

## PENGEMBANGAN SISTEM PERSEDIAAN BERBASIS WEB PADA UMKM

Vinny<sup>1</sup>, Jefri Junifer Pangaribuan<sup>2</sup>, Romindo<sup>3</sup><sup>1,2,3</sup>Sistem Informasi, Universitas Pelita Harapan, Medan, Indonesia  
E-mail: <sup>1</sup>vv70018@student.uph.edu, <sup>2</sup>jefri.pangaribuan@uph.edu, <sup>3</sup>romindo@uph.edu

*Abstract – Inventory management is one of the most important things for a company especially for those who engaged in buying and selling physical products and have many variations and stocks. Without inventory management, there will be chance for the company to experience overstock and stock-out which cause the business process to stop. To be able to control inventory, the min-max stock method will be used and the transaction data from the last 26 days will be used. Researcher decided to use the min-max stock method because to be able to control the inventory level, the minimum and maximum limit is needed to avoid overstock and stock-out. In this research, a web-based system will be planned which can help the company in inventory management and decision making in determining the reorder quantity. This research will produce a system with features such as to manage inventory so that overstock and stock out is avoided and to provide a prediction on re-order quantities for each product.*

**Keywords:** Information Systems, Inventory, Min-Max Method, Website

Abstrak – Manajemen persediaan merupakan salah satu hal yang sangat penting bagi suatu perusahaan terutama yang bergerak di bidang jual beli barang berbentuk fisik dan memiliki variasi serta stok yang cukup banyak. Tanpa adanya manajemen persediaan, maka akan membuka kemungkinan bagi perusahaan untuk memiliki stok yang berlebih atau terbungkalai (*overstock*) serta memungkinkan terjadinya *stock-out* yang dapat menyebabkan terhentinya proses bisnis. Untuk dapat mengontrol persediaan, salah satu metode yang dapat digunakan adalah metode *min-max* dan data yang digunakan adalah data transaksi keluar barang yang terjadi pada 26 hari terakhir. Peneliti memutuskan untuk menggunakan metode *min-max* karena untuk mengontrol level persediaan, diperlukan batas minimum dan maximum agar dapat menghindari *overstock* dan *stock-out*. Pada penelitian kali ini, peneliti akan merancang sebuah sistem berbasis web yang akan membantu perusahaan dalam hal manajemen persediaan dan pengambilan keputusan dalam menentukan jumlah *re-order*. Penelitian ini akan menghasilkan sistem dengan fitur untuk menjaga persediaan agar tidak terjadi *overstock* dan *stock-out* serta menghasilkan prediksi jumlah *re-order* untuk masing-masing produk.

**Kata Kunci:** Sistem Informasi, Manajemen Persediaan, Metode *Min-Max*, Website

## PENDAHULUAN

Manajemen persediaan merupakan suatu hal yang sangat penting bagi suatu perusahaan dan seiring berkembangnya teknologi, diharapkan perusahaan dapat dengan mudah memperoleh data persediaan. Tujuan dilakukannya manajemen persediaan adalah untuk mengetahui seberapa banyak barang yang harus tersedia dan seberapa banyak jumlah suatu barang yang harus dipesan ulang (*re-order*).

Untuk mengetahui persediaan barang, UMKM melakukan perhitungan fisik tiga bulan sekali terhadap persediaan dan laporan yang dihasilkan ditulis oleh karyawan yang bertugas menghitung persediaan. Perhitungan persediaan dilakukan secara kasar dan laporan yang dihasilkan hanya mengandung kode barang dengan jumlah barang yang dituliskan pada selembar kertas HVS. Dalam proses penjualan, pada UMKM terdapat seorang karyawan yang bertugas untuk melakukan pencatatan pengeluaran barang (penjualan) dimana setiap terjadi transaksi penjualan, karyawan tersebut akan mencatat barang apa yang dibeli dan berapa

jumlahnya. Kemudian, catatan tersebut akan digunakan untuk memperbarui buku barang yang ada di gudang. Selain itu, perusahaan ini sering sekali mengalami *stockout* dan *stockout* tersebut baru diketahui ketika seorang customer ingin membeli barang tersebut. Hal ini tentunya akan mengurangi penjualan UMKM dan mengurangi pelayanan yang diberikan.

Selain itu, pemesanan ulang pada UMKM juga masih dilakukan secara manual dan tidak tepat. Oleh karena itu, penulis akan membangun sebuah sistem yang dapat membantu proses pemesanan ulang dengan menggunakan suatu metode *min-max*. Metode *min-max* sangat sering digunakan sebagai metode untuk membangun *software* yang dirancang untuk sistem persediaan otomatis seperti penelitian yang pernah dilakukan untuk mengembangkan sistem informasi persediaan berbasis android [1] dan sistem informasi persediaan baku [2]

Selain untuk menentukan kuantitas pemesanan ulang, metode *min-max* juga dapat digunakan untuk mencegah

## Korespondensi

Jefri Junifer Pangaribuan | [jefri.pangaribuan@uph.edu](mailto:jefri.pangaribuan@uph.edu)

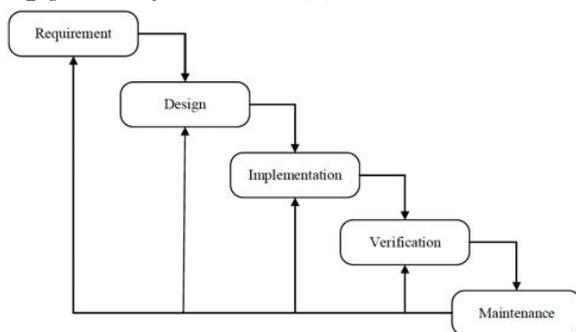
terjadinya *overstock*, *understock* dan *stockout*, penulis akan menggunakan metode *min-max stock* untuk menentukan *safety stock*, *minimum stock*, dan *maximum stock*. Sistem akan memberikan sebuah *alert* jika sebuah barang telah melebihi batas *maximum stock* atau berada di bawah *minimum stock*.

Dari penjelasan diatas, maka diketahui bahwa *inventory management* yang dilakukan oleh UMKM masih dilakukan secara manual dan tidak teratur. Untuk mengatasi masalah tersebut, diperlukan sebuah sistem yang dapat melakukan *inventory management*. Oleh karena itu dikembangkan Sistem Persediaan Berbasis Web. Sistem akan memberikan bantuan untuk menentukan kuantitas pemesanan ulang produk tersebut berdasarkan perhitungan penjualan produk tersebut pada 26 hari terakhir dan sistem akan menjaga *inventory* agar tetap berada di atas *minimum stock*. Untuk dapat memberikan fitur di atas, maka penulis akan menggunakan metode *min-max*.

### METODOLOGI PENELITIAN

Metode pengumpulan data pada penelitian ini dilakukan melalui wawancara, observasi, dan studi pustaka. Wawancara dilakukan secara langsung dengan pihak UMKM untuk memperoleh data mengenai kendala dan diskusi mengenai pengembangan sistem. Observasi dilakukan untuk memperoleh data dan merasakan perbedaan antara memakai sistem dan tidak menggunakan sistem. Sementara itu, studi pustaka dilakukan untuk menunjang pengembangan sistem serta membandingkan penelitian yang sudah dilakukan sebelumnya.

Pengembangan sistem dilakukan dengan menggunakan metode *waterfall*, dan menggunakan metode *min-max* sebagai metode pengendalian persediaan. Metode *waterfall* adalah SDLC yang sering digunakan pada pengembangan sistem informasi, dimana metode ini menggunakan pendekatan yang berurutan dan sistematis. Metode ini merekomendasikan suatu pendekatan pada pengembang perangkat lunak yang sekuensial, bermula dari tingkat kemajuan sistem pada analisis, desain, kode, pengujian, dan pemeliharaan [3].

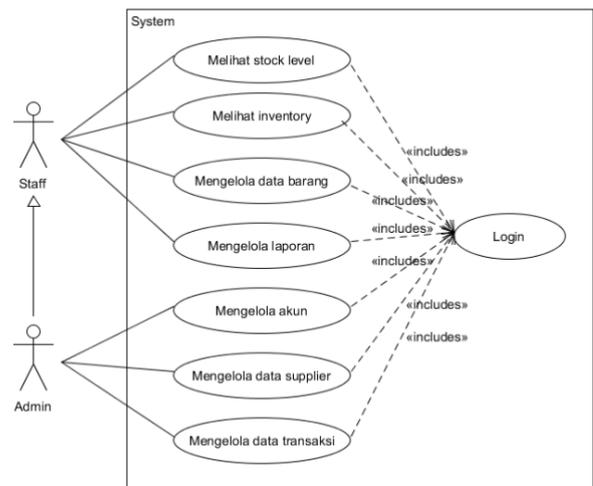


Gambar 1. Metode Waterfall [4]

### HASIL DAN PEMBAHASAN

Penelitian ini menghasilkan sebuah pengembangan sistem informasi persediaan untuk sebuah UMKM

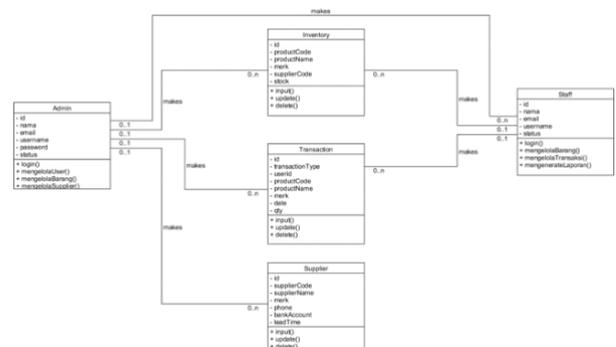
berbasis *web*. Gambar 2 menunjukkan *use case diagram* sistem informasi yang akan dikembangkan.



Gambar 2. Use Case Diagram Sistem yang Dikembangkan

*Use case diagram* merupakan diagram yang mendeskripsikan apa fungsionalitas dan perilaku dari sistem tersebut [5]. Komponen utama dari diagram ini adalah *use case* dan aktor. Dari Gambar 2 di atas, dapat dilihat bahwa *user* dibagi menjadi 2 tipe, yaitu *admin* dan *staff*. Kedua jenis user tersebut dapat melakukan *login*, melihat persediaan dan *stock level*. *Admin* memiliki otoritas khusus yaitu untuk mengelola akun, mengelola data *supplier*, mengelola data barang, mengelola data transaksi dan mengelola laporan. *Staff* dapat mengelola data transaksi namun terbatas hanya untuk 1 hari saja.

Gambar 3 di bawah ini adalah gambar *class diagram*, dimana pada gambar tersebut menunjukkan bahwa terdapat 6 kelas, yaitu *user*, *admin*, *staff*, *inventory*, *transaksi*, dan *supplier* dimana *admin* dan *staff* merupakan *inheritance* dari *user*. *Class diagram* merupakan diagram yang menggambarkan struktur sistem yang menampilkan kelas-kelas yang ada pada sistem tersebut sehingga dapat memberikan gambaran tentang sistem dan relasi apa saja yang ada pada sistem [6].



Gambar 3. Class Diagram Sistem Informasi Persediaan

### Perhitungan Metode Min-Max

Perhitungan menggunakan metode *min-max* yaitu dengan cara menghitung *safety stock*, *minimum stock*, *maximum stock* dan *re-order quantity* setiap produk dari

supplier yang berbeda. Metode *min-max* dilakukan dengan menentukan *safety stock* untuk mencegah risiko kekurangan persediaan [7]. Perhitungan ini hanya akan mengambil beberapa produk yang ada dari beberapa *supplier* yang ada.

#### A. Perhitungan untuk Bingkai 16x24 merk Boss

Berikut adalah tabel penjualan bingkai ukuran 16x24 dengan merk Boss.

Tabel 1. Tabel Penjualan Bingkai 16x24 dengan Merk Boss

NO	Tanggal	Qty (pcs)
1	1/3	20
2	2/3	8
3	8/3	4
4	9/3	4
5	12/3	16
6	16/3	12
7	17/3	4
8	20/3	8
9	22/3	8
10	25/3	8
11	26/3	12
12	29/3	12
<b>Total</b>		116
<b>Rata Rata (hari)</b>		4,46

Waktu pesanan (*Lead Time*) = 4 hari  
Rata-rata pemakaian barang (per hari) = 4  
Pemakaian maksimum = 20

##### 1. Perhitungan *Safety Stock*

$$SS = ( \text{Pemakaian maksimum} - T ) \times LT$$

$$= ( 20 - 4 ) \times 4$$

$$= 64$$

T = Pemakaian barang rata-rata (per hari)

LT = *Lead Time* (hari)

SS = *Safety Stock*

##### 2. Perhitungan *Minimum Stock (ROP)*

$$\text{Minimum (RoP)} = ( T \times LT ) + SS$$

$$= ( 4 \times 4 ) + 64$$

$$= 80$$

##### 3. Perhitungan *Maximum Stock*

$$\text{Maximum} = 2 \times ( T \times LT ) + \text{Safety stock}$$

$$= 2 \times ( 4 \times 4 ) + 64$$

$$= 96$$

##### 4. Perhitungan *Quantity Order*

$$Q = 2 \times T \times LT$$

$$= 2 \times 4 \times 4$$

$$= 32$$

Q = Jumlah pemesanan (unit/pesanan)

Dari hasil perhitungan di atas, dapat disimpulkan:

*Safety Stock*: 64pcs

*Minimum Stock (ROP)*: 80pcs

*Maximum Stock*: 96pcs

*Order Quantity*: 32pcs

#### B. Perhitungan untuk Bingkai 16x24 merk MyFrame

Berikut adalah tabel penjualan bingkai ukuran 16x24 dengan merk MyFrame.

Tabel 2. Tabel Penjualan Bingkai 16x24 dengan merk

MyFrame		
NO	Tanggal	Qty (pcs)
1	9/3	8
2	10/3	16
3	12/3	8
4	20/3	8
5	22/3	8
6	25/3	16
<b>Total</b>		64
<b>Rata Rata (hari)</b>		2,46

Waktu pesanan (*Lead Time*) = 7 hari

Rata-rata pemakaian barang (per hari) = 2

Pemakaian maksimum = 16

##### 1. Perhitungan *Safety Stock*

$$SS = ( \text{Pemakaian maksimum} - T ) \times LT$$

$$= ( 16 - 2 ) \times 7$$

$$= 98$$

##### 2. Perhitungan *Minimum Stock (ROP)*

$$\text{Minimum (RoP)} = ( T \times LT ) + SS$$

$$= ( 2 \times 4 ) + 98$$

$$= 106$$

##### 3. Perhitungan *Maximum Stock*

$$\text{Maximum} = 2 \times ( T \times LT ) + \text{Safety stock}$$

$$= 2 \times ( 2 \times 4 ) + 98$$

$$= 126$$

##### 4. Perhitungan *Quantity Order*

$$Q = 2 \times T \times LT$$

$$= 2 \times 2 \times 7$$

$$= 28$$

Dari hasil perhitungan di atas, dapat disimpulkan:

*Safety Stock*: 98pcs

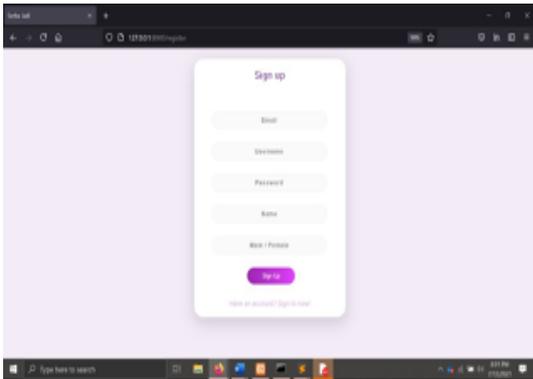
*Minimum Stock (ROP)*: 106pcs

*Maximum Stock*: 126pcs

*Order Quantity*: 28pcs

#### *Interface System*

Halaman *register* berfungsi untuk mendaftarkan *user* baru ke dalam sistem. *User* harus meng-*input* data-data yang diperlukan untuk dapat mendaftarkan dirinya seperti *email*, *password*, nama, *gender* dan memilih status mereka.

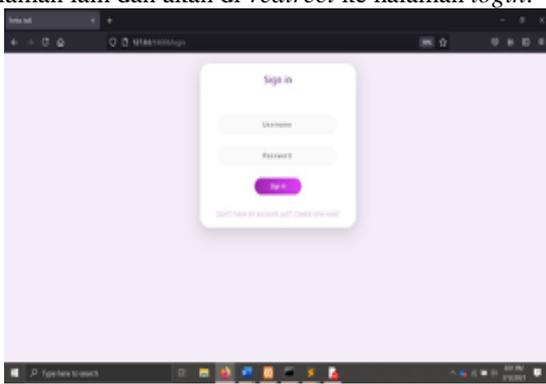


Gambar 4. Interface Halaman Register (Admin)



Gambar 7. Interface Halaman Users (Admin)

Halaman *login* berfungsi untuk memberikan akses sistem kepada *user* yang sudah mempunyai akun. *User* yang belum melakukan *login* tidak akan bisa masuk ke halaman lain dan akan di-*redirect* ke halaman *login*.



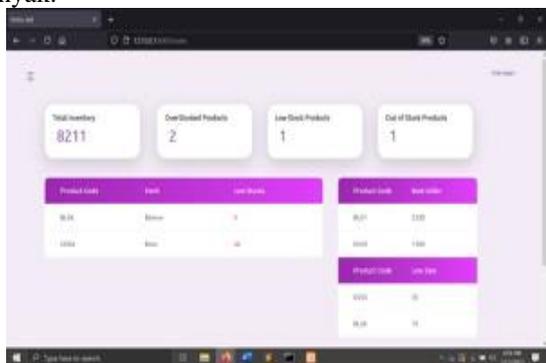
Gambar 5. Interface Halaman Login (Admin)

Untuk menambahkan *user* baru, semua data yang diperlukan pada *form* harus di-*input*. Setelah menekan tombol *submit*, maka *user* baru tersebut akan terdaftar dan data tersebut dapat langsung digunakan untuk *login* ke dalam sistem.



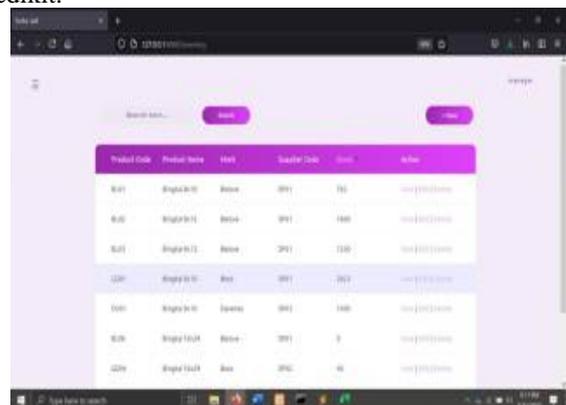
Gambar 8. Interface Halaman Add Users (Admin)

Setelah *user* berhasil *login*, *user* akan dibawa memasuki halaman *dashboard*. Pada halaman *dashboard*, *user* dapat melihat jumlah *inventory*, jumlah barang yang *overstock* dan *low-stock*, serta jumlah barang *out of stock*. Selain itu, *user* juga dapat melihat barang apa saja yang *stock*-nya tinggal sedikit dan barang apa saja yang terjual paling banyak.



Gambar 6. Interface Halaman Dashboard (Admin)

Pada halaman *inventory*, *admin* dapat melihat kode barang, nama barang, merk, kode *supplier* yang memproduksi barang serta *stock* barang tersebut. *Admin* juga dapat melihat *inventory card*, meng-*edit*, dan menghapus sebuah produk. Selain itu, sistem juga dilengkapi oleh fitur *search* sehingga *user* dapat dengan mudah mencari suatu produk tertentu serta fitur *sort by stock* agar *user* dapat mengetahui *stock* terbanyak dan sedikit.



Gambar 9. Interface Halaman Inventory (Admin)

Pada halaman *user*, *admin* dapat melihat akun yang terdaftar pada sistem serta dapat menambahkan *user* baru serta meng-*edit* maupun menghapus data *user* yang sudah terdaftar.

Pada Gambar 10 adalah form untuk menambahkan jenis produk baru. *Admin* perlu mengisi kode produk dan nama

produk serta memilih merk produk sesuai dengan yang terdaftar pada tabel *supplier*. *Stock* produk baru yang terdaftar adalah 0 dan untuk menambahkannya harus melalui form transaksi.



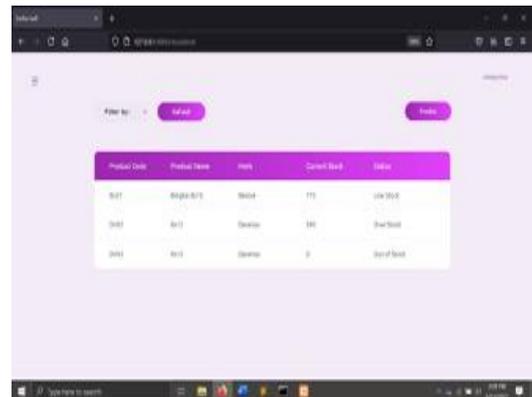
Gambar 10. Interface Halaman Form Add Item (Admin)

Pada halaman ini, *admin* dapat melihat *supplier* yang terdaftar dalam sistem. Data yang dapat dilihat adalah kode *supplier*, merk, nomor telepon, nomor rekening, serta hari yang diperlukan *supplier* untuk mengirimkan produk sejak dipesan (*Lead Time*). *Admin* juga dapat menambahkan *supplier* baru serta mengedit dan menghapus data *supplier* yang sudah ada.



Gambar 11. Interface Halaman Suppliers (Admin)

Halaman *Stock Level* merupakan halaman yang dapat diakses oleh semua *user*. Pada halaman ini, sistem akan menampilkan data barang dengan *stock* yang memiliki status *Low*, *Over*, dan *Out of Stock*. Pada Gambar 12, pada bagian atas terdapat tombol *predict*, fitur *filter* dan tombol *refresh*. *Filter* berfungsi untuk mem-*filter* data sesuai level *stock* dan tombol *refresh* berfungsi untuk me-*reset filter*. Tombol *predict* yang berfungsi untuk memprediksi jumlah *re-order* barang yang memiliki status *Low Stock* dan *Out of Stock*. Perhitungan dilakukan dengan menggunakan data 26 hari terakhir dari tanggal hari tersebut.



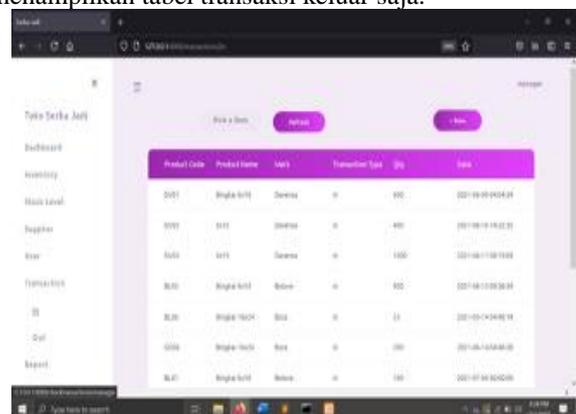
Gambar 12. Interface Halaman Stock Levels (Admin)

*Admin* dapat melakukan prediksi dan dapat men-*download* hasil prediksi dalam bentuk pdf untuk dicetak.



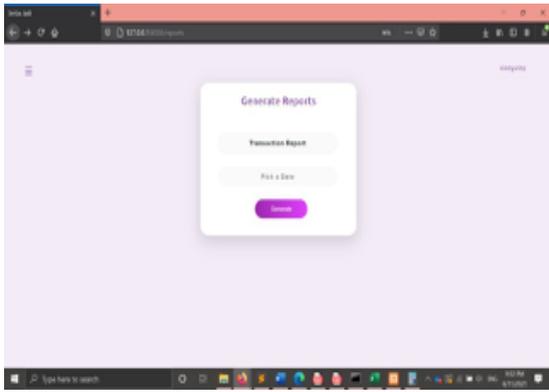
Gambar 13. Interface Hasil Prediksi (Admin)

Halaman *Transaction* merupakan halaman yang dapat diakses oleh semua jenis *user* namun akan memiliki tampilan yang berbeda. Kedua *user* dapat menambahkan transaksi baru. Halaman *transaction* ini dibagi menjadi 2 halaman yaitu *in* dan *out*. Untuk halaman *in* akan menampilkan tabel yang berisi hanya transaksi masuk saja dan sebaliknya dengan *out* yang akan hanya menampilkan tabel transaksi keluar saja.



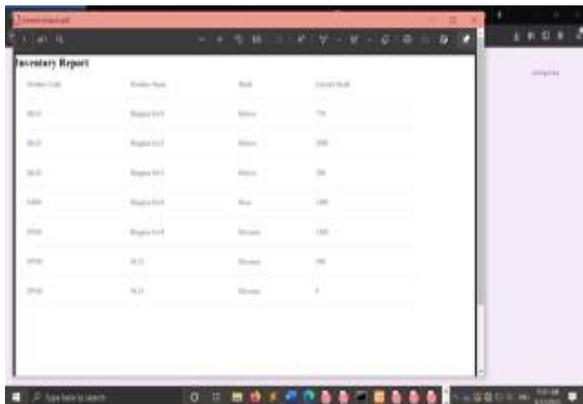
Gambar 14. Interface Halaman Transaction (Admin)

*Admin* dapat men-*generate* laporan *inventory* serta laporan transaksi. Untuk laporan transaksi, *user* dapat memilih untuk men-*generate* transaksi yang terjadi pada tanggal hari ini maupun sebelumnya.



Gambar 15. Interface Halaman *Reports – Choose Transaction Report (Admin)*

Gambar 16 menunjukkan hasil *download* laporan persediaan.



Gambar 16. Interface Hasil *Download Inventory Report (Admin)*

### Hasil Black-Box Testing

Berikut adalah hasil dari *black-box testing* yang telah dilakukan. Testing dilakukan oleh karyawan yang bekerja di UMKM tempat objek penelitian.

Tabel 3. Tabel Hasil *Black-Box Testing*

Skenario Pengujian	Kasus Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
<i>Login</i>	Isi <i>field username</i> dan <i>password</i> kemudian klik " <i>Login</i> "	Sistem memberikan akses <i>login</i> dan menampilkan <i>dashboard</i>	Sesuai	Normal
Tambah data barang	Masukkan semua data barang kemudian klik " <i>Add Item</i> "	Data barang berhasil ditambah dan disimpan dalam <i>database</i>	Sesuai	Normal

Tambah data <i>user (admin)</i>	Masukkan semua data <i>user</i> kemudian klik " <i>Add User</i> "	Data <i>user</i> berhasil ditambah dan disimpan dalam <i>database</i>	Sesuai	Normal
Tambah data transaksi	Masukkan semua data transaksi kemudian klik " <i>Add Transaction</i> "	Data transaksi berhasil ditambah dan disimpan dalam <i>database</i>	Sesuai	Normal
Edit data barang	Klik tombol <i>edit</i> pada barang yang ingin diubah dan ubah data lalu klik " <i>Update</i> "	Data berhasil diperbarui	Sesuai	Normal
Edit data <i>user (admin)</i>	Klik tombol <i>edit</i> pada <i>user</i> yang ingin diubah dan ubah data lalu klik " <i>Update</i> "	Data berhasil diperbarui	Sesuai	Normal
Hapus data barang	Klik tombol hapus pada barang yang ingin dihapus	Data berhasil dihapus	Sesuai	Normal
Hapus data <i>user (admin)</i>	Klik tombol hapus pada <i>user</i> yang ingin dihapus	Data berhasil dihapus	Sesuai	Normal
Mencari barang dengan <i>search bar</i>	Masukkan nama barang yang ingin dicari dan klik " <i>Search</i> "	Menampilkan barang dengan nama sesuai yang diketik oleh <i>user</i> pada <i>search bar</i>	Sesuai	Normal
Menampilkan data transaksi pada tanggal tertentu ( <i>admin</i> )	Memilih tanggal yang ingin ditampilkan	Menampilkan data transaksi sesuai dengan tanggal yang telah dipilih	Sesuai	Normal

Mendownload laporan <i>inventory</i>	Memilih pilihan "Inventory" dan klik tombol "Generate"	File pdf yang berisi laporan <i>inventory</i> berhasil didownload	Sesuai	Normal
Mendownload laporan transaksi	Memilih pilihan "Transaction" dan klik tombol "Generate"	File pdf yang berisi laporan <i>transaction</i> berhasil didownload	Sesuai	Normal
Mengakses halaman <i>stock level</i>	Memilih halaman "Stock Level" pada bagian menu navigasi	Menampilkan halaman <i>stock level</i> dan <i>status level</i> barang	Sesuai	Normal
Menampilkan barang dengan <i>stock level</i> tertentu	Pada bagian <i>filter</i> , pilih <i>level</i> yang diinginkan	Menampilkan barang dengan <i>level</i> yang telah dipilih <i>user</i>	Sesuai	Normal
Memprediksi jumlah yang harus dipesan ulang	Klik tombol "Predict" pada halaman <i>Stock Level</i>	Menampilkan tabel berisi jumlah pemesanan kembali	Sesuai	Normal
Mengurutkan <i>inventory</i> dari persediaan terbanyak ke tersedikit	Klik judul tabel "Qty" pada halaman <i>Inventory</i>	Mengurutkan barang sesuai "qty" dari yang tertinggi hingga terendah	Sesuai	Normal

## KESIMPULAN

Dari proses perancangan dan analisis sistem informasi ini, maka dapat disimpulkan bahwa:

1. Dengan menerapkan sistem *inventory* ini, proses *inventory management* perusahaan akan sangat terbantu karena sistem ini akan menjaga agar *stock* berada pada titik yang telah ditentukan (*minimum stock*, *safety stock*, dan *maximum stock*) dari perhitungan sehingga mencegah terjadinya *stock out* dan *overstock* dan proses bisnis dapat tetap berjalan.
2. Dengan menerapkan sistem *inventory* ini, perusahaan tidak perlu memeriksa dan menghitung *stock* yang ada secara manual karena sistem ini akan memberikan informasi *stock* secara *real-time*.
3. Sistem ini dilengkapi fitur untuk menghitung jumlah *re-order* jika *stock* mencapai titik *minimum*. Untuk mendapatkan jumlah *re-order*, sistem akan

melakukan perhitungan dengan menggunakan metode *min-max* dan dengan data yang tersedia 26 hari terakhir. Dengan demikian, perusahaan akan lebih mudah dalam menentukan jumlah yang harus dipesan kembali dari *supplier* jika *stock* berada di titik *minimum* atau *re-order point*.

4. Sistem juga dapat memberikan laporan mengenai *stock inventory* yang ada pada saat tersebut, laporan transaksi keluar masuk barang, dan laporan *inventory* suatu barang tertentu (*inventory card*).

Saran penelitian selanjutnya, penulis mengusulkan untuk tidak berhenti pada sistem persediaan, tetapi dapat mengembangkan ke sistem penjualan dan pembelian, seperti penelitian yang pernah dilakukan sebelumnya [8], atau dapat membuat sistem *Point of Sales* yang dapat digunakan ketika melakukan transaksi [9].

## DAFTAR PUSTAKA

- [1] P. Sari, A. Oklilas, and I. A., "Implementasi Metode Min-Max Stock Pada Sistem Informasi Persediaan Berbasis Android," *Jurnal Nasional Teknologi dan Sistem Informasi*, vol. 8, pp. 17–24, May 2022, doi: 10.25077/TEKNOSI.v8i1.2022.17-24.
- [2] A. P. Kinanthi, D. Herlina, and F. A. Mahardika, "Analisis Pengendalian Persediaan Bahan Baku Menggunakan Metode Min-Max (Studi Kasus PT. Djitoe Indonesia Tobacco)," *Performa: Media Ilmiah Teknik Industri*, vol. 15, no. 2, pp. 87–92, 2016.
- [3] R. Susanto and A. D. Andriana, "Perbandingan Model Waterfall Dan Prototyping Untuk Pengembangan Sistem Informasi," *Majalah Ilmiah UNIKOM*, vol. 14, no. 1, 2016, doi: 10.34010/miu.v14i1.174.
- [4] I. Pratiwi, S. Anardani, and A. R. Putera, "Rancang Bangun Sistem Informasi Penjadwalan Mata Pelajaran Dengan Metode Waterfall," *JDMIS: Journal of Data Mining and Information Systems*, vol. 1, no. 1, pp. 20–28, 2023, doi: 10.54259/jdmis.
- [5] Erlanie Sufarnap, Mirza Ilhami, and Jefri Junifer Pangaribuan, "Analisis dan Perancangan Sistem Informasi Penjualan pada Toko XYZ," *SATESI: Jurnal Sains Teknologi dan Sistem Informasi*, vol. 2, no. 2, pp. 170–176, Oct. 2022, doi: 10.54259/satesi.v2i2.1181.
- [6] D. S. Saputra, *Pengantar Sistem Informasi*, vol. 150, no. 1. Yogyakarta: Penerbit Andi, 2017. [Online]. Available: [https://www.google.co.id/books/edition/Pengantar\\_Sistem\\_Informasi/8VNLDwAAQBAJ?hl=id&gbpv=1](https://www.google.co.id/books/edition/Pengantar_Sistem_Informasi/8VNLDwAAQBAJ?hl=id&gbpv=1)
- [7] N. L. Rachmawati and M. Lentari, "Penerapan Metode Min-Max untuk Minimasi Stockout dan Overstock Persediaan Bahan Baku," *Jurnal INTECH Teknik Industri Universitas Serang Raya*, vol. 8, no. 2, pp. 143–148, Oct. 2022, doi: 10.30656/intech.v8i2.4735.

- [8] J. J. Pangaribuan, H. Margono, O. P. Barus, Y. A. Pratama, and A. Maulana, "Sales, Purchase, and Inventory Information System Design at SMEs," in *1st International Conference on Technology Innovation and Its Applications (ICTIIA)*, IEEE, Sep. 2022, pp. 1–6. doi: 10.1109/ictiia54654.2022.9935929.
- [9] O. Putra Barus, P. Jefri Junifer, P. Yudhistira Adhitya, M. Ade, and N. Faisal, "Peningkatan Kemampuan Pengelolaan Transaksi Melalui Implementasi Sistem Informasi PoS Untuk Para Peternak Arjuna Farm, Deli Serdang," *Jurnal Pengabdian kepada Masyarakat Nusantara (JPkMN)*, vol. 2, no. 2, pp. 110–118, Mar. 2022, doi: <https://doi.org/10.55338/jpkmn.v2i2.293>.