

PENERAPAN WATERFALL DALAM PENGEMBANGAN WEB INVENTORI EDC DENGAN SPRING BOOT DAN THYMELEAF

Nadira Adiva Wibowo^{1*}, Moh. Idris²

^{1,2}Universitas Islam Indonesia, Indonesia

*Corresponding author: 20523202@students.uii.ac.id

Abstract: Digital payments assisted by EDC or QREDC are a concrete example of financial technology that provides speed and practicality in payment activities. One of the scopes of EDC is inventory management. If management finds it difficult to deal with the ever-increasing number of EDCs to be processed and the need for accurate and up-to-date data, it is necessary to develop an EDC inventory web system. The system aims to control and supervise all EDC processes including procurement, processing, inventory, distribution, and return, with precise and up-to-date data. The system was also developed using the waterfall method, back-end technology (the Java programming language, Spring Boot framework, and PostgreSQL database), as well as front-end technology (Thymeleaf templates, jQuery libraries, and AJAX). The waterfall method and all these technologies make code work easy and fast by knowing comprehensive and detailed system requirements. As a result, the EDC inventory web system was successfully developed and operates well and precisely according to its purpose. This is demonstrated by the existence of two features (product repair and product repair items) which have succeeded in achieving the goal of returning them if there is a damaged EDC in the inventory.

Keywords: EDC, Inventory, Waterfall, Spring Boot, Thymeleaf.

Abstrak: Pembayaran digital yang dibantu dengan EDC atau QREDC adalah contoh wujud nyata dari *financial technology* yang menyediakan kecepatan dan kepraktisan dalam aktivitas pembayaran. Salah satu ruang lingkup EDC adalah manajemen inventornya. Apabila manajemennya merasakan kesulitan dalam menghadapi banyaknya EDC yang terus meningkat untuk diproses dan keperluan data yang tepat serta terkini, dibutuhkan pengembangan sistem web inventori EDC. Sistem tersebut bertujuan untuk mengontrol dan mengawasi semua proses EDC yang mencakup pengadaan, pemrosesan, persediaan, pendistribusian, dan pengembalian, dengan data yang tepat dan terkini. Sistem juga dikembangkan dengan metode *waterfall*, teknologi *back-end* (bahasa pemrograman Java, *framework* Spring Boot, dan *database* PostgreSQL), serta teknologi *front-end* (*template* Thymeleaf, *library* jQuery, dan AJAX). Metode *waterfall* dan semua teknologi tersebut membuat pengerjaan kode mudah dan cepat dengan mengetahui kebutuhan sistem yang komprehensif dan detail. Hasilnya, sistem web inventori EDC berhasil dikembangkan dan beroperasi dengan baik dan tepat sesuai tujuan. Hal tersebut ditunjukkan dengan adanya dua fitur (*product repair* dan *product repair item*) yang berhasil mencapai tujuan berupa pengembalian jika ada EDC yang rusak pada inventori.

Kata kunci: EDC, Inventori, *Waterfall*, *Spring Boot*, *Thymeleaf*.

Copyright (c) 2025 The Authors. This is an open-access article under the CC BY-SA 4.0 license (<https://creativecommons.org/licenses/by-sa/4.0/>)

PENDAHULUAN

Kemajuan teknologi yang berkembang dengan sangat cepat sudah memberikan dampak pada *lifestyle* manusia, termasuk memberikan modifikasi bagi manusia dalam

mengerjakan aktivitas. Salah satu contohnya adalah *fintech* atau *financial technology* yang berkonsentrasi pada pengadaan solusi keuangan yang menggunakan perangkat lunak modern sehingga dapat memudahkan keperluan aktivitas keuangan (Nafiah dan Faih, 2019). Dengan begitu, sekarang manusia lebih memilih aktivitas yang cepat dan praktis. Terdapat berbagai perusahaan di Indonesia yang bergerak di industri bidang *fintech*, salah satunya adalah PT XYZ.

PT XYZ sudah meluncurkan salah satu produk *fintech* berkualitas, yaitu QREDC (*Quick Response Electronic Data Capture*). QREDC merupakan gabungan dari EDC yang menggunakan QR berupa QRIS. EDC (*Electronic Data Capture*) merupakan alat elektronik yang membantu penjual menerima pembayaran elektronik dari pembeli dengan cepat dan praktis (Thoyyib dan Zulfikar, 2023), sedangkan QRIS (*Quick Response Code Indonesian Standard*) merupakan standar kode QR di Indonesia yang menyederhanakan pembayaran dengan alat elektronik (Thoyyib dan Zulfikar, 2023). Maka dari itu, QREDC merupakan pembayaran elektronik dari pembeli ke penjual dengan melakukan *scanning* ke QRIS yang dimunculkan melalui EDC. *Scanning* bisa melalui *e-wallet* atau *mobile banking*. Pembayaran tersebut juga secara tidak langsung sudah menggantikan cara pembayaran tradisional yang memakai uang tunai dan menawarkan kemudahan dan kecepatan karena uang digital dapat digunakan langsung melalui internet. Hal tersebut membuat perbedaan dengan pembayaran tradisional yang sebelumnya harus menarik uang tunai di ATM atau bank.

PT XYZ juga dapat membuat suatu produk yang masih berhubungan dengan *fintech* sebagai jasa untuk pihak lain atau klien mereka. Salah satunya adalah inventori EDC yang dimiliki oleh salah satu klien. Inventori atau persediaan merupakan sekumpulan barang yang diadakan untuk mencukupi kebutuhan pelanggan (Muhardi, 2019). Inventori tersebut memiliki dua kesulitan di dalamnya. Kesulitan pertama, yaitu banyaknya EDC yang terus meningkat untuk diproses. Peningkatan tersebut membuat kekacauan bagi klien karena tidak bisa membedakan EDC yang belum diproses dan yang sudah diproses. Terlebih lagi jika ada EDC yang rusak saat diproses sehingga membuat EDC tersebut harus dikirim lagi untuk dibetulkan. Kesulitan kedua, yaitu keperluan data yang tepat dan terkini.

Oleh karena itu, PT XYZ akhirnya mengembangkan sistem web inventori EDC. Sistem merupakan rangkaian komponen atau elemen yang saling berhubungan dan

berkolaborasi untuk memperoleh tujuan yang telah ditetapkan (Anjarwati et al., 2021). Web merupakan himpunan informasi yang disampaikan dalam bentuk halaman dan dapat dijangkau melalui internet (Romadhon et al., 2021). Sistem inventori merupakan sistem untuk menyimpan berbagai data persediaan barang ke *database* yang dapat diakses dan dipakai saat diperlukan sehingga mempunyai input dan *output* data yang jelas dan tepat (Gosal dan Rustam, 2022). Sistem web inventori EDC ini mempunyai satu tujuan utama yang mencakup kebutuhan, yaitu untuk mengontrol dan mengawasi semua proses EDC yang mencakup pengadaan, pemrosesan, persediaan, pendistribusian, dan pengembalian, dengan data yang tepat dan terkini. Tujuan tersebut akan menyelesaikan dua kesulitan tersebut.

Sistem web inventori EDC ini dikembangkan dengan teknologi bahasa pemrograman Java, *framework* Spring Boot, dan *database PostgreSQL* sebagai bagian dari *back-end*. *Back-end* merupakan kegiatan yang berkonsentrasi pada pemrosesan *database* dan struktur web sehingga dapat menyampaikan informasi dari *database* ke web (Sarmanela et al., 2023). Selain itu, dikembangkan juga dengan gabungan dari *template* web desain Metronic (HTML, CSS, *JavaScript*, *framework Bootstrap*), *template Thymeleaf*, *library* jQuery, dan AJAX sebagai bagian dari *front-end*. *Front-end* merupakan tampilan yang menyambungkan interaksi *user* dengan *back-end* memakai HTML dan CSS (Putri dan Sulianta, 2023).

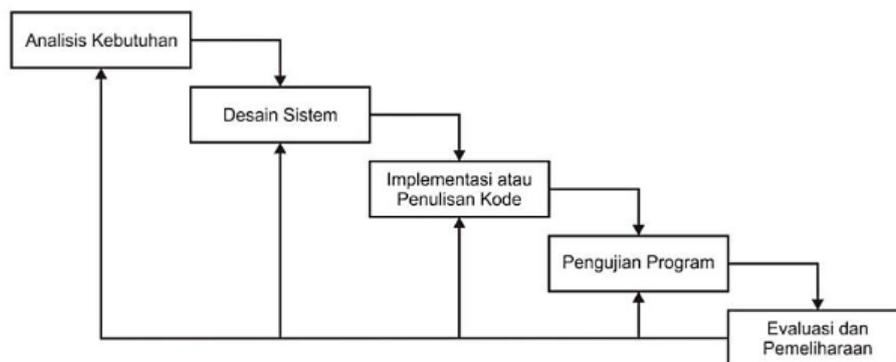
Bahasa pemrograman Java dipilih karena didukung dengan adanya *framework* Spring Boot. *Framework* merupakan kerangka kerja yang menyajikan kode-kode berupa *class* dan *function* yang siap dipakai secara berkali-kali oleh *programmer* sehingga mempercepat pengembangan (Yudhanto dan Prasetyo, 2019). Sementara itu, Spring Boot merupakan bagian terpadu dari *framework* Spring yang perlu dipakai dalam pengembangan aplikasi Spring karena akan menyederhanakan proses pengembangan (Khaqiqi dan Harani, 2023). Hal tersebut menjadi alasan mengapa Spring Boot dipilih dalam sistem web inventori EDC. Selain itu, Spring Boot memiliki ukuran yang ringan, lisensi *open source*, dan menyediakan berbagai dependensi (*Spring MVC*, *Spring Data*, Lombok, dkk) yang membantu proses pengembangan (Hartono dan Mailoa, 2024). *Database PostgreSQL* dipilih karena bersifat *open source* dan *free* yang memakai dan memperluas bahasa SQL yang bisa digabungkan dengan banyak fitur. Selain itu, bisa

dioperasikan di berbagai sistem operasi, seperti *Windows*, *Linux*, *macOS*, *Solaris*, dan *BSD* (Postgresql, 2024).

Template Thymeleaf dipilih karena *template* tersebut merupakan *serverside* Java *template engine* dan telah terpadu dengan *framework* *Spring Boot* sehingga memang ditujukan untuk mengembangkan aplikasi berbasis web yang menggunakan *Spring Boot* (Dagha dan Susetyo, 2021). *Library* jQuery dipilih karena menggabungkan interaksi antara JavaScript dan HTML dengan menyajikan berbagai kode yang sederhana sehingga mudah dimengerti (Sahi, 2020). AJAX dipilih karena pembaruan data dari *server* sebagai *response* atas *request* dari *client* dapat dipaparkan tanpa perlu pemuatan ulang pada semua halaman web (Lewenusa, 2020).

METODE

Metode untuk mengembangkan sistem web inventori EDC menggunakan *waterfall* yang dikenal sebagai salah satu model SDLC yang seringkali dipakai. SDLC atau *System Development Life Cycle* merupakan metode umum yang dipakai untuk mengembangkan sistem informasi. Metode *waterfall* mengilustrasikan pendekatan yang sistematis, berkesinambungan, dan bersifat linear dari tahapan awal sampai akhir. Tahapan berikutnya hanya boleh dilanjutkan setelah tahapan sebelumnya selesai, namun tahapan yang telah dilalui tidak dapat berbalik (Wahid, 2020). Maka dari itu, metode ini tepat untuk kebutuhan yang sudah teridentifikasi dan mempunyai sedikit potensi perubahan selama pengembangannya (Solehudin et al., 2023). Selain itu, diharapkan sistem yang dikembangkan menggunakan metode ini dapat beroperasi dengan baik, tepat dengan kebutuhan, dan berpengaruh baik sesuai lingkungannya (Fatman et al., 2023). Ada lima tahapan dalam metode *waterfall* yang ditampilkan pada gambar di bawah ini:



Gambar 1. Tahapan Metode *Waterfall*

(Sumber: Hermansyah et al., 2023)

1. Analisis Kebutuhan

Tahapan ini melibatkan pengumpulan informasi secara menyeluruh mengenai kebutuhan sistem, termasuk fitur-fitur detail yang diinginkan oleh *user* atau klien dan pembatasan sistemnya. Informasi biasanya diperoleh melalui wawancara, observasi, diskusi, atau dokumen dan dikaji untuk memastikan bahwa data sistem sudah tepat dengan kebutuhan (Vicky dan Syaripudin, 2022).

2. Desain Sistem

Tahapan ini melibatkan perancangan desain sistem yang mencakup *use case diagram*, *activity diagram*, *class diagram*, hingga *user interface*. Semua desain sistem tersebut akan memberikan pengetahuan kebutuhan sistem yang komprehensif dan detail sehingga mempercepat penulisan kode program.

3. Implementasi atau Penulisan Kode

Tahapan ini melibatkan penulisan kode program supaya pengembangan sistem bisa terwujud. Kode program menggunakan bahasa pemrograman Java, *framework Spring Boot*, dan *database PostgreSQL* sebagai bagian dari *back-end*. Selain itu, juga menggunakan *template* web desain Metronic, *template Thymeleaf*, *library jQuery*, dan *AJAX* sebagai bagian dari *front-end*. Implementasinya akan menggabungkan beragam fitur, dimana setiap fitur mempunyai fungsi-fungsi tersendiri yang lebih detail. Hal tersebut tepat dengan kebutuhan yang sudah diidentifikasi pada tahapan analisis kebutuhan dan desain sistem.

4. Pengujian Program

Tahapan ini melibatkan pengujian menyeluruh terhadap pengerjaan sistem yang sudah dikembangkan. Pengujian bertujuan untuk memerhatikan dan memverifikasi bahwa semua fitur sistem berfungsi secara optimal dan tepat dengan kebutuhan. Akan tetapi, pengujian belum dikerjakan dengan suatu metode tertentu karena pengujian masih dikerjakan dalam ruang lingkup *developer* saja.

5. Evaluasi dan Pemeliharaan

Tahapan ini melibatkan pemeliharaan sistem untuk menjaga kinerjanya tetap optimal. Juga mengerjakan pembetulan kepada sistem saat menemukan kekeliruan atau ketidakselarasan selama proses pemeliharaan untuk memastikan bahwa sistem dapat beroperasi dengan optimal lagi.

HASIL DAN PEMBAHASAN

Hasil

Berdasarkan pengembangan sistem web inventori EDC yang dikembangkan dengan metode *waterfall* dan berbagai teknologi mulai dari *framework Spring Boot* hingga AJAX, hasil akhirnya berupa berbagai tampilan halaman web pada sistem yang mengoperasikan kebutuhan melalui fitur-fitur di dalamnya. Berikut adalah hasil yang diperoleh selama proses pengembangan dengan metode *waterfall*:

1. Analisis Kebutuhan

Dari dokumen yang diberikan oleh klien, diperoleh informasi tentang kebutuhan sistem yang mendetail. Dokumen ini mencakup elemen-elemen utama, seperti fitur-fitur, ruang lingkup, dan batasan. Satu dari fitur utama adalah pengembalian EDC yang rusak. Fitur ini mencakup dua fitur lagi, yaitu pengelolaan suatu pengembalian (*product repair*) terkait beberapa EDC yang rusak saat diproses dan EDC yang rusak (*product repair item*) tersebut akan dihimpun sesuai jumlah tertentu untuk dikembalikan. *User* yang dapat berhubungan dengan dua fitur tersebut adalah klien *maker* dan *vendor arrival checker*.

2. Desain Sistem

User klien *maker* yang disebutkan dalam tahapan sebelumnya mempunyai hubungan rinci dengan dua fitur tersebut. Pada fitur *product repair*, yaitu membuat pengembalian, melihat semua pengembalian yang difasilitasi dengan filter, melihat pengembalian berdasarkan *id*, mengubah pengembalian berdasarkan *id*, menghapus pengembalian berdasarkan *id*, dan mengerjakan *request delivery*. Pada fitur *product repair item*, yaitu membuat EDC yang rusak dan melihat semua EDC yang rusak yang difasilitasi dengan filter.

Untuk *user* *vendor arrival checker* yang disebutkan dalam tahapan sebelumnya juga mempunyai hubungan rinci dengan dua fitur tersebut. Pada fitur *product repair*, yaitu melihat semua pengembalian yang hanya diperuntukkan kepada mereka dan difasilitasi dengan filter, melihat pengembalian tersebut berdasarkan *id*, mengerjakan *reject request delivery*, mengerjakan *approve request delivery*, dan mengerjakan *received delivery*. Pada fitur *product repair item*, yaitu melihat semua EDC yang rusak yang hanya diperuntukkan kepada mereka dan difasilitasi dengan filter.

Terdapat juga alur atau *flow* mengenai proses dua fitur tersebut dimulai dari adanya EDC yang rusak saat diproses oleh klien sehingga klien *maker* menambahkan

EDC tersebut ke *product repair item*. Setiap EDC mempunyai informasi tentang vendor *arrival* yang mengirimkannya. Penambahan tersebut akan memberikan dampak pada dua hal. Pertama, pengembalian baru tidak dibuat jika sudah ada pengembalian yang diperuntukkan pada vendor *arrival* yang sama dengan vendor *arrival* yang dipunyai EDC dan statusnya adalah *packing*. Akibatnya EDC yang rusak dihimpun pada pengembalian tersebut. Kedua, pengembalian baru dibuat jika tidak ada pengembalian yang diperuntukkan pada vendor *arrival* yang dipunyai EDC atau jika ada pengembalian yang diperuntukkan pada vendor *arrival* yang sama dengan vendor *arrival* yang dipunyai EDC dan statusnya tidak *packing*.

Kemudian, klien *maker* menentukan atau mengubah tanggal pengiriman pada suatu pengembalian dan mengerjakan *request delivery*. Vendor *arrival checker* lalu memverifikasi apakah permintaan pengembalian tersebut disetujui atau tidak disetujui. Jika disetujui, klien *maker* langsung mengirimkan pengembalian dan vendor *arrival checker* akan menerima pengembalian ketika sudah sampai. Jika tidak disetujui, klien *maker* menentukan lagi tanggal pengirimannya dan mengerjakan *request delivery* lagi. EDC yang rusak nantinya lekas dibetulkan oleh vendor *arrival* ketika sudah diterima dan vendor *arrival* mengirimkan lagi EDC yang sudah dibetulkan ke klien supaya diproses lagi sehingga status EDC yang rusak berubah menjadi *done*.

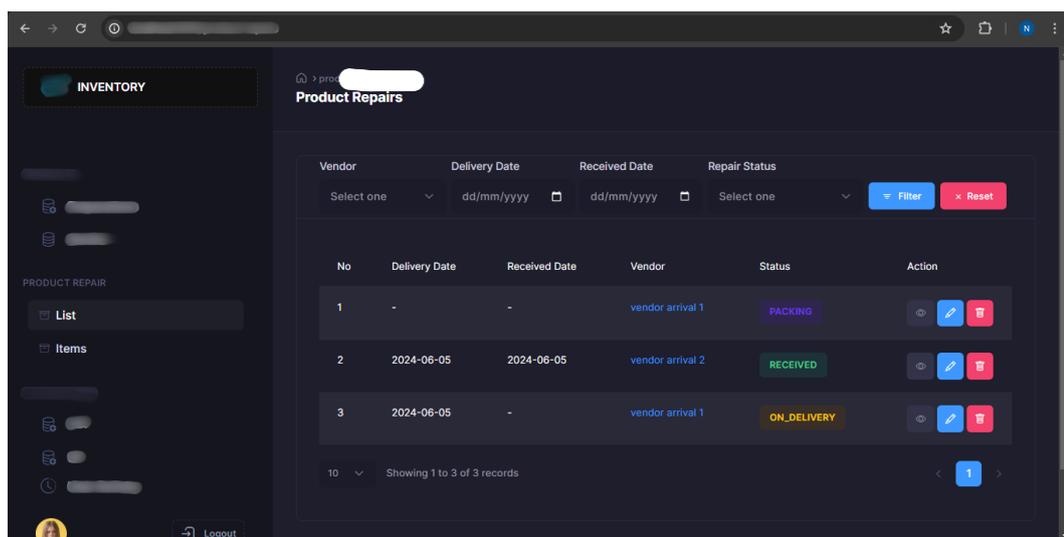
Terakhir, dua fitur tersebut mempunyai desain *database* masing-masing. Untuk fitur *product repair*, atribut-atributnya mencakup *id* sebagai *primary key*, *delivery date*, *received date*, *vendor id*, dan status. Untuk fitur *product repair item*, atribut-atributnya mencakup *id* sebagai *primary key*, *EDC id*, *serial number*, *remark*, *product repair id* sebagai *foreign key*, dan status. *Product repair* berhubungan *one to many* dengan *product repair item* sehingga setiap pengembalian dapat mencakup satu atau lebih EDC yang rusak. Hal tersebut ditunjukkan dengan *foreign key* pada *product repair item* yang mengacu dengan *primary key* pada *product repair*.

3. Implementasi atau Penulisan Kode

Penulisan kode program menggunakan teknologi bahasa pemrograman Java, *framework Spring Boot*, dan *database PostgreSQL* sebagai bagian dari *back-end* serta gabungan dari template web desain Metronic (HTML, CSS, *JavaScript*, *framework Bootstrap*), *template Thymeleaf*, *library jQuery*, dan AJAX sebagai bagian dari *front-*

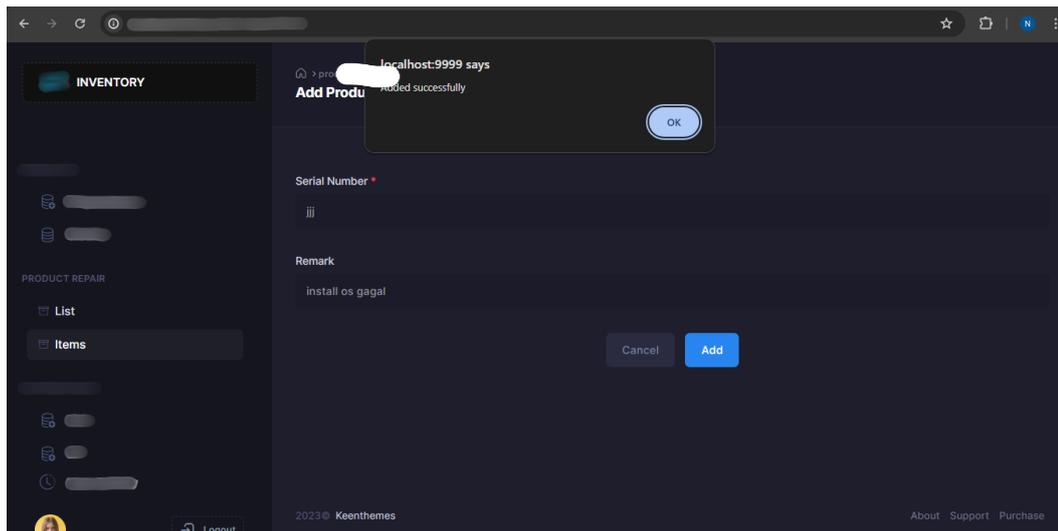
end. Berikut beberapa tampilan halaman pada fitur *product repair* dan *product repair item* sebagai hasil dari penulisan kode program:

Pada Gambar 2 adalah halaman *list* yang memperlihatkan semua pengembalian ke semua vendor *arrival* dengan model tabel karena *user* yang *login* adalah klien *maker*. Tabel dibuat dengan bantuan DataTables (*library* jQuery) dan mempunyai beberapa kolom, yaitu *delivery date*, *received date*, vendor *arrival* tujuan, status, dan *action* yang mempunyai tiga tombol untuk melihat pengembalian berdasarkan *id*, mengubah pengembalian berdasarkan *id*, dan menghapus pengembalian berdasarkan *id*.



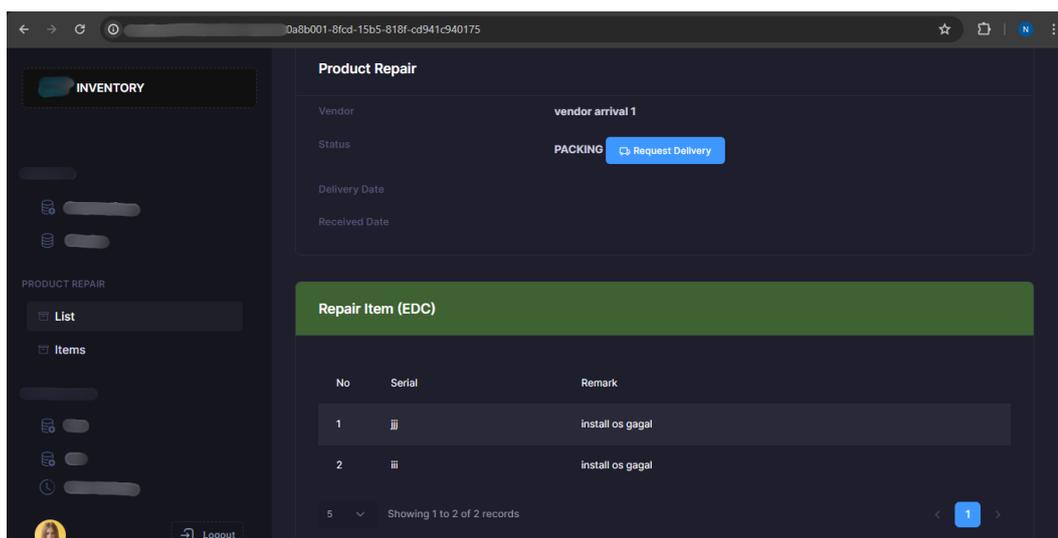
Gambar 2. Tampilan Halaman List pada Product Repair

Pada Gambar 3 adalah halaman *add-form* yang memperlihatkan kesuksesan penambahan EDC yang rusak dengan *serial number* *jjj* dan *remark* *install os* gagal oleh *user* klien *maker*. Hasil penambahan tersebut tidak membuat pengembalian baru karena vendor *arrival* yang dipunyai oleh EDC tersebut adalah vendor *arrival* 1 yang sama dengan vendor yang sudah ada pada pengembalian dan statusnya adalah *packing*. Dengan demikian, pada halaman *list* mempunyai tampilan pengembalian yang masih sama dengan Gambar 2.



Gambar 3. Tampilan Halaman *Add-Form* pada *Product Repair Item*

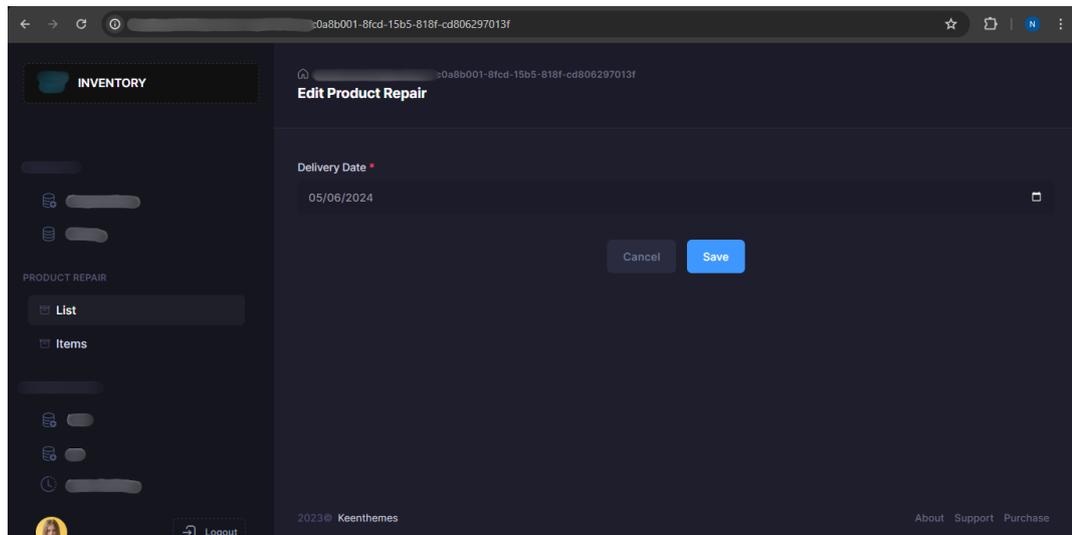
Pada Gambar 4 adalah halaman *view* yang memperlihatkan suatu pengembalian berdasarkan *id*. Selain itu, dapat mengerjakan *request delivery* pada tombol *request delivery* dan memperlihatkan semua EDC yang rusak pada pengembalian tersebut dengan model tabel. Terdapat dua EDC yang rusak yang mempunyai *serial number* masing-masing, yaitu *serial number* iii yang sebelumnya sudah ditambahkan sebelumnya dan *serial number* jii yang terakhir ditambahkan pada Gambar 4.



Gambar 4. Tampilan Halaman *View* pada *Product Repair*

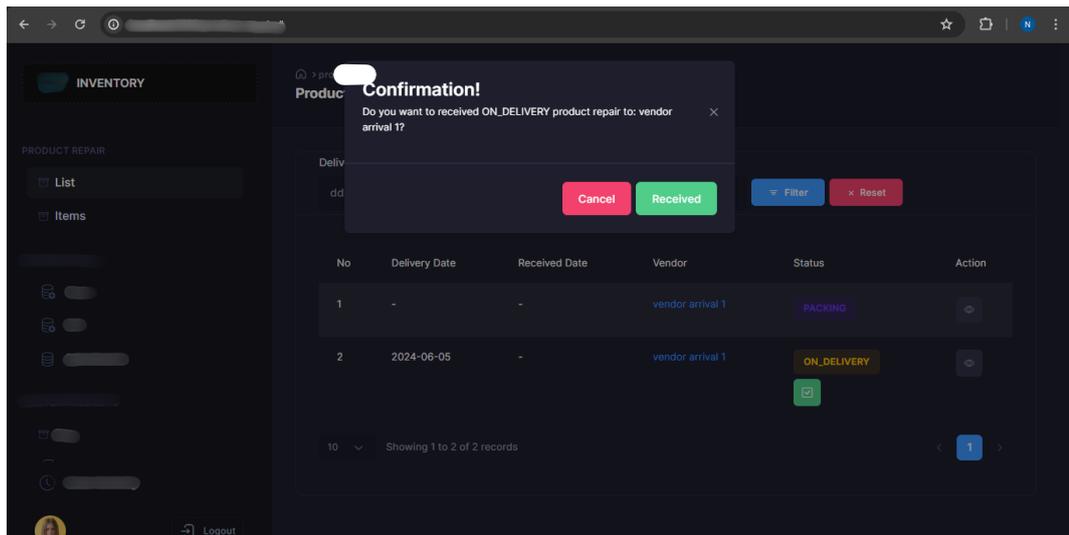
Pada Gambar 5 adalah halaman *edit-form* memperlihatkan suatu pengembalian berdasarkan *id* dan mengubah pengembalian tersebut. Atribut pada pengembalian yang dapat diubah adalah tanggal pengiriman. Jika tanggal pengiriman sudah ditentukan atau diubah, maka tombol *request delivery* pada Gambar 4 dapat ditekan.

Pengembalian yang ditampilkan pada Gambar 5 adalah pengembalian ke vendor *arrival 1* yang tanggal pengirimannya sudah diatur pada 5 Juni 2024 dan sekarang statusnya adalah *on delivery*.



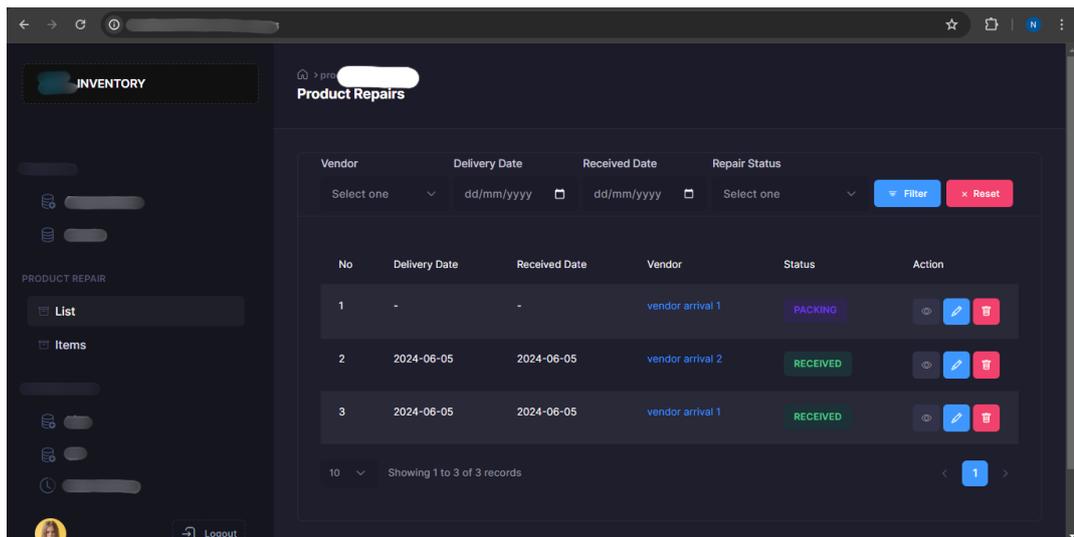
Gambar 5. Tampilan Halaman *Update* pada *Product Repair*

Pada Gambar 6 adalah halaman *list* yang memperlihatkan semua pengembalian ke vendor *arrival 1* saja dengan model tabel karena *user* yang *login* adalah vendor *arrival 1 checker*. Kolom *action* pada tabel hanya mempunyai satu tombol untuk melihat pengembalian berdasarkan *id*. Saat *user vendor arrival 1 checker* menekan tombol centang pada pengembalian yang statusnya adalah *on delivery*, sebuah *pop up modal* akan muncul. *Pop up modal* tersebut untuk memverifikasi bahwa pengembalian dengan tanggal pengiriman 5 Juni 2024 telah datang dan akan diterima. Tanggal penerimaan akan disesuaikan secara otomatis berdasarkan zona waktu *default*.



Gambar 6. Tampilan Halaman *List* pada *Product Repair* (2)

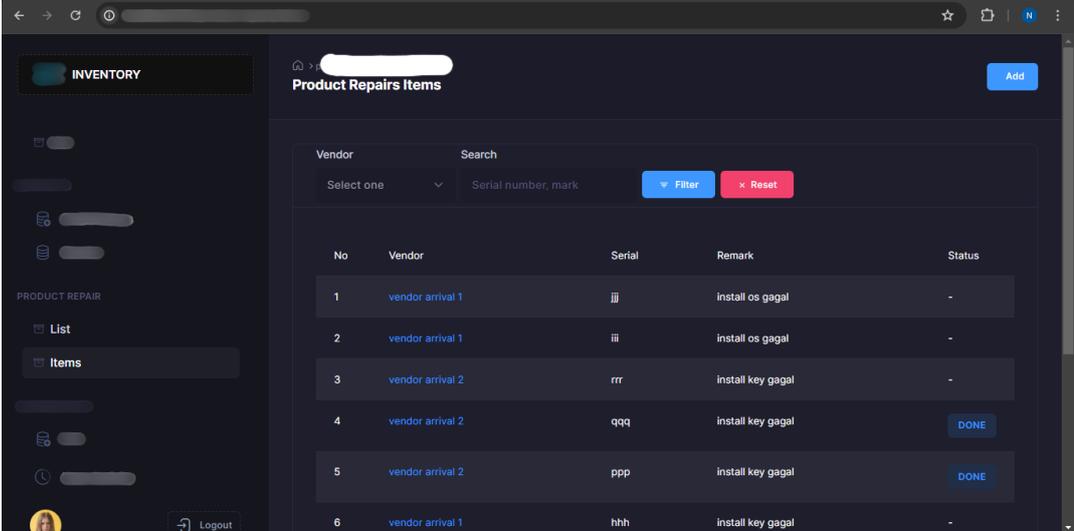
Pada Gambar 7 adalah halaman dengan tampilan yang sama seperti Gambar 2 dan diperoleh sehabis proses *checker* (konfirmasi) pada Gambar 6 dikerjakan. Akan tetapi, mempunyai perbedaan yaitu terdapat pengembalian yang statusnya adalah *received* dan tanggal penerimaan sudah berjumlah dua. Hal tersebut menandakan bahwa penerimaan pengembalian sukses dikerjakan.



Gambar 7. Tampilan Halaman *List* pada *Product Repair* (3)

Pada Gambar 8 **Error! Reference source not found.** adalah halaman *list-item* yang memperlihatkan semua EDC yang rusak ke semua vendor *arrival* dengan model tabel karena *user* yang *login* adalah klien *maker*. Terdapat beberapa kolom, yaitu *vendor*, *serial number*, *remark*, dan status. Status berupa *done* berarti EDC yang rusak

sudah dibetulkan oleh vendor *arrival* yang terkait dan dikirimkan lagi ke klien untuk diproses lagi.



No	Vendor	Serial	Remark	Status
1	vendor arrival 1	jjj	install os gagal	-
2	vendor arrival 1	iii	install os gagal	-
3	vendor arrival 2	rrr	install key gagal	-
4	vendor arrival 2	qqq	install key gagal	DONE
5	vendor arrival 2	ppp	install key gagal	DONE
6	vendor arrival 1	hhh	install key gagal	-

Gambar 8. Tampilan Halaman *List Item* pada *Product Repair Item*

4. Pengujian Program

Pengujian belum dikerjakan dengan suatu metode tertentu, seperti *black-box*. Saat ini pengujian masih dikerjakan dalam ruang lingkup *developer* untuk menguji bahwa setiap fitur yang sudah dikembangkan dapat beroperasi dengan baik dan tepat dengan kebutuhan. Dengan demikian, pengembangan sistem belum sampai pada tahap terakhir, yaitu evaluasi dan pemeliharaan.

Pembahasan

Secara umum, hasil pengembangan sistem web inventori EDC dengan metode pengembangan *waterfall* sudah dapat beroperasi dengan baik dan tepat dengan kebutuhan fitur-fitur yang telah diperoleh. Keberhasilan pengembangan tersebut tepat dengan teori metode *waterfall*, dimana kebutuhan telah diketahui dengan komprehensif dan detail pada tahap analisis kebutuhan dan desain sistem. Walaupun begitu, terdapat ketidaktepatan dengan teori metode *waterfall* berupa adanya penambahan kebutuhan dan alur sistem saat sudah berada di tahap implementasi sehingga harus balik ke tahap analisis kebutuhan dan desain sistem. Padahal tahapan yang telah dilewati pada teori metode *waterfall* seharusnya tidak dapat diulang.

Selanjutnya, teknologi yang digunakan pada *back-end* dan *front-end* tepat dengan teori yang telah dijelaskan. Dalam penggunaan *framework Spring Boot*, pengerjaan kode program mudah dan cepat dikerjakan berkat berbagai kelebihan yang dimiliki *Spring*

Boot, seperti *DI*, *Spring MVC*, *Spring Rest*, dan *Spring Data*. Dalam penggunaan *database PostgreSQL*, akses dan manipulasi data mudah dikerjakan dengan bahasa SQL yang dipunyai *PostgreSQL* serta dapat beroperasi dengan lancar di *Windows*. Dalam penggunaan *template Thymeleaf*, tampilan halaman web menjadi dinamis dengan perpaduan *Spring Boot* yang dipunyai oleh *Thymeleaf*. Dalam penggunaan *library jQuery*, pengerjaan kode program juga mudah dan cepat dikerjakan berkat kode-kode *JavaScript* yang siap dipakai. Dalam penggunaan *AJAX*, *user* tidak merasa terganggu saat menggunakan halaman web karena tidak butuh pemuatan ulang pada semua halaman web.

Terdapat juga persamaan dan perbedaan dari dua hasil penelitian terdahulu yang mirip atau hampir sama dengan hasil pengembangan sistem web inventori EDC ini. Penelitian pertama diambil dari judul “Pengembangan Sistem *Point of Sale* Berbasis Web (Studi Kasus: Bos Duren Malang)”. Penelitian tersebut membuat sistem POS berbasis web yang sudah mengerjakan pengujian menggunakan tiga metode pengujian, salah satunya dengan *black-box*, dan memperoleh hasil yang valid sesuai harapan (Dirosa dan Kurniawan, 2021). Penelitian kedua diambil dari judul “Sistem Informasi *Inventory* pada PT Djaya Buah Bersinar Denpasar Berbasis Web”. Penelitian tersebut membuat sistem *inventory* buah berbasis web yang sudah mengerjakan pengujian menggunakan metode *black-box* dan memperoleh hasil yang valid sesuai harapan (Desmayani et al., 2022).

Persamaan dari penelitian pertama, yaitu sukses mengembangkan sistem berbasis web dengan metode *waterfall* dan teknologi berupa bahasa pemrograman Java serta *framework* Spring Boot. Persamaan dari penelitian kedua, yaitu sukses mengembangkan sistem inventori berbasis web dengan metode *waterfall* yang bertujuan mengelola persediaan barang di toko atau perusahaan. Perbedaan dari penelitian pertama, yaitu sistem yang dikembangkan terkait *point of sale*, bukan persediaan barang. Perbedaan dari penelitian kedua, yaitu teknologi pada sistem yang digunakan adalah bahasa pemrograman PHP dan *database MySQL*, bukan bahasa pemrograman Java dan *database PostgreSQL*.

Dua penelitian tersebut juga mempunyai persamaan, yaitu mengerjakan pengujian pada hasil pengembangan sistem dengan metode *black-box* dan memperoleh hasil yang sukses atau valid sesuai harapan. Akan tetapi, hal tersebut berbeda dengan pengembangan sistem web inventori EDC yang pengujiannya belum dikerjakan oleh *quality engineer*

atau pihak yang bertanggung jawab dengan suatu metode pengujian, seperti *black-box*. Pengujian saat ini masih dikerjakan dalam ruang lingkup *developer* untuk menguji bahwa setiap fitur yang sudah dikembangkan dapat beroperasi dengan baik dan tepat dengan kebutuhan.

SIMPULAN

Berdasarkan penelitian ini, dapat disimpulkan bahwa sistem web inventori EDC dengan metode *waterfall* dan teknologi *back-end* (bahasa pemrograman Java, *framework* Spring Boot, *database* PostgreSQL) serta *front-end* (*template* Thymeleaf, *library* jQuery, AJAX) telah berhasil beroperasi dengan baik dan tepat sesuai tujuan yang mencakup kebutuhan. Hal tersebut ditunjukkan dengan adanya kesesuaian antara fitur yang berhasil dioperasikan dengan salah satu tujuan sistem, yaitu dua fitur berupa *product repair* dan *product repair item* yang berhasil mencapai tujuan pengembalian jika ada EDC yang rusak pada inventori.

Lebih detailnya, sistem web inventori EDC ini dibangun karena adanya kesulitan yang dipunyai oleh klien, yaitu banyaknya EDC yang terus meningkat untuk diproses dan keperluan data yang tepat serta terkini. Hal tersebut dapat teratasi dengan tujuan utama yang diharapkan pada pengembangan sistem web inventori EDC. Tujuan utamanya adalah mengontrol dan mengawasi semua proses pada EDC yang mencakup pengadaan, pemrosesan, persediaan, pendistribusian, dan pengembalian, dengan data yang tepat dan terkini. Hasil pengembangan sistem ini berupa berbagai tampilan halaman web yang mengoperasikan kebutuhan melalui fitur-fitur di dalamnya. Dari semua fiturnya yang berhasil dikembangkan dan dioperasikan, terdapat dua fitur yang dijelaskan dalam jurnal ini, yaitu *product repair* (pengembalian) dan *product repair item* (EDC yang rusak). Dua fitur tersebut menjadi konkret nyata bahwa salah satu tujuan, yaitu pengembalian jika ada EDC yang rusak pada inventori, berhasil terwujudkan dan beroperasi dengan baik. Jurnal ini juga berfungsi sebagai landasan, rujukan, panduan, atau model praktis bagi pembaca yang berminat dalam mengembangkan sistem web inventori suatu barang.

DAFTAR RUJUKAN

- Anjarwati, F., Febriyanti, O., Muthmainah, & Sufyana, C. M. (2021). Sistem Informasi Akuntansi Penjualan Makanan Ringan Berbasis Web Di Krenies INDONESIA. *Fair Value: Jurnal Ilmiah Akuntansi Dan Keuangan*, 4(Spesial Issue 1), 90-101. doi:<https://doi.org/10.32670/fairvalue.v4iSpesial Issue 1.567>
- Dagha, W. C., & Susetyo, Y. A. (2021). Web Event, Spring Boot, Java Pembangunan Aplikasi Web Event menggunakan Framework Spring Boot di PT XYZ. *JATISI*

- (*Jurnal Teknik Informatika dan Sistem Informasi*), 8(3), 1457-1469. doi:<https://doi.org/10.35957/jatisi.v8i3.1052>
- Desmayani, N. M., Wardani, N., Nugraha, P. G., Indrawan, I. Y., & Mahendra, G. S. (2022). Sistem Informasi Inventory pada PT. Djaya Buah Bersinar Denpasar Berbasis Web. *INSERT: Information System and Emerging Technology Journal*, 3(2), 82-92. doi:<https://doi.org/10.23887/insert.v3i2.54696>
- Dirosa, D., & Kurniawan, T. A. (2021). Pengembangan Sistem Point of Sale berbasis Web (Studi Kasus: Bos Duren Malang). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 5(4), 1615-1622. Retrieved from <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/8952>
- Fatman, Y., Hidayat, I. S., & Anadhiya, N. I. (2023). Rancang Bangun Sistem Informasi Tugas Akhir Menggunakan Metode Waterfall Di Fakultas Teknik Universitas Islam Nusantara. *EDUSAINTEK: Jurnal Pendidikan, Sains Dan Teknologi*, 11(1), 290-299. doi:<https://doi.org/10.47668/edusaintek.v11i1.988>
- Gosal, R., & Rustam, A. (2022). Perancangan Sistem Informasi Inventory Berbasis Web Pada Gudang Di PT. Spin Warriors. *Aisyah Journal of Informatics and Electrical Engineering, Volume 4 Issue 1*, 27-32. doi:<https://doi.org/10.30604/jti.v4i1.99>
- Hartono, M. A., & Mailoa, E. (2024). Pembuatan Website REST API Ensiklopedia Tumbuhan dengan Kombinasi Framework Spring Boot dan MyBatis Generator. *Jurnal Indonesia : Manajemen Informatika Dan Komunikasi*, 5(2), 1508-1520. doi:<https://doi.org/10.35870/jimik.v5i2.712>
- Hermansyah, Wijaya, R. F., & Utomo, R. B. (2023). Metode Waterfall Dalam Rancang Bangun Sistem Informasi Manajemen Kegiatan Masjid Berbasis Web. *KLIK: Kajian Ilmiah Informatika dan Komputer*, 3(5), 563-571. doi:<https://doi.org/10.30865/klik.v3i5.756>
- Khaqiqi, M. T., & Harani, N. H. (2023). *Penerapan Metode Gamifikasi Pada Rest Api Spring Boot*. Bandung: Penerbit Buku Pedia. Retrieved from https://books.google.co.id/books?hl=id&lr=&id=t76-EAAAQBAJ&oi=fnd&pg=PA1&dq=apa+itu+spring+boot&ots=VR3X5drFkK&sig=zc6XgauamVze_glLo-W5REv5PcI&redir_esc=y#v=onepage&q=apa%20itu%20spring%20boot&f=false
- Lewenusa, I. (2020). *Belajar Teknologi Ajax Untuk Pemula*. Retrieved from https://books.google.co.id/books?hl=id&lr=&id=zmj5DwAAQBAJ&oi=fnd&pg=PR1&dq=pengertian+ajax&ots=kMreki38Tl&sig=CfP3GBYwGELXW_RRN0-weBpNIHI&redir_esc=y#v=onepage&q=pengertian%20ajax&f=false
- Muhardi, M. (2019). Sistem E-Inventory Menggunakan Bahasa Pemrograman Visual Basic .NET. *Jurnal Ilmu Komputer*, 8(1), 138-143. doi:<https://doi.org/10.33060/JIK/2019/Vol8.Iss1.120>
- Nafiah, R., & Faih, A. (2019). Analisis Transaksi Financial Technology (Fintech) Syariah dalam Perspektif Maqashid Syariah. *Iqtishadia*, 6(2), 167-175. doi:<https://doi.org/10.19105/iqtishadia.v6i2.2479>

- PostgreSQL. (2024). *About*. Retrieved June 6, 2024, from <https://www.postgresql.org/about/>
- Putri, V. J., & Sulianta, F. (2023). Perancangan Front-End Aplikasi Kearsipan Berbasis Web (E-ARSIP) Di SMKN 6 Bandung. *Jurnal Darma Agung*, 31(4), 706-715. Retrieved from <http://jurnal.darmaagung.ac.id/index.php/jurnaluda/article/view/3217>
- Romadhon, M. H., Yudhistira, Y., & Mukrodin, M. (2021). Sistem Informasi Rental Mobil Berbasis Android Dan Website Menggunakan Framework Codeigniter 3 Studi Kasus : CV Kopja Mandiri. *Jurnal Sistem Informasi Dan Teknologi Peradaban*, 2(1), 30-36. Retrieved from <https://journal.peradaban.ac.id/index.php/jsitp/article/view/756>
- Sahi, A. (2020). Aplikasi Test Potensi Akademik Seleksi Saringan Masuk LP3I Berbasis Web Online menggunakan Framework Codeigniter. *TEMATIK*, 7(1), 120-129. doi:<https://doi.org/10.38204/tematik.v7i1.386>
- Sarmanela, B. C., Samiya, A. N., & Yuliana, M. E. (2023). Mengenal Backend Developer : Skill, dan Tanggung Jawab Menjadi Backend Developer. *Prosiding Seminar Nasional Teknologi Informasi dan Bisnis (SENATIB)*, 128-134. Retrieved from <https://ojs.uib.ac.id/index.php/Senatib/article/view/3019>
- Solehudin, A.-A., Fariz, N., Wahyu, N., Permana, R. F., & Saifudin, A. (2023). Rancang Bangun Digitalisasi Persediaan Barang Berbasis Web Menggunakan Metode Waterfall. *LOGIC: Jurnal Ilmu Komputer Dan Pendidikan*, 1(4), 1000-1005. Retrieved from <https://www.journal.mediapublikasi.id/index.php/logic/article/view/3163>
- Thoyyib, M. B., & Zulfikar, D. H. (2023). Desain UI/UX Website Referral untuk Program Gerakan Funding Culture Menggunakan Figma. *Prosiding Seminar Nasional Teknologi Komputer dan Sains*, 1(1), 232-241. Retrieved from <https://prosiding.seminars.id/prosainteks/article/view/67>
- Vicky, V. O., & Syaripudin, A. (2022). Perancangan Sistem Informasi Absensi Pegawai Berbasis Web Dengan Metode Waterfall (Studi Kasus : Kantor DBPR Tangerang Selatan). *OKTAL: Jurnal Ilmu Komputer Dan Sains*, 1(01), 17-26. Retrieved from <https://www.journal.mediapublikasi.id/index.php/oktal/article/view/2>
- Wahid, A. A. (2020). Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi. *Jurnal Ilmu-ilmu Informatika dan Manajemen STMIK*, 1-5. Retrieved from https://www.researchgate.net/publication/346397070_Analisis_Metode_Waterfall_Untuk_Pengembangan_Sistem_Informasi
- Yudhanto, Y., & Prasteyo, H. A. (2019). *Mudah Menguasai Framework Laravel*. Jakarta: PT Elex Media Komputindo. Retrieved from https://books.google.co.id/books?hl=id&lr=&id=8tKdDwAAQBAJ&oi=fnd&pg=PP1&dq=mudah+menguasai+framework+laravel&ots=lv241q-VVM&sig=uvh1PRdZqEaeY7rXfxijlswCYGg&redir_esc=y#v=onepage&q=mudah%20menguasai%20framework%20laravel&f=false