

Pelaksanaan Magang Kerja: *Full Stack Developer* Pada Mitra PT. Celindo Pratama

Mangapul Siahaan¹, Tevin Lim²

Universitas Internasional Batam

e-mail: mangapul.siahaan@uib.ac.id¹, 2031132.tevin@uib.ac.id²

Abstrak

PT. Celindo Pratama didirikan pada 2013 sebagai perusahaan Teknologi Informasi yang fokus pada dukungan IT, pemeliharaan, dan pemrograman. Mereka menyediakan berbagai layanan, termasuk desain sistem informasi akademik, program keuangan, dan *Point of Sale*. Klien sebelumnya meminta pengembangan sistem baru karena hambatan dalam proses kerja yang berpotensi merusak atau memperlambat pemeliharaan mesin. Masalah lain adalah kurangnya pengendalian terhadap *Data Master Customer* yang mengakibatkan ketidakpastian dalam konsistensi dan akurasi. Dalam pengembangan sistem, penulis menggunakan metode kualitatif yaitu wawancara untuk mengumpulkan data dalam pengembangan aplikasi dan mengadopsi metode *Software Development Life Cycle* (SDLC) untuk merencanakan dan melaksanakan pembuatan perangkat lunak berkualitas tinggi. Selama magang, penulis berhasil merancang sistem *Total Productive Maintenance* dan *Master Data Management*.

Abstract

PT. Celindo Pratama was established in 2013 as an Information Technology company focusing on IT support, maintenance, and programming. They offer various services, including academic information system design, financial programs, and Point of Sale solutions. Previous clients requested the development of a new system due to obstacles in their workflow that had the potential to disrupt or slow down machine maintenance. Another issue was the lack of control over the Master Customer Data, leading to uncertainty in consistency and accuracy. In the system development, the author employed a qualitative method, namely interviews, to gather data during the application development process, and adopted the Software Development Life Cycle (SDLC) method to plan and execute the creation of high-quality software. During the internship, the author successfully designed the Total Productive Maintenance system and Master Data Management.

Keywords: *Information Technology, Programming, Qualitative, Software Development Life Cycle (SDLC)*

Pendahuluan

Sistem informasi merupakan penerapan teknologi informasi dan komunikasi yang digunakan oleh perusahaan bisnis. Di era abad ke-21, pentingnya dukungan sistem informasi sangat signifikan mengingat adanya perubahan yang cepat dalam dunia yang terus berkembang (Primawanti & Ali, 2022). Hal ini menandakan adanya peningkatan

penggunaan sistem informasi pada berbagai bidang, antara lain bidang pendidikan, bidang ekonomi, bidang kesehatan, bidang pemerintahan, dan bidang sosial-budaya. Pada dasarnya, teknologi digunakan untuk memudahkan pekerjaan manusia. Saat ini, teknologi sudah menjadi kebutuhan primer manusia. Bahkan, teknologi sudah digunakan dalam segala aspek kehidupan manusia.

Berdasarkan data yang ditemukan di Google Play, pada awal tahun 2021 terdapat sekitar 3,48 juta aplikasi yang tersedia, sedangkan di Apple App Store terdapat sekitar 2,22 juta aplikasi (Harsono, 2022). Diperkirakan jumlah tersebut akan terus meningkat seiring berjalannya waktu. Itu semua merupakan hasil dari karya para *programmer* yang merancang dan mengelola aplikasi untuk berbagai kebutuhan pengguna. Singkatnya, pekerjaan sebagai *programmer* sangatlah penting. Oleh karena itu, tidak heran jika di era saat ini banyak berkembang produk-produk digital yang tersebar di berbagai *platform* virtual, baik berupa website maupun aplikasi berbasis desktop ataupun *mobile*. Pengembang perangkat lunak lebih fokus bekerja di ranah digital, yakni dalam mendesain produk digital seperti *landing page*, aplikasi, *dashboard*, dan website. Pengembang perangkat lunak perlu mengetahui apa yang diinginkan pengguna dan dapat membuat kode dengan cerdas untuk menciptakan aplikasi yang dapat diakses oleh pengguna. Tujuan dari pengembang perangkat lunak adalah meningkatkan kepuasan konsumen saat menggunakan antarmuka website, desktop, dan *mobile* (Prasetyo et al., 2022).

Pada program magang ini, penulis bekerja sebagai *Full Stack Developer*. Dimana penulis akan membangun sebuah sistem *Master Data Management* (MDM) dan *Total Productive Maintenance* (TPM) yang terdiri dari aplikasi *desktop* dan juga website. Penulis akan membuat dari segi tampilan *Frontend* dan *Backend* agar aplikasi ini dapat berjalan dengan baik dan benar.

Masalah

Mitra yang terlibat dalam rangkaian kegiatan magang kerja ini adalah PT. Celindo Pratama, sebuah entitas yang tengah membutuhkan kehadiran seorang pengembang perangkat lunak serba bisa, yang umumnya dikenal sebagai *Full Stack*

Developer. Mitra ini telah mengidentifikasi perlunya kehadiran individu dengan kompetensi yang luas dan mendalam sesuai dengan kebutuhan yang mereka hadapi. Tugas seorang *Full Stack Developer* ini akan menjadi sangat signifikan dalam menjalankan dan memajukan proyek-proyek vital dari mitra, yakni proyek *Master Data Management* (MDM) dan *Total Productive Maintenance* (TPM), yang mencakup pengembangan aplikasi desktop dan situs web.

Dalam periode ini, mitra menemui tantangan dalam hal kekurangan sumber daya yang mumpuni dalam bidang pengembangan perangkat lunak. Sebagai solusi atas tantangan ini, kesempatan diberikan kepada penulis untuk bergabung dalam program magang kerja dengan peran sebagai *Full Stack Developer* di bawah naungan mitra. Melalui peran ini, penulis berperan dalam mengembangkan proyek *Master Data Management* (MDM) dengan basis situs *web*, serta *Total Productive Maintenance* (TPM) yang meliputi aplikasi desktop dan juga situs *web*, semua sesuai dengan persyaratan dan kebutuhan yang ditetapkan oleh klien.

Partisipasi penulis sebagai *Full Stack Developer* memberikan manfaat ganda: pertama, mitra mendapatkan sumber daya yang kritis dalam upaya pengembangan sistem yang mendukung konsumen mereka; dan kedua, penulis mendapatkan peluang untuk mengaplikasikan pengetahuan dan keterampilan yang telah mereka peroleh selama studi mereka dalam lingkungan dunia nyata. Melalui kolaborasi ini, kebutuhan mitra akan sumber daya pengembang perangkat lunak terpenuhi dan penulis dapat mendapatkan pengalaman berharga yang berkontribusi pada pemahaman praktis serta perkembangan profesional mereka dalam industri teknologi informasi.

Metode

Teknik pengumpulan data yang digunakan penulis adalah wawancara secara daring. Wawancara merupakan suatu bentuk percakapan yang memiliki tujuan tertentu. Proses percakapan ini melibatkan dua pihak, yaitu peneliti atau pewawancara (*interviewer*) yang mengajukan pertanyaan, dan informan terwawancara yang memberikan jawaban terhadap pertanyaan tersebut (Vebrianto et al., 2022). Wawancara dilakukan untuk mengumpulkan data yang diperlukan dalam pengembangan aplikasi *Total Productive Maintenance* dan *Master Data Management* dengan menggunakan wawancara daring melalui aplikasi Zoom atau WhatsApp sebagai sarana komunikasi untuk memenuhi kebutuhan pengembangan aplikasi.

Proses perancangan sistem *Master Data Management* (MDM) dan *Total Productive Maintenance* (TPM) akan dilakukan dengan metode *Software Development Life Cycle* (SDLC). SDLC merupakan serangkaian tahapan kerja yang dilakukan oleh analis sistem dan *programmer* untuk membangun sistem informasi, serta merupakan metode yang digunakan dalam mengembangkan sistem tersebut (Mallisza et al., 2022). Tahapan-tahap dalam SDLC sebagai berikut:

1. Analisis Kebutuhan

Pada tahap yang signifikan ini, penulis mengambil langkah pertama dalam bentuk analisis kebutuhan sistem yang menjadi pijakan penting bagi pengembangan lebih lanjut. Proses ini didasarkan pada data yang telah berhasil dihimpun melalui wawancara secara daring dengan pihak-pihak terkait. Informasi yang diperoleh dari interaksi ini menjadi bahan mentah yang kemudian dianalisis secara seksama. Dari hasil analisis ini, tampak bermunculan fitur-fitur yang diidentifikasi sebagai elemen penting dalam implementasi sistem *Master*

Data Management (MDM) dan *Total Productive Maintenance* (TPM).

Fitur-fitur ini bukan hanya daftar biasa, tetapi adalah pilihan fitur yang telah diurutkan berdasarkan pentingnya. Tim pengembang menggunakan fitur-fitur ini sebagai dasar untuk langkah-langkah selanjutnya. Fitur-fitur ini sangat penting dalam menentukan bagaimana sistem akan berfungsi dan apa yang akan dicapai. Oleh karena itu, tahap ini sangat terhubung dengan bagaimana luaran sistem akan dirancang.

Secara singkat, tahap analisis kebutuhan sistem ini merupakan langkah pertama dalam proses pengembangan. Ia merangkum apa yang diperlukan oleh sistem berdasarkan informasi yang telah dikumpulkan, dan membentuk dasar untuk merencanakan langkah-langkah selanjutnya dalam merancang dan mengembangkan sistem.

2. Perancangan atau *Design*

Tahapan ini memperlihatkan peran penulis dalam merancang dan mendesain tampilan aplikasi dengan penuh keahlian. Tindakan ini dilakukan dengan merujuk pada kebutuhan yang telah ditetapkan oleh klien, serta mengintegrasikan prinsip-prinsip desain yang relevan dengan standar tampilan halaman proyek yang telah direncanakan sebelumnya. Desain yang dihasilkan dalam tahap ini bertujuan untuk menghasilkan tampilan awal atau *prototype* yang mencerminkan gambaran dasar dari bentuk akhir yang diinginkan.

Melalui proses ini, penulis memadukan aspek estetika dan fungsionalitas ke dalam satu kesatuan. Penekanan pada detail dan kreativitas memungkinkan pengembangan tampilan yang menarik dan intuitif bagi

pengguna akhir. Setiap elemen dalam desain ini harus memenuhi persyaratan klien serta terintegrasi dengan logika dan alur kerja dari aplikasi yang sedang dirancang.

Desain *prototype* ini menghasilkan gambaran visual yang lebih nyata dan komprehensif tentang bagaimana tampilan aplikasi akan tampak dan berinteraksi. Hal ini memfasilitasi komunikasi yang lebih baik dengan klien, karena mereka dapat secara langsung melihat bagaimana rancangan ini mencerminkan visi mereka. Desain ini juga membantu mengidentifikasi potensi masalah atau perbaikan sejak dini sebelum proses pengembangan lebih lanjut.

Secara keseluruhan, tahap perancangan tampilan aplikasi ini merupakan jembatan penting antara konsep abstrak dan realisasi nyata, dan menjadi landasan yang sangat penting bagi tahap-tahap pengembangan selanjutnya.

3. Pengkodean Sistem

Pada tahap ini, penulis terlibat dalam proses pengembangan komponen *frontend* dari *Total Productive Maintenance* dengan memanfaatkan teknologi *ASP.NET Web Forms applications* yang ditingkatkan dengan elemen-elemen *jquery* yang terintegrasi. Secara khusus, penggunaan *jquery* memberikan dinamika dan interaktivitas yang meningkatkan pengalaman pengguna.

Selanjutnya, dalam pengembangan *backend Total Productive Maintenance* berbasis situs *web*, penulis memilih untuk menggunakan bahasa pemrograman *C#* sebagai fondasi untuk membangun struktur yang kuat dan fungsional. Sementara itu, untuk bagian *backend* dari *Total Productive Maintenance*

yang berbasis desktop dan modul pengiriman email berbasis desktop, penulis memilih bahasa pemrograman *Visual Basic* untuk membangun solusi yang sesuai dengan konteks dan kebutuhan khusus.

Di sisi lain, dalam mengembangkan komponen *frontend* dari *Master Data Management*, penulis memutuskan untuk mengadopsi kerangka kerja *Bootstrap* yang memungkinkan tampilan yang responsif dan estetis. Penggunaan *HTML*, *CSS*, dan *Javascript* menjadi dasar dalam membangun tampilan yang menarik dan *user-friendly*.

Pada bagian *backend* dari *Master Data Management*, penulis memilih bahasa pemrograman *C#* sebagai tulang punggung untuk mengelola logika bisnis yang kompleks dan memastikan kohesifnya fungsionalitas seluruh sistem.

Sebagai fondasi untuk penyimpanan dan pengelolaan data, kedua proyek ini mengadopsi *SQL Server* sebagai sistem *database*. Ini memastikan keandalan dalam menyimpan, mengambil, dan memanipulasi data yang diperlukan oleh kedua sistem ini.

Secara keseluruhan, langkah-langkah dalam pengembangan ini mencerminkan strategi yang matang dan berorientasi pada kebutuhan. Pendekatan yang dipilih untuk masing-masing komponen *frontend* dan *backend* menunjukkan pemahaman mendalam tentang teknologi dan konteks penggunaan yang berbeda, sambil tetap memastikan bahwa keseluruhan sistem terintegrasi dengan baik.

4. Pengujian Sistem

Setelah proses integrasi selesai menggabungkan berbagai komponen

sistem menjadi kesatuan utuh, langkah selanjutnya adalah menghadapi sistem pada ujian ketat dengan metode pengujian *black box*. Misi dari tahap pengujian ini adalah memeriksa setiap sudut sistem dengan cermat untuk memastikan bahwa tidak ada celah yang mengakibatkan kesalahan (*error*), *bug*, atau masalah keamanan yang tidak terdeteksi.

Proses pengujian ini jauh lebih komprehensif daripada sekadar menguji *database* lokal. Selain menguji pada lingkungan lokal, sistem juga dijalankan pada *server* QA atau pengembangan. Hal ini dirancang untuk menciptakan simulasi lingkungan yang mendekati situasi dunia nyata. Lingkungan ini memungkinkan untuk mengidentifikasi potensi kesalahan yang mungkin timbul saat sistem beroperasi dalam skala yang lebih luas dan melibatkan interaksi yang lebih kompleks.

Tujuan utama dari pengujian ini adalah untuk menjaga kinerja sistem agar tetap optimal dan responsif, serta menghindari situasi dimana sistem dapat terhenti atau mengalami penurunan performa yang signifikan. Kemudian, memastikan bahwa tidak ada kesalahan atau cacat yang muncul di hadapan pengguna ketika sistem digunakan secara menyeluruh.

Dalam rangka mencapai tujuan ini, berbagai skenario pengujian dijalankan untuk menguji berbagai aspek sistem, termasuk fungsionalitas, interaksi pengguna, integrasi komponen, dan keamanan. Hasil pengujian akan digunakan untuk mengidentifikasi dan memperbaiki masalah yang ditemukan. Proses pengujian ini merupakan langkah kritis dalam memastikan bahwa sistem yang diimplementasikan akan bekerja

dengan lancar, efisien, dan aman saat digunakan oleh pengguna di seluruh jangkauan.

5. Evaluasi Sistem

Setelah pengujian dianggap mencapai standar yang memadai, sistem siap untuk diimplementasikan kepada pelanggan yang telah ditentukan oleh tim *IT Support* pelanggan. Meskipun pada tahap ini sistem telah secara resmi dapat diakses oleh publik, peran tim *IT Support* pelanggan tidak berakhir di sini. Sebaliknya, mereka menjalankan tanggung jawab krusial dalam memantau sistem secara terus-menerus untuk mengantisipasi potensi terjadinya kesalahan (*error*) atau *bug* saat sistem digunakan oleh pengguna akhir.

Dalam konteks ini, umpan balik dari pengguna memiliki peran penting. Jika ada laporan tentang kesalahan atau *bug* dari pengguna sistem, tim pengembang sistem akan segera merespons dengan memeriksa masalah yang dilaporkan. Perbaikan kode yang diperlukan akan segera dilakukan untuk memastikan bahwa masalah diatasi dan sistem berfungsi sesuai dengan harapan.

Pembahasan

1. Perancangan Luaran Kegiatan

Dalam konteks komitmen peningkatan, perhatian utama tertuju pada fase perancangan luaran kegiatan. Selama menjalani masa magang yang berharga di PT. Celindo Pratama, penulis telah berupaya dengan sungguh-sungguh untuk memberikan kontribusi yang berarti bagi perusahaan. Hasil jerih payah penulis terwujud dalam sejumlah luaran yang signifikan. Antara lain, penulis berhasil mengembangkan halaman dan fitur-

fitur baru yang telah disetujui baik oleh perusahaan maupun pelanggan. Beberapa diantaranya adalah *ETPM master data*, *ETPM monitoring request*, *ETPM check point for machine*, *ETPM request detail* dalam versi desktop, *ETPM job detail* dalam versi desktop, serta *ETPM report* dan *ETPM email sender*. Selain itu, penulis juga berhasil merancang *ETPM Check Points* dalam versi website, *ETPM request detail* dalam versi website, *ETPM job detail* dalam versi website, dan *MDM master data*. Semua pencapaian ini merupakan hasil dari dedikasi penulis dalam menghasilkan solusi yang sesuai dengan kebutuhan dan harapan perusahaan serta pelanggan.



Gambar 1. Contoh UI ETPM Versi Desktop

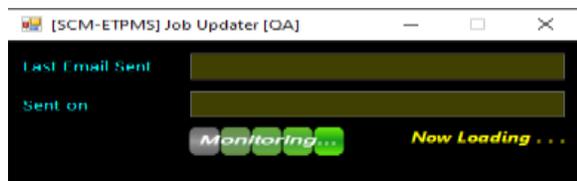
Proyek ETPM versi desktop melibatkan komponen *Master Data* yang menjadi sumber kebenaran tunggal dan konsisten untuk referensi dalam aktivitas operasional dan pengambilan keputusan di organisasi, terintegrasi dalam proyek *Total Productive Maintenance*. Fitur utama meliputi permintaan pemantauan ETPM yang memonitor berbagai jenis permintaan yang telah direncanakan maupun tak terduga, serta permintaan pemeliharaan yang sedang berlangsung. Fitur lainnya, poin pemeriksaan ETPM untuk mesin, berfungsi menetapkan jadwal pemeliharaan berdasarkan karakteristik

mesin, sehingga penjadwalan pemeliharaan mesin menjadi lebih efisien. Fitur khusus ini juga mempermudah pembuatan permintaan pemeliharaan sesuai jadwal yang telah ditetapkan. Solusi ini juga menangani permintaan pemeliharaan mesin yang tak terduga dengan efisien. Fitur laporan ETPM menjadi salah satu fitur penting untuk melihat hasil laporan pemeliharaan secara komprehensif dan terperinci, memberikan wawasan mendalam tentang aktivitas pemeliharaan sebelumnya.



Gambar 2. Contoh UI ETPM Versi Website

Dalam proyek ETPM versi website, terdapat fitur yang bertujuan mempermudah pelaksanaan tugas pemeliharaan mesin yang rutin direncanakan. Fitur ini tidak hanya mengoptimalkan penyelesaian tugas-tugas pemeliharaan yang telah dijadwalkan, tetapi juga berperan penting dalam menangani permintaan pemeliharaan yang tiba-tiba muncul pada mesin. Dengan menggunakan *platform* website yang disediakan, pengguna dapat lebih efisien mengeksekusi permintaan pemeliharaan tersebut. Secara keseluruhan, fitur ini bertujuan utama untuk mengelola permintaan pemeliharaan mesin yang tak terduga, memastikan proses pemeliharaan berjalan efisien dan tepat waktu melalui *platform* website yang tersedia.



Gambar 3. Contoh UI ETPM Email Sender

Dalam kerangka proyek ETPM *email sender*, fitur yang dihadirkan memiliki peran sentral dalam menyediakan beragam fungsi yang berkaitan dengan pengiriman email. Berbagai jenis email yang memiliki tujuan dan kegunaan yang berbeda-beda dikirimkan melalui fitur ini. Antara lain, fitur tersebut mencakup beberapa jenis email yang penting dalam menjalankan berbagai proses di dalam sistem, meliputi:

a. *Section Time Limit Email*

Jenis email ini dirancang untuk memberikan peringatan saat batas waktu suatu bagian atau tahap dalam proses ETPM mendekati atau melewati batas waktu yang telah ditetapkan. Dengan demikian, pemangku kepentingan dapat secara proaktif mengambil tindakan yang diperlukan untuk memastikan keteraturan dan kelancaran proses.

b. *ETPM Overdue Mail*

Email ini memiliki peran penting dalam memberikan notifikasi ketika suatu pekerjaan atau tugas dalam proses ETPM mengalami keterlambatan atau melewati tenggat waktu yang telah ditetapkan. Hal ini memungkinkan tim untuk segera merespon dan mengatasi keterlambatan yang terjadi.

c. *Approval Email*

Fitur ini berkaitan dengan proses persetujuan dalam ETPM. Email jenis ini digunakan untuk memberikan informasi tentang persetujuan yang telah diberikan oleh pihak terkait, sehingga semua pihak terkait memiliki pemahaman yang jelas mengenai status dan perkembangan proses.

d. *Job In Progress Email*

Jenis email ini bertujuan untuk memberikan pembaruan dan informasi tentang kemajuan suatu pekerjaan yang sedang berlangsung dalam proses ETPM. Pembaruan berkala melalui email ini membantu pemangku kepentingan tetap terinformasi tentang status pekerjaan.

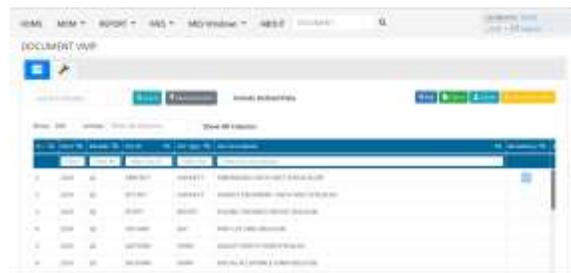
e. *Job Complete Email*

Fitur ini berfungsi untuk memberikan notifikasi dan konfirmasi ketika suatu pekerjaan atau tugas dalam proses ETPM telah selesai dilaksanakan. Ini membantu dalam menciptakan transparansi dan akuntabilitas terhadap penyelesaian tugas.

f. *NC Result Mail*

Email ini memiliki peran dalam menyampaikan hasil dan informasi terkait *Non-Conformance* (NC) atau hasil inspeksi yang tidak sesuai dengan standar atau persyaratan tertentu. Informasi ini penting untuk diakses oleh tim yang terkait dengan langkah-langkah perbaikan.

Seluruh jenis email yang diintegrasikan ke dalam fitur ETPM *email sender* ini bekerja secara sinergis untuk memastikan komunikasi yang efektif, pemantauan yang akurat, dan kesesuaian dengan jadwal dan persyaratan yang telah ditetapkan dalam proses ETPM.



Gambar 4. Contoh UI MDM Master Data

Dalam konteks proyek MDM, fitur yang diperkenalkan memiliki peranan utama dalam menjaga kendali terhadap *Data Master Customer*. Tujuan utama dari fitur ini adalah memastikan bahwa data tersebut senantiasa dipertahankan dalam keadaan konsisten, akurat, terkini, relevan, dan sesuai dengan konteksnya. Semua ini dilakukan dengan maksud untuk memenuhi kebutuhan bisnis yang beragam dan berbeda di berbagai aplikasi dan divisi yang beroperasi di dalam organisasi.

Dalam mengelola *data master* pelanggan, fitur ini berperan sebagai penjaga keandalan informasi yang berkaitan dengan pelanggan di seluruh perusahaan. Dengan memastikan data konsisten, setiap divisi atau aplikasi dalam organisasi dapat mengandalkan data yang sama untuk pengambilan keputusan yang lebih baik dan akurat. Keakuratan dan kekonsistenan data ini juga membantu dalam menghindari kesalahan dan ambiguitas yang dapat timbul akibat data yang tidak terkendali.

Selain itu, fitur ini juga memastikan bahwa data yang disimpan tetap terkini dan relevan. Dalam lingkungan bisnis yang terus berubah, informasi yang sudah tidak relevan atau sudah usang dapat merugikan efisiensi operasional dan pengambilan keputusan yang tepat. Oleh karena itu, fitur ini berperan sebagai penjaga agar hanya data yang aktual dan relevan yang tersimpan dan dapat diakses.

Integrasi fitur MDM ini menjadi penengah dalam menjembatani berbagai divisi dan aplikasi yang mungkin memiliki kebutuhan data yang berbeda. Hal ini menciptakan sinergi di seluruh organisasi dan memastikan bahwa data pelanggan yang dimiliki oleh berbagai entitas dalam perusahaan memiliki keselarasan yang konsisten, memperkuat visibilitas, serta meningkatkan efektivitas operasional secara keseluruhan.

2. Proses Implementasi Luaran

Proses implementasi dari luaran dimulai dengan tahap pengumpulan kebutuhan bersama klien dan *Product Manager*. Klien dan tim akan mengadakan pertemuan yang intensif untuk membahas perubahan atau penambahan fitur yang diinginkan. Hasil dari diskusi ini diwujudkan dalam bentuk dokumentasi berupa presentasi Power Point yang diberikan kepada penulis dan *Product Manager*.



Gambar 5. Contoh Dokumentasi Requirement Gathering

Selanjutnya, dalam persiapan tahap implementasi, penulis memiliki tugas penting untuk menyusun kueri-kueri yang diperlukan untuk menambahkan tabel pada *database SQL Server*. Selain itu, perancangan logika untuk operasi *Create, Read, Update, dan Delete (CRUD)* pada penambahan fitur harus dipersiapkan dengan cermat. Selanjutnya, langkah penambahan kode-kode yang sesuai dengan fitur yang diperlukan juga dilakukan.

Tahap berikutnya adalah pengecekan yang dilakukan oleh tim pengembang dan tim *IT Support customer*. Pengecekan ini menjadi langkah kritis untuk memastikan bahwa tidak ada kesalahan atau *bug* yang dapat mengganggu kinerja sistem. Proses pengecekan dilakukan di lingkungan *QA Server* yang merefleksikan kondisi realistis. Hal ini penting untuk meminimalkan risiko

masalah saat implementasi di lingkungan produksi.

Setelah melalui tahap pengecekan yang akurat, implementasi resmi dilakukan. File-file perubahan atau file terbaru yang telah dirancang akan di-zip menjadi satu file. File ini kemudian diunggah ke Google Drive yang dapat diakses oleh tim *IT Support Customer*. Dalam tahap implementasi, tim *IT Support Customer* juga turut membantu untuk memastikan bahwa perubahan diterapkan dengan lancar dan sesuai dengan rencana.

Keseluruhan proses implementasi ini mencerminkan kerja sama yang erat antara berbagai tim yang terlibat. Dari pengumpulan kebutuhan hingga tahap penerapan, setiap langkah dijalankan dengan cermat dan terkoordinasi untuk memastikan bahwa fitur-fitur baru atau perubahan yang diinginkan dapat diimplementasikan dengan sukses dan minimal risiko.

3. Kondisi Setelah Implementasi

Setelah tahap implementasi selesai, dampak positif yang terlihat adalah berkurangnya hambatan dalam proses kerja, mencegah potensi kerusakan, serta menghindari keterlambatan saat menjalankan pemeliharaan mesin. Selain itu, pengelolaan *Data Master Customer* kini mencerminkan konsistensi, akurasi, ketepatan waktu, relevansi, dan kontekstualitas yang penting untuk memenuhi beragam kebutuhan bisnis di berbagai aplikasi dan divisi.

Pengembangan proyek untuk sistem *Total Productive Maintenance* dan *Master Data Management* berhasil diselesaikan sesuai dengan spesifikasi yang telah diuraikan dalam PRD (*Product Requirement Document*) dan persyaratan yang telah dibahas bersama

klien. Hasil aplikasi ini diterima dengan baik oleh klien, menunjukkan kesesuaian antara yang diinginkan dan yang dihasilkan.

Lebih lanjut, kedua aplikasi ini telah diintegrasikan pada situs *web*, memberikan aksesibilitas yang lebih luas. Selain itu, sistem *Total Productive Maintenance* juga telah diimplementasikan dalam versi desktop, memungkinkan penggunaan yang lebih fleksibel dan mudah diakses.

Simpulan

Selama kegiatan magang kerja sebagai *Full Stack Developer* di PT. Celindo Pratama, berbagai kesimpulan dapat diambil. Pertama, proyek pengembangan sistem *Total Productive Maintenance* dan *Master Data Management* telah berhasil diselesaikan dengan sukses, sesuai dengan spesifikasi yang tercantum dalam PRD dan persyaratan yang telah disetujui bersama klien. Respons positif dari klien terhadap hasil aplikasi merupakan indikasi keberhasilan proyek ini. Kedua aplikasi ini telah berhasil diintegrasikan pada situs *web*, serta sistem *Total Productive Maintenance* telah berhasil diimplementasikan dalam versi desktop.

Kedua, penerapan metode *Software Development Life Cycle* (SDLC) telah berjalan dengan baik dalam merencanakan, mengelola, dan melaksanakan langkah-langkah penting dalam pengembangan perangkat lunak. Pendekatan ini terbukti membantu menghasilkan perangkat lunak berkualitas tinggi yang sesuai dengan kebutuhan dan harapan klien.

Terakhir, aplikasi yang telah dikembangkan telah berhasil diadopsi oleh pengguna dengan luas. Sistem *Total Productive Maintenance* memberikan manfaat dalam memastikan kelancaran proses kerja dan mencegah kerusakan serta keterlambatan saat pemeliharaan mesin. Sementara itu, sistem *Master Data*

Management membantu mengelola *Data Master Customer* agar tetap konsisten, akurat, terkini, relevan, dan kontekstual sesuai dengan kebutuhan bisnis yang beragam di berbagai aplikasi dan divisi. Keseluruhan hasil magang ini mencerminkan pencapaian yang positif dalam mengembangkan solusi perangkat lunak yang efektif dan sesuai dengan kebutuhan klien.

Daftar Pustaka

- Harsono, H. (2022). *FAKTOR-FAKTOR YANG MEMPENGARUHI SISTEM INFORMASI BERBASIS KOMPUTER: SISTEM OPERASI, SERVER, DAN PROGRAMMER (LITERATURE REVIEW EXECUTIVE SUPPORT SISTEM FOR BUSINESS)*. 3(2). <https://doi.org/10.38035/jmpis.v3i2>
- Mallisza, D., Hadi, H. S., & Aulia, A. T. (2022). *Implementasi Model Waterfall Dalam Perancangan Sistem Surat Perintah Perjalanan Dinas Berbasis Website Dengan Metode SDLC*. 1(1), 24–35.
- Prasetyo, S. M., Simanjuntak, H., Laksono, D. B., Gilang, M., Gunawan, N., Pamulang, U., Selatan, T., Experience, U., & Digital, P. (2022). *Ui ux developer*. 1(1), 50–58.
- Primawanti, E. P., & Ali, H. (2022). *PENGARUH TEKNOLOGI INFORMASI, SISTEM INFORMASI BERBASIS WEB DAN KNOWLEDGE MANAGEMENT TERHADAP KINERJA KARYAWAN (LITERATURE REVIEW EXECUTIVE SUPPORT SISTEM (ESS) FOR BUSINESS)*. 3(3), 267–285.
- Vebrianto, R., Anwar, A., Studi, P., Pendidikan, M., Madrasah, G., Islam, U., Sultan, N., & Riau, S. K. (2022). *B u a n a p e n d i d i k a n*. 18(1), 43–48.