adalah halaman informasi yang disediakan melalui jalur internet sehingga bisa diakses di seluruh dunia selama terkoneksi dengan jaringan internet. *Website* merupakan komponen atau kumpulan komponen yang terdiri dari teks, gambar, suara, dan animasi sehingga menarik untuk dikunjungi”.

Berdasarkan penjelasan – penjelasan tersebut, bisa disimpulkan bahwa *website* adalah suatu sistem yang saling berhubungan yang di dalamnya terdapat berbagai item yang tersimpan dalam *web server*.

### ***Framework***

Menurut Naista(2017), *Framework* adalah suatu struktur konseptual dasar yang digunakan untuk memecahkan atau menangani suatu masalah yang kompleks. Singkatnya, *framework* adalah wadah atau kerangka kerja dari sebuah website yang akan dibangun. Dengan menggunakan kerangka tersebut waktu yang digunakan dalam membuat website lebih singkat dan memudahkan dalam melakukan perbaikan.

Banyak keuntungan yang didapatkan jika menggunakan framework dalam membangun sebuah website. Salah satu keuntungan tersebut adalah memberikan struktur yang baik dalam program yang dibuat karena framework memiliki library atau fungsi yang bisa langsung digunakan. Selain itu, framework memermudah dalam pengerjaan program secara tim karena dalam membangun website harus melakukan penyesuaian dengan gaya framework yang dipakai.

### ***NextJS***

*NextJS* merupakan framework untuk membuat aplikasi *website* menggunakan bahasa pemrograman *JavaScript*. *NextJS* memberikan banyak manfaat bagi pengembang aplikasi web, termasuk kemampuan untuk membuat tampilan *(UI)* yang responsif dan *SEO-friendly* dengan cepat, mengelola *state* aplikasi dengan mudah, serta mengoptimalkan kinerja aplikasi web secara otomatis. Selain itu, *NextIS* s juga memiliki dukungan untuk *Server-side Rendering (SSR)* dan *Static Site Generation (SSG*), yang memungkinkan aplikasi web untuk dihasilkan dan diakses dengan lebih cepat.

### ***ExpressJS***

*ExpressJS* adalah framework web app untuk *Node.js* yang ditulis dengan bahasa pemrograman *JavaScript*. Express.js adalah framework *back end*. Artinya, ia bertanggung jawab untuk mengatur fungsionalitas website, seperti pengelolaan *routing* dan *session*, permintaan *HTTP*, penanganan *error*, serta pertukaran *data* di *server*.

Berbeda dengan kebanyakan *framework* yang punya aturan ketat mengenai penulisan kode, ExpressJS adalah *framework* yang menggunakan pendekatan *Unopinionated* dalam proses pengembangan. Artinya, pengguna punya kebebasan dalam menentukan metode yang akan digunakan untuk mengeksekusi suatu perintah.

### **Basis Data**

Menurut Indrajani (2015:70), “basis data adalah kumpulan data yang saling berhubungan secara logis dan didesain untuk mendapatkan data yang dibutuhkan oleh suatu organisasi”.

Menurut Jayanti & Sumiari, (2018:2), “Basis Data merupakan data yang terintegrasi, yang diorganisasi untuk memenuhi kebutuhan para pemakai di dalam suatu organisasi”. Berdasarkan diatas, dapat disimpulkan bahwa basis data merupakan data yang dapat didesain dan berintegrasi sehingga dapat memenuhi kebutuhan user dalam perusahaan atau organisasi.

Berdasarkan penjelasan – penjelasan tersebut, dapat disimpulkan bahwa basisdata adalah sekumpulan data yang terintegrasi dan terorganisir yang digunakan untuk memenuhi kebutuhan pemakai.

### ***PostgreSQL***

*PostgreSQL*, yang juga dikenal sebagai *Postgres*, adalah sistem pengelolaan basis data objek relasional yang bersifat sumber terbuka. Dalam istilah yang lebih sederhana, *PostgreSQL* adalah sistem yang digunakan untuk menyimpan dan mengambil data dengan aman dalam aplikasi perangkat lunak. Fitur tambahan yang dimiliki *PostgreSQL* memungkinkan penggunaannya sebagai sistem basis data relasional dengan kemampuan tambahan yang berorientasi pada objek. *PostgreSQL* juga mendukung lingkungan multiprosesor dengan baik. Dalam sistem pengelolaan basis data, transaksi adalah operasi logis yang dilakukan pada data tunggal. *PostgreSQL* mematuhi standar *ACID (Atomicity, Consistency, Isolation, Durability)* untuk semua transaksi basis data. Dengan demikian, *PostgreSQL* mendukung penggunaan *foreign keys, joins, views, triggers and stored procedures* tersimpan dalam berbagai bahasa pemrograman.(’Afiifah et al., 2022)

Kelebihan PostgreSQL meliputi arsitektur multiproses (*forking*) yang memberikan stabilitas yang tinggi. Jika ada proses anak yang mati, hal itu tidak akan menyebabkan seluruh proses server mati. Hal ini berbeda dengan pengalaman masa lalu yang sering terjadi. (A. T. Wibowo & Wiguna, 2019)

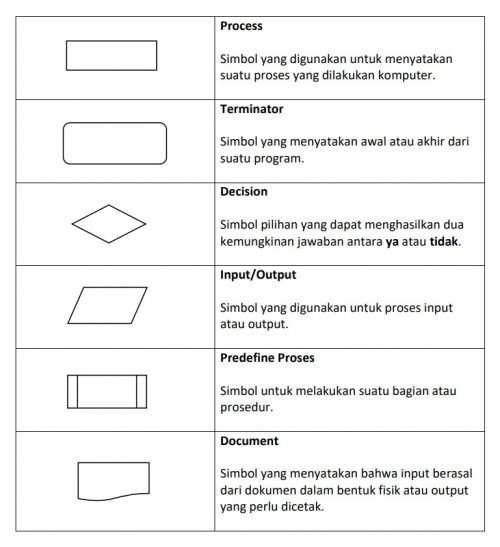
Dalam situasi beban tinggi dengan jumlah koneksi simultan yang besar, kecepatan *PostgreSQL* sering kali lebih baik daripada *MySQL* untuk *query* dengan klausa *JOIN* yang kompleks. Ini karena *PostgreSQL* mendukung penguncian (*locking*) pada level yang lebih rendah, yaitu pada tingkat baris (*row*). (A. T. Wibowo & Wiguna, 2019)

Salah satu fitur unggulan *PostgreSQL* adalah kemampuannya dalam pemrograman berorientasi objek (OO). Fitur ini mencakup pewarisan tabel dan tipe data, serta tipe data *array* yang berguna untuk menyimpan banyak item data dalam satu rekaman. Dengan adanya kemampuan OO ini, *PostgreSQL* memungkinkan pengguna untuk mendefinisikan sebuah tabel yang mewarisi definisi dari tabel lain. (A. T. Wibowo & Wiguna, 2019)

### **Diagram Alur*(Flowchart)***

Menurut Nurmalina (2017 : 86) *Flowchart* adalah representasi secara simbolik dari suatu algoritma atau prosedur untuk menyelesaikan suatu masalah, dengan menggunakan *flowchart* akan memudahkan pengguna melakukan pengecekan bagian-bagian yang terlupakan dalam analisis masalah, disamping itu *flowchart* juga berguna sebagai fasilitas untuk berkomunikasi antara pemrogram yang bekerja dalam tim suatu proyek.

*Flowchart* dapat membantu mengkomunikasikan jalannya program ke orang lain (bukan pemrogram) dengan lebih mudah.



(Sumber: dewaweb.com)

Gambar 3.5.9. Komponen Diagram Alur(Flowchart)

### ***Application Programming Interface(API)***

*Application Programming Interface (API)* adalah Sebuah bahasa dan pesan format yang digunakan oleh program aplikasi untuk berkomunikasi dengan sistem operasi atau program pengendalian lainnya seperti sistem manajemen database (*DBMS*) atau protokol komunikasi(Ida Fitriani, 2017).

*API* diimplementasikan dengan menulis panggilan fungsi dalam program, yang menyediakan hubungan ke subrutin yang diperlukan untuk eksekusi.

## **Implementasi Sistem**

### **Diagram Alur*(flowchart)* Sistem**

* + - 1. **Diagram Alur*(Flowchart)* *Login***



Gambar 3.6.1. 1. Diagram Alur(Flowchart) Login

Berdasarkan gambar 3.6.1.1., proses *login* dilakukan dengan memasukkan *email* dan *password* yang selanjutkan akan diverifikasi, jika proses verifikasi berhasil, maka akan dialihkan ke halaman *dashboard*.

* + - 1. **Diagram Alur*(Flowchart)* *Register***



Gambar 3.6.1. 2. Diagram Alur(Flowchart) register

Berdasarkan gambar 3.6.1.2, proses *register* dimulai dengan memasukkan *email* yang akan digunakan, selanjutnya akan dikirimkan *email* verifikasi yang mengarahkan pada proses *setup account*.

* + - 1. **Diagram Alur*(Flowchart)* *Invite People***



Gambar 3.6.1. 3. Diagram Alur(Flowchart) Invite People

Berdasarkan gambar 3.6.1.23, proses dilakukan dengan memasukkan *email* dari *user* yang akan dimasukkan ke dalam organisasi. Jika *email* tersebut belum terdaftar pada aplikasi, maka akan dikirimkan *invitation email*. Setelah itu, data anggota pada organisasi akan diperbaharui.

* + - 1. **Diagram Alur*(Flowchart)* *Create Workspace***



Gambar 3.6.1. 4. Diagram Alur(Flowchart) Create Workspace

Berdasarkan gambar 3.6.1.3., proses pembuatan *workspace* bisa dilakukan dengan hanya mengisi nama dari *workspace* saja. Setelah itu sistem yang akan otomatis membuat *workspace*.

* + - 1. **Diagram Alur*(Flowchart)* *Create Project***



Gambar 3.6.1. 5. Diagram Alur(Flowchart) Create Project

Berdasarkan gambar 3.6.1.5., proses *create project* dilakukan dengan mengisi nama project dan menentukan *client, workspace* dan *project handler.* Selanjutnya, sistem akan membuat project berdasarkan *input* yang diberikan oleh *user.*

* + - 1. **Diagram Alur*(Flowchart)* *Create Sprint***



Gambar 3.6.1. 6. Diagram Alur(Flowchart) Create Sprint

Berdasarkan gambar 3.6.1.6., proses *create* *sprint* bisa dilakukan cukup dengan menekan tombol *create sprint*. Selanjutnya, sistem akan membuat sprint dengan nama sementara yang nantinya bisa diubah.

* + - 1. **Diagram Alur*(Flowchart)* *Edit Sprint***



Gambar 3.6.1. 7. Diagram Alur(Flowchart) Edit Sprint

Berdasarkan gambar 3.6.1.7., proses edit *sprint* bisa dilakukan untuk mengubah nama *sprint*, menentukan waktu mulai dan waktu selesai *sprint*, juga menentukan *goals* dari *sprint* itu sendiri.

* + - 1. **Diagram Alur*(Flowchart)* *Start Sprint***



Gambar 3.6.1. 8. Diagram Alur(Flowchart) Start Sprint

Berdasarkan gambar 3.6.1.8., proses start sprint dilakukan dengan menekan tombol *start sprint* yang selanjutnya akan dialihkan ke halaman *kanban board.*

* + - 1. **Diagram Alur*(Flowchart)* *Create Task***



Gambar 3.6.1. 9. Diagram Alur(Flowchart) Create Task

Berdasarkan gambar 3.6.1.9., proses *create task* dilakukan dengan mengisikan nama *task*, menentukan *level*, *assignee* dan *reporter*, juga menentukan waktu pengerjaan paling cepat, normal dan paling lambat sebagai perhitungan untuk *story point* dengan menggunakan *pert method.*

* + - 1. **Diagram Alur*(Flowchart)* *Change Task Sprint***



Gambar 3.6.1. 10. Diagram Alur(Flowchart) Change Task Sprint

Berdasarkan gambar 3.6.1.10., proses *change task sprint* bisa dilakukan hanya dengan menarik dan lepas(*drag and drop*) task ke sprint yang diinginkan. Setelah itu sistem akan otomatis mengganti *sprint* *id* pada *task* ke *sprint* *id* yang baru.

* + - 1. **Diagram Alur*(Flowchart)* *Attach File***



Gambar 3.6.1. 11. Diagram Alur(Flowchart) Attach File

Berdasarkan gambar 3.6.1.11., proses *attach file* dilakukan dengan melakukan *upload file* yang selanjutnya akan divalidasi oleh sistem. Jika format file tidak sesuai dengan ketentuan, maka proses upload akan gagal. Namun, jika sudah sesuai maka proses *upload* akan berhasil dan *task* *activity* akan berubah.

* + - 1. **Diagram Alur*(Flowchart)* *Add Issue***



Gambar 3.6.1. 12. Diagram Alur(Flowchart) Add Issue

Berdasarkan gambar 3.6.1.12., proses *add issue* dilakukan dengan menentukan jenis isu yang dihadapi dan *task* yang memiliki isu tersebut. Selanjutnya *task* *activity* juga akan berubah.

* + - 1. **Diagram Alur*(Flowchart)* *Edit Task Description***



Gambar 3.6.1. 13. Diagram Alur(Flowchart) Edit Task Description

Berdasarkan gambar 3.6.1.13., proses *edit task description* dilakukan dengan menekan sebanyak 2 kali pada deskripsi *task.* Deskripsi tersebut selanjutnya akan berubah menjadi *textarea* yang bisa diisi dengan deskripsi yang baru. Setelah disimpan selanjutnya *task* *activity* akan berubah.

* + - 1. **Diagram Alur*(Flowchart)* Tambah atau Ubah *Task Parent***



Gambar 3.6.1. 14. Diagram Alur(Flowchart) tambah atau ubah Task Parent

Berdasarkan gambar 3.6.1.14., proses tambah atau ubah *task parent* dilakukan dengan menentukan *task* yang akan dijadikan *parent*. Seletah itu *task* *activity* akan berubah.

* + - 1. **Diagram Alur*(Flowchart)* ubah Detail *Task***



Gambar 3.6.1. 15. Diagram Alur(Flowchart) Ubah Detail Task

Berdasarkan gambar 3.6.1.15., proses ubah detail *task* dilakukan dengan memasukkan data *task* yang akan diubah yang selanjutnya akan disimpan. Setelah itu *task activity* akan berubah.

* + - 1. **Diagram Alur*(Flowchart)* *Create Task Status***



Gambar 3.6.1. 16. Diagram Alur(Flowchart) Create Task Status

Berdasarkan gambar 3.6.1.16., proses create *task* *status* dilakukan dengan memasukkan nama dari *task* baru. Selanjutnya sistem akan membuat *task status* baru berdasarkan data yang dimasukkan.

* + - 1. **Diagram Alur*(Flowchart)* Ubah *Task Status* Dengan *Kanban***



Gambar 3.6.1. 17. Diagram Alur(Flowchart) ubah Task Status dengan Kanban

Berdasarkan gambar 3.6.1.17., proses ubah *task status* dengan *kanban* dilakukan dengan menarik dan melepas(*drag and drop)* *task* di status yang diinginkan. Selanjutnya secara otomatis sistem akan memperbarui status dari *task.*

* + - 1. **Diagram Alur*(Flowchart)* *Complete Sprint***



Gambar 3.6.1. 18. Diagram Alur(Flowchart) Complete Sprint

Berdasarkan gambar 3.6.1.18., proses *complete sprint* dilakukan dengan menekan tombol *complete* yang selanjutnya akan dilakukan pengecekan terhadap jumlah sprint yang sedang berjalan. Jika *sprint* yang berjalan lebih dari 2 maka akan ditentukan *sprint* mana yang akan diselesaikan. Setelah itu akan ditentukan juga *sprint* pengalihan untuk *task* yang belum selesai.

* + - 1. **Diagram Alur*(Flowchart)* *Delete Project***



Gambar 3.6.1. 19. Diagram Alur(Flowchart) Delete Project

Berdasarkan gambar 3.6.1.19., proses *delete* project hanya bisa dilakukan oleh *superadmin* atau *admin*. Setelah dihapus, *user* akan dialihkan ke *dashboard*.

* + - 1. **Diagram Alur*(Flowchart)* Ubah *Project Name***



Gambar 3.6.1. 20. Diagram Alur(Flowchart) ubah Project Name

Berdasarkan gambar 3.6.1.20., proses ubah *project name* juga hanya bisa dilakukan oleh *admin* atau *superadmin*. Jika *user* tersebut memiliki *role* *admin* atau *superadmin* maka selanjutnya *user* hanya perlu memasukkan nama baru dari *project.*

* + - 1. **Diagram Alur*(Flowchart)* *Remove Project Handler***

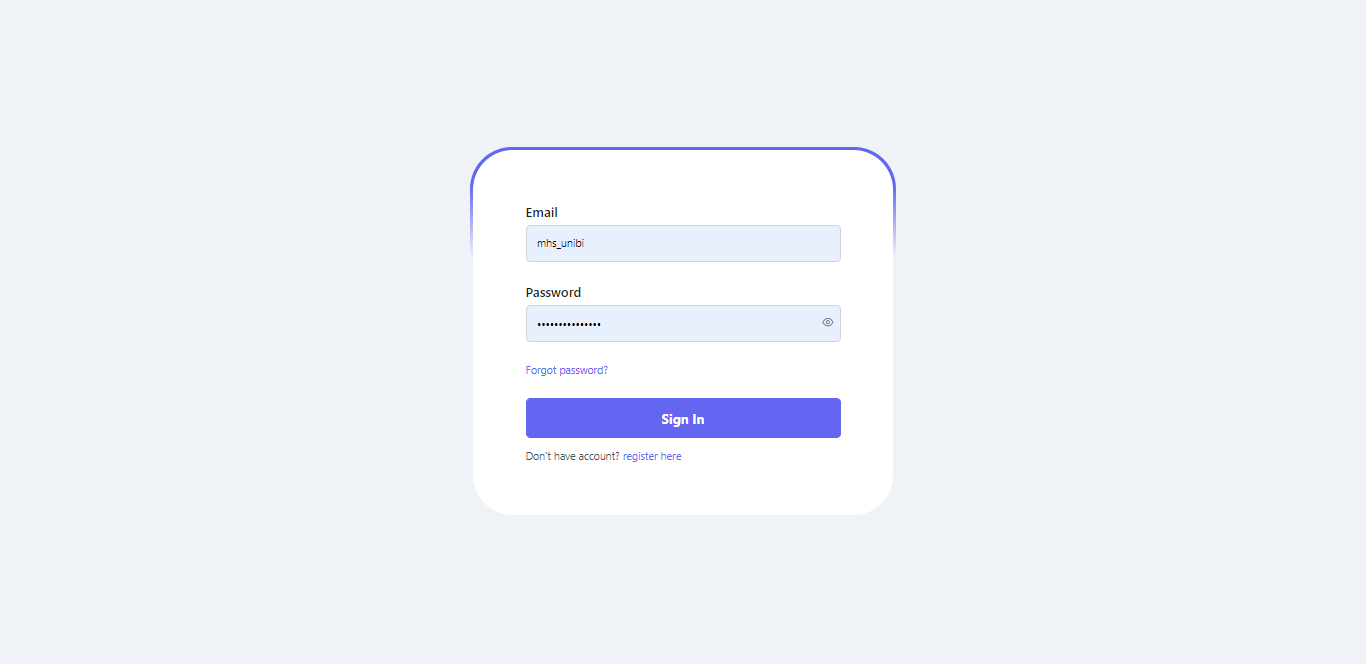


Gambar 3.6.1. 21. Diagram Alur(Flowchart) Remove Project Handler

Berdasarkan gambar 3.6.1.21., proses *remove project handler* hanya bisa dilakukan oleh *admin* atau *superadmin. Admin* atau *superadmin* hanya perlu menekan *icon* remove pada *handler* dan melakukan konfirmasi. Selanjutnya *handler* akan berhasil dihapus.

### ***Screenshoot* Aplikasi**

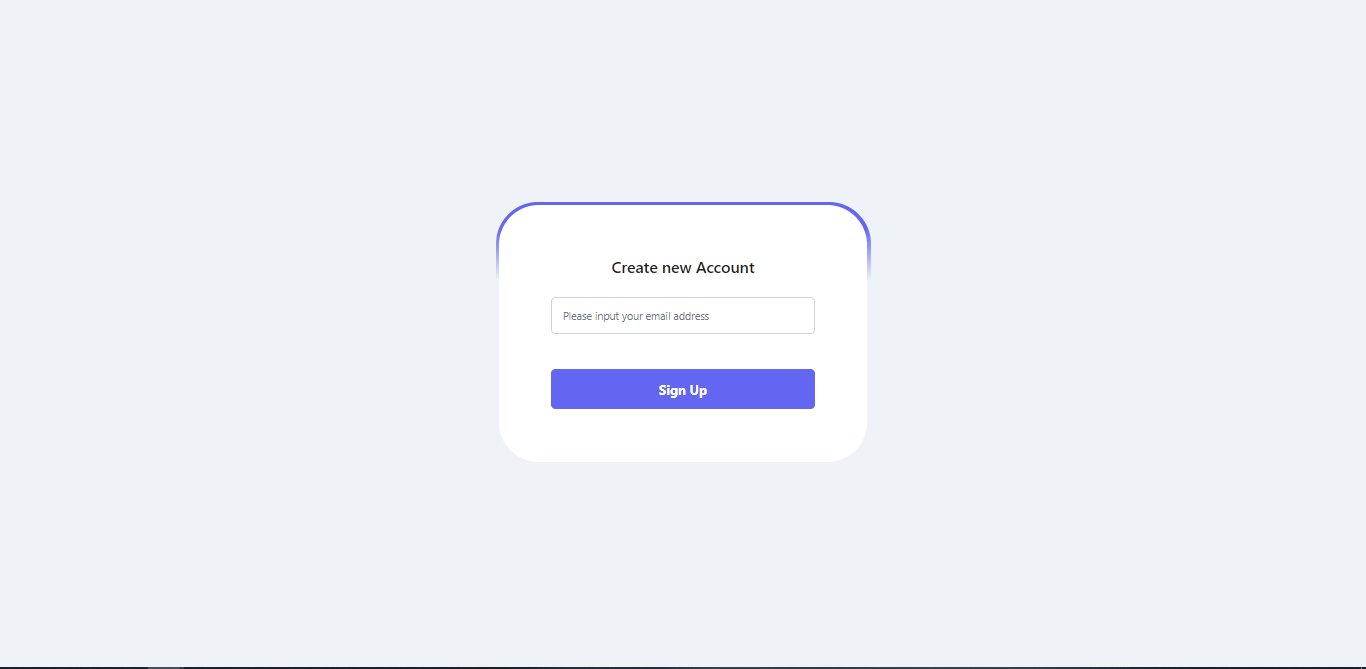
* + - 1. ***Screenshoot* Halaman *Login***



Gambar 3.6.2. 1. Screenshoot Login

Berdasarkan gambar 3.6.2.1., Halaman *login* merupakan halaman awal yang dilakukan untuk autentikasi user. Pada halaman ini terdapat *link* yang mengarahkan pada *forgot password* jika *user* lupa dengan *passwordnya* dan juga *link register* bagi *user* yang akan mendaftar.

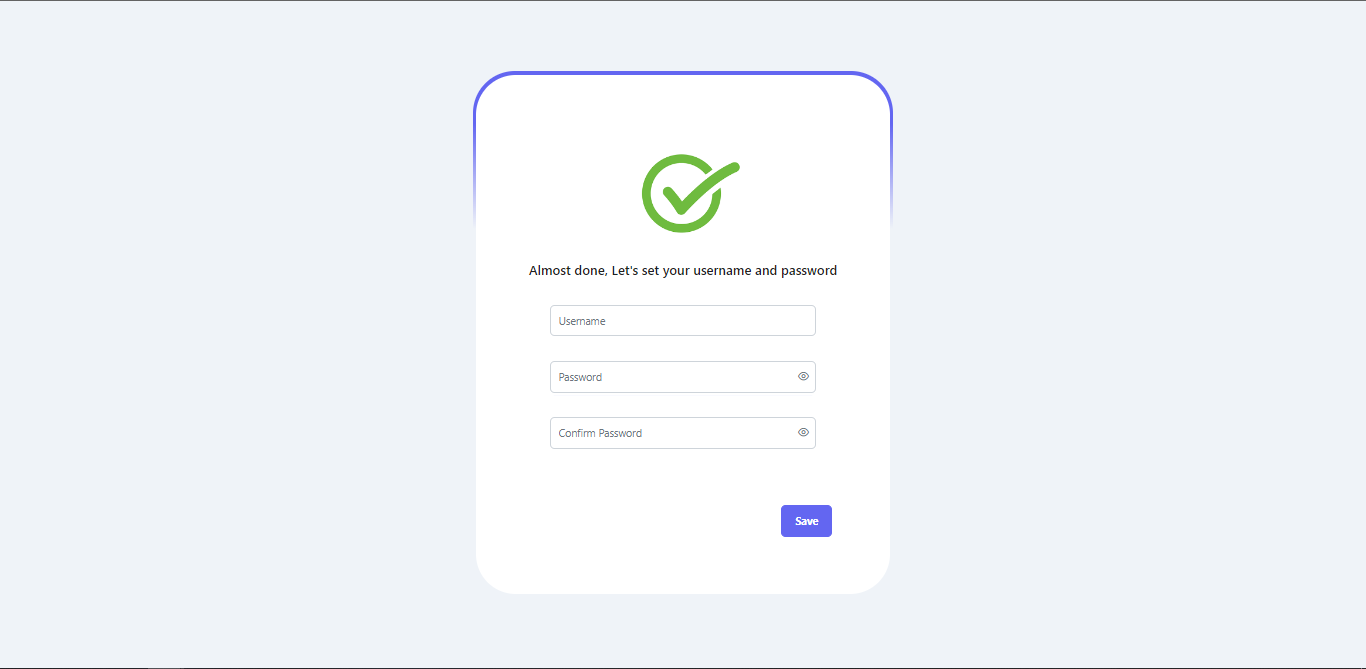
* + - 1. ***Screenshoot* Halaman *Register***



Gambar 3.6.2. 2. Screenshoot Register

Berdasarkan gambar 3.6.2.2., Halaman ini berisi *form* untuk mendaftar yang dimana isinya hanya terdapat *form* *input* *user.* Yang jika diisi maka akan dikirimkan *email* verifikasi dan *setup account*

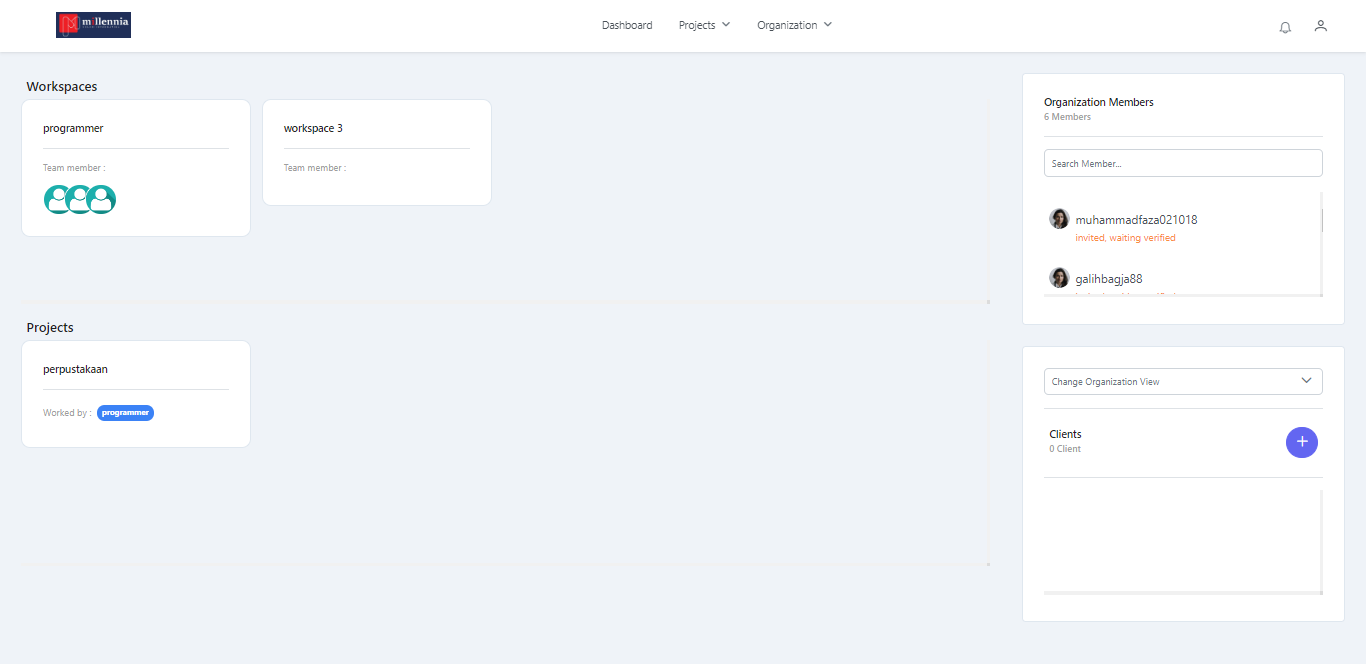
* + - 1. ***Screenshoot* Halaman *Setup Account***



Gambar 3.6.2. 3. Screenshoot Setup Account

Berdasarkan gambar 3.6.2.3., Halaman ini merupakan tahap akhir dari pendaftaran *user*. Pada halaman ini terdapat *form* yang berisi *input username, password* dan *confirm password* yang setelah selesai maka *user* akan dialihkan Kembali ke halaman *login*.

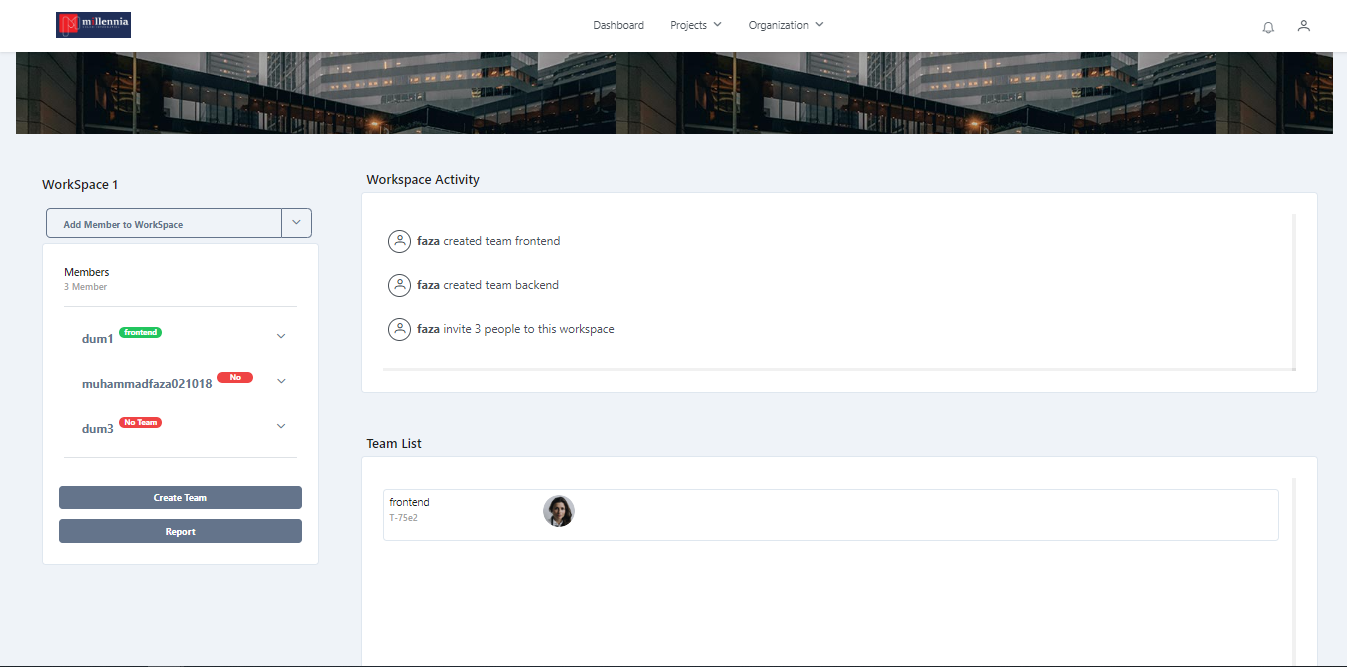
* + - 1. ***Screenshoot* Halaman *Dashboard***



Gambar 3.6.2. 4. Screenshoot Dashboard

Berdasarkan gambar 3.6.2.4., pada halaman ini terdapat daftar *workspace* dan *project* yang terdapat dalam organisasi. Terdapat juga daftar anggota dan Kelola klien disisi kanan. Pada bagian atas juga terdapat beberapa fungsi seperti *invite people, create workspace* dan *create project.*

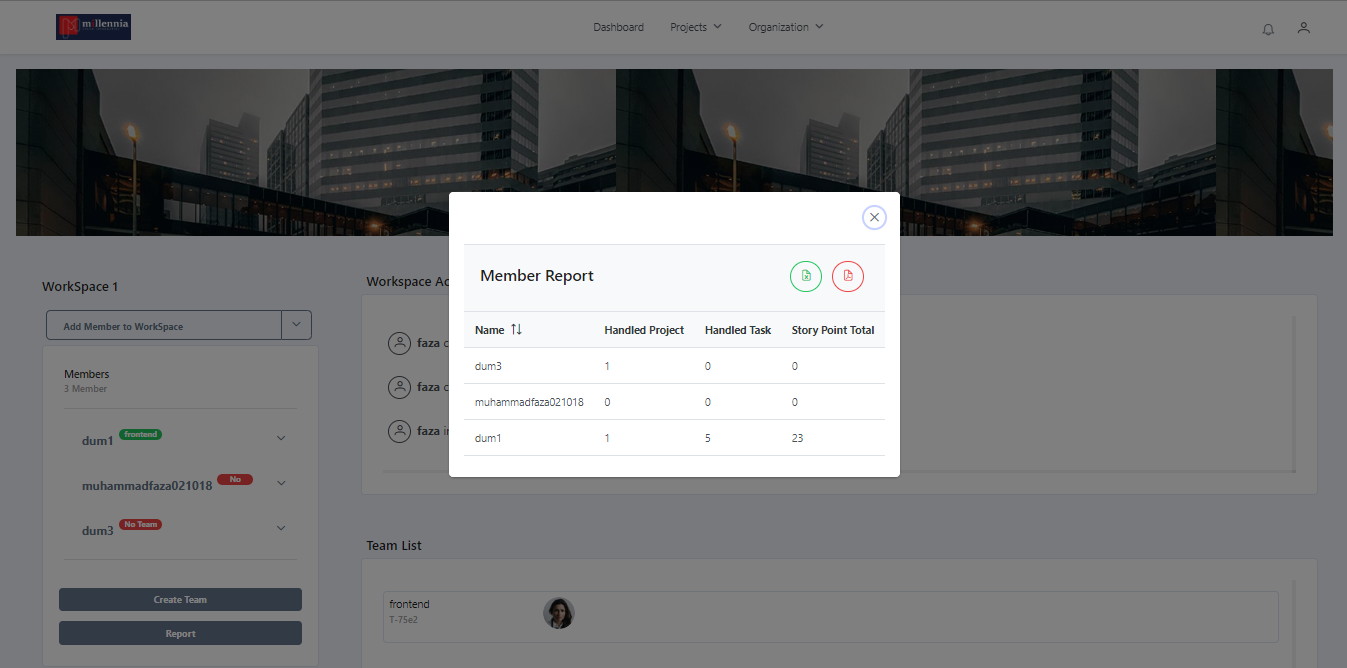
* + - 1. ***Screenshoot* Halaman *Workspace***



Gambar 3.6.2. 5. Screenshoot Halaman Workspace

Berdasarkan gambar 3.6.2.5., Halaman ini berisi detail dari *workspace* pada organisasi. Pada halaman ini juga *user* dapat meng-*invite* user lain ke dalam *workspace*. Pada halaman ini juga terdapat tombol untuk membuat *team* dan *member report.*

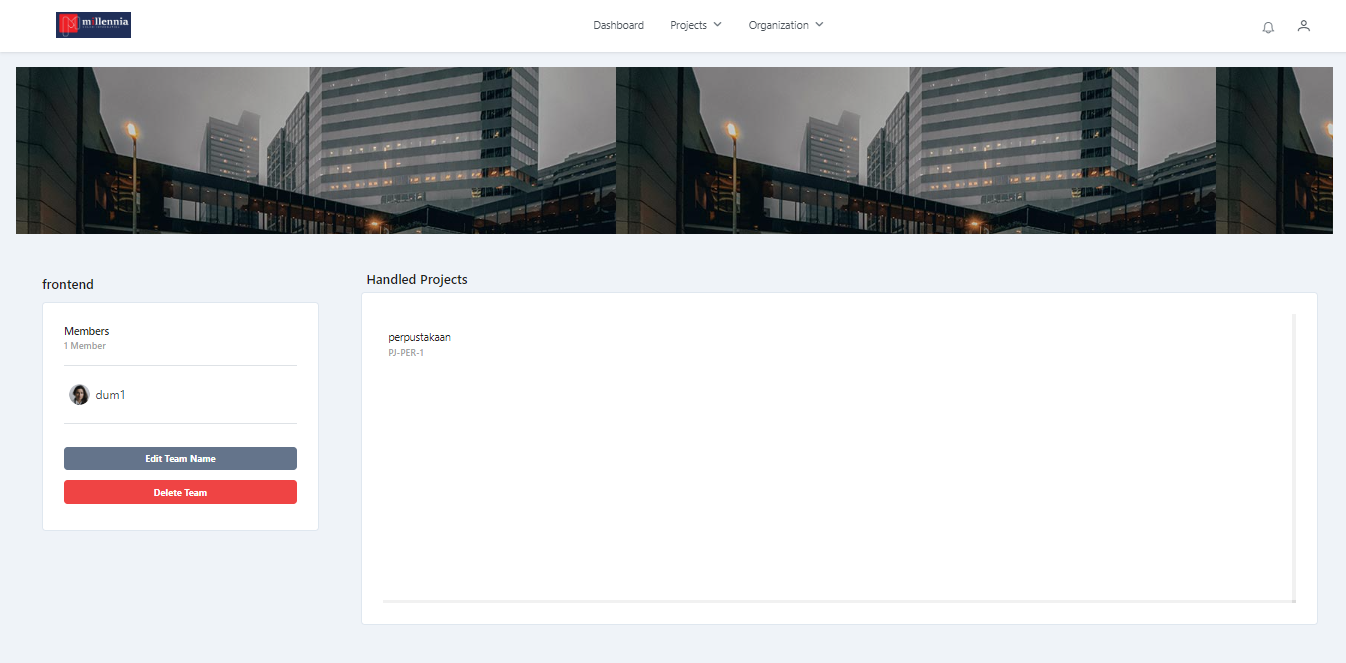
* + - 1. ***Screenshoot* Halaman *Member Report***



Gambar 3.6.2. 6. Screenshoot Halaman Member Report

Berdasarkan gambar 3.6.2.6., Halaman ini merupakan hasil *report* dari tiap member *workspace*. Halaman ini selanjutnya dapat di *generate* menjadi *excel* atau *pdf.*

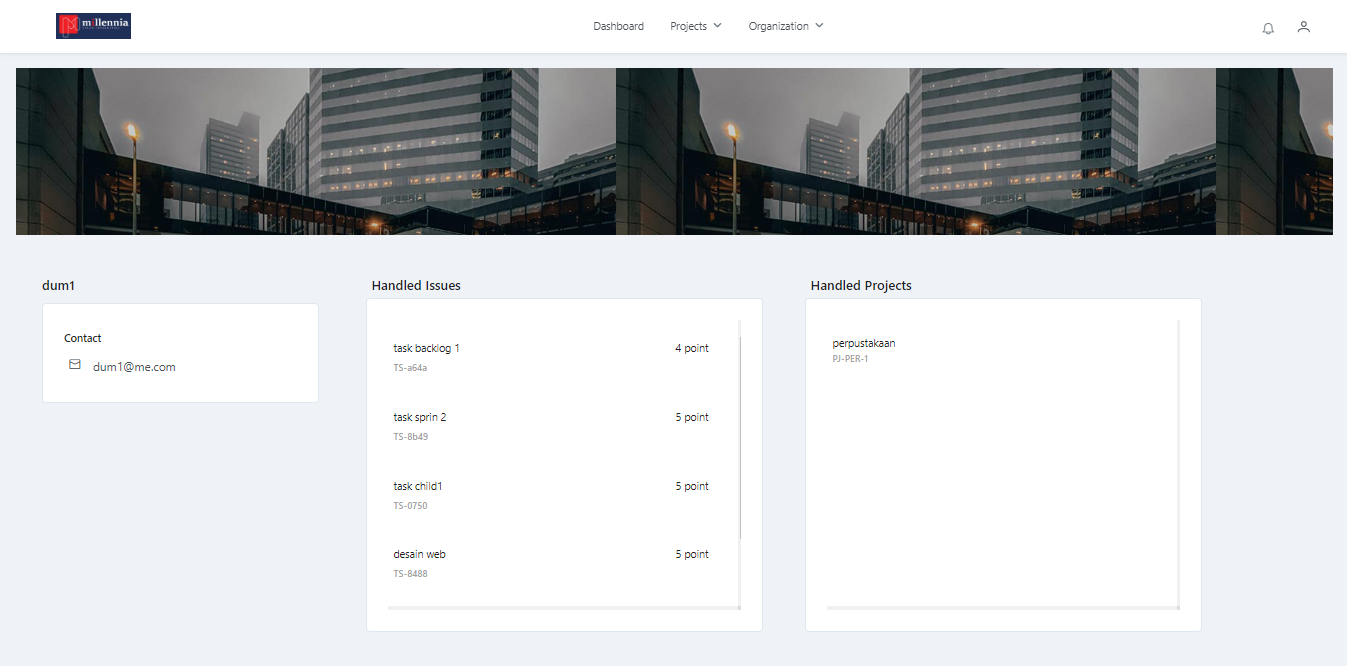
* + - 1. ***Screenshoot* Halaman *team***



Gambar 3.6.2. 7. Screenshoot Halaman Team

Berdasarkan gambar 3.6.2.7., Halaman *team* berisi detail dari *team* pada *workspace*. Di dalamnya terdapat daftar dari *project* yang di-*handle* oleh *team* dan daftar dari member *team.*

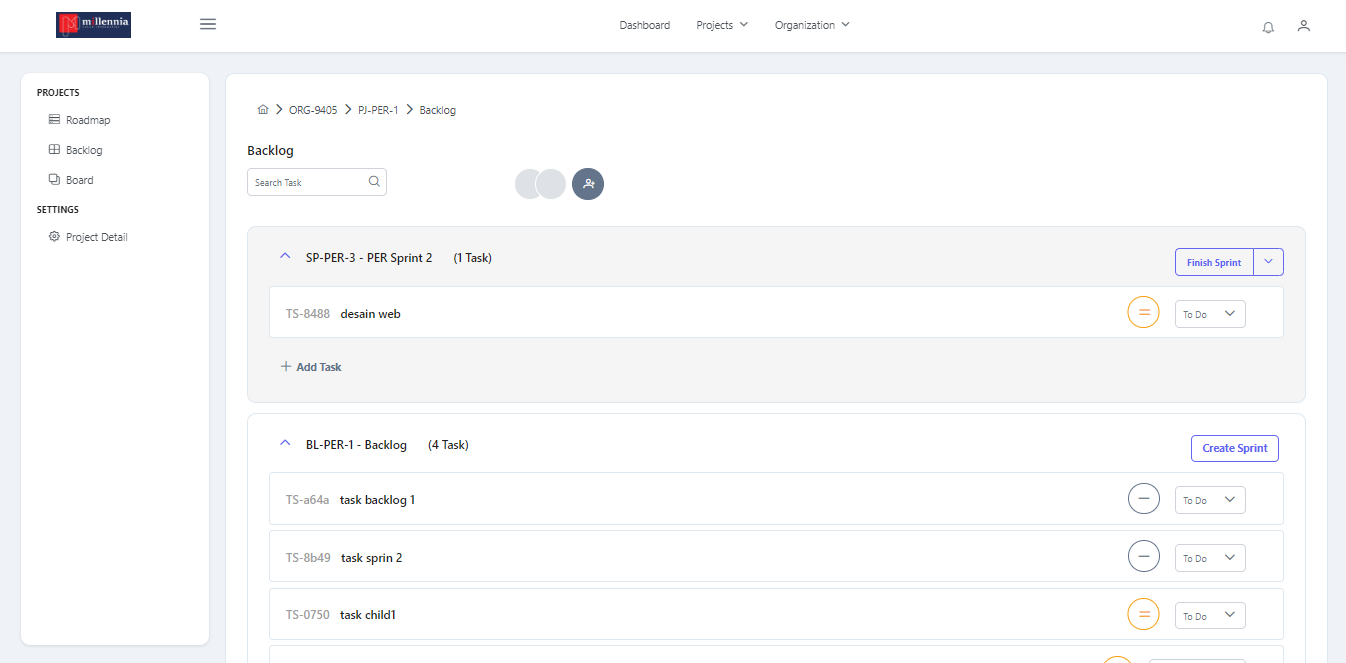
* + - 1. ***Screenshoot* Halaman *member***



Gambar 3.6.2. 8. Screenshoot Halaman Member

Berdasarkan gambar 3.6.2.8., Halaman ini berisi detail dari member pada organisasi. Di dalamnya bisa terlihat semua *project* dan *task* yang ditangani oleh member tersebut.

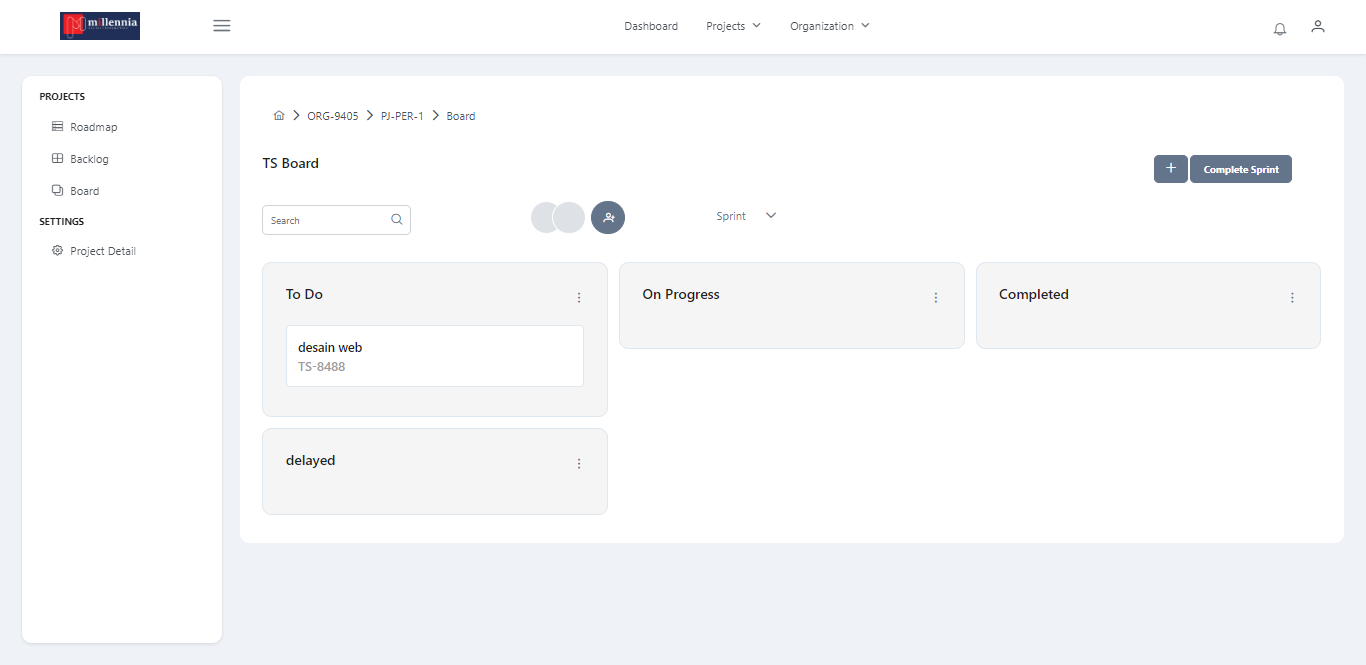
* + - 1. ***Screenshoot* HalamanBacklog *Project***



Gambar 3.6.2. 9. Screenshoot Halaman Backlog Project

Berdasarkan gambar 3.6.2.9., Halaman *backlog* berisi daftar *sprint* dan *backlog* pada *project*. Di dalamnya juga terdapat daftar *task* yang ada di tiap *sprint.*

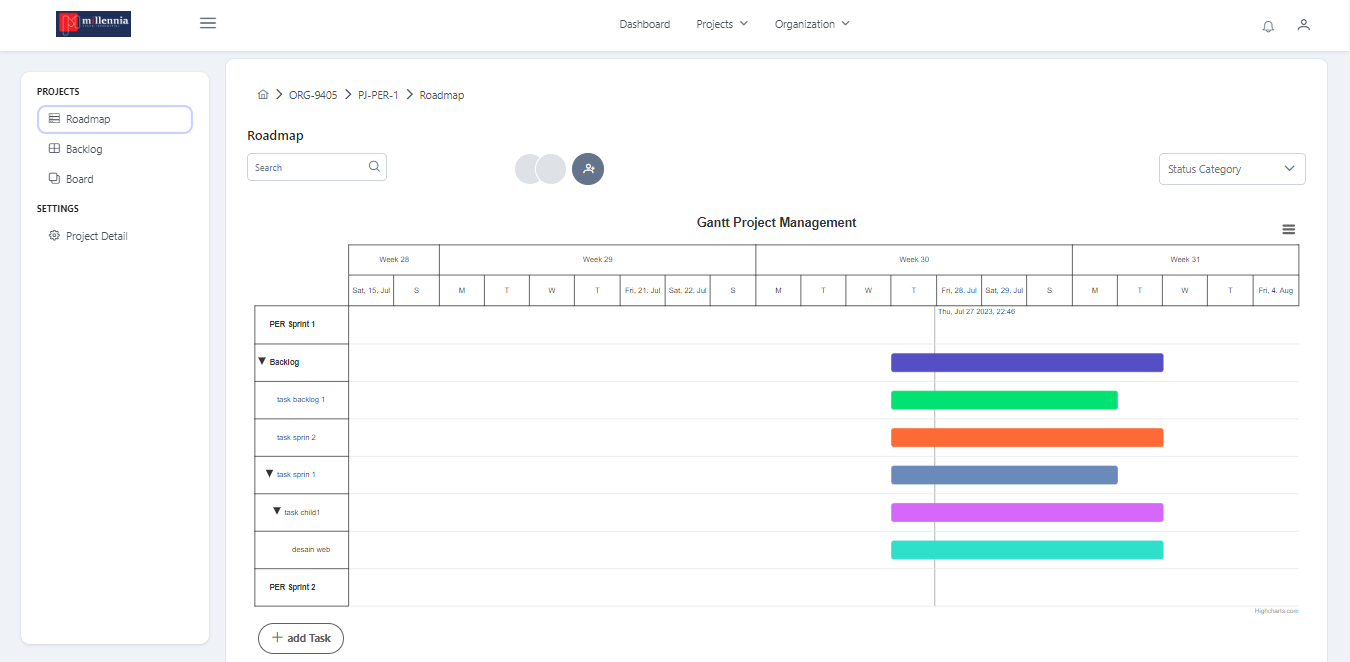
* + - 1. ***Screenshoot* Halaman *Kanban Board Project***



Gambar 3.6.2. 10. Screenshoot Halaman Kanban Board Project

Berdasarkan gambar 3.6.2.10., Halaman ini berisi *kanban* dari status pada project dimana di dalamnya juga terdapat *task* yang dikategorikan berdasarkan status. *Task* yang ditampilkan di halaman ini hanya *task* pada *sprint* yang sudah berjalan.

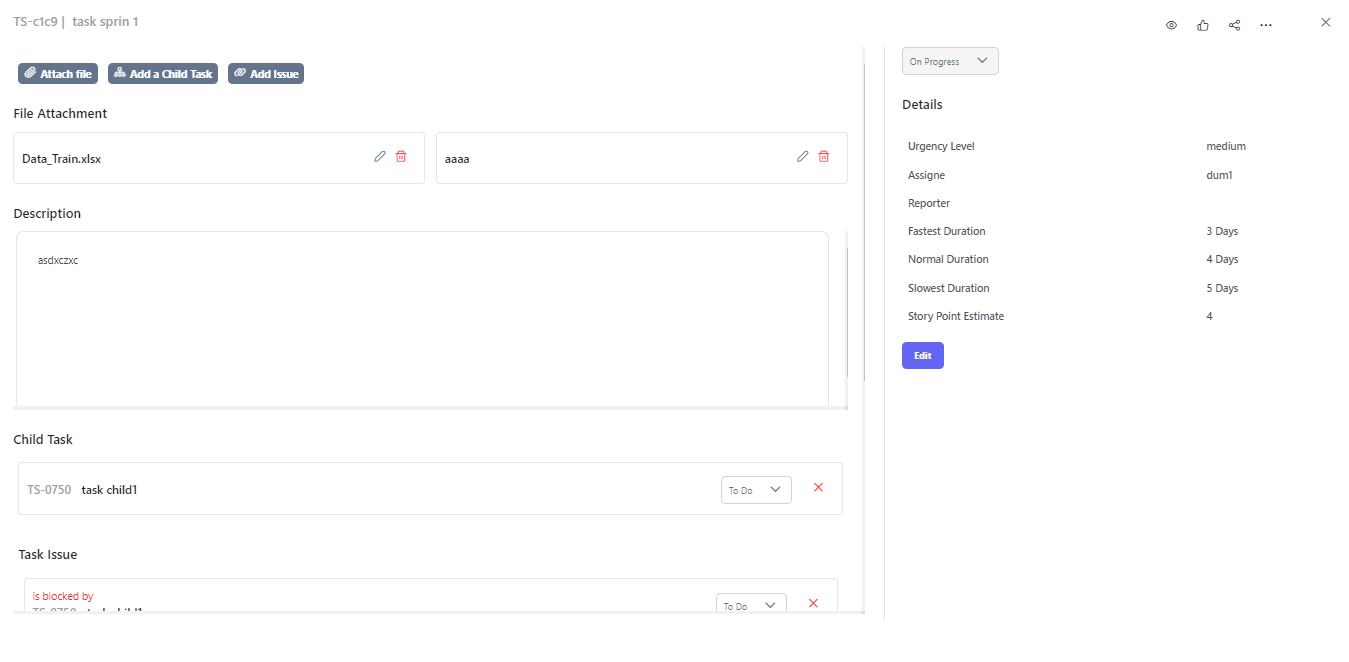
* + - 1. ***Screenshoot* Halaman *Roadmap Project***



Gambar 3.6.2. 11. Screenshoot Halaman Roadmap Project

Berdasarkan gambar 3.6.2.1., Halaman *roadmap* berisi *roadmap* dari tiap *task* yang dihitung berdasarkan waktu pengerjaan yang telah diinputkan sebelumnya.

* + - 1. ***Screenshoot* Halaman *Task Detail***



Gambar 3.6.2. 12. Screenshoot Halaman Task Detail

Berdasarkan gambar 3.6.2.12., Halaman ini berisi detail dari *task*. Di dalamnya terdapat berbagai data seperti *attachment* dari task, *child task, task issue, task description* dan lain sebagainya.

### **Pengujian Aplikasi dengan Metode *Black Box***

* + - 1. **Pengujian Autentikasi Akun**

Tabel 3.6.3. 1. Pengujian autentikasi akun

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Deskripsi | Prosedur  pengujian | masukan | Keluaran yang diharapkan | Hasil yang didapatkan | kesimpulan |
| 1 | Mengisi *email* dan *password* | Klik tombol *sign in* untuk autentikasi akun | *Email* dan *password* | Dialihkan ke *dashboard* organisasi | Dialihkan ke *dashboard* organisasi jika autentikasi berhasil, muncul pesan error jika gagal autentikasi | diterima |

* + - 1. **Pengujian *Register User***

Tabel 3.6.3. 2 Pengujian Register User

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Deskripsi | Prosedur  pengujian | masukan | Keluaran yang diharapkan | Hasil yang didapatkan | kesimpulan |
| 1 | Mengisi *email* | Klik tombol *sign up* untuk membuat akun | *email* | Mendapatkan *email* verifikasi | *Email* verifikasi diterima | diterima |
| 2 | Setup dan verifikasi akun | Klik tombol save untuk verifikasi akun | *Email*, *password* dan confirm *password* | Dialihkan ke halaman login jika berhasil setup dan verifikasi akun | Dialihkan ke halaman login | diterima |

* + - 1. **Pengujian *Dashboard***

Tabel 3.6.3. 3 Pengujian Dashboard

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Deskripsi | Prosedur  pengujian | masukan | Keluaran yang diharapkan | Hasil yang didapatkan | kesimpulan |
| 1 | Melakukan *invite* terhadap *user* di luar organisasi dan aplikasi | Klik tombol *invite* untuk *invite* *user* ke dalam organisasi | *Email* *user*,  role | Muncul pesan jika proses *invite* berhasil dan daftar member organisasi bertambah | Muncul pesan *invite* berhasil dan daftar member organisasi bertambah | Diterima |
| 2 | Membuat *workspace* | Klik tombol *create workspace* untuk membuat *workspace* | Nama *workspace* | Muncul pesan jika proses pembuatan berhasil dan daftar *workspace* bertambah | Muncul pesan pembuatan *workspace* berhasil dan daftar *workspace* bertambah | diterima |
| 3 | Tambah *client* | Klik tombol *add client* untuk menambahkan *client* | Nama *client* dan deskripsi *client* | Muncul pesan jika pembuatan  *Client* berhasil dan daftar *client* bertambah | Muncul pesan  *Client* berhasil dan daftar *client* bertambah | diterima |
| 4 | Ubah data *client* | Klik tombol *edit* untuk mengubah data *client* | Nama *client* dan deskripsi *client* | Muncul pesan jika ubah data  *Client* berhasil | Muncul pesan ubah data  *Client* berhasil | diterima |
| 5 | Hapus data *client* | Klik tombol *delete* untuk menghapus data *client* | - | Muncul pesan jika hapus data  *Client* berhasil | Muncul pesan hapus data  *Client berhasil* | diterima |
| 6 | Pembuatan *project* | Klik tombol *create project* untuk membuat *project* | Nama *project*, *client*, *workspace¸*dan member *workspace* yang dipilih | Muncul pesan jika proses pembuatan berhasil | Muncul pesan pembuatan *project* berhasil | diterima |

* + - 1. **Pengujian *Workspace***

Tabel 3.6.3. 4 Pengujian Workspace

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Deskripsi | Prosedur  pengujian | masukan | Keluaran yang diharapkan | Hasil yang didapatkan | kesimpulan |
| 1 | Melakukan *invite* terhadap *user* di luar *workspace*, organisasi atau aplikasi | Klik tombol *invite* untuk *invite* *user* ke dalam *workspace* | *Email* *user*,  role | Muncul pesan jika proses *invite* berhasil, daftar member *workspace* bertambah dan *workspace activity* berubah | Muncul pesan *invite* berhasil, daftar member *workspace* bertambah dan *workspace activity* berubah | Diterima |
| 2 | Membuat *team* | Klik tombol *create team* untuk membuat *team* | Nama *team* dan *user* yang akan dimasukan ke dalam *team* | Muncul pesan jika proses berhasil, daftar *team* bertambah dan *workspace activity* berubah | Muncul pesan proses berhasil, daftar *team* bertambah dan *workspace activity* berubah | Diterima |
| 3 | Ubah member *team* | Klik tombol *add team* atau *change team* untuk mengubah member *team* | - | Muncul pesan jika proses berhasil, data *team* berubah dan *workspace activity* berubah | Muncul pesan proses berhasil, data *team* berubah dan *workspace activity* berubah | diterima |
| 4 | Mengeluarkan member dari *workspace* | Klik tombol *kick member from workspace* untuk mengeluarkan member | - | Muncul pesan jika proses berhasil, data *workspace* berubah dan *workspace activity* berubah | Muncul pesan proses berhasil, data *workspace* berubah dan *workspace activity* berubah | diterima |

* + - 1. **Pengujian *Member Report***

Tabel 3.6.3. 5 Pengujian Member Report

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Deskripsi | Prosedur  pengujian | masukan | Keluaran yang diharapkan | Hasil yang didapatkan | kesimpulan |
| 1 | *Export report* | Klik *icon* *excel* untuk *export* ke *excel* dan klik tombol *pdf* untuk *export* ke dalam *pdf* | *-* | Dokumen berhasil di *generate* | Dokumen berhasil di *generate* | diterima |

* + - 1. **Pengujian *Teams***

Tabel 3.6.3. 6 Pengujian Teams

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Deskripsi | Prosedur  pengujian | masukan | Keluaran yang diharapkan | Hasil yang didapatkan | kesimpulan |
| 1 | Ubah nama *team* | Klik tombol save untuk mengubah nama *team* | Nama *team* | Muncul pesan jika proses berhasil, data *teams* berubah | Muncul pesan proses berhasil, data *team* berubah | diterima |
| 2 | Hapus *team* | Klik tombol *delete* *team* untuk menghapus *team* | - | Muncul pesan jika proses berhasil, data *di*alihkan ke halaman *workspace* | Muncul pesan proses berhasil, data *di*alihkan ke halaman *workspace* | diterima |

* + - 1. **Pengujian *Project***

Tabel 3.6.3. 7 Pengujian Project

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Deskripsi | Prosedur  pengujian | masukan | Keluaran yang diharapkan | Hasil yang didapatkan | kesimpulan |
| 1 | Ubah nama *project* | Klik tombol save untuk mengubah nama *project* | Nama *project* | Muncul pesan jika proses berhasil, data *project* berubah | Muncul pesan proses berhasil, data *project* berubah | diterima |
| 2 | Hapus *project* | Klik tombol *delete* *project* untuk menghapus *project* | - | Muncul pesan jika proses berhasil, data *di*alihkan ke halaman *dashboard* | Muncul pesan proses berhasil, data *di*alihkan ke halaman *dashboard* | diterima |
| 3 | Tambah *project handler* | Klik tombol *invite* untuk menambah *project handler* | Nama member | Muncul pesan jika proses berhasil, data *project* berubah | Muncul pesan proses berhasil, data *project* berubah | diterima |

* + - 1. **Pengujian *Backlog* *Project***

Tabel 3.6.3. 8 Pengujian Backlog Project

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Deskripsi | Prosedur  pengujian | masukan | Keluaran yang diharapkan | Hasil yang didapatkan | kesimpulan |
| 1 | Membuat sprint | Klik tombol *create sprint* untuk menambahkan sprint | - | Muncul *sprint* baru pada daftar *sprint* | Muncul *sprint* baru pada daftar *sprint* | diterima |
| 2 | Membuat *task* | Klik tombol *add task* untuk membuat *task* | Nama *task*, level, penerima tugas, pelapor tugas, waktu pengerjaan paling cepat, waktu pengerjaan normal, waktu pengerjaan paling lambar | Muncul *task* baru pada *sprint* | Muncul *task* baru pada *sprint* | diterima |
| 3 | Ubah *sprint* dari *task* | *Drag and drop* *task* ke *sprint* lain | - | Posisi *task* berpindah | Posisi *task* berpindah | diterima |
| 4 | Ubah status *task* | Klik status *task* yang akan dipilih | Status *task* | Status *task* berubah | Status *task* berubah | diterima |
| 5 | Memulai *sprint* | Klik tombol *start sprint* untuk memulai sprint | - | Dialihkan ke halaman board | Dialihkan ke halaman board | diterima |
| 6 | mencari *task* | Mengetik nama *task* yang dicari | Nama *task* | Daftar *task* hanya memunculkan *task* yang dicari | Daftar *task* hanya memunculkan *task* yang dicari | diterima |

* + - 1. **Pengujian *Board* *Project***

Tabel 3.6.3. 9 Pengujian Board Project

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Deskripsi | Prosedur  pengujian | masukan | Keluaran yang diharapkan | Hasil yang didapatkan | kesimpulan |
| 1 | Ubah status *task* | *Drag and drop* *task* ke status yang lain | - | Posisi *task* berpindah | Posisi *task* berpindah | diterima |
| 2 | Tambah status | Klik *icon* tambah untuk menambahkan status | Nama status | Muncul *kanban* status baru | Muncul *kanban* status baru | diterima |
| 3 | *Complete sprint* | Klik tombol *complete sprint* untuk menyelesaikan *sprint* | *Sprint* yang akan diselesaikan dan *sprint* pengalihan untuk *task* yang belum selesai | *Task* dari sprint yang diselesaikan akan berubah atau menghilang | *Task* dari sprint yang diselesaikan akan berubah atau menghilang | diterima |
| 4 | mencari *task* | Mengetik nama *task* yang dicari | Nama *task* | Daftar *task* hanya memunculkan *task* yang dicari | Daftar *task* hanya memunculkan *task* yang dicari | diterima |
| 5 | Memilih *task sprint* yang akan ditampilkan | Memilih sprint | - | Daftar *task* hanya memunculkan *task* dari *sprint* yang dipilih | Daftar *task* hanya memunculkan *task* dari *sprint* yang dipilih | diterima |

* + - 1. **Pengujian *Roadmap* *Project***

Tabel 3.6.3. 10 Pengujian Roadmap Project

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Deskripsi | Prosedur  pengujian | masukan | Keluaran yang diharapkan | Hasil yang didapatkan | kesimpulan |
| 1 | Membuat *task* | Klik tombol *add task* untuk membuat *task* | Nama *task*, level, sprint, penerima tugas, pelapor tugas, waktu pengerjaan paling cepat, waktu pengerjaan normal, waktu pengerjaan paling lambar | Muncul *task* baru pada *roadmap* | Muncul *task* baru pada *roadmap* | diterima |
| 2 | mencari *task* | Mengetik nama *task* yang dicari | Nama *task* | hanya memunculkan *task* yang dicari | hanya memunculkan *task* yang dicari | diterima |

* + - 1. **Pengujian *Task* *Detail***

Tabel 3.6.3. 11 Pengujian Task Detail

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Deskripsi | Prosedur  pengujian | masukan | Keluaran yang diharapkan | Hasil yang didapatkan | kesimpulan |
| 1 | Ubah status *task* | Klik status *task* yang akan dipilih | Status *task* | Status *task* berubah | Status *task* berubah | diterima |
| 2 | Ubah detail *task* | Klik tombol *edit* untuk mengubah detail *task* | Nama tak, level, penerima tugas, pelapor tugas, waktu pengerjaan paling cepat, waktu pengerjaan normal, waktu pengerjaan paling lambar | Muncul pesan edit berhasil dan detail *task* berubah | Muncul pesan edit berhasil dan detail *task* berubah | diterima |
| 3 | Mengawasi *task* | Klik tombol *start watching* untuk mulai mengawasi *task* | - | Muncul nama *user* yang mengawasi dan *Task Activity* berubah | Muncul nama *user* yang mengawasi dan *Task Activity* berubah | diterima |
| 4 | *vote* *task* | Klik tombol vote *this task* untuk mulai *vote* *task* | - | Muncul nama *user* yang melakukan *vote* dan *Task Activity* berubah | Muncul nama *user* yang melakukan *vote* dan *Task Activity* berubah | diterima |
| 5 | *Share* *task* | Klik *share* untuk mengirimkan detail *task* ke *user* lain atau klik icon *copy* untuk meng-*copy* *task URL* | Nama *user*, pesan | Muncul pesan jika proses berhasil dan *Task Activity* berubah | Muncul pesan proses berhasil dan *Task Activity* berubah | diterima |
| 6 | Ubah deskripsi *task* | Klik tombol *save* untuk menyimpan deskripsi baru | Deskripsi *task* | Muncul pesan jika proses berhasil dan *Task Activity* berubah | Muncul pesan jika proses berhasil dan *Task Activity* berubah | diterima |
| 7 | Membuat *child* *task* | Klik tombol *add task* untuk membuat *child task* | Nama *task*, level, sprint, penerima tugas, pelapor tugas, waktu pengerjaan paling cepat, waktu pengerjaan normal, waktu pengerjaan paling lambar | Muncul *task* baru pada *child task* dan *Task Activity* berubah | Muncul *task* baru pada *child task* dan *Task Activity* berubah | diterima |
| 7 | Membuat *task issue* | Klik tombol *add issue* untuk membuat *task issue* | Jenis *issue* dan *task* yang mengalami masalah | Muncul *task* baru pada *task issue* dan *Task Activity* berubah | Muncul *task* baru pada *task issue* dan *Task Activity* berubah | diterima |

* + - 1. **Pengujian *Attachment* *File***

Tabel 3.6.3. 12 Pengujian Attachment File

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Deskripsi | Prosedur  pengujian | masukan | Keluaran yang diharapkan | Hasil yang didapatkan | kesimpulan |
| 1 | *Upload attachment* | Klik tombol *upload* untuk meng-*upload task* | File yang akan diupload | List *task attachment* akanbertambah dan *Task Activity* berubah | List *task attachment* bertambah dan *Task Activity* berubah | diterima |
| 2 | Ubah nama *attachment* | Klik tombol *save* | Nama *attachment* | Nama *attachment* akan berubah | Nama *attachment* berubah | diterima |
| 3 | Hapus *attachment* | Klik *icon delete* untuk menghapus *attachment* | - | *Attachment* akan terhapus dan *Task Activity* berubah | *Attachment* terhapus dan *Task Activity* berubah | diterima |

* + - 1. **Pengujian *Child* *Task***

Tabel 3.6.3. 13 Pengujian Child Task

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Deskripsi | Prosedur  pengujian | masukan | Keluaran yang diharapkan | Hasil yang didapatkan | kesimpulan |
| 1 | Ubah status *task* | Klik status *task* yang akan dipilih | Status *task* | Status *task* berubah | Status *task* berubah | diterima |
| 2 | Menghapus *child task* | Klik *icon delete* untuk menghapus *child task* | - | *child task* akan terhapus dan *Task Activity* berubah | *child task* terhapus dan *Task Activity* berubah | diterima |

* + - 1. **Pengujian *Task* Issue**

Tabel 3.6.3. 14 Pengujian Task Issue

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Deskripsi | Prosedur  pengujian | masukan | Keluaran yang diharapkan | Hasil yang didapatkan | kesimpulan |
| 1 | Ubah status *task* | Klik status *task* yang akan dipilih | Status *task* | Status *task* berubah | Status *task* berubah | diterima |
| 2 | Menghapus *child issue* | Klik *icon delete* *child issue* | - | *child issue* akan terhapus | *child issue* terhapus | diterima |

* + - 1. **Pengujian *Task Activity***

Tabel 3.6.3. 15 Pengujian *Task Activity*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Deskripsi | Prosedur  pengujian | masukan | Keluaran yang diharapkan | Hasil yang didapatkan | kesimpulan |
| 1 | Memfilter *task comment* atau *task history* | Klik jenis aktivitas yang akan ditampilkan | - | Aktivitas yang muncul hanya jenis aktivitas yang dipilih | Aktivitas yang muncul hanya jenis aktivitas yang dipilih | diterima |
| 2 | Komentar *task* | Klik tombol save untuk *comment* *task* | komentar | *Task Activity* akan berubah | *Task Activity* berubah | diterima |
| 3 | Hapus komentar | Klik *icon delete* untuk menghapus komentar | - | komentar akan terhapus | komentar terhapus dan *Task Activity* berubah | diterima |

# BAB IV

# KESIMPULAN DAN SARAN

## **Kesimpulan**

Berdasarkan latar belakang dan ruang lingkup pada penelitian ini, penulis dapat menyimpulkan bahwa Aplikasi Manajemen Proyek Berbasis *Web* Pada PT Millennia Solusi Informatika sebagai berikut :

1. Aplikasi telah melewati tahap pengujian secara *black box* dan telah memenuhi persyaratan yang dibutuhkan oleh pihak Perusahaan.
2. Aplikasi telah dibangun berdasarkan permintaan dan kebutuhan Perusahaan dengan menggunakan *PostgreSQL* sebagai basis data dan berbasis *website*.
3. Fitur pada aplikasi telah disesuaikan dengan kebutuhan dari Perusahaan meliputi: *register, login, verification email, dashboard, workspace, team, project, backlog, sprint, task, child task, attachment, status, roadmap, board, client, profile setting*, dan dibantu dengan perhitungan *pert method* untuk mentukan prioritas *task* yang harus lebih penting untuk dikerjakan*.* Dengan fitur - fitur tersebut aplikasi ini dapat membantu dalam pengerjaan proyek pada perusahaan.

## **Saran**

Setelah melaksanakan kerja praktik selama 3 bulan pada perusahaan PT Millennia Solusi Informatika dan melakukan pembuatan Aplikasi Manajemen Proyek Berbasis *Web* ini kedepannya diharapkan dapat digunakan oleh perusahaan sebagai pengelolaan proyek dan dikembangkan lagi dengan membuat fitur-fitur baru yang membuat proyek yang dikerjakan di perusahaan dapat berjalan secara efektif. Selain itu, diharapkan aplikasi dapat diperbaharui Kembali oleh Perusahaan mengingat proses pembuatan dan pengujian aplikasi masih belum dilaksanakan dengan maksimal.

# DAFTAR PUSTAKA

Ambriani, D., & Nurhidayat, A. I. (2020). RANCANG BANGUN REPOSITORY PUBLIKASI ILMIAH DOSEN BERBASIS WEB. *Jurnal Manajemen Informatika. Volume 10 Nomor 01 Tahun 2020*, 58-66.

Anendya, A. (2023). *Mengenal Apa Itu Flowchart, Fungsi, Jenis, dan Contohnya*. Retrieved from https://www.dewaweb.com/blog/pengertian-flowchart/

Ariffudin, M. (2021). *Mengenal Express.js: Pengertian, Cara Kerja, Keunggulan, Tutorial*. Retrieved from https://www.niagahoster.co.id/blog/express-js-adalah

Christanto, E., & Wibowo, T. (2020). *menggunakan aplikasi Microsoft Excel*. *1*.

Darmawan, D., & Ratnasari, A. (2020). Rancang Bangun Sistem Informasi Manajemen Proyek Berbasis *Web* Pada Pt Seatech Infosys. *Jurnal Sisfokom (Sistem Informasi Dan Komputer)*, *9*(3), 365–372. https://doi.org/10.32736/sisfokom.v9i3.931

Hardiansyah, A. D., & Puspita Dewi, C. N. (2020). PERANCANGAN BASIS DATA SISTEM INFORMASI PERWIRA TUGAS BELAJAR (SIPATUBEL) PADA KEMENTERIAN PERTAHANAN. 223.

Institute, P. M. (2017). pengertian manajemen proyek. *Project Management Body of Knowledge*, 49

Laily, I. N. (2022). *Pengertian Website Menurut Para Ahli, Beserta Jenis dan Fungsinya*.https://katadata.co.id/safrezi/berita/6200a2a9697ec/pengertian-website-menurut-para-ahli-beserta-jenis-dan-fungsinya#google\_vignette

Rosanti, N., Setiawan, E., & Ayuningtyas, A. (2016). Penggunaan Metode Jalur Kritis Pada Manajemen Proyek (Studi Kasus: Pt. Trend Communications International). *Jurnal Teknologi*, *8*(1), 23. https://doi.org/10.24853/jurtek.8.1.23-30