

# Rancang Bangun Website Order Pada Bagian Bonding Agent Di Mold Departement PT. RCI

Suharyanti<sup>1)</sup>

Teknologi Informasi, Politeknik Gajah Tunggal  
[suharyanti@poltek-gt.ac.id](mailto:suharyanti@poltek-gt.ac.id)

Faishal Hizbullaah<sup>2)</sup>

Teknik Elektronika, Politeknik Gajah Tunggal  
[faishalhisbu@gmail.com](mailto:faishalhisbu@gmail.com)

Ari Kuswantori<sup>3)</sup>

Teknik Elektronika, Politeknik Gajah Tunggal  
[ari@poltek-gt.ac.id](mailto:ari@poltek-gt.ac.id)

## ABSTRAK

Perkembangan pesat dalam teknologi informasi (IT) telah mendorong berbagai sektor industri untuk mengadopsi digitalisasi guna meningkatkan efisiensi dan produktivitas. PT. RCI, sebuah perusahaan manufaktur di Indonesia, menghadapi tantangan dalam proses pemesanan dan *monitoring manual* yang memakan waktu dan rawan kesalahan. Penelitian ini bertujuan untuk mengatasi masalah tersebut dengan merancang dan membangun sistem *order insert* berbasis *website*. Proses digitalisasi ini diharapkan dapat mengurangi waktu pemesanan, meningkatkan efisiensi operasional, dan meminimalkan kesalahan. Metode yang digunakan mencakup pemodelan *Unified Modeling Language* (UML), dengan pengembangan aplikasi menggunakan *framework* CodeIgniter dan *database* PostgreSQL. Hasil penelitian menunjukkan bahwa digitalisasi dan penggunaan teknologi informasi dapat meningkatkan efisiensi dan efektivitas dalam pengelolaan data. Implementasi sistem ini diharapkan dapat membawa perbaikan signifikan dalam proses produksi PT. RCI serta memberikan manfaat yang lebih luas dalam industri manufaktur.

Kata Kunci : *Web Based Application, Unified Modelling Language* (UML), CodeIgniter, PostgreSQL.

## ABSTRACT

*The rapid advancement in information technology (IT) has prompted various industrial sectors to adopt digitalization to enhance efficiency and productivity. PT. RCI, a manufacturing company in Indonesia, faces challenges with time-consuming and error-prone manual ordering and monitoring processes. This study aims to address these issues by designing and building a website-based order insert system. The digitalization process is expected to reduce ordering time, improve operational efficiency, and minimize errors. The methods used include Unified Modeling Language (UML) modeling, with application development employing the CodeIgniter framework and PostgreSQL database. The study results indicate that digitalization and the use of information technology can improve data management efficiency and effectiveness. The implementation of this system is anticipated to bring significant improvements to PT. RCI's production processes and provide broader benefits to the manufacturing industry.*

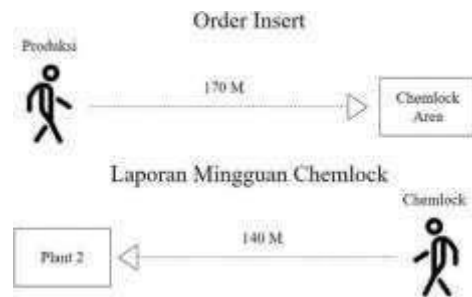
*Keywords: Digitalization in Manufacturing, Unified Modeling Language (UML), CodeIgniter, PostgreSQL.*

## I. PENDAHULUAN

Perkembangan teknologi yang pesat, terkhususnya dalam bidang *information technology* (IT) , telah mendorong berbagai sektor industri untuk mengikuti arus digitalisasi guna meningkatkan efisiensi dan produktivitas. Salah satu implementasi teknologi informasi yang signifikan adalah penggunaan *website* sebagai media tambahan informasi bagi berbagai pelaku industri, baik di bidang manufaktur, pendidikan maupun kesehatan. Dalam melakukan inovasi teknologi, industri tentunya dihadapkan pada tuntutan untuk meningkatkan produktivitas dan mengurangi biaya produksi sembari menjaga kualitas produk dan kepuasan pelanggan. Inovasi yang dilakukan haruslah berkelanjutan, yang sering kali diimplementasikan dalam konsep *kaizen*, yang menekankan perbaikan berkelanjutan dan berkesinambungan [1]. Salah satu bentuk implementasi *kaizen* yaitu melalui digitalisasi proses dengan harapan mampu menurunkan biaya produksi dan meningkatkan efisiensi operasional. PT. RCI adalah salah satu perusahaan terkemuka yang berperan dalam pengembangan industri manufaktur di Indonesia, khususnya dalam produksi berbagai produk berbasis karet. Perusahaan ini menawarkan berbagai macam produk untuk sektor pertanian, transportasi, industri, infrastruktur, otomotif, dan banyak lagi. Salah satu departemen krusial di PT. RCI adalah Departemen Mould, yang memiliki bagian Bonding Agent. Bagian ini bertanggung jawab atas proses *chemlock*, yaitu proses penyemprotan cairan kimia khusus pada permukaan *insert*. Secara umum, proses pemesanan dan *monitoring* yang dilakukan oleh operator di PT. RCI masih bersifat manual. Hal ini mengakibatkan lamanya waktu yang diperlukan untuk menyelesaikan proses tersebut. Proses pemesanan dapat memakan waktu hingga 39 menit 41 detik karena banyaknya tahapan yang harus dilalui. Tahapan-tahapan tersebut meliputi pengecekan stok di lini produksi, pembuatan data order, perjalanan dari bagian produksi *curing mould* ke bagian Bonding Agent, pemberian informasi pesanan kepada bagian terkait, dan perjalanan kembali ke lini produksi.

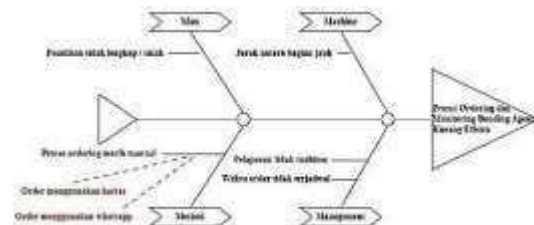
Gambar 1 menunjukkan jarak yang harus ditempuh oleh operator selama melaksanakan proses pemesanan. Jarak yang signifikan ini menambah waktu dan usaha yang diperlukan, sehingga mengurangi efisiensi dan produktivitas secara

keseluruhan. Selain itu, metode manual ini juga rentan terhadap kesalahan manusia dan kehilangan data, yang dapat berdampak negatif pada kualitas dan ketepatan waktu produksi.



Gambar 1. Kegiatan *Order* dan Pelaporan

Untuk menganalisis penyebab terjadinya masalah terkait lamanya durasi order dan *monitoring*, maka dibuat analisis *fishbone* seperti pada Gambar 2.



Gambar 2. *Fishbone* Diagram

Diagram *Fishbone* menunjukkan permasalahan utama dalam proses *ordering* dan *monitoring* di bagian Bonding Agent yang kurang efisien. Jarak antara bagian produksi *curing mould* dengan Bonding Agent menyebabkan proses pemesanan menjadi lambat dan tidak efisien. Penggunaan personal *chat* melalui WhatsApp untuk pemesanan tidak disarankan karena kurangnya transparansi dan sulitnya *monitoring* data. Selain itu, pelaporan yang tidak dilakukan secara *real-time* dan dikumpulkan pada akhir minggu menyebabkan risiko hilangnya kertas laporan dan masalah penyimpanan. Berdasarkan analisis di atas, penelitian ini akan fokus pada perancangan dan pembangunan digitalisasi sistem order *insert* berbasis *website*. Proses pembuatan *website* akan menggunakan *Unified Modeling Language* (UML) sebagai acuan untuk merancang alur kegiatan yang akan dilaksanakan oleh operator dalam penggunaan *website*. *Website* akan dibangun menggunakan *framework* CodeIgniter dan PostgreSQL sebagai *database* untuk penyimpanan data. Selain itu, XAMPP akan digunakan sebagai server lokal untuk pengujian, dan Visual Studio Code akan digunakan sebagai editor untuk pengembangan kode.

Berikut merupakan kajian pustaka yang digunakan pada penelitian:

Tabel 1. Kajian Pustaka

Sumber	Hasil Kajian
[1]	Penelitian ini mengubah sistem purchase order manual menjadi digital dengan metode <i>Rapid Application Development</i> (RAD). Proses pengembangan meliputi pemodelan bisnis, data, dan proses, pembentukan aplikasi, serta pengujian. Hasil penelitian menunjukkan bahwa aplikasi ini meningkatkan efisiensi pengelolaan data barang, data supplier, dan permintaan barang secara digital.
[2]	Penelitian ini menjelaskan penggunaan metode <i>black box</i> dalam pengujian perangkat lunak untuk aplikasi <i>Inventori</i> di PT. Aino Indonesia. Metode ini memastikan bahwa perangkat lunak berfungsi sesuai <i>test case</i> yang dibuat, sehingga aplikasi dinyatakan siap digunakan oleh pengguna.
[3]	Jurnal ini membahas penerapan <i>Framework CodeIgniter</i> untuk pengembangan aplikasi PHP berbasis web. Metode ini menyediakan desain fleksibel yang mengelola proses PQRS ( <i>Petitions, Complaints, Claims, and Suggestions</i> ), memungkinkan institusi mendapatkan informasi pengguna untuk memperbaiki proses dan meningkatkan layanan serta kepuasan.
[4]	Buku ini menjelaskan fungsi dari PostgreSQL, mencakup fitur-fitur terbaru, pembuatan tabel, keamanan basis data, serta teknik pemantauan, validasi cadangan, dan pemeliharaan rutin.
[5]	Jurnal ini menjelaskan perancangan dan pembuatan website menggunakan <i>Unified Modeling Language</i> (UML). Metode ini mencakup pembuatan diagram untuk merancang struktur dan fungsi situs, memastikan aplikasi <i>web</i> sesuai dengan rencana dan efektif dalam menyampaikan informasi kepada masyarakat dan calon siswa baru.

(Sumber : Hasil Kajian Penulis,2024)

### **Unified Modeling Language (UML)**

*Unified Modeling Language* adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, membangun, dan mendokumentasikan sebuah sistem pengembangan

*software* berbasis OO (*Object-Oriented*). UML sendiri juga memberikan standar penulisan sebuah sistem *blue print*, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema *database*, dan komponen-komponen yang diperlukan dalam sistem *software*.

### **Use Case Diagram**

*Use case diagram* adalah suatu *Diagram* yang menggambarkan interaksi antara aktor dalam sebuah sistem yang akan dibuat, *Use Case Diagram* terdiri dari beberapa aktor yang terhubung di dalamnya [1].

### **Activity Diagram**

*Activity Diagram* adalah diagram yang mengilustrasikan aliran kerja atau kegiatan dari suatu sistem atau proses bisnis, dengan kata lain diagram ini menggambarkan aktivitas yang dapat dilakukan oleh sistem tersebut [2].

### **Visual Studio Code**

Visual Studio Code dapat mengenali jenis bahasa pemrograman yang digunakan dan memberikan variasi warna yang sesuai untuk setiap fungsi dalam kode tersebut. Selain itu, aplikasi ini sudah terintegrasi dengan GitHub dan memiliki fitur untuk menambahkan ekstensi, memungkinkan pengembang untuk memperluas fungsionalitas yang tidak tersedia secara *default* di Visual Studio Code [3].

### **Framework CodeIgniter**

*Framework CodeIgniter* adalah alat yang membantu dalam membuat aplikasi web menggunakan bahasa pemrograman PHP. Dengan adanya berbagai pustaka yang tersedia, pengembangan aplikasi bisa dilakukan dengan cepat. Selain itu, antarmuka yang sederhana dan struktur *logisnya* membuat pengaksesan menjadi lebih mudah [4].

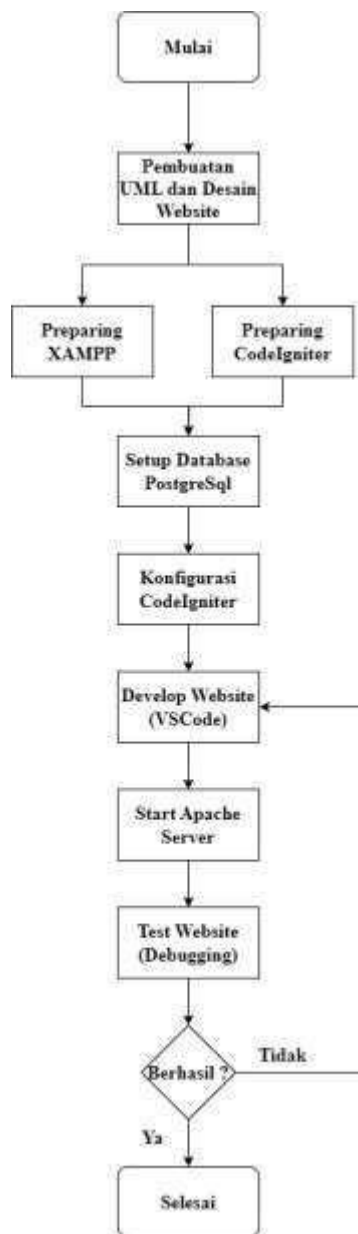
### **Database PostgreSQL**

*Database PostgreSQL* adalah sistem manajemen basis data (DBMS) sumber terbuka yang kuat dengan reputasi yang sangat baik dalam kinerja tinggi dan stabilitas yang menawarkan fitur-fitur seperti: berbagai jenis data yang berbeda dan sangat spesifik (*different and very specific types of data*), pemicu (*triggers*), fungsi, tampilan, dll [4].

### **Black Box Testing**

Pengujian *Black Box* atau lebih dikenal sebagai pengujian perilaku, dilakukan tanpa pengetahuan tentang struktur interior atau logika perangkat lunak yang diuji, pengujian mengacu pada spesifikasi kebutuhan dan tidak perlu menganalisis kode dan pengujian ini dilakukan dengan sudut pandang *user* akhir [5].

## II. METODE PENELITIAN



Gambar 3. Alur Penelitian  
(Sumber: Hasil Kajian Penulis, 2024)

Pada Gambar 3 memperlihatkan alur penelitian yang dilakukan penulis dalam penelitian kali ini, penjelasan mengenai alur penelitian yang dilakukan adalah sebagai berikut:

### A. Pembuatan UML dan Desain Website

Tahap pertama ini melibatkan pembuatan diagram UML untuk memodelkan struktur dan perilaku sistem yang akan dikembangkan sesuai dengan proses kegiatan yang berjalan. Selain itu, desain *website* juga dibuat untuk menentukan tampilan dan alur navigasi pada *website*.

### B. *Preparing Xampp dan CodeIgniter*

Tahap ini meliputi persiapan lingkungan pengembangan dengan menginstal XAMPP untuk server lokal dan *framework* CodeIgniter. XAMPP menyediakan Apache, PHP, dan MySQL yang diperlukan untuk pengembangan dan pengujian *website*.

### C. Setup Database PostgreSQL

Pada tahap ini, *database* PostgreSQL disiapkan dengan membuat *database* dan tabel yang dibutuhkan oleh aplikasi. Struktur tabel dan hubungan antar tabel didefinisikan sesuai dengan model UML yang telah dibuat sebelumnya.

### D. Konfigurasi CodeIgniter

Pada tahap ini, CodeIgniter dikonfigurasi agar dapat terhubung dengan *database* PostgreSQL yang telah disiapkan. Konfigurasi ini mencakup pengaturan *file* `database.php` dan penyesuaian pengaturan lainnya sesuai dengan kebutuhan aplikasi.

### E. Developing Website

Pada tahap ini, pengembangan *website* dilakukan menggunakan Visual Studio Code (VSCode) sebagai editor. Tahap ini mencakup penulisan kode untuk bagian *frontend* dan *backend* dari aplikasi, sesuai dengan desain dan model UML yang telah dibuat.

### F. Start Apache Server

Pada tahap ini, server Apache yang disediakan oleh XAMPP dijalankan untuk melakukan pengujian aplikasi secara lokal. Hal ini memungkinkan pengembang untuk melihat hasil pengembangan secara langsung di browser.

### G. Test Website (Debugging)

Tahap ini melibatkan pengujian dan *debugging* aplikasi untuk memastikan bahwa semua fungsi bekerja dengan baik dan tidak ada *bug*. Jika ditemukan kesalahan, perbaikan dilakukan hingga aplikasi berfungsi sesuai dengan yang diharapkan.

### H. Selesai

Tahap terakhir adalah penyelesaian proyek setelah semua tahap sebelumnya berhasil dilakukan dan aplikasi berjalan sesuai dengan yang diharapkan. Pada tahap ini, aplikasi siap untuk digunakan ke server produksi jika diperlukan

## III. HASIL DAN PEMBAHASAN

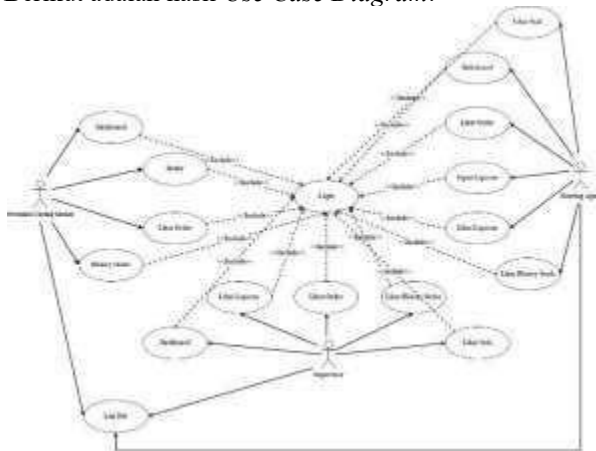
Pada Hasil dan Pembahasan bertujuan untuk mengetahui tahapan perancangan *website* ini sudah bisa digunakan dan sistem yang sudah dibuat dapat berjalan dengan baik sesuai dengan yang diharapkan.

### Pembuatan UML

Tahap perancangan ini menggunakan *Unified Modelling Language* (UML) sebagai alat dalam membantu perancangan, dimana UML yang di gunakan yaitu:

#### A. Use Case Diagram

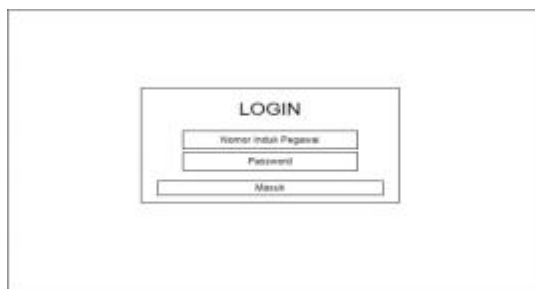
*Use Case Diagram* adalah suatu diagram yang menggambarkan interaksi antara aktor dalam sebuah sistem yang akan di buat, diagram ini terdiri dari beberapa aktor yang terhubung di dalamnya [1]. Berikut adalah hasil *Use Case Diagram*:



Gambar 4. Use Case Diagram

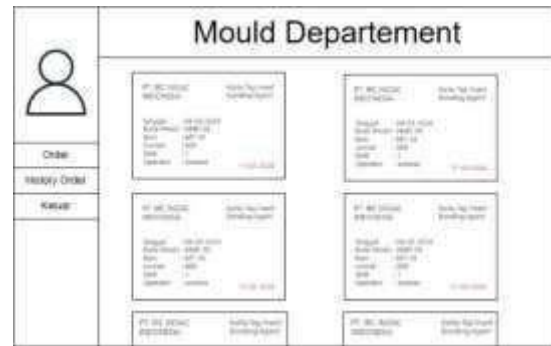
Gambar 4 memperlihatkan hasil *use case diagram* yang telah di buat dan dapat di lihat terdapat 3 aktor di dalam *use case* yaitu produksi *curing mould*, *bonding agent*, dan supervisor. Pada Gambar 2 dapat di lihat bahwa aktor produksi *curing* dapat melakukan *login*, *order*, melihat *order*, melihat *history order*, dan *log out*. Untuk aktor selanjutnya adalah *bonding agesnt* yang dapat melakukan *login*, menyelesaikan *order*, melihat *history* stok, memasukkan laporan, melihat laporan, melihat stok, dan *log out*.

### Desain Website



Gambar 5. Tampilan Desain Login

Gambar 5 menunjukkan *menu login* yang dirancang agar *user* masuk ke dalam akun dengan memasukkan Nomor Induk Pegawai (NIP) dan *password*.



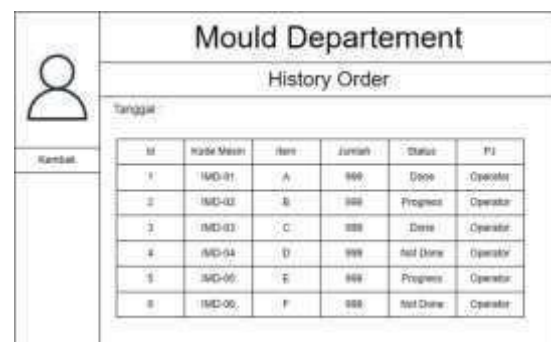
Gambar 6. Tampilan Desain Dashboard Produksi

Gambar 6 menunjukkan rancangan antarmuka untuk *Mould Department* bagian produksi, yang dirancang untuk memudahkan produksi dalam mengelola dan memantau status order *insert*. Antarmuka ini terdiri berbagai fitur yang dapat membantu *user* dalam mengelola dan memonitor pesanan yang masuk.



Gambar 7. Tampilan Desain Form Order

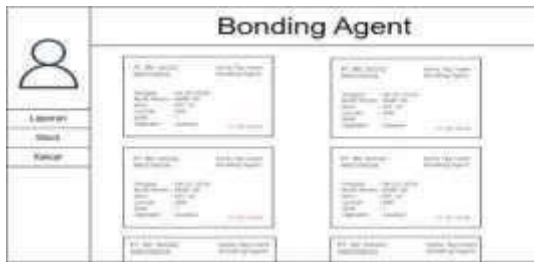
Gambar 7 menunjukkan rancangan antarmuka pengguna untuk memasukkan order *insert* yang dibutuhkan dalam jadwal produksi harian di *Mould Department*, Antarmuka ini dirancang untuk memudahkan proses produksi dalam mengelola dan memantau status order *insert*.



Gambar 8. Tampilan Desain History Order

Gambar 8 menunjukkan rancangan antarmuka pengguna yang dirancang untuk melihat riwayat (*history*) order *insert* yang sudah dilaksanakan oleh bagian produksi pada *Mould Department*. Antarmuka ini menyediakan berbagai informasi

yang memudahkan pengguna dalam mengelola dan memantau status order *insert*.



Gambar 9. Tampilan Dashboard Bonding Agent

Gambar 9 menunjukkan rancangan antarmuka untuk bagian *bonding agent*, yang dirancang untuk memudahkan produksi dalam mengelola status order cetakan.



Gambar 10. Tampilan Desain Stock on Hand

Gambar 10 menunjukkan rancangan antarmuka untuk bagian *bonding agent*, yang dirancang untuk memudahkan dalam memantau *stock on hand* yang terdapat pada *storage area*.



Gambar 11. Tampilan Desain Input Stock

Gambar 11 menunjukkan rancangan antarmuka pengguna untuk memasukkan *stock* yang datang pada hari tersebut, fungsi ini dibuat dengan tujuan agar pengguna dapat melihat aliran barang masuk dan keluar selama proses produksi.



Gambar 12. Tampilan Desain Laporan Harian

Gambar 12 menunjukkan rancangan antarmuka untuk bagian *bonding agent* dengan tujuan untuk memasukkan hasil produksi hari tersebut beserta poin – poin penting dalam pengerjaan produksi.



Gambar 13. Tampilan Desain Dashboard Supervisor

Gambar 13 menunjukkan rancangan antarmuka untuk *Supervisor Mould Department*, yang dirancang untuk memudahkan dalam memantau status order *insert*. Antarmuka ini terdiri berbagai fitur yang dapat membantu *user* dalam memonitor kegiatan proses.



Gambar 14. Tampilan Laporan Hasil Harian

Gambar 14 menunjukkan rancangan antarmuka untuk *Supervisor Mould Department*, yang dirancang untuk memudahkan dalam memantau hasil produksi harian sehingga data dapat dilihat secara *realtime*.

## Hasil Implementasi Website



Gambar 15. Hasil Tampilan *Login*

Gambar 15 menunjukkan halaman *login*, di mana pengguna harus memasukkan Nomor Induk Pegawai (NIP) dan kata sandi. Setelah data diverifikasi, pengguna diarahkan ke halaman sesuai dengan perannya, memastikan bahwa hanya pengguna yang memiliki hak akses yang dapat masuk ke dalam sistem sesuai dengan otoritas mereka.



Gambar 16. Hasil Tampilan *Dashboard* Produksi

Gambar 16 menunjukkan tampilan *dashboard* pada bagian produksi. Di halaman ini, pengguna dapat mengelola dan memasukkan data order mengenai *insert* yang akan diproduksi sesuai kebutuhan. Fitur ini memungkinkan pengguna untuk memperbarui informasi produksi secara *real-time*, memastikan efisiensi dan akurasi dalam proses manufaktur serta memantau status pesanan dengan mudah.



Gambar 17. Hasil Tampilan *Form Order*

Gambar 17 menunjukkan halaman order, di mana pengguna dapat memasukkan data mengenai *insert* yang dibutuhkan pada hari tersebut. Data ini kemudian ditampilkan pada bagian *bonding agent* untuk diproses lebih lanjut. Halaman ini memfasilitasi alur kerja yang efisien, memastikan setiap pesanan *insert* dikelola dengan tepat waktu dan sesuai kebutuhan produksi tanpa harus pengguna meninggalkan daerah kerja mereka.



Gambar 18. Hasil Tampilan *History Order*

Gambar 18 menunjukkan halaman riwayat pemesanan, di mana pengguna dapat melihat riwayat pesanan *insert* mereka. Halaman ini memungkinkan pengguna untuk memeriksa detail pesanan sebelumnya, melacak status, dan memastikan semua pesanan telah diproses dengan benar. Fitur ini membantu dalam memantau dan mengelola data pemesanan secara efisien, memberikan transparansi dan kemudahan dalam akses informasi historis.



Gambar 19. Hasil Tampilan Dashboard Bonding Agent

Gambar 19 menunjukkan halaman *dashboard* bagian *bonding agent*. Di halaman ini, pengguna dapat melihat pesanan *insert* yang harus mereka kerjakan. Fitur ini memudahkan pengguna untuk memantau dan mengelola pesanan secara efisien, memastikan setiap tugas diproses sesuai prioritas dan jadwal yang ditentukan, serta meningkatkan koordinasi dan produktivitas dalam proses produksi.



Gambar 20. Hasil Tampilan Detail Kartu

Gambar 20 menunjukkan halaman detail pesanan. Ketika kartu pesanan ditekan, detail pesanan seperti tanggal dan informasi lainnya akan ditampilkan. Halaman ini memudahkan pengguna untuk mengakses informasi mendalam tentang setiap pesanan, memastikan transparansi dan akurasi dalam proses pengelolaan pesanan, serta membantu dalam memantau status dan rincian pesanan dengan mudah dan efisien.



Gambar 21. Hasil Tampilan Form Hasil Harian

Gambar 21 menunjukkan halaman *input* manual data *chemical lock* (chemlock), di mana pengguna dapat memasukkan hasil kerja mereka secara manual tanpa perlu pesanan dari bagian produksi. Halaman ini dibuat untuk mengatasi pesanan yang tidak sesuai dengan jadwal PPIC, memastikan fleksibilitas dan efisiensi dalam mencatat dan memantau data produksi secara akurat dan independen dari pesanan resmi.



Gambar 22. Hasil Tampilan Laporan Harian

Gambar 22 merupakan tampilan hasil halaman laporan harian dimana pengguna dapat memasukkan hasil produksi yang dilaksanakan pada hari dan *shift* tersebut, data yang dikumpulkan berupa nilai kekentalan (*Viscositas*), berat cairan yang digunakan, suhu pada oven dan lain lain.



Gambar 23. Hasil Tampilan *Stock on Hand*

Gambar 23 menunjukkan halaman *stock on hand*, yang menampilkan data *insert* yang dimiliki oleh bagian *bonding agent*. Halaman ini memungkinkan pengguna untuk memantau ketersediaan stok secara *real-time*, memastikan bahwa bahan yang diperlukan selalu tersedia untuk produksi. Fitur ini membantu dalam pengelolaan inventaris yang efisien, mencegah kekurangan atau kelebihan stok, dan mendukung kelancaran operasi produksi.



Gambar 25. Hasil Tampilan *History Stock*

Gambar 25 menunjukkan halaman *history stock*, di mana pengguna dapat melihat data barang yang datang sesuai tanggal, tipe, dan jumlahnya. Halaman ini memudahkan pelacakan penerimaan stok secara detail, memungkinkan pengguna untuk memverifikasi riwayat masuknya barang, memastikan keakuratan inventaris, dan membantu dalam analisis pola pengiriman serta pengelolaan persediaan yang lebih efektif.



Gambar 24. Hasil Tampilan *Fom Input Stock*

Gambar 24 menunjukkan halaman *input stock*, di mana pengguna dapat memasukkan data barang yang baru datang. Halaman ini memfasilitasi proses pencatatan barang dengan mengisi informasi seperti jenis barang, jumlah, dan tanggal penerimaan. Fitur ini memastikan data inventaris diperbarui secara akurat dan *real-time*, membantu dalam pengelolaan stok yang efisien dan mengurangi kemungkinan kesalahan.



Gambar 26. Hasil Tampilan *Dashboard Supervisor*

Gambar 26 merupakan tampilan hasil halaman *dashboard supervisor*, dimana pengguna dapat memantau kegiatan produksi pada hari tersebut.



Gambar 27. Hasil Tampilan Laporan Harian

Gambar 27 menunjukkan halaman laporan harian, di mana pengguna dapat memantau hasil produksi harian pada hari yang dipilih. Halaman ini memberikan ringkasan kinerja produksi, memungkinkan pengguna untuk melihat detail

produksi harian, mengevaluasi efisiensi, dan mengidentifikasi area yang memerlukan perbaikan. Fitur ini membantu dalam pelacakan dan analisis data produksi secara akurat dan *real-time*.



The screenshot shows a web interface with a 'History' section. It contains a table with columns for 'Tanggal', 'Qty', 'Amp', 'Jumlah', 'Jumlah Bekas', 'Mesin', 'Merkas', and 'Kategori'. The table lists several rows of production data with corresponding values for each column.

Gambar 28. Hasil Tampilan History

ss

#### IV. KESIMPULAN

Pengembangan *website* menggunakan *Unified Modelling Language* (UML) yang melibatkan jenis *Use case diagram* dan *Activity diagram*, dengan *framework* CodeIgniter untuk pengembangan aplikasi dalam PHP dan PostgreSQL sebagai sistem manajemen basis data. Proses penelitian dimulai dari persiapan XAMPP dan CodeIgniter, dilanjutkan dengan setup *database* PostgreSQL, konfigurasi CodeIgniter, pengembangan *website* menggunakan Visual Studio Code, dan pengujian dengan menjalankan server Apache serta melakukan *debugging*. Penerapan sistem order berbasis *website* ini diharapkan dapat meningkatkan efisiensi dan kecepatan proses order. Selain itu, *monitoring* sistem dapat dilakukan dengan lebih baik, membantu dalam manajemen dan pengawasan proses order secara lebih efektif dan *real-time*.

#### DAFTAR PUSTAKA

- [1] A.-A. Solehudin, N. Fariz, N. Wahyu, R. F. Permana, and A. Saifudin, "Rancang Bangun Digitalisasi Persediaan Barang Berbasis Web Menggunakan Metode Waterfall," *Log. J. Ilmu Komput. dan Pendidik.*, vol. 1, no. 4, pp. 1000–1005, 2023, [Online]. Available: <https://journal.mediapublikasi.id/index.php/logic/article/view/3163>
- [2] F. Amelia Sari Lubis, S. Sahara Lubis, and B. Hendrik, "Perancangan Sistem

Inventory Untuk Stok Barang Herbisida Pada UD. Anugrah Jaya Tani Dengan Bahasa Pemrograman PHP dan Database MySql," *J. Sains Inform. Terap.*, vol. 2, no. 2, pp. 50–55, 2023, doi: 10.62357/jsit.v2i2.167.

- [3] D. Nufriana, "Jurnal Ilmiah INTECH: Information Technology Journal of UMUS," *Umus*, vol. 1, no. 02, pp. 1–12, 2019.
- [4] Y. Blanco-Jimenez, L. K. Ahumada-Torres, J. R. Castro-Suarez, and M. A. Chico-Ruiz, "Development of a web system for the management of PQRS in higher education institutions," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 844, no. 1, 2020, doi: 10.1088/1757-899X/844/1/012068.
- [5] A. C. Praniffa, A. Syahri, F. Sandes, U. Fariha, Q. A. Giansyah, and M. L. Hamzah, "Penguji Black Box Dan White Box Sistem Informasi Parkir Berbasis Web Black Box and White Box Testing of Web-Based Parking Information System," *J. Test. dan Implementasi Sist. Inf.*, vol. 1, no. 1, pp. 1–16, 2023.
- [6] M. Manuhutu and J. Wattimena, "Perancangan Sistem Informasi Konsultasi Akademik Berbasis Website," *J. Sist. Inf. Bisnis*, vol. 9, no. 2, p. 149, 2019, doi: 10.21456/vol9iss2pp149-156.