

Rancang Bangun Cycle Count Inspection System untuk Manajemen Stok Inventori pada Warehouse Berbasis Website

Ravena Auliya¹, Nelmiawati²

^{1,2} Teknik Informatika, Politeknik Negeri Batam

ravena.auliya@student.polibatam.ac.id¹, mia@polibatam.ac.id²

Article Info

Article history:

Received ...

Revised ...

Accepted ...

Keyword:

Cycle count,
FSN analysis,
Incremental Method,
ITR,
Inventory,
Warehouse.

ABSTRACT

Batam is an industrial city. An industrial company has a logistics department that plays an important role in planning, monitoring, and coordinating the flow of inventory in and out of the company. Inventory is a supply. To optimize inventory management, the logistics department is equipped with a warehouse. This warehouse is designed to store large quantities of goods and ensure the availability of goods while maintaining the inventory. The storage of large quantities of goods causes the problem that the actual stock of goods is sometimes not in line with the system data. This error must be resolved immediately because it will have a bad impact if at any time an audit is held to harmonize procedures and calculations. There is a solution that can be used to minimize these errors, namely making a website-based Cycle Count Inspection System. This system aims to help calculate the accuracy of inventory stock data to minimize the error of differences in inventory stock data. In addition, with this system, the previous calculation work that was still manual can be done more easily by utilizing a website technology system, so that the calculation work will be faster and more structured. Cycle count in this system is carried out periodically according to the FSN analysis method based on ITR (Inventory Turnover Ratio). Later this ITR will become a benchmark for the cycle count schedule to be carried out. This system is made using .Net Framework, namely ASP.NET MVC Web Application, and SQL Server database. With the Incremental Method, this system is built starting from the analysis stage, passing the build stage to testing. After going through various stages and testing, this system can be used to overcome the problem of data differences and help the cycle count work to be more structured.



This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.

I. PENDAHULUAN

Batam merupakan kota industri yang berada dalam gugusan Provinsi Kepulauan Riau. Sejak ditetapkan sebagai kawasan industri, secara bertahap, investor mulai banyak berdatangan. Hal ini dapat dibuktikan dengan banyaknya industri dari berbagai jenis dan skala yang sudah memberikan perannya di kota batam [1]. Salah satu industri yang ada di kota batam adalah industri semikonduktor.

Sebuah perusahaan industri semikonduktor memiliki berbagai departemen, salah satunya adalah departemen logistik. Departemen logistik adalah departemen yang

bertanggung jawab dalam mengelola arus barang, pengelola strategis ekspor dan impor, penyimpanan suku cadang, maupun bahan baku komponen semikonduktor. Departemen ini memiliki peranan penting untuk merencanakan, melakukan pengawasan, serta mengkoordinasikan aliran inventori yang masuk maupun keluar perusahaan [2]. Secara harfiah, inventori merupakan persediaan. Inventori yang dikelola perusahaan dapat berupa bahan mentah yang dibeli dan diubah menjadi suatu produk. Terdapat banyak jenis inventori yang dikelola oleh perusahaan yakni bahan jadi atau barang yang sudah siap di jual, bahan baku atau bahan yang

akan digunakan untuk proses produksi, stok pengaman dan lain sebagainya [3]. Untuk mengoptimalkan manajemen inventori, departemen logistik dilengkapi dengan *Warehouse* (gudang penyimpanan inventori) yang berfungsi sebagai tempat penyimpanan berbagai inventori, bahan baku, dan suku cadang produksi. *Warehouse* dirancang untuk tempat penyimpanan inventori dalam jumlah besar dan memastikan ketersediaan inventori yang akan digunakan untuk proses produksi sekaligus menjaga stok inventori dengan aman dan teratur [4]. Penyimpanan inventori dalam jumlah besar dapat menimbulkan sebuah permasalahan yaitu stok aktual inventori tidak selaras dengan data inventori yang ada dalam sistem perusahaan. Permasalahan ini akan menjadi besar bila sewaktu-waktu diadakannya audit untuk memeriksa dan memastikan akurasi, serta mengidentifikasi kesalahan pada prosedur pengelolaan.

Cycle counting merupakan proses perhitungan akurasi persediaan aktual inventori secara terus menerus dengan membandingkan data stok yang ada di sistem. Proses ini bertujuan untuk mengenali perbedaan stok dan memperbaiki kesalahan dalam pencatatan inventaris dengan rentang waktu harian, mingguan atau bulanan. Sehingga setiap tahun data inventori dapat dikalkulasikan beberapa kali [5]. Berdasarkan hasil diskusi dengan koordinator *warehouse*, dapat disimpulkan bahwa pengerjaan kalkulasi sebelumnya tidak dijalankan dengan baik. Kalkulasi sebelumnya tidak mempunyai jadwal yang jelas, dan pengerjaan masih berupa manual dengan mencetak data dari sistem. Kemudian, cetakan tersebut akan dicocokkan ke data aktual inventori. Hal ini dapat menciptakan celah bagi terjadinya kesalahan manusia dalam perhitungan, dan dikhawatirkan dapat menimbulkan potensi kecurangan. Selain diskusi, survei lokasi juga dilakukan dengan melihat rak penyimpanan inventori, tata letak inventori, dan mencoba proses kalkulasi manual. Dalam survei tersebut, proses kalkulasi dilakukan dengan mengambil sampel sebagai percobaan. Hasil percobaan menunjukkan adanya kesalahan perhitungan, yaitu jumlah data aktual berbeda dengan jumlah data di sistem.

Dengan adanya diskusi dan survei yang telah dilakukan, terdapat solusi yang ditawarkan untuk membuat sistem *cycle counting* berbasis *website*, yaitu **Cycle Count Inspection System**. Berdasarkan standar operasional prosedur perusahaan, sistem ini mengacu pada penerapan metode *FSN analysis* untuk mengatur jadwal kalkulasi dan mengklasifikasikan inventori berdasarkan besar dan kecilnya ITR inventori. Sistem ini mampu memastikan kalkulasi berjalan dengan baik. Sistem akan dibangun menggunakan .Net Framework yaitu ASP.NET MVC Web Application, serta *database* SQL Server. Sistem ini tidak menerapkan proses *login* dengan memasukkan *username* dan *password*, melainkan menggunakan otentikasi dan otorisasi. Ketika nama perangkat/PC terdaftar, maka pengguna akan otomatis masuk ke sistem tanpa perlu memasukkan *username* dan

password. Dengan adanya sistem ini, diharapkan dapat meminimalisir kesalahan perhitungan akurasi, pelaksanaan kalkulasi tepat waktu sehingga dapat selalu terpantau, dan pastinya dapat meminimalisir kesalahan pada tahap audit.

II. LANDASAN TEORI

A. Tinjauan Pustaka

Sebelumnya, beberapa sumber telah membuat sistem terkait pengelolaan data stok inventori dengan fokus pengembangan yang berbeda-beda. Berikut beberapa penelitian pengembangan sistem yang sudah pernah dilakukan seperti:

1. Penelitian mengenai Rancang Bangun Sistem Informasi Inventory Berbasis web menggunakan metode klasifikasi ABC. Yang sebelumnya sudah pernah dilakukan oleh Muhammad Zhafari [6]. Sistem ini bertujuan untuk membantu PT SKA dalam menerapkan sistem secara online dikarenakan proses pencatatan dan pengelolaan inventori masih bersifat semi manual dengan *Microsoft Office*. Sistem ini dibangun dengan menggunakan PHP sebagai bahasa pemrograman *back-end* dan Framework CodeIgniter 4, HTML, dan Framework CSS Bootstrap 5 dan MySQL sebagai *database*.
2. Penelitian mengenai Perancangan Sistem Inventory Berbasis Web Studi Kasus UD. Berkah Ananda. Penelitian ini dilakukan oleh Eel Susilowati [7]. Sistem inventori ini dibangun untuk membantu UD. Berkah Ananda dalam pengelolaan inventori. Aktifitas yang dilakukan masih dalam tahap manual di buku pencatatan, sehingga mengakibatkan adanya ketidakselarasan data inventori di Gudang dengan pembukuan serta sulitnya membuat laporan dikarenakan harus disesuaikan kembali dan membutuhkan waktu lama. Sistem ini dibangun menggunakan PHP, MySQL dan UML sebagai pemodelan sistem.
3. Penelitian mengenai Rancang Bangun Sistem Warehouse Multi Cabang Dengan Metode Klasifikasi ABC Berbasis Website Pada PT Young Multi Sarana. Penelitian ini dilakukan oleh Ferdinand Novan Christian [8]. Sistem warehouse ini dibangun untuk membantu PT Young Multi Sarana dalam proses pengelolaan produk yang sebelumnya masih manual. Masalah lainnya muncul ketika pengelolaan dan peletakan inventori masih dilakukan secara acak, akibatnya butuh waktu 10-15 menit untuk membuat daftar permintaan inventori. Sistem ini dibangun menggunakan PHP, Framework Laravel, dan *database* MySQL.

Berikut beberapa aspek pembeda pada setiap sistem inventori dari tinjauan Pustaka dengan sistem yang penulis buat.

TABEL I
Tabel Perbandingan sistem

Perbandingan	[6]	[7]	[8]	Tugas Akhir Penulis
Nama Perusahaan	PT SKA	UD. Berkah Ananda	PT Young Multi Sarana	Perusahaan Industri Semikonduktor Batam
Bahasa Pemrograman	PHP	PHP	PHP	HTML, CSS, JAVASCRIPT
Framework	CodeIgniter 4	-	Laravel	ASP.NET MVC Web Application
Fitur Utama	Pengelolaan data inventori masuk dan keluar.	Pengelolaan data inventori masuk dan keluar.	Pengelolaan permintaan inventori, rekap klasifikasi inventori, peletakan inventori sesuai metode ABC	Pengelolaan proses kalkulasi <i>cycle counting</i> ,
Metode	ABC (berdasarkan nilai investasi tinggi)	-	ABC (berdasarkan nilai investasi tinggi)	FSN (berdasarkan cepat lambatnya inventori keluar, <i>fast</i> , <i>slow</i> dan <i>non-moving</i>)
Fokus Penelitian	Pengelolaan data inventori	Pengelolaan data inventori	Penempatan inventori dan Pengelolaan permintaan inventori	Kalkulasi tanpa adanya kesalahan.

Berdasarkan Tabel 1 perbandingan sistem, ketiga penelitian tersebut dijadikan acuan dalam pembuatan sistem. Jurnal tersebut nantinya akan dikembangkan serta disesuaikan dengan permasalahan yang ada. Ketiganya mempunyai tujuan yang sama dengan dua jurnal yang mengangkat metode yang sama. Metode ABC sudah dipertimbangkan terlebih dahulu untuk mengelola stok inventori dalam sistem ini. Namun, setelah dilakukan survei dan diskusi, metode ABC tidak cocok digunakan karena merupakan metode klasifikasi berdasarkan nilai investasi tertinggi melalui pergerakan inventori berdasarkan omset. Sedangkan sistem yang akan dibuat mengklasifikasikan perhitungan stok inventori berdasarkan kecepatan inventori masuk dan keluar tanpa melibatkan perubahan lokasi peletakan.

B. Cycle Counting

Cycle counting merupakan sebuah metode pendekatan manajemen pergudangan untuk menghitung persediaan aktual inventori secara berkala dengan membandingkan data stok aktual inventori dengan data stok inventori yang ada di sistem yang bertujuan untuk mengidentifikasi perbedaan stok, memperbaiki kesalahan dalam pencatatan inventaris dan memastikan bahwa segala prosedur telah dijalankan secara terstruktur [9]. *Cycle counting* menjadi fitur utama dalam sistem ini. Sistem ini juga dapat membaca keselarasan antara data aktual yang nanti akan dimasukkan dengan data yang ada di sistem. Data tersebut nantinya akan di cocokkan dengan hasil persentase dan sedikit penjelasan sebagai hasil akhirnya.

C. FSN Analysis

FSN Analysis adalah sistem klasifikasi inventori

berdasarkan *Turnover Ratio*. Rasio ini mengacu pada kecepatan inventori tersebut keluar dan dapat diartikan sebagai rasio tahunan konsumsi inventori dibagi dengan persediaan rata-rata. Ada 3 kategori dalam *FSN Analysis* sebagai berikut [10]:

1. *Fast Moving* yaitu mengacu pada hasil pembagian lebih dari 3. Sebagai patokan bahwa dalam setahun inventori tersebut cepat keluar sehingga menjadi prioritas dalam jadwal kalkulasi.
2. *Slow Moving* yaitu mengacu pada hasil pembagian antara 1-3. Sebagai patokan bahwa dalam setahun inventori tersebut tidak selalu sering keluar seperti pada klasifikasi pertama.
3. *Non Moving* yaitu mengacu pada hasil pembagian kurang dari 1. Sebagai patokan bahwa dalam setahun inventori tersebut jarang keluar sehingga nantinya akan semakin jarang dilakukan kalkulasi.

Sistem ini akan menerapkan metode *FSN analysis* yang sudah menjadi standar operasional prosedur perusahaan sebagai penerapan jadwal kalkulasi. *FSN* akan mengkategorikan inventori berdasar nilai *ITR* setiap inventori. Dengan *FSN* ini, semakin cepat inventori tersebut keluar maka semakin rutin kalkulasi dilakukan pada inventori tersebut.

D. ITR (Inventory Turnover Ratio)

Inventory Turnover Ratio merupakan rumus rasio yang menunjukkan seberapa efektif persediaan di dapat kelola. *ITR* bertujuan untuk mengukur rata-rata persediaan yang di putar dalam suatu periode [11]. Dalam sistem ini *ITR* sudah menjadi standar operasional prosedur perusahaan dan sudah ada sewaktu data inventori dimasukkan ke dalam sistem. *ITR*

digunakan untuk mengelola jadwal kalkulasi. Semakin besar nilai ITR inventori maka jadwal kalkulasi akan lebih sering di lakukan dan begitu pula sebaliknya.

E. .NET Framework (ASP.NET MVC Web Application)

.Net Framework merupakan teknologi pengembangan yang terdiri dari beberapa alat, bahasa pemrograman, dan perpustakaan untuk membangun berbagai tipe aplikasi. Perangkat lunak ini di desain untuk bekerja terutama pada sistem operasi *Microsoft Windows* [12]. ASP.NET merupakan kerangka pengembangan perangkat lunak untuk membangun sebuah website yang menggunakan HTML, CSS, dan JavaScript. ASP.NET MVC merupakan metode pembangunan website dengan menggunakan *Model, View, dan Controller*. Cara ini memiliki pola yang memudahkan pembangunan website [13].

F. SQL Server

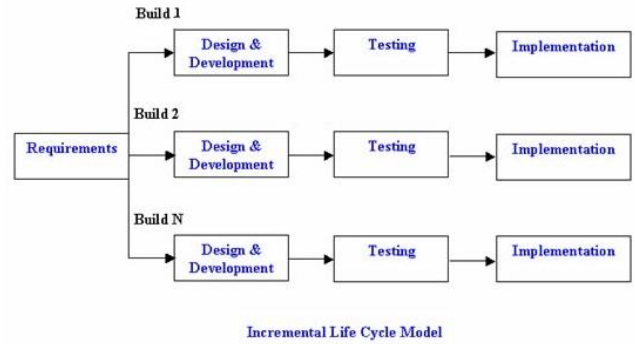
SQL (Structured Query Language) pada dasarnya digunakan sebagai media komunikasi dengan *database*. *Microsoft* menjadi salah satu sistem manajemen basis data rasional (RDBMS) yang memakai SQL server. SQL server merupakan perangkat lunak yang bersifat *client/server* yang komponennya berfungsi untuk menampilkan dan memanipulasi data [14].

G. Cycle Count Inspection System

Sistem *Cycle count* merupakan sebuah sistem yang menghitung akurasi persentase antara data stok aktual dengan data di sistem. Pengguna memasukkan data inventori beserta ITR yang sudah ditentukan ke dalam sistem. Setelah memasukkan data inventori ke dalam sistem, pengguna akan mendapatkan email pemberitahuan tentang jadwal kalkulasi hari ini berdasarkan kategori yang telah diemailkan. Berdasarkan ITR tersebut, pengguna akan menghitung akurasi dengan memasukkan data aktual dan sistem akan membandingkan dengan data yang sudah ada di sistem. Selain itu, sistem ini akan melaporkan hasil kalkulasi jika tidak mencapai 100% benar guna memastikan keakuratan perhitungan stok inventori selalu terpantau.

III. METODE

Sistem *Cycle count* ini dibuat dengan metode *Incremental* yang pengembangannya berdasarkan pada *requirement software* dan pembuatannya dipecah menjadi beberapa bagian/fungsi sehingga pengerjaannya akan dilakukan secara bertahap. Hasil dari pengembangan model ini semakin lama akan semakin lengkap sampai versi terlengkap sistem yang akan menjadi hasil akhir. Metode ini sesuai dengan pengembangan sistem *cycle count* karena mengalami pengulangan tahapan dan juga penambahan fungsi di setiap pertemuan *user* yang sudah dijadwalkan.



Gambar 1. Metode Incremental (Ilham Aji K, Brain Strom) Pada gambar 1 menjelaskan bahwa metode *Incremental* memiliki beberapa tahapan dalam pengembangannya, meliputi [15]:

A. Tahapan Requirement

Pada tahap ini dilakukan diskusi bersama koordinator *warehouse*. Berdasarkan diskusi tersebut dapat disimpulkan bahwa pengerjaan kalkulasi sebelumnya tidak mempunyai jadwal yang jelas dan pengerjaan masih berupa manual dengan mencetak data dari sistem dan cetakan tersebut akan di cocokkan ke data aktual inventori. Selain diskusi, survei lokasi dilakukan dengan mencoba kalkulasi menggunakan metode manual. Hasil percobaan tersebut tidak efektif dikarenakan terdapat kesalahan dalam proses perhitungan. Kesalahan tersebut muncul dikarenakan terdapat inventori yang memiliki perbedaan jumlah stok sistem dan aktual stok. Hal itu dikarenakan peletakan inventori yang tidak sesuai dengan sistem, dan kesalaham perhitungan manual. Dengan adanya diskusi dan survei ini, nantinya informasi ini dikumpulkan dan dianalisa untuk masuk ke tahap selanjutnya.

B. Tahapan Specification

Pada tahap ini menganalisa kebutuhan-kebutuhan yang cocok untuk sistem yang akan dibangun dan analisa kebutuhan tersebut akan menjadi acuan utama. Dalam sistem ini beberapa rancangan antara lain:

B.1 Rancangan Karakteristik Aktor

Aktor yang berperan dalam sistem ini ada 2 yaitu Admin dan *Warehouse* tim. Keduanya memiliki karakteristik yang berbeda.

TABEL II
Tabel Karakteristik Pengguna

Aktor	Karakteristik
Admin	Admin dapat menambah, mengubah dan menghapus data inventori, <i>role</i> akun dan email akun. Admin juga dapat mencetak laporan hasil kalkulasi. Admin hanya bisa melihat hasil kalkulasi namun tidak dengan melakukan kalkulasi.
<i>Warehouse</i> Tim	<i>Warehouse</i> tim dapat melakukan kalkulasi, mendapat email pengingat kalkulasi, mencetak laporan dan mendapat laporan jika kalkulasi tidak 100% berhasil.

B.2 Rancangan Kebutuhan Fungsional

Sistem ini memiliki kebutuhan fungsional sebagai gambaran bagaimana fungsi dalam sistem berjalan.

Tabel III
Tabel Kebutuhan Fungsional

No	Kebutuhan Fungsional
F001	Sistem menangani <i>login</i> otomatis ketika dengan akun <i>windows</i> yang sudah di daftarkan dan menangani akun berdasarkan <i>role</i> dan menampilkan fitur berdasarkan hak akses.
F002	Admin dapat mendaftarkan pengguna yang ingin menggunakan website sekaligus memberi <i>role</i> akses fitur.
F003	Admin dapat memasukkan, mengubah dan menghapus data stok inventori manual maupun secara otomatis dengan <i>excel</i> dan memilih kategori yang akan di kalkulasi.
F004	Sistem mengirimkan email pemberitahuan kalkulasi sesudah admin memasukkan data inventori dan memilih kategori.
F005	<i>Warehouse</i> tim dapat melakukan kalkulasi sesuai pemberitahuan email.
F006	Admin dan <i>warehouse</i> tim dapat mencetak laporan hasil kalkulasi dan mencetak laporan sebagai PDF.
F007	Sistem dapat mengkalkulasikan kesamaan stok aktual yang di masukkan dengan stok yang ada di sistem.
F008	Sistem dapat otomatis memberi email pemberitahuan kepada pimpinan jika terjadi ketidaksesuaian data stok aktual dengan data stok sistem.
F009	Sistem dapat memasukkan email pimpinan dan mengubah identitas <i>role</i> .

B.3 Rancangan Kebutuhan NonFungsional

Selain kebutuhan fungsional, sistem ini memiliki beberapa rancangan kebutuhan nonfungsional.

TABEL IV
Tabel Kebutuhan NonFungsional

No	Kebutuhan NonFungsional
NF001	Sistem dibangun dalam bentuk website <i>responsive</i> .
NF002	Sistem memiliki antar muka yang <i>user friendly</i> .
NF003	Hanya admin yang mempunyai hak untuk mengubah, mengedit dan menghapus data serta memberi akses.

B.4 Rancangan Tahapan *Build*

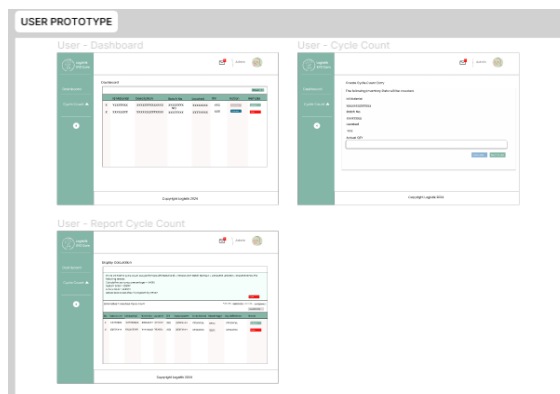
TABEL V
Tabel Rancangan *Build* Program

No	Keterangan
<i>Build 1</i>	Sistem dapat memasukkan data inventori ke dalam sistem dengan manual proses maupun otomatis proses dengan <i>excel</i> .
<i>Build 2</i>	Sistem mampu melakukan kalkulasi dengan baik.
<i>Build 3</i>	Sistem mampu melakukan email otomatis serta laporan hasil kalkulasi dalam bentuk dokumen.
<i>Build 4</i>	Sistem dapat memberikan alarm berupa email sesuai jadwal kalkulasi yang telah di tentukan. Dan mampu memberikan otorisasi pada setiap fitur.

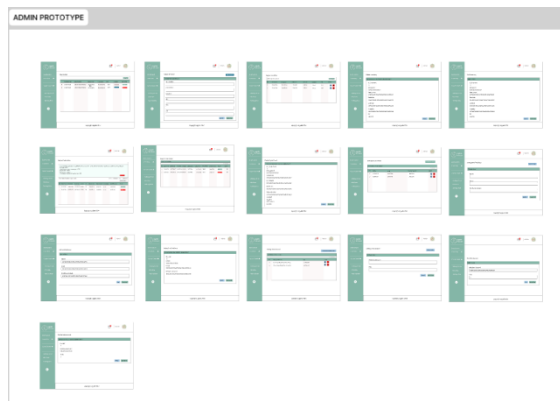
Berdasarkan Tabel V, rancangan *build* program. Hasil *build 1* dapat dilihat sesuai Gambar 11, hasil *build 2* dapat dilihat sesuai Gambar 9 & 10, hasil *build 3* dapat dilihat sesuai Gambar 12 untuk email laporan hasil kalkulasi, serta hasil *build 4* dapat dilihat sesuai Gambar 13 untuk email jadwal kalkulasi.

C. Tahapan *Architecture Design*

Pada tahap ini pengembang berfokus pada pembuatan design dan pengembangan *prototype*. Pengembangan *prototype* sistem ini terdapat meliputi halaman *warehouse* tim dan admin sesuai Gambar 2 & 3.



Gambar 2. *Prototype User Cycle Count Inspection System*



Gambar 3. *Prototype Admin Cycle Count Inspection System*

D. Tahapan *Development*

Pada tahap ini dilakukan pengembangan sistem dengan menyesuaikan bahasa pemrograman, fitur, dan analisis kebutuhan yang sudah di tentukan sebelumnya. Hasil dari pengembangan sistem ini dapat dilihat pada Gambar 8-13.

E. Tahapan *Testing*

Setelah selesai dalam tahap *development*, diperlukan *testing* atau pengujian agar dapat menentukan apakah sistem yang dikembangkan dapat berkerja sebagaimana semestinya atau tidak. Hasil pegujian sistem dapat dilihat pada Tabel VII.

F. Tahapan *Implementation*

Tahapan ini merupakan tahapan implemantasi sistem. Dimana sistem sudah di *hosting* dan bisa digunakan. Dalam sistem ini, tahapan implementasi dilakukan setelah semua *build* dan *testing* dilakukan.

Tahapan-tahapan diatas dilakukan secara berurutan. Pengembangan sistem yang sudah selesai akan dilakukan *testing* dan kemudian di diskusikan kepada *user*, apakah memerlukan perubahan atau sudah sesuai.

IV. ANALISIS DAN PERANCANGAN

Cycle Count Inspection System merupakan sistem *cycle counting* data stok inventori yang memiliki 2 jenis aktor yaitu, *warehouse* tim sebagai *user* dan admin sebagai aktor yang dapat mengelola semua fitur. *Warehouse* tim akan memiliki peran dengan hak akses 2 sedangkan admin memiliki peran dengan hak akses 1.

TABEL VI
Tabel Fitur dan *Role*

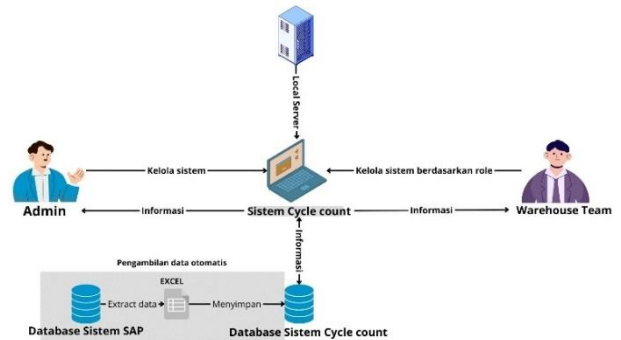
No	Fitur Sistem	Role	
		Warehouse tim	Admin
1	Dashboard		
2.	Inventory		
	a. Create Inventory		
	b. Display Inventory		
3.	Cycle Count		
	a. Check Calculation		
	b. Report Calculation		
	c. Display Calculation		
4.	Setting Email Address		
5.	Setting Account		
6.	Setting Role Account		

Pada Tabel VI *Warehouse* tim dan Admin memiliki perizinan berbeda-beda untuk menjalankan fitur-fitur tertentu. Di sistem ini, *Warehouse* tim hanya diberikan hak akses untuk mengkalkulasi data dalam fitur “Check

Calculation” dan mencetak laporan kalkulasi melalui fitur “Report Calculation”. Kalkulasi dilakukan dengan memasukkan data aktual dan menekan tombol “Calculate”. Setelah berhasil, *warehouse* tim akan diarahkan ke fitur “Report Calculation” dimana mereka dapat melihat hasil kalkulasi serta dapat mencetak laporan hasil kalkulasi. Fitur ini memiliki 2 kondisi, jika hasil kalkulasi adalah 100%, maka hasil akan menunjukkan bahwa kalkulasi berhasil. Namun jika hasil kalkulasi berada dalam rentang $\pm 100\%$, maka hasil menunjukkan bahwa kalkulasi tidak berhasil. Selanjutnya sistem akan mengirimkan email pemberitahuan kepada pimpinan terkait kesalahan kalkulasi, dan hasil yang salah akan dikalkulasi ulang. *Warehouse* tim tidak memiliki akses untuk melihat data inventori karena merupakan bentuk keamanan data kalkulasi sistem.

Admin memiliki hak akses pada semua fitur kecuali pada fitur kalkulasi. Admin berperan untuk menambahkan dan mengelola data inventori pada fitur “Create Inventory” dan “Display Inventory”. Kemudian admin memiliki akses untuk melihat dan mencetak hasil kalkulasi pada fitur “Report Calculation”. Terakhir, admin memiliki akses untuk mengelola *role* akun, *role* sistem serta data email. Untuk mempermudah analisis, berikut ini gambar diagram dari sistem *cycle count*.

A. Gambaran Umum Sistem



Gambar 4. Gambaran Umum *Cycle Count Inspection System*

Sesuai Gambar 4, sistem memiliki 2 aktor, keduanya dapat mengelola sistem berdasarkan *role* masing- masing. Data inventori didapat dari sistem SAP perusahaan. Admin akan menarik data saat jadwal kalkulasi tiba. Data tersebut kemudian masuk ke dalam *database cycle count*, dimana sistem akan menampilkan dan memungkinkan kedua pengguna untuk dapat menggunakannya.

B. Use Case Diagram

Pada Gambar 5, sistem memiliki 2 aktor yang dapat mengakses fitur setelah melalui otentikasi dan otorisasi (Otomatis masuk jika akun PC terdaftar). Sistem akan memeriksa apakah akun PC sudah terdaftar dan memiliki hak akses. Aktor pertama adalah *warehouse* tim, dimana mereka dapat mengkalkulasikan data inventori, melihat serta

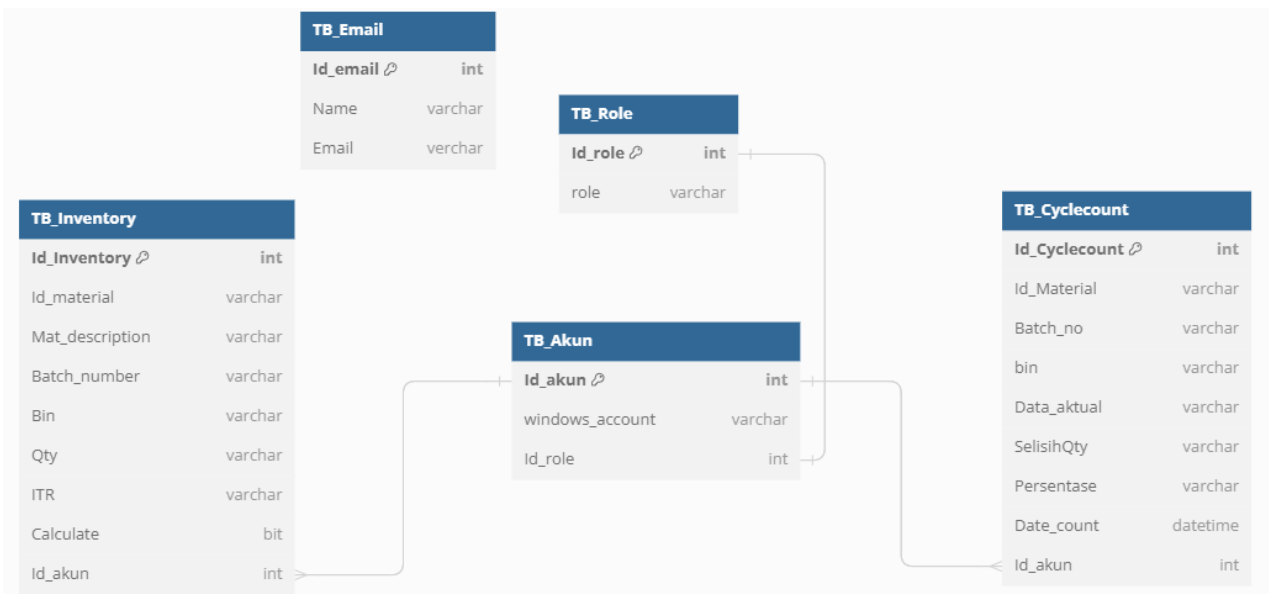
mencetak laporan hasil kalkulasi. Sedangkan aktor kedua yaitu Admin memiliki hak akses untuk menambah, mengubah dan menghapus data, melihat dan mencetak hasil kalkulasi, memilih kategori yang akan dikalkulasikan, mengelola *role* akun, *role* sistem serta data email.



Gambar 5. Use Case Cycle Count Inspection System

C. Skema Relational

Sesuai Gambar 6, tabel pertama adalah TB_Role, berfungsi mendeskripsikan peran akun dan menentukan hak akses pengguna. Di tabel ini, “Id_role” bertindak sebagai primary key. Kedua yaitu TB_Akun, sebagai tempat penyimpanan akun pengguna dan hak akses pengguna. Di tabel ini, “Id_akun” bertindak sebagai *primary key* sedangkan “Id_role” bertindak sebagai *foreign key*. Dimana keduanya berelasi *one to one*, yang berarti 1 akun PC terdaftar hanya memiliki 1 hak akses dalam sistem. Ketiga adalah TB_Inventory, berfungsi sebagai tempat penyimpanan data inventori. Di tabel ini, “Id_inventory” bertindak sebagai *primary key* sedangkan “Id_akun” bertindak sebagai *foreign key*. Dimana keduanya berelasi *one to many*, yang artinya 1 pengguna dapat memasukkan lebih dari 1 data inventori. Keempat yaitu TB_Cyclecount, berfungsi untuk menyimpan hasil kalkulasi. Di tabel ini, “Id_cyclecount” bertindak sebagai *primary key* dan “Id_role” yang bertindak sebagai *foreign key*. Keduanya berelasi *one to many*, artinya 1 akun dapat melakukan lebih dari sekali kalkulasi jika hasil kalkulasi sebelumnya salah. Terakhir adalah TB_Email, berfungsi sebagai tempat penyimpanan data email pimpinan yang akan digunakan sebagai alamat email pemberitahuan jika kalkulasi salah. Di tabel ini “Id_Email” bertindak sebagai *primary key* dan tidak berelasi dengan tabel manapun



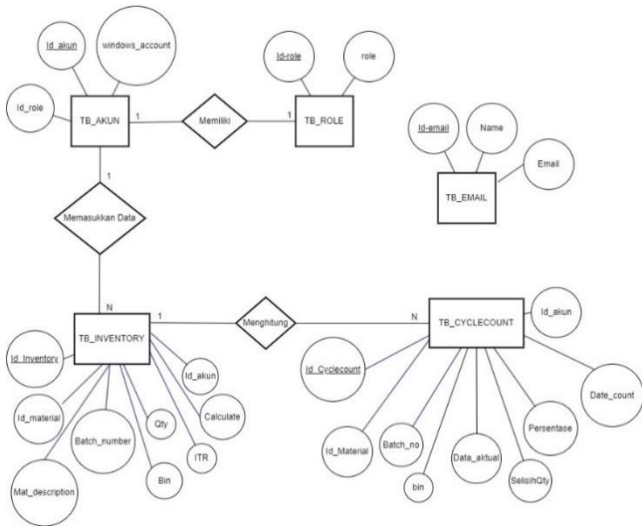
Gambar 6. Skema Relational Cycle Count Inspection System

D. ER Diagram

Pada gambar 7 menunjukkan bahwa setiap tabel memiliki entitas masing-masing. TB_role dan TB_Akun berelasi *one to one*, TB_Akun dengan TB_Inventory berelasi *one to many* begitu pula dengan TB_Akun dengan

TB_Cyclecount. Pada gambar 5 juga menjelaskan bahwa TB_akun memiliki hak akses yang sudah didefinisikan pada TB_Role. Ketika PC pengguna sudah memiliki hak akses, maka pengguna dapat menjalankan sistem sesuai hak akses yang sudah ditentukan. TB_Akun sebagai tempat untuk

mendaftarkan hak akses pengguna yang kemudian dapat menambahkan data inventori ke dalam TB_Inventori. Dari data inventori tersebut, pengguna dapat mengkalkulasikan data yang kemudian disimpan ke dalam TB_Cyclecount. Ketika hasil kalkulasi tidak berhasil, maka sistem akan mengirim email laporan ke pimpinan. Alamat akun email pimpinan didapat dari kumpulan email pada TB_Email.



Gambar 7. ER Diagram Cycle Count Inspection System

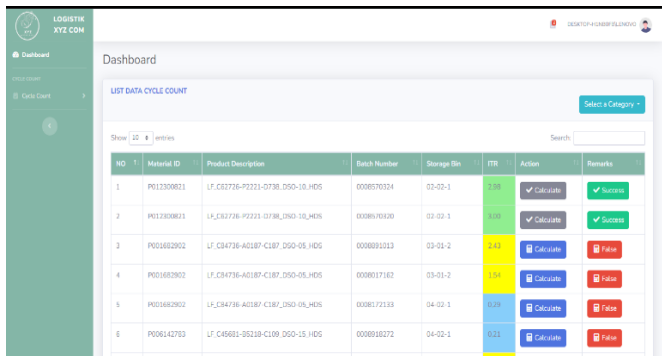
V. HASIL DAN PEMBAHASAN

A. Implementasi Antarmuka

Sistem ini memiliki dua aktor yang memiliki hak akses fitur dan halaman antar muka yang berbeda. Sistem ini memiliki halaman *user* (warehouse tim) dan halaman admin.

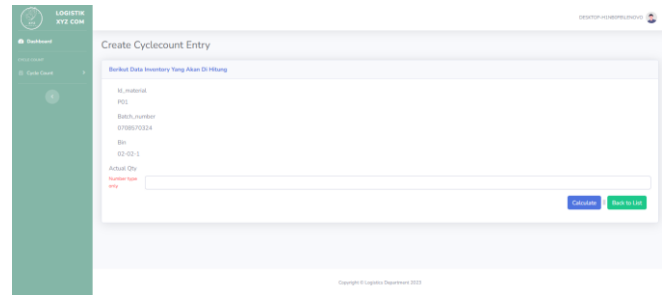
1) Halaman Warehouse tim

Pada halaman *warehouse* tim terdapat 2 fitur yaitu *dashboard* dan kalkulasi. Di halaman ini kalkulasi merupakan salah satu fitur utama sistem yang hanya dapat diakses oleh *warehouse* tim saja.



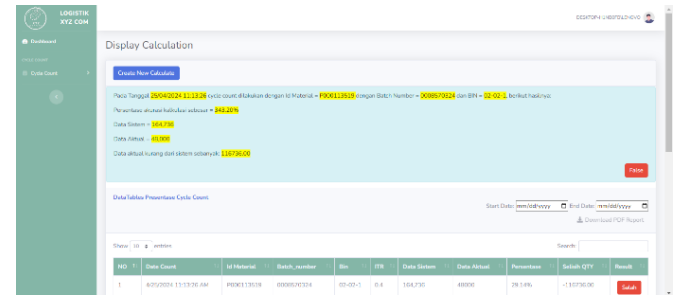
Gambar 8. Dashboard Warehouse tim

Pada halaman ini, *warehouse* tim memilih tombol "Calculate", lalu masukkan nilai data aktual kemudian tekan tombol "Calculate" kembali untuk melihat hasilnya.



Gambar 9. Fitur Create Cycle Count Warehouse tim

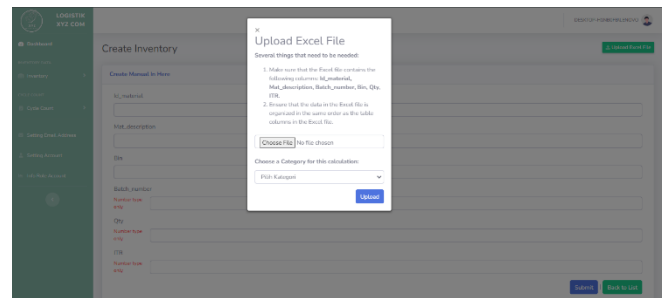
Setelah selesai kalkulasi, *warehouse* tim dapat melihat hasil kalkulasi pada halaman "Report Calculation" seperti Gambar 10.



Gambar 10. Fitur Report Cycle Count Warehouse tim

2) Halaman Admin

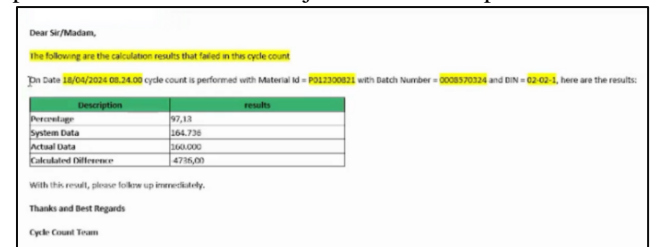
Pada halaman ini admin dapat mengelola sebagian besar fitur, namun admin tidak dapat mengakses fitur kalkulasi. Fitur inventori merupakan fitur utama yang hanya dapat diakses oleh admin saja untuk memasukkan data ke sistem. Admin memiliki 2 cara untuk memasukkan data inventori yaitu dengan memasukkan data secara manual dan masukkan data melalui *excel* yang terlihat pada Gambar 11.



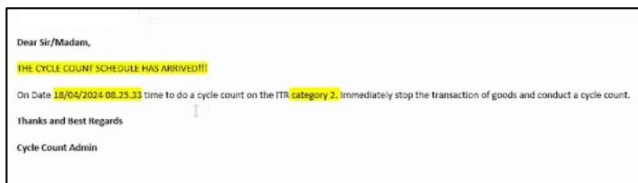
Gambar 11. Fitur Inventori Admin

3) Halaman Email

Halaman ini menunjukkan antarmuka pada saat pimpinan dan pengguna menerima email kalkulasi yang tidak berhasil pada Gambar 12 dan email jadwal kalkulasi pada Gambar 13.



Gambar 12. Email Laporan Kalkulasi Tidak Berhasil



Gambar 13. Email Jadwal Kalkulasi

B. Pengujian

Dalam tahap ini, sistem diuji menggunakan *Alpha testing*. *Alpha testing* atau pengujian *alpha* merupakan pengujian untuk memahami perilaku dan melakukan pengalaman pengguna pada sistem. Pengujian ini biasanya dilakukan oleh tim *developer* atau karyawan yang bertujuan

mencoba meniru 80% perilaku pengguna. Pengujian ini dilakukan sebelum sistem di rilis ke pengguna atau sebelum sistem ini siap di pakai. Hal ini dilakukan untuk mengidentifikasi *error* atau *bug* yang mungkin akan muncul [16].

Pengujian *alpha* pada sistem ini menggunakan metode *black box*. Metode *black box* merupakan salah satu metode penelitian perangkat lunak yang memeriksa spesifikasi perangkat lunak yang berdasarkan pada fungsional sistem untuk melihat apakah sistem mampu menghasilkan luaran yang di harapkan dan sesuai dengan fungsi sistem. Pengujian ini dilakukakan oleh penulis dengan mencoba meniru bagaimana perilaku pengguna pada saat memakai sistem [17]. Pada sistem ini, pengujian di dasarnya pada *use case* diagram guna untuk memeriksa fungsional sistem.

TABEL VII
Tabel Pengujian *Alpha Testing*

No	Use Case	Skenario	Yang diharapkan	Kesimpulan
1.	Mengelola data inventori (Admin)	1) Untuk menambahkan data, pengguna masuk ke dalam fitur “Create Inventory” dan masukkan data manual atau melalui file <i>excel</i> . 2) Untuk menghapus salah satu data inventori. 3) Menghapus semua data inventori 4) Mengubah data inventori	1) Sistem dapat menyimpan data <i>excel</i> ke dalam <i>database</i> dan akan menampilkan <i>error</i> jika <i>excel</i> masih ada yang kosong. 2) Untuk menghapus data inventori satu persatu, sistem menampilkan halaman detail inventori yang akan di hapus, setelah berhasil menghapus maka sistem akan kembali ke halaman sebelumnya. 3) Untuk menghapus data inventori sekaligus, sistem memunculkan <i>pop-up</i> pemberitahuan “apakah anda yakin ingin menghapus semua data”. Kemudian kembali ke halaman sebelumnya setelah berhasil menghapus data. 4) Sistem dapat mengubah data inventori di <i>database</i> .	Berhasil
2.	Mengkalkulasikan data inventori (User)	1) Pada halaman <i>dashboard</i> , tekan tombol “Calculate” pada salah satu tabel yang kemudian sistem akan mengarahkan ke halaman “Create Cycle Count”. Masukkan data aktual lalu tekan tombol “Calculate” untuk mengkalkulasikan data inventori. 2) Untuk admin, ketika halaman <i>dashboard</i> memilih “Calculate” pada salah satu tabel maka akan memunculkan <i>error</i> .	1) Sistem dapat mengkalkulasikan data. Dengan memasukkan aktual. Data tersebut kemudian akan dikurangkan dengan data inventori yang sudah ada di sistem. Jika sistem menemukan kesalahan, maka kesalahan perhitungan akan diberitahukan melalui <i>email</i> . 2) Jika akun pengguna tidak memiliki akses ke fitur kalkulasi, maka sistem akan memunculkan <i>error</i> 403 terkait hak akses.	Berhasil
3.	Melihat hasil kalkulasi (Admin dan User)	1) Untuk hak akses <i>user</i> , ketika sudah selesai melakukan kalkulasi, maka sistem akan mengarahkan ke halaman “Report Calculation” untuk melihat hasil kalkulasi. 2) Untuk hak akses admin, pengguna dapat melihat hasil kalkulasi dengan memilih fitur “Report Calculation”.	Sistem akan menampilkan hasil kalkulasi pada halaman “Report Calculation”. Jika hasil kalkulasi tidak berhasil, maka halaman tersebut akan menampilkan tombol dengan indikasi “salah” begitupula sebaliknya.	Berhasil
4.	Menghapus hasil kalkulasi (Admin)	Pada halaman “Display Inventory”, admin dapat memilih “delete” pada salah satu tabel untuk menghapusnya.	Sistem akan menampilkan halaman detail data yang akan di hapus untuk konfirmasi kembali apakah benar ingin menghapus data yang telah di pilih. Ketika sudah berhasil menghapus, maka	Berhasil

			sistem akan mengarahkan pengguna pada halaman sebelumnya.	
5.	Mencetak laporan hasil kalkulasi (Admin dan User)	Pada halaman "Report Calculation", pengguna masukkan "Tanggal awal" dan "Tanggal akhir" lalu tekan tombol "Download" untuk mencetak.	Sistem akan mencetak data hasil kalkulasi berdasarkan masukan "Tanggal awal" dan "Tanggal akhir". Jika keduanya tidak di masukkan maka tombol "Download" tidak bisa digunakan.	Berhasil
6.	Memilih kategori ITR yang akan di email untuk kalkulasi	Pada halaman untuk memasukkan data inventori, terdapat <i>form</i> untuk memilih ITR kategori mana yang akan di kalkulasikan.	Sistem dapat mengambil nilai kategori yang telah di pilih, kemudian mengirimkan hasil pilihan tersebut melalui email sebagai alarm/notifikasi jadwal kalkulasi. Jika ITR kosong, maka akan muncul <i>error</i> pada <i>form</i> "Data harus di isi".	Berhasil
7.	Mengelola akun pengguna (Admin)	1) Untuk menambahkan data, pengguna menekan tombol "Create New Akun" dan masukkan akun <i>windows</i> serta hak akses <i>role</i> nya. 2) Untuk menghapus salah satu data akun, pilih tombol "delate" pada salah satu tabel, kemudian hapus. 3) Mengubah data akun, pengguna memilih tombol "edit" untuk mengubah data.	Pada saat menambahkan data, sistem akan menyimpan data dalam <i>database</i> . Sistem mampu menghapus data akun dengan mengkonfirmasi kembali pada halaman yang menampilkan detail data yang akan di hapus. Sistem juga mampu mengubah data akun dengan menyediakan halaman untuk mengubah data.	Berhasil
8.	Mengelola data akun email (Admin)	1) Untuk menambahkan data, pengguna menekan tombol "Create New Email" dan masukkan "Name" dan "Email". 2) Untuk menghapus salah satu data email, pilih tombol "delate" pada salah satu tabel email, kemudian hapus. 3) Mengubah data email, dengan memilih tombol "edit" untuk mengubah data.	Pada saat menambahkan data, sistem akan menyimpan data dalam <i>database</i> . Sistem mampu menghapus data email dengan konfirmasi kembali pada halaman yang menampilkan detail data yang akan di hapus. Sistem juga mampu mengubah data email dengan menyediakan halaman untuk mengubah data.	Berhasil
9.	Mengelola <i>role</i> (Admin)	1) Untuk menambahkan data, pengguna menekan tombol "Create New Role" dan nama akses <i>role</i> baru. 2) Untuk menghapus salah satu data <i>role</i> , pilih tombol "delate" pada salah satu tabel <i>role</i> , kemudian hapus. 3) Mengubah data <i>role</i> , dengan memilih tombol "edit" untuk mengubah data.	Pada saat menambahkan data, sistem akan menyimpan data dalam <i>database</i> . Sistem mampu menghapus data <i>role</i> dengan konfirmasi kembali pada halaman yang menampilkan detail data yang akan di hapus Sistem juga mampu mengubah data <i>role</i> dengan menyediakan halaman untuk mengubah data.	Berhasil
10.	Otentikasi & Otorisasi	Masuk ke dalam link web.	Sistem akan mengkonfirmasi nama akun PC yang akan masuk. Jika akun sudah terdaftar, maka akun tersebut masuk ke dalam <i>dashboard</i> sesuai dengan fitur hak aksesnya. Jika tidak, maka akan menampilkan <i>error</i> 403 terkait akses.	Berhasil

VI. KESIMPULAN

Cycle Count Inspection System merupakan sistem *cycle counting* yang membantu melakukan perhitungan akurasi data stok inventori agar dapat meminimalisir adanya perbedaan data stok inventori. Selain mengacu pada ITR, sistem ini juga mengacu pada penerapan metode FSN *analysis* sesuai dengan standar operasional prosedur perusahaan untuk mengatur jadwal kalkulasi serta mengklasifikasikan inventori berdasarkan nilai ITR pada tiap inventori.

Berdasarkan pemaparan yang penulis berikan dan hasil pengujian sistem yang berjalan sesuai yang diharapkan. Dapat disimpulkan bahwa sistem ini mampu mengkalkulasikan data serta mengidentifikasi kesalahan kalkulasi, selain itu sistem juga mampu mengingatkan pengguna jika sudah waktunya melakukan kalkulasi, sistem mampu melaporkan dengan email jika terdapat kesalahan kalkulasi. Sistem ini mampu mengubah kalkulasi yang sebelumnya manual menjadi kalkulasi otomatis berbasis website. Dengan demikian, sistem ini dapat digunakan sebagai fasilitas elektronik untuk perhitungan inventori dan diharapkan dapat menjadi jawaban atas permasalahan perbedaan pada stok inventori.

VII. SARAN

Secara keseluruhan, sistem ini sudah memenuhi fitur yang diharapkan oleh perusahaan. Namun, masih perlu adanya pengembangan pada beberapa fitur-fitur tertentu. Pengembangan sangat dianjurkan untuk memperhatikan aspek kinerja, kegunaan, dan keamanan dalam sistem. Untuk pengembangan sistem selanjutnya, diharapkan dapat mengembangkan fitur berupa notifikasi jadwal kalkulasi otomatis yang mengingatkan pengguna berdasarkan tanggal yang sudah dijadwalkan dan mengembangkan proses penarikan data inventori dari sistem SAP perusahaan secara otomatis agar dapat lebih meningkatkan fungsionalitas sistem. Dengan demikian, sistem akan menjadi lebih kompleks serta dapat meningkatkan kemudahan dan kenyamanan bagi pengguna.

DAFTAR PUSTAKA

- [1] BP BATAM, "Batam: Kota Industri dan Perkembangannya di Masa Mendatang," bpbatam.go.id. Accessed: Mar. 29, 2024. [Online]. Available: <https://bpbatam.go.id/batam-kota-industri/>
- [2] Wiresix, "Divisi Logistik: Pengertian, Tugas, dan Kisaran Gajinya," <https://www.waresix.com/>. Accessed: Mar. 29, 2024. [Online]. Available: <https://www.waresix.com/id/divisi-logistik-adalah/>
- [3] N. widya Utami, "Mengenal Sistem Inventory & Cara Mudah Pengelolaannya," *Supply Chain Management*. Accessed: Mar. 29, 2024. [Online]. Available: <https://www.jurnal.id/id/blog/2018-mengenal-sistem-inventory-dan-cara-mudah-pengelolaannya/>
- [4] M. A. Puteri, M. P. Zabina, and E. Triputra, "Telaah Sistem Manajemen Pergudangan Dalam Berbagai Metode Inventory," *Ris. Sains dan Teknol. Kelaut.*, vol. 6, no. 1, pp. 40–46, 2023, doi: 10.62012/sensistek.v6i1.24246.
- [5] M. R. Wardana and Y. Sukmono, "Perancangan Cycle Inventory Policy Menggunakan Metode Cycle Counting Pada Gudang Pt. Badak Ngl," *Pros. Semin. Nas.*, pp. 78–87, 2019, [Online]. Available: <http://e-journals.unmul.ac.id/index.php/SEMNASTEK/article/view/2796>
- [6] M. Zhafari, "RANCANG BANGUN SISTEM INFORMASI INVENTORY BERBASIS WEB MENGGUNAKAN METODE KLASIFIKASI ABC," p. 371, 2022.
- [7] E. Susilowati, "Perancangan Sistem Inventory Berbasis Web Studi Kasus Ud. Berkah Ananda," *J. Esensi Infokom*, vol. 2, no. 1, pp. 54–63, 2018.
- [8] F. N. Christian, "Rancang Bangun Sistem Warehouse Multi Cabang Dengan Metode Klasifikasi ABC Berbasis Website Pada PT Young Multi Sarana.," no. 8.5.2017, pp. 2003–2005, 2022.
- [9] L. Schwarz, "Inventory Cycle Counting 101: Best Practices & Benefits," <https://www.netsuite.com/>. [Online]. Available: <https://www.netsuite.com/portal/resource/articles/inventory-management/using-inventory-control-software-for-cycle-counting.shtml>
- [10] D. Devarajan and M. S. Jayamohan, "Stock control in a Chemical Firm: Combined FSN and XYZ Analysis," *Procedia Technol.*, vol. 24, pp. 562–567, 2016, doi: 10.1016/j.protcy.2016.05.111.
- [11] Niko, "Cara Menghitung Rasio Perputaran Persediaan (Inventory Turnover)," <https://www.jurnal.id/>. Accessed: Nov. 11, 2023. [Online]. Available: <https://www.jurnal.id/id/blog/rasio-perputaran-persediaan/>
- [12] Microsoft, "What is .NET Framework?," <https://dotnet.microsoft.com/>. [Online]. Available: <https://dotnet.microsoft.com/en-us/learn/dotnet/what-is-dotnet-framework>
- [13] Microsoft, "Ringkasan ASP.NET." [Online]. Available: <https://learn.microsoft.com/id-id/aspnet/overview>
- [14] M. Salsabila, "Apa Itu SQL Server? Yuk Kulik Kelebihan dan Kekurangannya." [Online]. Available: <https://dqlab.id/apa-itu-sql-server-yuk-kulik-kelebihan-dan-kekurangannya>
- [15] A. K. Ilham, "TENTANG INCREMENTAL MODEL," *Brain Strom*. [Online]. Available: <https://ilhamajji.blogspot.com/2014/11/tentang-incremental-model.html>
- [16] A. S. Oktriwina, "Mengenal Alpha Testing, Tahap Pengujian Pertama sebelum Aplikasi Diluncurkan," <https://glints.com/>. Accessed: Mar. 31, 2024. [Online]. Available: <https://glints.com/id/lowongan/alpha-testing/>
- [17] B. A. B. Iv and P. D. A. N. Analisis, "Bab iv pengujian dan analisis," pp. 65–83.