

Optimasi Pengendalian Persediaan dengan Metode Reorder Point dalam Pengembangan Aplikasi Kontrol Stok Berbasis Web

Rakhmad Maulidi^{1*}, Prima Listianti^{2*}

* Teknik Informatika, Sekolah Tinggi Informatika & Komputer Indonesia (STIKI) Malang

maulidi@stiki.ac.id¹, 181116012@mhs.stiki.ac.id²

Article Info

Article history:

Received 2023-02-21

Revised 2023-07-02

Accepted 2023-07-19

Keyword:

Stock Control,
Reorder Point,
SDLC,
Prototype,
Blackbox Testing

ABSTRACT

Stock control is a process of keeping track of owned items, their location, and their movements in and out of storage. It helps businesses to manage their inventory efficiently and meet consumer demand while reducing the cost of storing goods. However, online shops like Omah Mode often struggle to meet the demand of consumers due to the lack of a stock control system, resulting in penalties from the marketplace. To solve this problem, a web-based stock control application was designed using the reorder point method. This method determines the minimum stock of goods that should be in the warehouse and the right time to order goods with low stock from suppliers. The purpose of this research is to develop an inventory control system that displays the minimum stock of goods needed in the warehouse. The study used the SDLC Prototype method, consisting of needs analysis, prototype making, prototype evaluation, system coding, system testing, system evaluation, and system use. The results showed that the system created has an 85.71% accuracy rate based on a comparison of manual calculations and system calculations of 28 sample items.



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

I. PENDAHULUAN

Salah satu hal yang diperlukan saat ini adalah teknologi informasi. Segala pekerjaan manusia lebih efektif dengan adanya teknologi informasi. Salah satunya contohnya adalah aplikasi. Aplikasi adalah program perangkat lunak yang berisi kode atau instruksi yang bisa diubah mengikuti kebutuhan [1]. Sistem kontrol stok barang adalah salah satu aplikasi yang berguna saat ini. Kontrol stok barang berarti mengetahui barang apa saja yang dimiliki, dimana barang tersebut disimpan dan kapan barang tersebut keluar dan masuk. Jika bisnis memiliki kontrol stok yang ketat, maka permintaan konsumen akan terpenuhi.

Omah Mode (OMode) merupakan salah satu online shop yang bergerak dalam bidang penjualan perlengkapan TNI / POLRI. Barang yang dijual oleh Omah Mode sangat bervariasi dan jumlahnya juga banyak. Akan tetapi OMode belum memiliki sistem untuk mengontrol stok barang. Sehingga, seringkali tidak bisa memenuhi tingginya permintaan pelanggan dan terkena penalti dari marketplace. Selain itu, omah mode juga tidak memiliki pencatatan yang baik untuk keluar masuknya barang.

Metode reorder point dapat menyelesaikan permasalahan penentuan minimal persediaan suatu barang untuk dilakukan pemesanan kembali, menurut penelitian tentang sistem persediaan stok barang berbasis web ini efektif digunakan untuk melakukan pendataan keluar masuknya barang dan ketersediaan stok akurat [2]. Penelitian tentang perhitungan pemesanan produk dengan menggunakan reorder point dapat meningkatkan jumlah permintaan karena adanya safety stock, permintaan pelanggan selalu terpenuhi dan ketersediaan barang selalu ada. [3]. Penelitian tentang pengembangan aplikasi proses perhitungan persediaan stok aman barang dan titik kapan barang harus kembali dipesan menggunakan metode safety stock dan reorder point memberikan kemudahan bagi pengguna [4]. Penggunaan Metode reorder point dan waterfall untuk mengembangkan aplikasi berbasis website setelah dilakukan uji fungsionalitas menggunakan blackbox testing, menunjukkan hasil berjalan dengan baik dan memberikan informasi tentang persediaan barang berupa jumlah stok, safety stock, reorder point [5]. Metode reorder point dapat dijadikan solusi untuk mengontrol persediaan barang dan menentukan titik pemesanan kembali dan safety stock [6].

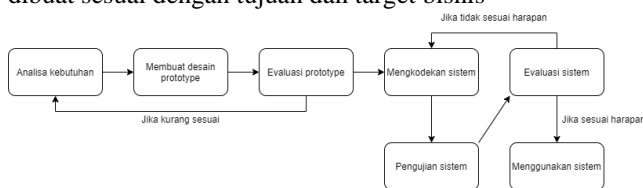
Berdasarkan permasalahan tersebut, dibuat aplikasi kontrol stok barang menggunakan metode reorder point. Reorder point adalah ketersediaan barang yang harus tetap ada saat pemesanan dilakukan atau disebut dengan titik pemesanan Kembali (*Reorder Point*) [7]. Metode ini dipilih karena bisa menentukan minimal stok yang harus ada di gudang dan waktu yang tepat untuk memesan kembali barang yang stoknya sudah menipis.

II. METODE

SDLC (Software Development Life Cycle) membantu dalam memastikan pengembangan perangkat lunak yang terorganisir dan terstruktur dengan tujuan akhir menghasilkan produk berkualitas tinggi yang memenuhi kebutuhan pelanggan. Keuntungan dari SDLC meliputi struktur yang teratur, pengendalian risiko yang lebih baik, peningkatan kualitas perangkat lunak, pengendalian biaya yang lebih efektif, dan komunikasi yang lebih baik. Dengan SDLC, tim pengembang dapat mengelola proyek dengan lebih efisien dan menjaga hubungan yang baik dengan stakeholder, sehingga memastikan proyek berjalan dengan baik [8]. Beberapa Metode SDLC yang biasa digunakan dalam proses pengembangan perangkat lunak diantaranya *prototype* [9], *waterfall* [10], *Scrum* [11], *Extreme Programming* (XP) [12].

Model SDLC yang digunakan dalam penelitian ini adalah *prototype*, model skemanya seperti pada Gambar 1 Metode Prototype

Model *prototype* memungkinkan pengembang dan pengguna untuk memahami lebih cepat tentang apa yang mereka inginkan dari perangkat lunak. Hal ini dapat dicapai dengan membuat prototipe yang dapat diuji coba dan diberikan umpan balik oleh pengguna sebelum perangkat lunak yang sebenarnya dibuat. Selain itu, model *prototype* juga memungkinkan pengembang untuk lebih fleksibel dan adaptif dalam mengubah kebutuhan pengguna atau lingkungan bisnis yang terus berubah-ubah [9]. Dengan melakukan iterasi dan revisi pada prototipe, pengembang dapat menyesuaikan perangkat lunak dengan kebutuhan pengguna dan memastikan bahwa perangkat lunak yang dibuat sesuai dengan tujuan dan target bisnis



Gambar 1 Metode Prototype

Pada metode ini, pengguna memiliki gambaran tentang aplikasi yang akan dikembangkan dan dapat melakukan pengujian aplikasi sebelum dirilis. Tahapan dari *prototype* antara lain:

1) Analisa kebutuhan

Pada tahapan ini, peneliti melakukan wawancara kepada pengguna untuk mengumpulkan kebutuhan pengguna dalam

upaya memahami persyaratan fungsional dan non-fungsional yang diperlukan dalam perangkat lunak. Selain itu juga melakukan penetapan lingkup perangkat lunak untuk memastikan bahwa prototipe fokus pada fungsi penting dan kritis.

2) Membuat prototype

Pada tahapan ini, dilakukan perancangan aplikasi sesuai dengan hasil Analisa kebutuhan. Pembuatan desain awal untuk menggambarkan arsitektur perangkat lunak dan memastikan bahwa kebutuhan pengguna dapat terpenuhi [9]. Dilanjutkan dengan membuat *prototype* dengan menggunakan teknik pengembangan cepat perangkat lunak untuk menghasilkan model awal yang dapat diuji dan dievaluasi oleh pengguna.

3) Evaluasi prototype

Pada tahapan ini, peneliti melakukan evaluasi terhadap *prototype* yang sudah dibuat dengan menguji *prototype* untuk memastikan bahwa perangkat lunak memenuhi persyaratan pengguna dan memperbaiki masalah yang teridentifikasi [9]. Kemudian dilanjutkan dengan meninjau kembali kebutuhan pengguna dan merevisi *prototype* sesuai dengan umpan balik pengguna.

4) Pengkodean sistem

Pada tahap ini, tim pengembang akan membuat kode program yang diarahkan pada tujuan tertentu, menggunakan sebuah framework PHP yakni Codeigniter. Kode program ini akan digunakan untuk mengimplementasikan fitur-fitur pada *prototype* dan menghasilkan sistem yang sesuai dengan kebutuhan pengguna [9]. Model *prototyping* biasanya melibatkan proses iterasi yang terus menerus, maka proses pengkodean sistem mungkin perlu diulang dan disesuaikan dengan umpan balik pengguna hingga sistem sudah sesuai dengan kebutuhan pengguna.

5) Pengujian sistem

Pada tahap ini, dilakukan pengujian sistem yang sudah dibuat dengan menggunakan *blackbox testing*. *Blackbox testing* adalah teknik pengujian pengembangan perangkat lunak di mana cara kerja internal perangkat lunak yang diuji tidak diketahui oleh pengujinya. Pengujian dilakukan dengan fokus pada fungsi perangkat lunak, tanpa pengetahuan terhadap kode, struktur, atau detail implementasi internalnya [11]. Jenis pengujian ini digunakan untuk memverifikasi bahwa perangkat lunak memenuhi persyaratan yang ditentukan dan berfungsi dengan baik dari perspektif pengguna akhir [10]. Metode lain yang bisa digunakan seperti *whitebox* atau *usability testing* [10]

6) Evaluasi sistem

Evaluasi sistem dilakukan pada akhir siklus pengembangan sistem. Tujuan dari evaluasi sistem adalah untuk memastikan bahwa sistem yang telah dibangun telah memenuhi kebutuhan pengguna dan dapat berfungsi dengan baik. Proses evaluasi sistem pada model *prototyping* di SDLC dilakukan dengan mengumpulkan umpan balik dari pengguna atau stakeholder yang telah menggunakan sistem. Dalam tahap ini, pengguna atau stakeholder akan diminta untuk menguji sistem dan memberikan umpan balik. Setelah

evaluasi sistem selesai dan sistem dinyatakan siap digunakan, sehingga tahap penggunaan sistem siap dilakukan.

7) *Penggunaan sistem*

Pada tahap ini, sistem akan diinstal pada lingkungan produksi dan disiapkan untuk digunakan oleh pengguna. Selain itu, tim pengembang juga akan memberikan pelatihan kepada pengguna mengenai cara penggunaan sistem yang telah dibangun. Selain itu, proses penggunaan sistem pada model prototyping di SDLC juga dapat memunculkan kebutuhan baru yang belum teridentifikasi sebelumnya. Kebutuhan baru ini kemudian akan dijadikan masukan untuk pengembangan sistem selanjutnya atau untuk membuat sistem baru yang lebih baik.

A. *Reorder Point*

Reorder point adalah ketersediaan barang yang harus tetap ada saat pemesanan dilakukan atau disebut dengan titik pemesanan kembali[2]. Rumus dari reorder point adalah sebagai berikut:

$$Reorder\ point = (D \times L) + SS \tag{1}$$

di mana:

- 1) *D* adalah rata-rata permintaan per periode waktu (biasanya dalam satuan bulan)
- 2) *L* adalah lama waktu untuk pengiriman atau lead time dari pemasok (biasanya dalam satuan bulan)
- 3) *SS* adalah safety stock atau stok keselamatan, yaitu jumlah persediaan tambahan yang dijaga untuk mengantisipasi ketidakpastian permintaan atau keterlambatan pengiriman.

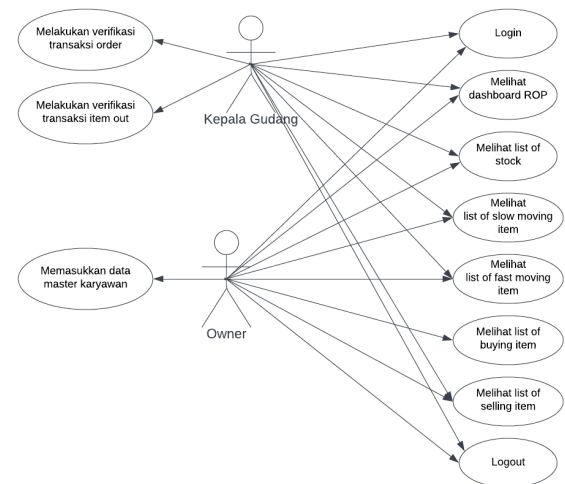
Dalam persamaan ini, (*D x L*) menunjukkan jumlah persediaan yang diperlukan untuk memenuhi permintaan selama lead time, dan *SS* ditambahkan untuk memastikan bahwa persediaan cukup untuk memenuhi permintaan selama lead time bahkan jika terjadi fluktuasi permintaan atau keterlambatan pengiriman. Oleh karena itu, ROP adalah jumlah persediaan minimum yang harus dicapai sebelum melakukan pemesanan kembali dari pemasok.

B. *Use Case Diagram*

Dalam UML (Unified Modeling Language), use case adalah teknik untuk menggambarkan interaksi antara sistem dan pengguna. Use case menjelaskan fungsionalitas sistem dari perspektif pengguna, dengan menunjukkan tindakan atau skenario yang dapat dilakukan oleh pengguna pada sistem. [13]. Sebuah diagram use case dalam UML biasanya terdiri dari dua komponen utama, yaitu aktor (actor) dan use case (use case). Aktor adalah entitas eksternal yang berinteraksi dengan sistem, sedangkan use case adalah fungsionalitas atau tindakan yang dapat dilakukan oleh pengguna pada sistem.



Gambar 2 Use Case Diagram Sistem Reorder Poin Untuk Admin

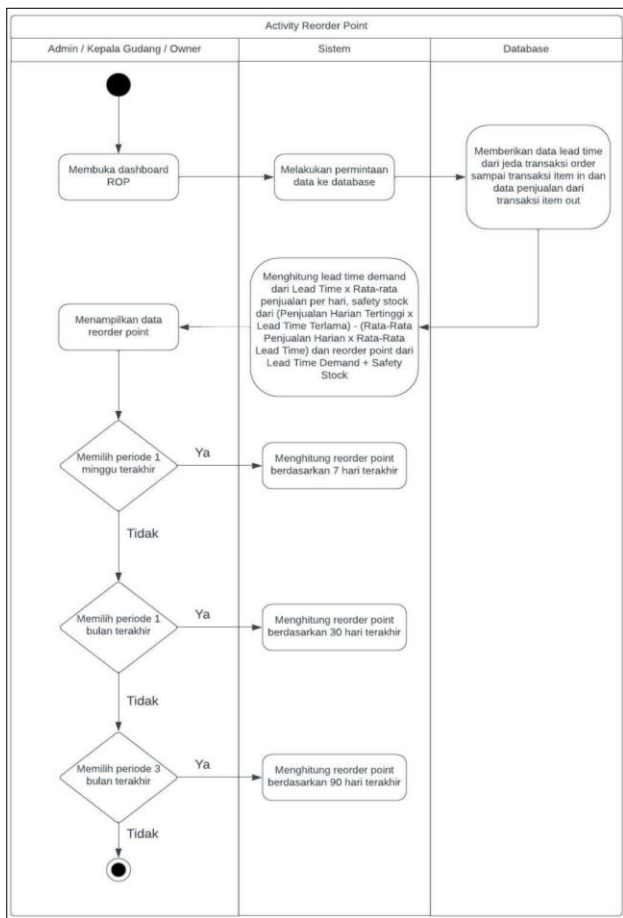


Gambar 3 Use Case Diagram Sistem Reorder Poin

Pada penelitian ini terdapat 3 aktor yaitu pemilik, kepala gudang dan admin seperti pada Gambar 2 dan Gambar 3. Setiap aktor bisa melakukan login, melihat dashboard ROP dan melihat laporan. Untuk pemilik, bisa memasukkan data master karyawan. Admin, bisa memasukkan data master selain master karyawan dan memasukkan transaksi. Sedangkan Kepala Gudang bisa melakukan verifikasi transaksi yang sudah dibuat oleh admin. Untuk menggunakan system ini setiap pengguna harus melakukan proses login untuk memastikan pengguna system adalah orang yang punya hak.

C. Activity Diagram

Activity diagram dalam UML (Unified Modeling Language) adalah salah satu jenis diagram yang digunakan untuk menggambarkan alur kerja (workflow) dari sebuah proses atau sistem. Diagram aktivitas menunjukkan serangkaian aktivitas yang terkait satu sama lain dan bagaimana aktivitas-aktivitas tersebut saling berhubungan [13]. Dalam activity diagram, aktivitas digambarkan sebagai persegi panjang, keputusan (decision) digambarkan sebagai rhombus, dan garis-garis panah yang menghubungkannya menunjukkan alur kerja. Selain itu, activity diagram dapat mencakup kondisi atau guard (biasanya digunakan dalam keputusan), input/output, dan mekanisme looping.



Gambar 4 Activity Diagram Sistem Reorder Point

Pada Gambar 4 merupakan activity diagram proses reorder point. Pada activity diagram tersebut, diawali dengan pengguna (admin / kepala gudang / owner) membuka fitur dashboard ROP. Lalu, sistem melakukan permintaan data transaksi order, item in dan item out ke database.

Setelah menerima permintaan data dari sistem, database mengirimkan data lead time yang didapatkan dari jarak transaksi order sampai transaksi item in lalu data penjualan dari transaksi item out. Data yang sudah diterima oleh sistem,

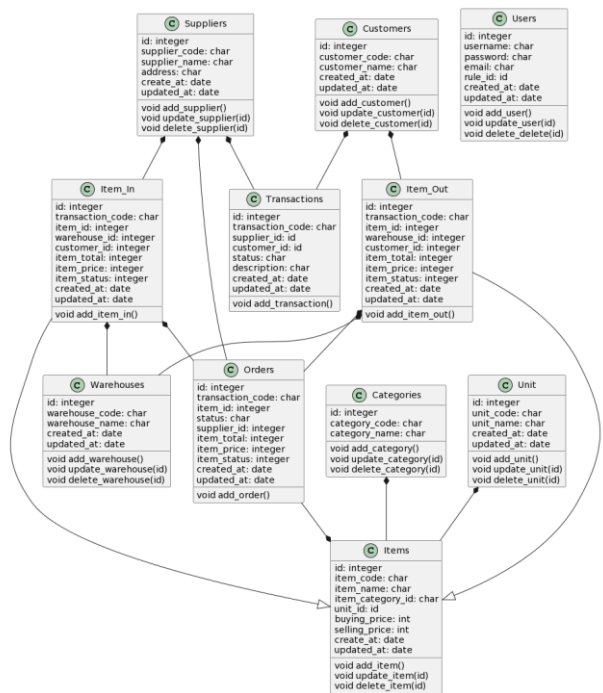
dihitung sesuai dengan rumus reorder point. Pengguna bisa memilih filter periode pada dashboard reorder point.

Jika pelanggan memilih filter 1 minggu terakhir, maka dashboard akan menampilkan data 7 hari terakhir. Jika memilih 1 bulan terakhir, maka dashboard akan menampilkan data 30 hari terakhir. Jika memilih 3 bulan terakhir, maka dashboard akan menampilkan data 90 hari terakhir.

D. Class diagram

Class diagram adalah salah satu diagram dalam UML (Unified Modeling Language) yang digunakan untuk menggambarkan struktur kelas, atribut, dan hubungan antara kelas-kelas pada suatu sistem. Class diagram menggambarkan objek-objek yang terkait dalam suatu sistem dan bagaimana objek-objek tersebut berinteraksi satu sama lain. [13].

Dalam class diagram, kelas digambarkan dalam bentuk persegi panjang dengan nama kelas di dalamnya. Atribut kelas digambarkan sebagai variabel yang terkait dengan kelas, sedangkan metode digambarkan sebagai fungsi atau operasi yang terkait dengan kelas. Hubungan antara kelas-kelas digambarkan dengan panah atau garis, seperti *association* (hubungan asosiasi), *inheritance* (pewarisan), *aggregation* (agregasi), dan *composition* (komposisi).



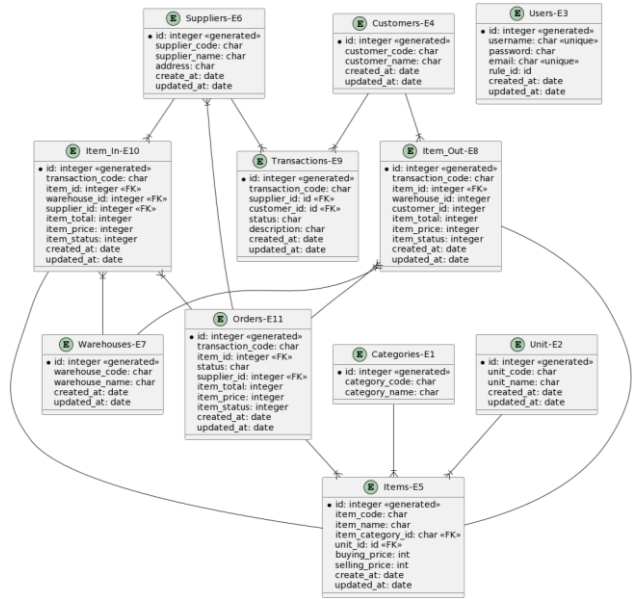
Gambar 5 Class Diagram Sistem Reorder Point

Pada Gambar 5 adalah class diagram untuk perancangan sistem ini. Terdapat 11 class yang saling berelasi diantaranya adalah class category, items, unit, users, customers, suppliers, warehouse, transactions, order, item_in dan item_out.

E. Skema basis data

Perancangan data pada sistem ini menggunakan *conceptual data model*(CDM) dan *physical data model*(PDM). CDM adalah bagian penting dalam desain sistem, karena model ini menggambarkan struktur data yang akan digunakan dalam sistem tersebut. Model ini menyajikan konsep-konsep tingkat tinggi yang ada dalam organisasi atau domain bisnis, sehingga memudahkan pemahaman tentang data apa yang harus disimpan dan bagaimana data tersebut terkait dengan data lainnya.

PDM adalah representasi konkrit dari model data yang menggambarkan bagaimana data akan disimpan dalam database pada desain sistem. Model ini meliputi detail teknis tentang struktur tabel, kolom, relasi, indeks, kunci, dan aturan validasi data. PDM menyediakan pandangan teknis tentang basis data yang akan diimplementasikan dan digunakan oleh sistem yang dirancang. PDM dibuat spesifik sesuai dengan basis data yang digunakan pada pengembangan sistem di penelitian ini yaitu *MySQL* [14].



Gambar 7 Physical Data Model Sistem Reorder Point

Physical data model pada Gambar 7 memiliki 11 tabel yang saling berelasi yaitu tabel *category*, *items*, *unit*, *users*, *customers*, *suppliers*, *warehouse*, *transactions*, *order*, *item_in* dan *item_out*. Primary key pada entitas ini memiliki tipe data *integer* yang *auto increment*

III. HASIL DAN PEMBAHASAN

A. Perhitungan Reorder Point

Penghitungan *reorder point* dilakukan manual dengan menggunakan bantuan Google Spreadsheet. Data yang dibutuhkan untuk perancangan ini adalah nama barang, periode pemesanan mulai barang dipesan ke supplier sampai barang masuk ke Gudang dan penjualan setiap barang. Contoh hasil perhitungan untuk barang PDL NINJA 1 Uk. 41 seperti yang terdapat pada Tabel 2 dan Tabel 3.

TABEL 1
PENJUALAN PDL NINJA 1 UK. 41

No.	Tanggal penjualan	Quantity penjualan	Satuan
1.	22 - 05 - 16	1	Pasang
2.	22 - 05 - 17	1	Pasang
3.	22 - 06 - 04	1	Pasang
4.	22 - 06 - 06	1	Pasang
5.	22 - 06 - 08	1	Pasang
6.	22 - 06 - 15	2	Pasang
7.	22 - 06 - 18	1	Pasang
8.	22 - 06 - 20	1	Pasang
9.	22 - 06 - 23	1	Pasang
10.	22 - 06 - 24	3	Pasang
11.	22 - 07 - 18	1	Pasang
12.	22 - 07 - 25	1	Pasang
13.	22 - 07 - 27	1	Pasang
Jumlah		16	
Rata – rata		1,230769231	
Penjualan tertinggi		3	

Gambar 6 Conceptual Data Model Sistem Reorder Point

Gambar 6 adalah *conceptual data model* pada sistem ini yang memiliki 11 entitas yang saling berhubungan yaitu entitas *category*, *items*, *unit*, *users*, *customers*, *suppliers*, *warehouse*, *transactions*, *order*, *item_in* dan *item_out*. Primary key pada entitas ini memiliki tipe data *integer* yang *auto increment*.

TABEL 2
LEAD TIME PDL NINJA 1 UK. 41

Lead Time	Lama Lead Time(hari)
Lead Time 1	7
Lead Time 2	4
Lead Time 3	10
Lead Time 4	5
Rata – rata	6,5
Lead time terlama	10

Tabel 2 berisi *lead time* yang diperoleh dari periode pemesanan barang. *Lead time* 1 merupakan pemesanan tanggal 9 Mei 2022 dan barang masuk ke Gudang tanggal 16 Mei 2022. *Lead time* 2 merupakan pemesanan tanggal 6 Juni 2022 dan barang masuk ke Gudang tanggal 10 Juni 2022. *Lead time* 3 merupakan pemesanan tanggal 6 Juli 2022 dan barang masuk ke Gudang tanggal 16 Juli 2022 dan yang terakhir *Lead time* 4 merupakan pemesanan tanggal 19 Juli 2022 dan barang masuk ke Gudang tanggal 24 Juli 2022. Berikut adalah hasil perhitungan Reorder poin.

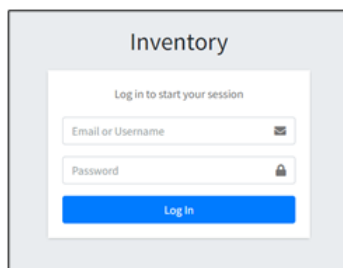
$$\begin{aligned} \text{Lead time demand} &= 5 \times 1,230769231 \\ &= 6,153846154 \\ \text{Safety Stock} &= (3 \times 10) - (1,230769231 \times 6,5) \\ &= 22 \\ \text{Reorder Point} &= 6,153846154 + 22 \\ &= 28,15384615 \\ &= 29 \end{aligned}$$

B. Implementasi Sistem

Implementasi sistem yang dihasilkan dari perancangan yang sudah dibuat, berikut ini adalah pembahasan hasil dari implementasi sistem untuk beberapa modul utama.

1) Halaman Login Sistem

Tujuan dari proses login adalah untuk memverifikasi identitas pengguna yang hendak mengakses aplikasi serta menentukan peran atau hak akses yang dimiliki oleh pengguna tersebut. Jika data pengguna tidak terdaftar dalam basis data aplikasi, maka akan ditampilkan pesan peringatan. Dalam konteks aplikasi ini Gambar 8, informasi yang diperlukan untuk login adalah nama pengguna (username) dan kata sandi (password).

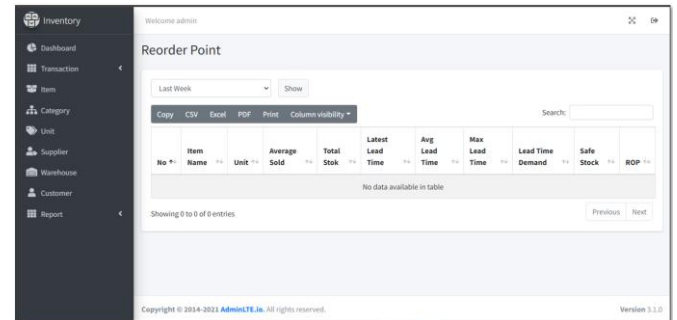


Gambar 8 Halaman Login Sistem

2) Halaman Dashboard Sistem

Halaman Dashboard memiliki peranan penting dalam sistem informasi karena memberikan tampilan visual yang

mudah dipahami mengenai performa suatu bisnis atau proses yang sedang berjalan. Dengan adanya informasi tentang penjualan rata-rata, stok barang, lead time, safety stock, dan reorder point pada satu tempat, pengguna sistem dapat dengan cepat mengetahui kondisi bisnis dan mengambil tindakan yang dibutuhkan. Selain itu, halaman Dashboard seperti pada Gambar 9 dapat membantu pengguna sistem dalam membuat keputusan bisnis yang lebih akurat dan tepat waktu.

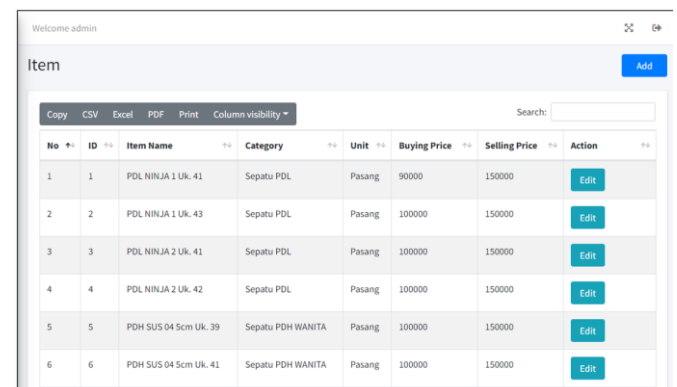


Gambar 9 Dashboard Sistem

3) Halaman Modul Item Barang

Modul ini digunakan untuk memasukkan data barang. Menu utama modul ini seperti pada Gambar 10

. Pada modul ini terdapat beberapa fitur yakni List item yang berfungsi untuk melihat daftar barang yang ada. Berikutnya fitur Add item yang berfungsi untuk menambahkan data barang baru. Selanjutnya fitur Edit item yang digunakan untuk mengubah data barang yang sudah ada. Dan yang terakhir fitur Delete item yang digunakan untuk menghapus data barang yang tidak digunakan.



Gambar 10 Modul Item Barang

4) Halaman Daftar Stok Barang

Fungsi dari halaman daftar stok barang adalah untuk menampilkan informasi terkini mengenai stok barang.

No	ID	Item Name	Category	Unit	Stok	Highest	Max Lead	Avg	AvgLead	Safety Stock	Buying Price
1		PDL NINJA 1 Uk. 41	Sepatu PDL	Pasang	5	3	10	1.3000	7	21	90000
2		PDL NINJA 1 Uk. 43	Sepatu PDL	Pasang	17		10		8.5	0	100000
3		PDL NINJA 2 Uk. 41	Sepatu PDL	Pasang	19		10		7	0	100000
4		PDL NINJA 2 Uk. 42	Sepatu PDL	Pasang	18		10		7	0	100000
5		PDH SUS 04 Scm Uk. 39	Sepatu PDH WANITA	Pasang	25		10		6.25	0	100000
6		PDH SUS 04 Scm Uk. 41	Sepatu PDH WANITA	Pasang	22		10		6.25	0	100000

Gambar 11 Modul Stok Barang

Pada halaman ini (Gambar 11), disajikan sebuah tabel yang berisi beberapa kolom seperti nomor, nama barang, kategori, satuan, stok, item terlaris, waktu tunggu (lead time) maksimum, rata-rata waktu tunggu, stok aman (safety stock), dan harga beli.

5) Modul Rencana Pemesanan Barang

Modul ini berfungsi untuk menampilkan daftar barang yang memerlukan pemesanan ulang ke supplier. Modul ini (Gambar 12), terdapat filter pencarian serta tabel yang memuat berbagai informasi seperti nomor barang, nama barang, kategori, satuan, stok, penjualan tertinggi, waktu tunggu maksimal, rata-rata penjualan, rata-rata waktu tunggu, stok keselamatan, dan jumlah pesanan yang sarankan untuk dipesanan kepada supplier.

No	Item Name	Category	Unit	Stok	Highest	Max Lead	Avg	AvgLead	Safety Stock	Must Order Items
1	Sepatu PDL NINJA 1 Uk. 43	Sepatu PDL	Pieces	-1	5	10	1.5385	7.33333333333333	39	40
2	Sepatu PDL NINJA 2 Uk. 41	Sepatu PDL	Pasang	5	2	10	1.2143	6.6	12	7
3	PDL NINJA 2 Uk. 42	Sepatu PDL	Pasang	4	2	10	1.0714	6.6	13	9
4	PDH SUS 04 Scm Uk. 39	Sepatu PDH WANITA	Pasang	4	2	10	1.3125	6.25	12	8

Gambar 12 Modul Rencana Pemesanan Barang

6) Halaman Perhitungan Reorder Point

Pada Gambar 13, terlihat hasil perhitungan reorder point yang dilakukan oleh sistem. Perhitungan ini didasarkan pada perhitungan manual yang telah dilakukan sebelumnya.

No	Item Name	Unit	Average Sold	Total Stok	Latest Lead Time	Avg Lead Time	Max Lead Time	Lead Time Demand	Safe Stock	ROP
4	PDL NINJA 1 Uk. 41	Pasang	1.2308	5	5 days	6.50 days	10 days	6.15	22	29

Gambar 13 Halaman Reorder Point

Menariknya, hasil perhitungan dari sistem dan perhitungan manual sama persis dan tidak ada perbedaan sedikit pun antara keduanya.

C. Pengujian fungsionalitas aplikasi

Pengujian yang dilakukan pada aplikasi ini menggunakan metode *blackbox testing* dengan menggunakan skenario tes. Hasil pengujian untuk setiap fitur berdasarkan hasil yang diharapkan sudah sesuai. Pengujian dilakukan untuk setiap fitur yang dimiliki dari sistem reorder poin yang sudah dirancang pada tahapan sebelumnya.

TABEL 3
SKENARIO PENGUJIAN

Fitur	Hasil yang diharapkan	Hasil pengujian
<i>Login</i>	Pengguna bisa melakukan login dengan memasukkan username & password yang sudah didaftarkan.	Sesuai
<i>Employee</i>	Pengguna bisa melihat daftar karyawan, menambah, mengubah dan menghapus data karyawan.	Sesuai
<i>Category</i>	Pengguna bisa melihat daftar kategori, menambah, mengubah dan menghapus data kategori.	Sesuai
<i>Unit</i>	Pengguna bisa melihat daftar satuan, menambah, mengubah dan menghapus data satuan.	Sesuai
<i>Item</i>	Pengguna bisa melihat daftar barang, menambah, mengubah dan menghapus data barang.	Sesuai
<i>Supplier</i>	Pengguna bisa melihat daftar supplier, menambah, mengubah dan menghapus data supplier.	Sesuai
<i>Customer</i>	Pengguna bisa melihat daftar pelanggan, menambah, mengubah dan menghapus data pelanggan.	Sesuai
<i>Warehouse</i>	Pengguna bisa melihat daftar gudang, menambah, mengubah dan menghapus data gudang.	Sesuai
<i>Order</i>	Pengguna bisa membuat transaksi order	Sesuai
<i>Item In</i>	Pengguna bisa membuat transaksi barang masuk	Sesuai
<i>Item Out</i>	Pengguna bisa membuat transaksi barang keluar	Sesuai
<i>Dashboard ROP</i>	Pengguna bisa melihat daftar barang beserta perhitungan ROP	Sesuai

Fitur	Hasil yang diharapkan	Hasil pengujian
List of Slow-Moving Item	Pengguna bisa melihat daftar barang yang kurang diminati	Sesuai
List of Fast-Moving Item	Pengguna bisa melihat daftar barang yang banyak diminati	Sesuai
List of Selling Item	Pengguna bisa melihat daftar barang terjual	Sesuai
List of Buying Item	Pengguna bisa melihat daftar barang yang dibeli	Sesuai

D. Pengujian Reorder Point

Pengujian perhitungan *reorder point* dari sistem yang dikembangkan dilakukan dengan mengambil *sample* barang yang akan diuji menggunakan rumus *slavin* pada persamaan nomer 2 dan rumus akurasi untuk menghitung tingkat keakuratan aplikasi. Pengujian ini dilakukan untuk memastikan perhitungan *reorder poin* sudah sesuai dengan hasil perhitungan manual menggunakan google spreadsheet.

$$n = \frac{N}{1+Ne^2} \quad (2)$$

Dimana n adalah jumlah *sample* yang dicari, N adalah jumlah populasi dan e merupakan margin error yang bisa ditoleransi. Jumlah populasi yang diuji sebanyak 30 barang berdasarkan pemesanan dan penjualan 3 bulan terakhir. *Margin error* yang ditoleransi sebesar 5% artinya, tingkat akurasinya mencapai 95%. Hasil perhitungan seperti berikut.

$$n = \frac{30}{1 + (30 \times 0,005^2)}$$

$$n = 27,9069767$$

Hasil perhitungan 27,9069767 dibulatkan keatas menjadi 28. Artinya, 28 barang akan diuji *reorder point* dengan membandingkan hasil perhitungan manual dengan perhitungan aplikasi. Dari hasil pengujian, terdapat 4 barang yang terdapat selisih antara hasil perhitungan manual dengan perhitungan aplikasi. Hasilnya, hasil pengujian dari aplikasi ini memiliki tingkat keakuratan sebesar 85,71%.

IV. KESIMPULAN

Kesimpulan yang didapatkan dari penelitian ini yang pertama pengembangan aplikasi kontrol stok dengan menerapkan metode *reorder poin* terbukti membantu pengguna dalam pengaturan stok agar optimal. Hasil uji fungsionalitas aplikasi dengan metode *blackbox testing* sudah sesuai dengan skenario yang sudah dibuat. Kersimpulan terakhir berdasarkan pengujian *reorder point*, aplikasi ini memiliki tingkat akurasi yang baik mencapai 85,71%.

DAFTAR PUSTAKA

- [1] A. Silberschatz, H. Korth and S. Sudarshan, Database System Concepts, 6th Ed, New York: McGraw-Hill Companies, Inc, 2011.
- [2] M. R. Sahputra, E. Rahayu and N. Nurjamiyah, "Penerapan Metode Reorder Point pada Persediaan Stok Barang Berbasis Website," *Jurnal Ilmiah Teknologi Harapan*, vol. 10, no. 2, pp. 68-74, 2022.
- [3] M. Mahwan, "Penerapan Metode Reorder Point (ROP) dalam Persediaan Sabun Cuci Merk "B-Light" pada UD. Dhofir Jaya di Desa Pemecutan Kaja Kecamatan Denpasar Utara," *Jurnal Ilmiah Akuntansi dan Humanika*, vol. 11, no. 2, pp. 199-205, 2021.
- [4] R. C. Pratiwi, C. Iswayudi and R. Y. Rachmawati, "Sistem Manajemen Persediaan Barang Dagang Menggunakan Metode Safety Stock Dan Reorder Point Berbasis Web (Studi Kasus: Art Kea Centro Plaza Ambarukmo Yogyakarta)," *Jurnal Script*, vol. 7, no. 2, pp. 213-222, 2019.
- [5] K. M. Thalia, E. D. Oktaviyani and F. Sylviana, "Sistem Informasi Inventory Berbasis Website (Studi Kasus : Pada Toko Obyth)," *JOINTECOMS : Journal of Information Technology and Computer Science*, vol. 1, no. 1, pp. 78-86, 2021.
- [6] H. Sutisna and M. Cahyati, "Implementasi Metode ROP Pada Perancangan Sistem Informasi Persediaan Produk Kecantikan pada CV BK Tasikmalaya," *Reputasi: Jurnal Rekayasa Perangkat Lunak*, vol. 2, no. 1, pp. 37-41, 2021.
- [7] A. H. Nobil, A. H. A. Sedigh and L. E. Cárdenas-Barrón, "Reorder point for the EOQ inventory model with imperfect quality items," *Ain Shams Engineering Journal*, vol. 11, no. 4, pp. 1339-1343, 2022.
- [8] J. Akinsola, A. Ogunbanwo, O. Okesola, I. Odun-Ayo, F. Ayegbusi and A. Adebisi, "Comparative Analysis of Software Development Life Cycle Models (SDLC)," in *Intelligent Algorithms in Software Engineering*, 2020.
- [9] L. V. Wijaya and S. R. Ramadhani, "Sistem Informasi Peminjaman Laboratorium pada Cross-Platform dengan Metode Prototyping (Studi Kasus: Politeknik Caltex Riau)," *Journal of Applied Informatics and Computing (JAIC)*, vol. 4, no. 1, pp. 22-27, 2020.
- [10] N. A. Rosa and S. R. Ramadhani, "Pengembangan Aplikasi Mobile Berbasis Android untuk Manajemen Antrian Bimbingan KPdan Proyek Akhir dengan Memanfaatkan Fitur Location Based Service," *Journal of Applied Informatics and Computing (JAIC)*, vol. 6, no. 1, p. 78086, 2022.
- [11] I. F. Ashari, M. F. Zuhdi, M. T. Gagaman and S. T. Denira, "Kolepa Mobile Application Development Based on Android Using SCRUM Method (Case Study: Kolepa Minigolf and Coffe Shop)," *Journal of Applied Informatics and Computing (JAIC)*, vol. 6, no. 1, pp. 104-112, 2022.
- [12] M. Silalahi and S. P. Saragih, "Sistem Informasi Manajemen Lembaga Pendidikan dan Pelatihan Madani (LP2M) dengan Metode Extreme Programming," *Journal of Applied Informatics and Computing (JAIC)*, vol. 3, no. 2, pp. 107-113, 2019.
- [13] M. N. Arifin and D. Siahaan, "Structural and Semantic Similarity Measurement of UML Use Case Diagram," *Lontar Komputer*, vol. 11, no. 2, pp. 88-100, 2022.
- [14] X. Yu, X. Jiao, C. Wang, H. Chen and M. Aloqaily, "Analysis and Design of University Teaching Equipment Management," *Journal of Cyber Security*, vol. 3, no. 3, pp. 177-185, 2021.