

PENGEMBANGAN WAREHOUSE MANAGEMENT SYSTEM MENGUNAKAN METODOLOGI SDLC PROTOTYPE (STUDI KASUS PT CENTURY BATTERIES INDONESIA)

Rayhan Putra Jafandi

Informatika, Universitas Singaperbangsa Karawang
Jalan HS. Ronggowaluyo, Karawang, Indonesia
2010631170028@student.unsika.ac.id

ABSTRAK

PT. Century Batteries Indonesia, produsen baterai asam timbal untuk berbagai keperluan, menghadapi tantangan manajemen stok di gudang produk jadi akibat pencatatan manual yang tidak efisien. Pengembangan *Warehouse Management System* (WMS) berbasis database menjadi solusi untuk meningkatkan efisiensi operasional, mengoptimalkan pencarian dan penempatan produk, mempercepat pengambilan barang, serta meningkatkan akurasi pencatatan persediaan. Observasi dan *meeting project* menunjukkan kendala seperti waktu penjemputan barang yang lama dan kesulitan identifikasi tipe baterai. WMS dapat mengurangi kesalahan dan meningkatkan efisiensi dengan sistem yang mencatat setiap transaksi di gudang. Penelitian ini bertujuan untuk merancang *warehouse management system* dengan metode *SDLC Prototype* dan menguji *Warehouse Management System* yang telah dikembangkan dapat mengurangi kesalahan, menghemat waktu, dan meningkatkan produktivitas operasional gudang. Sistem ini dirancang dengan meliputi enam tahap sesuai dengan metode yang digunakan yaitu *SDLC Prototype*: pengumpulan kebutuhan, desain, pembuatan *prototype*, pengembangan, pengujian, dan pemeliharaan. Desain menggunakan UML, sementara pengembangan dilakukan dengan *framework* CodeIgniter. Pengujian mencakup alpha dan beta *testing*, dengan model *Technology Acceptance Model* (TAM). *Technology Acceptance Model* (TAM) dengan hasil yaitu 94,3% dengan kategori "Sangat Baik". Implementasi WMS ini menjadi solusi efektif bagi PT. Century Batteries Indonesia untuk mengurangi kesalahan, menghemat waktu, dan meningkatkan produktivitas operasional gudang.

Kata kunci : *Warehouse Management System, SDLC Prototype, Efisiensi Gudang*

1. PENDAHULUAN

PT. Century Batteries Indonesia memiliki peran penting dalam produksi baterai asam timbal untuk kendaraan dan penggunaan selain otomotif. Perusahaan yang sudah beroperasi sejak 1971 dengan fasilitas produksi yang telah memenuhi standar internasional. PT. Century Batteries Indonesia telah menghasilkan berbagai jenis baterai otomotif dan non-otomotif dengan kualitas global.

Namun, manajemen stok yang efisien di gudang produk jadi (*warehouse finish goods*) menjadi tantangan. Awalnya, manajemen gudang menggunakan pencatatan secara manual yang kurang efisien dan rumit. Oleh karena itu, pengembangan WMS (*Warehouse Management System*) menjadi langkah penting untuk meningkatkan efisiensi pergudangan.

Berdasarkan hasil observasi yang telah dilakukan di PT. Century Batteries Indonesia dan *meeting project* dengan salah satu penanggung jawab di gudang produk jadi (*warehouse finish goods*) terdapat beberapa kendala yang dialami oleh staf di bagian tersebut. Menurut salah satu penanggung jawab di gudang produk jadi (*warehouse finish goods*), yaitu Indri Afrianti di divisi ISD (*Industrial System Development*) menyatakan bahwa: *Pick up time* yang lama ketika barang sudah berada didalam antrian untuk diletakan di rak yang kosong; staf melakukan pencatatan secara manual di *packing list customer* masing – masing; Penulisan tipe baterai pada bagian

sudut *packing* agar terlihat ketika barang diletakan di atas.

WMS (*Warehouse Management System*) adalah suatu sistem aplikasi komputer berbasis database, yang digunakan untuk meningkatkan efisiensi gudang dalam menjaga keakuratan data persediaan dengan melakukan pencatatan setiap transaksi dalam gudang [3].

Dengan penerapan WMS (*Warehouse Management System*) di PT. Century Batteries Indonesia dapat mengoptimalkan pencarian dan penempatan produk di gudang, mempercepat pengambilan barang, serta meningkatkan akurasi pencatatan persediaan. Fitur pemindai *QR Code*, terintegrasi dengan database akan mempercepat identifikasi produk dan mengurangi *human errors*. Melalui pemanfaatan *frameworks* seperti tersaji lebih *modern* dan *responsive* dalam *dashboard* yang intuitif dengan data *real-time* yang sesuai dengan stok produk yang ada di gudang sehingga dapat memberikan informasi akurat dan mudah dipahami. Dengan demikian semoga sistem akan dibuat dapat menjadi solusi untuk PT. Century Batteries Indonesia untuk mengurangi kesalahan, menghemat waktu, dan meningkatkan produktivitas operasional.

2. TINJAUAN PUSTAKA

2.1. Rancang Bangun

Rancang bangun adalah suatu cara pembentukan suatu sistem untuk menghasilkan sistem baru, dimana

sistem tersebut dapat digunakan untuk menggantikan atau menyempurnakan sistem yang ada secara keseluruhan atau sebagian dari sistem yang ada [1].

2.2. Sistem

Sistem merupakan kumpulan element – element yang saling berhubungan dan teritegrasi yang dirancang untuk mencapai tujuan. Element yang membentuk sistem tersebut adalah tujuan, *input*, *output*, *process*, *controll* dan *feedback* [4].

2.3. Website

Website merupakan gabungan dari halaman – halaman yang dapat menampilkan informasi berupa teks, video, gambar, suara, animasi dan kumpulan seluruhnya. Sehingga dapat bersifat statis atau dinamis dalam pembentukannya sehingga dapat saling terhubung antar halaman [2].

2.4. Warehouse Management System (WMS)

WMS (*Warehouse Management System*) adalah suatu sistem aplikasi komputer berbasis database, yang digunakan untuk meningkatkan efisiensi gudang dalam menjaga keakuratan data persediaan dengan melakukan pencatatan setiap transaksi dalam gudang [6]. Sistem ini bertujuan untuk mengontrol pergerakan dan penyimpanan persediaan dalam sebuah gudang dan memproses transaksi terkait dengan penerimaan, pemilihan, pengambilan dan pengiriman persediaan dalam gudang.

2.5. Framework

Framework adalah kumpulan kode program yang dapat digunakan secara langsung dengan mengikuti aturan penulisan tertentu dari masing – masing *framework*. Tujuan penggunaan *framework* adalah mempercepat pembuatan halaman website, karena *framework* telah menawarkan berbagai fitur siap pakai. Cukup gunakan fitur ini tanpa harus melakukan semuanya dari awal. Selain itu, aturan penulisan *framework* memaksa kita menggunakan metode yang baik [9].

2.6. Alpha Testing

Alpha testing merupakan pengujian pengujian terhadap aplikasi untuk memastikan aplikasi berfungsi dengan benar sesuai dengan kebutuhan dan tujuan yang diharapkan. *Alpha testing* fokus terhadap kebutuhan fungsional perangkat lunak dan fungsi yang diuji adalah fitur pada sistem [10].

2.7. Beta Testing

Beta testing merupakan pengujian yang dilakukan secara langsung dengan menyebarkan kuesioner yang dapat dihitung yang memungkinkan dibuatnya sebuah kesimpulan dari hasil kuesioner tersebut [11]. Pengujian *beta testing* terdapat pengambilan sampel dapat dilakukan melalui TAM (*Technology Acceptance Model*). Model ini adalah

salah satu teori yang menggunakan pendekatan perilaku yang digunakan untuk mengkaji proses adopsi suatu teknologi.

2.8. Codeigniter

Codeigniter adalah sebuah *framework* yang dibuat menggunakan bahasa pemrograman PHP yang bertujuan untuk mempermudah para developer website untuk membuat atau mengembangkan aplikasi berbasis website. *Codeigniter* memiliki eksekusi tercepat dibanding *framework* lainnya. Codeigniter bersifat *open source* dan menggunakan model berbasis MVC (*Model, View, Controller*) [5].

2.9. Software Development Life Cycle (SDLC)

SDLC Merupakan proses pengembangan atau memodifikasi *system* perangkat lunak dengan menggunakan model yang digunakan untuk mengembangkan suatu sistem perangkat lunak yang telah ada atau baru. SDLC memiliki beberapa model yang bisa digunakan dan dalam penelitian saat ini menggunakan model *prototype* [8].

2.10. Unified Modeling Language (UML)

Unified Modelling Language (UML) ialah suatu bahasa pemodelan yang berbentuk grafis yang digunakan untuk menggambarkan dan menetapkan spesifikasi sistem perangkat lunak. UML memberikan suatu standar dalam membuat model suatu sistem. Selain itu UML dapat diterapkan untuk pemahaman dan dokumentasi dari setiap sistem informasi. Pemanfaatan UML di sektor industri terus berkembang. Hal tersebut adalah standar terbuka yang membuatnya menjadi bahasa pemodelan yang banyak digunakan di industri perangkat lunak dan pengembangan sistem [7].

3. METODE PENELITIAN

Metodologi penelitian yang digunakan yaitu dengan SDLC (*Software Development Life Cycle*) *Prototype*. Metode ini dimulai sesuai dengan tahapan umum pada metode SDLC yaitu mulai dari pengumpulan kebutuhan (*requirement*), pembuatan desain (*design*), pembuatan *prototype* (*prototyping*), evaluasi *prototype* (*evaluating prototype*), pengembangan (*development*), pengujian sistem (*testing system*), pemeliharaan (*maintenance*). Metode ini sesuai karena dalam proses pengembangan terdapat bisa terlihat kesalahan apa saja yang dapat diperbaiki lebih awal.

3.1. Pengumpulan Kebutuhan (Requirement)

Tahap ini berfungsi sebagai penentuan elemen kunci yang menjadi dasar dalam menangani masalah pada warehouse finish goods, serta solusi sistem yang diperlukan dalam kasus ini. Hal ini dilakukan dengan mengevaluasi dan mempersempit masalah yang akan dipecahkan.

3.2. Pembuatan Desain (Design)

Tahap selanjutnya adalah pembuatan design. Pada tahap ini terbagi menjadi 2 bagian yaitu:

- a. Desain perancangan pembuatan UML (*Unified Modeling Language*)
Desain ini termasuk pembuatan Diagram *Use Case*, Diagram Aktivitas, Diagram *Sequence*, dan Diagram Kelas. Tahap ini dilakukan untuk menetapkan elemen – elemen yang diperlukan dalam membentuk komponen aplikasi yang akan direncanakan.
- b. Desain database
Desain ini merupakan pembuatan skema database dari *warehouse management system*.

3.3. Pembuatan Prototype (Prototyping)

Tahap selanjutnya adalah pembuatan *prototype*. Tahap ini dilakukan untuk membuat tampilan kasar atau *low-fidelity* untuk *warehouse management system*.

3.4. Pengembangan (Development)

Tahap ini merupakan implementasi dari desain dan *prototype* yang sudah disepakati ke dalam bahasa pemrograman yang digunakan, dan *framework* yang sudah disepakati.

3.5. Pengujian (Testing)

Langkah selanjutnya dalam model SDLC Prototype ini merupakan strategi yang digunakan pada penelitian, yang melibatkan pengujian. Pengujian sistem akan berkelanjutan hingga sistem pakar terbentuk sepenuhnya dan layak untuk dikembangkan dan ditingkatkan pengetahuan ahlinya. Sistem akan diuji dengan dua metode yaitu alpha testing dan beta testing.

3.6. Pemeliharaan (Maintenance)

Tahap terakhir adalah pemeliharaan program yang dilakukan dengan memperbaiki sistem. Oleh karena itu, langkah yang dapat diambil untuk menjaga *warehouse management system* adalah dengan melakukan pembaruan pada fitur sehingga sistem bisa lebih baik dalam mengelola *warehouse finish goods*. Pemeliharaan sistem dilakukan dalam bentuk pembaruan fitur dan pengguna dapat mengevaluasi apakah sistem telah sesuai seperti yang diharapkan.

4. HASIL DAN PEMBAHASAN

4.1. Hasil Penelitian

Hasil penelitian dibuat berdasarkan dengan langkah – langkah yang sudah direncanakan sesuai dengan rancangan penelitian agar *warehouse management system* yang sudah dibuat dapat berjalan sesuai dengan tujuan penelitian. Berikut merupakan rangkaian tahapan yang sesuai dengan metodologi SDLC (*System Development Life Cycle*) *Prototype* yang digunakan sebagai acuan dalam pembuatan rangkaian penelitian:

4.2. Pengumpulan Kebutuhan (Requirement)

Pengumpulan kebutuhan merupakan tahapan pertama dalam metode SDLC (*System Development Life Cycle*) *Recycle*. Tahap ini sangat penting dan berpengaruh untuk tahapan-tahapan selanjutnya. Tahap pengumpulan kebutuhan yang dilakukan diantaranya adalah:

- a. Analisis Masalah
Berdasarkan hasil *meeting project* yang dilakukan dengan Indri Afriyanti sebagai bagian dari divisi ISD (*Industrial System Development*) dan analisis permasalahan pada penelitian ini didapatkan beberapa masalah yaitu:
 - *Pick up time* atau waktu pengambilan barang yang sudah berada didalam antrian untuk diletakan di lokasi rak yang kosong terlalu lama.
 - *Staff* yang ditugaskan dalam *monitoring* barang masih menggunakan pencatatan secara manual dalam pengarsipan di *packing list customer* masing – masing.
 - Penulisan tipe baterai yang masih manual pada sudut box agar tipe baterai dapat terlihat dengan jelas.
- b. Analisis Kebutuhan
Dari analisis permasalahan tersebut, hal-hal yang diperlukan untuk menyelesaikan masalah tersebut adalah sebagai berikut:
 - Kebutuhan Fitur
Dari hasil analisis masalah yang sudah ada maka *warehouse management system* mempunyai kebutuhan fitur seperti pada tabel dibawah ini:

Tabel 1. Kebutuhan Fitur

No	Fitur	Fungsi	Jumlah
1.	<i>Count All Rack Location On, Off, Quantity Items</i>	Membuat agar disatu halaman dapat menampilkan semua lokasi rak yang terisi dan tidak terisi, serta dapat menghitung semua total <i>item</i> yang terdapat pada lokasi disetiap rak yang ada.	Berhasil
2.	<i>Scanning QR Code</i>	Membuat agar kamera dapat melakukan <i>scanning data</i> pada gambar <i>QR code</i> .	Berhasil
3.	<i>Input Data from QR (Manual Select Rack or Auto Select Rack)</i>	Membuat data hasil <i>scanning QR code</i> dapat di <i>store</i> pada database baik melalui pemilihan rak terlebih dahulu (<i>manual select rack</i>) atau tanpa melakukan pemilihan rak (<i>auto select rack</i>).	Berhasil
4.	<i>Update Data Item</i>	Dapat melakukan pembaruan <i>item</i> di database.	Berhasil
5.	<i>Reset Data Item</i>	Mengosongkan data di lokasi pada rak yang sudah terisi.	Berhasil
6.	<i>Bypass Data Item</i>	Memindahkan data di lokasi pada yang sudah terisi.	Berhasil

No	Fitur	Fungsi	Jumlah
7.	Add Half Item (Barang Sisa)	Dapat menambahkan Half Item (Barang Sisa) pada rak yang sudah terisi.	Berhasil
8.	List All Items	Menampilkan semua barang yang telah di input.	Berhasil
9.	Extract to Excel	Membuat resume dengan format excel dari semua data yang telah di input.	Berhasil

- **Kebutuhan Fungsional**
Tahap ini merupakan hasil kebutuhan fungsional yang diperlukan pada warehouse management system:

Tabel 2. Kebutuhan Fungsional

No	Deskripsi Kebutuhan
1.	Sistem dapat digunakan dalam mengatur item yang ada di gudang.
2.	Sistem harus dapat melakukan integrasi dengan API (Application Programming Interface) yang telah disediakan oleh perusahaan.
3.	Sistem dapat membuat kamera dapat digunakan sebagai scanner.
4.	Sistem dapat melakukan extract data kedalam format excel.

4.3. Pembuatan Desain (Design)

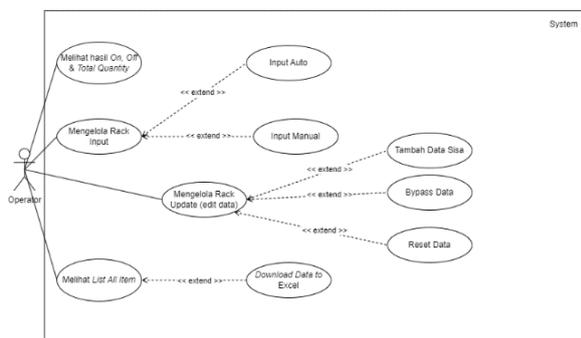
Tahap desain menggambarkan suatu rancangan yang dibangun agar dapat dipahami alur aplikasi yang akan digunakan. Oleh karena itu, tahap desain ini dibagi menjadi 3 sebagai berikut:

a. **Desain Arsitektur Aplikasi**

Pada tahap desain arsitektur aplikasi ini dirancang menggunakan diagram UML seperti use case diagram, activity diagram, sequence diagram dan class diagram. Untuk menjelaskan fungsi dan aliran data pada sistem pakar deteksi penyakit kaligata.

4.4. Use Case Diagram

Use Case Diagram digunakan untuk pemaparan interaksi langsung antara pasien dengan sistem.



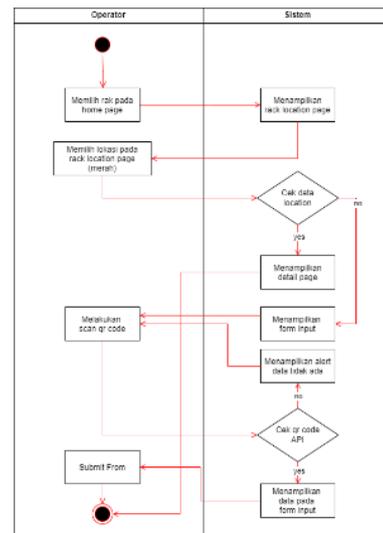
Gambar 1. Use Case Diagram

4.5. Activity Diagram

Setelah use case beserta skenario selesai dibuat, selanjutnya adalah membuat activity diagram. Activity diagram ini dirancang dengan tujuan untuk menjelaskan aktivitas yang dilakukan oleh user.

a. **Activity Diagram Mengelola Rack Input (Manual)**

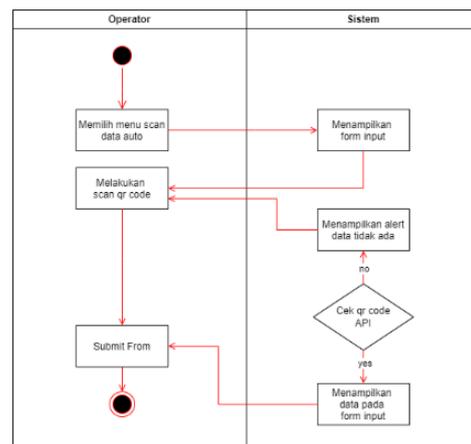
Gambar 2 di bawah menjabarkan proses input data menggunakan scan qr code yang dilakukan secara manual dengan memilih lokasi pada rak yang dipilih.



Gambar 2. Activity Diagram Mengelola Rack Input (Manual)

b. **Activity Diagram Mengelola Rack Input (Auto)**

Gambar 3 di bawah menjabarkan proses input data menggunakan scan qr code yang dilakukan secara otomatis untuk pemilihan rak dan lokasi.



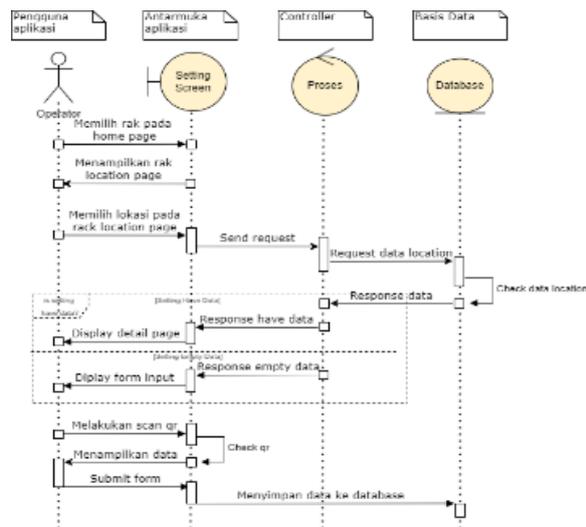
Gambar 3. Activity Diagram Mengelola Rack Input (Auto)

4.6. Sequence Diagram

Sequence diagram adalah suatu diagram yang memvisualisasikan interaksi antara objek dalam suatu sistem. Berikut ini adalah gambaran sequence diagram dari sistem yang akan dibuat.

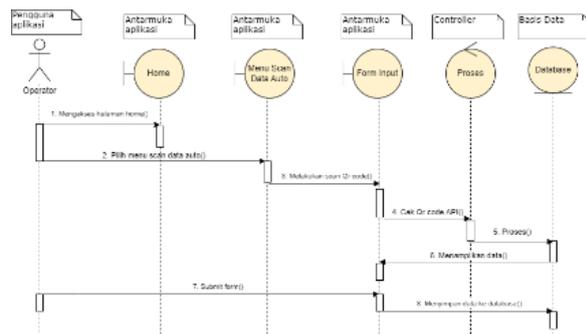
a. Sequence Diagram Mengelola Rack Input (Manual)

Gambar 4 di atas menjabarkan proses input data secara manual. Dimulai ketika operator memilih rak dan lokasi. Selanjutnya controller akan melakukan cek data apakah lokasi yang dipilih memiliki data atau tidak. Jika iya maka masuk ke menu detail dan jika tidak maka masuk ke form input. Lalu operator dapat melakukan scan qr code yang juga akan masuk ke pengecekan data dari qr code apakah data tersebut ada atau tidak. Jika ada maka data akan tampil pada form input, jika tidak maka tidak tampil apapun. Selanjutnya operator bisa melakukan submit data untuk menyimpan ke database.



Gambar 4. Sequence Diagram Mengelola Rack Input (Manual)

b. Sequence Diagram Mengelola Rack Input (Auto)



Gambar 5. Sequence Diagram Mengelola Rack Input (Auto)

Gambar 5 di atas menjabarkan proses input data secara auto. Dimulai ketika operator memilih menu input scan data auto, maka tampilan akan masuk ke form input. Lalu operator dapat melakukan scan qr yang juga akan masuk ke pengecekan qr apakah data tersebut ada atau tidak, jika ada maka data akan tampil pada form input, jika tidak maka tidak tampil apapun.

Selanjutnya operator bisa melakukan submit data untuk menyimpan ke database.

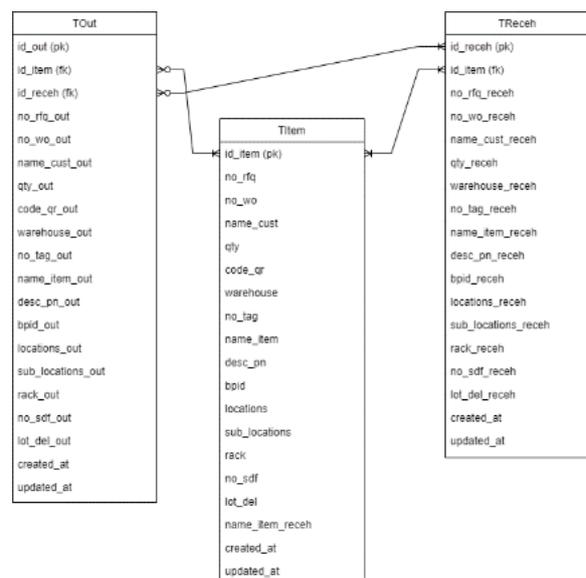
4.7. Class Diagram

Class diagram memberikan gambaran tentang kelas-kelas yang ada dalam sistem, hubungan antar kelas, atribut-atribut, dan dan fungsi-fungsi yang dimiliki oleh masing-masing kelas. Class diagram ini dibuat untuk merinci desain perangkat lunak sehingga dokumentasinya lebih terstruktur. Berikut merupakan class diagram dari sistem yang akan dibuat.



Gambar 6. Class Diagram

4.8. Desain Arsitektur Basis Data



Gambar 6. Entity Relationship Diagram Warehouse Management System

Setelah dilakukannya tahap perancangan arsitektur aplikasi, selanjutnya akan dilakukan tahap desain arsitektur basis data yang bertujuan untuk menggambarkan data-data yang saling berhubungan dengan sistem yang dirancang.

4.9. Pembuatan Prototype (Prototyping)

Desain *prototype* adalah tampilan kasar atau low-fidelity dari sistem yang dibuat. Desain *prototype* yang akan dibuat pada *Warehouse Management System*, yaitu sebagai berikut.

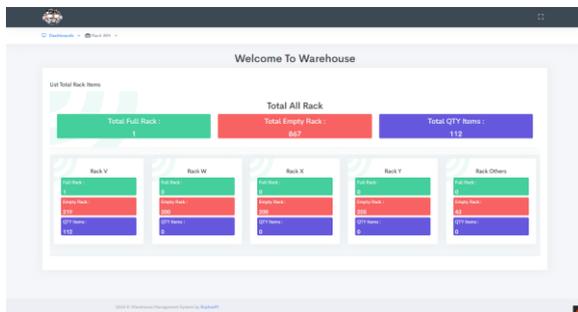
- a. Tampilan *Home*
Tampilan *home* menunjukkan perancangan *home* atau tampilan utama saat pertama kali mengakses aplikasi.
- b. Tampilan Menu *Form Input Manual*
Tampilan menu *form input* manual merupakan tampilan untuk operator melakukan *input* data secara manual. Sistem akan memberikan *form* yang dapat mengakses kamera supaya bisa melakukan *scanning qr code*.
- c. Tampilan Menu *Form Input Auto*
Tampilan menu *form input auto* merupakan tampilan untuk operator melakukan *input* data secara *auto*. Sistem akan memberikan *form* yang dapat mengakses kamera supaya bisa melakukan *scanning qr code*.

4.10. Pengembangan (Development)

Pada tahap development ini akan melakukan beberapa tahapan diantaranya melakukan tahapan pengkodean dari rancangan yang telah diberikan. Berikut merupakan tahapan pada fase development.

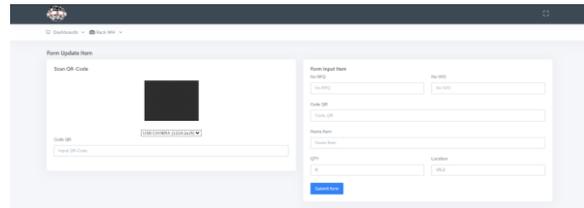
- a. *User Interface (UI)*
User Interface adalah tahap untuk implementasi dari desain *prototype* sebelumnya. Berikut *user interface* untuk *Warehouse Management System*.

- Tampilan *Home*
Pada gambar 7 menunjukkan *user interface home* saat pertama kali mengakses aplikasi



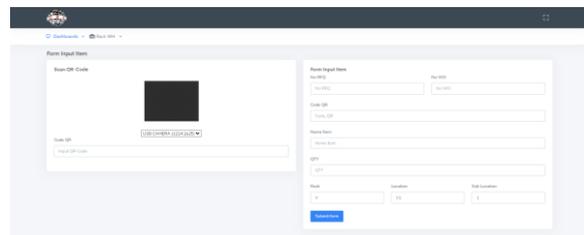
Gambar 7. User Interface Home

- Tampilan Menu *Form Input Manual*
Gambar 8 merupakan *user interface* untuk operator melakukan *input* data secara manual.



Gambar 8. User Interface Form Input Manual

- Tampilan Menu *Form Input Auto*
Gambar 9 merupakan *user interface* untuk operator melakukan *input* data secara *auto*.



Gambar 9. User Interface Form Input Auto

4.11. Pengujian (Testing)

- a. *Alpha Testing*
Alpha testing adalah suatu proses evaluasi sistematis terhadap perangkat lunak yang telah dikembangkan, dengan tujuan untuk mengidentifikasi kecacatan, memastikan kelayakan, dan memverifikasi apakah perangkat lunak tersebut berfungsi sesuai dengan persyaratan yang telah ditetapkan. Pada pengujian ini dilakukan dua testing yaitu dengan pengujian *black box testing* dan *white box testing*.

- *Black Box Testing*
Black box testing merupakan pengujian yang dilakukan untuk mengamati hasil dari eksekusi melalui data uji dengan menguji fungsionalitas sistem tanpa memiliki pengetahuan rinci tentang struktur internal atau implementasi perangkat lunak. Pengujian *black box testing* hanya mengevaluasi dari tampilan *interface*

Tabel 3. Black Box Testing Mengelola Rack Input

No.	Data Masukan	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	Memilih <i>menu input rack auto</i>	Melakukan <i>input auto</i>	Sesuai harapan	Diterima
2.	Memilih <i>input</i> secara manual	Melakukan <i>input</i> manual	Sesuai harapan	Diterima

3.	Scan qr code	Dapat melakukan scan qr code	Sesuai harapan	Diterima
4.	Mengisi form input	Dapat mengisi form tanpa harus melakukan scan	Sesuai harapan	Diterima
5.	Memilih tombol submit	Input data ke database	Sesuai harapan	Diterima

Berdasarkan hasil pengujian menggunakan *black box testing*, dapat disimpulkan bahwa sistem telah menjalani pengujian menyeluruh dari perspektif fungsionalitas pengguna. Hal ini memastikan bahwa setiap fitur dan fungsionalitas sistem beroperasi sesuai dengan yang diharapkan oleh pengguna.

• *White Box Testing*

Berdasarkan hasil pengujian menggunakan *white box testing* maka didapatkan nilai *cyclomatic complexity* (V(G)) untuk kode input data tersebut adalah 36. Ini menunjukkan bahwa terdapat 36 jalur yang berbeda dalam aliran kontrol program. Semakin tinggi nilai

cyclomatic complexity, maka akan semakin kompleks aliran kontrolnya.

b. *Beta Testing*

Tahap pengujian selanjutnya adalah pengujian beta testing menggunakan *Technology Acceptance Model* (TAM). Pengujian ini bertujuan untuk mengevaluasi apakah aplikasi yang telah dikembangkan memenuhi standar atau tidak, apakah sistem yang telah dibuat dapat membantu pekerjaan atau tidak. Pengujian ini dilakukan melalui questioner dengan indikator yang sudah disesuaikan. Berikut ini tabel pengujian menggunakan model TAM:

Tabel 4. Kode Variabel

No	Variabel	Kode	Indikator
1	Persepsi Kebermanfaatan (Perceived Usefulness)	PU1	Saya merasa Warehouse Management System (WMS) membantu memberikan informasi yang aktual
2		PU2	Saya merasa Warehouse Management System efektif dalam membantu pekerjaan?
3	Persepsi Kemudahan Penggunaan (Perceived Ease of Use)	PEU1	Saya mudah untuk mempelajari cara menggunakan Warehouse Management System?
4		PEU2	Saya mudah berinteraksi dengan fitur yang ada di Warehouse Management System?
5	Sikap terhadap penggunaan (Attitude Toward Using)	ATU1	Seberapa positif sikap Anda terhadap penggunaan Warehouse Management System dalam pekerjaan Anda?
6		ATU2	Saya menerima terhadap penggunaan Warehouse Management System setelah menggunakannya untuk waktu tertentu?
7	Intensitas perilaku Penggunaan (Behavioral Intention to Use)	BIU1	Seberapa kuat niat Anda untuk terus menggunakan Warehouse Management System di masa depan?
8		BUI2	Seberapa sering Anda berencana untuk menggunakan Warehouse Management System dalam pekerjaan sehari-hari Anda?
9	Penggunaan Sistem Secara Aktual (Actual System Use)	ASU1	Seberapa sering Anda menggunakan Warehouse Management System dalam pekerjaan sehari-hari Anda?
10		ASU2	Bagaimana pengalaman Anda menggunakan Warehouse Management System setelah menggunakannya didalam pekerjaan?

Berdasarkan responden yang terdiri dari 4 karyawan dari PT Century Batteries Indonesia. Berikut table data responden karyawan :

Tabel 5. Total Skor Variabel

No	Nama Variabel	Total	Skor Ideal
1	Persepsi Kebermanfaatan (Perceived Usefulness)	144	152
2	Persepsi Kemudahan Penggunaan (Perceived Ease of Use)	143	152
3	Sikap Terhadap Penggunaan (Attitude Toward Using)	148	152
4	Intensitas Perilaku Penggunaan (Behavioral Intention to Use)	142	152
5	Penggunaan Aktual Sistem (Perceived Ease of Use)	140	152
Jumlah		717	760

Tabel 6. Tabel Persentase

No	Skala	Tingkat Kepuasan
1.	0% - 20%	Tidak Setuju
2.	21% - 40%	Kurang Setuju
3.	41%-60%	Cukup
4.	61%-80%	Setuju
5.	81%-100%	Sangat Setuju

Dari 19 responden terdapat nilai total dari masing – masing variabel yaitu 717 point. Skor yang diperoleh kemudian dihitung dan diperoleh persentase kelayakan dengan rumus sebagai berikut:

$$Y = \frac{\sum(N.R)}{Skor\ Ideal} \times 100\%$$

$$Y = \frac{717}{760} \times 100\% = 94.3\%$$

Hasil dari perhitungan persentase kelaakan adalah 94,3% sehingga dapat disimpulkan bahwa system telah diterima dengan baik menggunakan

Technology Acceptance Model (TAM) dengan kategori “Sangat Baik” jika dilihat dari table persentase.

4.12. Pemeliharaan (*Maintenance*)

Pada penelitian ini, tahap pemeliharaan dilakukan selama 4 bulan aplikasi digunakan agar aplikasi dapat dikembangkan sesuai dengan perubahan yang terjadi sehingga pengetahuan sistem dapat lebih baik lagi dari waktu ke waktu, tetapi tidak membuat aplikasi baru. Pada rentang waktu 4 bulan terjadi beberapa perubahan untuk menambahkan tabel TOut pada database yang tujuannya untuk melihat data apa saja yang pernah ada pada warehouse finish goods.

Perubahan selanjutnya terdapat pada penambahan fitur bypass yang fungsinya untuk melakukan pemindahan data ke tempat yang dituju tanpa melakukan reset data terlebih dahulu. Pemeliharaan sistem memiliki peran penting dalam meningkatkan kinerja operator *warehouse finish goods*. Hal ini memastikan bahwa pengetahuan yang terdapat dalam program sistem terus berkembang sejalan dengan perkembangan zaman. Selain itu, pemeliharaan juga diperlukan untuk memastikan agar program dapat beroperasi secara optimal.

4.13. Pembahasan

Berdasarkan hasil penelitian yang telah dilakukan, *warehouse management system* berbasis web telah dirancang langkah demi langkah sesuai dengan prinsip-prinsip yang ditemukan dalam literatur dan saran dari para ahli. Sistem ini dirancang dengan menerapkan pendekatan kerangka kerja *System Development Life Cycle (SDLC) Prototype*. Metode ini memiliki enam tahapan yaitu, pengumpulan kebutuhan (*requirement*), pembuatan desain (*design*), pembuatan *prototype (prototyping)*, pengembangan (*development*), pengujian (*testing*), dan pemeliharaan (*maintenance*).

Tahap pertama yaitu pengumpulan kebutuhan (*requirement*) adalah tahap dimana analisis masalah dan analisis kebutuhan dibuat agar pada tahap ini system yang dibuat dapat diidentifikasi permasalahan dan kebutuhan untuk *system* yang dibuat. Tahap kedua yaitu pembuatan desain (*design*). Pada tahap ini pembuatan rancangan menggunakan UML (*Unified Modeling Language*) dilakukan termasuk pembuatan *use case diagram*, *activity diagram*, *sequence diagram* dan *class diagram*. Pada tahap ini juga ada pembuatan desain database, pembuatan database dilakukan untuk membuat skema tempat penyimpanan data yang diperlukan dalam *warehouse management system*.

5. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian yang menerapkan metode *Software Development Life Cycle Prototype* dalam pengembangan *Warehouse Management System* berbasis web, dapat disimpulkan bahwa aplikasi ini berhasil memenuhi kebutuhan dengan baik, terbukti dari skor tinggi dalam pengujian beta

testing menggunakan *Technology Acceptance Model* (TAM). Implementasi WMS ini memberikan solusi efektif bagi PT. Century Batteries Indonesia dalam mengurangi kesalahan, menghemat waktu, dan meningkatkan produktivitas operasional gudang.

Meskipun demikian, terdapat beberapa saran untuk penelitian mendatang guna meningkatkan kesempurnaan sistem, seperti penambahan fitur untuk memantau data masuk dan keluar dengan lebih visual serta pengembangan sistem ini untuk *platform* lain seperti android dan desktop. Dengan demikian, pengembangan lebih lanjut pada aspek tersebut dapat meningkatkan fungsionalitas dan fleksibilitas sistem sesuai dengan kebutuhan operasional yang berkembang.

DAFTAR PUSTAKA

- [1] P. Anjelita, "Rancang Bangun Sistem Informasi E-Learning Pada Smk Negeri 3 Batam," Skripsi thesis, Prodi Sistem Informasi, 2019.
- [2] P. Anjelita, and E. R. "Rancang Bangun Sistem Informasi E-Learning Pada SMK Negeri 3 Batam," Computer and Science Industrial Engineering (COMASIE), 2019.
- [3] I. G. Ayu, P. A., and I. Nyoman, N., "Pelnelrapan Warelhousel Managemelnt System Pada Pt Uniplastindo Intelrbuana Bali," J. Telknologi dan Opeln Sourcel, pp. 7216-7238, 2019.
- [4] A. J. Fajriyah, "Rancang Bangun Sistem Informasi Telndelr Karelt Delsa Jungai Melnggunakan Meltodel Watefrfall," J. SISFOKOM, 2017.
- [5] A. Pratama, Codelignitelr Uncovelr - Panduan Bellajar Codelignitelr Untuk Pelmula. Dunia Ilkom, 2022.
- [6] K. N. Ramaa, and T. M. Rangaswamy, "Impact of Warehouse Management System in a Supply Chain," International Journal of Computer Applications, vol. 54, no. 1, pp. 1–20, 2012.
- [7] A. S. Rosa, Analisis dan Desain Perangkat Lunak: Rekayasa Perangkat Lunak Untuk Pemrograman Terstruktur, Berorientasi Objek, dan Agile. Bandung: Informatika, 2022.
- [8] A. S. Rosa, and M. Shalahudin, Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek). Bandung: Informatika, 2018.
- [9] R. Setiawan, "APA ITU framework? developer Wajib Tahu," Dicoding Blog, Dec. 15, 2021. Available: <https://www.dicoding.com/blog/apa-itu-framework/>. [Accessed: Mar. 18, 2022].
- [10] L. Setiyani, "Pengujian Sistem Informasi Inventory Pada Perusahaan Distributor Farmasi Menggunakan Metode Black Box Testing," J. Ilmu Komputer Dan Teknologi Informasi, vol. 4, no. 1, pp. 20-27, 2019.
- [11] A. Suandi, F. N. Khasanah, and E. Retnoningsih, "Pengujian Sistem Informasi E-Commerce Usaha Gudang Cokelat Menggunakan Uji Alpha dan Beta," Information System For Educators and Professionals, 2017.