

BAB III PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dalam Kerja Magang

Selama pelaksanaan kerja magang, mahasiswa berada di bawah naungan Program Studi Informatika. Struktur koordinasi melibatkan dua pihak utama: pembimbing akademik dari kampus dan supervisor teknis dari tempat magang. Pembimbing akademik berperan memberikan arahan umum terkait tujuan dan luaran magang, serta memastikan keselarasan kegiatan magang dengan kurikulum program studi. Sementara itu, supervisor teknis memiliki tanggung jawab membimbing mahasiswa dalam aspek teknis pekerjaan sehari-hari, memberikan masukan konstruktif, dan mengevaluasi kinerja mahasiswa selama periode magang. Koordinasi yang baik antara kedua pihak ini diharapkan dapat menciptakan lingkungan belajar yang optimal bagi mahasiswa.

Tabel 3.1 Uraian Kerja

Minggu ke	Tanggal	Keterangan
1	25 - 29 Agustus 2025	- Pemetaan indikator UIGM - Sketsa dashboard
2	1 - 5 September 2025	- Setup project - Struktur database awal
3	8 - 12 September 2025	- Implementasi dashboard
4	15 - 19 September 2025	- Fitur evidence
5	22 - 26 September 2025	- Integrasi data
6	29 September - 3 Oktober 2025	- Refinement design
7	6 - 10 Oktober 2025	- Desain form input
8	13 - 17 Oktober 2025	- Implementasi form input

9	20 - 24 Oktober 2025	- Implementasi validasi data
10	27 - 31 Oktober 2025	- Role access
11	3 - 7 November 2025	- Integrasi input
12	10 - 14 November 2025	- Uji coba beberapa fitur/menu
13	17 - 21 November 2025	- Perbaikan berdasarkan UI
14	24 - 28 November 2025	- Implementasi sederhana
15	1 - 5 Desember 2025	- Uji coba seluruh hasil
16	8 - 12 Desember 2025	- Finalisasi input data

3.2 Tugas dan Uraian dalam Kerja Magang

Dalam periode magang ini, mahasiswa diberikan beberapa tugas yang terkait dengan perencanaan, pengembangan, dan implementasi *website* untuk mendukung pengelolaan data *GreenMetric*. Rincian tugas tersebut adalah sebagai berikut:

3.2.1 Perencanaan Implementasi *Dashboard*

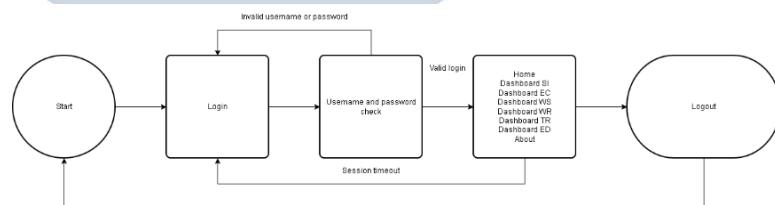
Tahap awal melibatkan perencanaan yang matang mengenai arsitektur dan teknologi yang akan digunakan dalam pengembangan *dashboard website*. Pemilihan *ReactJS* sebagai kerangka kerja (*framework*) utama didasarkan pada kemampuannya dalam membangun antarmuka pengguna (*UI*) yang interaktif dan responsif. Selain itu, *ReactJS* juga menawarkan komponen-komponen siap pakai yang dapat mempercepat proses pengembangan.

Untuk desain *UI/UX*, digunakan perangkat lunak *Figma*. *Figma* dipilih karena kemudahan kolaborasi tim, fitur *prototyping* yang canggih, dan kemampuan menghasilkan

desain yang sesuai dengan prinsip-prinsip desain yang berpusat pada pengguna (*user-centered design*). Desain yang dibuat di *Figma* akan menjadi acuan visual bagi pengembang dalam mengimplementasikan antarmuka *website*.

Pada sisi basis data (*database*), *MySQL* dipilih sebagai sistem manajemen basis data (DBMS) karena popularitasnya, stabilitasnya, dan ketersediaan sumber daya yang melimpah. Struktur basis data dirancang untuk menyimpan data *GreenMetric* secara terstruktur dan efisien, serta memungkinkan pengambilan data yang cepat dan akurat untuk ditampilkan di *dashboard*.

3.2.2 Flowchart Sistem



Gambar 3.2.2.1 Flowchart Sistem Login UI GreenMetric

Gambar di atas memperlihatkan diagram alir (*flowchart*) yang menggambarkan proses *login* pada antarmuka (*UI*) *GreenMetric*. *Flowchart* ini memberikan gambaran visual mengenai langkah-langkah yang terlibat dalam proses autentikasi pengguna, mulai dari *input* kredensial hingga pemberian akses ke sistem.

3.2.3 Identifikasi Permasalahan

Sebelum pengembangan *website*, pengelolaan data *GreenMetric* dilakukan secara manual. Proses ini memiliki beberapa kelemahan mendasar, di antaranya:

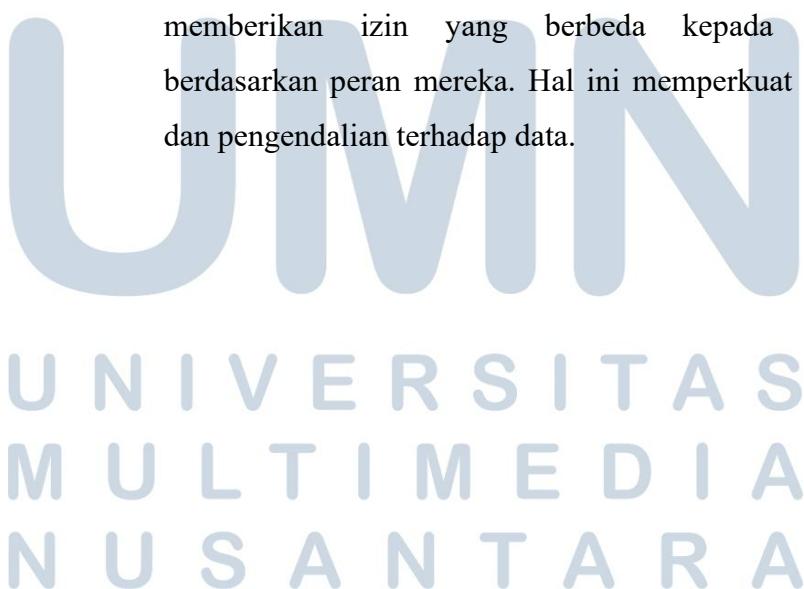
- Kesulitan dalam *Monitoring*: Data yang tersebar di berbagai dokumen dan *spreadsheet* membuat pemantauan (*monitoring*) terhadap kinerja *GreenMetric* menjadi sulit dan memakan waktu.
- Evaluasi yang Tidak Efisien: Proses evaluasi kinerja *GreenMetric* menjadi rumit karena data harus dikumpulkan dan dianalisis secara manual. Hal ini rentan terhadap kesalahan dan membutuhkan sumber daya yang besar.
- Kurangnya Transparansi: Data *GreenMetric* yang tidak terpusat membuat transparansi informasi menjadi terbatas. Pihak-pihak yang berkepentingan sulit mengakses data secara cepat dan akurat.
- Integrasi *Evidence* yang Terhambat: Pengumpulan dan pengelolaan bukti (*evidence*) pendukung kinerja *GreenMetric* menjadi sulit diintegrasikan ke dalam proses evaluasi.

Dengan demikian, pengembangan *website* menjadi solusi yang diperlukan untuk meningkatkan efisiensi, transparansi, dan integrasi *evidence* dalam pengelolaan data *GreenMetric*.

3.2.4 Solusi yang Diimplementasikan

Untuk mengatasi permasalahan yang telah diidentifikasi, *website GreenMetric* dikembangkan dengan fitur-fitur sebagai berikut:

- *Dashboard*: Menyajikan informasi kinerja *GreenMetric* secara visual dan interaktif. Pengguna dapat melihat tren, perbandingan, dan analisis data dengan mudah.
- *Form Input*: Memungkinkan pengguna memasukkan data *GreenMetric* secara terstruktur. Validasi data dilakukan pada saat *input* untuk memastikan kualitas data yang akurat.
- *Upload Evidence*: Pengguna dapat mengunggah dokumen atau berkas pendukung (*evidence*) yang relevan dengan data *GreenMetric* yang dimasukkan.
- Validasi Data: Sistem melakukan validasi terhadap data yang dimasukkan untuk memastikan keabsahan dan konsistensi data.
- *Role Access*: Sistem memiliki fitur pengaturan hak akses (*role access*) yang memungkinkan administrator memberikan izin yang berbeda kepada pengguna berdasarkan peran mereka. Hal ini memperkuat keamanan dan pengendalian terhadap data.



3.2.5 Tampilan Projek



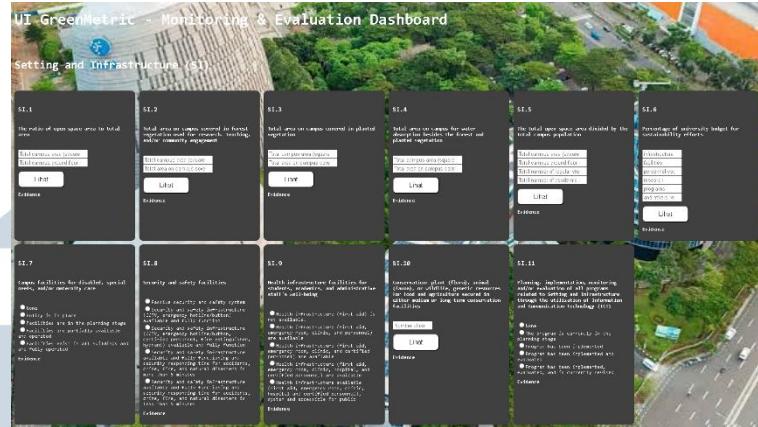
Gambar 3.2.5.1 Tampilan Halaman *Login Website UI GreenMetric*

Gambar 3.2.5.1 di atas memperlihatkan tampilan halaman *login website UI GreenMetric*. Halaman ini didesain sederhana agar pengguna dapat login dengan mudah.

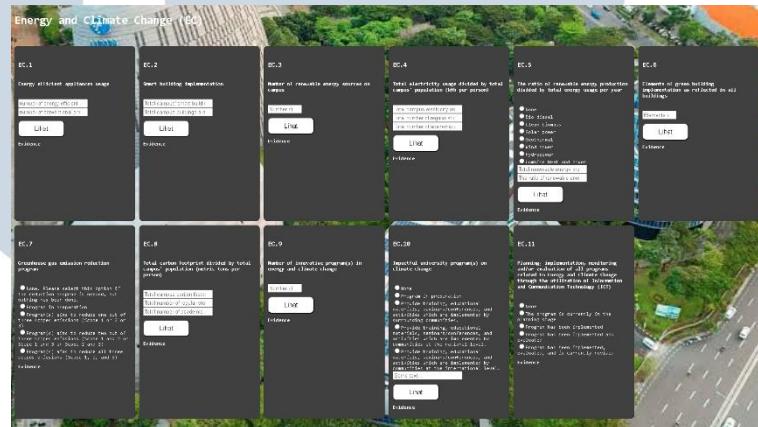


Gambar 3.2.5.2 Tampilan *Homepage Website UI GreenMetric Setelah Login*

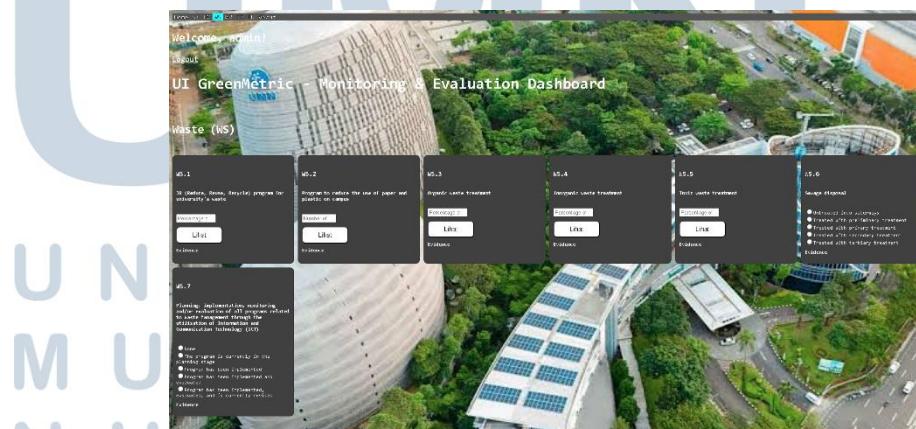
Gambar 3.2.5.2 di atas memperlihatkan tampilan *homepage website UI GreenMetric*. *Homepage* ini menyajikan pilihan *dashboard GreenMetric*. Pengguna dapat memilih salah satu *dashboard* di daftar atau menu di atas.



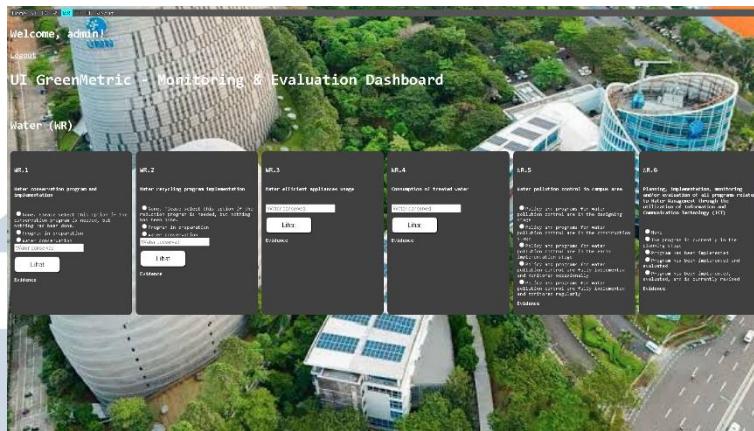
Gambar 3.2.5.3 Tampilan *Dashboard Website UI GreenMetric* Bagian SI



Gambar 3.2.5.4 Tampilan *Dashboard Website UI GreenMetric* Bagian EC



Gambar 3.2.5.5 Tampilan *Dashboard Website UI GreenMetric* Bagian WS



Gambar 3.2.5.6 Tampilan Dashboard Website UI GreenMetric Bagian WR



Gambar 3.2.5.7 Tampilan Dashboard Website UI GreenMetric Bagian ED

Gambar 3.2.5.3 s/d 3.2.5.7 di atas memperlihatkan tampilan *dashboard website UI GreenMetric*. *Dashboard* ini menyajikan informasi kinerja *GreenMetric* secara visual melalui grafik dan tabel. Pengguna dapat memantau perkembangan kinerja dari waktu ke waktu dan mengidentifikasi area yang memerlukan perbaikan.

3.3 Kendala dan Solusi

Selama pelaksanaan magang dan pengembangan situs web *UI GreenMetric*, Mahasiswa menghadapi beberapa tantangan yang memerlukan solusi yang tepat. Berikut

adalah rincian tantangan tersebut beserta cara penyelesaiannya:

Kesulitan Teknis pada *Framework ReactJS*:

Pada awalnya, mahasiswa mengalami kesulitan dalam memahami konsep pengelolaan *state* dan *component lifecycle* di *ReactJS*. Ini merupakan hal yang umum bagi pengembang yang baru mengenal *framework* ini.

Penyelesaian: Untuk mengatasi masalah ini, Mahasiswa melakukan langkah-langkah berikut:

- Studi Literatur Mendalam: Mahasiswa melakukan telah literatur tambahan dengan membaca berbagai artikel, tutorial, dan sumber daring lainnya yang membahas konsep-konsep *ReactJS* secara mendalam.
- Dokumentasi Resmi *ReactJS*: Mahasiswa mempelajari dan mengikuti dokumentasi resmi *ReactJS* secara seksama. Dokumentasi ini memberikan panduan yang jelas dan komprehensif tentang penggunaan *framework* ini.
- Diskusi dengan Dosen dan Tim: Mahasiswa berdiskusi secara aktif dengan dosen pembimbing dan rekan tim yang lebih berpengalaman. Diskusi ini membantu Mahasiswa untuk memahami konsep-konsep yang sulit dan menemukan solusi untuk masalah yang Mahasiswa hadapi.

Integrasi Database MySQL dengan *Front-end*:

Pada tahap awal pengembangan, Mahasiswa mengalami masalah dalam menghubungkan *front-end* (*ReactJS*) dengan *database MySQL* melalui *Application Programming Interface (API)*. Masalah ini menyebabkan *error* dalam proses pengambilan dan penyimpanan data.

Penyelesaian: Untuk mengatasi masalah ini, Mahasiswa melakukan langkah-langkah berikut:

- *Debugging Sistematis*: Mahasiswa melakukan proses *debugging* secara sistematis untuk mengidentifikasi penyebab *error*. Mahasiswa memeriksa kode *front-end*, API, dan konfigurasi *database* untuk menemukan kesalahan.
- Perbaikan Konfigurasi API: Mahasiswa memperbaiki konfigurasi API untuk memastikan bahwa API dapat berkomunikasi dengan *database MySQL* dengan benar.
- Pengujian *Endpoint* dengan *Postman*: Mahasiswa menggunakan *Postman* untuk menguji *endpoint* API sebelum diintegrasikan ke *front-end*. Pengujian ini membantu Mahasiswa untuk memastikan bahwa API berfungsi dengan baik dan mengembalikan data yang benar.

Desain Antarmuka yang Kurang Konsisten:

Desain antarmuka pengguna (UI) awal yang Mahasiswa buat di *Figma* belum sepenuhnya konsisten dengan prinsip-prinsip desain yang berpusat pada pengguna (*user-centered*

design). Hal ini dapat menimbulkan kebingungan dan kesulitan bagi pengguna.

Penyelesaian: Untuk mengatasi masalah ini, Mahasiswa melakukan langkah-langkah berikut:

- *Refinement* Desain: Mahasiswa melakukan *refinement* desain dengan memperhatikan masukan dari dosen pembimbing dan pengguna yang berpartisipasi dalam uji coba.
- Prinsip Desain Berpusat pada Pengguna: Mahasiswa memastikan bahwa desain UI Mahasiswa memenuhi prinsip-prinsip desain yang berpusat pada pengguna, seperti kemudahan penggunaan, aksesibilitas, dan estetika.
- Tampilan yang Intuitif: Mahasiswa berusaha untuk membuat tampilan UI yang intuitif dan mudah digunakan oleh pengguna. Mahasiswa menggunakan ikon, warna, dan tata letak yang jelas dan konsisten.

Validasi Data yang Kompleks:

Proses validasi data *GreenMetric* membutuhkan aturan yang cukup rumit untuk memastikan bahwa data yang dimasukkan akurat dan sesuai dengan standar yang ditetapkan. Tanpa validasi yang memadai, dapat terjadi kesalahan *input* yang dapat mempengaruhi hasil perhitungan *GreenMetric*.

Penyelesaian: Untuk mengatasi masalah ini, Mahasiswa melakukan langkah-langkah berikut:

- Penambahan Aturan Validasi Kustom: Mahasiswa menambahkan aturan validasi kustom pada formulir *input* untuk memastikan bahwa data yang dimasukkan memenuhi semua persyaratan yang diperlukan.
- Uji Coba Berulang: Mahasiswa melakukan uji coba berulang untuk memastikan bahwa aturan validasi berfungsi dengan benar dan mencegah kesalahan *input*.
- Standar Data: Mahasiswa memastikan bahwa data yang masuk sesuai dengan standar *GreenMetric* yang berlaku.

Manajemen Waktu dan *Deadline*:

Jadwal magang yang padat dan banyaknya fitur yang harus dikembangkan menyebabkan beberapa fitur tertunda penyelesaiannya. Manajemen waktu yang efektif sangat penting untuk memastikan bahwa proyek selesai tepat waktu.

Penyelesaian: Untuk mengatasi masalah ini, Mahasiswa melakukan langkah-langkah berikut:

- Pembuatan *Timeline* Kerja yang Detail: Mahasiswa membuat *timeline* kerja yang lebih detail untuk membagi tugas dan menetapkan *deadline* yang realistik.
- Prioritasi Fitur Inti: Mahasiswa memprioritaskan fitur inti (seperti *dashboard*, *input* data, dan bukti pendukung) untuk memastikan bahwa fitur-fitur penting ini selesai terlebih dahulu.

- Penyelesaian Fitur Tambahan: Mahasiswa menyelesaikan fitur tambahan setelah fungsi utama berjalan stabil.

Keterbatasan Dokumentasi Internal:

Pada awal proyek, belum tersedia dokumentasi teknis yang lengkap mengenai kebutuhan sistem *GreenMetric*. Kurangnya dokumentasi dapat menyulitkan pengembang dalam memahami sistem dan mengembangkan fitur baru.

Penyelesaian: Untuk mengatasi masalah ini, Mahasiswa melakukan langkah-langkah berikut:

- Penyusunan Dokumentasi Mandiri: Mahasiswa menyusun dokumentasi sendiri yang mencakup *flowchart*, *Entity Relationship Diagram (ERD)*, dan catatan teknis lainnya.
- Pengembangan Terstruktur: Dokumentasi ini membantu Mahasiswa untuk mengembangkan sistem secara lebih terstruktur dan efisien.
- Kemudahan Pemahaman: Dokumentasi ini juga memudahkan pihak lain (seperti pengembang lain atau pengguna sistem) untuk memahami sistem *GreenMetric*.