

BAB III

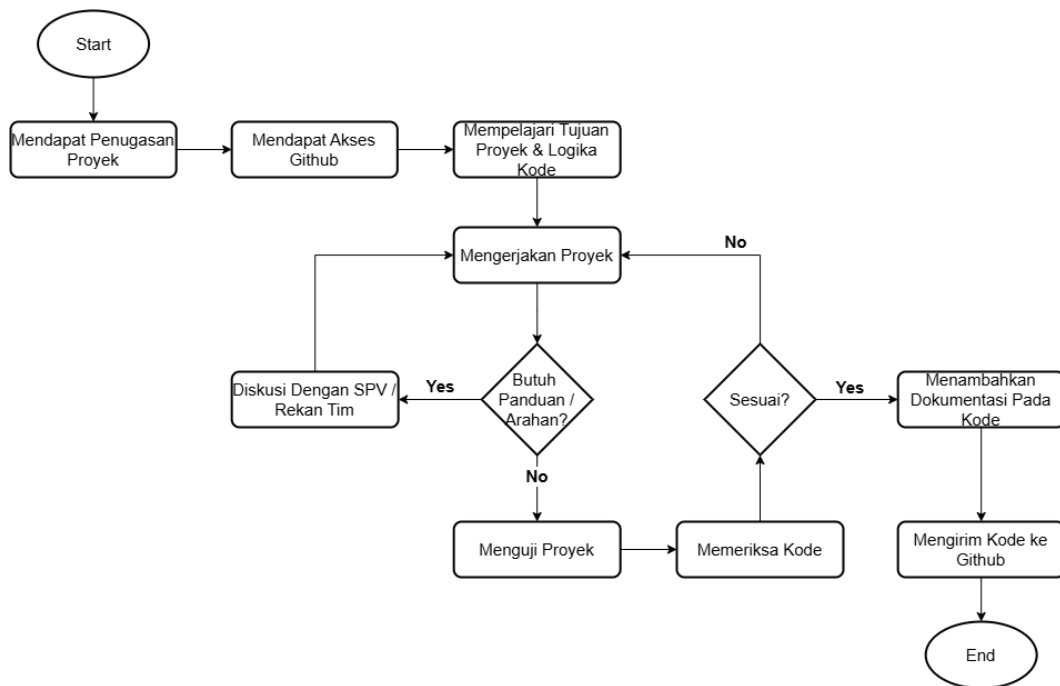
PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Selama kegiatan magang di PT IAS Support Indonesia sebagai *Fullstack Developer*, kewajiban utama peserta magang adalah mengembangkan serta memperbarui fitur pada proyek *website* yang dikerjakan. Tanggung jawab tersebut antara lain mempelajari kebutuhan proyek dan logika kode yang sudah ada, mengimplementasikan solusi untuk sisi *front-end* maupun *back-end*, melakukan pengujian awal terhadap hasil kerja, serta memastikan kesesuaian fitur dengan kebutuhan pengguna. Selain itu, dokumentasi berupa komentar pada bagian-bagian penting kode wajib diberikan oleh peserta agar tim atau pengembang selanjutnya mudah memahami fungsi atau alur program. Setelah proses pengujian dan pengecekan selesai, hasil kerja dikirim melalui *repository* GitHub sesuai prosedur tim yang berlaku.

Untuk menyelesaikan tanggung jawab tersebut, koordinasi dilakukan secara aktif antara peserta magang dengan supervisor dan rekan satu tim, terutama ketika menemui masalah teknis ataupun ketidakjelasan dalam hasil yang diinginkan. Diskusi dilakukan untuk mendapatkan arahan dan memastikan solusi yang dikembangkan sesuai dengan standar tim. Kolaborasi ini juga berjalan melalui *platform* GitHub, yang menjadi tempat pengumpulan pekerjaan dan dokumentasi teknis. Dengan koordinasi tersebut, proses pengembangan dapat berjalan efektif dan menghasilkan output yang sesuai dengan kebutuhan pengguna.

Manfaat dari hasil pekerjaan peserta magang dirasakan oleh dua pihak, yaitu pihak eksternal dan internal. Pihak eksternal yang dimaksud adalah pelanggan perusahaan, yaitu perusahaan yang membutuhkan dan menggunakan layanan digital dari PT IASS, yang dimana layanannya adalah pengembangan sistem. Disisi lain, pihak internal yang juga ikut merasakan manfaatnya adalah tim pengembang yang mendapatkan dukungan tambahan dalam menyelesaikan proyek, serta divisi terkait yang sistemnya dikembangkan atau diperbarui.



Gambar 3.1 Koordinasi dan Alur Kerja Magang

Supervisi langsung peserta magang dalam proses magang dilakukan oleh Bapak Harival Tivani, yang menjabat sebagai Technology Operations & Maintenance Senior Officer di PT IAS Support Indonesia. Dalam perannya di perusahaan, supervisor bertanggung jawab untuk mengoordinasikan tim, memetakan kebutuhan *manpower* terhadap proyek atau tugas, menjalin komunikasi lintas unit, mengambil keputusan di *layer* pertama, serta memastikan setiap tugas harian dapat diselesaikan dengan baik. Sebagai pembimbing selama program magang, supervisor memberikan arahan, solusi teknis, serta evaluasi berkala terhadap tugas-tugas yang diberikan kepada peserta magang. supervisor juga menjadi penghubung antara peserta magang dengan tim pengembang atau divisi lain yang terlibat dalam proyek, serta memastikan bahwa pekerjaan yang dilakukan selaras dengan standar dan kebutuhan perusahaan.

Sekitar empat proyek pengembangan, meliputi tampilan *dashboard* dan fitur-fitur pada *website* sistem perusahaan, telah diselesaikan oleh peserta magang sepanjang program magang. Setiap proyek memberikan tantangan tersendiri yang memerlukan ketelitian dalam penulisan kode serta pemahaman mendalam terhadap kebutuhan pengguna. Hasil dari pengerjaan tugas antara lain perbaikan fitur yang

sudah ada, pengembangan modul baru, serta peningkatan tampilan antarmuka agar lebih intuitif dan responsif.

Melalui pengalaman tersebut, banyak wawasan berharga yang diperoleh terkait praktik kerja profesional. Wawasan ini mencakup pentingnya komunikasi di dalam tim maupun antar divisi, pemahaman terhadap permasalahan dari akar penyebabnya sebelum solusi ditentukan, serta penyusunan strategi logika kode yang efisien dan sesuai kebutuhan pengguna. Selain itu, pentingnya dokumentasi yang terstruktur serta penulisan kode yang bersih dan mudah dipahami untuk pengembangan di masa depan juga dipelajari.

3.2 Tugas dan Uraian Kerja Magang

Sesuai dengan tanggung jawab yang diberikan kepada peserta magang, tugas-tugas yang telah dilaksanakan terbagi menjadi lima bagian utama. Proyek pertama adalah pengembangan sistem e-Office milik Taman Mini Indonesia Indah (TMII). Dalam proyek ini, sejumlah perbaikan dan penyesuaian telah dilakukan, antara lain pada tampilan surat, tipe dokumen yang dikirim, format penomoran surat, penyesuaian tanggal surat, serta penambahan fitur notifikasi untuk pengguna. Setelah itu, pembelajaran teknologi-teknologi yang akan digunakan di proyek kedepannya dan yang mungkin akan digunakan dimasa depan juga telah dilakukan oleh peserta magang.

Proyek kedua yang dikerjakan adalah Baggage Reconciliation System (BRS) milik PT Gapura. Pada proyek ini, pengembangan *dashboard* publik yang dapat diakses tanpa autentikasi login telah dilakukan, dengan tujuan memberikan akses informasi secara terbuka. Selanjutnya, proyek ketiga, yaitu pengembangan sistem internal untuk divisi SBU ICT berupa Tenant Management juga ditugaskan kepada peserta magang. Proyek ini diperlukan perubahan pada alur bisnis (*business process*) dan struktur basis data, sehingga diperlukan penambahan beberapa *field* baru dan penyesuaian pada tampilan *dashboard*.

Proyek keempat adalah *Toilet Management System* (TMS), di mana pembangun *dashboard* visual berupa grafik untuk menampilkan data pemantauan secara

informatif menjadi tugas peserta magang. Setelah menyelesaikan keempat proyek tersebut, pengembangan pada sistem e-Office TMII kembali ditugaskan dan dilanjutkan, dengan fokus pada penambahan fitur pencarian, penyesuaian konten surat, serta pengelolaan status surat.

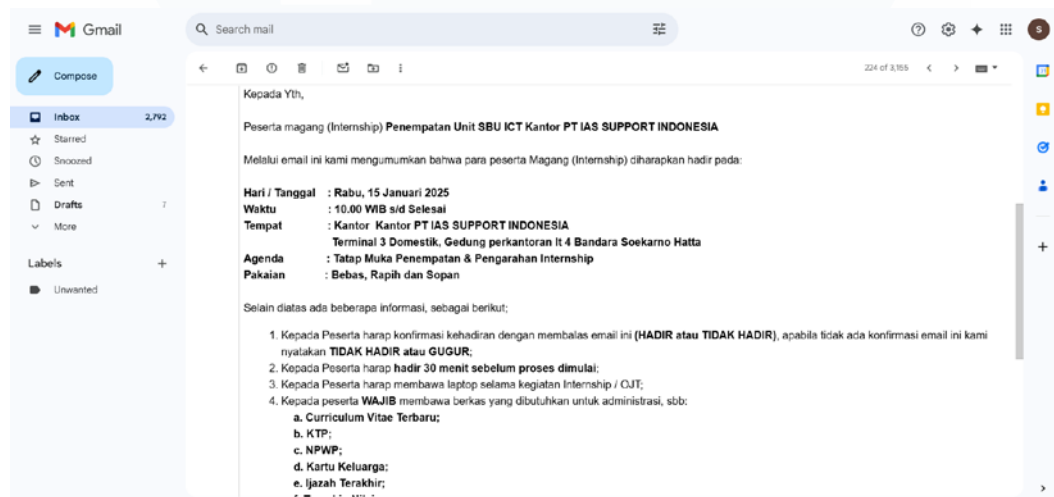
Tabel 3.1 Uraian Tugas Selama Pelaksanaan Magang

No.	Aktivitas	Minggu Ke-	Waktu Mulai	Waktu Selesai
1	Pengenalan Lingkungan	1	15 Januari 2025	15 Januari 2025
Proyek 1: E-Office				
2	Mempelajari Dokumentasi & Penugasan	1	15 Januari 2025	17 Januari 2025
3	Perbaikan format pdf surat	2	20 Januari 2025	21 Januari 2025
4	Penyesuaian Tipe Dokumen	2-3	22 Januari 2025	30 Januari 2025
5	Revisi format pdf	3	30 Januari 2025	31 Januari 2025
6	Penyesuaian Nomor Surat	4	3 Februari 2025	4 Februari 2025
7	Menambahkan fitur notifikasi	4	5 Februari 2025	7 Februari 2025
8	Memperbaiki <i>field</i> tujuan surat	5	10 Februari 2025	10 Februari 2025
9	<i>Testing & Debugging</i>	5	11 Februari 2025	14 Februari 2025
10	Briefing penugasan Tenant Management	5	14 Februari 2025	14 Februari 2025
11	Membuat dokumentasi & <i>code review</i>	6	17 Februari 2025	21 Februari 2025
Pembelajaran				
12	Mempelajari Tailwind	7	24 Februari 2025	28 Februari 2025
13	Mempelajari Laravel	8-11	3 Maret 2025	24 Maret 2025

No.	Aktivitas	Minggu Ke-	Waktu Mulai	Waktu Selesai
14	Mempelajari Graph Database	11-14	25 Maret 2025	15 April 2025
Proyek 2: BRS				
15	Testing Aplikasi di lapangan	14	16 April 2025	17 April 2025
16	Membuat <i>dashboard</i>	15	21 April 2025	23 April 2025
Proyek 3: Tenant Management				
17	Menambahkan <i>master table</i> business line dan <i>product category</i>	15	24 April 2025	25 April 2025
18	Menyesuaikan <i>dashboard</i>	16	28 April 2025	29 April 2025
19	Presentasi progress & penambahan kebutuhan	16	30 April 2025	30 April 2025
20	Menambahkan <i>form</i> tenant dan menyesuaikan <i>form</i> company	16-17	2 Mei 2025	8 Mei 2025
21	Penyesuaian grafik pada <i>dashboard</i>	17	9 Mei 2025	9 Mei 2025
22	Menyesuaikan form survey	18	14 Mei 2025	15 Mei 2025
23	Membuat format surat quotation & surat berlangganan	18-19	16 Mei 2025	19 Mei 2025
24	Penyesuaian <i>form</i> tenant	19	20 Mei 2025	20 Mei 2025
25	Menyesuaikan <i>form</i> register	19	21 Mei 2025	23 Mei 2023
Proyek 4: Toilet Management System				
26	Membuat grafik <i>dashboard</i>	20	26 Mei 2025	27 Mei 2025
Melanjut Proyek 1: E-Office TMII				
27	Mengurutkan Tembusan pada Surat	20	28 Mei 2025	28 Mei 2025
28	Menambahkan Status Disposisi	20-21	28 Mei 2025	2 Juni 2025
29	Menambahkan Tujuan Akhir Surat	21	2 Juni 2025	2 Juni 2025
30	Menambahkan Fitur Approval	21-22	3 Juni 2025	10 Juni 2025
31	<i>Testing, Debugging & Code Review</i>	22	11 Juni 2025	13 Juni 2025

3.2.1 Pengenalan Lingkungan

Di hari pertama kegiatan magang, 15 Januari 2025, peserta telah diundang secara resmi melalui email oleh pihak *Human Resource* (HR) untuk hadir di kantor PT IAS Support Indonesia. Setibanya di lokasi, peserta magang disambut oleh PIC HR yang menangani program *internship*. Pada sesi awal tersebut, penjelasan umum mengenai perusahaan diterima oleh peserta magang, termasuk informasi keadaan terbaru PT IAS Support Indonesia, susunan kepemimpinan yang baru, aturan berpakaian selama magang, serta ketentuan administratif yang harus dipenuhi seperti absensi dan perizinan. Penjelasan diakhiri dengan penandatanganan dokumen penting, yaitu surat pernyataan pemagangan, yang berfungsi sebagai kontrak kerja sementara selama periode magang berlangsung.



Gambar 3.2 Screenshot Email dari HR

Setelah sesi pengenalan awal, peserta diajak berkeliling lingkungan kantor untuk diperkenalkan pada ruangan-ruangan kerja dan unit-unit yang menempatnya. Hal ini bertujuan agar tata letak organisasi di dalam perusahaan diketahui oleh peserta. Selanjutnya, peserta diperkenalkan kepada anggota tim SBU ICT Solutions, unit tempat peserta ditempatkan sebagai *Fullstack Developer Intern*. Pada waktu yang sama, peserta juga dikenalkan langsung dengan supervisor, yaitu Harival Tivani selaku Technology Operations & Maintenance Senior Officer, yang menjadi penanggung jawab utama selama

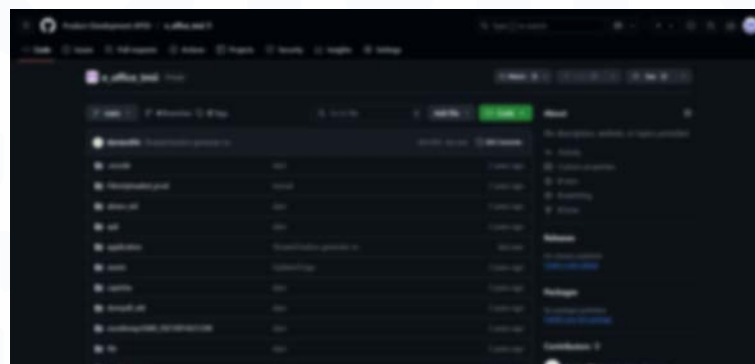
kegiatan magang. Peserta kemudian diperkenalkan kepada General Manager SBU ICT Solutions oleh supervisor dan menjelaskan lingkungan kerja di unit dan proyek apa saja yang akan diberikan kepada peserta.

3.2.2 Proyek E-Office (Minggu ke-1 Sampai Minggu ke-6)

Proyek pengembangan dan perbaikan sistem E-Office TMII dilaksanakan oleh peserta magang selama periode 15 Januari 2025 hingga 21 Februari 2025. Sistem ini sebelumnya terakhir mengalami pembaruan pada tahun 2023 dan memiliki fungsi utama sebagai fasilitas pertukaran dokumen antar individu dalam perusahaan, termasuk pembuatan dan distribusi surat resmi internal maupun pengumuman kepada seluruh karyawan. Selain itu, sistem juga digunakan untuk membuat dokumen dengan format formal yang ditujukan kepada pihak eksternal, seperti vendor atau mitra kerja lainnya.

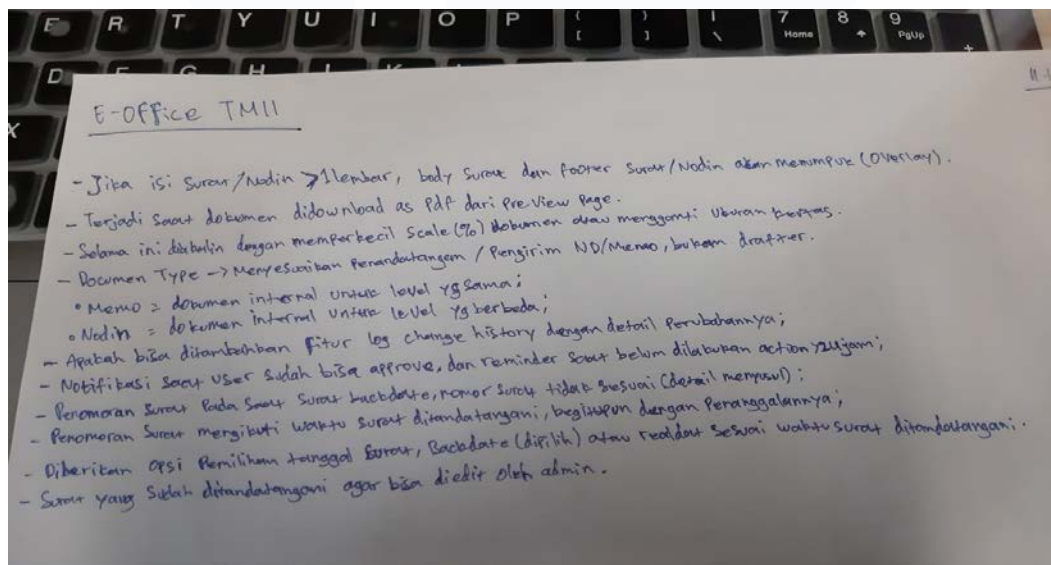
3.2.2.1 Mempelajari Dokumentasi dan Penugasan

Pada penugasan proyek pertama, peserta magang diberikan tugas untuk melakukan pengembangan dan perbaikan sistem e-Office milik Taman Mini Indonesia Indah (TMII). Akses terhadap proyek diberikan melalui *repository* GitHub, di mana peserta diundang oleh supervisor setelah menyerahkan alamat email yang digunakan. Proyek ini telah dibangun menggunakan *framework* CodeIgniter 4 dengan versi PHP 7, menggunakan MySQL sebagai basis data, dan dikelola melalui DBMS HeidiSQL. Untuk bisa melakukan pengujian dan pengembangan pada mesin lokal, peserta diberikan salinan basis data dari sebenarnya yang telah diisi beberapa data dummy juga.



Gambar 3.3 Repository Github E-Office

Proses setup lingkungan pengembangan memakan waktu sekitar tiga hari, termasuk konfigurasi awal yang dibantu langsung oleh supervisor. Proses tersebut diiringi dengan memahami struktur folder dan kode juga. Dalam periode ini, kendala terkait pemahaman proses bisnis internal sistem dihadapi karena tidak ada dokumentasi resmi yang diberikan. Sebelum menerima kebutuhan sistem, peserta terlebih dahulu ditugaskan untuk memahami struktur dokumen dan melakukan eksplorasi terhadap kode program yang ada. Hal ini mencakup penelusuran file-file utama dalam sistem, termasuk *controller*, *model*, dan *view* untuk memahami alur logika aplikasi. Informasi kebutuhan pengembangan disampaikan secara informal melalui catatan tulisan tangan dari supervisor. Tahap awal ini menjadi fondasi penting dalam memahami sistem secara menyeluruh sebelum berlanjut ke proses implementasi dan perbaikan fitur.



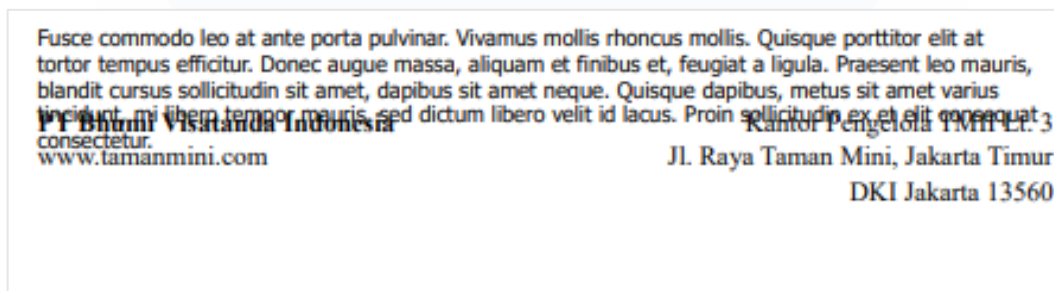
Gambar 3.4 Penugasan E-Office

Penugasan yang diberikan mencakup beberapa kebutuhan perbaikan sistem, antara lain perbaikan tampilan *footer* surat yang mengalami tumpang tindih / *overlay* dengan isi surat, penyesuaian tipe dokumen berdasarkan pihak yang menandatangani surat, penambahan notifikasi *reminder*, serta penyesuaian sistem penomoran surat. Terdapat pula beberapa permintaan dari pengguna yang tidak dikerjakan berdasarkan keputusan supervisor, di

antaranya adalah fitur *log change* karena memerlukan perubahan pada struktur basis data, serta fitur pengeditan surat yang telah ditandatangani oleh admin karena berpotensi menimbulkan penyalahgunaan atau penyelewengan dokumen.

3.2.2.2 Perbaikan Format PDF Surat

Fitur pertama yang diperbaiki oleh peserta magang adalah fitur cetak dokumen (PDF). Permasalahan yang muncul adalah ketika isi surat memiliki panjang melebihi satu halaman, bagian *footer* surat tertumpuk dengan isi suratnya, sehingga mengganggu struktur dan pembacaan dokumen. Solusi sementara yang sebelumnya dilakukan oleh pengguna adalah dengan mengecilkan skala tampilan atau mengubah ukuran kertas pada pengaturan *print*-nya, namun cara ini tidak selalu efektif dan cenderung tidak konsisten.



Gambar 3.5 Sebelum Perbaikan Tampilan PDF

Tampilan surat yang bertumpuk ini sangat mengganggu keterbacaan dan membuat surat yang dibuat menjadi tidak profesional. Untuk mengatasi permasalahan tersebut, logika pada bagian JavaScript ditambahkan untuk mengatur tampilan sebelum proses cetak dimulai. Logika ini bekerja dengan mendeteksi panjang konten yang apabila panjang konten melebihi panjang halaman yang tersedia dikurangi tinggi *footer*, maka sistem secara otomatis memotong bagian konten tersebut dan menyisipkan ruang kosong (*invisible spacer*) di antaranya sebagai ruang untuk *footer*-nya. Dengan pendekatan ini, tampilan *footer* tidak lagi tertindih oleh isi konten surat, sehingga hasil cetak menjadi lebih rapi dan profesional.

3.2.2.3 Penyesuaian Tipe Dokumen

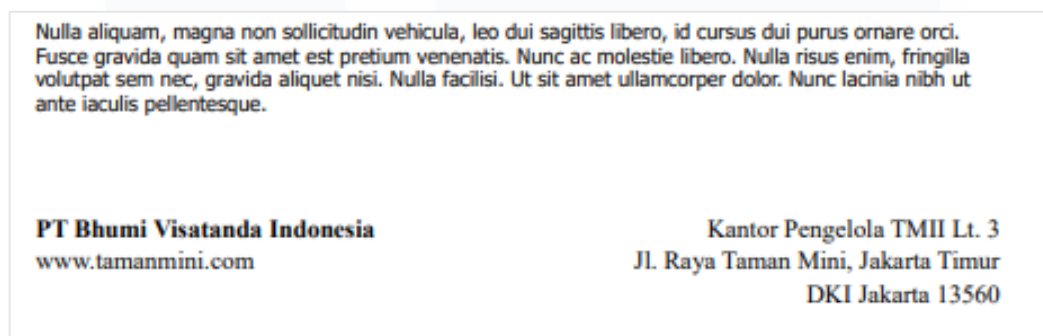
Fitur kedua yang diperbaiki berkaitan dengan logika penentuan tipe dokumen saat proses pembuatan dan pengiriman surat, khususnya ketika surat masih berstatus *draft*. Pada sistem sebelumnya, setiap surat yang dibuat wajib memilih tipe dokumen terlebih dahulu, dan pilihan tujuan surat akan otomatis difilter berdasarkan tipe dokumen serta level jabatan pengirim. Namun, sistem ini menimbulkan masalah ketika surat dikirim sebagai *draft* dan kemudian ingin diedit oleh pihak lain yang bukan pembuat surat awal. Sebagai contoh: seorang staf membuat surat dan mengirimnya sebagai *draft* kepada manajer, dengan tujuan akhir surat tersebut adalah manajer dari divisi lain (jabatan setara). Secara manual, staf harus menetapkan tipe dokumen sebagai Nota Dinas (Nodin) karena tujuan suratnya adalah manajer (jabatan tidak setara). Ketika manajer menerima *draft* tersebut dan ingin mengubah tipe dokumen menjadi *Memo* karena tujuan akhirnya memiliki level jabatan yang setara, sistem tidak mengizinkan pengubahan tipe dokumen. Akibatnya, tipe dokumen menjadi tidak sesuai dengan standar surat perusahaan.

Untuk mengatasi permasalahan ini, perubahan pada mekanisme filter tujuan surat dilakukan oleh peserta. Filter tujuan berdasarkan tipe dokumen dinonaktifkan agar pengguna dapat memilih tujuan surat secara fleksibel, tanpa dibatasi oleh tipe dokumen. Selanjutnya, sistem akan secara otomatis menetapkan tipe dokumen berdasarkan level jabatan tujuan akhir dari *draft*, bukan dari pembuat awal surat. Dengan cara ini, tipe dokumen akan lebih dinamis dan sesuai dengan alur komunikasi internal yang ditetapkan perusahaan.

3.2.2.4 Revisi Format PDF

Solusi sebelumnya untuk fitur cetak surat mengalami revisi karena memiliki kekurangan dalam fleksibilitas dan konsistensi tampilan, terutama saat pengguna mengubah ukuran kertas pada *print preview*. Dengan cara sebelumnya, ruang kosong untuk menghindari tabrakan antara konten surat dan *footer* ditambahkan secara dinamis menggunakan JavaScript. Namun, cara ini

memiliki kelemahan, ketika pengguna memilih ukuran kertas yang berbeda dari kertas awal, maka posisi konten dan ruang kosong akan ikut bergeser. Hal ini menyebabkan konten surat kembali bertumpuk dengan bagian footer, sehingga solusi menjadi tidak efektif. Masalah lainnya adalah JavaScript hanya dapat menambahkan ruang kosong sebelum *print preview* muncul dan tidak memiliki kendali terhadap perilaku *browser* saat pengaturan halaman dilakukan oleh pengguna. Dengan demikian, setelah *print dialog* aktif, penyesuaian yang dilakukan oleh JavaScript tidak lagi berpengaruh terhadap hasil cetakan akhir.



Gambar 3.6 Hasil Format PDF yang Sudah Diperbaiki

Untuk mengatasi masalah tersebut, pendekatan yang lebih stabil diterapkan dengan mengganti struktur layout surat menjadi format *table* HTML. Dalam implementasi ini, isi surat ditempatkan sebagai *body* dari *table*, sementara bagian *footer* ditempatkan sebagai *table footer*. Dengan memanfaatkan kemampuan *native browser* dalam mencetak struktur tabel, footer secara otomatis di-*repeat* di setiap halaman cetak tanpa perlu manipulasi manual menggunakan JavaScript. Hasilnya, posisi konten dan *footer* menjadi lebih presisi dan konsisten meskipun terjadi perubahan ukuran atau orientasi kertas di *print preview*.

3.2.2.5 Penyesuaian Nomor Surat

Perbaikan selanjutnya yang dilakukan peserta magang berkaitan dengan sistem penanggalan dan penomoran surat. Sebelumnya, tanggal surat secara otomatis mengikuti waktu pembuatan dokumen dan tidak bisa disesuaikan saat

edit oleh pengguna lain. Hal ini menjadi permasalahan ketika dokumen perlu penyesuaian untuk tanggal pada dokumen, karena sistem tidak menyediakan pilihan *back date* maupun *real date* secara fleksibel. Selain itu, penomoran surat juga mengikuti tanggal pembuatan, bukan tanggal saat surat ditandatangani, yang menyebabkan ketidaksesuaian pada urutan nomor jika surat dikirim tidak pada hari yang sama dengan pembuatannya.

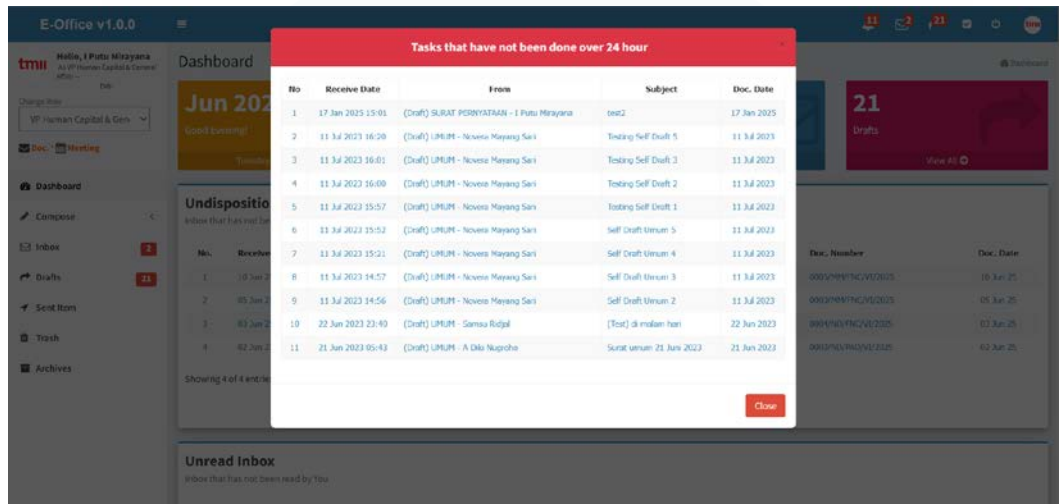
Gambar 3.7 Tampilan Pilihan Tanggal Dokumen pada Edit Draft

Untuk mengatasi hal ini, perancangan solusi dilakukan untuk memungkinkan pengguna memilih jenis tanggal surat pada saat proses pengeditan. Dua opsi disediakan dengan *radio button* dengan nilai *realdte* untuk menggunakan tanggal hari ini, dan *backdate* yang membuat pengguna bisa memilih tanggal dengan lebih fleksibel dengan nilai awal (*default*) adalah tanggal pembuatan surat. Tanggal yang dipilih ini akan ditampilkan sebagai tanggal resmi dalam isi surat. Namun, sistem penomoran tetap menggunakan tanggal saat surat ditandatangani (*real date*) untuk menjaga konsistensi dan urutan nomor sesuai dengan alur waktu. Cara ini digunakan dengan pengaturan tampilan dan logika saat surat diedit sebelum ditandatangani agar tidak perlu merubah struktur pada basis data. Hak untuk memilih tanggal ini dibatasi hanya pada pengguna yang sedang mengedit surat atau yang diminta melakukan *approval*, sedangkan dokumen yang sudah ditandatangani tidak dapat diubah dan bersifat final.

3.2.2.6 Menambahkan Fitur Notifikasi

Kebutuhan pengguna selanjutnya adalah fitur notifikasi. Fitur ini diharapkan untuk secara otomatis mengingatkan pengguna saat *draft* surat belum ditindaklanjuti selama lebih dari 24 jam sejak pembuatan. Permintaan

penambahan fitur ini muncul dari pengguna sistem yang menginginkan adanya pengingat tugas yang belum dikerjakan agar tidak terlewatkan.



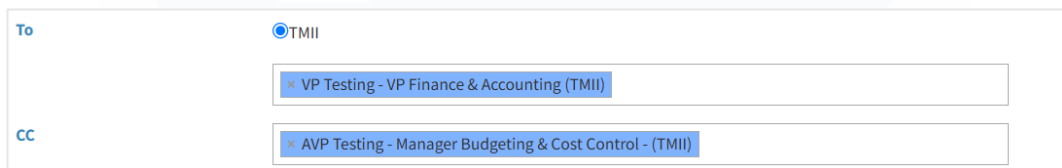
Gambar 3.8 Tampilan Notifikasi E-Office

Untuk mewujudkan hal tersebut, notifikasi dirancang dalam bentuk modal *popup* yang muncul secara otomatis ketika pengguna pertama kali masuk ke sistem. Modal ini menampilkan daftar dokumen yang telah melewati batas waktu 24 jam dan belum mengalami perubahan status. Setiap entri dalam daftar ini dilengkapi dengan tautan langsung menuju halaman pengeditan draft yang bersangkutan, sehingga memudahkan pengguna untuk segera menindaklanjuti dokumen tersebut.

Hal tersebut dilakukan dengan menambahkan logika pada modal untuk melakukan *filter* terhadap seluruh *draft* yang dimiliki oleh pengguna yang sedang *login*. Draft yang memenuhi kriteria waktu lebih dari 24 jam akan dimasukkan dalam daftar notifikasi. Untuk memastikan notifikasi tidak muncul berulang-ulang dalam satu sesi, peserta menambahkan penanda (*flag*) pada *controller* untuk mendeteksi apakah modal notifikasi sudah pernah ditampilkan selama sesi *login* berjalan. Dengan pendekatan ini, pengguna hanya melihat notifikasi sekali saja, sehingga tidak selalu muncul saat kembali ke *home*.

3.2.2.7 Memperbaiki Field Tujuan Surat

Permasalahan ditemukan pada bagian pengeditan surat, di mana field tujuan surat / *To* dan tembusan / *CC* berada dalam satu kolom data yang sama. Hal ini menyebabkan kebingungan bagi pengguna dalam membedakan antara penerima utama dan penerima tembusan. Untuk mengatasi permasalahan tersebut, kolom *receiveAs* pada tabel *tx_inbox_detail* dimanfaatkan sebagai penanda atau pembeda antara jenis penerima. Dalam proses penarikan data, sistem akan memeriksa nilai dari *receiveAs*, jika bernilai sebagai penerima utama, maka data akan ditampilkan di *field To*, sedangkan jika bernilai sebagai tembusan, maka akan ditampilkan di *field CC*. Dengan pemisahan logika ini, informasi yang akan diberikan kepada penerima surat dapat disajikan dengan lebih jelas dan akurat saat proses pengeditan maupun peninjauan surat.



The screenshot shows a form with two sections: 'To' and 'CC'. The 'To' section has a radio button labeled 'TMII' which is selected. Below it is a text box containing 'VP Testing - VP Finance & Accounting (TMII)'. The 'CC' section has a text box containing 'AVP Testing - Manager Budgeting & Cost Control - (TMII)'.

Gambar 3.9 Tampilan Tujuan dan Tembusan Surat Pada Halaman Edit

Tujuan dan tembusan ini dikhususkan untuk pembuatan surat internal. Format yang berbeda ditampilkan pada pembuatan surat external. Tujuan dan tembusan internal bisa ditujukan kepada lebih dari satu orang untuk memudahkan proses pengiriman surat.

3.2.2.8 Testing dan Debugging

Pada tahap ini, pengujian sistem dilakukan dengan menjalankan keseluruhan alur bisnis dari awal hingga akhir, tujuannya untuk memastikan bahwa setiap fitur berjalan sesuai dengan kebutuhan pengguna. Proses pengujian dilakukan secara manual dengan memeriksa apakah data yang dimasukkan telah tersimpan dengan benar di dalam basis data, apakah surat sampai kepada penerima yang tepat, serta memastikan informasi seperti tanggal dan nomor surat tampil dengan akurat. Validasi dilakukan bersama supervisor untuk memastikan bahwa hasil pengembangan sesuai dengan

standar dan ekspektasi sistem yang telah ditetapkan. Proses *debugging* dilaksanakan secara bertahap, dengan mengidentifikasi kesalahan fungsional atau tampilan dan memperbaikinya hingga sistem berjalan secara optimal. Hasil dari *testing* tidak menunjukkan adanya masalah, dan semua sudah berjalan sebagaimana mestinya dengan data yang disimpan dan ditampilkan sudah tepat dengan kebutuhan.

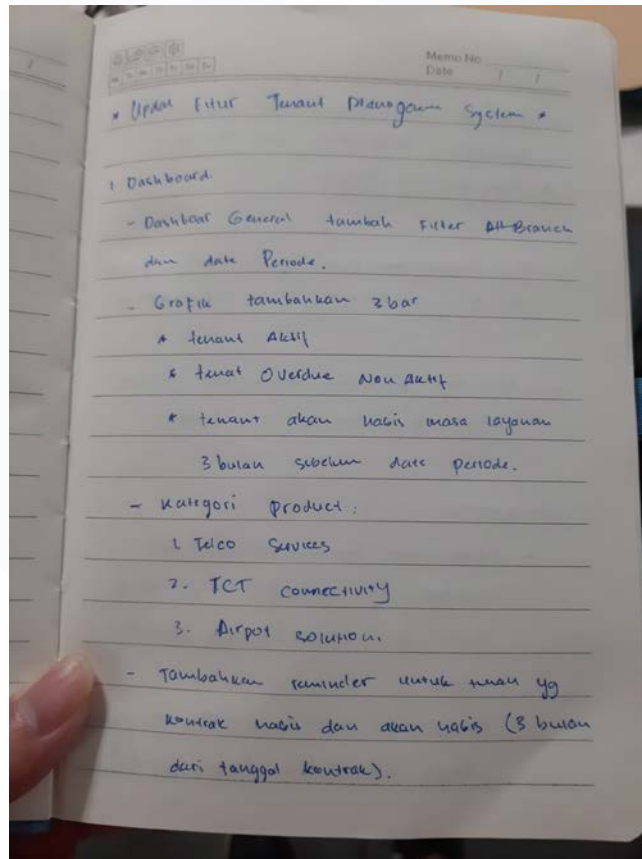
3.2.2.9 Briefing Penugasan Tenant Management



Gambar 3.10 Ruang Rapat untuk Briefing Tenant Management

Setelah proyek E-Office TMII mendekati tahap akhir, penugasan baru berupa pengembangan sistem Tenant Management milik SBU ICT Solutions diberikan kepada peserta. *Briefing* proyek ini dilakukan secara langsung di ruang rapat oleh supervisor dan para pengguna sistem. Pada sesi tersebut,

peserta diperkenalkan dengan sistem yang telah dikembangkan menggunakan *framework* Laravel Voyager, namun belum digunakan secara menyeluruh dalam operasional internal perusahaan.



Gambar 3.11 Catatan Penugasan Tenant Management

Dalam pemaparan tersebut, pengguna sistem mempresentasikan alur sistem secara umum dan menjelaskan bahwa hampir seluruh komponen sistem memerlukan penyempurnaan, baik dari sisi struktur data, alur kerja, maupun tampilan antarmuka. Waktu yang diberikan untuk memahami sistem tidak diberi batas karena peserta belum menerima akses terhadap kode sumber atau basis data. Sebagai hasil dan pengingat dari *briefing* ini, pencatatan berbagai kebutuhan sistem yang disampaikan dilakukan peserta sebagai penugasan dalam proses pengembangan dan perbaikan berikutnya, yang dikerjakan secara mandiri.

3.2.2.10 Membuat Dokumentasi dan *Code Review*

Setelah seluruh kebutuhan pengguna pada proyek telah berhasil diimplementasikan, tahap akhir pengembangan dilanjutkan dengan melakukan peninjauan ulang terhadap kode program. Proses ini dilakukan bersama supervisor, yang memeriksa aspek kerapian penulisan kode, keterbacaan logika, dan efisiensi implementasi fungsi yang digunakan. Tidak ada standar khusus penulisan kode di dalam tim, yang terpenting gaya penulisan kodenya jelas dan mudah dipahami.

```
// checkpoint untuk notifikasi task hanya muncul 1x
$data['showModal'] = false;
if ($this->session->userdata('firstLogin')) {
    $data['showModal'] = true;
    $this->session->unset_userdata('firstLogin'); // Hapus setelah ditampilkan sekali
}
```

Gambar 3.12 Contoh Comment pada Kode

Setelah kode dinyatakan sesuai, komentar-komentar penjelasan pada bagian-bagian penting dari kode ditambahkan oleh peserta, dengan tujuan memudahkan pemahaman bagi pengembang berikutnya. Setelah itu, proses *push* kode dilakukan ke *repository* GitHub ke dalam *branch* baru yang terpisah dari *main branch*. Hal ini dilakukan agar supervisor dapat melakukan pengecekan akhir bagian mana saja yang berubah tanpa risiko konflik kode.

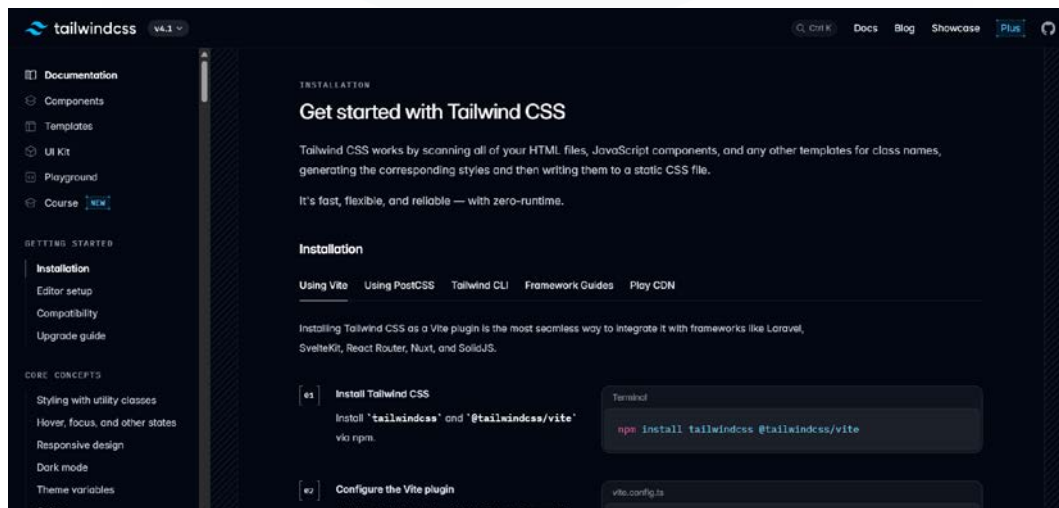
3.2.3 Pembelajaran (Minggu Ke-7 Sampai Minggu Ke-14)

Setelah menyelesaikan seluruh penugasan pada proyek *E-Office*, peserta awalnya dijadwalkan untuk langsung mengerjakan proyek Tenant Management milik SBU ICT Solutions. Akan tetapi, pelaksanaan proyek mengalami penundaan karena repositori GitHub proyek tersebut masih dipegang oleh *developer* sebelumnya dan membutuhkan waktu untuk proses serah terima akses kepada supervisor. Sambil menunggu akses tersebut, peserta diarahkan oleh supervisor untuk memanfaatkan waktu dengan mempelajari teknologi-teknologi pendukung yang relevan dan berpotensi dibutuhkan dalam proyek berikutnya.

Selama periode ini, pembelajaran dilakukan secara mandiri tanpa modul atau panduan khusus dari perusahaan. Pembelajaran dilakukan secara aktif di kantor setiap hari seperti hari kerja biasanya, meskipun tidak terdapat batas waktu resmi dari perusahaan mengenai kapan akses proyek akan diberikan. Proses ini berlangsung kurang lebih dari minggu ke-7 hingga minggu ke-14 masa magang. Hasil pembelajaran tidak diwajibkan untuk dilaporkan secara formal.

3.2.3.1 Mempelajari Tailwind CSS

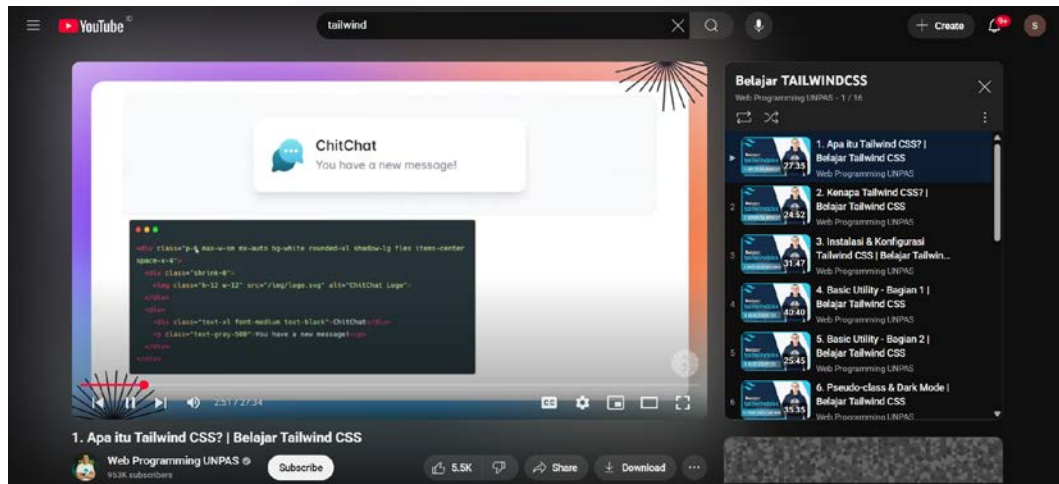
Selama masih menunggu akses proyek berikutnya, peserta diarahkan oleh supervisor untuk memanfaatkan waktu dengan mempelajari teknologi pendukung yang relevan dengan kebutuhan pengembangan sistem, salah satunya adalah Tailwind CSS. Tailwind CSS merupakan *utility-first* CSS *framework* yang memungkinkan pengembangan antarmuka pengguna secara cepat dan fleksibel. *Framework* ini lumayan populer dikalangan pengembang *website*.



Gambar 3.13 Website Dokumentasi Tailwind CSS

Tailwind CSS dipelajari secara mandiri selama kurang lebih satu minggu. Proses pembelajaran dilakukan melalui dokumentasi resmi yang tersedia di situs resmi Tailwind CSS dan beberapa video pembelajaran dari YouTube. Pembelajaran difokuskan pada pemahaman dasar penggunaan kelas-kelas

utility yang disediakan Tailwind, seperti pengaturan *margin*, *padding*, warna, dan responsivitas tampilan.



Gambar 3.14 Video Tutorial Tailwind CSS

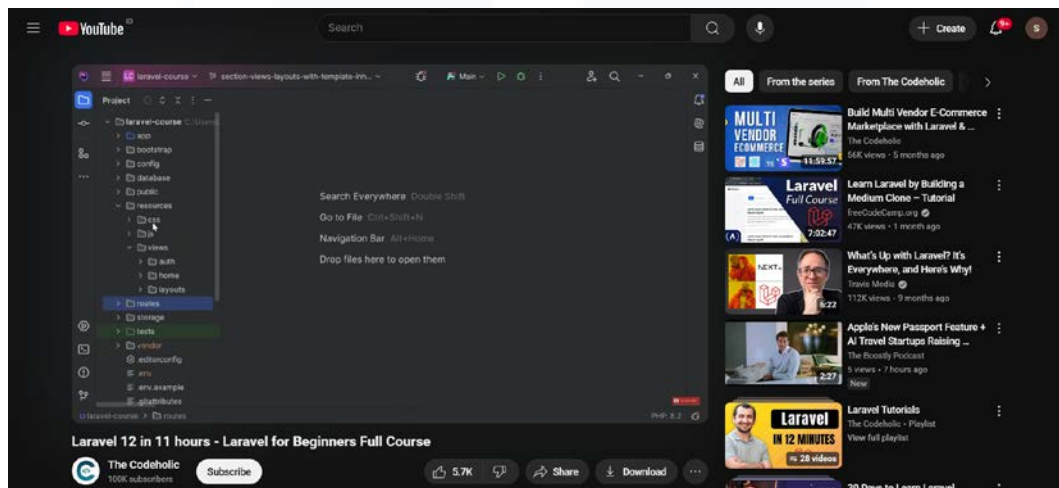
Untuk menguji pemahaman, percobaan membuat sebuah layout sederhana berupa kalkulator sebagai latihan praktis juga telah dilakukan. Tantangan utama yang dihadapi peserta adalah banyaknya variasi class yang tersedia, sehingga dibutuhkan ketelitian dalam memilih class yang tepat untuk menghasilkan tampilan yang diinginkan. Meskipun demikian, fleksibilitas yang dimiliki Tailwind CSS dirasa lebih tinggi dibandingkan *framework* CSS tradisional seperti Bootstrap, karena kontrol lebih rinci terhadap styling elemen dapat diberikan.

3.2.3.2 Mempelajari Laravel

Selain Tailwind CSS, arahan lainnya yang didapatkan peserta dari supervisor adalah mempelajari Laravel. Hal ini dikarenakan Laravel merupakan sebuah *framework* PHP yang masih sangat aktif dikembangkan dan digunakan secara luas oleh komunitas developer. *Framework* ini direkomendasikan karena tingkat stabilitasnya, pembaruan yang konsisten, serta dokumentasi yang lengkap.

Meskipun pengalaman sebelumnya dalam penggunaan Laravel telah dimiliki, pembelajaran tetap difokuskan pada penguatan konsep dasar seperti

penggunaan *routing*, pembuatan *controller*, migrasi *database*, dan proses *seeding*. Sebagai media latihan, proyek kalkulator yang dibuat pada pembelajaran Tailwind sebelumnya juga dimanfaatkan peserta untuk menerapkan materi yang dipelajari.

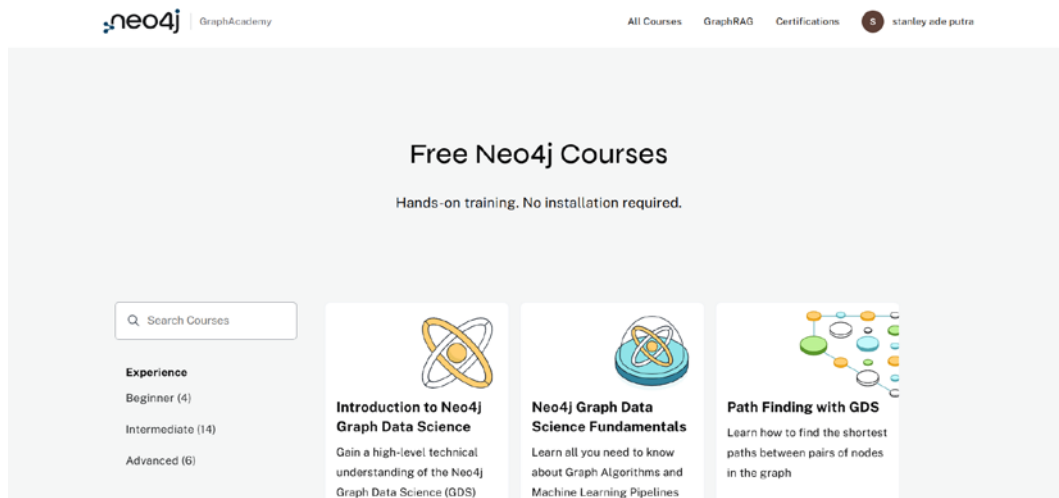


Gambar 3.15 Video Tutorial Laravel

Metode pembelajaran dilakukan secara mandiri melalui dokumentasi resmi Laravel dan video pembelajaran di YouTube. Selama proses tersebut, perbedaan ditemukan pada struktur folder dan pendekatan pengembangan antara Laravel versi 10 yang pernah digunakan dengan versi 12 yang sedang dipelajari. Perbedaan tersebut menjadi tantangan sekaligus peluang untuk memahami evolusi *framework* ini secara lebih komprehensif. Hal ini juga menandakan bahwa Laravel terus mengalami peningkatan dari segi fitur dan fleksibilitas, sehingga sangat relevan untuk digunakan dalam pengembangan aplikasi web ke depannya.

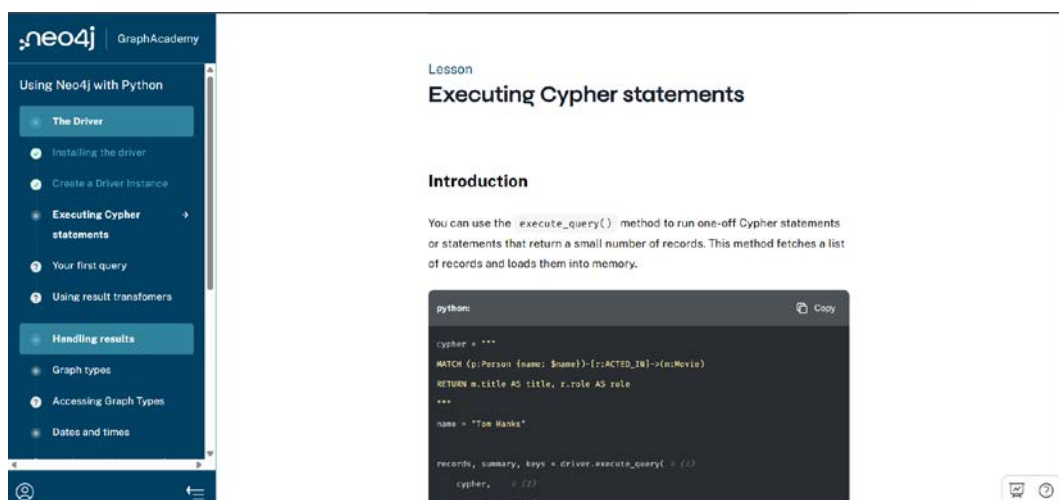
3.2.3.3 Mempelajari *Graph Database*

Pada minggu ke-11 hingga ke-14 masa magang, repositori proyek selanjutnya masih belum juga diterima. Maka dari itu, pembelajaran dilanjutkan dengan memperluas pengetahuan di bidang basis data dengan mempelajari teknologi *graph database*. Pemilihan topik ini didasarkan atas ketertarikan pribadi peserta serta relevansinya dengan penjurusan yang diambil peserta, yaitu *database*.



Gambar 3.16 Website Neo4J GraphAcademy

Graph database merupakan jenis basis data yang memiliki keunggulan dalam merepresentasikan hubungan antar data secara eksplisit dan fleksibel. Topik ini dipelajari melalui *platform* pembelajaran daring resmi dari Neo4j, yaitu Graph Academy. *Platform* ini menyediakan berbagai kursus yang dirancang bertingkat, dimulai dari tingkat dasar hingga tingkat lanjut. Kursus yang diikuti yaitu pembelajaran tentang konsep fundamental Neo4j, dasar-dasar bahasa *query* Cypher, serta prinsip-prinsip dasar pemodelan *graph*.



Gambar 3.17 Contoh Course Neo4J

Selama proses pembelajaran, Materi yang disediakan, simulasi studi kasus, dan kuis interaktif diikuti dan dikerjakan secara langsung pada *platform*

tersebut. Dari pembelajaran tersebut, pemahaman tentang elemen-elemen utama dalam graph database telah diperoleh, seperti node, relationship, dan property, serta bagaimana cara melakukan query menggunakan bahasa Cypher. Di akhir setiap kursus, sertifikat penyelesaian didapatkan sebagai bukti keberhasilan dalam materi yang telah dipelajari. Salah satu hal yang paling menarik bagi peserta adalah kemampuan *graph database* untuk mendefinisikan relasi yang memiliki nama berbeda-beda antar entitas, yang secara konseptual lebih fleksibel dibandingkan model relasional biasa. Meskipun teknologi ini belum secara langsung diterapkan dalam proyek magang yang sedang berlangsung, pemahaman ini diyakini peserta akan berguna dalam proyek-proyek mendatang yang membutuhkan manajemen data dengan relasi yang kompleks.

3.2.4 Proyek BRS (Minggu Ke-14 Sampai Ke-15)

Kesempatan diberikan kepada peserta untuk terlibat dalam proyek Baggage Reconciliation System (BRS) PT Gapura yang merupakan proyek aktif milik supervisor. Sistem ini dirancang untuk melakukan pelacakan terhadap barang bawaan penumpang pesawat yang masuk ke dalam bagasi, dimulai dari proses *check-in* hingga proses *check-out*. Tujuan utama dari sistem ini adalah untuk memastikan setiap barang yang dibawa oleh penumpang tercatat dan terpantau dengan baik selama berada dalam kendali pihak bandara. Proyek ini menggunakan Laravel versi 8 dengan PHP versi 8 dan sudah berjalan dengan menggunakan MySQL.

3.2.4.1 Testing Aplikasi di Lapangan

Kesempatan didapatkan peserta untuk turut serta dalam kegiatan pengujian aplikasi secara langsung di lapangan sebagai bagian dari proyek BRS. Aplikasi tersebut diinstal pada perangkat Android Zebra yang difungsikan sebagai pemindai tag barang penumpang. Kegiatan ini bertujuan untuk memastikan bahwa sistem dapat berjalan sesuai dengan fungsinya serta mengidentifikasi permasalahan yang terjadi atau berpotensi muncul dalam proses penggunaannya di lingkungan operasional. Pengujian dilakukan di area

outbound dan *makeup* area, dengan melibatkan tim dari pihak supervisor serta petugas lapangan yang memang menjadi pengguna langsung dari sistem tersebut. Dalam kegiatan ini, peserta turut berperan aktif membantu proses instalasi aplikasi ke perangkat yang tersedia serta mendokumentasikan seluruh proses pengujian melalui pengambilan foto dan video untuk keperluan evaluasi dan pelaporan.



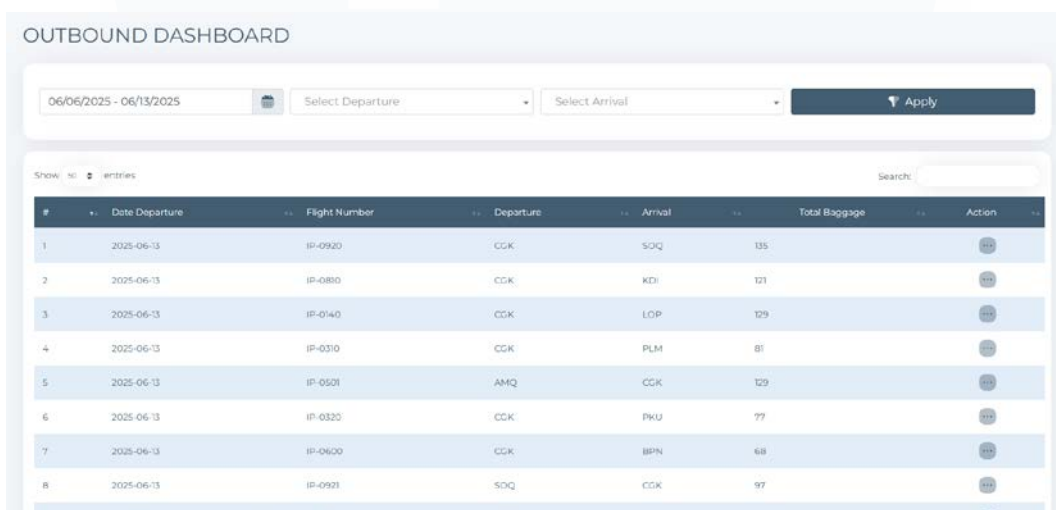
Gambar 3.18 Pengujian Aplikasi di Makeup Area

Pada pengujian ini, dilakukan pengecekan apakah dari semua proses mulai dari barang turun dari pesawat sampai barang diletakkan di *conveyor belt* untuk diambil penumpang sudah berjalan dengan benar dan tidak ada barang penumpang yang tidak ter-*scan*. Selain dilihat secara langsung, pengecekan ini dilihat juga melalui aplikasinya. Masalah yang sering terjadi pada proses ini adalah tag yang tidak ter-*scan* karena proses kerja yang dituntut untuk cepat sehingga petugas terburu-buru. Dari sisi sistem, tidak ada masalah yang terjadi.

3.2.4.2 Membuat *Dashboard*

Pada tahap ini, kontribusi dalam pengembangan sistem *monitoring* untuk proyek BRS diberikan kepada peserta dengan membuat *public dashboard* yang dapat diakses tanpa proses *login*. *Dashboard* ini bertujuan untuk memberikan kemudahan bagi supervisor lapangan dalam memantau data bagasi penumpang

secara real-time. Perancang dua jenis *dashboard* dilakukan peserta, yaitu untuk proses *outbound* dan *inbound*. Informasi yang ditampilkan meliputi tanggal keberangkatan, nomor penerbangan, dan total bagasi, serta detail masing-masing bagasi seperti nama pemilik tas, nomor tempat duduk, dan nomor tag bagasi. Seluruh data disajikan dalam bentuk tabel yang dilengkapi fitur pencarian dan filter untuk memudahkan pengguna dalam menemukan informasi yang dibutuhkan. Tampilan *dashboard* ini dirancang secara mandiri, kemudian *routing* dan *controller*-nya diatur, sementara untuk bagian *model* menggunakan yang telah dibuat sebelumnya. Seluruh *dashboard* ini diintegrasikan dalam sistem dan di-*hosting* melalui server internal perusahaan.



OUTBOUND DASHBOARD

06/06/2025 - 06/13/2025 Select Departure Select Arrival Apply

Show: 10 entries Search:

#	Date Departure	Flight Number	Departure	Arrival	Total Baggage	Action
1	2025-06-13	IP-0920	CGK	SOQ	135	
2	2025-06-13	IP-0880	CGK	KDI	121	
3	2025-06-13	IP-0140	CGK	LOP	129	
4	2025-06-13	IP-0310	CGK	PLM	81	
5	2025-06-13	IP-0501	AMQ	CGK	129	
6	2025-06-13	IP-0320	CGK	PKU	77	
7	2025-06-13	IP-0600	CGK	BPJ	68	
8	2025-06-13	IP-0921	SOQ	CGK	97	

Gambar 3.19 Tampilan Dashboard BRS

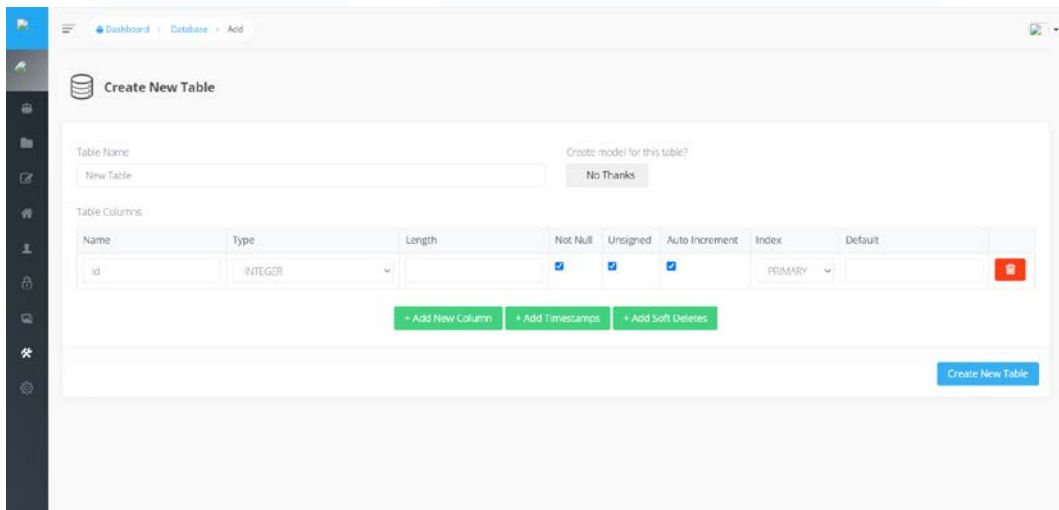
3.2.5 Proyek Tenant Management (Minggu Ke-15 Sampai Ke-19)

Pada minggu ke-15, akses ke repositori GitHub proyek Tenant Management akhirnya didapatkan peserta. Sistem ini digunakan oleh SBU ICT Solutions untuk mengelola pelanggan yang membeli layanan dari mereka, seperti pemasangan Wi-Fi, internet, VLAN, dan layanan terkait lainnya. Proyek ini dibangun menggunakan Laravel versi 6 dengan PHP versi 8, serta basis data MySQL. Selain itu, sistem ini juga memanfaatkan Laravel Voyager, sebuah *library* yang menyediakan antarmuka administrasi berbasis CRUD secara otomatis untuk mempercepat proses pengembangan. Sistem ini

memerlukan sejumlah penyesuaian karena adanya perubahan dalam proses bisnis yang terjadi di alur operasional SBU ICT Solutions.

3.2.5.1 Menambahkan *Master Table Business Line* dan *Product Category*

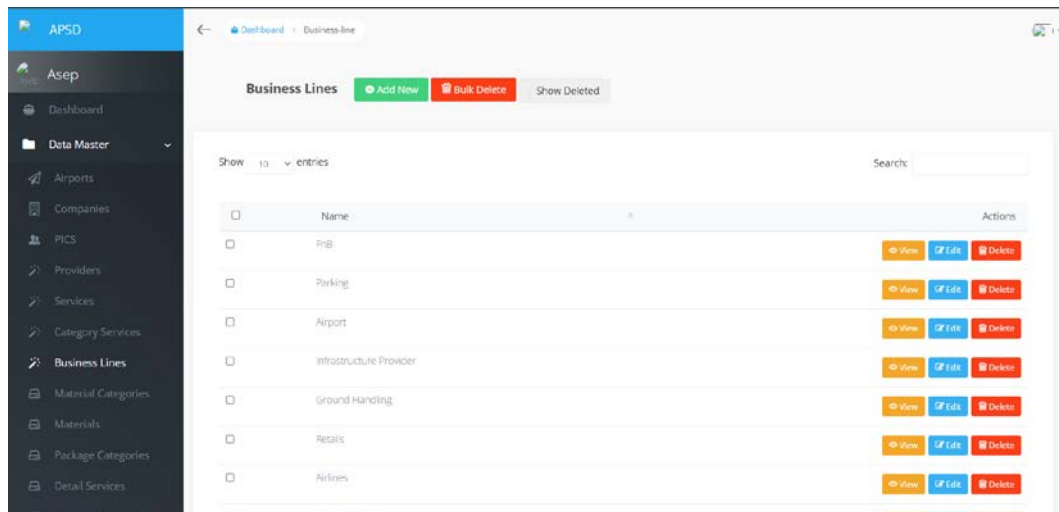
Pada tahap awal pengembangan, dilakukan penambahan tabel master oleh peserta, yaitu tabel *ms_business_line* dan *ms_product_categories*. Proses penambahan ini memanfaatkan fitur Laravel Voyager yang telah terintegrasi langsung dengan basis data. Tabel dapat dibuat melalui antarmuka Voyager dengan hanya memasukkan nama tabel, nama kolom, serta tipe data masing-masing kolom.



Name	Type	Length	Not Null	Unsigned	Auto Increment	Index	Default
id	INTEGER		☒	☒	☒	PRIMARY	

Gambar 3.20 Tampilan Menambah Tabel pada Database

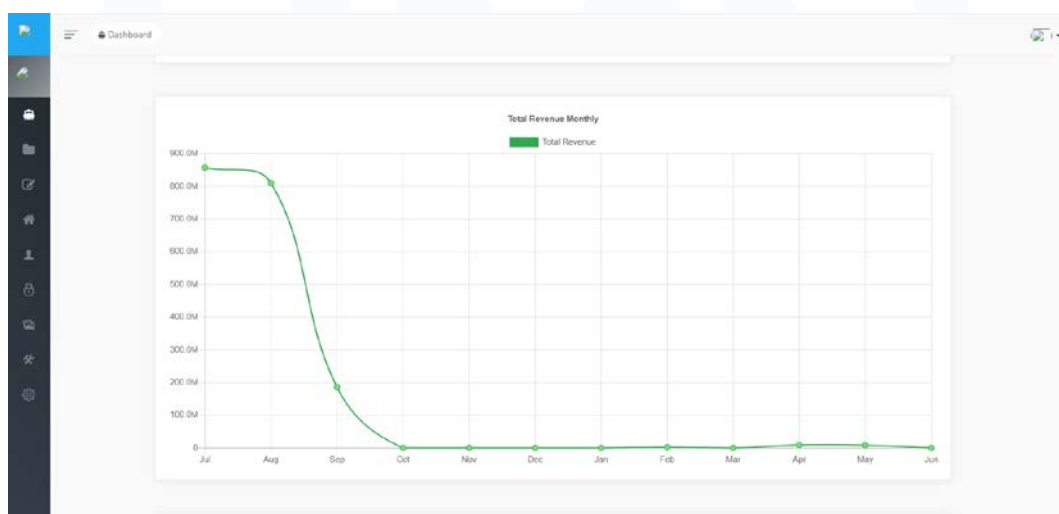
Setelah tabel berhasil ditambahkan, model dimasukkan kedalam direktori proyek agar dikenali oleh Voyager. Secara otomatis, Voyager kemudian menghasilkan antarmuka formulir yang memungkinkan pengguna untuk melakukan operasi tambah, ubah, maupun hapus data. Penyesuaian tambahan juga dapat dilakukan, seperti pengubahan tipe input pada form sesuai kebutuhan. Setelah seluruh proses pembuatan tabel dan formulir selesai, data awal dimasukkan langsung melalui formulir tersebut.



Gambar 3.21 Tampilan Form Setelah Tabel Dibuat dan Diisi Data

3.2.5.2 Menyesuaikan *Dashboard*

Pekerjaan kedua yang dilakukan oleh peserta difokuskan pada pembaruan tampilan *dashboard* sistem. *Dashboard* tersebut terdiri dari dua bagian utama, yaitu *dashboard* umum (*general*) yang menampilkan informasi seperti jumlah tenant, kategori tenant, serta layanan yang digunakan, dan *dashboard* keuangan (*finance*) yang menyajikan data pendapatan serta status pembayaran dari masing-masing *tenant*.

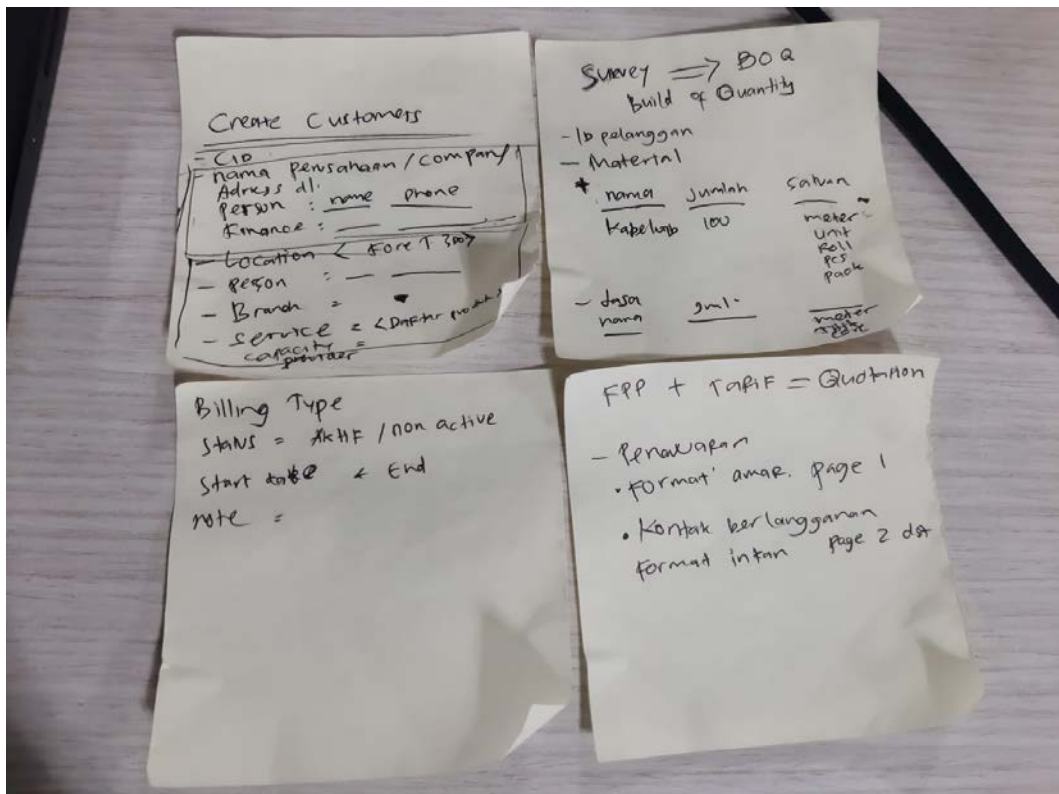


Gambar 3.22 Grafik pada Dashboard Finance

Sejumlah grafik tambahan telah dibuat oleh peserta berdasarkan template grafik yang sudah tersedia pada sistem sebelumnya. Pada *dashboard* umum, sebuah *bar chart* ditambahkan untuk menampilkan status *tenant* berdasarkan klasifikasi aktif, *overdue*, atau nonaktif. Sementara pada *dashboard* keuangan, ditambahkan *line chart* yang menampilkan tren pendapatan, yang dapat difilter berdasarkan bulan atau tahun untuk memudahkan analisis historis. Grafik-grafik ini dirancang dengan menggunakan *library* Chart.js, dan datanya diambil berdasarkan dari basis data asli yang kemudian disalin ke lingkungan lokal milik peserta.

3.2.5.3 Presentasi Progress & Penambahan Kebutuhan

Sesi presentasi terkait progres proyek telah diselenggarakan secara langsung di kantor oleh General Manager SBU ICT Solutions. Dalam kegiatan tersebut, para anggota SBU ICT selaku calon pengguna sistem turut diundang untuk memberikan umpan balik. Progres pengerjaan proyek dipaparkan oleh peserta, dan sejumlah masukan langsung disampaikan oleh para pengguna. Selain itu, kebutuhan tambahan serta penyesuaian sistem juga disampaikan agar sistem dapat lebih disesuaikan dengan kebutuhan operasional yang ada. Seluruh tanggapan dan permintaan yang diberikan oleh pengguna dicatat dan dijadikan acuan untuk pengembangan sistem lebih lanjut.



Gambar 3.23 Catatan Hasil Presentasi Progress

Kebutuhan tambahan pengguna digambarkan pada catatan tersebut. Pengguna membutuhkan Customer ID pada *form customer* atau *tenant* dengan *field-field* yang dibutuhkan. Kemudian *form* untuk *survey* juga digambarkan untuk memperjelas kebutuhan pengguna.

3.2.5.4 Menambahkan Form Tenant dan Menyesuaikan Form Company

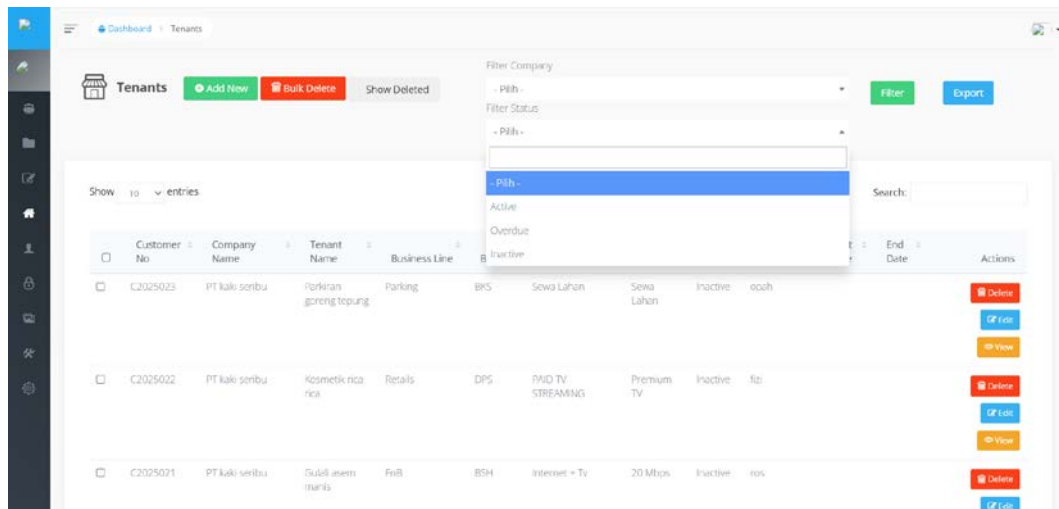
Berdasarkan hasil evaluasi pada sesi presentasi sebelumnya, dilakukan penyesuaian terhadap formulir pengisian data pelanggan. Sebelumnya, sistem menerapkan satu *Customer ID* untuk satu perusahaan, sehingga seluruh layanan yang digunakan berada dalam satu entitas. Namun, struktur tersebut diubah menjadi satu perusahaan memiliki satu *tenant*, dan setiap *tenant* dengan satu layanan memiliki satu *Customer ID* tersendiri. Oleh karena itu, kolom *Customer ID* pada formulir perusahaan dihilangkan dan dipindahkan ke formulir *tenant*, yang sebelumnya belum ada pada sistem.

Gambar 3.24 Tampilan Form Tenant dengan Penyesuaiannya

Formulir tenant kemudian dibuat dan disesuaikan, termasuk penambahan isian untuk *PIC tenant*, *Service* dan *Provider*. Karena sistem menggunakan Laravel Voyager sebagai *admin interface*, fleksibilitas pengelolaan formulir terbatas, sehingga penyesuaian perlu dilakukan secara manual pada *view* dan *controller*. Formulir perusahaan kini dapat digunakan untuk menyimpan data perusahaan beserta informasi *PIC*-nya, sedangkan formulir tenant digunakan untuk menyimpan data *PIC tenant*, *Service* dan *Provider*. Penyesuaian ini diperlukan karena Voyager tidak mendukung pengisian data ke beberapa tabel dalam satu formulir secara langsung melalui antarmuka, sehingga modifikasi kode harus dilakukan secara manual agar proses input dapat berjalan sebagaimana mestinya.

3.2.5.5 Penyesuaian Grafik pada *Dashboard*

Berdasarkan masukan yang disampaikan langsung oleh pengguna, penyesuaian pada grafik status tenant di *dashboard* telah dilakukan. Grafik yang dibangun menggunakan Chart.js dimodifikasi agar setiap *bar* dapat diklik dan mengarahkan pengguna ke halaman daftar *tenant* yang terpisah. Pada halaman tersebut, data secara otomatis difilter berdasarkan status yang dipilih.



Gambar 3.25 Fitur Filter Pada Tampilan Daftar Tenant

Untuk mendukung fungsionalitas ini, logika pemrosesan status tenant ditambahkan pada *controller*. Status "Overdue" diberikan apabila *end_date* kurang dari tiga bulan ke depan dan status tenant masih aktif; status "Aktif" apabila *end_date* melebihi tanggal saat ini dan status masih aktif; dan sisanya dikategorikan sebagai "Non Aktif".



Gambar 3.26 Tampilan Grafik Status yang Dapat Memfilter Daftar Tenant

Setelah filter berhasil diterapkan, fungsi *onClick* pada grafik ditambahkan agar saat grafik diklik, pengguna secara otomatis diarahkan ke halaman daftar tenant dengan parameter *query string* sesuai status yang diklik. Seluruh implementasi dilakukan menggunakan Chart.js, dan tampilan daftar

tenant diatur agar secara otomatis merespons parameter *filterStatus* yang terdapat dalam query string.

3.2.5.6 Menyesuaikan *Form Survey*

Form survei digunakan untuk mencatat kebutuhan material dan jasa yang diperlukan dalam pelaksanaan layanan yang dipesan oleh pelanggan, termasuk jumlah dari masing-masing item tersebut. Penyesuaian kode dilakukan pada bagian *controller* dan *view* agar *form* dapat menginput data ke beberapa tabel sekaligus. Data dikirim dari *form* dalam bentuk *array*, kemudian dipecah dalam *controller* menjadi beberapa entri sesuai dengan panjang *array* sebelum disimpan ke dalam database. Setiap entri kebutuhan berdasarkan pada data master yang tersimpan di tabel *ms_material*, di mana jenis item diklasifikasikan melalui kolom *category* yang berisi nilai 'material' atau 'jasa'. Relasi antara kebutuhan dalam survei dan data master dilakukan melalui tabel *tx_boq_survey*, yang menghubungkan entri dari *tx_survey* dengan *ms_material*. *Form* ini dirancang agar baris input dapat ditambah secara dinamis sesuai kebutuhan pengguna internal.

Provider	Service	Category	Quantity	Location
1 Icon+	METRO-E	1 Mbps	1	Terminal 1 lantai 2 Bandara Udara Internasional Sultan Syarif Kasim II

Gambar 3.27 Tampilan Form Survey

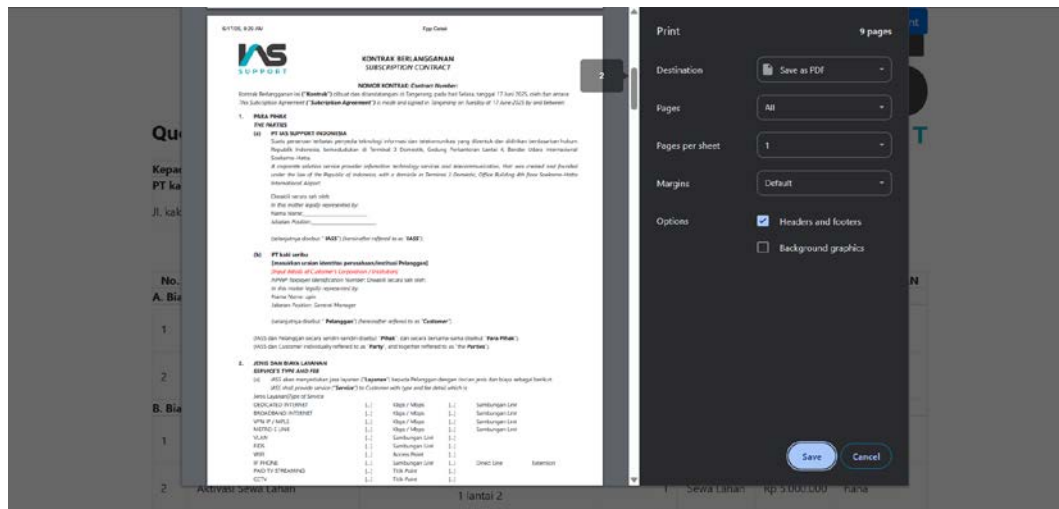
Setelah proses registrasi selesai dilakukan, tahap survei segera dijalankan. Langkah pertama dimulai dengan memilih nomor registrasi melalui *dropdown*, yang secara otomatis akan memfilter dan menampilkan nomor

pelanggan terkait. Setelah nomor pelanggan dipilih, detail layanan beserta lokasinya akan ditampilkan secara otomatis. Selanjutnya, data seperti *distance* dan *type* diisi berdasarkan hasil observasi langsung yang dilakukan oleh pengguna di lokasi pemasangan.

Selain itu, daftar material yang diperlukan untuk mendukung layanan juga diinput dalam *form* ini, lengkap dengan jumlahnya. Jumlah material yang dibutuhkan dapat lebih dari satu, begitu pula dengan jasa yang diperlukan, keduanya dapat ditambahkan dalam beberapa baris input. Setelah seluruh data terisi dengan lengkap, proses penyimpanan dilakukan dengan menekan tombol *save* pada sistem.

3.2.5.7 Membuat Format Surat *Quotation* & Surat Berlangganan

Format surat *quotation* dan surat berlangganan ditempatkan pada proses registrasi pelanggan sebagai hasil, di mana surat akan dibentuk secara otomatis berdasarkan data yang dimasukkan. Penyesuaian terhadap format surat dilakukan karena terdapat perbedaan standar dokumen yang diterapkan oleh unit terkait, termasuk penyesuaian isi pasal-pasal di dalamnya. *Template* surat disusun menggunakan *view* dan ditampilkan dalam halaman web, yang kemudian dapat dicetak dalam bentuk PDF oleh pengguna. Data-data seperti nama perusahaan, nama *tenant*, layanan yang dipesan, harga layanan, serta PIC ditampilkan secara dinamis berdasarkan input pengguna saat proses registrasi. Format dasar surat disediakan oleh pihak user dan digunakan sebagai contoh dalam pembuatan *template*.



Gambar 3.28 Tampilan Format Surat Kontrak Berlangganan

Surat berlangganan diisi secara otomatis menggunakan data yang telah dimasukkan pada form registrasi sebelumnya. Karena informasi yang dibutuhkan untuk surat *quotation* dan surat berlangganan memiliki kesamaan, kedua surat tersebut digabungkan dalam satu berkas PDF yang sama, namun ditampilkan pada halaman yang berbeda untuk menjaga kejelasan dan keterpisahan isi masing-masing dokumen.

3.2.5.8 Penyesuaian Form Tenant

Penyesuaian pada *form tenant* dilakukan pada bagian input *service*, *provider*, dan *capacity*, yang dibuat untuk saling bergantung satu sama lain. *Dropdown provider* hanya akan ditampilkan setelah opsi *service* dipilih, dan datanya difilter berdasarkan *service* terkait. Selanjutnya, *dropdown capacity* akan difilter berdasarkan kombinasi *service* dan *provider* yang telah dipilih. Proses pemanggilan data ini diimplementasikan menggunakan AJAX dengan Fetch API, dan mengembalikan respons dalam bentuk JSON. Data pada setiap *dropdown* diambil dari tabel yang terpisah di basis data, dan diformat secara dinamis di sisi klien. Selain itu, pada kondisi edit, nilai sebelumnya secara otomatis ditampilkan agar pengguna dapat melakukan perubahan tanpa harus mengisi ulang dari awal.

The image shows a web form for creating a tenant. It includes several dropdown menus and text input fields. The 'Provider' dropdown is currently open, displaying a list of providers: Lintasarta, Telkom, Neutiz, Icon+, and CBN. Other fields include 'Location', 'Service' (set to VPN-IP), 'Capacity' (set to Select Capacity), 'Billing Type' (set to Monthly), 'End Date' (with a date picker), and 'Tenant PIC Email'.

Gambar 3.29 Field yang Disesuaikan pada Form Tenant

Pada form ini, proses pemilihan layanan dilakukan secara berurutan, dimulai dari pemilihan jenis layanan, dilanjutkan dengan penyedia layanan / *provider*, dan diakhiri dengan pemilihan kapasitas. Urutan ini diterapkan untuk memastikan bahwa kapasitas yang tersedia memang ditawarkan oleh *provider* yang dipilih, dan *provider* tersebut memang menyediakan layanan yang telah dipilih sebelumnya.

3.2.5.9 Menyesuaikan *Form Register*

Dikarenakan terdapat beberapa penyesuaian dan penambahan *field* data, *form* registrasi juga perlu disesuaikan berdasarkan format surat dan kebutuhan *field* terbaru. *Form* ini dirancang agar perusahaan harus dipilih terlebih dahulu sebelum *field* lainnya ditampilkan. Setelah perusahaan dipilih, *field tenant* akan ditampilkan dalam bentuk *dropdown* yang berisi tenant-tenant milik perusahaan tersebut, sesuai dengan data yang telah diinput sebelumnya. Ketika tenant dipilih, detail layanan seperti lokasi, *service*, *provider*, dan *capacity* akan otomatis ditampilkan berdasarkan data sebelumnya. Nilai *quantity* akan diset secara *default* ke angka 1 namun tetap dapat disesuaikan oleh pengguna. Biaya aktivasi dan biaya berlangganan diambil secara otomatis dari database sesuai dengan kombinasi *service*, *provider*, dan kapasitas yang dipilih, namun tetap bisa disesuaikan tergantung dengan kesepakatan dengan pelanggan. *Field* tambahan berupa *note* juga ditambahkan untuk mencatat keterangan lebih lanjut.

Gambar 3.30 Tampilan Form Register

Dalam proses bisnisnya, satu registrasi perusahaan dapat mencakup lebih dari satu tenant dan layanan. Oleh karena itu, data layanan *tenant* yang dipilih akan terlebih dahulu dikumpulkan dalam bentuk *array* dan kemudian disimpan sebagai satu *field* dalam tabel *tx_register*. Seluruh proses input ini dilakukan oleh pengguna internal melalui formulir kustom yang dikembangkan di luar sistem Voyager.

#	Service	Provider	Location	Qty	Action
1	PAID TV STREAMING 1 Mbps	Intas	Jalan Nenas 1 Bandar Udara Depati Amir	1	

Save

Gambar 3.31 Tampilan Data yang Sudah di Add dan Ditampung Terlebih Dahulu

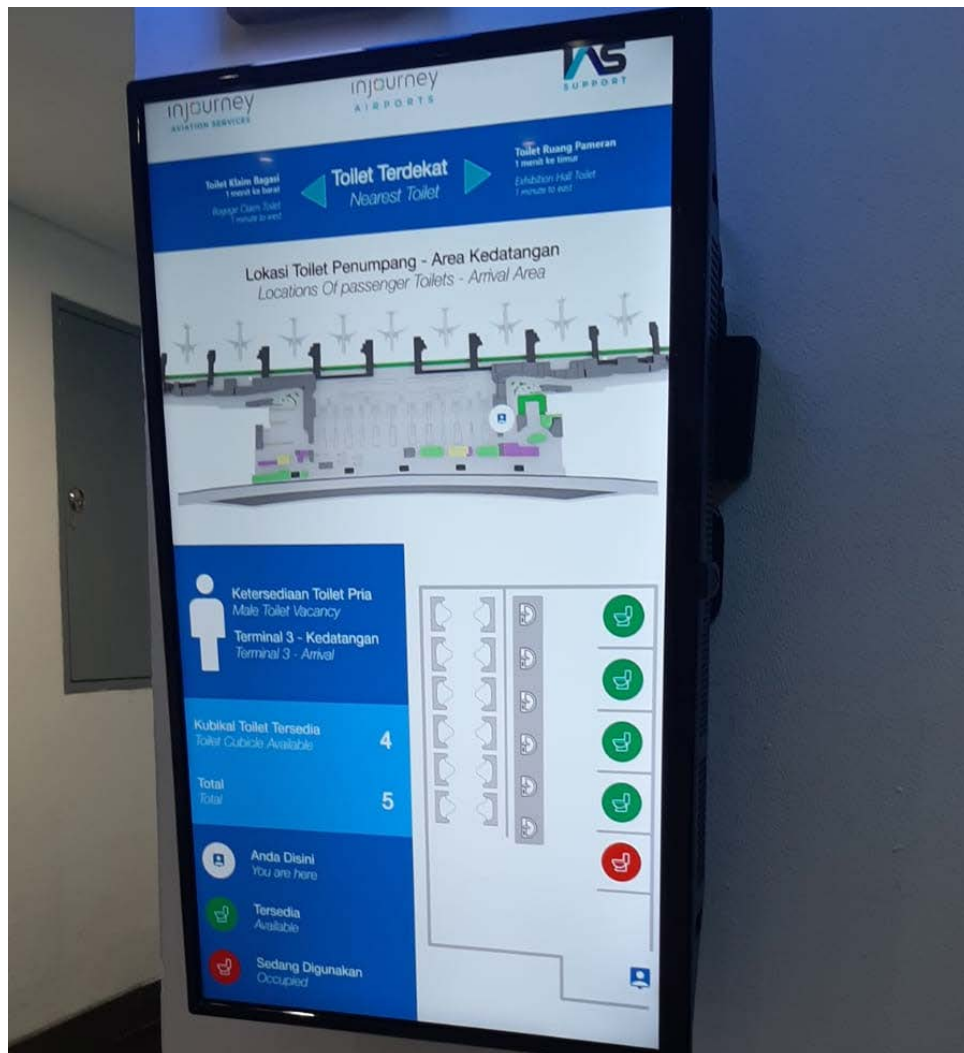
Proses registrasi diawali dengan memilih perusahaannya terlebih dahulu, kemudian akan muncul data *PIC* dari perusahaan tersebut. Setelah itu, tenant akan dipilih dari *dropdown* dan akan muncul detail dari layanan yang dipesan

oleh tenant tersebut. Pada bagian ini, jumlah layanan bisa disesuaikan beserta dengan harga berlangganan dan harga aktivasinya sesuai dengan kesepakatan yang dilakukan sebelumnya, serta catatan untuk keterangan agar memperjelas layanan atau biayanya. Setelah itu, tombol *add* ditekan dan data tersebut akan ditampung terlebih dahulu seiring menunggu layanan lain yang ingin dipesan dengan cara yang sama sekaligus dalam proses registrasi ini. Saat registrasi sudah sesuai, data baru akan dimasukkan ke basis data dengan tombol *save*.

3.2.6 Proyek Toilet Management System (Minggu Ke-20)

Toilet Management System merupakan salah satu proyek milik supervisor yang telah berjalan dan di-*deploy* untuk digunakan langsung di lapangan. Sistem ini bertujuan untuk memantau status penggunaan toilet, termasuk mengetahui apakah toilet sedang digunakan dan seberapa sering digunakan. Proyek ini dibangun menggunakan *framework* Laravel versi 9 dengan PHP versi 8 serta menggunakan MySQL sebagai basis datanya. Sistem ini terintegrasi dengan sensor yang dipasang pada pintu toilet, sehingga mampu mendeteksi secara otomatis status toilet dan menyajikan datanya dalam bentuk yang informatif.





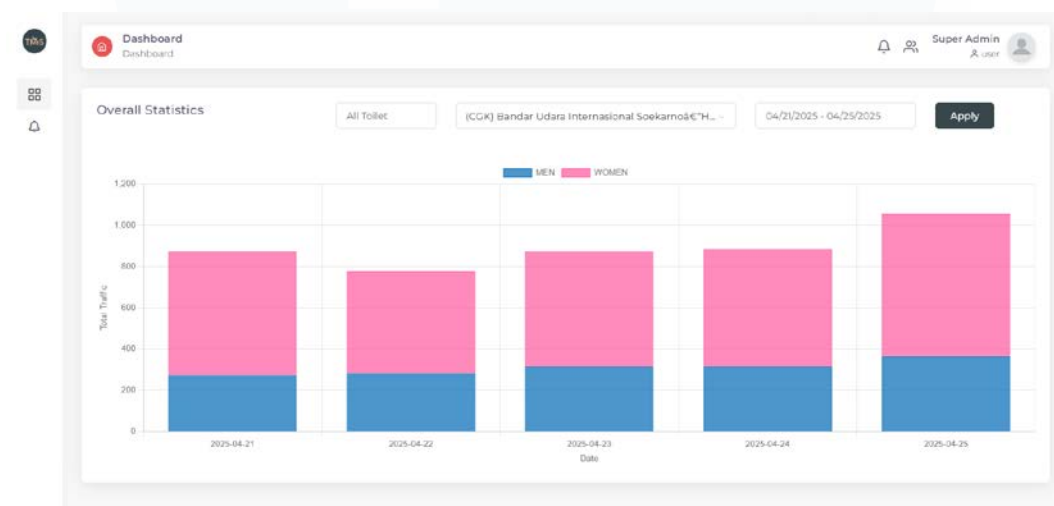
Gambar 3.32 Penggunaan Toilet Management System di Lapangan

Monitor ini dipasang di area toilet pria pada kedatangan domestik Terminal 3. Layar tersebut menampilkan informasi secara *real-time* mengenai lokasi toilet, status setiap bilik apakah sedang digunakan atau tersedia, serta pemetaan visual yang memudahkan pengguna dalam melihat kondisi toilet secara langsung. Tampilan ini bertujuan untuk meningkatkan kenyamanan dan efisiensi penggunaan fasilitas, dengan memberikan informasi yang jelas dan terkini kepada pengunjung bandara.

3.2.6.1 Membuat Grafik Dashboard

Pada proyek Toilet Management System, tugas untuk mengembangkan grafik pada *dashboard* admin yang menampilkan data frekuensi penggunaan

toilet harian diberikan kepada peserta. Grafik tersebut dirancang dalam bentuk *stacked bar chart* dan dipisahkan berdasarkan jenis kelamin pengguna, yaitu pria dan wanita. Data yang ditampilkan dapat difilter berdasarkan toilet, bandara, serta tanggal yang dipilih, sehingga memungkinkan analisis historis dilakukan dengan lebih fleksibel. Untuk proses pengambilan data, digunakan metode AJAX yang terhubung dengan database hasil salinan dari server produksi ke lingkungan lokal. Tampilan grafik dirancang secara mandiri oleh peserta tanpa panduan desain sebelumnya. Setelah proses pengembangan selesai, hasil pekerjaan ini telah ditinjau dan disetujui oleh supervisor sebagai bagian dari integrasi sistem *dashboard* yang berjalan di lapangan.



Gambar 3.33 Dashboard Grafik Traffic Toilet

Jumlah pengunjung toilet dapat di lacak dengan grafik ini dengan penanda warna biru sebagai toilet pria dan warna pink sebagai toilet wanita. Jika filter toilet dilakukan maka grafik akan menunjukkan data yang lebih detail, yaitu jumlah pengunjung per bilik. Kekurangan dari sistem ini adalah baterai dari sensor yang hanya bertahan sekitar 5 hari dan bisa lebih singkat jika frekuensinya pengunjungnya lebih banyak atau pintu toilet sering dibuka tutup. Sehingga petugas diharuskan memantau sensor agar tidak mati dan data yang tercatat tidak hilang.

3.2.7 Proyek E-Office (Minggu Ke-20 Sampai Minggu Ke-22)

Pada minggu ke-20 hingga minggu ke-22, peserta kembali diminta untuk melanjutkan proyek E-Office yang sebelumnya sempat diselesaikan. Hal ini disebabkan oleh munculnya kebutuhan tambahan dari pengguna yang baru disampaikan setelah proyek dianggap selesai. Kebutuhan-kebutuhan tersebut disampaikan dalam surat resmi dan kemudian difilter oleh supervisor, dengan mempertimbangkan bahwa tidak semua permintaan dapat direalisasikan karena keterbatasan teknis maupun bisnis prosesnya. Pengembangan lanjutan kemudian dikerjakan oleh peserta berdasarkan daftar kebutuhan yang telah disampaikan.

3.2.7.1 Mengurutkan Tembusan pada Surat

Tembusan Yth:

1. PRESIDENT DIRECTOR - PT. TMII;
2. FINANCE & HUMAN CAPITAL DIRECTOR - PT. TMII;
3. OPERATION DIRECTOR - PT. TMII;
4. VP HUMAN CAPITAL & GENERAL AFFAIR - PT. TMII;
5. MANAGER BUDGETING & COST CONTROL - PT. TMII;
6. MANAGER PARTNERSHIP - PT. TMII;
7. MANAGER REGIONAL PAVILION - PT. TMII;
8. ASMEN CSR & EXTRENAL RELATIONSHIP - PT. TMII;
9. ASMEN ACCOUNTING & VERIFICATION - PT. TMII;
10. ASMEN ASSET MANAGEMENT - PT. TMII;

Gambar 3.34 Tembusan Surat yang Sudah Berurutan

Pada sistem sebelumnya, urutan tembusan surat ditampilkan tidak sesuai dengan struktur jabatan, karena kolom *levelOrder* yang menunjukkan urutan jabatan disimpan dan diproses sebagai tipe data *string*. Akibatnya, urutan yang muncul tidak berurutan dan hanya mengikuti pola *string*, bukan urutan jabatan sebenarnya. Untuk menyelesaikan permasalahan ini, modifikasi pada model Laravel yang menangani pemanggilan data tembusan surat telah dilakukan dengan mengubah logika pengurutan agar *levelOrder* dibaca dan diurutkan sebagai *integer*. Dengan perubahan ini, daftar tembusan untuk surat sekarang ditampilkan berdasarkan jenjang jabatan secara berurutan, baik pada tampilan

antarmuka sistem maupun pada hasil cetak PDF. Perubahan ini berjalan lancar tanpa kendala, dan telah divalidasi bersama supervisor.

3.2.7.2 Menambahkan Status Disposisi

Pada tampilan sebelumnya, daftar disposisi dalam sistem E-Office tidak menampilkan apakah surat disposisi tersebut telah dibaca oleh penerimanya atau belum. Hal ini menyebabkan pengirim tidak memiliki informasi mengenai status surat yang telah dikirimkan. Untuk mengatasi permasalahan tersebut, penambaham logika penanda status baca dilakukan oleh peserta dengan memanfaatkan tabel *tx_inbox_unread*. Jika data terkait surat masih terdapat di dalam tabel tersebut, maka dianggap surat belum dibaca, dan sebaliknya, jika tidak ditemukan maka surat dianggap telah dibaca. Logika pemanggilan data terlatak pada model, sedangakn logika pewarnaan status ada dalam view. Status surat menampilkan status *read* atau *unread* dan didukung dengan perubahan warna latar belakang baris: warna kuning menandakan surat belum dibaca, dan warna hijau menandakan surat telah dibaca.

Historical Disposition

No	No Dispo	Hal	Desc. Dispo	Signature Position	From	To	Status	Date
1	DISPA.0001/HCD/00/05/2025	test 31 jan 25	poiuytrewqwertyuio[poiuytrewqwertyuipyrewq	VP Human Capital & General Affair	I Putu Mirayana VP Human Capital & General Affair	Lilik Kusdarwati Admin Corporate Secretary	Read	28 May 2025
2	DISPA.0001/HCD/00/05/2025	test 31 jan 25	poiuytrewqwertyuio[poiuytrewqwertyuipyrewq	VP Human Capital & General Affair	I Putu Mirayana VP Human Capital & General Affair	Agus Supian Asmen Asset Management	Read	28 May 2025

Close

Gambar 3.35 Tampilan Status Surat Disposisi

3.2.7.3 Menambahkan Tujuan Akhir Surat

Salah satu kebutuhan lainnya dari pengguna adalah menampilkan informasi mengenai tujuan akhir surat pada tampilan daftar surat *draft* dan *sent*. Tujuan akhir ini penting untuk memberikan kejelasan kepada pembuat maupun penerima surat terkait arah pengiriman surat tersebut, baik ke pihak internal maupun eksternal. Karena tujuan surat dapat bersifat internal maupun

eksternal, maka menyesuaikan pengambilan data pada model dilakukan agar mendukung kedua kondisi tersebut.

Draft Item June 2025 View

Show 25 entries Search:

No.	Received Date	From	Subject	Doc. Number	Doc. Date	Send Draft to	Final Destination	Action
1.	11 Jun 2025 15:58	(Draft) NOTA DINAS - Staff Testing	a	DRAFT	11 Jun 2025	SPV Budgeting & Cost Control	VP Finance & Accounting	
2.	10 Jun 2025 14:31	(Draft) URMUM - Staff Testing	test approval ext 3	DRAFT	10 Jun 2025	VP Finance & Accounting	kopi kapal air	
3.	05 Jun 2025 16:40	(Draft) MEMO - Staff Testing	test approval int 1	DRAFT	05 Jun 2025	SPV Budgeting & Cost Control	VP Human Capital & General Affair	
4.	05 Jun 2025 15:54	(Draft) URMUM - Staff Testing	test approval ext 2	DRAFT	05 Jun 2025	VP Finance & Accounting	pt makan makan	
5.	05 Jun 2025 15:17	(Draft) URMUM - Staff Testing	test approval ext 1	DRAFT	05 Jun 2025	VP Finance & Accounting	pt suka suka	

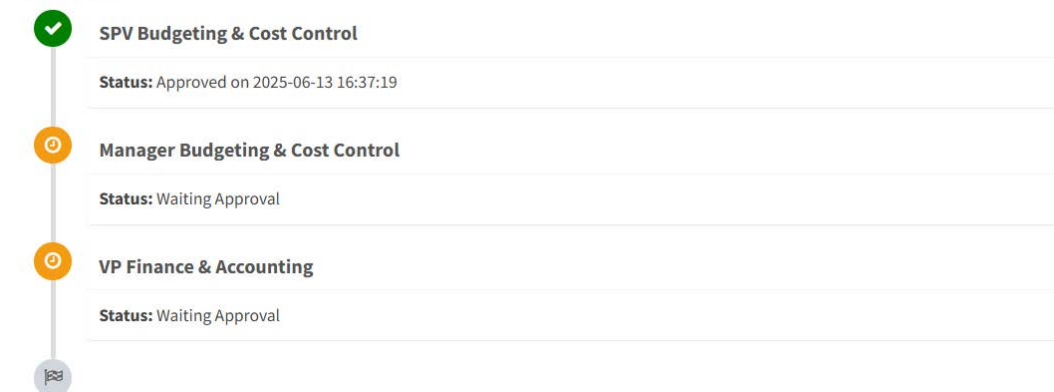
Gambar 3.36Tampilan Daftar Draft Surat

Untuk surat internal, data diambil dari tabel *tx_inbox_detail* menggunakan kolom *receiveAs* yang berisi nilai 'to_konsep', sedangkan untuk surat eksternal, data diambil dari tabel *tx_inbox_header* melalui kolom *to_institute*. Dalam penggunaannya, hanya salah satu dari dua sumber tersebut yang memiliki isi, sehingga pada controller ditambahkan logika pengecekan, jika *receiveAs* kosong maka akan menampilkan *to_institute*, dan sebaliknya. Informasi tujuan akhir ini ditampilkan dalam bentuk nama jabatan untuk surat internal dan nama perusahaan untuk surat eksternal. Selain itu, kolom ini juga ditambahkan ke dalam fitur pencarian agar pengguna dapat mencari surat berdasarkan tujuan akhirnya secara langsung dari daftar *draft* dan *sent*.

3.2.7.4 Menambahkan Fitur Approval

Sistem sebelumnya memiliki masalah terkait sistem yang tidak secara otomatis memastikan bahwa *draft* surat melewati atasan langsung pembuat surat. Hal ini menyebabkan surat dapat langsung dikirim ke atasan yang lebih tinggi tanpa melalui proses persetujuan bertingkat. Hal ini mengabaikan struktur hierarki dalam perusahaan yang merupakan komponen penting. Selain itu, sistem juga belum memiliki tampilan yang menunjukkan alur persetujuan secara menyeluruh, sehingga pengguna tidak dapat memantau sejauh mana surat telah diproses.

Approval View

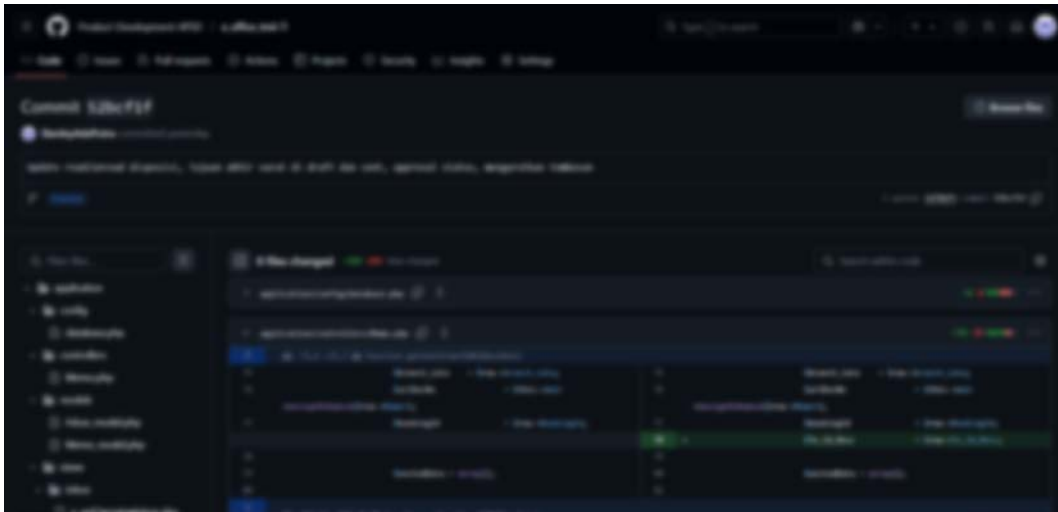


Gambar 3.37 Tampilan Urutan Approval Surat

Untuk mengatasi permasalahan ini, peserta menambahkan logika pencarian atasan langsung berdasarkan kolom yang telah tersedia pada data pengguna, yang menunjukkan *ID role* dari atasan langsung. Proses pencarian dilakukan secara iteratif menggunakan logika *looping while* pada *controller* hingga mencapai tujuan akhir surat. Dengan demikian, seluruh atasan yang terhubung langsung akan dilibatkan secara berurutan dalam proses *approval*.

Penambahan tampilan *approval* juga dilakukan dalam bentuk *timeline*, sehingga baik pembuat surat maupun para *approver* dapat melihat siapa saja yang telah atau belum menyetujui surat tersebut. Status *approval* ditandai dengan warna hijau untuk yang telah disetujui beserta keterangan waktu persetujuan, dan kuning untuk yang masih menunggu persetujuan. Selama proses pengembangan, kesulitan sempat dialami peserta dalam memahami urutan *approval* serta alur data yang masuk ke dalam basis data, namun berhasil menyelesaikannya setelah memahami struktur logika sistem secara menyeluruh.

3.2.7.5 Testing, Debugging & Code Review



Gambar 3.38 Push kode ke repository GitHub

Pada tahap akhir, sistem kembali diuji secara manual untuk memastikan alur bisnis berjalan sesuai dengan yang diharapkan serta untuk mendeteksi adanya kesalahan atau bug yang mungkin masih muncul. Pengujian dilakukan bersama supervisor, dengan memastikan bahwa seluruh data yang diproses telah sesuai dengan kebutuhan pengguna dan *output* yang dihasilkan telah akurat. Komentar-komentar penjelas juga ditambahkan ke dalam kode guna meningkatkan keterbacaan dan membantu pengembang lain dalam memahami fungsi setiap bagian kode. Setelah seluruh fungsi dinyatakan sesuai, kode melalui proses *code review* akhir oleh supervisor. Selanjutnya, kode di-*push* ke *branch* milik peserta pada repositori untuk dilakukan peninjauan lebih lanjut sebelum nantinya digabungkan ke *branch* utama.

3.3 Kendala yang Ditemukan

Selama menjalankan kegiatan magang di PT IAS Support Indonesia, peserta magang menghadapi beberapa kendala yang memengaruhi kelancaran proses kerja. Adapun kendala-kendala tersebut adalah sebagai berikut:

1. Kurangnya dokumentasi teknis dan proses bisnis. Dokumentasi yang tersedia dalam proyek sebagian besar hanya berupa komentar-komentar pada potongan kode program. Tidak terdapat dokumentasi teknis yang

menjelaskan secara menyeluruh fungsi-fungsi yang digunakan maupun struktur alur kerja sistem. Selain itu, dokumentasi mengenai proses bisnis yang mendasari pengembangan sistem juga tidak tersedia, sehingga muncul kebingungan terhadap alur kerja sistem yang dikerjakan.

2. Koneksi Wi-Fi yang tidak stabil. Meskipun kecepatan internet yang disediakan cukup memadai ketika koneksi aktif, kendala sering terjadi karena jaringan Wi-Fi di kantor mengalami gangguan. Dalam beberapa kondisi, perangkat terhubung ke jaringan Wi-Fi namun tidak dapat mengakses internet.
3. Ketidadaan sistem manajemen proyek. Dalam pelaksanaan proyek, tidak ada penggunaan sistem manajemen proyek yang terstruktur, seperti Trello, Jira, atau sejenisnya. Hal ini mengakibatkan peserta tidak memiliki acuan resmi terkait tenggat waktu penyelesaian tugas atau skala prioritas pekerjaan. Koordinasi mengenai progress dan deadline dilakukan secara lisan, sehingga alur kerja terkadang kurang terarah.
4. Penugasan yang tidak terstruktur. Tugas yang diberikan kepada peserta belum dalam kondisi siap untuk langsung dikerjakan karena ketergantungan terhadap pihak lain. Contohnya adalah proyek Tenant Management yang baru dapat dimulai setelah peserta menunggu selama kurang lebih sembilan minggu sejak awal penugasan.

3.4 Solusi atas Kendala yang Ditemukan

Untuk mengatasi berbagai kendala yang muncul selama menjalani kegiatan magang di PT IAS Support Indonesia, beberapa solusi telah diterapkan oleh peserta agar proses pembelajaran dan pengerjaan tugas tetap berjalan secara optimal. Solusi-solusi tersebut antara lain:

1. Melakukan konsultasi langsung kepada supervisor. Untuk mengatasi kurangnya dokumentasi, komunikasi dengan supervisor dilakukan oleh peserta secara aktif untuk memperoleh pemahaman yang lebih jelas terkait alur kerja sistem, kebutuhan fungsional, serta standar implementasi yang

diharapkan. Dengan demikian, miskomunikasi dan kesalahan implementasi dapat diminimalkan.

2. Menggunakan koneksi internet pribadi. Untuk mengatasi kendala jaringan internal kantor yang sering mengalami gangguan, koneksi pribadi berupa *hotspot* dari *handphone* dimanfaatkan oleh peserta. Langkah ini dilakukan agar pengembangan sistem tetap dapat dilanjutkan tanpa terhambat akses ke *repository* GitHub.
3. Menyelesaikan penugasan secepat mungkin. Dalam kondisi tanpa sistem manajemen proyek yang terstruktur, menyelesaikan setiap penugasan seefisien mungkin menjadi langkah yang diambil oleh peserta. Estimasi waktu pengerjaan dilakukan secara mandiri berdasarkan kompleksitas tugas, dan hasil pekerjaan selalu dikonsultasikan untuk memperoleh validasi dari supervisor. Pendekatan ini membantu menjaga progres proyek agar tetap berjalan.
4. Mengisi Waktu dengan Kegiatan yang Produktif. Untuk menyikapi keterlambatan dalam pemberian tugas utama, tindakan inisiatif untuk melakukan kegiatan pembelajaran mandiri dilakukan oleh peserta. Selain itu, supervisor juga memberikan beberapa tugas kecil seperti membuat *dashboard*.