

BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Organisasi

Kedudukan pada pelaksanaan program kerja magang di PT. Pranala Ragam Karya berdasarkan struktur organisasi adalah bagian *programmer*. *Programmer* bertanggung jawab langsung kepada *manager proyek* terhadap tugas-tugas yang dilakukan. *Proyek manager* pada proyek *PaperWork* merangkap sebagai mentor saat program kerja magang.

Dipercayakan untuk berada di posisi *Software Developer* yang ditugaskan dalam membangun *front-end* dan *back-end* pada proyek utama magang. Sebagai karyawan magang pada perusahaan PT. Pranala Ragam Karya memiliki proyek utama yaitu *website PaperWork*. Pengerjaan proyek ini dilaksanakan *Full WFH (Work From Home)*.

Sebelum mengembangkan proyek utama *PaperWork*, pemegang di PT. Pranala Ragam Karya melaksanakan proses pelatihan selama dua bulan awal. Pemegang dikoordinasi oleh *HRD* dan mentor setiap divisi. Kendala atau pertanyaan dari pemegang dapat ditanyakan ke mentor divisi masing-masing.

3.2 Tugas yang Dilakukan

Selama kerja magang pada perusahaan PT. Pranala Ragam Karya diawali dengan *training* selama dua bulan dan baru masuk ke proyek utama perusahaan. Proyek utama *PaperWork* ini menggunakan *React.js* dan *Node.js* dengan *library zod formik, redux toolkit*, menggunakan *database MySQL* dan *prisma orm, express.js* dan *typescript*. Tugas-tugas yang sudah dilakukan selama program kerja magang adalah sebagai berikut:

1. Mengeksplorasi dan mempelajari *website PaperWork*
2. Membuat *template* pengiriman *email* ke *user* menggunakan *CSS*
3. Membuat fungsi pengiriman *email* ke *user* menggunakan *nodemailer*, selama proses percobaan pengiriman *email* menggunakan *smtp mailtrap*
4. Mempelajari *framework CSS Tailwind*

5. Melakukan *re-design card* aplikasi
6. Membuat fitur menu untuk menampilkan aplikasi yang tersedia
7. Melakukan *re-design* pada saat nama agar muncul *popup dialog* berupa detail dari nama yang diklik
8. Membuat fitur baru *task* mulai dari *front-end* hingga *back-end*

3.3 Uraian Pelaksanaan Magang

Proses pelaksanaan kerja magang dilakukan selama empat bulan. Tugas yang diberikan pada pemegang diberikan pada *software* Jira dan aplikasi Skype. Pelaksanaan kerja magang diuraikan setiap minggu seperti pada Tabel 3.1.

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

Minggu Ke -	Pekerjaan yang dilakukan
1	Melakukan instalasi dan mempelajari <i>ReactJS, NodeJS, ubuntu</i> .
2	Melakukan proses <i>training</i> perusahaan dengan membuat tampilan <i>login</i> dan <i>register</i> akun pada <i>website</i> .
3	Melakukan proses <i>training</i> perusahaan dengan membuat tampilan yang sudah dibuat dapat berfungsi. Membuat <i>backend</i> dengan <i>MySQL</i> untuk validasi proses <i>login</i> dan <i>register</i> .
4	Melakukan proses <i>training</i> perusahaan dengan membuat fungsi <i>forget password</i> menggunakan <i>nodemailer</i> .
5	Melakukan proses <i>training</i> perusahaan dengan mengimplementasikan <i>zod formik</i> pada validasi form <i>login</i> dan <i>register</i> yang sudah dibuat.
6	Melakukan proses <i>training</i> perusahaan dengan mengimplementasikan <i>redux toolkit</i> setelah proses <i>login</i> dan membuat fitur <i>log out</i> .
7	Melakukan proses <i>training</i> perusahaan dengan mengimplementasikan <i>local storage</i> dan <i>library redux-persist</i> .
8	Melakukan proses <i>training</i> perusahaan dengan tugas <i>rewrite backend</i> agar menggunakan Prisma ORM.
Lanjut ke halaman berikutnya	

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

Minggu Ke -	Pekerjaan yang dilakukan
9	Masuk ke proyek utama mengerjakan tampilan <i>template email</i> verifikasi akun dengan <i>CSS</i> .
10	Melakukan percobaan pengiriman <i>template email</i> yang sudah dibuat menggunakan <i>nodemailer</i> .
11	Mengerjakan <i>re-design</i> pada tampilan <i>card</i> aplikasi.
12	Mengerjakan fitur baru berupa <i>icon menu</i> yang dapat diklik untuk menampilkan aplikasi yang tersedia.
13 - 14	Mengerjakan <i>re-design</i> pada bagian halaman anggota agar ketika nama anggota yang diklik muncul profil, nama, email, tanggal bergabung, dan status.
15	Melakukan perencanaan fitur baru berdasarkan ide individu untuk proyek utama.
16 - 19	Membuat fitur <i>task</i> untuk halaman admin yang berisi formulir untuk <i>input</i> tugas dan <i>list</i> tugas yang sudah dibuat. Membagi <i>list</i> tugas menjadi kategori <i>backlog</i> , <i>progress</i> , <i>review</i> , dan <i>done</i> . Mengimplementasi <i>backend</i> dengan menggunakan <i>MySQL</i>

3.3.1 Perencanaan

Pada proses pelaksanaan magang di perusahaan PT. Pranala Ragam Karya, tim karyawan magang divisi *software developer* ditugaskan untuk mempelajari *React JS*, *Node JS*, *nodemailer*, *library zod* dan *formik*, *library redux toolkit*, *redux-persist*, *local storage*, *prisma ORM*, *library express handlebar*, *CSS tailwind*, *library tailwind*, *TypeScript*, *JavaScript*, *Redux Toolkit Query*. Berikut merupakan perangkat lunak dan perangkat keras yang digunakan selama masa magang di perusahaan PT. Pranala Ragam Karya:

A. Perangkat Lunak

Perangkat Lunak yang digunakan saat membuat *website PaperWork* pada perusahaan PT. Pranala Ragam Karya sebagai berikut:

- Sistem Operasi: Linux (Ubuntu versi 22.04.3 LTS)
- Visual Studio Code untuk melakukan proses pemrograman

- Bitbucket untuk tempat pengumpulan hasil proyek yang sudah dikerjakan
- Jira untuk tempat pemberian tugas harian
- Skype dan WA untuk komunikasi dengan perusahaan PT. Pranala Ragam Karya

B. Perangkat Keras

Perangkat keras yang digunakan saat membuat *website PaperWork* pada perusahaan PT. Pranala Ragam Karya sebagai berikut:

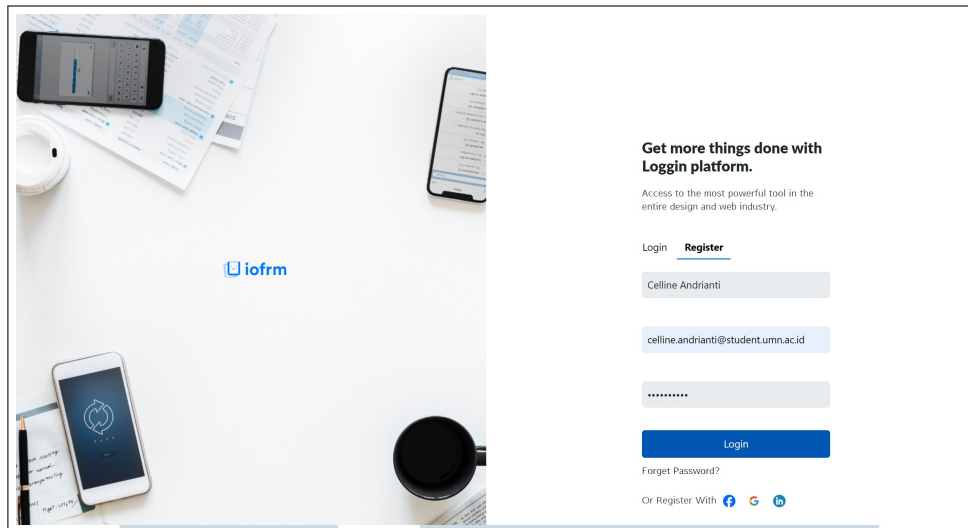
- Prosesor: AMD Ryzen 7 5800H with Radeon Graphics 3.20 GHz
- Memori: 16,0 GB
- SSD: 1 Tera
- Grafik: NVIDIA GeForce RTX 3060

3.3.2 Pengerjaan Website Login dan Register Pada Masa Training

Pada proses pelaksanaan magang diawali dengan tahap training untuk mempelajari library nodemailer, library formik, library zod, library redux toolkit, Prisma ORM, library express handlebar. Proyek ini menjadi tanggung jawab dari masing-masing anggota pada tim programmer yang berjumlah tiga orang. Setiap anggota membuat proyeknya masing-masing terkait tugas website *login* dan *register* ini. Tugas yang diberikan pada masa training bermanfaat untuk membangun proyek utama. Proyek pada saat masa training adalah membuat website untuk *login*, *register*, dan log out. Berikut ini merupakan tampilan website proyek pada masa training.

A. Halaman Register

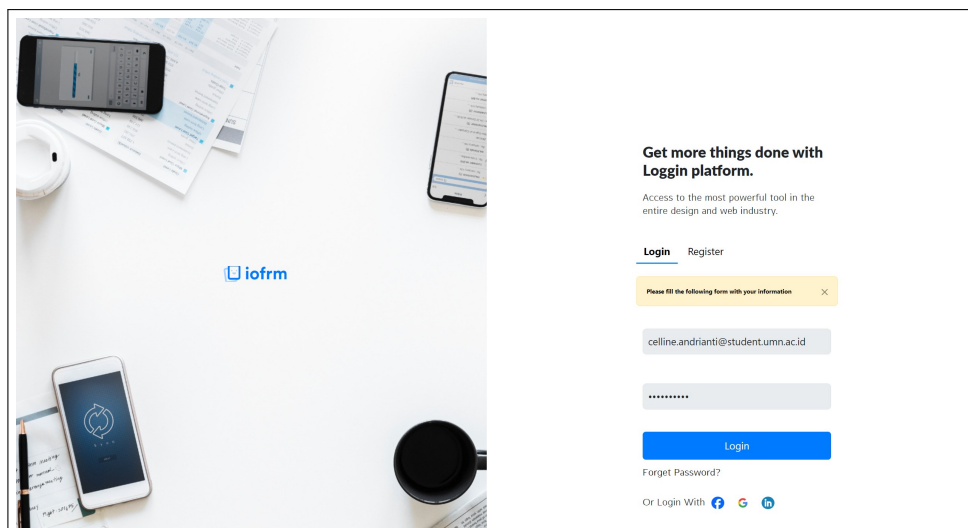
Pada halaman *register*, mengimplementasikan *library zod* dan *formik* untuk menangani validasi pada formulir seperti pada Gambar 3.1 [4]. Pengguna akan diminta untuk mengisi nama lengkap, *email* yang valid, serta *password*. Data akan tersimpan pada *database* dan digunakan pada saat mengecek kredensial proses *login*.



Gambar 3.1. Halaman *Register* Proyek *Training*

B. Halaman *Login*

Pada halaman *login*, mengimplementasikan *library zod* dan *formik* untuk menangani validasi pada formulir seperti pada Gambar 3.2. Pengguna akan diminta untuk mengisi *email* yang valid dan *password*. Sistem akan mengecek kredensial *login*, ketika sesuai dengan *database* maka akan masuk ke halaman *dashboard*.

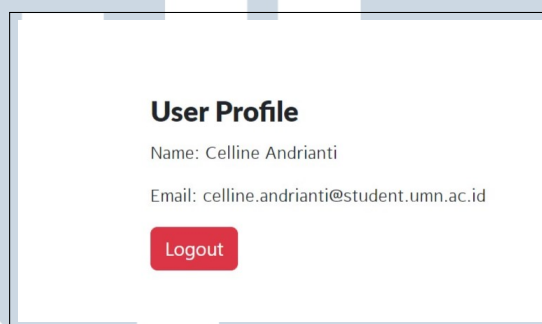


Gambar 3.2. Halaman *Login* Proyek *Training*

C. Halaman *Dashboard*

Pada halaman *dashboard* ini mengimplementasikan *library redux toolkit* untuk memudahkan *handle* dan *passing state*, sehingga ketika pengguna sedang

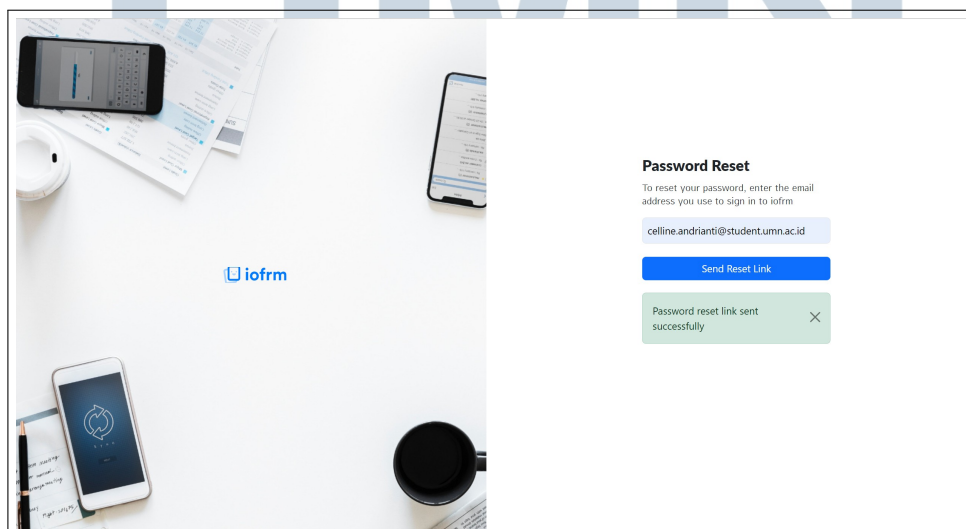
login, pada setiap halaman akan dengan mudah mengambil info pengguna tersebut tanpa perlu *passing props* antar halaman ataupun komponen seperti pada Gambar 3.3. Halaman *dashboard* juga mengimplementasikan *library localStorage* sehingga ketika sudah berhasil *login*, maka nama dan *email* akan tersimpan di *localStorage*, jadi ketika halaman dibuka kita mendapatkan nama dan *email* yang sudah tersimpan di *localStorage* sehingga ketika di-*refresh* tidak hilang [5]. Pada tombol *log out* memiliki fungsi untuk menghapus *state* yang ada di *localStorage* sehingga kembali lagi ke halaman *login*.



Gambar 3.3. Halaman *Login* Proyek *Training*

D. Halaman Lupa *Password*

Pada halaman *dashboard* ini mengimplementasikan *library nodemailer* untuk mengirimkan *email* berisi *URL token* untuk melakukan *reset password* seperti pada Gambar 3.4 [6]. Pengguna akan diminta mengisi *email* yang valid agar bisa mendapatkan *URL token* untuk melakukan *reset password* pada *database*.



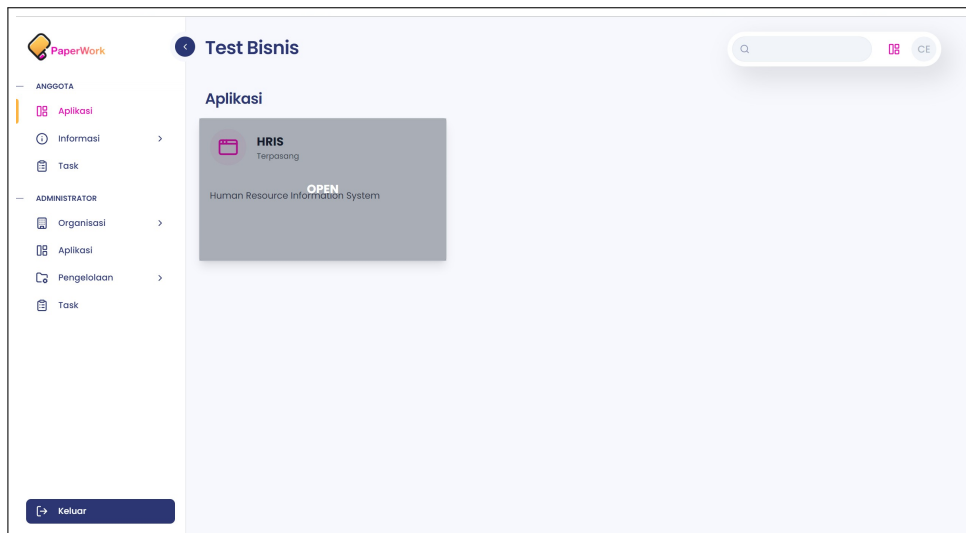
Gambar 3.4. Halaman Lupa *Password* Proyek *Training*

3.3.3 Pengerjaan Website PaperWork

Website PaperWork menjadi proyek utama pada proses magang. Dalam proyek ini terdapat berbagai *library* seperti *nodemailer*, *library tailwind*, *TypeScript*, *Redux Toolkit Query*, *library formik*, *library zod*, *Prisma ORM*, *library express handlebar*. Mentor memberikan tugas untuk melakukan *re-design* beberapa elemen dan menambah fitur baru pada proyek ini. Mentor memberikan tugas *re-design* karena desain awalnya kurang modern. Diberikan tugas untuk menambah fitur baru karena kebutuhan *client* terkait dengan *website* yang akan digunakan untuk mengelola bisnis. Mentor magang mempercayai karyawan magang bagian *programmer* untuk mengerjakan tugas ini karena sudah mempelajari *library-library* yang digunakan dan mempelajari proyek utama. Berikut merupakan tampilan *website* yang sudah dikerjakan.

A. *Re-design Card Aplikasi*

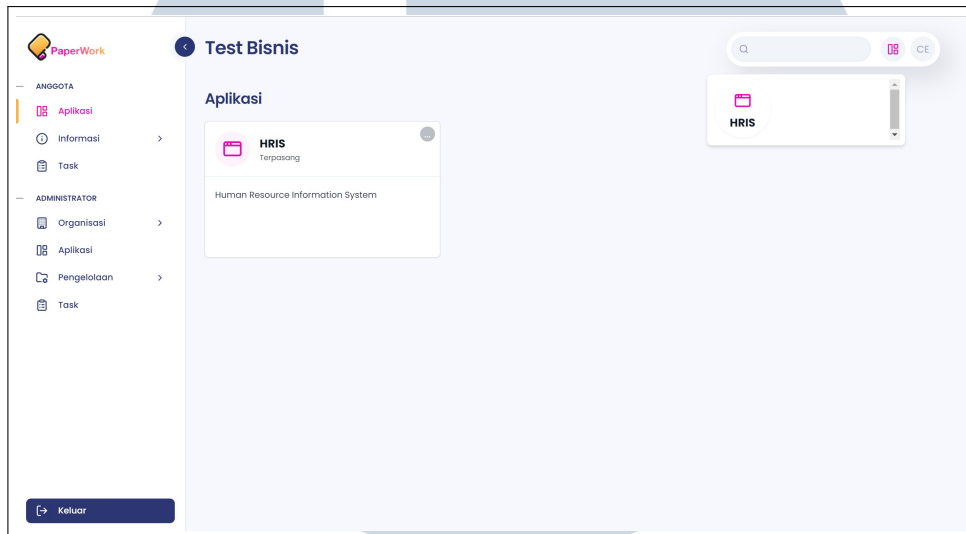
Halaman ini merupakan *dashboard* dari *website PaperWork*. Pada halaman anggota dan administrator telah dilakukan *re-design* terhadap *card* aplikasi agar terlihat lebih modern dengan menambah elemen *popup* dan tambahan teks "*open*" seperti pada Gambar 3.5.



Gambar 3.5. *Re-design Card Aplikasi Website PaperWork*

B. Menambah Fitur Menu Aplikasi

Penambahan fitur pada *navbar* agar ada menu aplikasi yang ketika diklik maka akan berisi aplikasi yang tersedia pada *website PaperWork* seperti pada Gambar 3.6. Ketika logo aplikasi diklik maka akan langsung *direct* ke halaman aplikasi masing-masing.

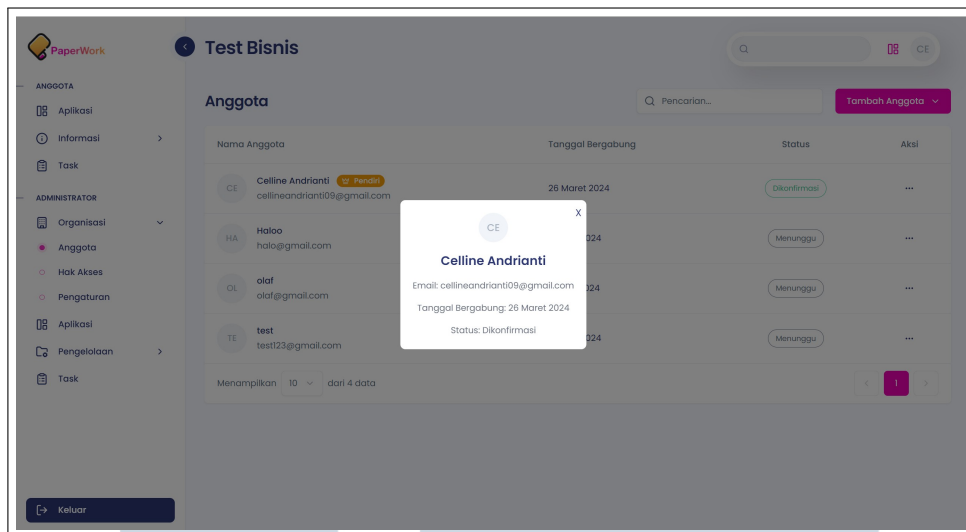


Gambar 3.6. Menambah Fitur Menu Aplikasi *Website PaperWork*

C. *Re-design* Halaman Anggota

Halaman anggota sebelum dilakukan *re-design*, ketika nama diklik maka tidak muncul aksi apapun sehingga dilakukan *re-design* pada saat nama diklik maka akan langsung muncul *popup* dialog seperti pada Gambar 3.7.

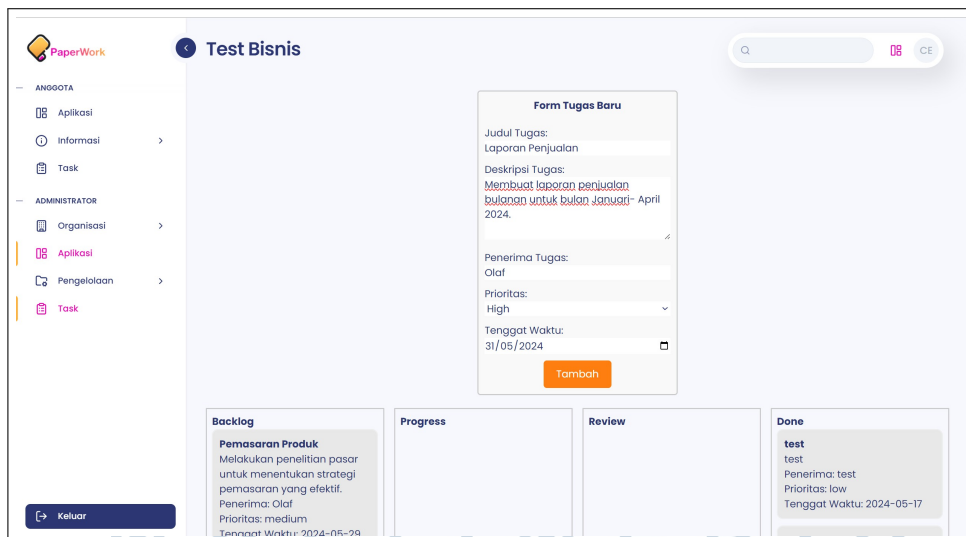
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.7. Re-design Halaman Anggota Website PaperWork

D. Fitur Task Administrator

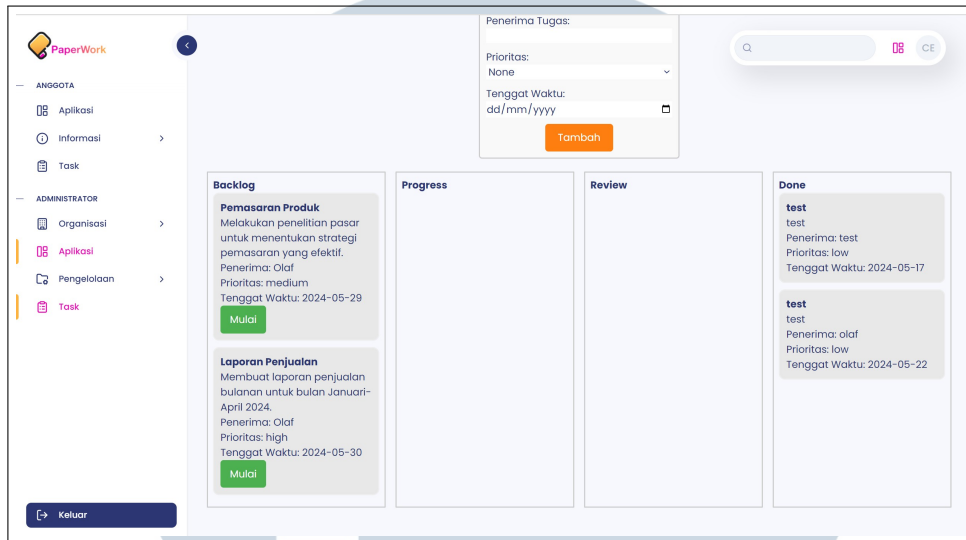
Halaman ini merupakan tampilan yang bisa diakses oleh administrator untuk mengisi tugas yang akan diberikan pada karyawan. Seluruh kolom *input form* sifatnya wajib diisi, jika tidak diisi maka tidak bisa ditambah tugasnya seperti pada Gambar 3.8.



Gambar 3.8. Tampilan Form Tambah Task Website PaperWork

Halaman ini merupakan lanjutan dari tahapan pengisian *form*. Setelah *form* tambah tugas diisi maka tugas akan otomatis berada pada kolom *backlog*. Pada tampilan ini ada empat kolom yaitu *backlog*, *progress*, *review*, dan *done* seperti pada Gambar 3.9. Pada administrator terdapat tombol *done* pada kolom *review* sehingga

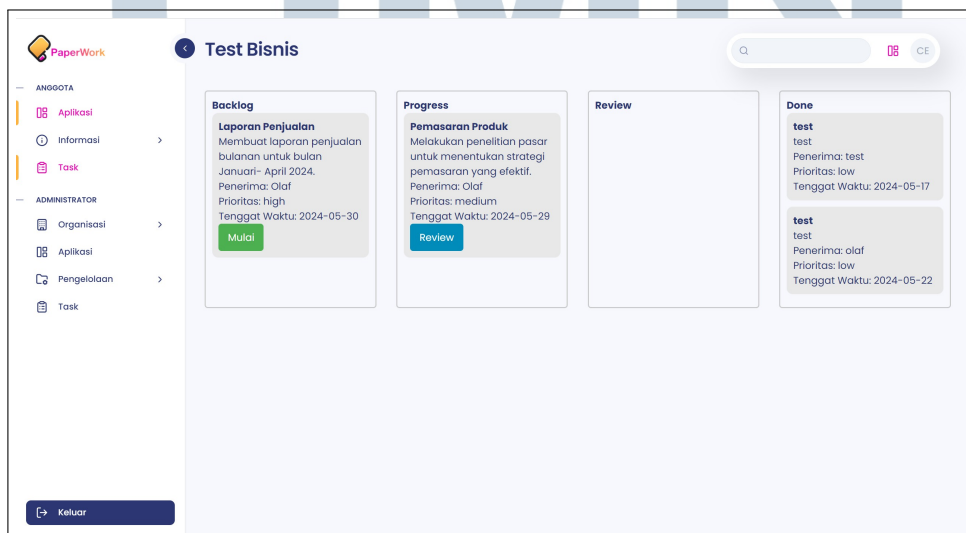
ketika diklik tombol *done* akan pindah ke kolom *done* dan tampilan di anggota juga berubah ke kolom *done*.



Gambar 3.9. Tampilan *List Task* Website *PaperWork*

E. Fitur *Task* Anggota

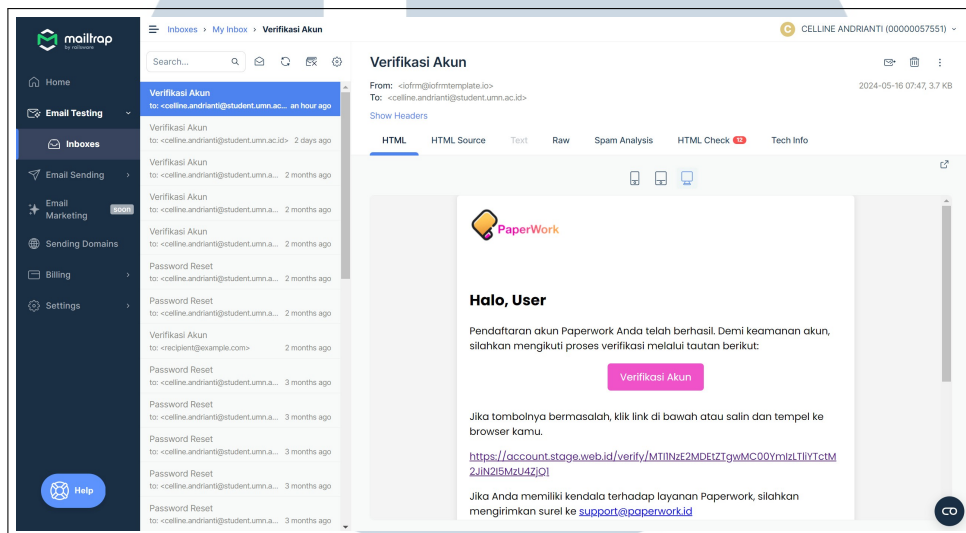
Pada halaman anggota, tampilan tugas juga terbagi menjadi empat kolom yaitu *backlog*, *progress*, *review*, dan *done*. Ketika tugas ada pada kolom *backlog* maka akan ada tombol mulai dan otomatis pindah ke kolom *progress*. Setelah di kolom *progress* maka akan muncul tombol *review* sehingga ketika anggota sudah menyelesaikan tugasnya maka tugas dapat di *review* oleh administrator seperti pada Gambar 3.10.



Gambar 3.10. Fitur *Task* Anggota Website *PaperWork*

F. Re-design Template Email Verifikasi

Pada tampilan Gambar 3.11 dibuat dengan CSS murni agar tidak mengalami perubahan pada perangkat yang digunakan pengguna. Pendaftaran di *website PaperWork*, pengguna wajib memverifikasi akun dengan mengklik tombol atau *link* yang dikirimkan melalui *email* sehingga akun dapat aktif.



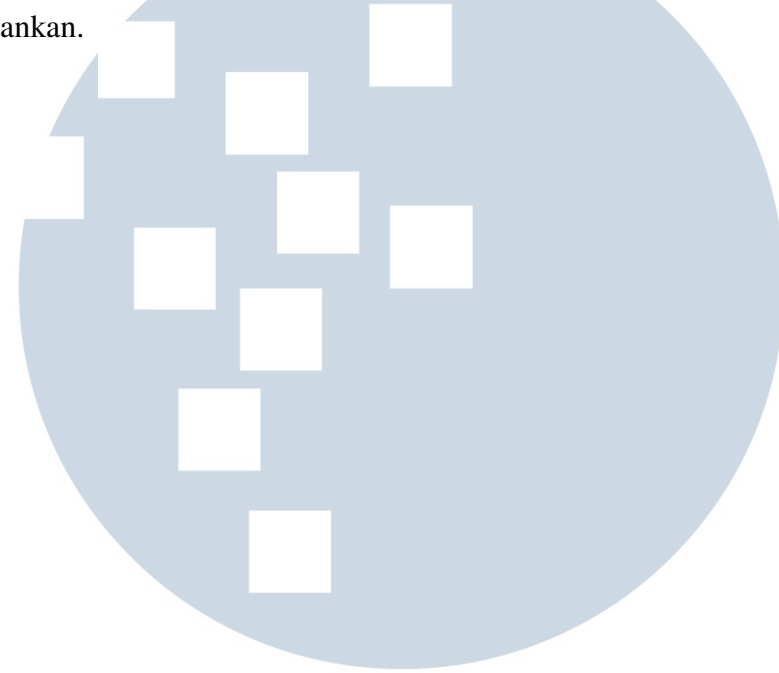
Gambar 3.11. Re-design Template Email Verifikasi

3.4 Kendala dan Solusi yang Ditemukan

Pada masa pelaksanaan magang di perusahaan PT. Pranela Ragam Karya sebagai *software developer* terdapat kendala yang dialami seperti:

- Kendala yang dialami yaitu perbedaan sistem operasi yang digunakan. Sebelumnya menggunakan sistem operasi *Windows*, sedangkan di perusahaan PT. Pranela Ragam Karya menggunakan sistem operasi *Linux*. Kendala dapat diselesaikan karena mendapat solusi dari mentor yaitu dengan cara menginstall *Ubuntu* sehingga *Linux* dapat berjalan di dalam *Windows*.
- Pada proses *training* mengalami kendala pada proyek pembuatan *website login* dan *register*. Ketika proses *login* berhasil dan di-*refresh* maka akan data *user* hilang. Kendala dapat diselesaikan karena mendapat solusi dari mentor yaitu mengimplementasikan *redux-persist* atau *local storage* sehingga ketika proses di-*refresh* dilakukan maka data *user* tidak hilang dan tetap ditampilkan pada halaman *dashboard*.

- Terdapat kendala error ketika proses menjalani proyek utama pertama kali. Setelah bertanya kepada mentor ternyata perlu dilakukan *npm run build* di folder *paperwork-ui* baru setelah itu folder *business-react* diberikan perintah *npm install*. Setelah mengikuti instruksi mentor proyek sudah berhasil dijalankan.



UMMN

UNIVERSITAS
MULTIMEDIA
NUSANTARA