

BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Kedudukan yang diberikan adalah sebagai *Intern Fullstack Developer* dengan tanggung jawab untuk membangun dan mengembangkan *User Interface* dan *User Experience* pada halaman web Helpernet, melakukan pengujian terhadap *User Interface* dan *User Experience* yang telah dibuat, serta melakukan pengujian terhadap API yang telah dibuat. Pekerjaan magang dilakukan di bawah bimbingan dan tanggung jawab kepada klien, direktur, serta Vincentius Marco Melandri.

3.2 Tugas yang Dilakukan

Tugas yang telah dilakukan selama menjalani masa magang di PT. CreateIT Solution meliputi pekerjaan sebagai *Frontend Developer*, yaitu membuat dan melakukan pengujian terhadap UI/UX yang telah dikembangkan, serta pekerjaan sebagai *Backend Developer*, yaitu melakukan pengujian terhadap API termasuk fitur-fitur yang telah dibuat. Berikut adalah rincian tugas yang telah dilakukan saat kerja magang:

- Membuat *User Interface* dan *User Experience* halaman autentikasi (*login* dan *register*).
- Menambahkan fitur multibahasa pada halaman web Helpernet, yaitu bahasa Inggris, German, dan Indonesia.
- Membuat dan mengembangkan *User Interface* dan *User Experience* halaman *Dashboard*.
- Membuat dan mengembangkan *User Interface* dan *User Experience* halaman *Event Overview*.
- Membuat dan mengembangkan *User Interface* dan *User Experience* halaman *User Management*.
- Membuat dan mengembangkan *User Interface* dan *User Experience* halaman *Helper Management*.

- Membuat dan mengembangkan *User Interface* dan *User Experience* halaman *Helper Work Hour Report*.
- Membuat dan mengembangkan *User Interface* dan *User Experience* halaman *Settings*.
- Melakukan testing *API* untuk *Authentication* (*login* dan *register*).
- Melakukan testing *API* untuk fitur *Create*, *Read*, *Update*, dan *Delete event* atau proyek.

3.3 Uraian Pelaksanaan Magang

Pelaksanaan kerja magang dijelaskan melalui pembagian mingguan yang disusun pada Tabel 3.1, yang memuat rincian pekerjaan yang dilakukan setiap minggu selama masa kerja magang.

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

Minggu Ke-	Pekerjaan yang Dilakukan
1	Perkenalan lingkungan kantor, pengenalan proyek HR App. Pengenalan fitur HR App seperti absensi, 2FA, registrasi user. Desain strategi autentikasi dan JWT authentication. Implementasi JWT dan datatable pada mini project.
2	Self Learning: setup microservice dan docker-compose. Developing landing page, auth page, toast notification. Mulai pengembangan HR Project, login/register tenant/user. Finalisasi frontend/backend auth service. Finalisasi frontend/backend: fitur reset password.
3	Mulai dashboard, section project (frontend). Pengembangan backend shift. Finalisasi autentikasi + <i>bugfix</i> .
4	Perbaikan UI frontend, API component, refresh token. Penyelesaian Shift Frontend. Finishing tenant shift. Build frontend tenant lease. Lanjutan tenant lease frontend.
	Lanjutan pada halaman berikutnya

Minggu Ke-	Pekerjaan yang Dilakukan
5	Revamp UI autentikasi. Fix payload, assign form, major UI fix. Perubahan assign form, <i>bug</i> fix. Update ShiftForm dan assign form. Mulai proyek baru, FE auth page.
6	Lanjut proyek baru (Helpernet), meeting, <i>bug</i> check.
7	Lanjut proyek baru (Helpernet), meeting, <i>bug</i> check.
8	Lanjut proyek baru (Helpernet), meeting, <i>bug</i> check. Kerjakan halaman event. Lanjut halaman event, <i>bug</i> check.
9	Lanjut halaman event, <i>bug</i> check. Penyesuaian halaman event overview & event-tab. Lanjut event-tab: department group.
10	Lanjut event-tab: department group. Multilanguage event-tab. Bugfix dan pengujian department group (Cypress).
11	CRUD helper dengan multilanguage. Bugfix form worker. Refactor state dan auto-refresh setelah create. Perbaiki fitur search helper.
12	Update department group list. Revamp event-tab. Fix payload create department dan <i>bug</i> lainnya di helper/event-tab.
13	Lanjut pengembangan event-tab. Refactor department dialog. Fix minor helper & shift. Awal halaman dashboard.
14	Fix dashboard dan manajemen user. Feedback halaman dari Thomas. Edit helper. Bugfix dan form reset.
15	Enhancement halaman helper & event. Work area di event-tab. Update shift edit. Validasi form dan input.
16	Revamp dashboard. Bugfix assign shift. Selesai fitur ubah password. Settings SMTP & multilanguage.
17	Fitur pencarian shift assign. Backend testing (testify). Refactor halaman helper & available dates. Fix SMTP.
18	Refactor register form & timezone shift. Laporan total jam kerja helper. CRUD item-list department group.
19	Integrasi item-list ke event. Tampilan tabel multibahasa. Infinite scroll. Halaman laporan helper selesai.

3.4 Dokumentasi Hasil Kerja Magang

Pada bagian ini ditampilkan spesifikasi sistem, alur kerja sistem, dan implementasi antarmuka pengguna *User Interface* dan pengalaman pengguna *User Experience* pada halaman web Helpernet.

3.4.1 Spesifikasi sistem Helpernet

Sistem Helpernet secara keseluruhan dibangun dengan spesifikasi teknis sebagai berikut:

- **Frontend:** Next.js TypeScript dan TailwindCSS untuk pengembangan antarmuka yang responsif dan modern.
- **Backend:** Bahasa pemrograman Go dengan menggunakan *framework* Fiber untuk membangun RESTful API yang cepat dan efisien.
- **Database:** MySQL sebagai sistem manajemen basis data relasional untuk menyimpan dan mengelola data aplikasi.
- **Link Produksi:** <https://helpernet.lagilapar.com>

3.4.2 Kebutuhan Perangkat Keras dan Lunak

Bagian ini menjelaskan spesifikasi perangkat keras dan lunak minimum maupun yang disarankan untuk melakukan proses pengembangan dan menjalankan sistem Helpernet.

A Perangkat Keras (Hardware Requirements)

Komponen	Minimum	Disarankan
CPU	Intel Core i3 / Ryzen 3	Intel Core i5 / Ryzen 5 ke atas
RAM	4 GB	8 GB atau lebih
Storage	1GB (kode dan database)	Diatas 1GB

Tabel 3.2. Spesifikasi Minimum dan Disarankan Perangkat Keras

Tabel 3.2 menunjukkan kebutuhan perangkat keras minimum dan disarankan untuk mengembangkan serta menjalankan sistem Helpernet secara lokal maupun di lingkungan server.

B Perangkat Lunak (Software Requirements)

Komponen	Versi atau Kebutuhan
Node.js	v18.x atau lebih baru [3]
NPM / PNPM	NPM v9+ atau PNPM v8+
Go	v1.20 atau lebih baru [4]
MySQL	v8.x [5]
Sistem Operasi	Windows 10 / Linux Ubuntu 20.04+ / macOS
Git	Terpasang (untuk version control)
Code Editor	Visual Studio Code (disarankan)
Browser	Google Chrome / Microsoft Edge / Safari

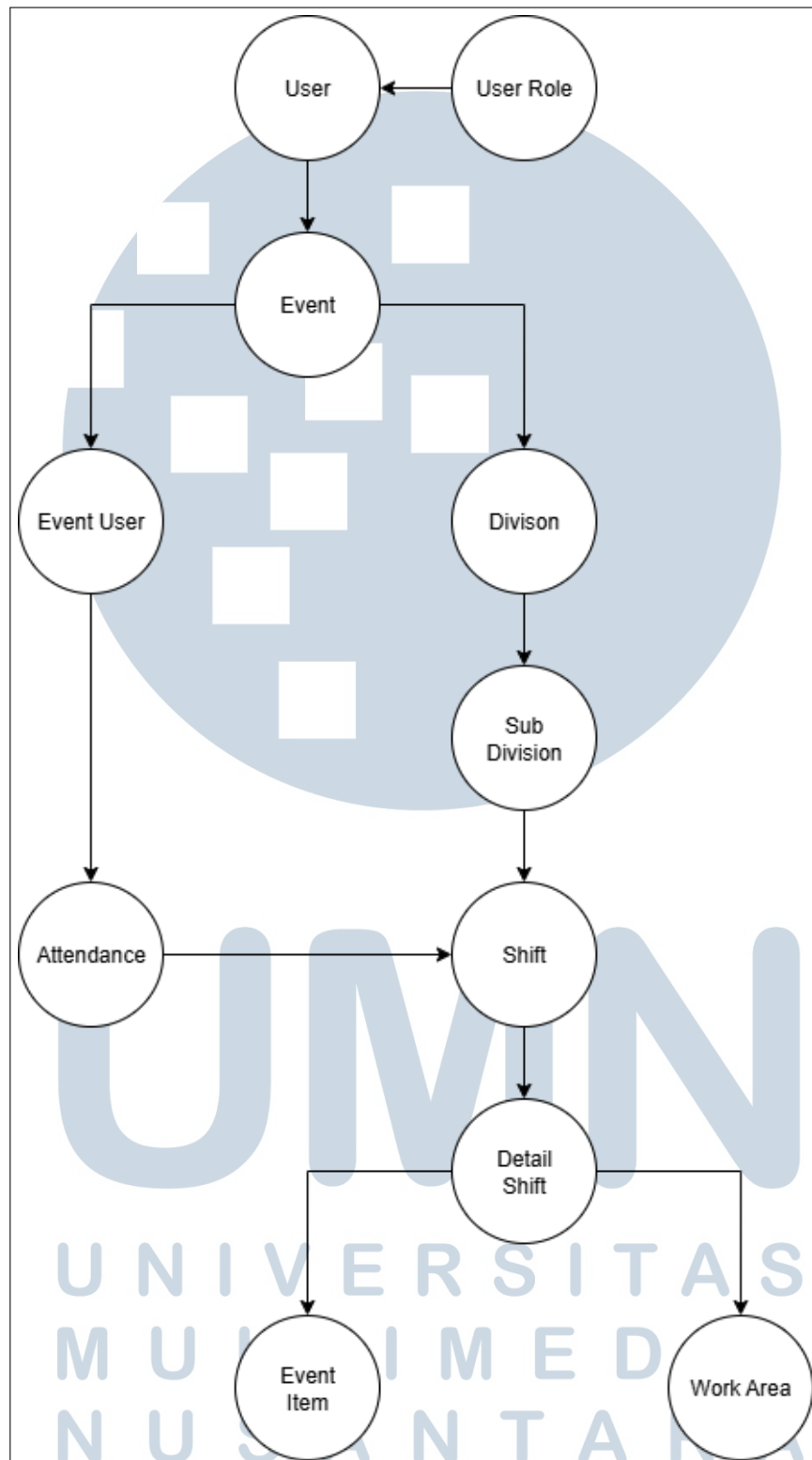
Tabel 3.3. Spesifikasi Perangkat Lunak yang Dibutuhkan

Tabel 3.3 merangkum perangkat lunak utama yang diperlukan dalam proses pengembangan dan eksekusi sistem Helpernet.

3.4.3 Alur Kerja Aplikasi

Bagian ini menjelaskan alur kerja aplikasi Helpernet berdasarkan modul utama sistem. Setiap alur kerja dijelaskan melalui diagram flowchart yang menggambarkan proses utama dalam modul tersebut.

UMMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.1. Diagram alur sistem Helpernet

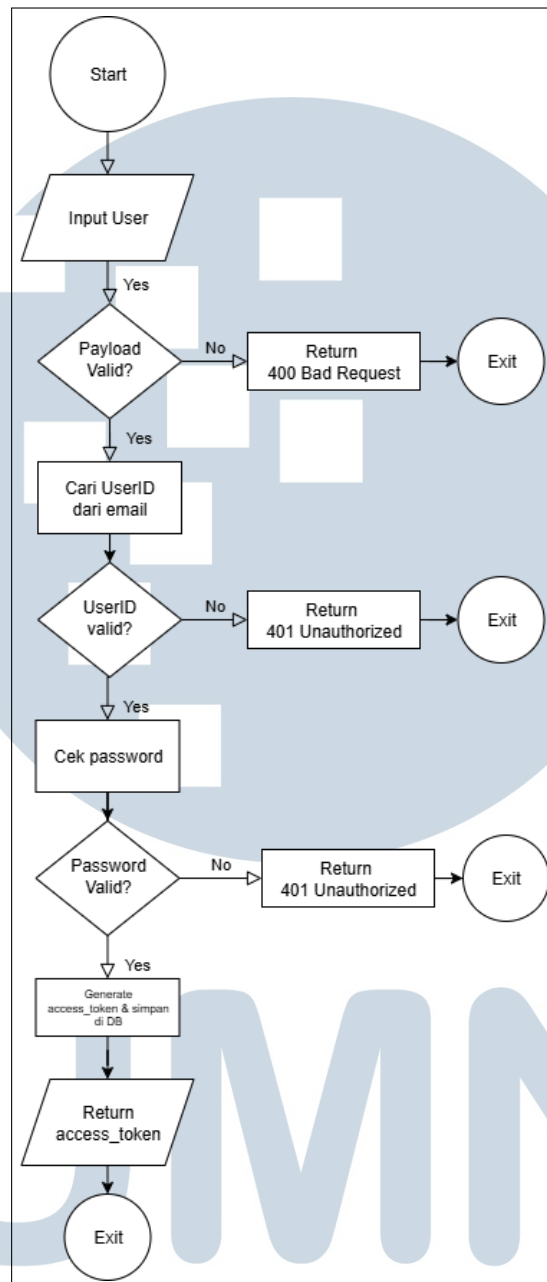
Gambar 3.1 memberikan gambaran visual hubungan antar entitas dalam sistem, seperti User, Event, Department, Shift, dan Absensi Helper.

1. **Login:** Pengguna wajib login terlebih dahulu ke dalam sistem menggunakan akun yang telah terdaftar. Proses autentikasi dilakukan untuk memastikan identitas dan hak akses pengguna.
2. **Pembuatan Event atau proyek:** Setelah berhasil login, pengguna dapat membuat *event* atau proyek baru. Pembuatan event menjadi langkah awal sebagai wadah seluruh konfigurasi seperti departemen, shift, dan panitia.
3. **Pembuatan Department Group dan Department:** Setelah event dibuat, pengguna perlu menyusun struktur organisasi proyek dengan membuat *department group*, kemudian menambahkan *department* di dalamnya. Struktur ini bertujuan untuk mengelompokkan area kerja dalam event secara sistematis.
4. **Pembuatan Shift:** Setiap department yang telah dibuat dapat memiliki satu atau lebih *shift* kerja. Shift ini bersifat fleksibel dan dapat diatur berbeda antar department sesuai kebutuhan operasional.
5. **Penugasan Helper:** Setelah shift dibuat, panitia atau *helper* yang telah terdaftar dapat ditugaskan atau di-*assign* ke shift tertentu. Penugasan ini disesuaikan dengan department tempat helper tersebut tergabung.
6. **Absensi Helper:** Helper yang telah mendapatkan shift akan melakukan proses *check-in* dan *check-out* sebagai bentuk absensi, sesuai dengan jadwal yang telah ditetapkan. Data absensi ini akan tercatat dalam sistem dan dapat dimonitor oleh admin event.

Alur tersebut dirancang untuk memastikan seluruh proses pengelolaan event berjalan sistematis dan efisien, serta mempermudah pengawasan terhadap jam kerja dan keaktifan panitia selama pelaksanaan proyek.

A Alur Kerja *Login*

Berikut adalah *flowchart* atau alur kerja sistem untuk proses *login*:

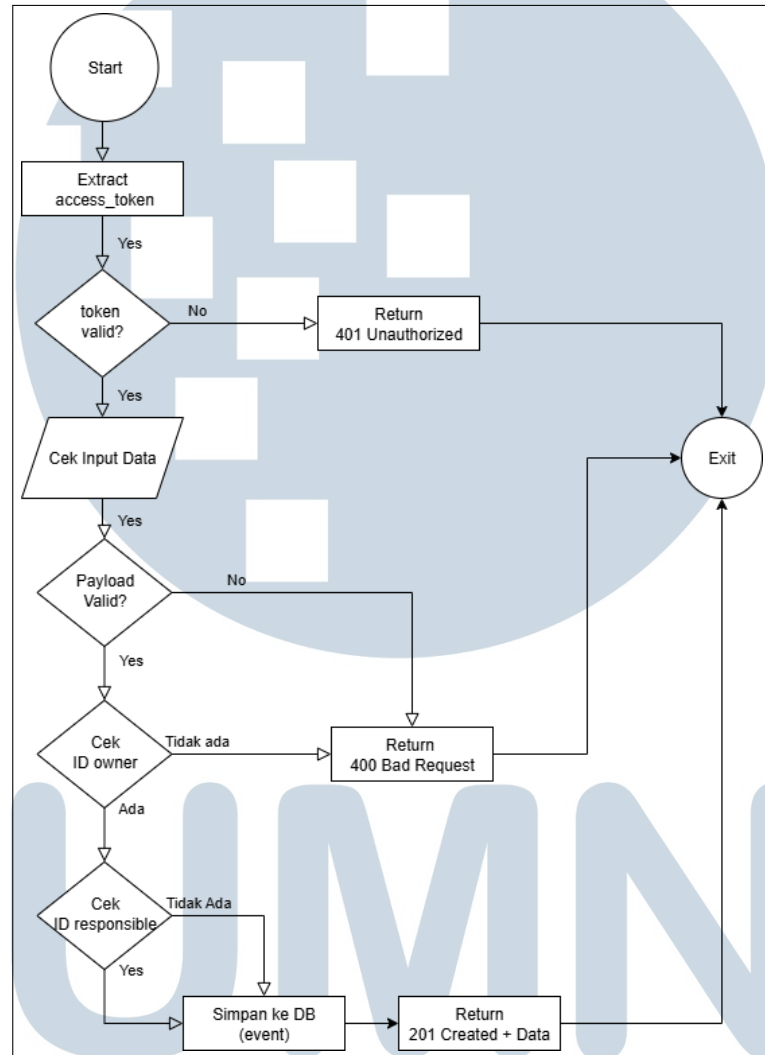


Gambar 3.2. Tampilan *flowchart* proses login

Gambar 3.2 menunjukkan alur proses login. Sistem akan menerima input data, melakukan validasi terhadap data tersebut, dan memastikan bahwa pengguna terdaftar. Jika seluruh proses validasi berhasil, sistem akan menghasilkan *access token* dan mengirimkannya kepada pengguna sebagai bagian dari respons berhasil.

B Alur Kerja Membuat Proyek

Berikut adalah *flowchart* atau alur kerja sistem untuk pembuatan proyek baru:

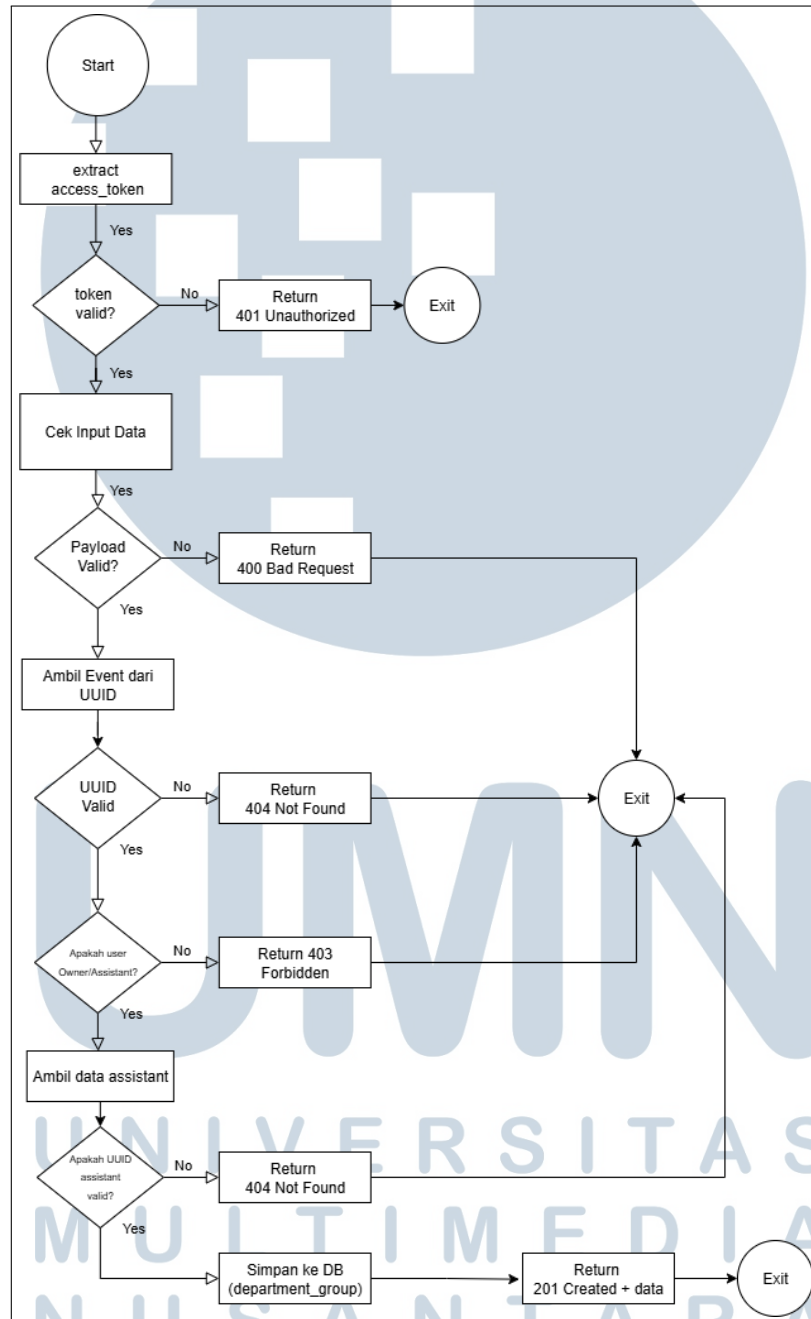


Gambar 3.3. Tampilan *flowchart* untuk *create project*

Gambar 3.3 menampilkan alur proses pembuatan proyek. Sistem akan menerima input data event, melakukan validasi terhadap data tersebut, dan memastikan bahwa pengguna memiliki otorisasi. Jika semua validasi berhasil, data proyek akan disimpan ke dalam basis data dan sistem mengembalikan respons berhasil berupa status 201 beserta data proyek yang telah dibuat.

C Alur Kerja *Create Department Group*

Berikut adalah *flowchart* atau alur kerja sistem untuk proses pembuatan *Department Group*:



Gambar 3.4. Tampilan *flowchart* proses pembuatan *Department Group*

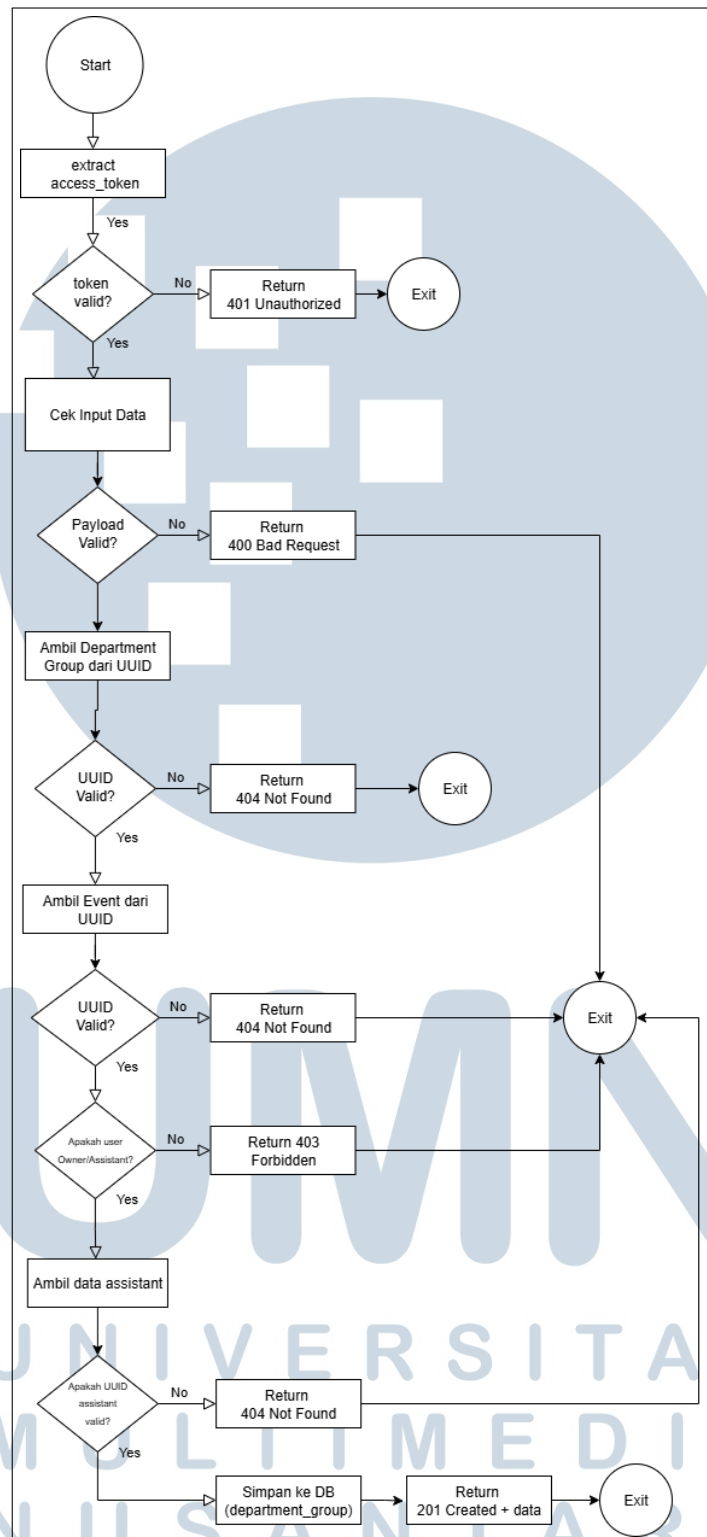
Gambar 3.4 menunjukkan alur proses pembuatan *Department Group*. Sistem akan memverifikasi token pengguna, memvalidasi data input, dan

memastikan bahwa pengguna memiliki hak sebagai pemilik atau asisten event. Jika UUID yang diberikan valid dan, apabila ada, pengguna yang bertanggung jawab juga valid, maka data akan disimpan ke dalam basis data, dan sistem akan mengembalikan respons 201 Created beserta data yang telah dibuat.

D Alur Kerja *Create Department*

Berikut adalah *flowchart* atau alur kerja sistem untuk proses pembuatan *Department*:





Gambar 3.5. Tampilan *flowchart* proses pembuatan *Department*

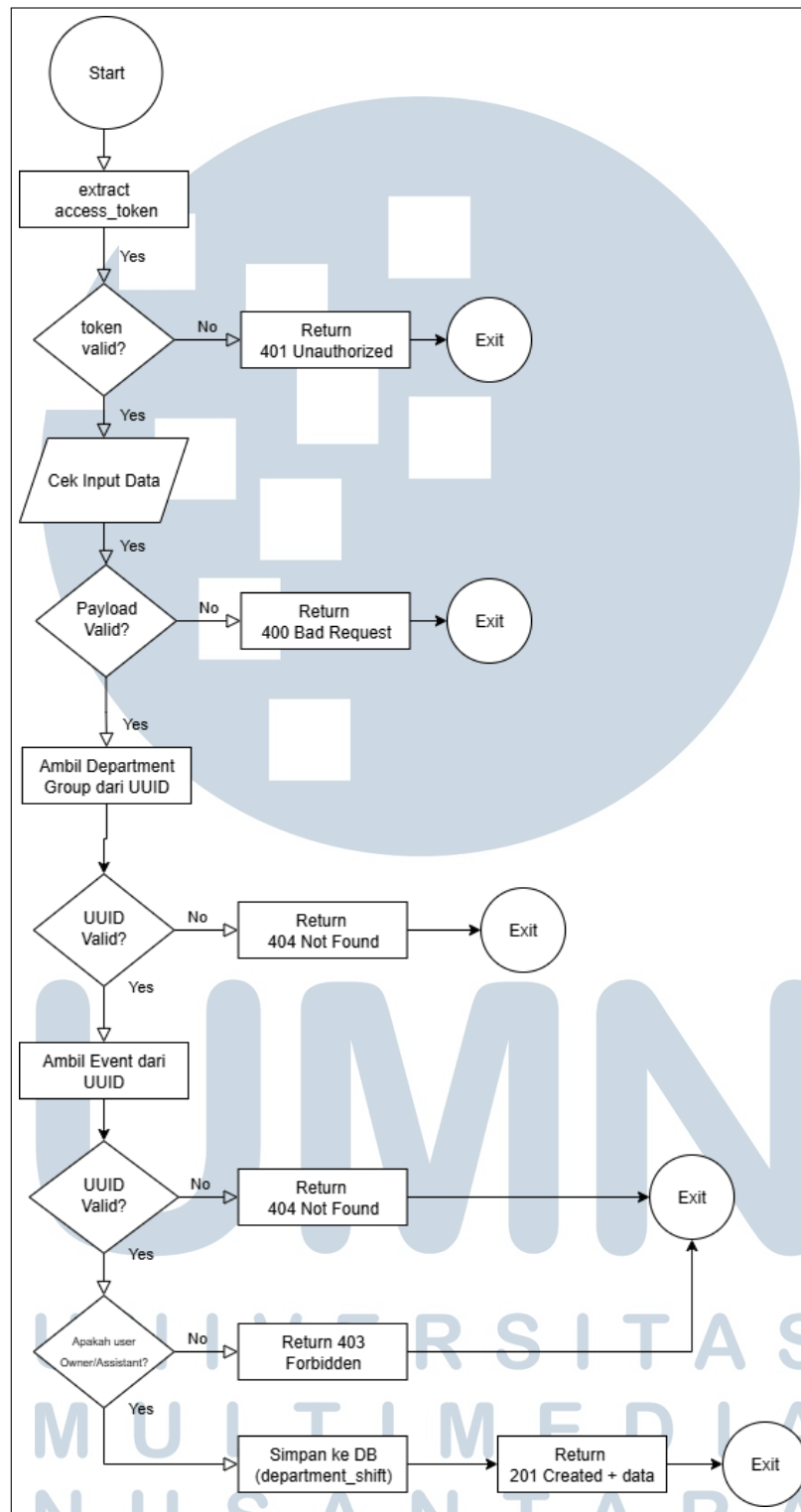
Gambar 3.5 menjelaskan alur proses pembuatan *Department*. Setelah

token diverifikasi dan data payload tervalidasi, sistem akan memastikan bahwa Department Group serta Event yang terkait benar-benar ada dan bahwa pengguna memiliki otorisasi. Jika semua validasi berhasil, dan jika diberikan, UUID dari pengguna yang bertanggung jawab juga valid, maka data Department akan disimpan ke dalam basis data dan sistem akan memberikan respons 201 Created beserta data hasil input.

E Alur Kerja Pembuatan Department Shift

Berikut adalah *flowchart* atau alur kerja sistem untuk proses membuat *department shift* di *project* terkait:





Gambar 3.6. Tampilan flowchart untuk *create department shift*

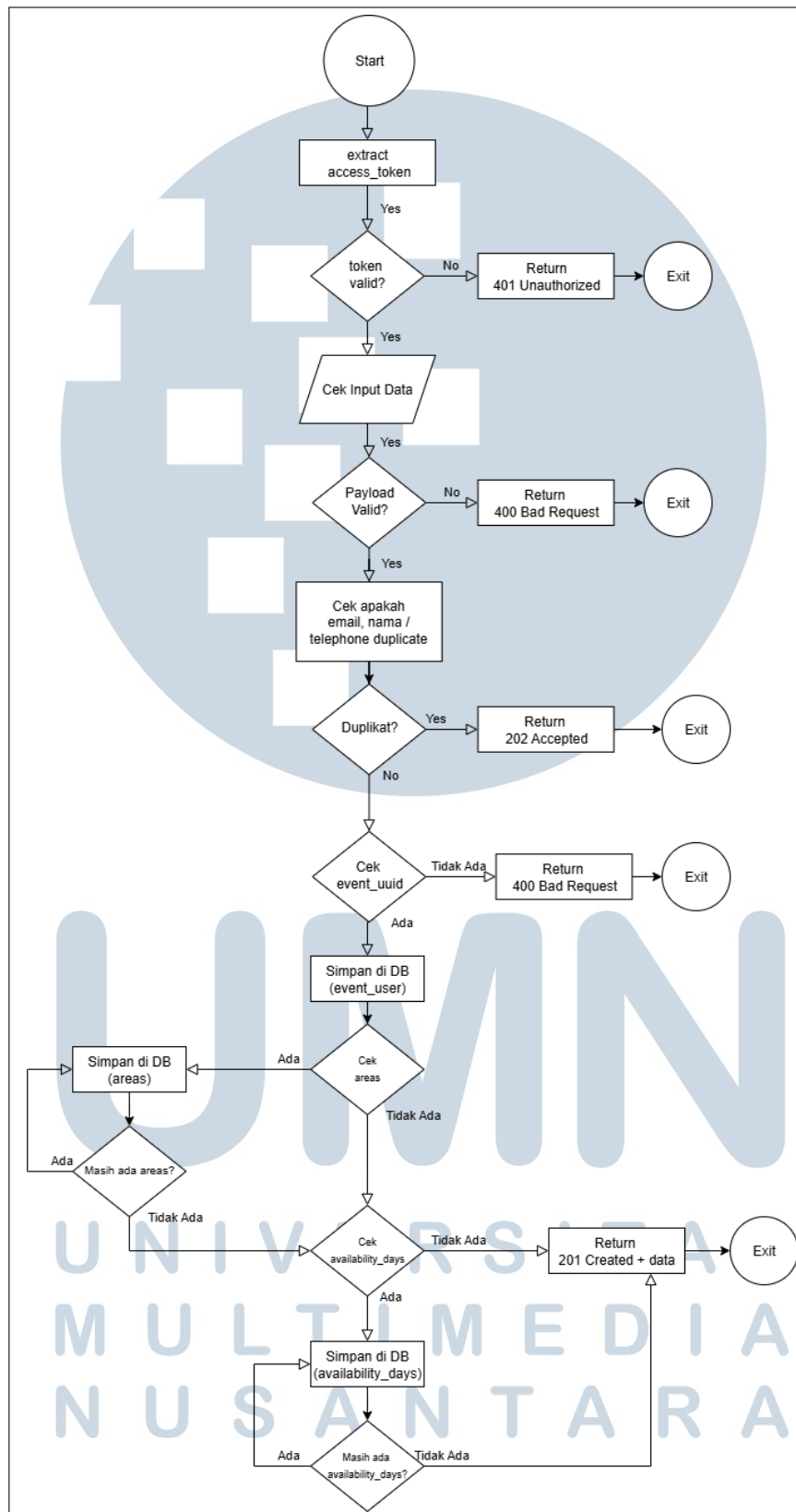
Gambar 3.6 menampilkan alur proses pembuatan *department shift*. Sistem akan mengekstrak *token* dari pengguna, memverifikasi *token*, dan melakukan

validasi terhadap data *payload* yang dikirim. Setelah itu, sistem akan mengambil data *department*, *group*, dan *event* yang terkait berdasarkan UUID. Selanjutnya, sistem akan memastikan apakah pengguna memiliki hak akses sebagai pemilik atau asisten *event*. Jika seluruh proses validasi berhasil, maka data shift akan disimpan ke dalam basis data dan sistem akan mengembalikan respons berhasil berupa status 201 beserta data *shift* yang telah dibuat.

F Alur Kerja Membuat Admin

Berikut adalah *flowchart* atau alur kerja sistem untuk proses membuat admin di *project* terkait:



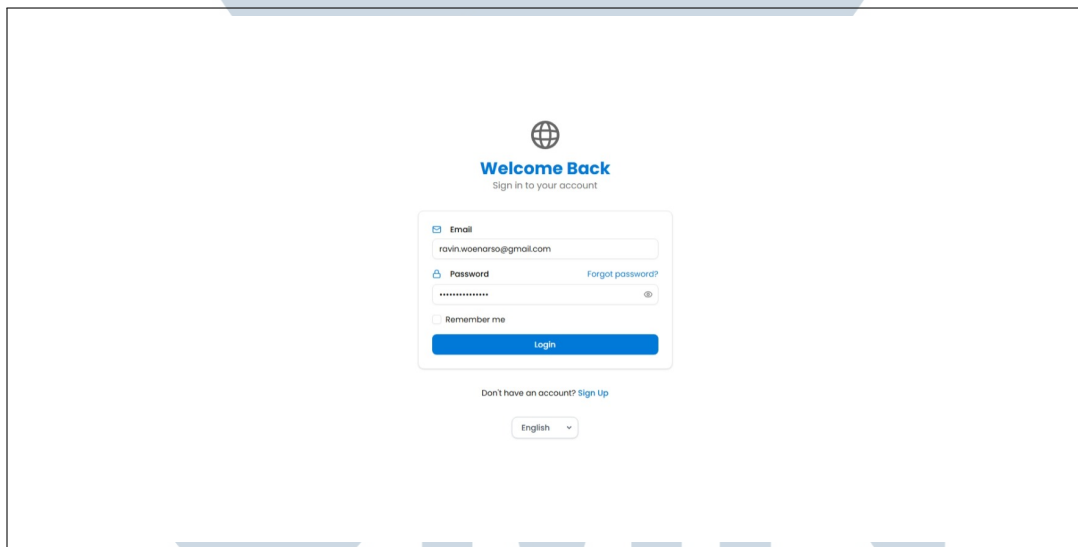


Gambar 3.7. Tampilan *flowchart* untuk proses pembuatan admin

Gambar 3.7 menampilkan alur proses pembuatan pengguna secara manual oleh admin. Sistem akan menerima input data pengguna, melakukan validasi terhadap data tersebut, dan memastikan bahwa pengguna belum terdaftar sebelumnya. Jika seluruh proses validasi berhasil, data pengguna akan disimpan ke dalam basis data beserta relasinya terhadap event, area, dan hari ketersediaan. Sistem kemudian akan mengembalikan respons berhasil berupa status 201 beserta data pengguna yang telah dibuat.

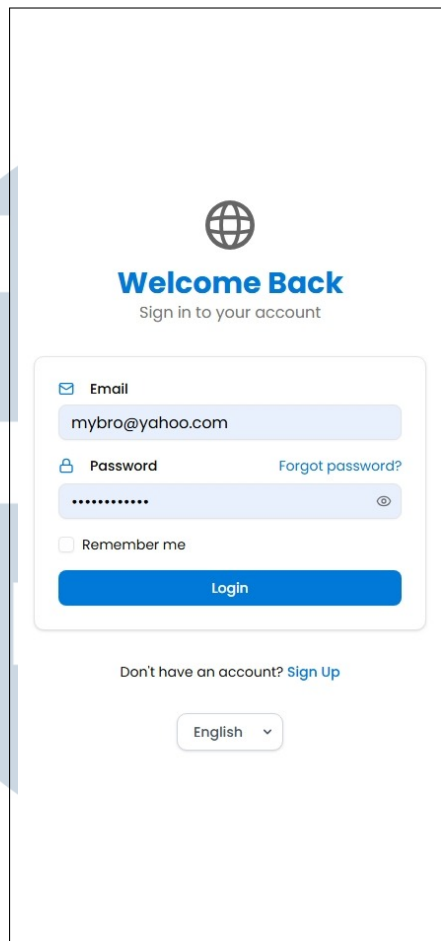
3.4.4 Halaman Autentikasi

Halaman autentikasi merupakan bagian awal dari sistem yang berfungsi untuk mengelola proses *login* dan registrasi pengguna. Halaman yang dibuat sudah mencakup validasi form, dukungan *view* untuk tampilan *mobile (responsive)*, serta integrasi dengan backend untuk proses autentikasi pengguna.



Gambar 3.8. Tampilan halaman *login* versi *desktop*

Gambar 3.8 menampilkan tampilan halaman *login* untuk versi *desktop* di halaman web Helpernet. Untuk *login* ke halaman web Helpernet, pengguna harus memasukkan informasi email dan password yang terdaftar di database Helpernet.



Gambar 3.9. Tampilan halaman *login* versi *mobile*

Gambar 3.9 menampilkan tampilan halaman *login* untuk versi *mobile* di halaman web Helpernet. Sama seperti tampilan *desktop*, tampilan halaman *login* versi ini mengharuskan pengguna untuk memasukkan informasi email dan password yang terdaftar.

Untuk fitur *register* di halaman web Helpernet, menggunakan sistem *stepper*, dan membagi menjadi 4 langkah, yaitu sebagai berikut:

Gambar 3.10. Tampilan halaman *Register* - Step 1

Gambar 3.10, menampilkan halaman registrasi dengan sistem *stepper*, di tahap yang pertama yaitu *Personal Information*. Di halaman ini pengguna harus memasukkan *first name*, *last name*, email yang digunakan, *gender*, dan tanggal lahir.

Gambar 3.11. Tampilan halaman *Register* - Step 2

Gambar 3.11, menampilkan halaman registrasi dengan sistem *stepper*, di tahap yang kedua yaitu *Contact Information*. Di halaman ini pengguna harus memasukkan kode negara beserta nomor handphone.

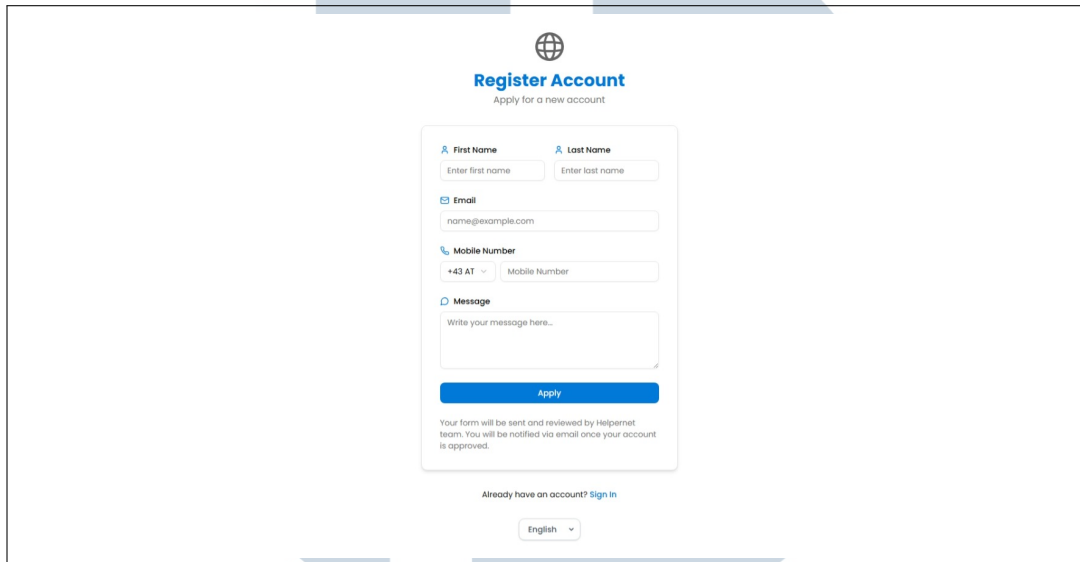
Gambar 3.12. Tampilan halaman *Register* - Step 3

Gambar 3.12, menampilkan halaman registrasi dengan sistem *stepper*, di tahap yang ketiga yaitu *Address Information*. Di halaman ini pengguna harus memasukkan *city*, *address*, dan *zipcode*.

Gambar 3.13. Tampilan halaman *Register* - Step 4

Gambar 3.13, menampilkan halaman registrasi dengan sistem *stepper*, di tahap yang keempat yaitu *Security Information*. Di halaman ini pengguna harus memasukkan *password* yang akan digunakan, dan mengonfirmasi *password* yang sudah dimasukkan.

Untuk yang versi terbaru, halaman registrasi diminta untuk diperbarui agar mencegah penggunaan aplikasi halaman web Helpernet oleh pihak yang tidak berwenang. Tampilan halaman registrasi yang telah diperbarui ditunjukkan pada Gambar 3.14.

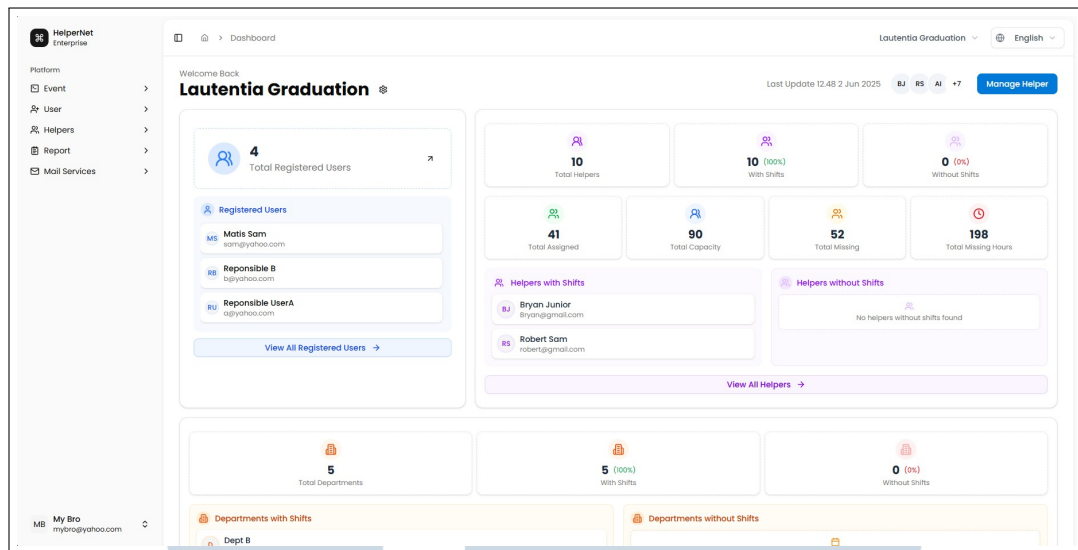


Gambar 3.14. Tampilan halaman *Register* yang baru

Gambar 3.14 menampilkan tampilan halaman registrasi yang baru. Di halaman tersebut, pengguna diharuskan mengisi *first name* dan *last name*, alamat email, kode negara beserta nomor *handphone*, dan alasan atau keperluan pengguna untuk menggunakan halaman web Helpernet.

3.4.5 Halaman Dashboard

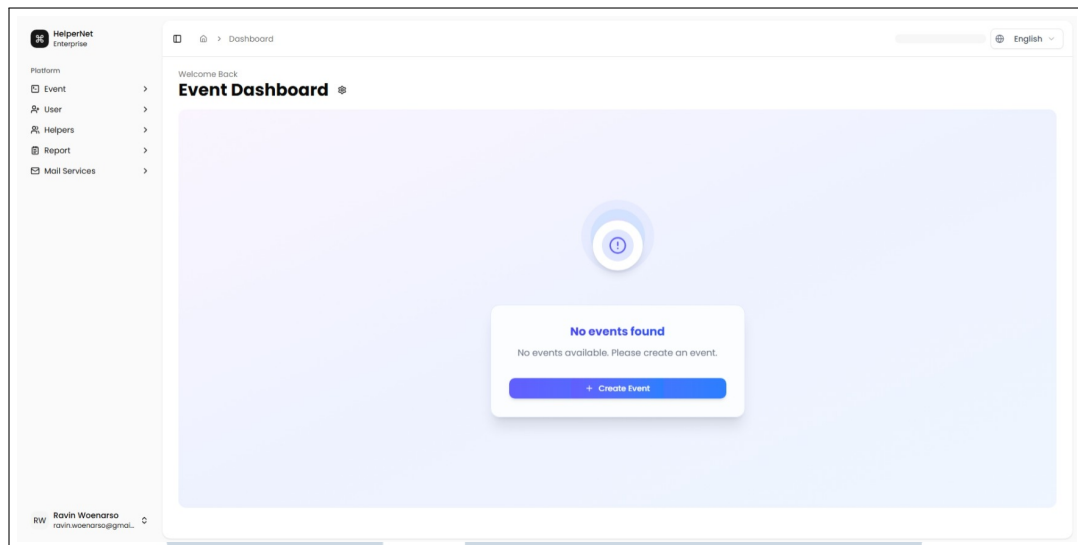
Halaman dashboard berfungsi sebagai pusat informasi utama yang menampilkan ringkasan data terkait *User*, *Helper*, *Department* dan *Shift* dari *event* atau proyek yang telah dibuat dan dipilih. Melalui halaman dashboard, pengguna dapat memantau statistik penting secara menyeluruh untuk *event* atau proyek yang sedang dipilih, seperti jumlah *User* yang sudah didaftarkan, status *shift*, dan yang lainnya.



Gambar 3.15. Tampilan halaman Dashboard - Bagian 1

Gambar 3.15 menampilkan halaman dashboard yang memberikan informasi dasar terkait *event* atau proyek yang sedang dipilih. Informasi dasar yang diberikan adalah sebagai berikut:

- Jumlah *user* yang ada di *event* atau proyek yang sedang dipilih.
- Jumlah panitia atau *helper*, jumlah *helper* yang memiliki dan yang tidak memiliki *shift* atau jadwal kerja, total kapasitas *shift*, total *shift* yang masih kekurangan *helper*, dan total jam kerja dari *shift* yang kekurangan *helper* tersebut.
- Jumlah *department*, total *department* dengan *shift*, dan yang tidak memiliki *shift*.
- *Shift* yang telah dibuat, mencakup informasi tanggal *shift*, jam mulai dan selesai, kapasitas *shift*, serta jumlah slot dan jam kerja yang masih kurang.



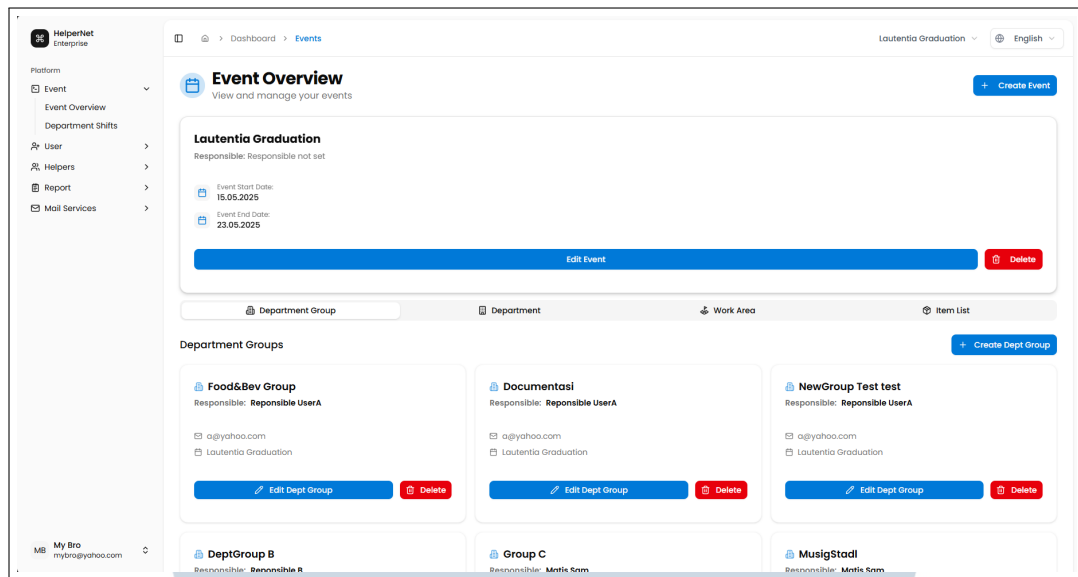
Gambar 3.16. Tampilan halaman Dashboard - Tanpa *Event*

Jika pengguna baru pertama kali masuk ke halaman web Helpernet, dan belum membuat *event* atau proyek sama sekali, maka tampilan dashboard akan seperti gambar 3.16.

3.4.6 Halaman *Event Overview*

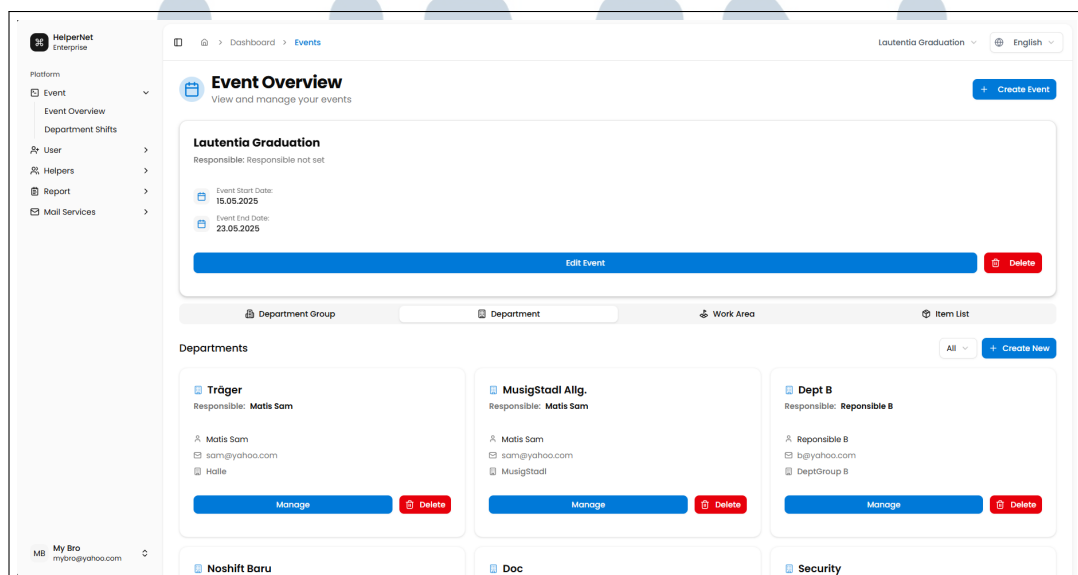
Halaman *Event Overview* merupakan halaman yang menampilkan konfigurasi serta struktur dari *event* atau proyek yang sudah dibuat dan dipilih. Pada halaman ini terdapat informasi mengenai *event*, dengan beberapa tab utama, seperti *Department Group*, *Department*, dan *Event Working Area*, dan *Event Item List* yang masing-masing berfungsi untuk mengelola pembagian kelompok kerja, departemen, area kerja, dan barang barang yang dibutuhkan dari *event*.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.17. Tampilan halaman Dashboard - Tab *Department Group*

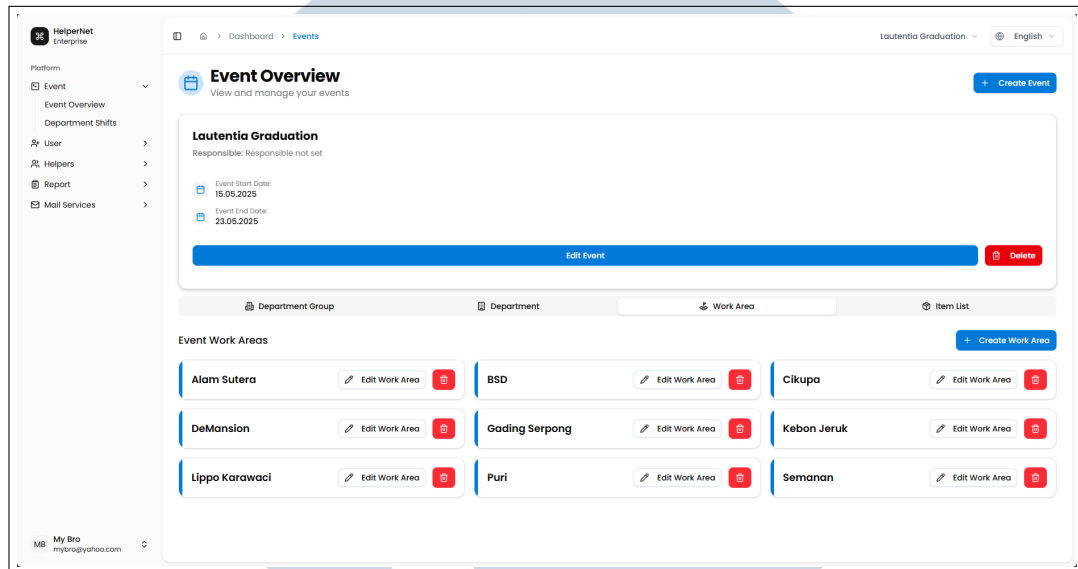
Gambar 3.17 menampilkan halaman *Event Overview*, dengan tab *Department Group* yang ada di *Event* tersebut. Halaman tersebut menampilkan informasi *Event*, seperti nama, orang yang bertanggung jawab, tanggal mulai dan tanggal selesai. Selain informasi *Event*, informasi *Department Group* juga ditampilkan, seperti nama *Department Group*, nama dan email orang yang bertanggung jawab.



Gambar 3.18. Tampilan halaman Dashboard - Tab *Department*

Gambar 3.18 menampilkan halaman *Event Overview*, dengan tab *Department*

Department yang ada di *event* tersebut. Tab dibawah memberikan informasi *department*, seperti nama *department*, nama dan email orang yang bertanggung jawab atas *department* tersebut, dan informasi asal *Department Group*.



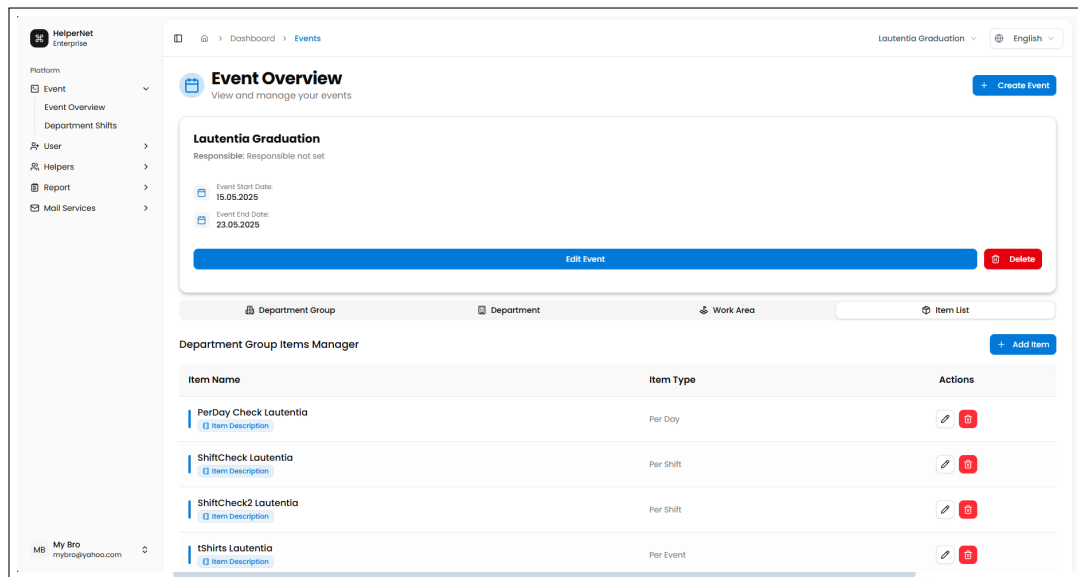
Gambar 3.19. Tampilan halaman Dashboard - Tab *Event Work Area*

Gambar 3.19 menampilkan halaman *Event Overview*, dengan tab *Work Area* yang ada di *event* tersebut. Tab dibawah memberikan informasi *Work Area*, seperti nama tempat yang akan digunakan atau yang tersedia di *event* yang dipilih.

3.4.7 Tab *Item List Management*

Tab *Item List Management* berfungsi untuk mengelola *item-item* atau barang barang apa saja yang dibutuhkan di *event* yang dipilih, yang nanti akan digunakan saat *helper* melakukan absensi dilapangan.

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

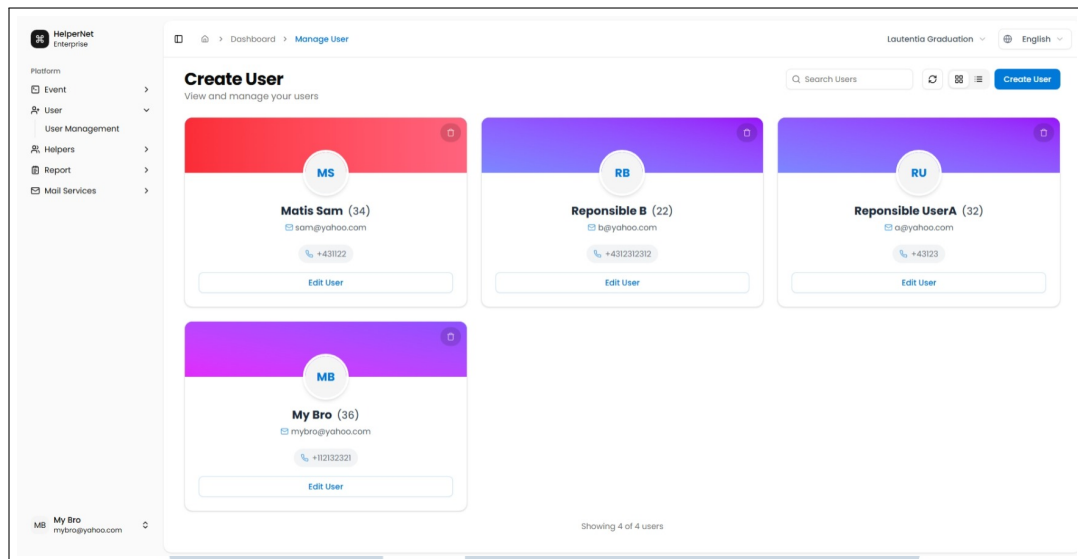


Gambar 3.20. Tampilan tab *Item List Management*

Gambar 3.20 menampilkan halaman *Event Overview* dengan tab *item list* yang ada di *event* tersebut. *Item* yang ditampilkan adalah *item* yang akan digunakan di dalam proyek atau *event* yang sedang dipilih.

3.4.8 Halaman *User Management*

Halaman *User Management* berfungsi untuk mengelola pengguna (*User*) pada *event* yang sedang dipilih, halaman tersebut mencakup fitur *Create*, *Read*, *Update*, *Delete* untuk mempermudah mengelola *event* yang dipilih.

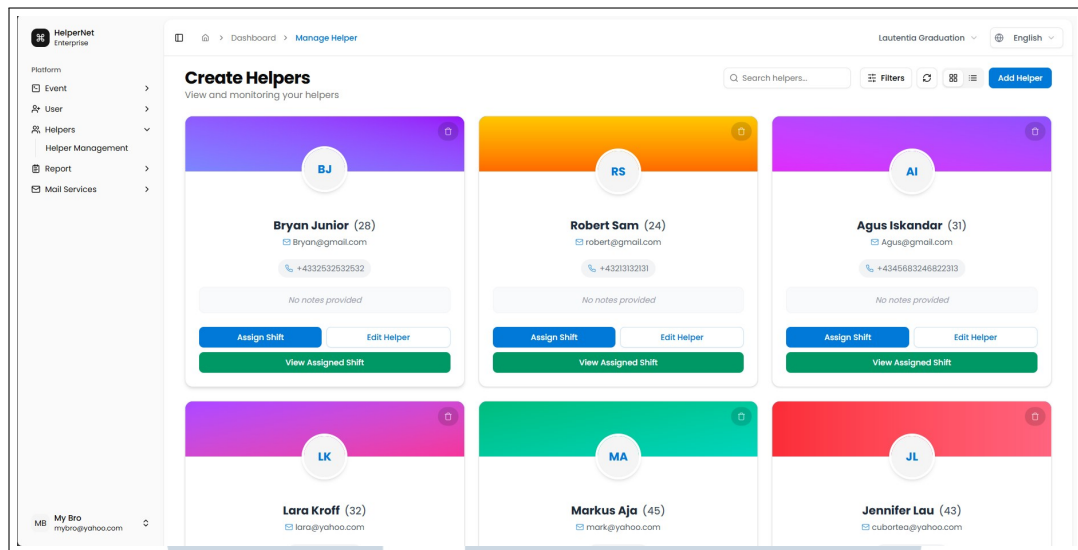


Gambar 3.21. Tampilan halaman *User Management*

Gambar 3.21 menampilkan halaman *User Management*, yang memberikan informasi mendetail terkait *user* yang ada di *event* yang dipilih, seperti nama, umur, email, dan nomor telepon.

3.4.9 Halaman *Helper Management*

Halaman *Helper Management* berfungsi untuk mengelola panitia atau *helper* yang bekerja di *Event* atau proyek yang sedang dipilih. Halaman ini mencakup fitur *Create*, *Read*, *Update*, *Delete*, serta dilengkapi fitur untuk menetapkan atau *assign* helper ke *shift* yang sudah dibuat sebelumnya di halaman *Department Shift*.

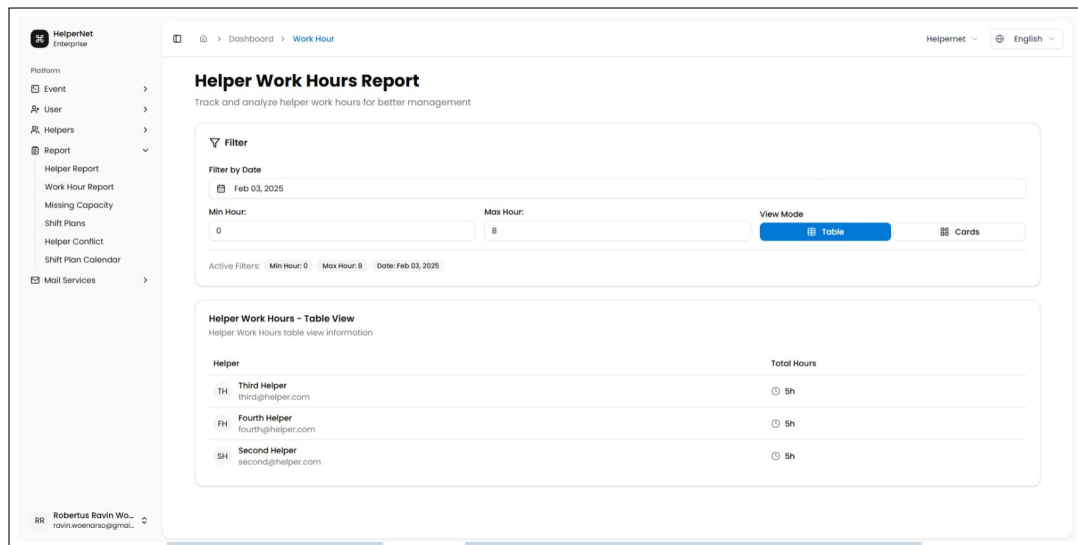


Gambar 3.22. Tampilan halaman *Helper Management*

Gambar 3.22 menampilkan halaman *Helper Management*, yang memberikan informasi mendetail terkait *helper* yang ada di *event* yang dipilih, seperti nama, umur, email, dan nomor telepon, dan *shift* apa saja yang mereka ikuti.

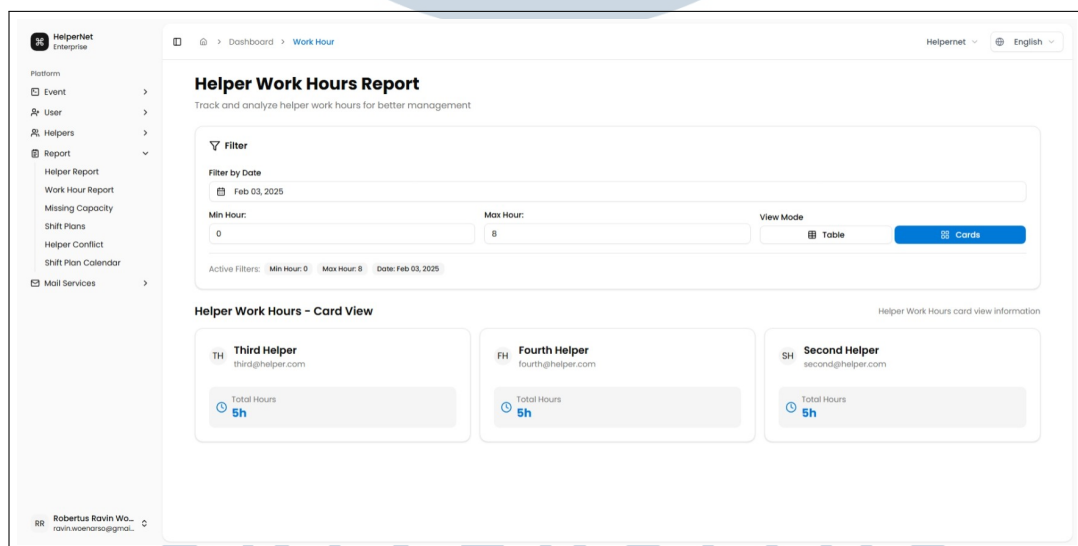
3.4.10 Halaman *Helper Work Hours Report*

Halaman *Helper Work Hours Report* berfungsi untuk menampilkan semua total jam kerja *Helper*, yang sudah di *assign* di *Department Shift* yang sudah dibuat. Halaman ini memiliki fitur untuk *filter* berdasarkan hari, minimal jam kerja, dan maksimal jam kerja.



Gambar 3.23. Tampilan halaman *Helper Work Hours Report - Table View*

Gambar 3.23 menampilkan halaman *Helper Work Hours Report*, yang memberikan informasi atau ringkasan terkait total jam kerja *helper* yang telah di *assign* di shift yang sudah disediakan dalam bentuk tabel.



Gambar 3.24. Tampilan halaman *Helper Work Hours Report - Card View*

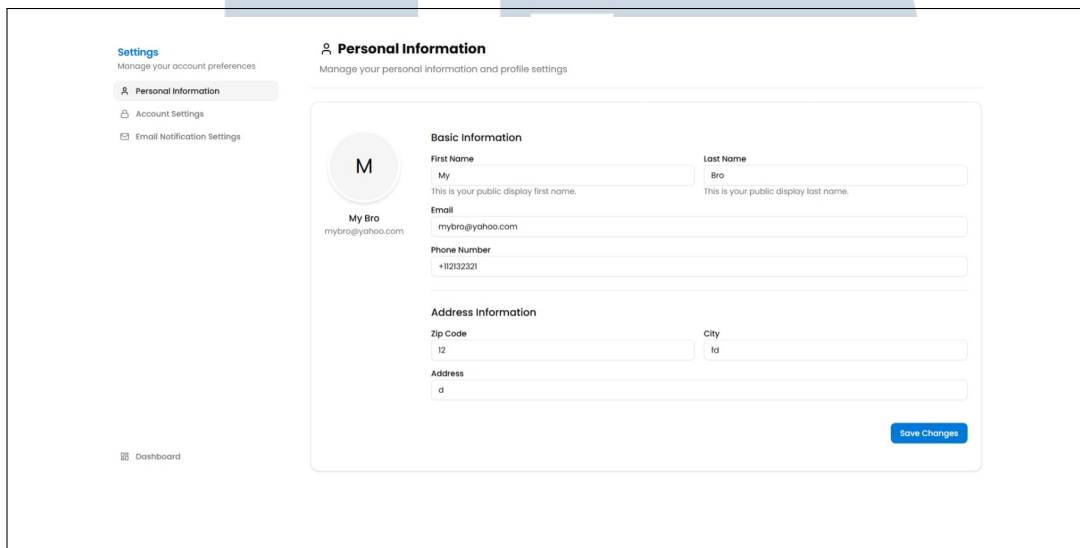
Gambar 3.24 menampilkan halaman *Helper Work Hours Report*, yang memberikan informasi atau ringkasan terkait total jam kerja *helper* yang telah di *assign* di shift yang sudah disediakan dalam bentuk card, sehingga tampilan lebih terbaca ketika digunakan di *handphone*.

3.4.11 Halaman *Settings*

Halaman *settings* berfungsi untuk mengelola preferensi dan data pribadi pengguna dalam halaman web Helpernet. Melalui halaman ini, pengguna dapat memperbarui informasi penting yang berkaitan dengan akun mereka.

Saat ini, halaman setting memiliki tiga fitur utama, yaitu:

1. *Personal Information*:

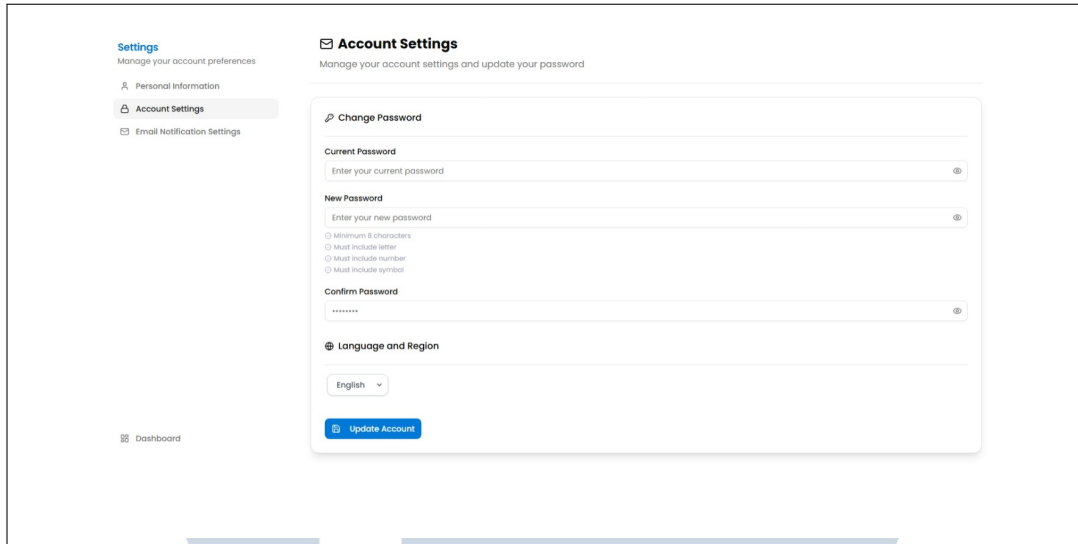


Gambar 3.25. Tampilan halaman *Settings - Personal Information*

Fitur ini memungkinkan pengguna melihat dan mengubah informasi pribadi atau akun yang sudah dibuat, seperti *first name*, *last name*, *email*, *phone number*, dan yang lainnya. Tampilan halaman ini bisa dilihat pada gambar 3.25.

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

2. *Account Settings:*



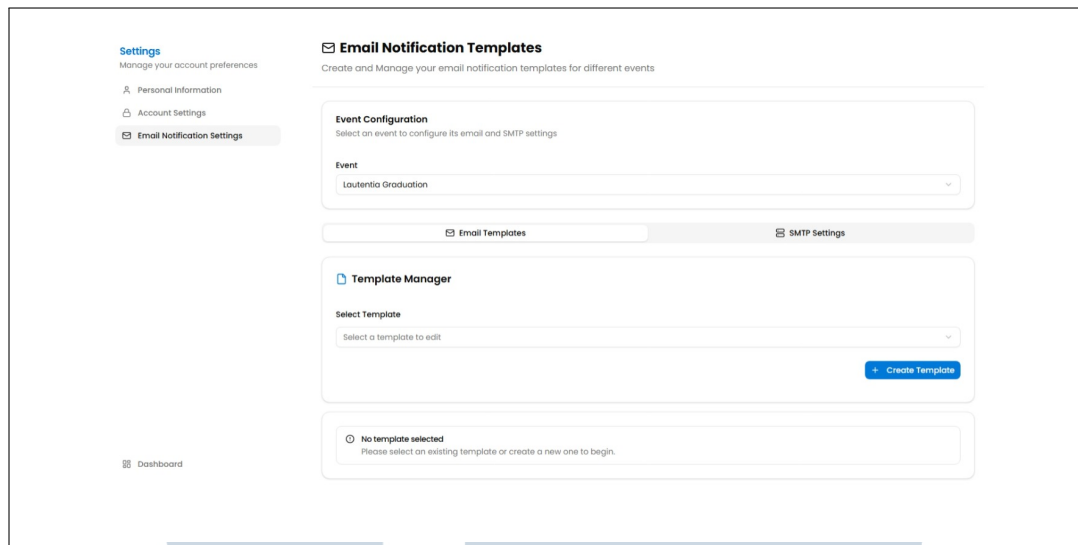
Gambar 3.26. Tampilan halaman *Settings - Account Settings*

Fitur ini disiapkan untuk mengatur hal-hal terkait akun pengguna, untuk sekarang halaman tersebut hanya untuk mengganti *password* akun yang digunakan untuk menggunakan Helpernet, dan pemilihan bahasa. Tampilan halaman ini bisa dilihat pada gambar 3.26.

3. *Email Notification Settings:*

Fitur ini digunakan untuk mengelola sistem notifikasi email ke *helper*. Terdapat dua fitur utama dalam halaman ini, yaitu:

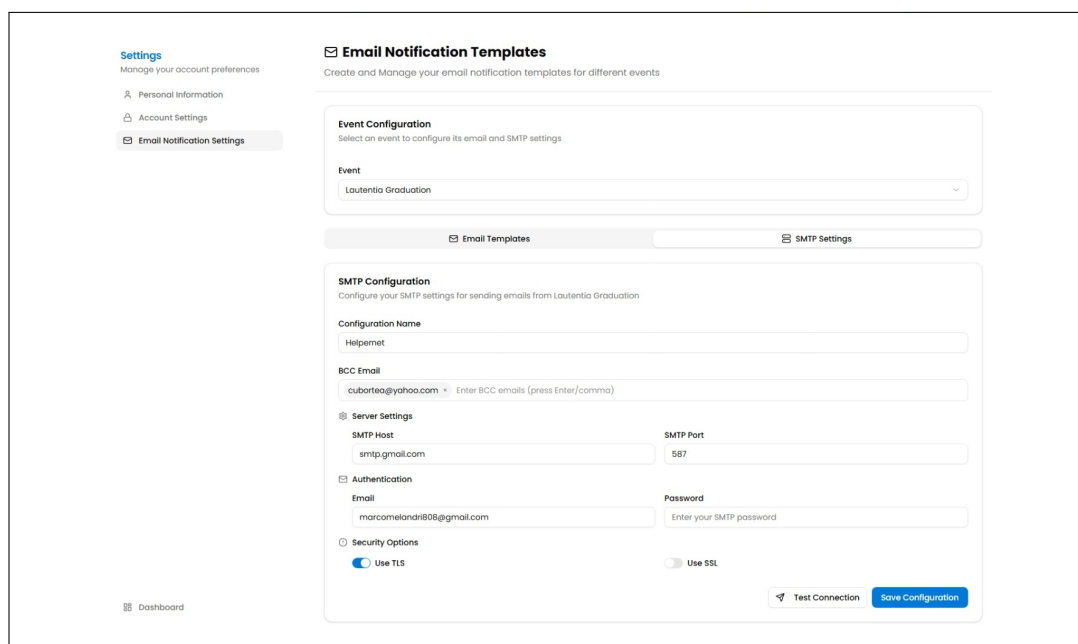
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.27. Tampilan halaman *Settings - Notification Settings - Email Template*

- **Email Template:**

Berfungsi untuk membuat dan mengatur *template* email yang akan digunakan dalam pengiriman pesan kepada panitia atau pihak yang dipilih. Template ini dapat disesuaikan sesuai kebutuhan *event* atau proyek, atau notifikasi tertentu, yang telah ditampilkan pada gambar 3.27.



Gambar 3.28. Tampilan halaman *Settings - Notification Settings - SMTP Configuration*

- **SMTP Settings:**

Digunakan untuk mengatur detail teknis server email *SMTP* yang akan digunakan untuk mengirimkan email dari aplikasi web Helpernet. Pengguna dapat menentukan alamat email pengirim, host server, port, serta kredensial lainnya yang dibutuhkan. Tampilan halaman *SMTP Settings* bisa dilihat di gambar 3.28.

3.4.12 Hasil Pengujian Aplikasi

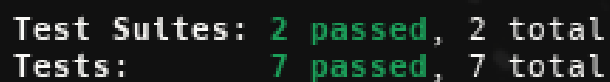
Pengujian aplikasi Helpernet dilakukan secara bertahap dan menyeluruh oleh beberapa pihak yang terlibat dalam pengembangan sistem. Proses pengujian dilakukan dalam beberapa tahap untuk memastikan bahwa fitur-fitur yang dikembangkan berjalan sesuai dengan fungsinya. Pengujian dibagi menjadi dua bagian utama sebagai berikut:

1. **Frontend** menggunakan framework *Jest*
2. **Backend** menggunakan framework testing Go dengan library *Testify*

Pengujian awal dilaksanakan oleh tim pengembang secara internal, dilanjutkan dengan pengujian fungsional oleh salah satu anggota tim, Olivia Syahma, dan diakhiri dengan tahap validasi akhir yang dilakukan oleh direktur.

A Frontend

Pengujian frontend dilakukan menggunakan framework *Jest* untuk memastikan bahwa komponen antarmuka pengguna dan logika interaktif berfungsi dengan benar.



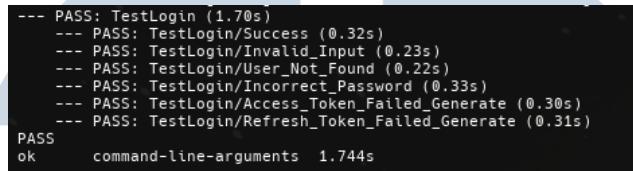
```
Test Suites: 2 passed, 2 total
Tests:       7 passed, 7 total
```

Gambar 3.29. Hasil Pengujian Unit Test Frontend dengan Jest

Gambar 3.29 menunjukkan bahwa *test suites* dan *unit tests* berhasil dijalankan dengan status **passed**. Sebanyak 2 *test suites* dan 7 *unit tests* telah dijalankan, menandakan bahwa fitur yang diuji dari sisi frontend telah berfungsi sesuai dengan spesifikasi yang ditentukan.

B Backend

Pengujian backend berfokus pada controller, menggunakan *Go test* dan library *Testify*.



```
--- PASS: TestLogin (1.70s)
--- PASS: TestLogin/Success (0.32s)
--- PASS: TestLogin/Invalid_Input (0.23s)
--- PASS: TestLogin/User_Not_Found (0.22s)
--- PASS: TestLogin/Incorrect_Password (0.33s)
--- PASS: TestLogin/Access_Token_Failed_Generate (0.30s)
--- PASS: TestLogin/Refresh_Token_Failed_Generate (0.31s)
PASS
ok      command-line-arguments  1.744s
```

Gambar 3.30. Hasil Pengujian Unit Test Backend untuk Fitur Login

Gambar 3.30 menunjukkan bahwa seluruh skenario pengujian untuk fitur login berhasil dijalankan dengan hasil **PASS**. Skenario yang diuji meliputi:

- Proses login berhasil
- Input tidak valid (gagal validasi)
- Pengguna tidak ditemukan
- Kata sandi salah
- Gagal menghasilkan access token
- Gagal menghasilkan refresh token

3.5 Kendala dan Solusi yang Ditemukan

3.5.1 Kendala yang ditemukan

Selama pelaksanaan kerja magang pada proyek Helpernet, terdapat sejumlah kendala dan tantangan yang dihadapi, antara lain sebagai berikut:

1. Miskomunikasi antar tim:

Selama proses pengembangan aplikasi, terkadang terjadi miskomunikasi antara anggota tim, terutama dalam penyampaian kebutuhan atau pembagian tugas.

2. Adanya *bug* pada aplikasi:

Dalam proses implementasi dan pengembangan fitur, beberapa *bug* ditemukan, seperti kesalahan logika, tampilan yang tidak sesuai ekspektasi, atau kesalahan saat integrasi antar modul.

3. Perubahan tugas yang tidak konsisten:

Selama pelaksanaan magang, terdapat beberapa kasus di mana instruksi pekerjaan mengalami perubahan setelah tugas selesai dikerjakan. Sebagai contoh, pada awalnya diberikan arahan untuk membuat fitur tertentu, namun setelah pengerjaan selesai, arahan tersebut berubah menjadi permintaan untuk mengimplementasikan fungsi yang berbeda. Perubahan ini tidak disampaikan secara jelas sejak awal, sehingga mengakibatkan ketidakefisienan waktu serta perlunya melakukan ulang pekerjaan yang telah diselesaikan.

3.5.2 Solusi yang diterapkan

Untuk mengatasi kendala dan tantangan yang dihadapi selama pelaksanaan kerja magang, berikut ini adalah solusi yang diterapkan:

1. Membuat *ticketing* untuk pekerjaan yang dilakukan:

Untuk meminimalkan miskomunikasi, setiap permintaan atau tugas dicatat dalam bentuk *ticket* menggunakan *tools* manajemen proyek di Gitea. Dengan sistem ini, setiap tugas memiliki deskripsi yang jelas, prioritas, serta riwayat komunikasi, sehingga mengurangi adanya miskomunikasi antar tim.

2. Melakukan *testing* terhadap fitur yang sudah dibuat:

Setiap fitur yang sudah dibuat, fitur tersebut harus diuji dengan skenario yang berbeda sehingga menghindari adanya bug saat digunakan oleh pengguna.

3. Menunggu konfirmasi sebelum menyelesaikan tugas sampai selesai:

Untuk menghindari revisi akibat perubahan instruksi, pekerjaan tidak langsung diselesaikan sepenuhnya. Sebelum melanjutkan ke tahap akhir, hasil awal dikonsultasikan terlebih dahulu kepada direktur dan project manager agar arah pengerjaan benar dan sesuai harapan.