

BAB III

PELAKSANAAN KERJA

3.1 Kedudukan dan Koordinasi

Bagian ini menjelaskan mengenai kedudukan peserta magang dalam struktur organisasi perusahaan serta mekanisme koordinasi yang diterapkan selama pelaksanaan proyek. Pembahasan mencakup posisi kerja peserta magang, peran dan tanggung jawab yang dijalankan, serta alur komunikasi dan koordinasi dengan supervisor lapangan dan pihak terkait lainnya dalam mendukung kelancaran proses pengembangan sistem.

3.1.1 Kedudukan

Selama menjalani program kerja magang di PT CS-ERP Indonesia, peserta magang ditempatkan pada posisi *Fullstack Web Developer*. Posisi ini memberikan lingkup kerja yang cukup luas, mencakup proses perancangan, implementasi, pengujian, hingga pemeliharaan sistem berbasis web yang dikembangkan oleh perusahaan. Penempatan pada posisi ini memberikan kesempatan bagi peserta magang untuk memahami proses pengembangan sistem secara menyeluruh, baik dari sisi frontend maupun backend.

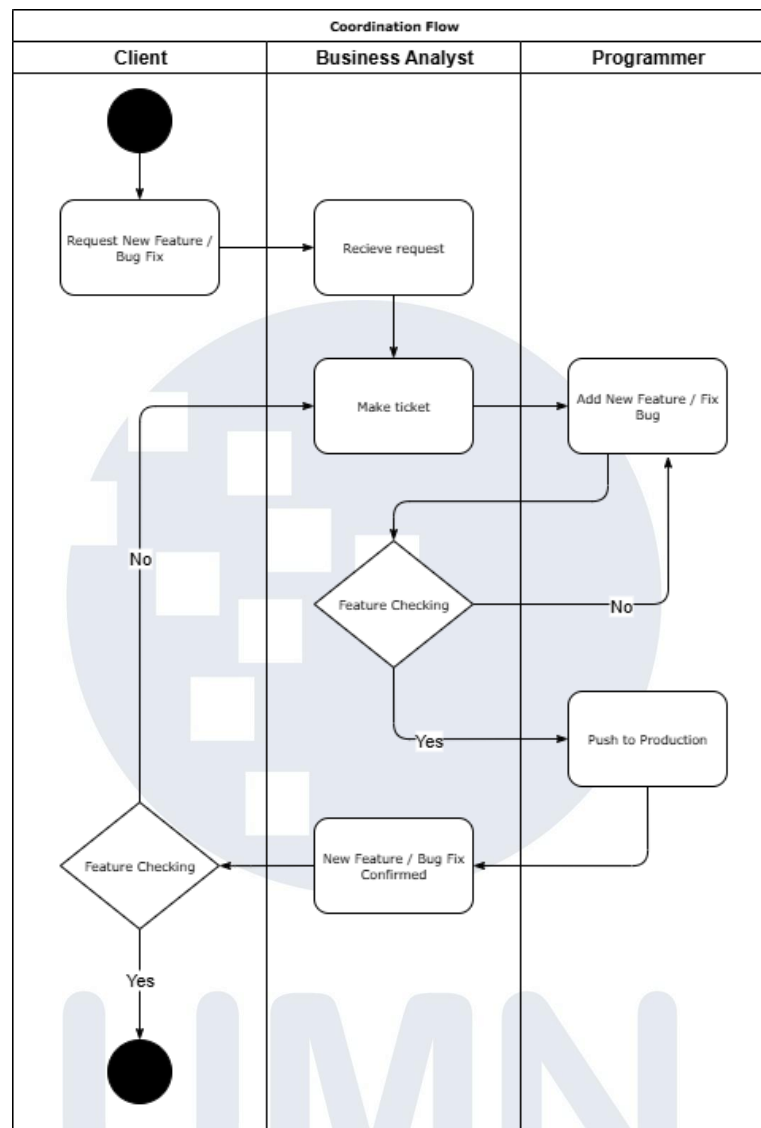
Mengacu pada struktur organisasi yang telah ditampilkan pada Gambar 2.1 pada bab sebelumnya, *programmer intern* berada di bawah supervisi langsung seorang *senior programmer*. *Senior programmer* memiliki peran utama dalam mengarahkan, mengawasi, serta memastikan bahwa setiap aktivitas pengembangan sistem berjalan sesuai dengan standar teknis dan kebutuhan perusahaan. Dalam hal ini, peserta magang berperan sebagai pendukung teknis yang membantu menyelesaikan berbagai tugas pengembangan, khususnya pada proyek-proyek yang memiliki tingkat prioritas tinggi atau belum dapat diselesaikan sepenuhnya oleh *senior programmer* akibat keterbatasan waktu dan tingginya beban kerja.

Dalam pelaksanaan tugas sehari-hari, peserta magang dituntut untuk bersikap mandiri dan proaktif dalam mempelajari teknologi, *framework*, serta alur kerja yang digunakan di perusahaan. Peserta magang juga didorong untuk memanfaatkan teknologi pendukung, termasuk *Artificial Intelligence* (AI), sebagai alat bantu dalam proses pengembangan. Meskipun demikian, untuk aspek-aspek yang memiliki tingkat kompleksitas tinggi, khususnya terkait pengelolaan dan integrasi *database* perusahaan, peserta magang secara rutin melakukan konsultasi dengan *senior programmer* serta *business analyst*. Koordinasi ini bertujuan untuk memastikan bahwa implementasi teknis yang dilakukan tetap selaras dengan struktur sistem yang ada serta kebutuhan bisnis klien.

3.1.2 Koordinasi

Untuk mendukung kelancaran pelaksanaan proyek, PT CS-ERP Indonesia menerapkan mekanisme koordinasi yang terstruktur antara pihak-pihak yang terlibat. Mekanisme ini dirancang untuk memastikan bahwa komunikasi, distribusi tugas, serta validasi hasil pekerjaan dapat berjalan secara efektif dan terkontrol. Koordinasi yang terstruktur juga berperan dalam meminimalkan miskomunikasi serta memastikan kesesuaian antara kebutuhan bisnis dan implementasi teknis yang dilakukan oleh tim pengembang.

Dalam pelaksanaannya, koordinasi dilakukan secara rutin antara klien, business analyst, dan tim programmer sesuai dengan peran dan tanggung jawab masing-masing. Melalui mekanisme ini, proses pengembangan sistem dapat dipantau dengan baik dan setiap perubahan atau kendala yang muncul dapat segera ditindaklanjuti. Alur koordinasi tersebut ditunjukkan pada Gambar 3.1.



Gambar 3.1 Alur koordinasi

Berdasarkan Gambar 3.1, alur koordinasi dalam pelaksanaan proyek diawali dari interaksi langsung antara klien dan *business analyst*. Pada tahap awal, klien menyampaikan kebutuhan, permintaan pengembangan fitur baru, maupun permasalahan yang dihadapi pada sistem melalui sesi pertemuan atau komunikasi rutin dengan business analyst. Informasi yang diperoleh dari klien kemudian dianalisis dan dituangkan ke dalam bentuk tiket, yang berisi deskripsi detail mengenai kebutuhan sistem, ruang lingkup pekerjaan, serta tingkat prioritas pengerjaan. Tiket ini selanjutnya diteruskan kepada divisi *programmer* sebagai dasar pelaksanaan pekerjaan teknis.

Setelah tiket diterima, tim *programmer* yang terdiri atas *senior programmer* dan *programmer intern* melakukan pembagian tugas secara fleksibel. Pembagian tugas ini mempertimbangkan tingkat kesibukan, ketersediaan waktu, serta kompleksitas pekerjaan yang harus diselesaikan. Dengan mekanisme tersebut, seorang *programmer* dapat menangani lebih dari satu tiket dalam satu waktu, termasuk tiket yang berasal dari klien yang berbeda. Pola kerja ini bertujuan untuk menjaga fleksibilitas perusahaan terhadap kebutuhan klien sekaligus memastikan seluruh permintaan dapat diselesaikan secara tepat waktu.

Setelah proses pengembangan atau perbaikan sistem selesai dilakukan, hasil pekerjaan diserahkan kembali kepada *business analyst* untuk dilakukan pengecekan dan validasi. *Business analyst* bertanggung jawab memastikan bahwa fitur yang dikembangkan atau perbaikan yang dilakukan telah sesuai dengan kebutuhan operasional klien serta spesifikasi yang telah disepakati sebelumnya. Apabila hasil pekerjaan telah dinyatakan sesuai dan memenuhi standar perusahaan, *senior programmer* akan melanjutkan ke tahap akhir, yaitu melakukan proses *push* ke *environment production*. Tahap koordinasi ini diakhiri dengan *business analyst* yang menyampaikan konfirmasi kepada klien terkait penyelesaian permintaan atau permasalahan yang diajukan. Dengan alur koordinasi yang terstruktur ini, proses pengembangan, debugging, dan pembaruan sistem di PT CS-ERP Indonesia dapat berjalan secara efisien, terkontrol, dan konsisten dengan kebutuhan klien.

3.2 Tugas yang Dilakukan

Selama pelaksanaan magang, peserta magang menjalankan tugas yang terbagi ke dalam beberapa tahapan kerja yang disesuaikan dengan tingkat adaptasi, kemampuan teknis, serta kebutuhan perusahaan. Pada tahap awal magang, peserta magang diberikan waktu selama kurang lebih dua minggu untuk beradaptasi dengan lingkungan kerja serta memperkuat pemahaman dasar terkait pengembangan web,

khususnya penggunaan HTML, CSS, dan JavaScript. Tahap adaptasi ini bertujuan untuk menyamakan pemahaman peserta magang terhadap standar pengembangan yang diterapkan di perusahaan sebelum terlibat langsung dalam proyek aktif.

Memasuki minggu ketiga setelah fase adaptasi, peserta magang mulai dilibatkan secara langsung dalam proses pengembangan sistem dan mendapatkan tanggung jawab untuk mengerjakan sebuah proyek baru sebagai pengembang tunggal selama kurang lebih dua bulan. Pada tahap ini, peserta magang bertanggung jawab dalam menerjemahkan kebutuhan bisnis yang telah dirumuskan melalui koordinasi dengan unit terkait ke dalam bentuk sistem yang fungsional, dengan tetap berada di bawah pengawasan *senior programmer* dan *business analyst*.

Setelah fase pengembangan utama tersebut selesai, peran peserta magang beralih ke tahap pemeliharaan (*maintenance*) terhadap sistem lain yang telah aktif digunakan oleh klien. Tugas pada tahap ini meliputi proses *debugging*, pembaruan fitur, serta penyesuaian sistem sesuai dengan perubahan kebutuhan operasional klien. Dengan demikian, tugas yang dijalankan peserta magang tidak hanya berfokus pada pengembangan sistem baru, tetapi juga mencakup pemeliharaan dan peningkatan kualitas sistem yang sudah berjalan.

3.3 Uraian Pelaksanaan Kerja

Bagian ini menjelaskan secara umum pelaksanaan kerja peserta magang, yang mencakup alur proses pekerjaan, mekanisme koordinasi, serta pola kerja yang diterapkan selama masa magang. Uraian difokuskan pada proses kerja secara menyeluruh, mulai dari tahap perencanaan, pengembangan, hingga pemeliharaan sistem, tanpa menitikberatkan pada proyek tertentu.

3.3.1 Proses Pelaksanaan

Proses pelaksanaan kerja peserta magang di PT CS Indonesia dilaksanakan secara bertahap dan terstruktur, mengikuti alur kerja pengembangan sistem yang telah ditetapkan perusahaan. Secara umum, alur

pelaksanaan kerja terdiri dari tiga tahap utama, yaitu rapat rutin dengan unit bisnis, proses pengembangan sistem, serta tahap code review dan quality improvement. Ketiga tahap ini saling berkaitan dan dilaksanakan secara berkesinambungan selama periode magang.

Tahap pertama dalam proses pelaksanaan adalah rapat rutin dengan unit bisnis, yang berlangsung secara berkala sepanjang periode magang. Rapat ini menjadi titik awal dalam setiap proses pengembangan maupun pemeliharaan sistem. Dalam rapat tersebut, *business analyst* menyampaikan kebutuhan klien yang diperoleh melalui hasil komunikasi dan diskusi langsung dengan klien. Kebutuhan yang dibahas mencakup permintaan pengembangan fitur baru, perubahan kebutuhan sistem yang sudah berjalan, serta permasalahan operasional yang dihadapi klien dalam penggunaan sistem.

Selain membahas kebutuhan, rapat rutin ini juga dimanfaatkan untuk melakukan evaluasi terhadap progres pengembangan yang sedang berlangsung. Tim *programmer* bersama *business analyst* mendiskusikan kendala teknis, meninjau pencapaian pekerjaan sebelumnya, serta menyusun rencana kerja untuk periode selanjutnya. Melalui mekanisme ini, perusahaan berupaya memastikan bahwa pengembangan sistem tetap selaras dengan kebutuhan bisnis klien dan dapat diselesaikan sesuai dengan prioritas yang telah ditentukan.

Setelah kebutuhan dan rencana kerja disepakati dalam rapat unit bisnis, proses pelaksanaan berlanjut ke tahap kedua, yaitu pengembangan sistem. Tahap ini merupakan inti dari pelaksanaan kerja peserta magang, karena pada fase inilah kebutuhan yang telah dirumuskan sebelumnya diterjemahkan ke dalam bentuk sistem yang fungsional. Proses pengembangan sistem mencakup tiga aktivitas utama, yaitu mempelajari alur kerja aplikasi dan struktur kode yang sudah ada, mengembangkan tampilan antarmuka pengguna (*user interface*), serta menyusun dan mengoptimalkan logika *backend* beserta *query SQL* aplikasi.

Pada tahap awal pengembangan sistem, peserta magang terlebih dahulu mempelajari alur kerja aplikasi dan struktur kode yang telah ada. Aktivitas ini dilakukan untuk memahami arsitektur sistem, pola pengembangan yang digunakan, serta relasi antar modul dan *database*. Pemahaman terhadap struktur kode menjadi sangat penting, terutama karena sistem yang dikembangkan maupun dipelihara memiliki kompleksitas tinggi, baik dari sisi logika bisnis maupun struktur data. Dalam tahap ini, peserta magang juga mempelajari struktur *database* perusahaan yang memiliki jumlah tabel cukup banyak dengan relasi yang saling bergantung, sehingga diperlukan pemahaman yang mendalam sebelum melakukan pengembangan lebih lanjut.

Setelah memahami alur kerja aplikasi dan struktur kode, peserta magang mulai mengerjakan pengembangan tampilan antarmuka pengguna (UI). Pengembangan UI bertujuan untuk menyediakan tampilan sistem yang informatif, mudah digunakan, dan sesuai dengan kebutuhan operasional pengguna. Pada tahap ini, peserta magang mengimplementasikan desain antarmuka yang mendukung alur bisnis, seperti penyajian data transaksi, status proses, serta action yang dapat dilakukan oleh pengguna sesuai dengan hak akses masing-masing. Proses pengembangan UI dilakukan secara terintegrasi dengan backend agar setiap interaksi pengguna dapat diproses dan ditampilkan secara real-time.

Seiring dengan pengembangan antarmuka, peserta magang juga bertanggung jawab dalam penyusunan dan optimalisasi logika *backend* serta *query* SQL aplikasi. Aktivitas ini mencakup pengolahan data, validasi input, pengaturan alur proses bisnis, serta pengambilan dan manipulasi data dari *database*. Optimalisasi *query* SQL dilakukan untuk memastikan sistem dapat berjalan secara efisien, terutama ketika menangani data dalam jumlah besar. Dengan adanya integrasi yang baik antara *frontend*, *backend*, dan *database*, sistem yang dikembangkan diharapkan dapat berfungsi secara baik dan sesuai dengan kebutuhan pengguna.

Tahap terakhir dalam proses pelaksanaan kerja adalah *code review* dan *quality improvement*. Setelah fitur atau perbaikan sistem selesai dikembangkan, kode yang dihasilkan akan melalui proses *code review*. Proses ini bertujuan untuk memastikan bahwa kode telah memenuhi standar *coding* perusahaan, baik dari sisi struktur, keterbacaan, keamanan, maupun efisiensi. Selain itu, *code review* juga berfungsi untuk mengidentifikasi potensi *bug* atau masalah lain pada sistem sebelum diimplementasikan ke lingkungan produksi.

Berdasarkan hasil *code review* dan pengujian oleh unit bisnis, peserta magang kemudian melakukan perbaikan *bug* dan *error* yang ditemukan. Aktivitas ini mencakup penyesuaian logika program, perbaikan tampilan antarmuka, serta penyempurnaan *query database* apabila diperlukan. Dengan adanya tahap *quality improvement* ini, sistem yang dikembangkan tidak hanya berfungsi sesuai kebutuhan, tetapi juga memiliki kualitas yang lebih baik, stabil, dan siap digunakan oleh klien dalam mendukung kegiatan operasional mereka.

Untuk mendukung seluruh rangkaian proses pelaksanaan kerja tersebut, mulai dari tahap koordinasi kebutuhan, pengembangan sistem, hingga *code review* dan pemeliharaan, diperlukan dukungan berbagai *tools* pengembangan yang sesuai dengan kebutuhan industri. Penggunaan *tools* ini berperan penting dalam meningkatkan efisiensi kerja, menjaga kualitas hasil pengembangan, serta memastikan proses pengelolaan proyek dan kode program dapat berjalan secara terstruktur dan terdokumentasi dengan baik. *Tools* yang digunakan dalam pengembangan dan penyelesaian proyek-proyek selama pelaksanaan kerja magang di PT CS Indonesia terdiri dari berbagai perangkat lunak pendukung yang berfungsi untuk meningkatkan efisiensi, kualitas, serta keteraturan proses pengembangan sistem. Penggunaan *tools* tersebut membantu peserta magang dalam menjalankan tugas-tugas pengembangan, pemeliharaan, hingga koordinasi proyek secara lebih terstruktur. Adapun *tools* yang digunakan antara lain sebagai berikut.

1. Cursor

Cursor merupakan *AI-powered code editor* yang dirancang untuk meningkatkan efisiensi proses pengembangan perangkat lunak melalui integrasi teknologi *Artificial Intelligence* secara langsung di dalam *workspace* pengembangan. *Cursor* berbasis pada *Visual Studio Code*, sehingga memiliki antarmuka dan ekosistem yang familiar bagi *developer*. *Tool* ini menyediakan berbagai fitur unggulan, seperti *context-aware code suggestion*, *inline code completion*, *background agent*, serta *AI-assisted refactoring*. Fitur-fitur tersebut membantu *programmer* dalam menulis kode, memperbaiki kesalahan, serta mengoptimalkan struktur program dengan lebih cepat dan akurat melalui penggunaan *prompt* yang relatif singkat.

Selain berfungsi sebagai *code editor*, *Cursor* juga dilengkapi dengan terminal internal yang memungkinkan pengguna menjalankan perintah berbasis *Command Line Interface (CLI)* secara langsung di dalam aplikasi. Terminal ini digunakan untuk berbagai keperluan, seperti manajemen repositori *Git*, menjalankan *script*, mengakses dan mempelajari struktur database, membantu proses *debugging*, serta membuat dan menjalankan *unit testing*. Dengan adanya integrasi editor, AI, dan terminal dalam satu lingkungan kerja, *Cursor* memberikan kemudahan bagi peserta magang dalam mengelola seluruh aktivitas pengembangan secara efisien.

2. XAMPP

XAMPP merupakan perangkat lunak *open-source* yang digunakan sebagai lingkungan pengembangan web berbasis server lokal (*local web server environment*). *Tool* ini menyediakan beberapa komponen utama, seperti *Apache* sebagai *web server* dan *MySQL* sebagai sistem manajemen basis data. Selama pelaksanaan magang, *XAMPP* digunakan

untuk menjalankan dan menguji aplikasi berbasis web secara lokal sebelum diimplementasikan ke lingkungan production.

Penggunaan XAMPP memungkinkan peserta magang untuk melakukan pengembangan dan pengujian sistem tanpa harus terhubung langsung dengan server eksternal milik perusahaan atau klien. Maka dari itu, proses *debugging* dan pengujian fitur dapat dilakukan secara lebih aman dan fleksibel. Selain itu, penggunaan lingkungan lokal juga membantu meminimalkan risiko kesalahan yang dapat berdampak langsung pada sistem yang sedang digunakan oleh klien.

3. MySQL Workbench dan SQL Server

MySQL Workbench dan SQL Server merupakan dua sistem manajemen basis data yang digunakan selama pelaksanaan kerja magang di PT CS Indonesia. Penggunaan dua sistem *database* ini disesuaikan dengan kebutuhan masing-masing klien, mengingat setiap klien perusahaan dapat menggunakan jenis *database* yang berbeda. MySQL Workbench digunakan untuk mengelola *database* berbasis MySQL, sedangkan SQL Server digunakan untuk mengelola database berbasis Microsoft SQL Server.

Kedua *tools* ini digunakan untuk melakukan proses *restore database* yang telah diunduh dari server klien, sehingga database tersebut dapat digunakan dalam lingkungan pengembangan lokal. Selain itu, MySQL Workbench dan SQL Server juga dimanfaatkan untuk melakukan pengelolaan struktur tabel, relasi antar data, penulisan dan pengujian *query* SQL, serta analisis data yang dibutuhkan dalam proses pengembangan dan pemeliharaan sistem. Dengan menggunakan kedua *tools* ini, peserta magang dapat memahami perbedaan karakteristik masing-masing database serta menyesuaikan implementasi sistem sesuai dengan kebutuhan klien.

4. Github

GitHub merupakan platform manajemen versi berbasis *cloud* yang digunakan untuk menyimpan, mengelola, dan mengoordinasikan pengembangan kode sumber secara kolaboratif. Selama menjalani program magang di PT CS Indonesia, GitHub digunakan sebagai *tools* utama dalam penerapan *version control* berbasis Git. Setiap perubahan kode yang dilakukan oleh *programmer*, baik oleh *senior programmer* maupun *programmer intern*, dicatat secara terstruktur melalui *commit*, sehingga riwayat perubahan dapat dilacak dengan jelas.

Penggunaan GitHub membantu meminimalkan risiko kehilangan kode serta memudahkan proses kolaborasi antar anggota tim. Selain itu, GitHub juga mendukung proses review kode melalui mekanisme *pull request*, sehingga kualitas kode yang akan diintegrasikan ke dalam sistem utama dapat terjaga. Dengan adanya sistem *version control* ini, pengembangan sistem dapat dilakukan secara lebih terorganisir dan aman, terutama ketika menangani beberapa proyek atau klien secara bersamaan.

5. Jira

Jira merupakan *tools* manajemen proyek dan manajemen tiket yang digunakan untuk mengelola kebutuhan pengembangan sistem serta permintaan perbaikan dari klien. Selama pelaksanaan magang di PT CS Indonesia, Jira berfungsi sebagai media utama untuk mencatat, memantau, dan mengelola tiket yang dibuat oleh *business analyst* berdasarkan hasil komunikasi dengan klien. Setiap tiket berisi informasi terkait deskripsi pekerjaan, jenis permintaan (pengembangan fitur baru atau perbaikan bug), tingkat prioritas, serta status pengerjaan.

Melalui Jira, tim *programmer* dapat mengetahui daftar pekerjaan yang harus diselesaikan, membagi tugas secara lebih terstruktur, serta memantau progres pengerjaan masing-masing tiket. Penggunaan Jira membantu memastikan bahwa setiap permintaan klien terdokumentasi dengan baik dan tidak terlewatkan. Selain itu, Jira juga mempermudah

proses koordinasi antara *business analyst* dan tim *programmer*, karena setiap perubahan status tiket dapat dipantau secara *real-time*. Dengan adanya *tools* manajemen tiket ini, proses pengembangan dan pemeliharaan sistem di PT CS Indonesia dapat berjalan lebih terorganisir, transparan, dan efisien.

Berbagai *tools* yang digunakan selama pelaksanaan magang di PT CS Indonesia berperan sebagai pendukung utama dalam menjalankan aktivitas pengembangan dan pemeliharaan sistem secara efektif. Setiap *tools* memiliki fungsi yang saling melengkapi, mulai dari pengembangan kode, pengelolaan *database*, pengujian sistem, hingga koordinasi dan kolaborasi antaranggota tim. Pemanfaatan *tools* tersebut tidak berdiri secara terpisah, melainkan terintegrasi dalam alur kerja harian yang dijalani oleh *programmer intern* sesuai dengan kebutuhan proyek yang sedang berlangsung.

Penggunaan *tools* tersebut dapat dilihat secara langsung dalam aktivitas kerja yang dilakukan selama periode magang. Peserta magang tidak hanya dituntut untuk memahami fungsi teknis dari setiap *tools*, tetapi juga mampu menerapkannya secara tepat pada setiap tahapan pekerjaan, mulai dari tahap koordinasi awal, proses pengembangan sistem, hingga evaluasi dan perbaikan kualitas aplikasi. Oleh karena itu, untuk memberikan gambaran yang sistematis mengenai bagaimana aktivitas kerja tersebut direalisasikan selama masa magang, diperlukan pemetaan kegiatan dalam bentuk *timeline* pelaksanaan kerja. *Timeline* pada Tabel 3.1 ini disusun untuk menunjukkan rangkaian aktivitas utama yang dijalankan oleh *programmer intern* beserta durasi pelaksanaannya selama periode magang.

Tabel 3.1 *Timeline Realisasi Agenda Programmer Intern*

No	Aktivitas	Minggu ke-	Tanggal Mulai	Tanggal Selesai
1	Rapat Rutin dengan Unit Bisnis	1 Juli – 1 Desember	1 Juli 2025	2 Desember 2025
1.1	Mendiskusikan kebutuhan dan perubahan requirement sistem	1 Juli – 1 Desember	1 Juli 2025	2 Desember 2025
1.2	Melakukan evaluasi progres pengembangan dan penyusunan rencana kerja selanjutnya	1 Juli – 1 Desember	1 Juli 2025	2 Desember 2025
2	Mengembangkan sistem (Frontend, Backend, SQL Query)	3 Juni – 3 Desember	15 Juni 2025	18 Desember 2025
2.1	Mempelajari alur kerja aplikasi dan struktur <i>code</i> yang sudah ada	3 Juni – 1 November	16 Juni 2025	1 November 2025
2.2	Mengembangkan tampilan antarmuka pengguna (UI)	1 Juli – 2 Desember	1 Juli 2025	11 Desember 2025
2.3	Menyusun dan mengoptimalkan logika <i>backend</i> serta <i>query SQL</i> aplikasi	4 Juli – 3 Desember	21 Juli 2025	18 Desember 2025
3	Melakukan Code Review dan Quality Improvement	4 Agustus – 3 Desember	18 Agustus 2025	18 Desember 2025
3.1	Melakukan <i>code review</i> untuk memastikan standar coding yang konsisten	4 Agustus – 3 Desember	18 Agustus 2025	18 Desember 2025
3.2	Memperbaiki <i>bug</i> dan <i>error</i> berdasarkan hasil pengujian unit bisnis	4 Agustus – 3 Desember	18 Agustus 2025	18 Desember 2025

Berdasarkan *timeline* realisasi agenda *programmer intern* pada Tabel 3.1, dapat diketahui bahwa pelaksanaan magang di PT CS Indonesia berlangsung secara berkelanjutan dan terstruktur. Aktivitas yang dilakukan tidak hanya berfokus pada satu jenis pekerjaan, melainkan mencakup berbagai tahapan penting dalam siklus pengembangan sistem, seperti koordinasi dengan unit bisnis, proses pengembangan *frontend* dan *backend*, pengelolaan *database*, hingga kegiatan evaluasi dan peningkatan kualitas kode. Setiap aktivitas tersebut saling berkaitan dan membentuk

alur kerja yang utuh dalam mendukung keberhasilan proyek yang dijalankan perusahaan.

Selain itu, pembagian aktivitas ke dalam beberapa agenda utama menunjukkan bahwa peran *programmer intern* tidak hanya bersifat teknis, tetapi juga melibatkan pemahaman terhadap proses bisnis dan kebutuhan klien. Melalui keterlibatan langsung dalam setiap tahapan tersebut, peserta magang memperoleh pengalaman praktis mengenai bagaimana sebuah sistem dikembangkan, diuji, dan disempurnakan di lingkungan industri. Untuk memberikan penjelasan yang lebih mendalam dan terperinci, pada subbab berikutnya akan diuraikan pelaksanaan masing-masing aktivitas utama sesuai dengan urutan yang tercantum dalam timeline, dimulai dari kegiatan rapat rutin dengan unit bisnis sebagai tahapan awal dalam proses pengembangan sistem.

3.3.1.1 Mengadakan Rapat Rutin dengan Unit Bisnis

Seluruh proyek yang dijalankan dalam perusahaan diawali melalui proses rapat rutin antara tim programmer dengan unit bisnis. Rapat ini dilaksanakan sebagai wadah koordinasi dan diskusi awal terkait kebutuhan klien, ruang lingkup pekerjaan, serta rencana pengembangan proyek baru yang akan dilaksanakan. Melalui rapat ini, perusahaan berupaya memastikan bahwa setiap sistem yang dikembangkan memiliki dasar kebutuhan yang jelas dan selaras dengan tujuan operasional perusahaan maupun ekspektasi klien. Tahapan ini menjadi fondasi penting dalam proses pengembangan sistem karena berperan dalam meminimalkan kesalahpahaman, mengurangi risiko perubahan kebutuhan di tengah proses pengembangan, serta meningkatkan efektivitas perencanaan proyek.

Rapat rutin ini diadakan secara berkala pada setiap awal bulan, dengan unit bisnis berperan menyampaikan berbagai kebutuhan klien yang diperoleh dari hasil komunikasi dan pertemuan langsung dengan

klien. Informasi yang disampaikan mencakup kebutuhan fungsional sistem, permintaan penyesuaian fitur, serta kendala operasional yang dihadapi klien pada sistem yang telah berjalan. Dalam forum ini, tim programmer turut berperan aktif dengan memberikan pandangan teknis terkait tingkat kompleksitas, kelayakan implementasi, serta estimasi waktu dan sumber daya yang dibutuhkan untuk merealisasikan kebutuhan tersebut. Apabila terdapat permintaan yang dinilai tidak realistis atau sulit untuk diimplementasikan dalam kondisi tertentu, maka tim programmer dan unit bisnis akan melakukan diskusi lebih lanjut untuk mencari solusi alternatif yang tetap mampu memenuhi kebutuhan utama klien tanpa mengorbankan kualitas sistem.

Selain membahas kebutuhan dan rencana pengembangan proyek baru, rapat rutin ini juga dimanfaatkan sebagai sarana evaluasi terhadap sistem-sistem yang telah dikembangkan sebelumnya. Evaluasi dilakukan untuk menilai apakah sistem yang telah diimplementasikan berjalan sesuai dengan spesifikasi, memenuhi standar operasional perusahaan, serta memberikan manfaat yang optimal bagi klien. Unit bisnis akan menyampaikan masukan, kendala, maupun keluhan yang diterima dari klien terkait penggunaan sistem tersebut. Selanjutnya, tim programmer akan melakukan analisis lebih lanjut untuk menentukan langkah perbaikan, pengoptimalan, atau pengembangan lanjutan yang diperlukan. Hal ini bertujuan agar perusahaan dapat menjaga kualitas sistem yang dikembangkan sekaligus memastikan keberlanjutan dan layanan kepada klien.

3.3.1.2 Mengembangkan Sistem

Setelah fase koordinasi dan perumusan kebutuhan melalui rapat rutin dengan unit bisnis, proses pengerjaan proyek memasuki fase 2,

yaitu pengembangan sistem. Fase ini merupakan inti dari pelaksanaan kerja peserta magang, karena pada tahap inilah seluruh kebutuhan bisnis, hasil diskusi, serta kesepakatan teknis yang telah ditetapkan sebelumnya diterjemahkan ke dalam bentuk sistem yang dapat digunakan secara nyata oleh klien. Proses pengembangan tidak hanya berfokus pada pembuatan fitur, tetapi juga mencakup pemahaman sistem yang akan dikembangkan, penerapan standar pengembangan yang berlaku di perusahaan, serta pengujian awal untuk memastikan sistem berjalan sesuai dengan kebutuhan operasional.

Sesuai dengan timeline pada Tabel 3.1, fase pengembangan sistem mencakup tiga aktivitas utama, yaitu mempelajari alur kerja aplikasi dan struktur kode yang sudah ada, mengembangkan tampilan antarmuka pengguna (UI) serta integrasi dengan backend, dan menyusun serta mengoptimalkan logika backend beserta query SQL aplikasi. Ketiga aktivitas tersebut dilakukan secara berkesinambungan dan saling berkaitan, sehingga setiap tahapan menjadi dasar bagi tahapan berikutnya.

a. Mempelajari Alur Kerja Aplikasi dan Struktur Kode

Tahap awal dalam fase pengembangan sistem diawali dengan proses mempelajari alur kerja aplikasi dan struktur kode yang telah ada. Kegiatan ini menjadi langkah fundamental bagi peserta magang agar dapat memahami cara kerja sistem secara menyeluruh sebelum terlibat langsung dalam pengembangan fitur baru maupun perbaikan sistem yang sudah berjalan. Proses pemahaman ini dilakukan dengan menelusuri struktur *folder* dan *file* pada aplikasi berbasis *framework* Laravel, mempelajari penerapan konsep *Model–View–Controller* (MVC), serta memahami alur request dan response antara *frontend*, *backend*, dan *database*.

Selain memahami struktur teknis, peserta magang juga mempelajari bagaimana sistem digunakan dalam konteks operasional

klien. Hal ini dilakukan dengan menelusuri alur bisnis yang direpresentasikan dalam sistem, seperti proses pengelolaan data, alur persetujuan, serta hubungan antar modul. Dengan memahami keterkaitan tersebut, peserta magang dapat melihat bagaimana kebutuhan bisnis diterjemahkan ke dalam logika sistem, sehingga pengembangan yang dilakukan tidak hanya bersifat teknis, tetapi juga relevan dengan proses bisnis yang berjalan.

Pada tahap ini, peserta magang juga mempelajari standar penulisan kode dan konvensi pengembangan yang diterapkan oleh perusahaan, seperti pola penamaan variabel dan fungsi, pemisahan logika bisnis, serta penggunaan komponen atau helper tertentu. Pemahaman terhadap standar ini bertujuan untuk menjaga konsistensi kode, mempermudah proses pemeliharaan sistem, serta mengurangi potensi kesalahan integrasi ketika fitur baru dikembangkan. Tahap pemahaman alur kerja dan struktur kode ini menjadi landasan penting sebelum melanjutkan ke tahap pengembangan sistem yang lebih kompleks.

b. Mengembangkan Tampilan Antarmuka Pengguna (UI)

Setelah proses pemahaman alur kerja aplikasi dan struktur kode selesai dilakukan, peserta magang mulai memasuki tahap pengembangan *user interface* (UI). Tahap ini bertujuan untuk mengimplementasikan kebutuhan bisnis yang telah dirumuskan pada fase rapat dengan unit bisnis ke dalam bentuk tampilan sistem yang fungsional, informatif, dan mudah digunakan oleh pengguna.

Dalam pengembangan UI, peserta magang terlibat dalam pembuatan dan penyesuaian berbagai komponen antarmuka, seperti tabel data, *form input*, tombol *action*, *modal dialog*, serta elemen visual pendukung lainnya. Pengembangan dilakukan dengan mempertimbangkan alur kerja pengguna (*user flow*), sehingga setiap fitur dapat diakses secara logis dan efisien sesuai dengan kebutuhan

operasional. Selain itu, konsistensi desain antar halaman juga menjadi perhatian utama agar sistem memiliki tampilan yang seragam dan mudah dipahami oleh pengguna.

Tahap ini tidak hanya berfokus pada aspek visual, tetapi juga mencakup integrasi antara frontend dan backend. Peserta magang memastikan bahwa setiap elemen antarmuka dapat berkomunikasi dengan backend melalui pemanggilan API *Endpoint*, serta mampu menampilkan data yang diambil dari *database* secara akurat *dan real-time*. Dengan demikian, pengembangan UI menjadi jembatan antara kebutuhan pengguna dan logika sistem yang berjalan di sisi server.

c. Menyusun dan Mengoptimalkan Logika Backend serta Query SQL

Seiring dengan pengembangan antarmuka pengguna, peserta magang juga terlibat dalam penyusunan dan pengoptimalan logika *backend* serta *query SQL* aplikasi. Tahap ini berperan penting dalam memastikan bahwa setiap fungsi sistem dapat berjalan sesuai dengan kebutuhan bisnis, stabil, dan memiliki performa yang baik ketika digunakan dalam skala operasional.

Pada tahap ini, peserta magang berkontribusi dalam pengembangan *controller* untuk mengelola alur proses sistem, penyusunan validasi data untuk menjaga integritas input pengguna, serta implementasi logika bisnis yang sesuai dengan aturan perusahaan. Selain itu, peserta magang juga terlibat dalam penulisan dan penyesuaian *query SQL* yang digunakan untuk pengambilan, penyimpanan, serta pembaruan data dalam database.

Optimalisasi *query* menjadi salah satu fokus utama pada tahap ini, terutama untuk memastikan sistem tetap responsif ketika menangani data dalam jumlah besar. Beberapa upaya yang dilakukan meliputi pembatasan data melalui pagination, pengelolaan relasi antar

tabel secara efisien, serta penerapan *filtering* data sesuai kebutuhan pengguna. Hal ini diharapkan agar sistem tidak hanya berfungsi dengan baik secara fungsional, tetapi juga mampu memberikan performa yang optimal dan stabil dalam jangka panjang.

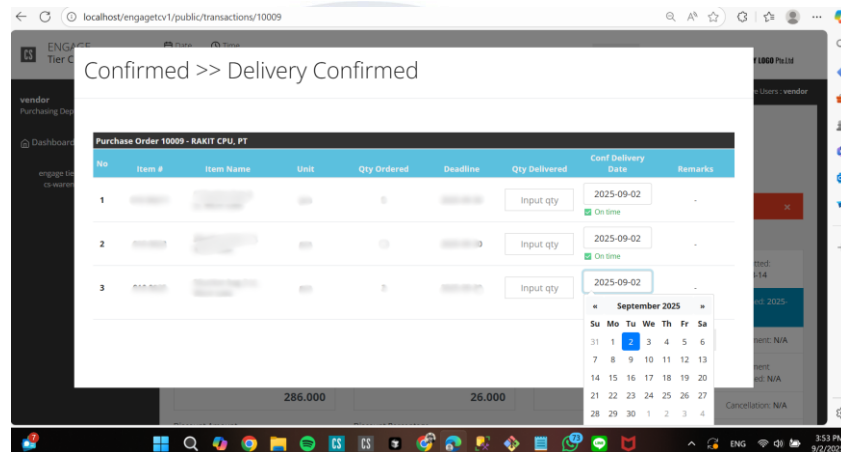
d. Proyek yang Dikembangkan

Setelah proses penyusunan dan optimalisasi logika *backend* serta *query* SQL dilakukan, hasil pengembangan tersebut kemudian diimplementasikan secara langsung dalam berbagai proyek sistem yang dikembangkan selama fase ini. Implementasi proyek ini menjadi bentuk nyata dari penerapan kebutuhan bisnis yang telah dianalisis sebelumnya, sekaligus sarana bagi peserta magang untuk mengintegrasikan aspek frontend, backend, serta pengelolaan data dalam satu sistem yang utuh dan dapat digunakan secara operasional oleh klien.

Salah satu sistem utama yang dikembangkan pada fase ini adalah Engagetcv, yaitu aplikasi web berbasis *framework* Laravel yang berfungsi sebagai sistem manajemen dan monitoring purchase order (PO) serta data vendor. Engagetcv dikembangkan sebagai subsistem pendukung dari sistem ERP utama perusahaan dan berperan dalam memastikan proses procurement berjalan secara transparan, terdokumentasi, dan terkontrol. Sistem ini tidak digunakan untuk membuat PO baru, melainkan memproses dan memantau data PO yang diimpor dari ERP untuk kemudian dikelola lebih lanjut oleh klien dan vendor.

Dalam pengembangan antarmuka pengguna, peserta magang terlibat dalam pembuatan tampilan yang bersifat informatif dan operasional, sehingga pengguna dapat dengan mudah memahami status transaksi serta melakukan tindakan yang diperlukan sesuai

dengan peran masing-masing. Sistem Engagetcv menerapkan konsep role-based access control (RBAC), di mana hak akses dan fitur yang tersedia disesuaikan dengan jenis pengguna, yaitu klien dan vendor. Berikut ini Gambar 3.2, Gambar 3.3, dan Gambar 3.4 yang menggambarkan beberapa fitur yang dimiliki oleh Engagetcv.



Gambar 3.2 Proses Pengiriman Barang

Gambar 3.2 menampilkan sebuah *modal dialog* yang digunakan oleh pihak vendor untuk melakukan konfirmasi pengiriman barang. Modal ini muncul ketika vendor memilih aksi pengiriman pada transaksi *purchase order* tertentu. Pada tampilan ini, vendor diwajibkan mengisi beberapa informasi penting, seperti tanggal pengiriman barang dan jumlah barang yang dikirim. Data yang diinput pada tahap ini akan diproses oleh sistem dan disimpan ke dalam *database* sebagai bagian dari histori transaksi. Fitur ini berperan penting dalam mendukung proses pelacakan distribusi barang secara *real-time*, sehingga klien dapat mengetahui status pengiriman tanpa harus melakukan komunikasi manual dengan vendor. Selanjutnya fitur pemantauan transaksi yang sedang berjalan dapat terlihat pada Gambar 3.3.

Gambar 3.4 Detail Transaksi

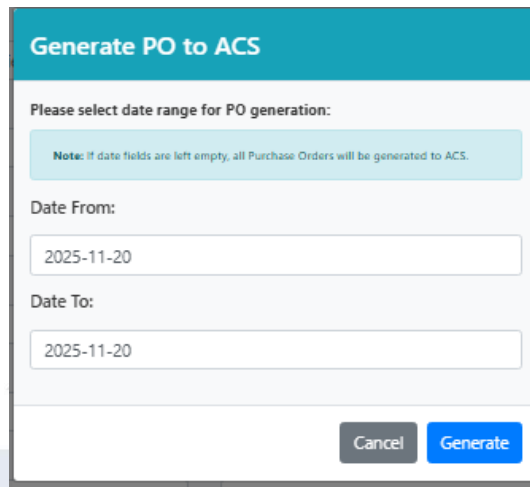
Gambar 3.4 menunjukkan halaman detail transaksi yang dapat diakses melalui tombol *View* pada tabel transaksi. Halaman ini menyajikan informasi transaksi secara lebih lengkap dan rinci, mulai dari data umum PO, informasi vendor, daftar item yang dipesan, status tiap tahapan proses, hingga histori aktivitas yang telah dilakukan. Dengan adanya halaman detail ini, pengguna dapat melakukan penelusuran transaksi secara menyeluruh, sehingga memudahkan proses evaluasi, validasi, maupun tindak lanjut apabila terjadi kendala dalam proses pengadaan barang.

Selain pengembangan sistem Engagetcv, peserta magang juga terlibat dalam pengembangan sistem Arrival Checksheet (ACS) yang digunakan oleh klien untuk mengelola proses kedatangan barang dari supplier. Sistem ini berfungsi sebagai alat kontrol untuk memastikan bahwa barang yang diterima telah sesuai dengan dokumen pengiriman dan standar pemeriksaan perusahaan. Melalui ACS, setiap aktivitas yang berkaitan dengan kedatangan barang, seperti pencatatan data awal, validasi informasi pengiriman, dan hasil pemeriksaan material, dapat dipantau secara jelas dan sistematis. Sistem ini membantu memastikan bahwa setiap barang yang masuk telah melalui proses verifikasi yang sesuai standar perusahaan. Tampilan utama pada halaman ACS dapat dilihat seperti Gambar 3.5.

Gambar 3.5 Arrival Checksheet

Gambar 3.5 menampilkan tampilan utama modul Arrival Checksheet yang berisi daftar data kedatangan barang. Pada halaman ini, pengguna dapat melihat informasi terkait supplier, nomor PO, status pemeriksaan, serta melakukan berbagai aksi lanjutan melalui tombol yang tersedia. Modul ini dirancang untuk mendukung alur kerja pemeriksaan barang secara terstruktur, mulai dari pencatatan kedatangan, pengecekan kondisi fisik material, hingga penentuan status barang apakah dapat diterima, ditolak, atau memerlukan pengujian tambahan. Dengan adanya sistem ini, seluruh proses penerimaan barang terdokumentasi dengan baik dan dapat ditelusuri kembali apabila diperlukan.

Sebagai pengembangan lanjutan dari sistem ACS, dilakukan pula peningkatan pada fitur Generate Purchase Order ke Arrival Checksheet agar lebih fleksibel dan sesuai dengan kebutuhan operasional pengguna. Sebelumnya, proses generate hanya dapat dilakukan untuk seluruh PO sekaligus, sehingga kurang sesuai ketika pengguna hanya membutuhkan data pada periode tertentu yang dapat dilihat pada Gambar 3.6.



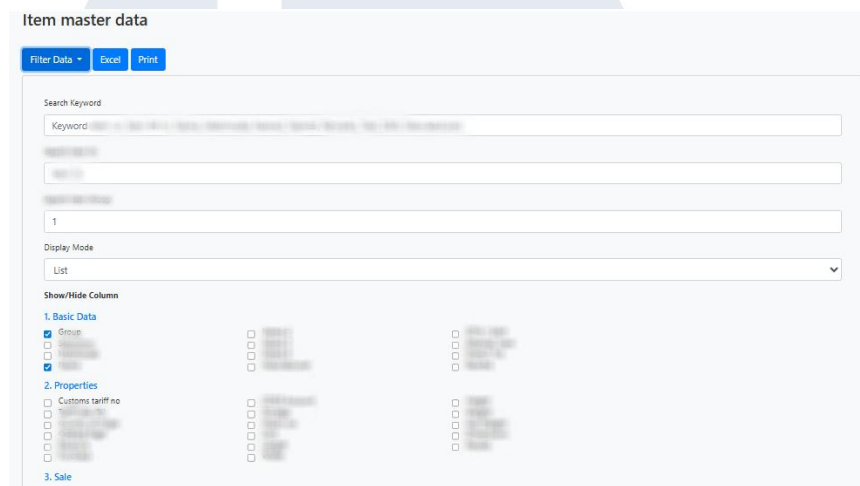
Gambar 3.6 Date Filter pada Button Generate PO to ACS

Melalui pembaruan ini sesuai dengan Gambar 3.6, sistem kini memungkinkan pengguna melakukan generate PO berdasarkan rentang tanggal yang dapat diatur melalui modal dialog. Selain itu, opsi untuk tetap generate seluruh PO tetap tersedia. Sistem juga dilengkapi validasi dasar untuk memastikan input tanggal benar dan sesuai.

Perubahan ini membantu meningkatkan efisiensi kerja karena pengguna dapat menghasilkan data PO yang lebih relevan dengan kebutuhan operasional, sekaligus mengurangi risiko kesalahan melalui mekanisme pengecekan sederhana sebelum proses dijalankan. Pembaruan ini juga memperkuat aspek kontrol dan dokumentasi dalam sistem Arrival Checksheet.

Selain pengembangan sistem yang berkaitan langsung dengan proses purchase order dan penerimaan barang, peserta magang juga terlibat dalam pengembangan dan penyempurnaan modul Item Master Data. Modul ini berfungsi sebagai pusat pengelolaan data master item dalam sistem *Warenwirtschaft* dan menjadi fondasi penting bagi berbagai proses operasional perusahaan, seperti *procurement*, pergudangan, *quality control*, dan distribusi barang.

Modul Item Master Data menyediakan tampilan antarmuka yang membantu pengguna untuk melihat, mencari, serta mengelola informasi produk secara menyeluruh dan terstruktur. Keberadaan modul ini sangat krusial karena data item yang tersimpan akan digunakan oleh berbagai modul lain dalam sistem, sehingga konsistensi dan akurasi data harus selalu terjaga. Tampilan utama modul Item Master Data dapat dilihat pada Gambar 3.7 dan Gambar 3.8.



Gambar 3.7 Item Master Data 1

Gambar 3.7 menampilkan tampilan utama daftar item dalam sistem. Pada halaman ini, pengguna dapat melihat data item dalam bentuk tabel yang memuat informasi dasar seperti nama produk, nomor item, grup item, serta atribut utama lainnya yang digunakan dalam proses operasional perusahaan. Tabel ini dirancang untuk memberikan gambaran umum mengenai seluruh item yang terdaftar dalam sistem sehingga pengguna dapat dengan cepat memahami kondisi data yang tersedia.

The screenshot displays a web-based interface for managing item master data. It features a form with three main sections: '3. Sale', '4. Purchase', and '5. User Defined Fields'. Each section contains a 'Legend' checkbox and a grid of input fields. A 'Display' button is located at the bottom right of the form. Below the form, a pagination bar shows 'Total Records : 585 Record(s)' and navigation links for 'First', 'Previous', '1', '2', '3', 'Next', and 'Last'. The main data is presented in a table with the following columns: Item Nr, Group, Name, Legend, Purchase Legend, User Defined Fields 1, and Action. The table contains three rows of data, each with a corresponding 'Action' button.

Item Nr	Group	Name	Legend	Purchase Legend	User Defined Fields 1	Action
1000000000	00	Item 1000000000			Item 1000000000	[Action]
1000000000	00	Item 1000000000			Item 1000000000	[Action]
1000000000	00	Item 1000000000			Item 1000000000	[Action]

Gambar 3.8 Item Master Data 2

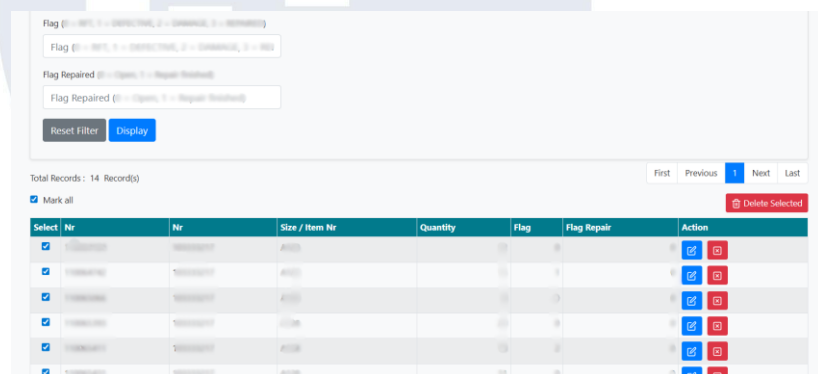
Gambar 3.8 menampilkan tampilan lanjutan dari modul Item Master Data yang menunjukkan detail serta fitur pendukung pengelolaan data. Pada halaman ini, sistem menyediakan fitur pencarian dan filter yang memungkinkan pengguna melakukan pencarian item berdasarkan kata kunci tertentu, nomor item, maupun klasifikasi grup item. Fitur ini sangat membantu terutama ketika jumlah data item yang dikelola cukup besar, karena pengguna tidak perlu menelusuri data secara manual satu per satu.

Untuk menjaga performa sistem tetap optimal meskipun menangani data dalam jumlah besar, modul Item Master Data juga dilengkapi dengan mekanisme pagination. Dengan adanya pagination, data ditampilkan secara bertahap per halaman sehingga waktu pemuatan halaman tetap cepat dan pengalaman pengguna tetap nyaman. Pendekatan ini memastikan bahwa pengelolaan data master dapat dilakukan secara efisien tanpa mengorbankan kinerja sistem.

Seiring dengan semakin banyaknya modul yang dikembangkan dan bertambahnya volume data yang dikelola dalam sistem, muncul kebutuhan untuk meningkatkan efisiensi dalam pengelolaan data operasional. Tidak hanya terbatas pada penyajian dan pencarian data, sistem juga dituntut mampu mendukung proses pemeliharaan data

secara cepat dan aman. Berdasarkan kebutuhan tersebut, dilakukan pengembangan fitur tambahan yang berfokus pada efisiensi pengelolaan data dalam jumlah besar, salah satunya melalui implementasi fitur batch delete data.

Melalui fitur batch delete, pengguna dapat memilih beberapa baris data pada tabel dengan menggunakan *checkbox*, atau memilih seluruh data dalam satu halaman melalui opsi *Mark All*. Setelah data yang ingin dihapus ditentukan, pengguna dapat menekan tombol *Delete Selected* untuk memulai proses penghapusan, seperti yang ditunjukkan pada Gambar 3.9.



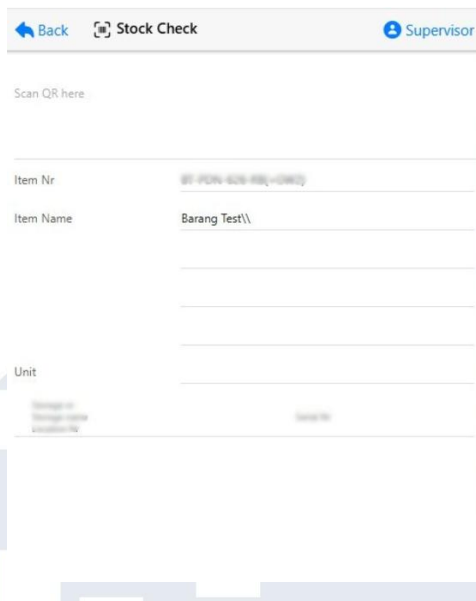
Gambar 3.9 Fitur Batch Delete

Gambar 3.9 menampilkan implementasi fitur *batch delete* pada salah satu modul sistem. Sebelum proses penghapusan dijalankan, sistem akan menampilkan dialog konfirmasi yang berisi informasi mengenai jumlah data yang akan dihapus. Dialog ini berfungsi sebagai mekanisme pengaman agar pengguna dapat memastikan kembali tindakannya dan meminimalkan risiko penghapusan data yang tidak diinginkan. Setelah dikonfirmasi, sistem akan menghapus seluruh data terpilih dalam satu proses, memperbarui tampilan tabel secara otomatis, serta menampilkan notifikasi bahwa proses penghapusan telah berhasil dilakukan.

Pada awalnya, fitur *batch delete* hanya diterapkan pada modul *quality control* (QC) barang manufaktur untuk mempercepat proses pembersihan data hasil pemeriksaan. Namun, karena fitur ini terbukti meningkatkan efisiensi kerja secara drastis, penerapannya kemudian diperluas ke berbagai modul lain yang memiliki kebutuhan penghapusan data dalam jumlah besar. Maka dari itu, fitur ini telah menjadi salah satu komponen pendukung penting dalam menjaga efektivitas dan kecepatan operasional sistem secara keseluruhan.

Selain peningkatan efisiensi dalam pengelolaan data, pengembangan sistem juga diarahkan untuk mendukung keakuratan dan keandalan pertukaran data antar modul serta integrasi dengan perangkat eksternal. Dalam operasional gudang dan inventori, kebutuhan akan pemrosesan data yang cepat dan minim kesalahan menjadi sangat penting, terutama pada proses identifikasi barang. Oleh karena itu, dikembangkan pula mekanisme *encoding* dan *decoding* data yang diimplementasikan pada fitur *Stock Check*.

Pada fitur *Stock Check*, pengguna dapat memasukkan data melalui field *Scan QR here*. Setelah data dimasukkan atau *QR code* dipindai, sistem akan menampilkan informasi dasar item serta daftar lokasi penyimpanan yang memiliki stok tersedia. Seluruh hasil pengecekan disajikan dalam bentuk tabel yang memuat informasi penting seperti nomor item, lokasi penyimpanan, dan jumlah stok yang tersedia. Tampilan fitur ini dapat dilihat pada Gambar 3.10.



Gambar 3.10 Encoding dan Decoding

Gambar 3.10 menunjukkan implementasi proses encoding dan decoding yang digunakan dalam fitur Stock Check. Ketika QR code dipindai, sistem akan melakukan proses decoding untuk mengekstrak informasi penting seperti nomor item dan nomor serial sesuai dengan format data yang telah ditentukan. Di sisi lain, proses encoding item number digunakan untuk memastikan bahwa data yang mengandung karakter khusus dapat dikirim dan diproses oleh API tanpa menyebabkan error atau kegagalan sistem.

Penerapan mekanisme encoding dan decoding ini memastikan bahwa proses pengecekan stok dapat berjalan dengan lancar dan konsisten, baik melalui pemindaian QR code maupun input manual. Dengan demikian, sistem mampu memberikan informasi stok secara cepat, akurat, dan dapat diandalkan, yang sangat mendukung kebutuhan operasional perusahaan dalam pengelolaan inventori.

3.3.1.3 Melakukan *Code Review* dan *Quality Improvement*

Setelah seluruh proses pengembangan sistem pada fase sebelumnya dilaksanakan, kegiatan magang dilanjutkan ke fase 3, yaitu *code review* dan *quality improvement*. Fase ini merupakan

tahapan penting dalam siklus pengembangan perangkat lunak karena berfungsi untuk memastikan bahwa sistem yang telah dikembangkan tidak hanya berjalan secara fungsional, tetapi juga memenuhi standar kualitas, keamanan, dan *maintainability* yang ditetapkan oleh perusahaan.

Pada tahap ini, peserta magang terlibat dalam proses code review yang telah dikembangkan terhadap kode yang dibuat secara mandiri. Proses *code review* dilakukan dengan arahan langsung dari *senior programmer*, yang berperan sebagai penanggung jawab kualitas sistem. Melalui proses ini, peserta magang mempelajari standar penulisan kode yang diterapkan di PT CS Indonesia, termasuk konsistensi struktur kode, penamaan variabel dan fungsi, penerapan konsep modularisasi, serta pemanfaatan fitur *framework* Laravel secara optimal.

Code review tidak hanya berfokus pada aspek sintaks dan struktur kode, tetapi juga mencakup evaluasi terhadap logika program, efisiensi *query database*, serta potensi terjadinya *bug* atau *error* pada kondisi tertentu. Setiap temuan dalam proses *review* didiskusikan bersama *senior programmer* untuk menentukan langkah perbaikan yang paling tepat. Dengan ini, peserta magang tidak hanya melakukan perbaikan kode secara teknis, tapi juga pemahaman yang lebih mendalam mengenai alasan di balik setiap keputusan pengembangan.

Selain *code review*, fase ini juga mencakup kegiatan *quality improvement* yang bertujuan untuk meningkatkan kualitas sistem secara keseluruhan. Kegiatan *quality improvement* meliputi proses *debugging* berdasarkan laporan dari unit bisnis atau *business analyst*, penyempurnaan fitur yang telah dikembangkan sebelumnya, serta optimalisasi performa sistem agar lebih stabil saat digunakan oleh klien. Permintaan perbaikan umumnya disampaikan dalam bentuk tiket yang berisi deskripsi masalah atau kebutuhan penyesuaian,

sehingga setiap perbaikan dapat dilakukan secara terstruktur dan terdokumentasi dengan baik.

Dalam pelaksanaannya, peserta magang turut dilibatkan dalam proses perbaikan *bug* yang ditemukan selama tahap pengujian maupun setelah sistem digunakan oleh klien. Proses ini mengharuskan peserta magang untuk menganalisis sumber permasalahan, menelusuri alur logika program, serta memastikan bahwa perbaikan yang dilakukan tidak menimbulkan dampak negatif pada fitur lain. Setelah perbaikan selesai, kode akan diuji kembali sebelum akhirnya dinyatakan siap untuk diimplementasikan.

Tahap akhir dari fase ini adalah proses validasi dan persiapan implementasi ke lingkungan produksi. Setelah seluruh perbaikan dan peningkatan kualitas dinyatakan selesai, *senior programmer* melakukan pengecekan akhir dan melanjutkan proses *deployment* ke server produksi. *Business analyst* kemudian melakukan konfirmasi kepada klien terkait penyelesaian pekerjaan yang diminta. Dengan adanya fase *code review* dan *quality improvement* ini, perusahaan dapat memastikan bahwa sistem yang diserahkan kepada klien memiliki kualitas yang baik, stabil, serta sesuai dengan kebutuhan operasional.

3.3.2 Kendala yang Ditemukan

Selama menjalani proses kerja magang di PT CS Indonesia, terdapat beberapa kendala dan tantangan yang dihadapi dalam pelaksanaan kegiatan sehari-hari. Kendala-kendala ini muncul seiring dengan keterlibatan langsung peserta magang dalam pengembangan dan pemeliharaan sistem yang digunakan oleh klien perusahaan. Meskipun demikian, kendala tersebut menjadi bagian dari proses pembelajaran yang memberikan pengalaman nyata mengenai dinamika kerja di lingkungan industri teknologi informasi. Adapun

beberapa kendala utama yang dihadapi selama pelaksanaan magang adalah sebagai berikut.

1. Salah satu kendala utama yang dihadapi adalah proses pemahaman terhadap struktur dan mekanisme kerja sistem yang telah berjalan, khususnya pada pemanfaatan database perusahaan. *Database* yang digunakan memiliki jumlah tabel dan relasi yang cukup banyak dengan tingkat kompleksitas yang tinggi. Selain itu, sebagian penamaan tabel dan *field* menggunakan bahasa Jerman, sehingga memerlukan waktu tambahan untuk memahami fungsi serta keterkaitan antar data yang ada. Di sisi lain, dokumentasi *database* yang tersedia masih terbatas, sehingga proses adaptasi dan pemahaman sistem perlu dilakukan melalui eksplorasi mandiri serta penelusuran langsung terhadap struktur data dan kode program yang digunakan.
2. Keterbatasan waktu dalam penyelesaian proyek pada periode tertentu. Kendala lain yang dihadapi adalah adanya periode tertentu di mana penyelesaian proyek atau fitur harus dilakukan dalam waktu yang relatif singkat. Hal ini terjadi akibat tuntutan kebutuhan klien maupun jadwal pengembangan internal perusahaan yang cukup padat. Kondisi tersebut menuntut peserta magang untuk mampu mengatur waktu dan menentukan prioritas pekerjaan secara lebih efektif, agar seluruh tugas dapat diselesaikan sesuai dengan target waktu yang telah ditetapkan tanpa mengorbankan kualitas hasil kerja.
3. Minimnya penjelasan dari sisi bisnis dalam pengerjaan proyek. Kendala berikutnya berkaitan dengan keterbatasan penjelasan dari sisi bisnis dalam proses pengerjaan proyek, baik pada pengembangan sistem baru maupun pada kegiatan pemeliharaan sistem yang telah berjalan. Arahkan yang diberikan kepada peserta magang cenderung lebih berfokus pada aspek teknis, sementara penjelasan mengenai konteks bisnis, tujuan fitur, serta dampak sistem terhadap operasional klien sering kali masih terbatas. Kondisi ini mengharuskan peserta magang untuk melakukan eksplorasi dan analisis secara mandiri guna memahami kebutuhan bisnis

yang mendasari pengembangan atau perbaikan sistem, sehingga dapat menentukan solusi teknis yang tepat dan relevan.

3.3.3 Solusi atas Kendala yang Ditemukan

Berdasarkan berbagai kendala yang dihadapi selama pelaksanaan kerja magang, peserta magang berupaya melakukan beberapa langkah dan pendekatan sebagai solusi untuk mengatasi permasalahan tersebut. Upaya-upaya ini dilakukan secara bertahap dengan menyesuaikan kondisi kerja, arahan dari supervisor, serta kebutuhan proyek yang sedang dikerjakan. Adapun solusi yang diterapkan antara lain sebagai berikut.

1. Pemanfaatan sumber daya pendukung dan koordinasi dengan pihak terkait. Untuk mengatasi kesulitan dalam memahami struktur sistem dan database, peserta magang memanfaatkan berbagai sumber daya pendukung yang tersedia, termasuk penggunaan bantuan teknologi *Artificial Intelligence* (AI) sebagai alat bantu dalam memahami struktur kode dan relasi *database*. Selain itu, peserta magang secara aktif melakukan diskusi dan konsultasi dengan *supervisor* serta *business analyst* yang memiliki pemahaman lebih mendalam terkait sistem dan *database* perusahaan. Melalui koordinasi ini, proses adaptasi dapat dilakukan dengan lebih terarah dan kesalahan pemahaman dapat diminimalkan.
2. Peningkatan efisiensi kerja melalui pengelolaan waktu dan prioritas. Dalam menghadapi keterbatasan waktu pada periode tertentu, peserta magang berupaya meningkatkan efisiensi kerja dengan melakukan pengelolaan waktu dan penentuan prioritas tugas secara lebih baik. Apabila memungkinkan, pengerjaan dilakukan terhadap beberapa fitur atau tugas secara paralel dengan tetap memperhatikan tingkat urgensi dan kompleksitas pekerjaan. Pendekatan ini membantu memastikan bahwa seluruh tugas dapat diselesaikan sesuai tenggat waktu yang ditentukan tanpa mengurangi kualitas hasil pengembangan.

3. Eksplorasi mandiri dan pendalaman konteks bisnis sistem. Untuk mengatasi keterbatasan penjelasan dari sisi bisnis, peserta magang melakukan eksplorasi secara mandiri terhadap sistem yang sedang dikerjakan, baik melalui pengamatan langsung terhadap alur aplikasi, analisis kebutuhan pengguna, maupun penelusuran fungsi sistem yang telah ada. Apabila ditemukan kendala atau ketidakjelasan, peserta magang melakukan konsultasi lanjutan dengan *business analyst* guna memperoleh penjelasan yang lebih jelas dan akurat. Dengan ini, pemahaman terhadap cara kerja bisnis dan tujuan pengembangan sistem dapat meningkat, sehingga solusi teknis yang dihasilkan tetap selaras dengan kebutuhan operasional klien dan tujuan strategis perusahaan.

